

Datei: 2014\_10\_12\_000000\_create\_users\_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

Datei: 2014\_10\_12\_100000\_create\_password\_resets\_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreatePasswordResetsTable extends Migration
{
    /**
     * Run the migrations.
     *
     */
```

```

* @return void
*/
public function up()
{
    Schema::create('password_resets', function (Blueprint $table) {
        $table->string('email')->index();
        $table->string('token');
        $table->timestamp('created_at')->nullable();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('password_resets');
}
}

```

Datei: 2019\_08\_19\_000000\_create\_failed\_jobs\_table.php

```
<?php
```

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

```

```

class CreateFailedJobsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('failed_jobs', function (Blueprint $table) {
            $table->id();
            $table->text('connection');
            $table->text('queue');
            $table->longText('payload');
            $table->longText('exception');
            $table->timestamp('failed_at')->useCurrent();
        });
    }

    /**
     * Reverse the migrations.
     *

```

```

    * @return void
    */
    public function down()
    {
        Schema::dropIfExists('failed_jobs');
    }
}

```

Datei: AccountController.php

```
<?php
```

```

namespace App\Http\Controllers;

use App\Http\Requests\RegisterRequest;
use App\Http\Resources\UserResource;
use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Auth;

use Propaganistas\LaravelPhone\PhoneNumber;

class AccountController extends Controller
{
    public function init(Request $request) {

        $user = $request->user();

        if (!$user) {
            $token = $request->input('token', null);
            $token = Str::isUuid($token) ? $token : null;

            if ($token) {
                $user = User::whereUniqueId($token)->first();
            }

            if (!$token || !$user) {
                $user = User::create();
            }

            Auth::login($user, true);
        } else {
            $request->session()->regenerate();
        }

        return new UserResource($user);
    }

    public function directContactConsent(Request $request) {
        $user = $request->user();
    }
}

```

```

    $user->direct_contact_consent = $request->gives_consent;

    $user->save();

    return new UserResource($user);
}

public function register(RegisterRequest $request) {
    $user = $request->user();

    $telephone = $request->input('telephone') ? PhoneNumber::make($request->telephone, 'DE')->formatNational() : null;

    $user->first_name = $request->input('first_name');
    $user->last_name = $request->input('last_name');
    $user->email = $request->input('email');
    $user->telephone = $telephone;

    $user->unique_id = null;

    $user->save();

    return new UserResource($user);
}
}

```

Datei: Add.vue

```

<template>
  <Page ref="page" title="Besuch einfügen" v-bind="{ ...$props, ...$attrs }">
    <!-- <Page ref="page" title="Besuch einfügen" extension-height="96px" v-bind="{ ...$props, ...$attrs }"> -->
    <!-- <template #extension>
      <div class="d-flex justify-space-around" style="width: 100%">
        <v-btn
          icon
          x-large
          light
          color="primary"
          style="background-color: white; border-color: rgb(222 222 222); pointer-events: none"
          v-for="icon in ['masks', '6_ft_apart', 'clean_hands']"
          :key="icon"
        >
          <v-icon :class="{ 'hide-icon': icon === '6_ft_apart' }">{{ icon }}</v-icon>
        </v-btn>
      </div>
    </template> -->
    <v-sheet color="primary" class="d-flex justify-space-around align-center" height="96px" >
      <div class="extended-container" />
        <v-btn
          icon
          x-large
          light

```



```

        tabindex="-1"
        color="primary"
        style="background-color: white; border-color: rgb(222 222 222); pointer-events: none"
        v-for="icon in ['masks', '6_ft_apart', 'clean_hands']"
        :key="icon"
    >
    <v-icon :class="{ 'hide-icon': icon === '6_ft_apart' }">{{ icon }}</v-icon>
    </v-btn>
</v-sheet>

<v-list>
<!-- <v-list style="padding-top: 152px"> -->
    <div class="body-1 text--secondary ma-6">
        Bitte tragen Sie eine Maske. Halten Sie Abstand zu anderen. Waschen Sie Ihre Hände
        regelmäßig.
    </div>
    <v-form v-model="valid" ref="form" @submit.prevent="submit">
        <RoomItem v-model="room" />

        <v-expand-transition>
            <DateTimeItem v-model="datetime" v-if="visit_in_past" />
        </v-expand-transition>

        <v-fade-transition>
            <v-list-item class="px-3 mt-3" v-if="!visit_in_past">
                <v-btn
                    outlined
                    rounded
                    block
                    color="secondary lighten-3"
                    class="body-2"
                    style="border-color: #ccc"
                    @click="visit_in_past = true"
                >
                    Datum/Uhrzeit eingeben
                    <v-icon right>keyboard_arrow_down</v-icon>
                </v-btn>
            </v-list-item>
        </v-fade-transition>

    </v-form>
</v-list>

<v-spacer></v-spacer>

<v-card-actions>
    <LoadingButton
        :query="promise" @click="submitForm" />
</v-card-actions>

<PrivacyFooter />
</Page>
</template>

```

```

<script>
import RoomItem from "@components/add/RoomItem";
import DateTimeItem from "@components/add/DateTimeItem";

import { setVisit } from "@requests/api";

export default {
  data: () => ({
    room: null,
    valid: false,
    datetime: {}},
  visit_in_past: false,
  promise: null
  }),

  props: {
    query: [String, Array]
  },

  async created() {

    if (this.query) {
      // If scanning through QR-Code, be sure, that the room list has been loaded
      this.getRooms().then(() => {
        // Check if query is an array, use the first
        const query = Array.isArray(this.query) ? this.query[0] : this.query;

        this.room = this.rooms.find(room => {
          const room_without_name = room.split(' ')[0];

          return room === query
            || query.endsWith(room_without_name);
        });
      });
    }

    if (this.room !== null) {
      this.$router.replace({ ...this.$router.currentRoute, query: null });
    }
  },

  methods: {
    submitForm() {
      console.log("submitForm()");
      // this.$refs.form.submit();
      this.submit();
    },
    async submit() {

      console.log("async submit()");

      this.$refs.form.validate();
    }
  }
}

```

```

    await this.$nextTick();

    if (!this.valid) return;

    this.promise = setVisit(this.room, this.datetime.visited_on, this.datetime.exited_on)
      .then(() => this.$refs.page.close());
  },
  },

  components: {
    RoomItem,
    DateTimeItem
  },
};
</script>

```

```

<style lang="scss" scoped>
.hide-icon {
  margin-top: 8px;

  &::before {
    content: "";
    background-color: white;
    width: 20px;
    height: 10px;
    position: absolute;
    bottom: 0;
  }
}

.extended-container {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  height: 160px;
  background: var(--v-primary-base);
}
</style>

```

Datei: AddButton.vue

```

<template>

  <v-speed-dial
    v-if="actions.length > 1"
    v-model="fab"

    fixed
    right
    bottom
    class="mb-3"
    :class="{ 'expanded-speed-dial': !isMobile }"
  >

```

```

transition="slide-y-reverse-transition"
>
<template v-slot:activator>
  <v-overlay :value="fab" z-index="-1" />
  <v-btn
    v-model="fab"
    ref="fab"
    color="accent"
    class="elevation-6"
    aria-label="Besuch einfügen"
    dark
    large
    :fab="isMobile"
    :rounded="!isMobile"
  >
    <v-icon :left="!isMobile">{{ fab ? 'close' : 'add' }}</v-icon>

    <span v-if="!isMobile">
      Besuch einfügen
    </span>
  </v-btn>
</template>

<div
  v-for="action in actions"
  :key="action.icon"
  style="position:relative" @click="$router.push({ name: action.route })">
  <v-chip
    label
    color="background"
    class="elevation-2 speed-dial__tooltip subtitle-1">
    {{ action.text }}
  </v-chip>
  <v-btn fab color="background" :aria-label="action.text">
    <v-icon style="font-size: 28px" color="secondary">
      {{ action.icon }}
    </v-icon>
  </v-btn>
</div>

</v-speed-dial>

<v-btn
  v-else
  aria-label="Besuch einfügen"
  large
  fixed
  right
  bottom
  ref="fab"
  to="new"
  color="accent"
  style="z-index: 10"
  class="mb-3 elevation-6"

```

```

    :fab="isMobile"
    :rounded="!isMobile"
  >
    <v-icon :left="!isMobile">{{ fab ? 'close' : 'add' }}</v-icon>

    <span v-if="!isMobile">
      Besuch einfügen
    </span>
  </v-btn>

</template>

<script>
export default {
  data: () => ({
    fab: false,
    canScanBarcodes: false,
    actions: []
  }),

  created() {
    this.canScanBarcodes = /MobilAndroid/i.test(navigator.userAgent);
  },

  mounted() {
    // Sequence is important
    if (this.suggested && this.suggested.length) {
      this.actions.push({
        route: 'suggested',
        text: 'Meistbesucht',
        icon: 'star_outline'
      });
    }

    this.actions.push({
      route: 'add',
      text: 'Manuelle Eingabe',
      icon: 'post_add'
    });

    if (this.canScanBarcodes) {
      this.actions.push({
        route: 'scanner',
        text: 'QR Code scannen',
        icon: 'qr_code_scanner'
      });
    }
  },

  watch: {
    fab() {
      const pulse = this.$refs.fab.$el.querySelector('.pulse');

      if (pulse) pulse.className = "";
    }
  }
}

```

```

    // if (this.actions.length > 1) {
    //   document.querySelector('html').classList.toggle('overflow-y-hidden', this.fab);
    // }
  },
  suggested() {
    if (this.actions.find(a => a.route === 'suggested')) return;

    if (this.suggested && this.suggested.length) {

      this.actions.unshift({
        route: 'suggested',
        text: 'Meistbesucht',
        icon: 'star_outline'
      });
    }
  },
  visits() {
    if (this.visits.length > 0) return;
    const hasPulse = this.$refs.fab.$el.querySelector('.pulse');
    if (hasPulse) return;

    const pulse = document.createElement('div');
    pulse.className = "pulse accent--text";
    pulse.setAttribute(this.$options._scopeId, "");

    this.$refs.fab.$el.appendChild(pulse);
  }
}
</script>

```

```

<style lang="scss" scoped>
::v-deep {
  & .speed-dial__tooltip {
    position: absolute;
    right: 80px;
    margin: 18px 0;
    max-width: inherit;
  }

  @media (prefers-color-scheme: dark) {
    .v-overlay--active.theme--dark .v-overlay__scrim {
      background-color: black !important;
      opacity: 0.6 !important;
    }
  }
}

```

```

.expanded-speed-dial::v-deep .v-speed-dial__list {
  align-items: flex-end;
}

```

```

.pulse {

```

```

content: "";
  animation: pulse 2s ease-out 3s 5;
  height: 100%;
  width: 100%;
  border-radius: 100px;
position: absolute;
}

@keyframes pulse {
  0% {
    box-shadow: 0 0 0 0 currentColor;
    opacity: 0.4;
  }
  70% {
    box-shadow: 0 0 0 100px currentColor;
    opacity: 0;
  }
  100% {
    box-shadow: 0 0 0 0 currentColor;
    opacity: 0;
  }
}
</style>

```

Datei: App.vue

```

<template>
  <v-app id="main">
    <v-main class="background">
      <router-view></router-view>

      <v-snackbar v-model="snackbar" top color="error" timeout="10000">

        <template v-if="globalError">
          <span v-html="globalError.message || globalError"></span>
        </template>

        <template v-slot:action="{ attrs }">
          <template v-if="!globalError"></template>

          <v-btn v-else-if="globalError.action === 'reload'" text v-bind="attrs"
@click="reload">Neuladen</v-btn>

          <v-btn v-else text v-bind="attrs" @click="snackbar = false">Close</v-btn>
        </template>
      </v-snackbar>

      <v-snackbar
top
vertical
timeout="-1"
color="info"
:value="updateExists"
>

```

```

        <span>Ein Update ist verfügbar</span>
            <template v-slot:action="{ attrs }">
                <v-btn
                    text
                    rounded
                    v-bind="attrs"
                    class="px-4"
                    @click="reload">Aktualisieren</v-btn>
            </template>
        </v-snackbar>
    </v-main>
</v-app>
</template>

<script>
// import axios from "@requests/connection";

import themeMixin from "@mixins/themeMixin"
import backgroundRecolorMixin from "@mixins/backgroundRecolorMixin"

export default {
  mixins: [themeMixin, backgroundRecolorMixin],
  data: () => ({
    updateExists: false,
  }),

  created() {
    navigator.serviceWorker.addEventListener('controllerchange', () => {
      this.updateExists = true
    });
  },

  methods: {
    reload() {
      window.location.reload();
    },
  },

  computed: {
    snackbar: {
      get() {
        return !!this.globalError;
      },
      set(error) {
        this.setError(error);
      }
    }
  },
};
</script>
<style>

html {

```



```
/* background-color: var(--v-background-base); */
background-color: var(--v-primary-base);
-ms-overflow-style: none; /* IE and Edge */
scrollbar-width: none; /* Firefox */
overscroll-behavior-x: none;
}

html.scrolled {
  background-color: var(--v-background-darken2) !important;
}

html::-webkit-scrollbar {
  display: none;
}

body {
  -webkit-user-select: none;
  -webkit-touch-callout: none;
  -webkit-tap-highlight-color: transparent;
}
</style>
```

```
<style lang="scss" scoped>
@import './node_modules/typeface-roboto/index.css';
```

```
#app {
  font-family: "Avenir", Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  // color: #2c3e50;
}
```

```
::v-deep {
  .opacity-25 {
    opacity: 0.25;
  }

  .opacity-50 {
    opacity: 0.5;
  }

  .opacity-75 {
    opacity: 0.75;
  }

  .strike-through > *:not(.dont-strike-through) {
    text-decoration: line-through;
  }

  .dont-strike-through {
    text-decoration: initial;
  }

  @for $i from 1 through 5 {
```

```

        .row__column#{$i} {
            flex-basis: $i * 32px;
            flex-shrink: 1;
            flex-grow: 0;
        }
    }
}

.background {
    background-color: var(--v-background-darken2) !important;
}

</style>

```

Datei: AppServiceProvider.php

```

<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;

class AppServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     *
     * @return void
     */
    public function register()
    {
        //
    }

    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {
        //
    }
}

```

Datei: Application.vue

```

<template>
  <v-container fill-height fluid class="px-8 py-6 background flex-grow-1">
    <Drawer />

```

```

    <router-view></router-view>
  </v-container>
</template>

<script>
import Drawer from "@components/drawer/Drawer";
import { mapState } from 'vuex';

export default {
  data: () => ({}),

  computed: {
    ...mapState('userConfig', {
      active_role: state => state.active_role
    }),
  },

  beforeRouteUpdate (to, from, next) {
    // called when the route that renders this component has changed,
    // but this component is reused in the new route.
    // For example, for a route with dynamic params `foo:id`, when we
    // navigate between `foo/1` and `foo/2`, the same `Foo` component instance
    // will be reused, and this hook will be called when that happens.
    // has access to `this` component instance.

    console.log("Before route update application");

    next();
  },

  components: {
    Drawer
  }
};
</script>

```

Datei: AuthBasicSearchApi.php

```

<?php

namespace App\Http\Middleware;

use Closure;

class AuthBasicSearchApi
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
    */

```

```

*/
public function handle($request, Closure $next)
{
    if ($request->ip() != Config('api.search_ip')) return response('Unauthorized', 401);

    $sapi_username = Config('api.search_username');
    $sapi_password = Config('api.search_password');

    if (!($sapi_username && $sapi_password)) {
        return response('Unauthorized', 401);
    }

    if ($request->getUser() !== $sapi_username || $request->getPassword() !== $sapi_password) {
        $headers = array('WWW-Authenticate' => 'Basic');
        return response('Unauthorized', 401, $headers);
    }

    return $next($request);
}
}
}

```

Datei: AuthServiceProvider.php

```
<?php
```

```
namespace App\Providers;
```

```
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
use Illuminate\Support\Facades\Gate;
```

```
class AuthServiceProvider extends ServiceProvider
{
    /**
     * The policy mappings for the application.
     *
     * @var array
     */
    protected $policies = [
        // 'App\Model' => 'App\Policies\ModelPolicy',
    ];

    /**
     * Register any authentication / authorization services.
     *
     * @return void
     */
    public function boot()
    {
        $this->registerPolicies();

        //
    }
}

```

Datei: AuthUser.php

```
<?php
```

```
namespace App\Helpers;
```

```
class AuthUser {
    public $ldap_id;
    public $first_name;
    public $last_name;
    public $email;

    public function __construct($ldap_id, $first_name, $last_name, $email)
    {
        $this->ldap_id = $ldap_id;
        $this->first_name = $first_name;
        $this->last_name = $last_name;
        $this->email = $email;
    }
}
```

Datei: Authenticate.php

```
<?php
```

```
namespace App\Http\Middleware;
```

```
use Illuminate\Auth\Middleware\Authenticate as Middleware;
```

```
class Authenticate extends Middleware
{
    /**
     * Get the path the user should be redirected to when they are not authenticated.
     *
     * @param \Illuminate\Http\Request $request
     * @return string|null
     */
    protected function redirectTo($request)
    {
        if (!$request->expectsJson()) {
            return route('login');
        }
    }
}
```

Datei: BasicEnum.php

```
<?php
```

```

namespace App\Utils;

use ReflectionClass;

abstract class BasicEnum {
    private static $constCacheArray = NULL;

    public static function getConstants() {
        if (self::$constCacheArray == NULL) {
            self::$constCacheArray = [];
        }
        $calledClass = get_called_class();
        if (!array_key_exists($calledClass, self::$constCacheArray)) {
            $reflect = new ReflectionClass($calledClass);
            self::$constCacheArray[$calledClass] = $reflect->getConstants();
        }
        return self::$constCacheArray[$calledClass];
    }

    public static function allValues() {
        return array_values(self::getConstants());
    }

    public static function isValidName($name, $strict = false) {
        $constants = self::getConstants();

        if ($strict) {
            return array_key_exists($name, $constants);
        }

        $keys = array_map('strtolower', array_keys($constants));
        return in_array(strtolower($name), $keys);
    }

    public static function isValidValue($value, $strict = true) {
        $values = array_values(self::getConstants());
        return in_array($value, $values, $strict);
    }
}

```

Datei: BroadcastServiceProvider.php

```

<?php

namespace App\Providers;

use Illuminate\Support\Facades\Broadcast;
use Illuminate\Support\ServiceProvider;

class BroadcastServiceProvider extends ServiceProvider
{
    /**

```

```

    * Bootstrap any application services.
    *
    * @return void
    */
    public function boot()
    {
        Broadcast::routes();

        require base_path('routes/channels.php');
    }
}

```

Datei: CheckForMaintenanceMode.php

```

<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode as Middleware;

class CheckForMaintenanceMode extends Middleware
{
    /**
     * The URIs that should be reachable while maintenance mode is enabled.
     *
     * @var array
     */
    protected $except = [
        //
    ];
}

```

Datei: ClearInactiveAccounts.php

```

<?php

namespace App\Console\Commands;

use App\User;
use Carbon\Carbon;
use Illuminate\Console\Command;

class ClearInactiveAccounts extends Command
{
    /**
     * The name and signature of the console command.
     *
     * @var string
     */
    protected $signature = 'users:clear';
}

```

```

/**
 * The console command description.
 *
 * @var string
 */
protected $description = 'Clear inactive accounts';

/**
 * Create a new command instance.
 *
 * @return void
 */
public function __construct()
{
    parent::__construct();
}

/**
 * Execute the console command.
 *
 * @return int
 */
public function handle()
{
    $weeks_ago = Carbon::parse('-4 weeks');

    return User::whereNull('first_name')
        ->whereNull('last_name')
        ->whereDate('created_at', '<', $weeks_ago)
        ->delete();
}
}

```

Date: ClearOldVisits.php

```

<?php

namespace App\Console\Commands;

use App\Visit;
use Carbon\Carbon;
use Illuminate\Console\Command;

class ClearOldVisits extends Command
{
    /**
     * The name and signature of the console command.
     *
     * @var string
     */
    protected $signature = 'visits:clear';

```



```

/**
 * The console command description.
 *
 * @var string
 */
protected $description = 'Clear old visits after the defined Corona time';

/**
 * Create a new command instance.
 *
 * @return void
 */
public function __construct()
{
    parent::__construct();
}

/**
 * Execute the console command.
 *
 * @return int
 */
public function handle()
{
    $weeks_ago = Carbon::parse('-4 weeks');

    return Visit::whereDate('visited_on', '<', $weeks_ago)->delete();
}
}

```

Datei: Controller.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;

class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
}

```

Datei: CurrentVisitDirect.vue

```

<template>

```

```

<v-card>
  <div class="overline px-4 pt-2">Gesamtkontakte im aktuellen Raum</div>

<v-timeline align-top dense class="body-2 text--secondary flex-grow-1 ml-n3 pt-0 mt-3 visit">
  <v-timeline-item
    small
    fill-dot
    class="py-2"
    color="info lighten-1"
  >
    <template #icon>
      <v-icon small color="background">face</v-icon>
      <div class="pulse" />
    </template>

    <v-list-item class="pl-0">
      <v-list-item-content class="py-0 align-self-start">
        <v-list-item-title class="mb-1">{{ item.room }}</v-list-item-title>
        <v-list-item-subtitle class="text-caption">Besucht um {{ formatTime(item.visited_on)
}}</v-list-item-subtitle>
      </v-list-item-content>
      <v-list-item-action class="mt-n1 ml-0 align-self-start">
        <v-btn
          text
          large
          rounded
          outlined
          :ripple="!interval"
          :loading="refresh_loading"
          color="secondary lighten-5"
          @click="reloadVisits">
          <v-icon
            left
            :size="24"
            class="mr-3"
            color="secondary lighten-5"
            :style="{ opacity: opacity }"
            >refresh</v-icon>
          <v-progress-circular
            :rotate="24"
            :size="16"
            :width="2"
            :value="waiting"
            class="mr-3"
            color="secondary lighten-5"
            style="position: absolute; left: -3px;"
          >
          </v-progress-circular>
          <span
            class="text-h4 font-weight-light text--primary persons_count"
            :class="{ 'persons_count--active': interval }">
            {{ item.total_visits.during }}
          </span>
        </v-btn>

```

```

        </v-list-item-action>
    </v-list-item>
</v-timeline-item>
</v-timeline>

        <v-card-actions>
            <v-btn
                @click="exitRoom(item.id)"
                small
                :depressed="success"
                :text="!success"
                :loading="loading"
                :color="success ? 'success' : 'info'"
            >
                <v-icon v-if="success" left>check</v-icon>
                {{ success ? 'Verlassen' : 'Raum Verlassen'}}
            </v-btn>
</v-spacer />

<!-- <v-btn
    small text rounded
    :loading="refresh_loading || !!interval"
    :disabled="!!interval"
    color="secondary lighten-4"
    @click="reloadVisits">
    <v-icon color="secondary lighten-3">refresh</v-icon>
    <template v-slot:loader>
        <v-progress-circular
            :indeterminate="!interval"
            :rotate="-90"
            :size="17"
            :width="2"
            :value="waiting"
            color="secondary lighten-5"
        >
            </v-progress-circular>
    </template>
</v-btn -->

<!-- <v-btn
    v-if="!interval"
    small text rounded
    :loading="refresh_loading"
    color="secondary lighten-4"
    @click="reloadVisits">
    <v-icon left color="secondary lighten-5">refresh</v-icon>
</v-btn>
<v-progress-circular
    v-else
    :rotate="-90"
    :size="18"
    :width="2"
    :value="waiting"
    class="mx-4"

```

```

        color="secondary lighten-5"
      >
    </v-progress-circular> -->
      </v-card-actions>
    </v-card>
  </template>

<script>
import { format, parse } from "date-fns";

import statesMixin from "@/mixins/statesMixin";

import { exitVisit, showVisits } from "@/requests/api";

export default {
  data: () => ({
    loading: false,
    success: false,
    waiting: 0,
    interval: false,
    refresh_loading: false
  }),
  props: {
    item: Object,
  },

  mixins: [statesMixin],

  created() {
    this.setStateCount(2);
    this.setState(1);
  },

  computed: {
    opacity() {
      if (!this.interval) return 1;
      // We have a delay when looping through the visits result
      // so we cannot directly react to the
      return (this.waiting - 30) / 100;
    }
  },

  methods: {
    formatTime(datetime) {
      return format(
        parse(datetime, "yyyy-MM-dd HH:mm:ss", new Date()),
        "HH:mm"
      );
    },

    reloadVisits() {
      if (this.interval) return;

      this.refresh_loading = true;
      showVisits().then(() => {

```

```

    this.interval = setInterval(() => {
      if (this.waiting >= 90) {
        clearInterval(this.interval);
        this.interval = null;
      } else {
        this.waiting += 5
      }
    }, 250)
  }).catch(() => this.refresh_loading = false);

  // showVisits().then(() => {
  //   this.interval = setInterval(() => {
  //     if (this.waiting === 100) {
  //       this.waiting = 0;
  //       clearInterval(this.interval);
  //       this.interval = null;
  //     }
  //     this.waiting += 5
  //   }, 250)
  // })
  // .catch(() => this.refresh_loading = false)
  // .finally(() => console.log("FInally loading"));
},
    exitRoom(id) {
      this.loading = true;

      exitVisit(id)
        .then(() => this.success = true)
        .catch(() => console.log("COULDNT EXIT"))
        .finally(() => this.loading = false);
    },
  },

  watch: {
    visits(){
      // if (!this.refresh_loading) return;

      this.refresh_loading = false;
    },
    interval() {
      if (!this.interval) this.waiting = 0;
    }
  }
};
</script>

<style lang="scss" scoped>

::v-deep {
  &.v-timeline {
    &::before {
      height: calc(100% - 10px) !important;
    }
  }
}

```

```

    & .v-timeline-item {
        padding-bottom: 8px !important;

        & .v-timeline-item__inner-dot {
            position: relative;
        }
    }

& .v-progress-circular__underlay {
    opacity: 0;
}

    & .visit {
        &::before {
            content: "";
            position: absolute;
            left: calc(48px - 1px);
            bottom: 10px;
            width: 2px;
            height: 80px;
            background-color: var(--v-info-lighten1);
        }

        &--done::before {
            background-color: var(--v-background-darken3) !important;
        }
    }
}

.pulse {
    animation: pulse 2s infinite;
    height: 100%;
    width: 100%;
    border-radius: inherit;
    position: absolute;
}

@keyframes pulse {
    0% {
        box-shadow: 0 0 0 0 currentColor;
        opacity: 0.4;
    }
    70% {
        box-shadow: 0 0 0 10px currentColor;
        opacity: 0;
    }
    100% {
        box-shadow: 0 0 0 0 currentColor;
        opacity: 0;
    }
}
</style>

```

Datei: DarkDialog.vue

```
<template>
  <v-dialog
    class="mx-2"
    max-width="500"
    :overlay-color="$vuetify.theme.dark ? 'black' : undefined"
    :overlay-opacity="$vuetify.theme.dark ? 0.7 : undefined"
    v-bind="{ ...$props, ...$attrs }"
    v-on="$listeners"
  >
    <template v-for="(_, slot) of $scopedSlots" v-slot:[slot]="scope">
      <slot :name="slot" v-bind="scope" />
    </template>
  </v-dialog>
</template>

<script>
export default {
  inheritAttrs: false
}
</script>
```

Datei: DatabaseSeeder.php

```
<?php

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     *
     * @return void
     */
    public function run()
    {
        // \App\Models\User::factory(10)->create();
    }
}
```

Datei: DateTimeItem.vue

```
<template>
  <div>
    <v-list-item>
      <v-list-item-action>
        <v-icon>event</v-icon>
      </v-list-item-action>
    </v-list-item>
  </div>
</template>
```

```

<v-list-item-content>
  <v-text-field
    label="Datum"
    type="date"
    v-model="date"
    hide-details="auto"
    :rules="[rules.required, rules.beforeToday]"
    :max="today"
    :min="fourWeeksAgo"

    @change="changed"
  ></v-text-field>
</v-list-item-content>
</v-list-item>
<v-list-item>
  <v-list-item-action>
    <v-icon>arrow_forward</v-icon>
  </v-list-item-action>

  <v-list-item-content>
    <v-text-field
      label="Betreten um"
      type="time"
      ref="time_enter"
      v-model="time_enter"
      hide-details="auto"
      :disabled="!date"
      :rules="[rules.required, rules.ifTodayBeforeNow]"
      @change="changed"
    ></v-text-field>
  </v-list-item-content>
</v-list-item>
<v-list-item>
  <v-list-item-action>
    <v-icon>arrow_back</v-icon>
  </v-list-item-action>

  <v-list-item-content>
    <v-text-field
      type="time"
      ref="time_exit"
      v-model="time_exit"
      hide-details="auto"
      :disabled="!time_enter"
      :rules="[rules.isAfterEnter, rules.ifTodayBeforeNow, rules.requiredIfNotToday]"
      @change="changed"
    >
    <template #label>
      Verlassen um <span class="ml-2" v-if="isNotToday()">(optional)</span>
    </template>
  </v-text-field>
</v-list-item-content>
</v-list-item>

```



```
</div>
</template>
```

```
<script>
```

```
import { parse, format, startOfToday, isPast, isAfter, isEqual, isToday, subDays } from "date-fns"
import { de } from "date-fns/locale";
```

```
export default {
  data: () => ({
    modal: null,
    rules: null,
    date: null,
    time_enter: null,
    time_exit: null
  }),
```

```
  props: {
    value: Object
  },
```

```
  created() {
    this.rules = {
      required: v => !!v || 'Datum ist erforderlich',
      beforeToday: v => v <= this.today || 'Datum muss in der Vergangenheit sein',
      isAfterEnter: v => this.isAfterEnter(v) || 'Uhrzeit muss nach dem Eintritt sein',
      ifTodayBeforeNow: v => this.beforeNow(v) || 'Datum muss in der Vergangenheit sein',
      requiredIfNotToday: v => this.isNotToday(v) || 'Enduhrzeit wird benötigt, wenn der Besuch an
einem anderen Tag statt gefunden hat.'
    }
  },
```

```
  mounted() {
    const [date, time] = format(new Date(), 'yyyy-MM-dd HH:mm').split(' ');

    this.date = date;
    this.time_enter = time;
  },
```

```
  methods: {
    beforeNow(time) {
      if (!time) return true;

      const selected = parse(`${this.date} ${time}`, 'yyyy-MM-dd HH:mm', new Date());

      return isPast(selected);
    },
    isAfterEnter(time) {
      if (!time) return true;

      const enter = parse(`${this.date} ${this.time_enter}`, 'yyyy-MM-dd HH:mm', new Date());
      const selected = parse(`${this.date} ${time}`, 'yyyy-MM-dd HH:mm', new Date());

      return isAfter(selected, enter) || isEqual(selected, enter);
    },
```

```

isNotToday(time) {
  // Date hasn't been filled, nothing to check
  if (!this.visited_on) return true;
  // Date is today, end time not necessarily required
  if (isToday(this.visited_on)) return true;
  // Date is in the past, time is required
  return !!time;
},
changed() {
  if (!this.date || !this.time_enter) return;

  this.$emit('input', {
    visited_on: this.visited_on,
    exited_on: this.exited_on,
  });
}
},

computed: {
  today() {
    return format(startOfToday(), 'yyyy-MM-dd');
  },
  fourWeeksAgo() {
    return format(subDays(new Date(), 28), 'yyyy-MM-dd');
  },
  visited_on() {
    try {
      return parse(`${this.date} ${this.time_enter}`, 'yyyy-MM-dd HH:mm', new Date(), { locale:
de })
    } catch(e) {
      return undefined;
    }
  },
  exited_on() {
    try {
      return parse(`${this.date} ${this.time_exit}`, 'yyyy-MM-dd HH:mm', new Date(), { locale: de
})
    } catch(e) {
      return undefined;
    }
  },
},
},

watch: {
  date() {
    this.$refs.time_enter.validate();

    if (this.$refs.time_exit.hasError) {
      this.$refs.time_exit.validate();
    }

    this.changed();
  },
  time_enter() {

```

```
        this.$refs.time_exit.validate();
        this.changed();
    },
    time_exit() {
        this.changed();
    }
}
};
</script>
```

```
<style>
</style>
```

Datei: EncryptCookies.php

```
<?php

namespace App\Http\Middleware;

use Illuminate\Cookie\Middleware\EncryptCookies as Middleware;

class EncryptCookies extends Middleware
{
    /**
     * The names of the cookies that should not be encrypted.
     *
     * @var array
     */
    protected $except = [
        //
    ];
}
```

Datei: EventServiceProvider.php

```
<?php

namespace App\Providers;

use Illuminate\Auth\Events\Registered;
use Illuminate\Auth\Listeners\SendEmailVerificationNotification;
use Illuminate\Foundation\Support\Providers\EventServiceProvider as ServiceProvider;
use Illuminate\Support\Facades\Event;

class EventServiceProvider extends ServiceProvider
{
    /**
     * The event listener mappings for the application.
     *
     * @var array
     */
```

```

protected $listen = [
    Registered::class => [
        SendEmailVerificationNotification::class,
    ],
];

/**
 * Register any events for your application.
 *
 * @return void
 */
public function boot()
{
    parent::boot();

    //
}
}

```

Datei: ExitRooms.php

```
<?php
```

```
namespace App\Console\Commands;
```

```
use App\Visit;
use Carbon\Carbon;
use Illuminate\Console\Command;
use Illuminate\Support\Facades\DB;
```

```
class ExitRooms extends Command
{
    /**
     * The name and signature of the console command.
     *
     * @var string
     */
    protected $signature = 'visits:exit';

    /**
     * The console command description.
     *
     * @var string
     */
    protected $description = 'Exit rooms at the end of the day';

    /**
     * Create a new command instance.
     *
     * @return void
     */
    public function __construct()

```

```

{
    parent::__construct();
}

/**
 * Execute the console command.
 *
 * @return int
 */
public function handle()
{
    // $end_of_day = Carbon::now('Europe/Berlin')->endOfDay();
    return Visit::whereNull('exited_on')->update([
        // visited_on + 1day - 1second => 23:59 of that day
        'exited_on' => DB::raw("date_trunc('day', visited_on) + INTERVAL '1 DAY' - INTERVAL '1
SECOND'")
    ]);
}
}

```

Datei: Handler.php

```
<?php
```

```
namespace App\Exceptions;
```

```
use Illuminate\Foundation\Exceptions\Handler as ExceptionHandler;
```

```
use Illuminate\Validation\ValidationException;
```

```
use Throwable;
```

```
class Handler extends ExceptionHandler
```

```

{
    /**
     * A list of the exception types that are not reported.
     *
     * @var array
     */
    protected $dontReport = [
        //
    ];

    /**
     * A list of the inputs that are never flashed for validation exceptions.
     *
     * @var array
     */
    protected $dontFlash = [
        'password',
        'password_confirmation',
    ];

    /**
     * Report or log an exception.

```

```

*
* @param \Throwable $exception
* @return void
*
* @throws \Exception
*/
public function report(Throwable $exception)
{
    parent::report($exception);
}

/**
 * Render an exception into an HTTP response.
 *
 * @param \Illuminate\Http\Request $request
 * @param \Throwable $exception
 * @return \Symfony\Component\HttpFoundation\Response
 *
 * @throws \Throwable
 */
public function render($request, Throwable $exception)
{
    return parent::render($request, $exception);
}
}

```

Datei: HasScrambledId.php

```
<?php
```

```
namespace App\Traits;
```

```
use Illuminate\Support\Str;
```

```
trait HasScrambledId {
```

```
    public static function bootHasScrambledId()
```

```
    {
        static::creating(function($model) {
            $model->scrambled_id = $model->scrambled_id ?? Str::random(16);
        });
    }

```

```
/**
 * Get the route key for the model.
 *
```

```
 * @return string
 */
```

```
public function getRouteKeyName()
```

```
{
    return 'scrambled_id';
}

```

```
}
```

Datei: Home.vue

```
<template>
  <div class="pa-0 align-start d-flex flex-column">
    <Toolbar />
    <v-sheet
      class="flex-grow-1 align-self-center d-flex flex-column px-1"
      :class="{ 'px-4': $vuetify.breakpoint.width > 360 }"
      color="rgba(255, 0, 0, 0)"
      style="margin-top: -36px"
      width="100%"
      max-width="700px"
      min-height="calc(100vh - 160px + 36px)"
    >

    <v-expand-transition>
      <CurrentVisit
        class="mb-4"
        v-if="active_visit"
        :item="active_visit"
      />
    </v-expand-transition>

    <InstallItem />

    <VisitsLoading v-if="loading" />
    <template v-else-if="!hasVisits">
      <VisitsEmpty class="mb-4" />
      <!-- This is floating -->
      <StartHere />
    </template>
    <template v-else>
      <SuggestedItem class="mb-4" />
      <TotalEncounters class="mb-4" />
    </template>

    <v-list color="transparent" flat two-line v-if="hasVisits">
      <v-card
        id="all_values"
        v-intersect.quiet="{
          handler: onIntersect,
          options: {
            threshold: 0,
            rootMargin: '-128px 0px 0px 0px'
          }
        }"
        :elevation="all_values_sticking ? 2 : 0"
        :class="{ 'sticking' : all_values_sticking }"
      >
      <!-- :color="all_values_sticking ? 'inherit' : 'transparent'" -->
      <!-- <div class="mb-3" style="top: 76px" id="sticker"> -->
    </v-list>
  </div>
</template>
```

```

<!-- <div class="mb-3" style="position: sticky; top: 76px" id="sticker"> -->
  <div class="d-flex">
    <div class="overline pl-4 pt-2 pb-3 flex-grow-1">Alle Einträge</div>
    <v-tooltip top eager>
      <template v-slot:activator="{ on, attrs }">
        <v-icon
          v-on="on"
          v-bind="attrs"

          small
          color="secondary lighten-4">info_outlined</v-icon>
        </template>
        <span>Max. 4 Wochen werden gespeichert</span>
      </v-tooltip>
    </div>
    <!-- <div class="hide_shadow" /> -->
  </v-card>
  <v-card tile elevation="0" class="hide_shadow" :style="{ 'opacity': all_values_sticking ? 1 : 0
}></v-card>
  <!-- <v-card
    id="all_values"
    class="elevation-1">
    <div class="d-flex">
      <div class="overline pl-4 pt-2 flex-grow-1">Alle Einträge</div>
      <v-tooltip top>
        <template v-slot:activator="{ on, attrs }">
          <v-icon
            v-on="on"
            v-bind="attrs"

            small
            color="secondary lighten-4">info_outlined</v-icon>
          </template>
          <span>Max. 4 Wochen werden gespeichert</span>
        </v-tooltip>
      </div>
      <div class="mb-n1 hide_shadow"></div>
    </v-card> -->
    <v-card
      v-for="(items, key, index) in groupedItems"
      :key="key"
      class="mb-4 py-3 elevation-1"
      :class="{ 'pt-1': index === 0 }"
      :style="{ 'border-radius': all_values_sticking ? '0 0 4px 4px' : '4px' }"
    >
      <div class="overline pl-4">{{ key }}</div>
      <v-list-group
        v-for="item in items"
        :key="item.id"
        no-action

        color="info"
      >
        <template v-slot:activator>

```



```

        <VisitItem :item="item" />
    </template>

    <TimelineVertical :item="item" />
</v-list-group>
</v-card>
</v-list>

    <v-spacer />
    <PrivacyFooter />
    </v-sheet>

    <router-view />

    <AddButton />

</div>
</template>

<script>
import { format, parse } from "date-fns";
import { de } from "date-fns/locale";

import Toolbar from "@components/toolbar/Toolbar";

import AddButton from "@components/home/AddButton";

import VisitItem from "@components/home/VisitItem";
import TimelineVertical from "@components/home/TimelineVertical";

import StartHere from "@components/home/StartHere";
import VisitsEmpty from "@components/home/VisitsEmpty";
import CurrentVisit from "@components/home/CurrentVisitDirect";
import VisitsLoading from "@components/home/VisitsLoading";
import SuggestedItem from "@components/home/SuggestedItem";
import InstallItem from "@components/home/InstallItem";

import TotalEncounters from "@components/home/TotalEncounters";

import axios from "@requests/connection";
import { init } from "@requests/api";

export default {
    data: () => ({
        loading: true,
        active_visit: null,
        groupedItems: [],
        all_values_sticking: false
    }),

    async mounted() {
        await axios.get("/sanctum/csrf-cookie");

        if (this.navigateTo !== 'account' && this.mustLogin) {

```

```

    this.$router.push({ name: "account" }).catch(() => {});
  }
  else if (this.navigateTo === 'account' && this.isLoggedIn) {
    this.$router.replace({ name: "home" }).catch(() => {});
  }

  // this.loading = true;

  await init();
  this.loading = false;
  // Don't await
  this.getRooms();
  },

  beforeRouteEnter(to, from, next) {
    next((vm) => vm.navigateTo = to.name);
  },

  beforeRouteUpdate(to, from, next) {
    this.$router.history._startLocation = '/';
    next();
  },

  methods: {
    onIntersect(entries) {
      this.all_values_sticking = !entries[0].isIntersecting;
    },

    prepareItems(items) {
      this.active_visit = false;

      return items.reduce((a, item) => {
        (item.date = parse(
          item.visited_on,
          "yyyy-MM-dd HH:mm:ss",
          new Date()
        )),
        (item.date_string = format(item.date, "EEE, MMM dd", {
          locale: de,
        }));

        if (!item.exited_on) {
          this.active_visit = item;
          return a;
        }

        const key = item.date_string;

        (a[key] = a[key] || []).push(item);
        return a;
      }, {});
    },
  },
},

```

```

    watch: {
      visits() {
        if (!this.visits) return;

        this.groupedItems = this.prepareItems(this.visits);
      },
    },

    components: {
      Toolbar,
      CurrentVisit,
      InstallItem,
      StartHere,
      VisitsEmpty,
      VisitsLoading,
      SuggestedItem,
      TotalEncounters,

      AddButton,
      VisitItem,

      TimelineVertical,
    },
  };
</script>

<style lang="scss" scoped>
::v-deep {
  & .v-skeleton-loader.v-skeleton-loader--is-loading {
    display: flex;
    flex-direction: column;
    justify-content: center;

    > div {
      height: auto;
    }
  }
}

#all_values {
  border-radius: 4px 4px 0 0;
  position: sticky;
  top: 0;
  z-index: 1;
  transition: background-color 0.1s linear, box-shadow 0.1s linear;

  &:not(.sticking) {
    background-color: transparent;
  }
}

.hide_shadow {
  height: 12px;
}

```

```
z-index: 1;
position: sticky;
top: 36px;
margin-top: -8px;
margin-bottom: -4px;
transition: opacity 0.1s linear;
}
```

```
</style>
```

Datei: InstallButton.vue

```
<template>
  <div v-if="canBeInstalled">
    <v-divider />
    <v-list-item
      v-if="installing"
      class="mt-2">
      <v-list-item-icon>
        <v-progress-circular
          size="24"
          indeterminate
          width="2"
          color="secondary lighten-3"></v-progress-circular>
      </v-list-item-icon>
      <v-list-item-title>Wird installiert...</v-list-item-title>
    </v-list-item>
    <v-list-item
      v-else
      class="mt-2"
      id="install"
      @click.stop.prevent="install">
      <v-list-item-icon><v-icon>add_to_home_screen</v-icon></v-list-item-icon>
      <v-list-item-title>App installieren</v-list-item-title>
      <v-list-item-action class="ml-8">
        <v-icon small color="accent">circle</v-icon>
      </v-list-item-action>
    </v-list-item>

    <DarkDialog v-model="dialog">
      <InstallDialog :type="browser" />
    </DarkDialog>
  </div>
</template>
```

```
<script>
import InstallDialog from "@components/home/InstallDialog";

import installMixin from "@mixins/installMixin"

export default {
  mixins: [installMixin],
  data: () => ({
```

```

    dialog: false,
  }),
  methods: {
    install() {
      this.installing = true;

      if (this.deferred_prompt) {
        this.deferred_prompt.prompt();
      }
      else if (this.isSafari || this.isFirefox) {
        this.dialog = true;
      }
    }
  },
  watch: {
    dialog(active) {
      if (!active) this.installing = false;
    }
  },
  components: {
    InstallDialog,
  }
}
</script>

```

```

<style>
@media all and (display-mode: standalone) {
  #install {
    display: none !important;
  }
}
</style>

```

Datei: InstallDialog.vue

```

<template>
  <!-- <div @click="cycleState" style="overflow: hidden">
    <v-card v-if="state === 0" class="text-center">
      <div class="d-flex justify-space-between pt-10 pb-2 mx-n2">
        <v-sheet
          height="50"
          width="50"
          color="primary opacity-25"
          rounded
        />
        <v-sheet
          height="50"
          width="50"
          color="primary opacity-50"
          rounded
        />
      </div>
    </v-card>
  </div>

```

```

    <v-img
      class="rounded-lg"
      height="50"
      width="50"
      contain
      style="background-color: var(--v-primary-base)"
      :src="`./img/icons/apple-touch-icon-152x152.png`"
    />
  </div>
  <v-sheet
    height="50"
    width="50"
    color="primary opacity-50"
    rounded
  />
  <v-sheet
    height="50"
    width="50"
    color="primary opacity-25"
    rounded
  />
</div>

<div class="text-h5 font-weight-bold py-3">Webapp installieren</div>

<div class="subtitle-2 font-weight-regular pb-4 px-6">
  Installieren Sie diese Anwendung auf Ihrem Startbildschirm für
  schnellen und einfachen Zugriff
</div>

<v-card-actions
  style="background-color: var(--v-background-darken2)"
  class="caption justify-center pt-3"
>
  <template v-if="type === 'Firefox'"
    >Tippen Sie auf
    <v-icon
      color="background"
      class="rounded-circle safari-icon"
      style="background-color: var(--v-background-darken5)"
      >more_vert</v-icon>
    >
    dann 'Installieren'</template>
  >
  <template v-else-if="type === 'Safari'"
    >Tippen Sie auf <shareIcon class="safari-icon" /> dann 'Zum Home-Bildschirm'</template>
  >
</v-card-actions>
</v-card>
<v-card v-else-if="state === 1">
  <v-card-title>
    Webapp installieren
  </v-card-title>
  <div class="d-flex">

```

```

<div class="body-1 font-weight-regular pb-4 pl-4">
  Installieren Sie diese Anwendung auf Ihrem Startbildschirm für
  schnellen und einfachen Zugriff
</div>
<v-sheet class="d-flex flex-wrap flex-shrink-0 pb-4 px-2" width="140px">
  <v-sheet
    v-for="i in 4"
    :key="`before${i}`"
    height="32"
    width="32"
    color="secondary"
    :style="{ opacity: i/20 }"
    class="ma-1"
    rounded
  />

  <div>
    <v-img
      class="rounded-lg ma-1"
      height="32"
      width="32"
      contain
      style="background-color: var(--v-primary-base)"
      :src="`./img/icons/apple-touch-icon-152x152.png`"
    />
  </div>

  <v-sheet
    v-for="i in 7"
    :key="`after${i}`"
    height="32"
    width="32"
    color="secondary"
    :style="{ opacity: 0.4 - i/20 }"
    class="ma-1"
    rounded
  />

</v-sheet>
</div>

<v-card-actions
  style="background-color: var(--v-background-darken2)"
  class="caption justify-center pt-3"
>
  <template v-if="type === 'Firefox'"
  >Tippen Sie auf
  <v-icon
    color="background"
    class="rounded-circle safari-icon"
    style="background-color: var(--v-background-darken5)"
    >more_vert</v-icon

```

```

    >
    dann 'Installieren'</template
  >
  <template v-else-if="type === 'Safari'"
    >Tippen Sie auf <shareIcon class="safari-icon" /> dann 'Zum Home-Bildschirm'</template
  >
</v-card-actions>
</v-card>
<v-card v-else-if="state === 2">
  <v-card-title>
    Webapp installieren
  </v-card-title>
  <div class="d-flex">

    <div class="body-1 font-weight-regular pb-4 pl-4">
      Installieren Sie diese Anwendung auf Ihrem Startbildschirm für
      schnellen und einfachen Zugriff
    </div>
    <v-sheet class="d-flex flex-wrap flex-shrink-0 pb-4 px-2" width="140px">
      <v-sheet
        v-for="i in 4"
        :key="`before${i}`"
        height="32"
        width="32"
        color="secondary"
        :style="{ opacity: 0.2 }"
        class="ma-1"
        rounded
      />

      <div>
        <v-img
          class="rounded-lg ma-1"
          height="32"
          width="32"
          contain
          style="background-color: var(--v-primary-base)"
          :src="`./img/icons/apple-touch-icon-152x152.png`"
        />
      </div>

      <v-sheet
        v-for="i in 7"
        :key="`after${i}`"
        height="32"
        width="32"
        color="secondary"
        :style="{ opacity: 0.2 }"
        class="ma-1"
        rounded
      />

    </v-sheet>
  </div>

```



```

<v-card-actions
  style="background-color: var(--v-background-darken2)"
  class="caption justify-center pt-3"
>
  <template v-if="type === 'Firefox'"
    >Tippen Sie auf
    <v-icon
      color="background"
      class="rounded-circle safari-icon"
      style="background-color: var(--v-background-darken5)"
      >more_vert</v-icon
    >
    dann 'Installieren'</template
  >
  <template v-else-if="type === 'Safari'"
    >Tippen Sie auf <shareIcon class="safari-icon" /> dann 'Zum Home-Bildschirm'</template
  >
</v-card-actions>
</v-card>

```

```

</div> -->

```

```

<v-card>
  <v-card-title>
    Webapp installieren
  </v-card-title>
  <div class="d-flex">

```

```

  <div class="body-1 font-weight-regular pb-4 pl-4">
    Installieren Sie diese Anwendung auf Ihrem Startbildschirm für
    schnellen und einfachen Zugriff
  </div>

```

```

  <v-sheet class="d-flex flex-wrap flex-shrink-0 pb-4 px-2" width="140px">
    <v-sheet
      v-for="i in 4"
      :key="`before${i}`"
      height="32"
      width="32"
      color="secondary"
      :style="{ opacity: 0.2 }"
      class="ma-1"
      rounded
    />

```

```

  <div>
    <v-img
      class="rounded-lg ma-1"
      height="32"
      width="32"
      contain
      style="background-color: var(--v-primary-base)"
      :src="`./img/icons/apple-touch-icon-152x152.png`"
    />

```

```

    </div>

    <v-sheet
      v-for="i in 7"
      :key="` after${i}`"
      height="32"
      width="32"
      color="secondary"
      :style="{ opacity: 0.2 }"
      class="ma-1"
      rounded
    />

  </v-sheet>
</div>

<v-card-actions
  style="background-color: var(--v-background-darken2)"
  class="caption justify-center pt-3"
>
  <template v-if="type === 'Firefox'"
    >Tippen Sie auf
    <v-icon
      color="background"
      class="rounded-circle safari-icon"
      style="background-color: var(--v-background-darken5)"
      >more_vert</v-icon
    >
    dann 'Installieren'</template
  >
  <template v-else-if="type === 'Safari'"
    >Tippen Sie auf <shareIcon class="safari-icon" /> dann 'Zum Home-Bildschirm'</template
  >
</v-card-actions>
</v-card>
</template>

<script>
import statesMixin from "@/mixins/statesMixin";
import shareIcon from "@/assets/ios-share-icon.svg";

export default {
  mixins: [statesMixin],
  props: {
    type: String
  },
  data: () => ({
    publicPath: null
  }),
  created() {
    this.publicPath = process.env.BASE_URL;
    this.setStateCount(3);
  },
}

```

```

    components: {
      shareIcon
    }
  };
</script>

<style lang="scss" scoped>

.safari-icon {
  height: 24px;
  margin: 0 8px 8px;
  fill: var(--v-info-base);
}

</style>

```

Datei: InstallItem.vue

```

<template>
  <v-expand-transition>
    <div @click="cycleState" v-if="showInstall">
      <v-card
        v-if="state === 0"
        class="mb-4" color="info" dark>
        <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
        <v-card-subtitle>Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die Webapp auf Ihrem
Gerät installieren.</v-card-subtitle>
        <v-card-actions>
          <v-spacer />
          <v-btn
            small
            rounded
            text
            class="px-4"
            @click.stop="askLater"
          >
            Nicht jetzt
          </v-btn>
          <v-btn
            small
            rounded
            depressed
            light
            class="px-4"
            @click.stop.prevent="install"
          >
            Installieren
          </v-btn>
        </v-card-actions>
      </v-card>
    </v-card>

```

```

v-else-if="state === 1"
class="mb-4" style="border-left: var(--v-accent-base) solid 8px">
<v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
  <v-card-subtitle>Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die Webapp auf Ihrem
Gerät installieren.</v-card-subtitle>
  <v-card-actions>
    <v-spacer />
    <v-btn
      small
      rounded
      text
      class="px-4"
      @click.stop="askLater"
    >
    Nicht jetzt
  </v-btn>
  <v-btn
    small
    rounded
    depressed
    light
    class="px-4"
    color="primary"
  >
  Installieren
</v-btn>
</v-card-actions>
</v-card>
<v-card
  v-else-if="state === 2"
  class="mb-4" color="#6e819e" dark>
  <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
  <v-card-subtitle>Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die Webapp auf Ihrem
Gerät installieren.</v-card-subtitle>
  <v-card-actions>
    <v-spacer />
    <v-btn
      small
      rounded
      text
      class="px-4"
      @click.stop="askLater"
    >
    Nicht jetzt
  </v-btn>
  <v-btn
    small
    rounded
    depressed
    light

```

```

        class="px-4"
        @click.stop.prevent="install"
    >
    Installieren
</v-btn>
</v-card-actions>
</v-card>
<v-card
    v-else-if="state === 3"
    class="mb-4" color="#395580" dark>
    <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
    <v-card-subtitle>Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die Webapp auf Ihrem
Gerät installieren.</v-card-subtitle>
    <v-card-actions>
    <v-spacer />
    <v-btn
        small
        rounded
        text
        class="px-4"
        @click.stop="askLater"
    >
    Nicht jetzt
</v-btn>
    <v-btn
        small
        rounded
        outlined
        dark
        class="px-4"
        @click.stop.prevent="install"
    >
    Installieren
</v-btn>
</v-card-actions>
</v-card>
<v-card
    v-else-if="state === 4"
    class="mb-4" color="#f6cd61">
    <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
    <v-card-subtitle>Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die Webapp auf Ihrem
Gerät installieren.</v-card-subtitle>
    <v-card-actions>
    <v-spacer />
    <v-btn
        small
        rounded
        text
        class="px-4"
        @click.stop="askLater"

```

```

    >
    Nicht jetzt
  </v-btn>
  <v-btn
    small
    rounded
    depressed
    light
    class="px-4"
    @click.stop.prevent="install"
  >
  Installieren
  </v-btn>
</v-card-actions>
</v-card>
<v-card
  v-else-if="state === 5"
  class="mb-4" color="#fe8a71">
  <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
  <v-card-subtitle>Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die Webapp auf Ihrem
Gerät installieren.</v-card-subtitle>
  <v-card-actions>
    <v-spacer />
    <v-btn
      small
      rounded
      text
      class="px-4"
      @click.stop="askLater"
    >
    Nicht jetzt
  </v-btn>
  <v-btn
    small
    rounded
    depressed
    light
    class="px-4"
    @click.stop.prevent="install"
  >
  Installieren
  </v-btn>
</v-card-actions>
</v-card>
<v-card
  v-else-if="state === 6"
  class="mb-4" color="#adcbe3">
  <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
  <v-card-subtitle>Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die Webapp auf Ihrem
Gerät installieren.</v-card-subtitle>

```

```

<v-card-actions>
  <v-spacer />
  <v-btn
    small
    rounded
    text
    class="px-4"
    @click.stop="askLater"
  >
  Nicht jetzt
</v-btn>
  <v-btn
    small
    rounded
    depressed
    light
    class="px-4"
    @click.stop.prevent="install"
  >
  Installieren
</v-btn>
</v-card-actions>
</v-card>
<v-card
  v-else-if="state === 7"
  class="mb-4" color="#42317b" dark>
  <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
  <v-card-subtitle>Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die Webapp auf Ihrem
Gerät installieren.</v-card-subtitle>
  <v-card-actions>
    <v-spacer />
    <v-btn
      small
      rounded
      text
      class="px-4"
      @click.stop="askLater"
    >
    Nicht jetzt
  </v-btn>
    <v-btn
      small
      rounded
      depressed
      light
      class="px-4"
      @click.stop.prevent="install"
    >
    Installieren
  </v-btn>
</v-card-actions>
</v-card>

```

```

<v-card
  v-else-if="state === 8"
  class="mb-4" color="#6497b1" dark>
  <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
  <div class="d-flex pb-3">
    <v-icon large class="ml-5 mr-3">add_to_home_screen</v-icon>
    <v-card-subtitle class="py-0">Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die
Webapp auf Ihrem Gerät installieren.</v-card-subtitle>
  </div>

```

```

<v-card-actions>
  <v-spacer />
  <v-btn
    small
    rounded
    text
    class="px-4"
    @click.stop="askLater"
  >
  Nicht jetzt
</v-btn>
  <v-btn
    small
    rounded
    depressed
    light
    class="px-4"
    @click.stop.prevent="install"
  >
  Installieren
</v-btn>
</v-card-actions>

```

```

</v-card>
<v-card
  v-else-if="state === 9"
  class="mb-4" color="#b3cde0">
  <v-card-title class="subtitle-1 font-weight-normal"><span class="first-line">THWi
Kontaktnachverfolgung auf Ihren</span><span class="second-line">Home-Bildschirm</span></v-card-
title>
  <div class="d-flex pb-3">
    <v-icon large class="ml-5 mr-3">add_to_home_screen</v-icon>
    <v-card-subtitle class="py-0">Greifen Sie schneller auf Ihre Inhalte zu, indem Sie die
Webapp auf Ihrem Gerät installieren.</v-card-subtitle>
  </div>

```

```

<v-card-actions>
  <v-spacer />
  <v-btn
    small
    rounded
    text
    class="px-4"

```



```

        @click.stop="askLater"
      >
      Nicht jetzt
    </v-btn>
    <v-btn
      small
      rounded
      depressed
      light
      class="px-4"
      @click.stop.prevent="install"
    >
      Installieren
    </v-btn>
  </v-card-actions>
</v-card>

  <DarkDialog v-model="dialog">
    <InstallDialog :type="browser" />
  </DarkDialog>
</div>
</v-expand-transition>
</template>

```

```

<script>
import statesMixin from "@/mixins/statesMixin"
import installMixin from "@/mixins/installMixin"
import InstallDialog from "@/components/home/InstallDialog";

```

```

export default {
  data: () => ({
    dialog: false,
  }),
  // mixins: [statesMixin],
  mixins: [statesMixin, installMixin],
  created() {
    this.setState(8);
    this.setStateCount(10);

    console.log("Should show install", this.canBeInstalled, this.showInstall);
  },
  methods: {
    askLater() {
      this.deferInstall();
    },
    install() {
      this.installing = true;
      if (this.deferred_prompt) {
        this.deferred_prompt.prompt();
      }
      else if (this.isSafari || this.isFirefox) {
        this.dialog = true;
      }
    }
  }
}

```

```
    },
    watch: {
        dialog(active) {
            if (!active) this.installing = false;
        }
    },
    components: {
        InstallDialog
    }
}
</script>
```

```
<style lang="scss" scoped>
```

```
.first-line::after {
    content: '';
    display: inline-flex;
}
```

```
.second-line {
    display: inline-flex;
}
```

```
.second-line::before {
    content: "\a ";
    white-space: pre;
}
```

```
</style>
```

Datei: Kernel.php

```
<?php
```

```
namespace App\Console;
```

```
use Illuminate\Console\Scheduling\Schedule;
use Illuminate\Foundation\Console\Kernel as ConsoleKernel;
```

```
class Kernel extends ConsoleKernel
```

```
{
    /**
     * The Artisan commands provided by your application.
     *
     * @var array
     */
    protected $commands = [
        //
    ];

    /**
     * Define the application's command schedule.
     *
     */
}
```

```

* @param \Illuminate\Console\Scheduling\Schedule $schedule
* @return void
*/
protected function schedule(Schedule $schedule)
{
    $schedule->command('visits:exit')->daily();
    $schedule->command('visits:clear')->daily();

    // $schedule->command('disposable:update')->weekly();
}

/**
 * Register the commands for the application.
 *
 * @return void
 */
protected function commands()
{
    $this->load(__DIR__.'/Commands');

    require base_path('routes/console.php');
}
}

```

Datei: LoadingButton.vue

```

<template>
    <v-btn
        block
        large
        rounded
        :loading="loading"
        :disabled="$attrs.disabled || loading"
        :depressed="success"
        :ripple="!success"
        :color="success ? 'success' : 'primary'"
        @click="click"
    >
        <v-icon left>{{ success ? 'check' : this.icon }}</v-icon>
        {{ success ? this.success_text : this.text }}
    </v-btn>
</template>

<script>
export default {
    props: {
        query: [Function, Promise],
        icon: {
            type: String,
            default: 'send'
        },
    },
}

```

```

    text: {
      type: String,
      default: 'Senden'
    },
    success_text: {
      type: String,
      default: 'Danke'
    },
  },
  data: () => ({
    success: false,
    loading: false,
  }),
  // created() {
  //   console.log("Attrs", this.$attrs)
  // },
  // computed: {
  //   icon() {
  //     return this.icon || 'send';
  //   }
  // },
  methods: {
    click() {
      if (this.success) return;

      this.$emit('click', this);
    },
  },
  watch: {
    query() {
      if (!this.query) return;

      this.loading = true;

      this.query.then(() => this.success = true)
        .finally(() => this.loading = false);
    }
  }
};
</script>

<style>
</style>

```

Datei: Login.vue

```

<template>
  <v-sheet color="primary">

```

```

<div class="pill">
  <!-- <object type="image/svg+xml" :data="require('th-wildau-logo.svg')">
    <style>
      .partnerLogo {
        width: 100%;
        height: 100px;
        stroke: red;
        fill: red;
      }
    </style>
  </object -->
</div>
</v-sheet>
</template>

```

```

<script>
export default {

```

```

}
</script>

```

```

<style>

```

```

</style>

```

Datei: LoginHelper.php

```

<?php

```

```

namespace App\Helpers;

```

```

use Illuminate\Support\Arr;
use Illuminate\Support\Facades\Http;

```

```

require_once(' ../Ldap.php');
class LoginHelper {

```

```

/**
 * Undocumented function
 *
 * @param [type] $username
 * @param [type] $password
 * @return App\Helpers\AuthUser
 */

```

```

public static function attemptLogin($username, $password) {
    $data = self::loginFacade($username, $password);

```

```

    if (!$data) return null;

```

```

    [
        'ldap_id' => $ldap_id,
        'firstname' => $first_name,
        'lastname' => $last_name,
    ]

```

```

        'email' => $email
    ] = $data;

    return new AuthUser($ldap_id, $first_name, $last_name, $email);
}

private static function loginFacade($username, $password) {
    $entry = \Ldap::authenticate($username, $password);

    return [
        'ldap_id' => $entry['id'],
        'firstname' => $entry['firstname'],
        'lastname' => $entry['lastname'],
        'email' => $entry['email']
    ];
}
}

```

Datei: LoginRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'username' => 'required|alpha_dash',
            'password' => 'required'
        ];
    }
}

```

Datei: Page.vue

```
<template>
  <v-dialog
    ref="page"
    :value="open" :transition="transition"
    content-class="page__background"
    fullscreen
    persistent no-click-animation> <!-- this is a workaround for closing dialog when clicking on
snackbar -->
  <v-card flat tile class="d-flex flex-column">
    <v-app-bar
      dark
      fixed
      :color="color"
      elevate-on-scroll
      scroll-target=".v-dialog--active"
      :extension-height="extensionHeight"
      :class="{ 'gradient': gradient }"
      ref="toolbar"
    >
      <v-toolbar-title>{{ title }}</v-toolbar-title>

      <v-spacer></v-spacer>

      <v-progress-circular
        v-if="closable"
        :value="closing"
        width="15"
        rotate="-90"
        class="mr-n1"
        color="rgba(255,255,255, 0.3)"
      >
        <v-btn icon @click="navigateAway">
          <v-icon>close</v-icon>
        </v-btn>
      </v-progress-circular>

      <template v-slot:extension v-if="!!this.$slots.extension">
        <v-sheet class="d-flex flex-grow-1 justify-center" color="transparent">
          <v-sheet :max-width="maxWidth" width="100%" class="align-self-center d-flex flex-
column" color="transparent">
            <slot name="extension" />
          </v-sheet>
        </v-sheet>
      </template>
    </v-app-bar>

    <v-sheet
      :max-width="maxWidth"
      :style="{ paddingTop: `${toolbar_padding}px` }"
      width="100%"
```

```

        class="align-self-center d-flex flex-column flex-grow-1"
        color="transparent">
        <slot></slot>
    </v-sheet>

    </v-card>
</v-dialog>
</template>

<script>
export default {
  data: () => ({
    open: true,
    closing: 0,
    transition: null,
    toolbar_padding: 0
  }),

  props: {
    maxWidth: {
      type: String,
      default: '700px'
    },
    closable: {
      type: Boolean,
      default: true
    },
    gradient: {
      type: Boolean,
      default: false
    },
    color: {
      type: String,
      default: 'primary'
    },
    title: {
      type: String,
      required: true
    },
    extensionHeight: {
      type: [Number, String],
      default: undefined
    }
  },

  created() {
    this.transition = !this.is_opened_directly ? 'dialog-bottom-transition' : 'appear-transition';
  },

  mounted() {
    document.querySelector('html').classList.add('overflow-y-hidden');
    this.transition = 'dialog-bottom-transition';

    if (this.color !== 'transparent') {

```



```

    const content_rect = document.querySelector('.v-
toolbar__content').getBoundingClientRect().height || 0;
    const extension_rect = document.querySelector('.v-
toolbar__extension')?.getBoundingClientRect().height || 0;

    this.toolbar_padding = content_rect + extension_rect;
  }

  // console.log("Test");

  // var _overlay = document.getElementById('overlay');
  // var _clientY = null; // remember Y position on touch start

  // _overlay.addEventListener('touchstart', function (event) {
  //   if (event.targetTouches.length === 1) {
  //     // detect single touch
  //     _clientY = event.targetTouches[0].clientY;
  //   }
  // }, false);

  // _overlay.addEventListener('touchmove', function (event) {
  //   if (event.targetTouches.length === 1) {
  //     // detect single touch
  //     disableRubberBand(event);
  //   }
  // }, false);

  // function disableRubberBand(event) {
  //   var clientY = event.targetTouches[0].clientY - _clientY;

  //   if (_overlay.scrollTop === 0 && clientY > 0) {
  //     // element is at the top of its scroll
  //     event.preventDefault();
  //   }

  //   if (isOverlayTotallyScrolled() && clientY < 0) {
  //     //element is at the top of its scroll
  //     event.preventDefault();
  //   }
  // }

  // function isOverlayTotallyScrolled() {
  //   // https://developer.mozilla.org/en-
  //   // US/docs/Web/API/Element/scrollHeight#Problems_and_solutions
  //   return _overlay.scrollHeight - _overlay.scrollTop <= _overlay.clientHeight;
  // }

  // console.log("REFS", this.$refs.page);

  // this.$refs.page.$el.addEventListener('touchstart', function (event) {
  //   console.log("Scrolling", event.pageY);
  //   this.allowUp = (this.scrollTop > 0);
  //   this.allowDown = (this.scrollTop < this.scrollHeight - this.clientHeight);
  //   this.slideBeginY = event.pageY;

```

```

// });

// this.$refs.page.$el.addEventListener('touchmove', function (event) {
//   var up = (event.pageY > this.slideBeginY);
//   var down = (event.pageY < this.slideBeginY);
//   this.slideBeginY = event.pageY;
//   if ((up && this.allowUp) || (down && this.allowDown)) {
//     event.stopPropagation();
//   }
//   else {
//     event.preventDefault();
//   }
// });
},

computed: {
  is_opened_directly() {
    return this.$router.history._startLocation !== '/'
  },
},

methods: {
  close() {
    this.closing = 100;

    setTimeout(this.navigateAway, 2000);
  },
  navigateAway() {
    this.open = false;

    setTimeout(() => {
      if (this.is_opened_directly) {
        this.$router.replace({ name: "home" }).catch(() => {});
      } else {
        console.log("GOing back");
        this.$router.go(-1);
      }
    }, 200);
  },
},
}
</script>

```

```
<style lang="scss" scoped>
```

```

// Because iOS overscrolls modal windows, we need to hide whats behind
.theme--light .v-dialog__content::v-deep .page__background {
  background-color: white !important;
}
.theme--dark .v-dialog__content::v-deep .page__background {
  background-color: rgb(30 30 30) !important;
}

::v-deep {

```

```

& .v-progress-circular__overlay {
  transition: all 1.75s ease-in;
}

& .v-progress-circular__underlay {
  opacity: 0;
}

& .v-toolbar.gradient {
  background: linear-gradient(180deg, rgb(0 0 0 / 80%) 0%, rgb(255 255 255 / 0%) 87%) !important;
}
}
</style>

```

Datei: PleaseLoginItem.vue

```

<template>
  <v-card>
    <template v-if="state === 0">
      <div class="overline px-4 pt-2">Gesamtkontakte</div>
      <div class="d-flex">
        <div class="body-2 px-4 text--secondary flex-grow-1">
          Meldungen Sie sich an um informiert werden zu können, wenn es ein Ansteckungsgefahr statt
          findet
        </div>

        <div class="mr-4 text-h4 font-weight-light d-flex align-center justify-center" style="min-
          width: 48px">
          {{ encounters }}
        </div>
      </div>
    </template>

    <v-card-text v-if="state === 1" class="pb-0">
      <div>Begegnungen</div>
      <p class="display-1 text--primary">
        {{ encounters }}
      </p>
      <div class="text--primary">
        Meldungen Sie sich an um informiert werden zu können, wenn es ein Ansteckungsgefahr statt
        findet
      </div>
    </v-card-text>

    <v-card-actions>
      <v-spacer />
      <v-btn
        color="primary lighten-2"
        text
        to="account"
      >
        Anmelden
      </v-btn>
    </v-card-actions>
  </v-card>
</template>

```

```

    </v-card-actions>
  </v-card>
</template>

<script>
import statesMixin from "@/mixins/statesMixin"

export default {
  mixins: [statesMixin],
  props: {
    encounters: {
      type: Number,
      required: true
    }
  },
  created() {
    this.setStateCount(2);
  }
};
</script>

<style>
</style>

```

Datei: PrivacyFooter.vue

```

<template>
  <div style="height: 52px" class="mt-2">
    <div
      class="background darken-2 d-flex py-4 px-4"
      style="position: absolute; left: 0; right: 0;">
      <a
        class="secondary--text text--lighten-2 body-2 px-2"
        :href="privacy_link"
        target="_blank">Datenschutzerklärung</a>

      <a
        class="secondary--text text--lighten-2 body-2 px-2"
        :href="imprint_link"
        target="_blank">Impressum</a>
    </div>

    <v-spacer />
  </div>
</template>

<script>
export default {
  computed: {
    privacy_link() {
      return process.env.VUE_APP_PRIVACY_LINK;
    },
  },
};

```

```

    imprint_link() {
      return process.env.VUE_APP_IMPRINT_LINK;
    }
  }
}
</script>

```

Datei: QrScanner.vue

```

<template>
  <Page ref="page" title="Raum QR-Code scannen"
    gradient
    color="transparent">

    <qrcode-stream @decode="onDecode" @init="onInit"></qrcode-stream>

    <div class="font-weight-light text-h4 text--secondary d-flex flex-column align-start justify-center"
      style="position: absolute; left: 0; right: 0; bottom: 0; top: 0;">
      <div v-if="error" class="px-4">
        <v-icon size="48" class="mb-4" color="error">warning</v-icon>
        <div>{{ error }}</div>
      </div>
      <template v-if="loading">
        <v-skeleton-loader
          type="image"
          width="100%"
          height="100%"
          max-width="100%"
          max-height="100%" />
        <v-progress-circular
          style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%)"
          indeterminate
          size="64"
        ></v-progress-circular>
      </template>
    </div>
  </Page>
</template>

<script>
import { QrCodeStream } from "vue-qrcode-reader";
import { setVisit } from "@requests/api";

export default {
  data: () => ({
    error: null,
    loading: true,
    hasCameraPermission: false
  }),
  created() {
    navigator.mediaDevices.enumerateDevices().then(
      (devices) => {
        const cameras = devices.filter(d => d.kind === 'videoinput');

```

```

        // If the camera has a label, then it has been given permission
        this.hasCameraPermission = cameras.some(c => c.label !== "");
    });
},
mounted() {

    // setTimeout(() => {
    //     setVisit('13-027').then(() => this.$refs.page.navigateAway());
    // }, 5000);
},
    methods: {
        async onInit(promise) {
            console.log("INIT");
            // show loading indicator
            this.loading = true;
            try {
                const { capabilities } = await promise;

                console.log("Capabilities", capabilities);
                // successfully initialized

                await this.getRooms();
            } catch (error) {
                console.log("CATCHING", error, JSON.stringify(error));
                if (error.name === "NotAllowedError") {
                    // user denied camera access permission
                    this.error = "Zugriff auf die Kamera wurde nicht erlaubt. Bitte aktivieren Sie dies in die
Einstellungen Ihren Browser."
                } else if (error.name === "NotFoundError") {
                    // no suitable camera device installed
                    this.error = "Keine Kamera gefunden."
                } else if (error.name === "NotReadableError") {
                    // maybe camera is already in use
                    this.error = "Es konnte nicht auf die Kamera zugegriffen werden."
                } else if (error.name === "OverconstrainedError") {
                    // did you requested the front camera although there is none?
                    this.error = "Vorderkamera nicht unterst\u00fctzt."
                } else if (error.name === "StreamApiNotSupportedError") {
                    // browser seems to be lacking features
                    this.error = "Ihren Browser wird nicht unterst\u00fctzt."
                }
            }

            // } else if (error.name === "NotSupportedError") {
            //     // page is not served over HTTPS (or localhost)
            // }
            } finally {
                // hide loading indicator
                this.loading = false;
            }
        },
        onDecode(string) {
            const room = this.rooms.find(room => {
                const room_without_name = room.split(' ')[0];

```

```

        return string.endsWith(room_without_name);
    });

    if (room) {
        // There's at least one
        console.log("ROOM FOUND!", room);
        setVisit(room).then(() => this.$refs.page.navigateAway());

        // this.$router.replace({ name: 'add', query: { room } });
    }

    else {
        console.log("Decoded", room, string);
    }
    },
},
    components: {
        QrcodeStream,
    },
};
</script>

<style scoped>
.v-toolbar {
    background: linear-gradient(180deg, rgb(0 0 0 / 60%) 0%, rgb(255 255 255 / 0%) 87%) !important;
}

</style>

<style lang="scss" scoped>
.qrcode-stream-wrapper {
    height: 100vh !important;
    width: 100vw !important;
    // margin-left: -16px;
    // margin-right: -16px;
}

::v-deep {
    &.v-toolbar__content {
        background: linear-gradient(180deg, rgb(0 0 0 / 60%) 0%, rgb(255 255 255 / 0%) 87%) !important;
    }

    &.v-skeleton-loader__image {
        height: 100%;
    }
}

// .v-toolbar {
//     background: linear-gradient(180deg, rgb(0 0 0 / 60%) 0%, rgb(255 255 255 / 0%) 87%) !important;
// }

</style>

```

Datei: RedirectIfAuthenticated.php

```
<?php

namespace App\Http\Middleware;

use App\Providers\RouteServiceProvider;
use Closure;
use Illuminate\Support\Facades\Auth;

class RedirectIfAuthenticated
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @param string|null $guard
     * @return mixed
     */
    public function handle($request, Closure $next, $guard = null)
    {
        if (Auth::guard($guard)->check()) {
            return redirect(RouteServiceProvider::HOME);
        }

        return $next($request);
    }
}
```

Datei: Register.vue

```
<template>
  <Page title="Anmelden" ref="page" :closable="!mustLogin" v-bind="{ ...$props, ...$attrs }">
    <template #extension>
      <v-tabs v-model="tab" background-color="transparent" >
        <v-tabs-slider color="accent"></v-tabs-slider>

        <v-tab active-class="white--text" v-for="item in items" :key="item">{{ item }}</v-tab>
      </v-tabs>
    </template>

    <div class="px-4">

      <div class="body-2 text--secondary my-6">
        Wir erheben und verarbeiten die dabei erhobenen personenbezogenen Daten gemäß Art. 5 und
        6 der DSGVO, § 5 und 6 BbgDSG nur für den Zweck der Kontaktnachverfolgung nach der
        Eindämmungsverordnung des Landes Brandenburg.
      <br /><br />
    </div>
  </Page>
</template>
```



Eine Zusammenfassung des zuständigen Ministeriums für Wissenschaft, Forschung und Kultur, unter welchen Bedingungen Präsenzveranstaltungen an Brandenburger Hochschulen durchgeführt werden können, finden Sie unter:

```
<br />
<a
  target="_blank"
  href="https://mwfk.brandenburg.de/mwfk/de/ministerium/umgang-mit-corona-
pandemie/">https://mwfk.brandenburg.de/mwfk/de/ministerium/umgang-mit-corona-pandemie/</a>
<br />(Abgerufen am 02.10.2020)
</div>
```

```
<v-tabs-items v-model="tab">
  <v-form ref="form" v-model="valid" @submit.prevent="submit">
    <v-tab-item :key="Login">
      <Login v-if="tab === 0" :ref="`formContent${tab}`" />
    </v-tab-item>
    <v-tab-item :key="Gast">
      <RegisterGuest v-if="tab === 1" :ref="`formContent${tab}`" />
    </v-tab-item>

    <input type="submit" name="update" value="Einloggen"
      style="position: absolute; height: 0px; width: 0px; border: none; padding: 0px;"
      hidefocus="true" tabindex="-1"/>
    </v-form>
  </v-tabs-items>
```

```
</div>
```

```
    <v-spacer />
    <v-card-actions>
      <LoadingButton
        :icon="tab === 0 ? 'account_circle' : undefined"
        :text="tab === 0 ? 'Einloggen' : undefined"

        :query="promise" @click="submit" />
    </v-card-actions>
```

```
<PrivacyFooter />
```

```
</Page>
```

```
</template>
```

```
<script>
```

```
import Login from "@components/register/Login";
import RegisterGuest from "@components/register/RegisterGuest";
```

```
export default {
  data() {
    return {
      valid: true,

      promise: null,
      closing: 0,

      tab: null,
      items: ["Login", "Gast"],
```

```

    },
    methods: {
      close() {
        this.$router.replace({ name: "home" });
      },
      async submit() {
        this.$refs.form.validate();

        await this.$nextTick();

        if (!this.valid) return;

        const ref = this.$refs[`formContent${this.tab}`];

        const promise = ref.getRequestPromise();
        promise.then((response) => {
          if (response) this.$refs.page.close();
        });

        this.promise = promise;
      },
    },
    components: {
      Login,
      RegisterGuest,
    },
  };
</script>

```

```

<style>
input,
input:-webkit-autofill,
input:-webkit-autofill:hover,
input:-webkit-autofill:focus {
  -webkit-box-shadow: 0 0 0 30px inherit inset;
}

```

```
</style>
```

```
<style lang="scss" scoped>
```

```

::v-deep {
  & .v-input__slot {
    margin-bottom: 8px !important;
  }
}
</style>

```

Datei: RegisterGuest.vue

```

<template>
  <div>
    <v-text-field
      filled
      type="text"
      label="Vorname"
      hide-details="auto"
      v-model="data.first_name"
      autocomplete="given-name"
      :rules="[rules.required]"
    />
    <v-text-field
      filled
      type="text"
      label="Nachname"
      hide-details="auto"
      v-model="data.last_name"
      autocomplete="family-name"
      :rules="[rules.required]"
    />
  </div>

```

```

<div class="body-2 text--secondary mt-6 mb-2">

```

Das Gesundheitsamt braucht ihren Telefonnummer oder E-Mail-Adresse um sich mit Ihnen in Kontakt zu setzen.

```

</div>
<v-select
  v-model="contact_type"
  filled
  :items=["E-Mail", "Telefon"]
  label="Kontakt"
  hide-details
></v-select>

  <v-text-field
    v-if="contact_type === 'E-Mail'"
    filled
    type="email"
    label="E-Mail"
    v-model="data.email"
    autocomplete="email"
    :error-count="errors.email.length || 1"
    :error-messages="errors.email"
    :rules="[rules.required, rules.validEmail]"
  />

  <v-text-field
    v-else
    filled
    type="tel"
    label="Telefon"
    v-model="telephone"
    autocomplete="tel"
  />

```

```

        :error-count="errors.telephone.length || 1"
        :error-messages="errors.telephone"
        :rules="[rules.required, rules.validTelephone]"
    />

    <v-checkbox v-model="data.accept" :rules="[rules.required]" hide-details>
        <template #label>
            <span class="text-body-2">Ich habe die Datenschutzbestimmungen zur
Kenntnis genommen.</span>
        </template>
    </v-checkbox>
</div>
</template>

<script>
import { register } from "@requests/api";
import { AsYouType } from "libphonenumber-js";

export default {
    data() {
        return {
            contact_type: 'E-Mail',
            rules: {
                required: (v) => !!v || "Feld erforderlich",
                validTelephone: () =>
                    this.phoneFormatter.isValid() || "Keine gültige Telefonnummer",
                validEmail: (v) => /^[^\s@]+@[^\s@]+\.[^\s@]+$/ .test(v) || 'Keine gültige E-Mail Adresse'
            },
            data: {
                first_name: null,
                last_name: null,
                telephone: null,
                email: null,
                accept: null,
            },
            errors: {
                telephone: [],
                email: []
            },
            phoneFormatter: new AsYouType("DE"),
        };
    },
    created() {
        this.data = {
            first_name: this.user.first_name,
            last_name: this.user.last_name,
            telephone: this.user.telephone,
            email: this.user.email,
        };

        this.telephone = this.user.telephone;
    },

```

```

    methods: {
      getRequestMethod() {
        if (this.contact_type === 'E-Mail') {
          this.data.telephone = null;
        } else {
          this.data.email = null;
        }
      }

      const promise = register(this.data);

      promise.catch((error) => {
        const error_bag = error?.response?.data?.errors || {};

        for (const [key, value] of Object.entries(error_bag)) {
          if (key in this.errors) {
            this.errors[key] = value;
          }
        }
      });

      return promise;
    },
  },

  computed: {
    telephone: {
      get() {
        return this.data.telephone;
      },
      set: function (telephone) {
        this.data.telephone = null;

        if (telephone === null) {
          return;
        }

        this.$nextTick(() => {
          this.phoneFormatter.reset();
          let formatted = this.phoneFormatter.input(telephone);
          this.data.telephone = formatted;
        });
      },
    },
  },

  watch: {
    "data.telephone": {
      this.errors.telephone = [];
    },
    "data.email": {
      this.errors.email = [];
    }
  },
};

```

```
</script>
```

```
<style>  
</style>
```

Datei: RegisterGuestWithAddress.vue

```
<template>  
  <div>  
    <v-text-field  
      filled  
      type="text"  
      label="Vorname"  
      hide-details="auto"  
      v-model="data.first_name"  
      autocomplete="given-name"  
      :rules="[rules.required]"  
    />  
    <v-text-field  
      filled  
      type="text"  
      label="Nachname"  
      hide-details="auto"  
      v-model="data.last_name"  
      autocomplete="family-name"  
      :rules="[rules.required]"  
    />  
  
    <v-text-field  
      filled  
      type="text"  
      label="Straße"  
      hide-details="auto"  
      v-model="data.street"  
      autocomplete="street-address"  
      :error-count="errors.street.length"  
      :error-messages="errors.street"  
      :rules="[rules.required]"  
    />  
  
    <v-text-field  
      filled  
      type="text"  
      label="Stadt"  
      hide-details="auto"  
      v-model="data.city"  
      autocomplete="address-level2"  
      :error-count="errors.city.length"  
      :error-messages="errors.city"  
      :rules="[rules.required]"  
    />  
  
    <v-text-field
```

```

        filled
        type="number"
        label="PLZ"
        hide-details="auto"
        v-model="data.postal_code"
        autocomplete="postal-code"
        :error-count="errors.postal_code.length"
        :error-messages="errors.postal_code"
        :rules="[rules.required]"
    />

    <v-text-field
        filled
        type="tel"
        label="Telefon"
        hide-details="auto"
        v-model="telephone"
        autocomplete="tel"
        :error-count="errors.telephone.length"
        :error-messages="errors.telephone"
        :rules="[rules.required, rules.validTelephone]"
    />

    <v-checkbox v-model="data.accept" :rules="[rules.required]" hide-details>
        <template #label>
            <span class="text-body-2">Ich habe die Datenschutzbestimmungen zur
Kenntnis genommen.</span>
        </template>
    </v-checkbox>
</div>
</template>

<script>
import { register } from "@requests/api";
import { AsYouType } from "libphonenumber-js";

export default {
    data() {
        return {
            rules: {
                required: (v) => !!v || "Feld erforderlich",
                validTelephone: () =>
                    this.phoneFormatter.isValid() ||
                    "Keine gültige Telefonnummer",
            },
            data: {
                first_name: null,
                last_name: null,
                telephone: null,
                street: null,
                city: null,
                postal_code: null,
                accept: null,
            },
        },
    },
};

```

```

        errors: {
            telephone: [],
            street: [],
            city: [],
            postal_code: [],
        },
        phoneFormatter: new AsYouType("DE"),
    };
},

created() {
    this.data = {
        first_name: this.user.first_name,
        last_name: this.user.last_name,
        telephone: this.user.telephone,
        street: this.user.street,
        city: this.user.city,
        postal_code: this.user.postal_code,
    };

    this.telephone = this.user.telephone;
},

methods: {
    getRequestPromise() {
const promise = register(this.data);

promise.catch((error) => {
    const error_bag = error?.response?.data?.errors || {};

    for (const [key, value] of Object.entries(error_bag)) {
        if (key in this.errors) {
            this.errors[key] = value;
        }
    }
});

return promise;
},
},

computed: {
    telephone: {
        get() {
            return this.data.telephone;
        },
        set: function (telephone) {
            this.data.telephone = null;

            if (telephone === null) {
                return;
            }

            this.$nextTick(() => {

```



```

        this.phoneFormatter.reset();
        let formatted = this.phoneFormatter.input(telephone);
        this.data.telephone = formatted;
    });
    },
},
},

watch: {
    "data.telephone"() {
        this.errors.telephone = [];
    },
    "data.street"() {
        this.errors.street = [];
    },
    "data.city"() {
        this.errors.city = [];
    },
    "data.postal_code"() {
        this.errors.postal_code = [];
    },
},
};
</script>

<style>
</style>

```

Datei: RegisterRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegisterRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
}

```

```

public function rules()
{
    $rules = [
        'accept' => ['required', 'accepted'],
        'first_name' => ['required', 'string', 'min:1'],
        'last_name' => ['required', 'string', 'min:1']
    ];

    if($this->filled('email')){
        $rules['email'] = ['required_if:telephone,null', 'email:rfc,dns,spoof', 'indisponible'];
    }
    elseif ($this->filled('telephone')) {
        $rules['telephone'] = ['required_if:email,null', 'phone:DE'];
    }

    return $rules;
}

public function withValidator($validator) {

    $validator->after(function ($validator) {
        if (!$this->filled('email') || $this->filled('telephone')) {
            $email = __('validation.attributes.email');
            $telephone = __('validation.attributes.telephone');

            $message = __('validation.required', ['attribute' => "$email oder $telephone"]);

            $validator->errors()->add('general', $message);
        }
    });
}

/**
 * Get the error messages for the defined validation rules.
 *
 * @return array
 */
public function messages()
{
    return [
        'accept.*' => 'Bitte akzeptieren Sie die Datenschutzbestimmungen'
    ];
}
}

```

Datei: RoomController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Helpers\RoomsHelper;

class RoomController extends Controller
{
    public function show() {
        return RoomsHelper::getRooms();
    }
}
```

Datei: RoomEnum.php

```
<?php
```

```
namespace App;
```

```
use App\Utils\BasicEnum;
```

```
class RoomEnum extends BasicEnum {
    const ROOMS = [
        "10-0-01 Bibliothek Eingang",
        "10-0.08 Besprechungsraum",
        "10-008 Vortragsraum",
        "10-011 Lager Verkaufswaren",
        "10-012 Stuhllager",
        "10-025 Arbeitsraum Küchenleiter",
        "10-026 Kalte Küche",
        "10-027 Garküche",
        "10-028 Topfspüle",
        "10-029 Flur Küche",
        "10-030 Lager Getränke",
        "10-102 Ausleihe",
        "10-103 Schließfächer",
        "10-104 Foyer Bibliothek",
        "10-108 Erste Hilfe / Teeküche",
        "10-111 Kopierraum",
        "10-2.08 Besprechungsraum",
        "10-2.18 Besprechungsraum",
        "10-201 Arbeitsraum",
        "10-203 Arbeitsraum",
        "10-213 Ltr. Bibliothek",
        "10-215 Arbeitsraum",
        "10-216 Auskunft",
        "10-218 Mediathek",
        "10-3.02 Besprechungsraum",
        "10-3.12 Besprechungsraum",
        "10-3.13 Besprechungsraum",
        "10-3.15 Besprechungsraum",
        "10-301 Carrel",
        "10-302 Carrel",
        "10-313 Carrel",
        "10-315 Carrel",
        "10-K01 Geschirreinigung",
    ]
}
```

"10-K03 Abstellraum Leergut",  
"10-K04 Abstellraum Pumi",  
"10-K05 Lüftung Mensa",  
"10-K06 Heizung",  
"10-K08 Sprinklerzentrale",  
"10-K09 ELT",  
"10-K10 Batterie",  
"10-K11 EDV-Server",  
"10-K12 Lagerraum Pumi",  
"10-K16 Lüftungszentrale",  
"13-001 Haupteingang / Empfang",  
"13-002 Arbeitsraum",  
"13-003 Besprechungsraum",  
"13-004 WIT",  
"13-005 Büro WIT",  
"13-006 Teeküche WIT",  
"13-007 Arbeitsraum",  
"13-010 Eltern-Kind-Zimmer",  
"13-010a Lehrmittelraum",  
"13-013 Kopiererei",  
"13-014 Kopiererei",  
"13-015 Kopiererei",  
"13-016 Abstellraum",  
"13-017 Arbeitsraum",  
"13-018 Sportraum",  
"13-019 I-undP-Amt",  
"13-020 I-undP-Amt",  
"13-021 I-undP-Amt",  
"13-022 Leiter I-undP-Amt",  
"13-023 Flur/Vorraum",  
"13-024 I-undP-Amt",  
"13-025 I- und P-Amt",  
"13-026 I-undP-Amt",  
"13-027 Seminarraum",  
"13-028 Büro",  
"13-029 Besprechungsraum",  
"13-031 Arbeitsraum",  
"13-032 Arbeitsraum",  
"13-033 Arbeitsraum",  
"13-034 Arbeitsraum",  
"13-035 Arbeitsraum",  
"13-036 Seminarraum",  
"13-037 Seminarraum",  
"13-038 Arbeitsraum",  
"13-039 Arbeitsraum",  
"13-040 Arbeitsraum",  
"13-041 TIB-Stelle/SeWiTec",  
"13-043 TIB-Stelle",  
"13-044 Arbeitsraum",  
"13-045 Arbeitsraum",  
"13-047 Empfang / Pförtner",  
"13-052 betriebstechn. Anlagen",  
"13-053 Stiller Raum",  
"13-054 Seminarraum",

"13-101 Arbeitsraum Kanzler",  
"13-102 Sekretariat Präsidentin",  
"13-103 Arbeitsraum Präsidentin",  
"13-104 Senatssaal",  
"13-105 Teeküche",  
"13-106 Arbeitsraum",  
"13-108 Arbeitsraum",  
"13-109 Ltr. Bauangelegenheiten",  
"13-110 Sachbearb. Bau",  
"13-111 Ltr.Techn.Hausverw.",  
"13-113 Arbeitsraum",  
"13-114 Arbeitsraum",  
"13-115 Arbeitsraum",  
"13-116 Labor Workstation",  
"13-117 Ltr. HRZ",  
"13-118 betriebstechn.Anlagen",  
"13-119 Arbeitsraum",  
"13-120 Arbeitsraum",  
"13-121 Arbeitsraum",  
"13-123 Seminarraum",  
"13-125 Arbeitsraum",  
"13-126 Arbeitsraum",  
"13-127 Arbeitsraum",  
"13-128 Ltr Org./Controlling",  
"13-129 Arbeitsraum",  
"13-130 Arbeitsraum",  
"13-131 Arbeitsraum",  
"13-132 Arbeitsraum",  
"13-133 Arbeitsraum",  
"13-134 Arbeitsraum",  
"13-135 Ltr. Personal",  
"13-136 Arbeitsraum",  
"13-136A Arbeitsraum",  
"13-137 Arbeitsraum",  
"13-138 Arbeitsraum",  
"13-139 Arbeitsraum",  
"13-140 Arbeitsraum",  
"13-141 Arbeitsraum",  
"13-142 Besprechungsraum",  
"13-143 Arbeitsraum",  
"13-147 Abstellraum Reinigung",  
"13-149 Arbeitsraum",  
"13-150 HRZ Supportcenter",  
"13-151 Arbeitsraum",  
"13-152 Arbeitsraum",  
"13-156 Teeküche",  
"13-157 Arbeitsraum",  
"13-200 Technisches Labor",  
"13-200A Labor Brennstoffzelle",  
"13-200B Arbeitsraum",  
"13-200C Arbeitsraum",  
"13-201 Besprechungsraum",  
"13-202 EDV-Serverraum",  
"13-203 Personalrat",

"13-204 Arbeitsraum",  
"13-205 Arbeitsraum",  
"13-206 Arbeitsraum",  
"13-207 Arbeitsraum",  
"13-208 Graduate School",  
"13-209 Arbeitsraum",  
"13-210 Arbeitsraum",  
"13-211 Arbeitsraum",  
"13-212 Seminarraum",  
"13-213 Arbeitsraum",  
"13-214 Arbeitsraum",  
"13-214A Serverraum",  
"13-215 Arbeitsraum",  
"13-216 Ltr. Hochschulmarketing",  
"13-217 Systemakkreditierung",  
"13-218 Arbeitsraum",  
"13-219 Arbeitsraum",  
"13-220 TQM",  
"13-221 Zentrum f Qualitaetssicher",  
"13-222 Zentr f Qualitaetssicher",  
"13-223 Zentrum f Qualitaetss",  
"13-224 Arbeitsraum",  
"13-225 Besprechungsraum",  
"13-226 Regenerative Energien",  
"13-227 Arbeitsraum",  
"13-230 Teeküche",  
"13-231 Reinigung",  
"13-252A ELT",  
"13-K01 Konferenzraum",  
"13-K02 Abstellraum",  
"13-K05 Abstellraum",  
"13-K07 Abstellraum",  
"13-K08 Abstellraum",  
"13-K09 Abstellraum",  
"13-K10 Abstellraum",  
"13-K11 Umkleideraum",  
"13-K13 Umkleideraum",  
"13-K14 Abstellraum",  
"13-K15 Abstellraum",  
"13-K16 Abstellraum",  
"13-K17 Abstellraum",  
"13-K18 HA-Raum Trinkwasser",  
"13-K19 Abstellraum",  
"13-K20 Abstellraum",  
"13-K21 Abstellraum",  
"13-K22 Abstellraum",  
"13-K23 Abstellraum",  
"13-K24 Abstellraum",  
"13-K25 Abstellraum",  
"13-K26 Abstellraum",  
"13-K27 Abstellraum",  
"13-K28 Abstellraum",  
"13-K30 Klimatechnik",  
"13-K32 Kältezentrale",

"13-K33 Abstellraum",  
"13-K33A Heizung",  
"13-K35 Abstellraum",  
"13-K39 Aufzugsmaschinenraum",  
"13-K45 Abstellraum",  
"13-K46A Elektroeinspeisung",  
"13-K46B HRZ",  
"13-K48 EDV-Anschlussraum",  
"13-K51 Abstellraum",  
"13-K52 Abstellraum",  
"13-K53 Flur",  
"13-K54 Abstellraum",  
"13-K55 Abstellraum",  
"13-K56 Fernmeldetechnik",  
"13-K57 Abstellraum",  
"13-K58 Abstellraum",  
"14-A001 Hörsaal",  
"14-A001A Elektro FM",  
"14-A002 Regieraum",  
"14-A003 Reinigung",  
"14-A004 Hausmeister / Erste-Hilfe",  
"14-A005 Lab. Fertigungsmesstechn",  
"14-A006 Lab. Werkzeugkonstruktion",  
"14-A007 EDV-Anwendung/RIB",  
"14-A008 Lab. Trenntechnik",  
"14-A009 Lab. Trenntechnik",  
"14-A010 Schweißtechnik/Therm.Spri",  
"14-A011 Labor Therm. Spritzen",  
"14-A012 Werkstoffprüftechnik II",  
"14-A013 Werkstoffprüftechnik I",  
"14-A014 Laser- u. Plasmatechnik",  
"14-A104 Lab. f. NatKom / RFID",  
"14-A105 Software Engineering I",  
"14-A106 Labor Telematik",  
"14-A106A Vorbereitungsraum",  
"14-A107 Software Engineering II",  
"14-A108 Telematikanwendungen",  
"14-A109 Büroraum Laboringenieur",  
"14-A110 Phys. Grundlagen",  
"14-A111 Lab Laser-Plasmatechnik",  
"14-A112 Büroraum Laboringenieur",  
"14-A113 Sozialraum",  
"14-A114 Büroraum",  
"14-A115 Elektrotechnik / EMA",  
"14-A201 Arbeitsraum",  
"14-A202 Arbeitsraum",  
"14-A203 Arbeitsraum",  
"14-A204 Arbeitsraum",  
"14-A205 Arbeitsraum",  
"14-A206 Arbeitsraum",  
"14-A207 Arbeitsraum",  
"14-A208 Arbeitsraum",  
"14-A209 Arbeitsraum",  
"14-A210 Arbeitsraum",

"14-A211 Arbeitsraum",  
"14-A212 Arbeitsraum",  
"14-A214 Kopierraum",  
"14-A215 Arbeitsraum",  
"14-A216 Arbeitsraum",  
"14-A217 Arbeitsraum",  
"14-A218 Arbeitsraum",  
"14-A219 Arbeitsraum",  
"14-A220 Arbeitsraum",  
"14-A221 Sozialraum",  
"14-A223 Arbeitsraum",  
"14-A224 Arbeitsraum",  
"14-A225 Arbeitsraum",  
"14-A226 Arbeitsraum",  
"14-A227 Arbeitsraum",  
"14-A228 Arbeitsraum",  
"14-A229 Arbeitsraum",  
"14-A230 Arbeitsraum",  
"14-A231 Arbeitsraum",  
"14-A232 Arbeitsraum",  
"14-A233 Arbeitsraum",  
"14-A234 Arbeitsraum",  
"14-AK01 Sprinklerzentrale",  
"14-AK01A Wasserspeicher",  
"14-AK02 Lüftungstechnik",  
"14-AK03 Druckluft / Heizung",  
"14-AK04 Abstellraum",  
"14-AK05 Elt 2",  
"14-AK06 Batterie",  
"14-AK07 IT",  
"14-AK08 IT",  
"14-AK09 Elt 1",  
"14-AK10 SWH + KWS",  
"14-AK15 Kaltwassersatz",  
"14-AK16 Pumi",  
"14-AK17 Aufzugstechnik",  
"14-B001 Seminarraum",  
"14-B002 Seminarraum",  
"14-B003 Lab.f.FMT/QM",  
"14-B004 Produktionsvorbereitung",  
"14-B005 Lab. Fabrikplanung",  
"14-B006 Laser-/Plasmatechnik",  
"14-B007 Oberflächentechnik",  
"14-B008 Rasterelektronenmikroskop",  
"14-B009 Lab. f. Materialographie",  
"14-B009A Lab. f. Materialografie",  
"14-B009B Lab. f. Materialografie",  
"14-B010 Oberflächentechnik",  
"14-B101 Seminarraum",  
"14-B102 Lab. Bildverarbeitung",  
"14-B103 Dozentenraum",  
"14-B104 Lab. f.Konstruktion/CAD",  
"14-B105 Lab. f. Konstruktion/CAD",  
"14-B106 Lab. f. Konstruktion/CAD",



"14-B107 Phys. GL / reg. Energiet.",  
"15-003 Lab. Automatisierungstech",  
"15-004 Aufzugsanlagentechnikraum",  
"15-005 Lab. f. Maschinendynamik",  
"15-008 Lab. f. Maschinendynamik",  
"15-009 Technikum Therm. VT",  
"15-010 Raum f.Heizg.",  
"15-012 Techn.Thermische VT Ex",  
"15-013 Gefahrstofflager",  
"15-015 Putzraum",  
"15-022 Labor Biosystemtechnik",  
"15-022a Lab. Biosystemtechnik",  
"15-023 Labor Biosystemtechnik",  
"15-025 NRT",  
"15-026 Labor Biosystemtechnik",  
"15-027 NS",  
"15-028 Notstromversorgung",  
"15-029 MS",  
"15-030 Trafo",  
"15-031 Trafo",  
"15-033 Gas",  
"15-034 Heizraum",  
"15-035 Lab.f.Biosystemtechnik",  
"15-036 Lab.f.Biosystemtechnik",  
"15-038 Lab.f.Biosystemtechnik",  
"15-039 Abstellraum",  
"15-041 Lab. Mikrosystemtechnik",  
"15-043 Abstellraum/Betriebst.",  
"15-044 Lagerraum Mikrosystemtech",  
"15-045 Abstellr Mikrosystemtech",  
"15-046 Labor",  
"15-047 Abstellraum",  
"15-049 Reinraumlabor",  
"15-050 Labor Mikrosystemtechnik",  
"15-051 Chemikalienlager",  
"15-052 Lab.f.Materialographie",  
"15-053 Lab.f.Aufbereit.u.Kreisl.",  
"15-054 Arbeitsraum",  
"15-055 Heizraum",  
"15-101 Besprechungsraum",  
"15-103 Besprechungsraum",  
"15-104 Meßwarte / Arbeitsraum",  
"15-105 Schleuse",  
"15-107 Lab. f. PAA",  
"15-111 Putzraum",  
"15-112 Teeküche",  
"15-113 Fachbereichssprecher",  
"15-114 Abstellraum",  
"15-116 ELT",  
"15-117 Fachbereichsverwaltung",  
"15-118 Fachbereichsverwaltung",  
"15-119 Lab.f. PAA",  
"15-121 Lab.Prozeß-u.Anlagenautom",  
"15-122 Serverraum",

"15-123 Serverraum",  
"15-124 Lab.f.anorg.Chemie",  
"15-125 Glaslager",  
"15-127 Arbeitsraum",  
"15-128 Arbeitsraum",  
"15-129 Arbeitsraum",  
"15-130 Arbeitsraum",  
"15-131 Abstellraum",  
"15-132 ELT",  
"15-133 Besprechungsraum",  
"15-136 Arbeitsraum",  
"15-137 Meßraum",  
"15-137a Abstellraum",  
"15-138 Chemielager 2",  
"15-139 Chemielager 1",  
"15-140 Lab.f.org.Chemie",  
"15-143 Labor",  
"15-144 Labor",  
"15-145 Labor",  
"15-147 Arbeitsraum",  
"15-148 Arbeitsraum",  
"15-149 Arbeitsraum",  
"15-150 ELT",  
"15-151 Labor",  
"15-151a Kühlraum",  
"15-155 Arbeitsraum",  
"15-156 Arbeitsraum",  
"15-157 Labor",  
"15-158 Labor",  
"15-160 ELT",  
"15-201 Seminarraum",  
"15-202 Seminarraum",  
"15-203 Seminarraum",  
"15-204 Seminarraum",  
"15-205 Arbeitsraum",  
"15-207 Putzraum",  
"15-211 Arbeitsraum",  
"15-212 Arbeitsraum",  
"15-213 Arbeitsraum",  
"15-214 ELT",  
"15-215 Abstellraum",  
"15-216 Arbeitsraum",  
"15-217 Arbeitsraum",  
"15-219 Arbeitsraum",  
"15-220 Arbeitsraum",  
"15-221 Abstellraum",  
"15-222 Arbeitsraum",  
"15-223 Arbeitsraum",  
"15-225 Abstellraum",  
"15-226 Arbeitsraum",  
"15-227 Arbeitsraum",  
"15-228 Arbeitsraum",  
"15-229 Arbeitsraum",  
"15-230 ELT",

"15-233 Arbeitsraum",  
"15-234 Arbeitsraum",  
"15-K00 Betriebstechnische Anlage",  
"15-K01 Betriebstechnische Anlage",  
"15-K02 Zuluft/Abluft",  
"15-K05 Sanitär",  
"15-K06 Pumi",  
"15-K07 Abstellraum",  
"15-K08 Betriebstechnische Anlage",  
"16-0002 Technikzentrale",  
"16-0003 Brandmeldezentrale",  
"16-0004 Lager",  
"16-0005 Lager",  
"16-0006 Teeküche",  
"16-0009 Aufenthaltsraum Campulino",  
"16-0010 Teeküche Campulino",  
"16-0011 Lager Campulino",  
"16-0014 Wickelraum Campulino",  
"16-0015 Stillen/1.Hilfe Campulino",  
"16-0030 Seminarraum",  
"16-0031 Büro",  
"16-0032 Büro",  
"16-0033 Büro",  
"16-0034 Lab f Autom. LifeSciences",  
"16-0035 Labor Molekularbiologie",  
"16-0036 Lab f Molekulare Biotechn",  
"16-0037 Zellkultur",  
"16-0038 Labor Molekularbiologie",  
"16-0039 Lager",  
"16-0040 Teeküche",  
"16-0041 Abstellraum",  
"16-0042 Elektrophorese",  
"16-0044 Lager",  
"16-0045 Kühlraum",  
"16-0046 Labor f.Bioprozesstechnik",  
"16-0047 Labor f.Bioprozesstechnik",  
"16-0054 Hochschulrechenzentrum",  
"16-0055 Demonstrationswindkanal",  
"16-0056 Flugsimulator",  
"16-0057 Luftfahrttechnik",  
"16-0060 Lager",  
"16-0061 Kopierraum",  
"16-0062 Labor Produktionslogistik",  
"16-0063 Besprechungsraum",  
"16-0064 E-Techniklabor",  
"16-0065 Mechanikwerkstatt",  
"16-0066 Harzwerkstatt",  
"16-0067 Labor Avionik",  
"16-0068 Triebwerkslabor",  
"16-0069 Labor Luftfahrttechnik",  
"16-0075 Reinigung",  
"16-0076 FabLab Kreativlabor",  
"16-0080 Labor Intralogistik",  
"16-0081 Labor Verkehrslogistik",

"16-0082 Lager",  
"16-0083 Kopierraum",  
"16-0084 Teeküche",  
"16-0085 Büro",  
"16-0086 Büro",  
"16-0087 Büro",  
"16-0088 Büro",  
"16-0089 Büro",  
"16-0090 Büro",  
"16-0091 Büro",  
"16-0092 Büro",  
"16-0093 Seminarraum",  
"16-0095 Seminarraum",  
"16-0100 TRH 01",  
"16-0200 TRH 02",  
"16-0300 TRH 03",  
"16-0400 TRH 04",  
"16-0500 TRH 05",  
"16-0600 TRH 06",  
"16-1001 Labor Bioinformatik",  
"16-1002 Labor Bioinformatik",  
"16-1003 Büro",  
"16-1004 Büro",  
"16-1005 Besprechungsraum",  
"16-1006 Büro",  
"16-1007 Büro",  
"16-1008 Büro",  
"16-1009 Labor Bioinformatik",  
"16-1010 Teeküche",  
"16-1011 Kopierraum",  
"16-1012 Reinigung",  
"16-1013 Büro",  
"16-1014 Büro",  
"16-1022 Reinigung",  
"16-1030 Büro",  
"16-1031 Büro",  
"16-1032 Büro",  
"16-1033 Labor CAD/CAM",  
"16-1034 Labor Bioinformatik",  
"16-1035 Seminarraum",  
"16-1036 Lager",  
"16-1037 Kopierraum",  
"16-1038 Teeküche",  
"16-1039 Gruppenraum",  
"16-1040 Labor Bioinformatik",  
"16-1041 Labview/Logistikelemente",  
"16-1042 AutoID-/RFID-Testcenter",  
"16-1043 Seminarraum",  
"16-1044 Seminarraum",  
"16-1045 Seminarraum",  
"16-1054 Erste-Hilfe-Raum",  
"16-1055 Seminarraum",  
"16-1060 Büro",  
"16-1061 Büro",

"16-1062 Gruppenraum",  
"16-1063 Gruppenraum",  
"16-1064 Teeküche",  
"16-1065 Kopierraum",  
"16-1074 Reinigung",  
"16-1075 Büro",  
"16-1081 Labor CAD/CAM",  
"16-1082 Lager",  
"16-1083 Büro",  
"16-1084 Büro",  
"16-1085 Büro",  
"16-1086 Büro",  
"16-1087 Büro",  
"16-1088 Büro",  
"16-1089 Büro",  
"16-1090 Büro",  
"16-1091 Büro",  
"16-1092 Büro",  
"16-1093 Büro",  
"16-1094 Seminarraum",  
"16-1095 Seminarraum",  
"16-1096 Multimedia-Management",  
"16-1097 Multimedia-Management",  
"16-1100 TRH 01",  
"16-1200 TRH 02",  
"16-1300 TRH 03",  
"16-1400 TRH 04",  
"16-1500 TRH 05",  
"16-1600 TRH 06",  
"16-2001 Büro",  
"16-2002 Büro",  
"16-2003 Büro",  
"16-2004 Büro",  
"16-2005 Besprechung",  
"16-2006 Büro",  
"16-2007 Büro",  
"16-2008 Büro",  
"16-2009 Erste-Hilfe-Raum",  
"16-2010 Teeküche",  
"16-2011 Kopierraum",  
"16-2012 Pumi",  
"16-2013 Büro",  
"16-2014 Büro",  
"16-2022 Kopierraum",  
"16-2030 Büro",  
"16-2031 Büro",  
"16-2032 Büro",  
"16-2033 Arbeitsraum",  
"16-2034 Seminarraum",  
"16-2035 Genomiklabor",  
"16-2036 Proteomiklabor",  
"16-2037 Pathogenlabor",  
"16-2038 Archiv",  
"16-2039 Archiv",

"16-2040 Seminarraum",  
"16-2041 Seminarraum",  
"16-2042 Teeküche",  
"16-2070 THBV",  
"16-2075 Büro",  
"16-2076 Büro",  
"16-2077 Büro",  
"16-2078 Teeküche",  
"16-2080 Büro",  
"16-2081 Büro",  
"16-2082 Büro",  
"16-2083 Büro",  
"16-2084 Büro",  
"16-2085 Büro",  
"16-2086 Büro",  
"16-2087 EDV-Labor",  
"16-2088 Seminarraum",  
"16-2089 Gründungsservice",  
"16-2090 Arbeitsraum",  
"16-2091 Arbeitsraum",  
"16-2092 Büro",  
"16-2093 Besprechung",  
"16-2094 Seminarraum",  
"16-2095 VINN:Lab",  
"16-2100 TRH 01",  
"16-2200 TRH 02",  
"16-2300 TRH 03",  
"16-2500 TRH 05",  
"16-2600 TRH 06",  
"16-K001 Lager",  
"16-K002 Tank Sprinklierung H17",  
"16-K003 Sprinklerzentrale",  
"16-K004 Technik 1",  
"16-K005 Heizungszentrale",  
"16-K006 Datenverteiler 1",  
"16-K007 Elektro 3",  
"16-K008 Elektro 2",  
"16-K009 Elektro 1",  
"16-K010 Hausanschlußraum",  
"16-K080 Datenverteiler 2",  
"16-K081 Serverraum",  
"16-K082 Lüftungszentrale",  
"16-K200 TRH 2",  
"16-K203 Technik 1",  
"16-K300 TRH 3",  
"16-K401 Mediengang",  
"16-K501 Mediengang",  
"16-K600 TRH 6",  
"17-0001 Auditorium Maximum",  
"17-0002 Datenraum 1",  
"17-0004 Lüftungszentrale",  
"17-0005 Lüftungszentrale",  
"17-0006 FIBS",  
"17-0020 Seminarraum",

"17-0021 Seminarraum",  
"17-0022 Büro",  
"17-0023 Lager",  
"17-0024 Servicecenter",  
"17-0025 Büro",  
"17-0026 ELT 1",  
"17-0030 Seminarraum",  
"17-0031 Reinigung",  
"17-0032 Hausanschlussraum",  
"17-0033 Datenraum 2",  
"17-0034 Video",  
"17-0035 Sistro",  
"17-0036 ELT",  
"17-0037 SiBel",  
"17-0100 TRH 1",  
"17-0101 Windfang",  
"17-0102 Foyer 1",  
"17-0104 Foyer 2",  
"17-0200 TRH 2",  
"17-1001 Foyer Auditorium",  
"17-1002 Regieraum",  
"17-1003 Fortluft",  
"17-1004 Seminarraum",  
"17-1020 Seminarraum",  
"17-1021 Seminarraum",  
"17-1022 Seminarraum",  
"17-1023 Büro",  
"17-1024 Büro",  
"17-1025 Büro",  
"17-1026 Kopierraum",  
"17-1027 Büro",  
"17-1028 Büro",  
"17-1029 ELT 2",  
"17-1030 Teeküche",  
"17-1100 TRH 1",  
"17-1200 TRH 2",  
"17-1300 TRH 3",  
"17-2031 Technikraum",  
"17-2032 Lüftungszentrale 2A",  
"17-2300 TRH 3",  
"17-2301 Technikgeschoß",  
"17-2302 Technikgeschoß",  
"17-2303 Installationsfläche 2.OG",  
"19-001 Kita Gruppenraum 1",  
"19-002 Kita Gruppenraum 2",  
"19-003 Kita Teeküche",  
"19-004 Kita Gruppenraum 3",  
"19-005 Kita Sanitärraum",  
"19-020 Studentenclub",  
"19-021 Studentenclub - Garderobe",  
"19-026 Studentenclub Lager",  
"19-030 TRH 2",  
"19-031 Pumi",  
"19-035 Teeküche",

"19-037 Dozentenraum",  
"19-038 Büro",  
"19-039 Besprechungsraum",  
"19-050 TRH 1",  
"19-120 Studentenclub",  
"19-130 TRH 2",  
"19-133 Dekanat FB INW",  
"19-134 Dekanat FB INW",  
"19-135 Dekanat",  
"19-138 Studierendenrat",  
"19-139 Büro",  
"19-140 Büro",  
"19-141 Büro",  
"19-144 Abstellraum",  
"19-150 TRH 1",  
"19-K01 Batterieraum",  
"19-K02 ELT-Raum",  
"19-K03 Heizung",  
"19-K04 Lager",  
"19-K07 Lager Kita",  
"19-K08 BMA",  
"19-K09 Serverraum",  
"19-K30 TRH 2",  
"100-U01 Seminarraum",  
"100-U03 Arbeitsraum",  
"100-U04 Abstellraum",  
"100-U05 Seminarraum",  
"100-U06 Küche",  
"100-U07 Wasserversorgungsraum",  
"100-U08 EDV-Technik",  
"100-U10 Aufzugsmaschinenraum",  
"100-U11 HLS-Raum",  
"100-U12 Elektro-Installation",  
"100-U13 Serverraum",  
"100-U14 Seminarraum",  
"100-U15 Seminarraum",  
"100-001 Seminarraum",  
"100-003 Arbeitsraum",  
"100-004 Arbeitsraum",  
"100-005 Arbeitsraum",  
"100-006 Arbeitsraum",  
"100-007 Arbeitsraum",  
"100-008 Arbeitsraum",  
"100-014 Lab.f.Software E/D",  
"100-015 Vorbereitungsraum",  
"100-019 Verteilerraum",  
"100-020 Labor f. Programmierung",  
"100-021 Labor f. Kommunikation",  
"100-101 Seminarraum",  
"100-103 Arbeitsraum",  
"100-104 Arbeitsraum",  
"100-105 Arbeitsraum",  
"100-106 Arbeitsraum",  
"100-107 Arbeitsraum",



"100-108 Arbeitsraum",  
"100-109 Besprechungsraum",  
"100-115 Lab. E-Learning u Mathe",  
"100-116 Vorbereitungsraum",  
"100-119 Verteilerraum",  
"100-120 Kopierraum",  
"100-121 Labor f. Medientechnik",  
"100-122 Lab. Medienintegr. Verw.",  
"100-201 Seminarraum",  
"100-203 Arbeitsraum",  
"100-204 Arbeitsraum",  
"100-205 Arbeitsraum",  
"100-206 Dekanat / Dekan",  
"100-207 Besprechung Dekanat",  
"100-208 Dekanat / Sekretariat",  
"100-214 Seminarraum multimedial",  
"100-215 Vorbereitungsraum",  
"100-218 Verteilerraum",  
"100-219 Teeküche Dekanat",  
"100-220 Arbeitsraum",  
"100-221 Arbeitsraum",  
"100-222 Seminarraum",  
"100-223 EDV-Labor",  
"100-301 Seminarraum",  
"100-303 Arbeitsraum",  
"100-304 Arbeitsraum",  
"100-305 Arbeitsraum",  
"100-306 Arbeitsraum",  
"100-307 Arbeitsraum",  
"100-308 Arbeitsraum",  
"100-309A Arbeitsraum",  
"100-309B Arbeitsraum",  
"100-316 Verteilerraum",  
"100-317 Teeküche",  
"100-318 Arbeitsraum",  
"100-319 Arbeitsraum",  
"100-320 Seminarraum",  
"100-321 Lab.f.Inform./Netzwerke.",  
"100-401 Seminarraum",  
"100-403 Dozentenraum",  
"100-404 Arbeitsraum",  
"100-405 Arbeitsraum",  
"100-406 Arbeitsraum",  
"100-407 Arbeitsraum",  
"100-408 Sprachlabor",  
"100-415 Verteilerraum",  
"100-416 Teeküche",  
"100-417 Arbeitsraum",  
"100-418 Arbeitsraum",  
"100-419 Mediothek",  
"100-420 Seminarraum",

];  
}

Datei: RoomItem.vue

```
<template>
  <v-list-item>
    <v-list-item-action>
      <v-icon>meeting_room</v-icon>
    </v-list-item-action>

    <v-list-item-content id="room_input">
      <v-autocomplete
        label="Raum"
        name="room"
        ref="room_selection"

        v-model="room"
        :items="rooms || []"
        :loading="!rooms || !rooms.length"
        :clearable="true"
        :rules="[v => !!v || 'Bitte wählen Sie einen Raum aus']"

        hide-no-data
        return-object
        hide-details="auto"
        :menu-props="select_props"
        :attach="#room_input"

        :search-input.sync="search"
        :filter="filterRooms"

        @click:clear="room = null"
        @blur="closeMenu"
        @keydown.enter="nextOrSubmit"
      >
      <template v-slot:item="{ parent, item }">
        <v-list-item-content>
          <v-list-item-title
            v-html="parent.genFilteredText(item.split(' ')[0])" />
          <v-list-item-subtitle
            v-html="parent.genFilteredText(item.split(' ').slice(1).join(' ').replace(/[/]/g, ''))" />
          </v-list-item-content>
        </template>
      </v-autocomplete>
    </v-list-item-content>
  </v-list-item>
</template>

<script>
import Bowser from "bowser";

export default {
  props: {
    value: String
  },
}
```

```

data: () => ({
  maxHeight: 0,
  search: "",
  homepage_scroll_position: null
}),

created() {
  this.maxHeight = (this.$vuetify.breakpoint.height / 3) - 16;
  this.browser = Browser.getParser(window.navigator.userAgent);
  console.log("Browser", this.browser);
},

mounted() {
  // Because Safari cannot set our page as fixed, it scrolls with the background
  // we set the background to scroll as far as possible and then scroll back up
  // later when the user focuses the room input field.
  if (this.isSafari) {
    this.homepage_scroll_position = window.scrollY;
    // Get scroll position, then return to old scroll position on close.
    setTimeout(() => this.$vuetify.goTo(1000, { duration: 0 }), 250);
  }
},

beforeDestroy() {
  if (this.isSafari) {
    // Get scroll position, then return to old scroll position on close.
    this.$vuetify.goTo(this.homepage_scroll_position, { duration: 0 });
  }
},

methods: {
  filterRooms(item, query_text, item_text) {
    const query = query_text.replace(/[()]/g, "");
    const text = item_text.replace(/[()]/g, "");
    return text.toLocaleLowerCase().indexOf(query.toLocaleLowerCase()) > -1
  },

  closeMenu() {
    this.$refs.room_selection.menuIsActive = false;
  },

  nextOrSubmit(e) {

    if (this.browser.getPlatformType() === 'desktop') {
      e.preventDefault();
      e.stopPropagation();
      return;
    }

    var focussableElements = 'input:not([type=hidden]):not([disabled])';

    var elements = [...document.activeElement.form.querySelectorAll(focussableElements)];

```

```

const next_index = elements.indexOf(document.activeElement) + 1;
const next_element = elements[next_index] || null;

if (next_element) {
  this.$refs.room_selection.blur();
  next_element.focus();

  e.preventDefault();
  e.stopPropagation();
}
}
},

computed: {
  select_props() {
    let p = {
      nudgeLeft: -72
    };
    if (this.$vuetify.breakpoint.xs) {
      return {
        ...p,
        top: true,
        maxHeight: this.maxHeight, // (this.$vuetify.breakpoint.height / 3) - 16,
      };
    } else {
      return {
        ...p,
        nudgeTop: '-16px',
      };
    }
  },

  small() {
    return this.$vuetify.breakpoint.xs;
  },

  room: {
    set(room) {
      this.$emit('input', room);
    },
    get() {
      return this.value;
    }
  },

  isSafari() {
    return this.browser.satisfies({
      mobile: {
        safari: '>=11.3',
      }
    });
  }
},

watch: {
  search(val) {

```

```

console.log("Search changed");
const target = document.getElementById('room_input');
// 553 => 137 => height / 4 => (553/3)-48
// 719 => 179 => => (553/3)-48 = 191

if (val && val.length === 1) {

    if (this.isSafari) {
        this.$vuetify.goTo(target, {
            duration: 350,
            offset: this.$vuetify.breakpoint.height / 4
        });
    } else {
        target.scrollIntoView(false);
    }
}
}
}
}
}
}
</script>

<style>

</style>

```

Datei: RoomResource.php

```

<?php

namespace App\Http\Resources;

use Illuminate\Http\Resources\Json\JsonResource;

class RoomResource extends JsonResource
{
    /**
     * Transform the resource into an array.
     *
     * @param \Illuminate\Http\Request $request
     * @return array
     */
    public function toArray($request)
    {
        return parent::toArray($request);
    }
}

```

Datei: RoomResourceCollection.php

```

<?php

```

```

namespace App\Http\Resources;

use Illuminate\Http\Resources\Json\ResourceCollection;

class RoomResourceCollection extends ResourceCollection
{
    /**
     * Transform the resource collection into an array.
     *
     * @param \Illuminate\Http\Request $request
     * @return array
     */
    public function toArray($request)
    {
        return parent::toArray($request);
    }
}

```

Datei: RoomsHelper.php

```

<?php

namespace App\Helpers;

use App\RoomEnum;

class RoomsHelper {

    private static $rooms = null;

    public static function getRooms() {
        if (!self::$rooms) {
            self::$rooms = collect(self::roomFacade()->filter());
        }

        return self::$rooms;
    }

    public static function getRoomData($room) {
        [$room_id, $room_title] = explode(' ', $room, 2);

        if (!self::$rooms) {
            self::getRooms();
        }

        if (count(self::$rooms) && is_object(self::$rooms->first())) {
            $data = self::$rooms->find(function($r) use ($room_id) {
                $r['id'] = $room_id;
            });

            return $data;
        }
    }
}

```

```

        return [
            'room_data' => [
                'id' => $room_id,
                'title' => preg_replace('/{0}/', "", $room_title)
            ]
        ];
    }

    private static function roomFacade() {
        return RoomEnum::ROOMS;
    }
}

```

Datei: RouteServiceProvider.php

```
<?php
```

```
namespace App\Providers;
```

```
use Illuminate\Foundation\Support\Providers\RouteServiceProvider as ServiceProvider;
use Illuminate\Support\Facades\Route;
```

```
class RouteServiceProvider extends ServiceProvider
```

```
{
    /**
     * This namespace is applied to your controller routes.
     *
     * In addition, it is set as the URL generator's root namespace.
     *
     * @var string
     */
    protected $namespace = 'App\Http\Controllers';

    /**
     * The path to the "home" route for your application.
     *
     * @var string
     */
    public const HOME = "";

    /**
     * Define your route model bindings, pattern filters, etc.
     *
     * @return void
     */
    public function boot()
    {
        //

        parent::boot();
    }
}

```

```

/**
 * Define the routes for the application.
 *
 * @return void
 */
public function map()
{
    $this->mapApiRoutes();

    $this->mapWebRoutes();

    //
}

/**
 * Define the "web" routes for the application.
 *
 * These routes all receive session state, CSRF protection, etc.
 *
 * @return void
 */
protected function mapWebRoutes()
{
    Route::middleware('web')
        ->namespace($this->namespace)
        ->group(base_path('routes/web.php'));
}

/**
 * Define the "api" routes for the application.
 *
 * These routes are typically stateless.
 *
 * @return void
 */
protected function mapApiRoutes()
{
    Route::prefix('api')
        ->middleware('api')
        ->namespace($this->namespace)
        ->group(base_path('routes/api.php'));
}
}

```

Datei: ScanHereItem.vue

```

<template>
  <v-card>
    <div class="overline px-4 pt-2">Zusammen gegen Corona</div>
    <div class="d-flex">
      <div class="body-2 px-4 text--secondary flex-grow-1">
        Scannen Sie ihren ersten Raum-Barcode.
      </div>
    </div>
  </v-card>
</template>

```



```

    </div>

    <div class="mr-4 text-h4 font-weight-light d-flex align-center justify-center" style="min-width:
48px">
      <v-icon size="48" class="mb-3">qr_code_scanner</v-icon>
    </div>
  </div>
</v-card>
</template>

```

```

<script>
import statesMixin from "@/mixins/statesMixin"

```

```

export default {
  mixins: [statesMixin],

```

```

  created() {
    this.setStateCount(2);
  }
};

```

```

</script>

```

```

<style>
</style>

```

Datei: Scanner.vue

```

<template>
  <video ref="video"></video>

```

```

</template>

```

```

<script>
import QrScanner from 'qr-scanner';

```

```

import QrScannerWorkerPath from '!!file-loader!qr-scanner/qr-scanner-worker.min.js';
QrScanner.WORKER_PATH = QrScannerWorkerPath;

```

```

// QrScanner.WORKER_PATH = 'path/to/qr-scanner-worker.min.js';

```

```

export default {
  data: () => ({
    scanner: null
  }),

```

```

  mounted() {
    this.scanner = new QrScanner(this.$refs.video, result => console.log('decoded qr code:', result));
    this.scanner.start();
  },

```

```

  beforeDestroy() {
    this.scanner.destroy();
    this.scanner = null;
  }
};

```

```

    },
    // beforeRouteLeave() {
    //   this.scanner.stop();
    // }
  }
</script>

```

```
<style>
```

```
</style>
```

Datei: SearchController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use App\User;
use Carbon\Carbon;
use Error;
use Illuminate\Support\Str;
use Illuminate\Http\Request;
use Illuminate\Support\Collection;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Validator;

```

```

class SearchController extends Controller
{

```

```
    public function show(Request $request) {
```

```

        $validator = Validator::make($request->all(), [
            'date' => ['required', 'date_format:Y-m-d', 'before_or_equal:' . Carbon::now('Europe/Berlin')->format('Y-m-d')],
            'consent' => ['sometimes', 'boolean']
        ]);

```

```

        // return array of errors to client with status code 400
        if ($validator->fails()) {
            return response($validator->messages()->toArray(), 400);
        }

```

```
        $users = $this->getUsers($request);
```

```

        if ($users->count() > 1) {
            $response = $users->map(function($user) {
                return $this->parseUser($user);
            });

```

```
        } else {
```

```

            $response = $users->map(function($user) use ($request) {
                $parsed = $this->parseUser($user);
                $parsed['encounters'] = $this->getEncounters($user, $request->date, $request->input('consent', false));
            });

```

```

        return $parsed;
    });
}

return response()->json(['users' => $response]);
}

protected function getEncounters($user, $date, $consent = false) {
    $date = Carbon::parse($date);

    $encounter_ids = $user->getEncountersAfterDate($date)->unique('user_id')->pluck('user_id');
    $encounters = User::findMany($encounter_ids)->whereNotNull('first_name');

    if ($consent) {
        $encounters = $encounters->where('direct_contact_consent', $consent);
    }

    return collect($encounters->map(function($e) {
        return $this->parseUser($e);
    }));
}

```

```

protected function getUsers(Request $request) {

    $users = null;

    if ($ldap_id = $request->input('ldap_id')) {
        $users = User::whereLdapId($ldap_id)->get();
    }

    elseif($id = $request->input('user_id')) {
        $users = User::whereId($id)->get();
    }

    elseif ($name = $request->input('name')) {
        $users = $this->getUserByName($name);

        if ($users->count() && ($email = $request->input('email'))) {
            $users = $this->filterResultsByColumn($users, 'email', $email);
        }

        if ($users->count() && ($telephone = $request->input('telephone'))) {
            $users = $this->filterResultsByColumn($users, 'telephone', $telephone);
        }
    }

    return $users;
}

```

```

private function filterResultsByColumn(Collection $results, String $column, String $value,
$levenshtein_value = 4) {

```

```

    $users = $results->toQuery()->where($column, $value)->get();

```

```

    if ($users->count()) return $users;

    $users = $results->toQuery()->where($column, 'like', "%$value%")->get();

    if ($users->count()) return $users;

    $users = $results->toQuery()->whereRaw("levenshtein($column, '$value') < $levenshtein_value")->get();

    return $users;

    return collect();
}

protected function parseUser($user) {
    if ($user->ldap_id) {
        return collect([
            'id' => $user->id,
            'ldap_id' => $user->ldap_id,
            'name' => $user->last_name . " , " . $user->first_name,
            'first_name' => $user->first_name,
            'last_name' => $user->last_name,
            'email' => $user->email,
            'telephone' => null
        ]);
    }

    return collect([
        'id' => $user->id,
        'name' => $user->last_name . " , " . $user->first_name,
        'first_name' => $user->first_name,
        'last_name' => $user->last_name,
        'email' => $user->email,
        'telephone' => $user->telephone
    ]);
}

private function getUserByName($name) {
    return User::whereRaw("CONCAT(first_name, ' ', last_name) % ?", [$name])
        ->orderByRaw("SIMILARITY(CONCAT(first_name, ' ', last_name), ?) desc", [$name])
        ->get();
}
}

```

Datei: Settings.vue

```

<template>
  <Page
    eager
    title="Einstellungen"
    extension-height="0"

```

```

v-bind="{ ...$props, ...$attrs }"
>
<v-list subheader>
  <v-subheader>Einwilligung</v-subheader>

  <v-list-item>
    <v-list-item-content>
      <v-list-item-title>Direkt kontaktieren</v-list-item-title>
    </v-list-item-content>
  <v-list-item-action>
    <v-switch
      color="info"
      :input-value="consent"
      :loading="direct_contact_loading"
      @click.stop.prevent="changeDirectContactConsent"
    ></v-switch>
  </v-list-item-action>
</v-list-item>

<DarkDialog v-model="dialog" scrollable>
  <template v-slot:activator="{ on, attrs }">
    <div class="body-2 px-4 text--secondary pb-3">
      Ich stimme zu, dass ich im Falle einer Covid-19-Infektion einer Kontaktperson direkt
durch die {{ place }} informiert werden darf.
      <v-btn x-small text color="info" v-bind="attrs" v-on="on">
        Mehr erfahren
      </v-btn>
    </div>
  </template>

  <v-card>
    <v-card-title>Freiwillige Einwilligung</v-card-title>
    <v-card-text style="height: 100%;">
      <p>
        Die hochschulinterne Kontaktnachverfolgung ermöglicht es Nutzern, freiwillig
zustimmen,
        dass sie telefonisch oder per E-Mail durch die {{ place }} informiert werden, wenn sie an
der
        {{ place }} in den letzten 14 Tagen Kontakt mit einer Covid-19-infizierten Person hatten.
      </p>
      <p>
        Grundlage dafür ist die freiwillige Zustimmung der/des Infizierten über ihre/seine
Infektion mit
        Covid-19. Dieses hochschulinterne Verfahren ist von der Kontaktnachverfolgung durch
die zuständigen
        Gesundheitsämter getrennt zu betrachten und bedarf daher einer zusätzlichen freiwilligen
Einwilligung.
      </p>
    </v-card-text>
    <v-card-actions>
      <v-spacer></v-spacer>
      <v-btn
        text
        color="info"

```

```

        @click="dialog = false"
      >
        Verstanden
      </v-btn>
    </v-card-actions>
  </v-card>
</DarkDialog>

<v-divider></v-divider>
<v-subheader>Anzeige</v-subheader>

<DarkDialog v-model="dialog_theme">
  <template v-slot:activator="{ on, attrs }">
    <v-list-item v-bind="attrs" v-on="on">
      <v-list-item-content>
        <v-list-item-title>Thema</v-list-item-title>
        <v-list-item-subtitle>{{ options.find(o => o.value === theme).text }}</v-list-item-
subtitle>
      </v-list-item-content>
    </v-list-item>
  </template>

  <v-list>
    <v-list-item-group color="info" :value="theme" mandatory @change="setTheme">
      <v-list-item
        v-for="option in options"
        :key="option.value"
        :value="option.value"
      >
        <v-list-item-icon>
          <v-icon :color="theme === option.value ? 'secondary lighten-2'">
            {{ option.icon }}
          </v-icon>
        </v-list-item-icon>

        <v-list-item-content>
          <v-list-item-title v-text="option.text"></v-list-item-title>
        </v-list-item-content>
      </v-list-item>
    </v-list-item-group>
  </v-list>
</DarkDialog>

<!--
      <v-list-item>
        <v-list-item-content>
          <v-list-item-title>Gerät einfügen</v-list-item-title>
          <v-list-item-subtitle>

        </v-list-item-subtitle>
      </v-list-item-content>
    </v-list-item-action>
    <v-switch
      v-model="consent"

```

```

        :loading="direct_contact_loading"

        @change="setDirectContactConsent"
    ></v-switch>

</v-list-item-action>
    </v-list-item>
</v-div></v-div -->
</v-list>
<v-list subheader>
</v-div></v-div>
    <v-subheader>Weiteres</v-subheader>

<v-list-item
  target="_blank"
  rel="noopener noreferrer"
  :href="about">
  <v-list-item-icon><v-icon color="secondary lighten-2">new_releases</v-icon></v-list-item-
icon>
  <v-list-item-content>
  <v-list-item-title>Über diese App</v-list-item-title>
  </v-list-item-content>
  <v-list-item-action class="ml-8">
    <v-icon small color="secondary lighten-5">open_in_new</v-icon>
  </v-list-item-action>
</v-list-item>

<v-list-item
  target="_blank"
  rel="noopener noreferrer"
  :href="report">
  <v-list-item-icon><v-icon color="secondary lighten-2">feedback</v-icon></v-list-item-icon>
  <v-list-item-title>Ansteckung melden</v-list-item-title>
  <v-list-item-action class="ml-8">
    <v-icon small color="secondary lighten-5">mail_outline</v-icon>
  </v-list-item-action>
</v-list-item>

<v-list-item
  class="mb-2"
  target="_blank"
  rel="noopener noreferrer"
  :href="support_mail">
  <v-list-item-icon><v-icon color="secondary lighten-2">question_answer</v-icon></v-list-
item-icon>
  <v-list-item-title>Support</v-list-item-title>
  <v-list-item-action class="ml-8">
    <v-icon small color="secondary lighten-5">mail_outline</v-icon>
  </v-list-item-action>
</v-list-item>

</InstallButton />
  </v-list>
</Page>

```

```

</template>

<script>
import { setDirectContact } from "@requests/api";

import InstallButton from "@components/toolbar/InstallButton"

import themeMixin from "@mixins/themeMixin"

export default {
  mixins: [themeMixin],

  data: () => ({
    dialog: false,
    dialog_theme: false,
    consent: false,
    direct_contact_loading: false
  }),

  mounted() {
    console.log("Settings loaded");
    this.consent = this.direct_contact_consent;

    console.log("THEME", this.$vuetify);
  },
  methods: {
    changeDirectContactConsent() {
      this.direct_contact_loading = true;

      this.consent = !this.consent;
      const state = this.direct_contact_consent;

      setDirectContact(this.consent)
        .catch(() => {
          this.consent = !state;
          this.$nextTick(() => this.consent = state);
        })
        .finally(() => this.direct_contact_loading = false);
    },

    // changeDirectContactConsent() {
    //   this.direct_contact_loading = true;

    //   this.consent = !this.consent;
    //   const state = this.direct_contact_consent;

    //   setDirectContact(this.consent)
    //     .catch(() => {
    //       this.consent = !state;
    //       this.$nextTick(() => this.consent = state);
    //     })
    //     .finally(() => this.direct_contact_loading = false);
    // }
  },
}

```



```

    computed: {
      support_mail() {
        return
`mailto:${process.env.VUE_APP_SUPPORT_MAIL}?subject=Kontaktnachverfolgung&body=Sehr%2
0geehrtes%20Support-Team%2C%0D%0A%0D%0A%0D%0A`;
      },
      report() {
        return
`mailto:${process.env.VUE_APP_REPORT_MAIL}?subject=Kontaktnachverfolgung%20Meldung`;
      },
      about() {
        return process.env.VUE_APP_ABOUT_APP;
      },
      place() {
        return process.env.VUE_APP_PLACE;
      }
    },

    watch: {
      direct_contact_consent: {
        handler() {
          if (this.isLoggedIn && this.direct_contact_consent === null) {
            this.direct_contact_loading = true;
          } else {
            this.consent = this.direct_contact_consent;
            this.direct_contact_loading = false;
          }
        },
        immediate: true
      }
    },

    components: {
      InstallButton
    },
  };
</script>

<style>
</style>

```

Datei: ShortSession.php

```

<?php

namespace App\Http\Middleware;

use Closure;

class ShortSession
{

```

```

/**
 * Handle an incoming request.
 *
 * @param \Illuminate\Http\Request $request
 * @param \Closure $next
 * @return mixed
 */
public function handle($request, Closure $next)
{
    // 10 minutes
    config(['session.lifetime' => 10]);
    return $next($request);
}
}

```

Date: StartHere.vue

```

<template>
  <div style="position: fixed; bottom: 60px; max-width: 260px; width: 100%;"
    :style="{ 'right': isMobile ? '40px' : '190px' }">
    <!-- <div class="text--secondary body-2 mb-8">
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed scelerisque finibus
      aliquet. Nunc sit amet aliquet erat.
      Maecenas eu auctor libero. Quisque molestie dolor ornare aliquet consequat.
    </div> -->
    <div class="d-flex justify-end mr-16">
      <ArrowToFabSVG />
    </div>
  </div>
</template>

<script>
import ArrowToFabSVG from "@/assets/arrow_to_fab.svg";

export default {
  components: {
    ArrowToFabSVG
  }
};
</script>

<style lang="scss" scoped>
svg {
  height: 100px;
  fill: var(--v-secondary-lighten3);
}

@media (prefers-color-scheme: dark) {
  img#arrow-svg {
    filter: invert(1);
  }
}

```

```
</style>
```

Datei: Suggested.vue

```
<template>
  <SuggestedBottomSheet />
</template>
```

```
<script>
import SuggestedBottomSheet from "@/components/home/SuggestedBottomSheet"
export default {
  components: {
    SuggestedBottomSheet
  }
}
</script>
```

```
<style>
```

```
</style>
```

Datei: SuggestedBottomSheet.vue

```
<template>
  <v-bottom-sheet
    inset
    v-model="sheet"
    :overlay-color="$vuetify.theme.dark ? 'black' : undefined"
    :overlay-opacity="$vuetify.theme.dark ? 0.7 : undefined"
  >
    <v-list two-line shaped :disabled="loading">
      <v-subheader>Raum betreten</v-subheader>
      <v-list-item-group
        color="info"
        v-model="selected"
      >
        <v-list-item
          v-for="room in suggested"
          :key="room"
          :value="room"
          :disabled="selected && selected !== room"
        >
          <v-list-item-content>
            <v-list-item-title>{{ room.split(' ')[0] }}</v-list-item-title>
            <v-list-item-subtitle>{{ room.split(' ').slice(1).join(' ').replace(/[/]/g, " ) }}</v-list-item-
subtitle>
          </v-list-item-content>
          <v-list-item-action>
            <v-btn
              v-if="selected === room"
              icon
              small
            >
```

```

        :loading="loading"
        :color="success ? 'success' : ""
      >
        <v-icon v-if="success">check_circle</v-icon>
      </v-btn>
    </v-list-item-action>
  </v-list-item>
</v-list-item-group>
</v-list>

</v-bottom-sheet>
</template>

<script>

import { setVisit } from "@requests/api";

export default {
  data: () => ({
    sheet: true,
    loading: false,
    success: false,
    selected: null
  }),

  methods: {
    close() {
      if (this.is_opened_directly) {
        this.$router.push({ name: "home" }).catch(() => {});
      } else {
        console.log("GOing back");
        this.$router.go(-1);
      }
    },

    async visit(room) {
      this.loading = true;

      await new Promise((res) => setTimeout(() => res(), 1000));

      setVisit(room)
        .then(() => this.success = true)
        .catch(() => console.log("COULDNT ENTER"))
        .finally(() => {
          this.loading = false;
          setTimeout(() => {
            this.success = false;
            this.selected = null;
            this.close();
          }, 1000);
        });
    },
  },
},

```

```

watch: {
  sheet(state) {
    if (!state) {
      setTimeout(this.close, 350);
    }
  },
  selected(room) {
    if (room) {
      this.visit(room);
    }
  }
}
};
</script>

```

Datei: SuggestedItem.vue

```

<template>
  <v-card>
    <div class="overline px-4 pt-2">Schnellanmeldung</div>
    <v-slide-group
      show-arrows
      v-model="model"
      class="px-1 pb-3 pt-1"
      :key="uniqueSuggested"
    >
      <v-slide-item
        v-for="room in suggested"
        :key="room"
        v-slot:default="{ active, toggle }"
      >
        <v-btn
          x-large
          rounded
          depressed
          :input-value="active"
          active-class="accent"
          class="mx-2 subtitle-2 font-weight-medium"
          @click="toggle"
        >
          <div>
            {{ room.split(' ')[0] }}
            <div class="text-caption text-capitalize" style="">
              {{ room.split(' ').slice(1).join(' ').replace(/[/\()]/g, " ") }}
            </div>
          </div>
        </v-btn>
      </v-slide-item>
    </v-slide-group>

    <v-expand-transition>
    <v-card-actions>
      <v-spacer />

```

```

      <!-- @click="exitRoom(item.id)" -->
    <v-btn
      small
      :text="!success"
      :loading="loading"
      :depressed="success"
      :disabled="isNaN(model)"
      :color="success ? 'success' : 'info'"
      @click="visit(suggested[model])"
    >
      <v-icon v-if="success" left>check</v-icon>
      {{ success ? 'Erfolgreich' : 'Raum betreten'}}
    </v-btn>

    <!-- <v-btn
      small
      text

      color="primary"
    >
      Raum betreten
    </v-btn> -->
  </v-card-actions>
  </v-expand-transition>
</v-card>
</template>

<script>
import { setVisit } from "@requests/api";

export default {
  data: () => ({
    model: undefined,
    loading: false,
    success: false,
  }),
  methods: {
    visit(room) {
      this.loading = true;

      setVisit(room)
        .then(() => this.success = true)
        .catch(() => console.log("COULDNT ENTER"))
        .finally(() => {
          this.loading = false;
          setTimeout(() => {
            this.model = 0;
            this.success = false;
            setTimeout(() => this.model = undefined, 0);
          }, 1000);
        });
    },
  },

```

```

    },
    computed: {
      uniqueSuggested() {
        return this.suggested.map(s => s.split(' ')[0]).join('');
      }
    }
  };
</script>

```

```

<style lang="scss" scoped>
::v-deep {
  .v-slide-group__prev {
    position: absolute;
    top: 0;
    bottom: 0;
    z-index: 1;
    background: rgb(255 255 255);
    transition: opacity 0.2s ease-in;

    &--disabled {
      opacity: 0;
    }
  }

  .v-slide-group__next {
    transition: opacity 0.2s ease-in;

    &--disabled {
      opacity: 0;
    }
  }

  .v-slide-group__content {
    padding-left: 24px !important;
    padding-right: 4px !important;
  }
}
</style>

```

Datei: SuggestedItemButton.vue

```

<template>
  <v-card>
    <div class="overline px-4 pt-2" @click="cycleStates">Schnellanmeldung</div>

    <v-slide-group
      v-if="state === 0"
      show-arrows
      v-model="model"
      class="px-1 pb-3 pt-1"
    >
      <v-slide-item

```

```

      v-for="n in suggested"
      :key="n"
      v-slot:default="{ active, toggle }"
    >
      <v-btn
        x-large
        rounded
        depressed
        :input-value="active"
        active-class="accent"
        class="mx-2 subtitle-2 font-weight-medium"
        @click="toggle"
      >
        <div>
          {{ n.split(' ')[0] }}
          <div class="text-caption text-lowercase" style="">
            {{ n.split(' ').slice(1).join(' ') }}
          </div>
        </div>
      </v-btn>
    </v-slide-item>
  </v-slide-group>

```

```

      <v-expand-transition>
    <v-card-actions>
      <v-spacer />
      <v-btn
        small
        text
        :disabled="isNaN(model)"
        color="primary"
      >
        Raum betreten
      </v-btn>
    </v-card-actions>
  </v-expand-transition>
</v-card>
</template>

```

```

<script>
import statesMixin from "@/mixins/statesMixin"

```

```

export default {
  mixins: [statesMixin],
  data: () => ({
    model: undefined,
  }),
  created() {
    this.setStatesCount(2);
  }
};
</script>

```

```

<style lang="scss" scoped>

```



```

::v-deep {
  .v-slide-group__prev {
    position: absolute;
    top: 0;
    bottom: 0;
    z-index: 1;
    background: rgb(255 255 255);
    transition: opacity 0.2s ease-in;

    &--disabled {
      opacity: 0;
    }
  }

  .v-slide-group__next {
    transition: opacity 0.2s ease-in;

    &--disabled {
      opacity: 0;
    }
  }

  .v-slide-group__content {
    padding-left: 24px;
    padding-right: 4px;
  }
}
</style>

```

Datei: TimelineVertical.vue

```

<template>
  <v-timeline align-top dense class="text--secondary pt-2">
    <v-timeline-item small fill-dot color="background">
      <template #icon>
        <v-icon color="secondary" small>people_outline</v-icon>
      </template>
      <span class="text-caption">{{ item.total_visits.before }} Personen vor Ihnen
an diesem Tag</span>
    </v-timeline-item>

    <template>
      <v-timeline-item
        small
        fill-dot
        class="py-3"
        icon="arrow_forward"
        color="background darken-4"
      >
        <div class="body-2">Besucht um {{ formatTime(item.visited_on)
}}</div>
        <div class="text-caption">{{ item.total_visits.during }}
Begegnungen</div>

```

```

        </v-timeline-item>
      <v-timeline-item
        small
          fill-dot
          icon="arrow_back"
          color="background darken-4"
          class="pt-6 visit visit--done">
        <div class="body-2">Bis {{ formatTime(item.exited_on) }}</div>
      </v-timeline-item>
    </template>
  </v-timeline>
</template>

```

```

<script>
import { format, parse } from "date-fns";

import statesMixin from "@mixins/statesMixin";

import { exitVisit } from "@requests/api";

export default {
  data: () => ({
    loading: false,
    success: false
  }),
  props: {
    item: Object,
  },
  mixins: [statesMixin],

  created() {
    this.setStateCount(2);
    this.setState(1);
  },
  methods: {
    formatTime(datetime) {
      return format(
        parse(datetime, "yyyy-MM-dd HH:mm:ss", new Date()),
        "HH:mm"
      );
    },
    exitRoom(id) {
      this.loading = true;

      exitVisit(id)
        .then((r) => {
          this.success = true;

          setTimeout(() => {
            this.$emit('updated', r.visits)
          }, 1000);
        });
    }
  }
};

```

```

        })
        .finally(() => this.loading = false);
    }
},
};
</script>

```

```

<style lang="scss" scoped>

```

```

::v-deep {
  &.v-timeline {
    &::before {
      height: calc(100% - 10px) !important;
    }
  }

  & .v-timeline-item {
    padding-bottom: 8px !important;

    & .v-timeline-item__inner-dot {
      position: relative;
    }
  }

  & .visit {
    &::before {
      content: "";
      position: absolute;
      left: calc(48px - 1px);
      bottom: 10px;
      width: 2px;
      height: 80px;
    }

    &--done::before {
      background-color: var(--v-background-darken3) !important;
    }
  }
}

```

```

.pulse {
  animation: pulse 2s infinite;
  height: 100%;
  width: 100%;
  border-radius: inherit;
  position: absolute;
}

```

```

@keyframes pulse {
  0% {
    box-shadow: 0 0 0 0 currentColor;
    opacity: 0.4;
  }
  70% {
    box-shadow: 0 0 0 10px currentColor;
  }
}

```

```

        opacity: 0;
    }
    100% {
        box-shadow: 0 0 0 0 currentColor;
        opacity: 0;
    }
}
</style>

```

Datei: Toolbar.vue

```

<template>
  <v-sheet
    class="px-4 pt-8"
    color="primary"
    height="160px"
    width="100%"
    style="position: sticky; top: -84px;"
    id="toolbar"
    v-scroll="onScroll"
  >
    <v-img
      height="60"
      contain
      :src="logo"
      :style="{ 'opacity': opacity, 'transform': `scale(${scale})` }"
      style="position: fixed; top: 24px;"
    />
    <v-app-bar color="transparent" flat dark fixed :style="{ 'opacity': opacity }">
      <v-badge
        :value="installWasDefered"
        color="accent"
        bottom
        dot
        overlap
        offset-x="16"
        offset-y="16"
        transition="scale-transition"
      >
        <v-btn
          icon
          class="ml-n3"
          to="settings"
          aria-label="Einstellungen">
          <v-icon>settings</v-icon>
        </v-btn>
      </v-badge>

      <v-spacer></v-spacer>

      <v-btn icon to="account" v-if="!isLoggedIn" aria-label="Anmelden">
        <v-icon>account_circle</v-icon>

```

```

        </v-btn>

<v-menu
  v-else
  v-model="menu"
  :nudge-width="300"
  :close-on-content-click="false"
  transition="slide-y-transition"
  >
  <template v-slot:activator="{ on, attrs }">
    <!-- <v-btn
      icon
      text
      color="white"
      class="pa-1"
      v-bind="attrs"
      v-on="on">
      <v-avatar
        color="white"
        size="36"
        style="border: 2px var(--v-primary-base) solid"
        v-ripple="{ class: 'primary--text' }">
        <span class="primary--text">{{ initials }}</span>
      </v-avatar>
    </v-btn -->

    <v-avatar color="white" size="36"
      v-ripple="{ class: 'primary--text' }"
      v-bind="attrs"
      v-on="on">
      <span class="primary--text">{{ initials }}</span>
    </v-avatar>
  </template>

<v-card>
  <v-list>
    <v-list-item>
      <v-list-item-content>
        <v-list-item-title>Angemeldet als</v-list-item-title>
        <v-list-item-subtitle>{{ name }}</v-list-item-subtitle>
      </v-list-item-content>

      <v-list-item-action>
        <v-btn icon @click="menu = false">
          <v-icon>close</v-icon>
        </v-btn>
      </v-list-item-action>
    </v-list-item>
  </v-list>
</v-card>
</v-menu>

  </v-app-bar>
</v-sheet>

```

```

</template>

<script>
import installMixin from "@/mixins/installMixin"

// import InstallButton from "@/components/toolbar/InstallButton"

import { logout } from "@/requests/api"

export default {
  data: () => ({
    opacity: 1,
    scale: 1,
    menu: false,
    deleting: false,
  }),

  mixins: [installMixin],

  methods: {
    onScroll() {
      this.opacity = 1 - window.scrollY / 80;
      this.scale = Math.min(1 - window.scrollY / 160, 1);
    },

    deleteUser() {
      this.deleting = true;
      logout().finally(() => this.deleting = false);
    }
  },

  computed: {
    name() {
      return `${this.user.first_name} ${this.user.last_name}`
    },
    initials() {
      let names = [this.user.first_name, this.user.last_name];
      return names.map(n => n.charAt(0).toUpperCase()).join("");
    },
    logo() {
      return `${process.env.BASE_URL}img/common/logo.png`;
    }
  },

  watch: {
    isLoggedIn() {
      if (!this.isLoggedIn) this.menu = false;
    }
  },

  components: {
    // InstallButton
  }
};

```

```
</script>
```

```
<style>  
</style>
```

Datei: TotalEncounters.vue

```
<template>  
  <v-card>  
    <div class="overline px-4 pt-2">Aktivität in den letzten 14 Tage</div>  
    <!-- Raumkontakt hinzufügen -> pfeil und pulsieren -->  
    <div class="py-2">  
      <v-list-item>  
        <v-list-item-icon class="pl-2">  
          <v-icon color="info">list_alt</v-icon>  
        </v-list-item-icon>  
        <v-list-item-content>  
          <v-list-item-title>Einträge</v-list-item-title>  
        </v-list-item-content>  
  
        <v-list-item-action class="pr-3">  
          <span class="text-h5 font-weight-light">  
            {{ total_encounters.visits_count }}  
          </span>  
        </v-list-item-action>  
      </v-list-item>  
      <v-list-item>  
        <v-list-item-icon class="pl-2">  
          <v-icon color="info">meeting_room</v-icon>  
        </v-list-item-icon>  
        <v-list-item-content>  
          <v-list-item-title>Besuchte Räume</v-list-item-title>  
        </v-list-item-content>  
  
        <v-list-item-action class="pr-3">  
          <span class="text-h5 font-weight-light">  
            {{ total_encounters.room_count }}  
          </span>  
        </v-list-item-action>  
      </v-list-item>  
      <v-list-item>  
        <v-list-item-icon class="pl-2">  
          <v-icon color="info">people</v-icon>  
        </v-list-item-icon>  
        <v-list-item-content>  
          <v-list-item-title>Kontaktpersonen</v-list-item-title>  
        </v-list-item-content>  
  
        <v-list-item-action class="pr-3">  
          <span class="text-h5 font-weight-light">  
            {{ total_encounters.people_count }}  
          </span>  
        </v-list-item-action>  
      </v-list-item>
```

```

        </v-list-item>
    </div>
    <!-- <div class="d-flex">
        <div>
            <span class="text-h4">13</span>
            <span class="text-caption">Räume</span>
        </div>
        <div>
            <span class="text-h4">132</span>
            <span class="text-caption">Personen</span>
        </div>
    </div> -->
    </v-card>
</template>

<script>
export default {

}
</script>

<style>

</style>

```

Datei: TrimStrings.php

```

<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\TrimStrings as Middleware;

class TrimStrings extends Middleware
{
    /**
     * The names of the attributes that should not be trimmed.
     *
     * @var array
     */
    protected $except = [
        'password',
        'password_confirmation',
    ];
}

```

Datei: TrustHosts.php

```

<?php

namespace App\Http\Middleware;

```



```
use Illuminate\Http\Middleware\TrustHosts as Middleware;
```

```
class TrustHosts extends Middleware
{
    /**
     * Get the host patterns that should be trusted.
     *
     * @return array
     */
    public function hosts()
    {
        return [
            $this->allSubdomainsOfApplicationUrl(),
        ];
    }
}
```

Datei: TrustProxies.php

```
<?php
```

```
namespace App\Http\Middleware;
```

```
use Fideloper\Proxy\TrustProxies as Middleware;
use Illuminate\Http\Request;
```

```
class TrustProxies extends Middleware
{
    /**
     * The trusted proxies for this application.
     *
     * @var array|string|null
     */
    protected $proxies;

    /**
     * The headers that should be used to detect proxies.
     *
     * @var int
     */
    protected $headers = Request::HEADER_X_FORWARDED_FOR |
Request::HEADER_X_FORWARDED_HOST | Request::HEADER_X_FORWARDED_PORT |
Request::HEADER_X_FORWARDED_PROTO | Request::HEADER_X_FORWARDED_AWS_ELB;
}
```

Datei: User.php

```
<?php
```

```

namespace App;

use App\Traits\UsesUuid;
use Carbon\Carbon;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

use Illuminate\Support\Facades\DB;

class User extends Authenticatable
{
    use Notifiable, UsesUuid;

    public function getRouteKeyName()
    {
        return 'unique_id';
    }

    public function visits() {
        return $this->hasMany(Visit::class)->orderBy('visited_on', 'desc')->orderBy('exited_on', 'desc');
    }

    public function getVisitsAfterDate(Carbon $date = null) {
        if ($this->visits()->count() === 0) return collect([]);

        if ($date === null) {
            $date = $this->visits->last()->visited_on->startOfDay();
        }

        $visits = $this->visits()->whereDate('visited_on', '>=', $date)->get();

        return $visits;
    }

    public function getEncountersAfterDate(Carbon $date = null) {
        if ($this->visits()->count() === 0) return collect([]);

        if ($date === null) {
            $date = $this->visits->last()->visited_on->startOfDay();
        }

        $visits = $this->visits()->whereDate('visited_on', '>=', $date)->get();

        return $visits->pluck('encounters')->flatten();
    }

    public function mostVisited() {
        if ($this->visits()->count() === 0) return collect([]);

        return DB::table('visits')
            ->select('room')
            ->where('user_id', $this->id)
            ->orderByRaw('SUM(coalesce (exited_on, now())) - visited_on desc')
            ->groupBy('room')->pluck('room');
    }
}

```

```
}  
}
```

Datei: UserResource.php

```
<?php
```

```
namespace App\Http\Resources;
```

```
use App\Helpers\RoomsHelper;
```

```
use Carbon\Carbon;
```

```
use Illuminate\Http\Resources\Json\JsonResource;
```

```
class UserResource extends JsonResource
```

```
{
```

```
    /**
```

```
     * Transform the resource into an array.
```

```
     *
```

```
     * @param \Illuminate\Http\Request $request
```

```
     * @return array
```

```
     */
```

```
    public function toArray($request)
```

```
    {
```

```
        $date = Carbon::now('Europe/Berlin')->modify('-14 days')->startOfDay();
```

```
        $visits = $this->getVisitsAfterDate($date);
```

```
        $encounters = $this->getEncountersAfterDate($date);
```

```
        $suggested = $this->mostVisited()->take(6);
```

```
        return [
```

```
            'token' => $this->unique_id,
```

```
            'user' => $this->only(['first_name', 'last_name']),
```

```
            'direct_contact_consent' => $this->direct_contact_consent,
```

```
            'total_encounters' => [
```

```
                'visits_count' => $visits->count(),
```

```
                'people_count' => $encounters->unique('user_id')->count(),
```

```
                'room_count' => $visits->unique('room')->count()
```

```
            ],
```

```
            'suggested' => $suggested,
```

```
            'most_visited' => $suggested->map(function($r) { return RoomsHelper::getRoomData($r); }),
```

```
            'visits' => VisitResource::collection($this->visits)
```

```
        ];
```

```
    }
```

```
}
```

Datei: UserSyncController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use App\Http\Resources\UserResource;

use App\User;
use App\Helpers\LoginHelper;
use App\Http\Requests\LoginRequest;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Validation\ValidationException;

class UserSyncController extends Controller
{
    public function store(LoginRequest $request) {
        $user = $request->user();

        try {
            $auth_user = LoginHelper::attemptLogin($request->username, $request->password);
        } catch (\Throwable $th) {
            return $this->sendFailedLoginResponse($request);
        }

        if (!$auth_user) {
            return $this->sendFailedLoginResponse($request);
        }

        $existing_user = User::whereLdapId($auth_user->ldap_id)->first();

        if ($existing_user) {
            $user->visits()->update([
                'user_id' => $existing_user->id
            ]);

            $user->delete();
            $request->session()->invalidate();

            $existing_user->first_name = $auth_user->first_name;
            $existing_user->last_name = $auth_user->last_name;
            $existing_user->unique_id = null;

            $existing_user->save();

            Auth::guard('web')->login($existing_user, true);

            return new UserResource($existing_user);
        } else {
            /**
             * @todo fill data?
             */
            $user->ldap_id = $auth_user->ldap_id;
            $user->first_name = $auth_user->first_name;
            $user->last_name = $auth_user->last_name;
            $user->unique_id = null;

            $user->save();
        }
    }
}

```

```

    return new UserResource($user);
}

/**
 * Get the failed login response instance.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Symfony\Component\HttpFoundation\Response
 *
 * @throws \Illuminate\Validation\ValidationException
 */
protected function sendFailedLoginResponse(Request $request)
{
    throw ValidationException::withMessages([
        'username' => [trans('auth.failed')],
    ]);
}
}

```

Datei: UsesUuid.php

```

<?php

namespace App\Traits;

use Illuminate\Support\Str;

trait UsesUuid
{
    protected static function bootUsesUuid()
    {
        static::creating(function ($model) {
            if (!$model->unique_id) {
                $model->unique_id = (string) Str::uuid();
            }
        });
    }
}
}

```

Datei: VerifyCsrfToken.php

```

<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;

class VerifyCsrfToken extends Middleware
{
    /**

```

```

    * The URIs that should be excluded from CSRF verification.
    *
    * @var array
    */
    protected $except = [
        //
    ];
}

```

Datei: Visit.php

```
<?php
```

```

return [
    'date_conflict' => 'There is already a visit overlapping the given times.',
    'active' => 'You entered a room after the given time. Please input the time when you left the room.'
];

```

Datei: VisitController.php

```
<?php
```

```

namespace App\Http\Controllers;

use App\Helpers\RoomsHelper;
use App\Http\Requests\VisitRequest;
use App\Http\Resources\UserResource;
use App\Http\Resources\VisitResource;
use App\RoomEnum;
use App\User;
use App\Visit;
use Carbon\Carbon;
use Illuminate\Support\Str;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\ValidationException;

class VisitController extends Controller
{
    public function show(Request $request) {
        $user = $request->user();

        return $this->response($user);
    }

    public function destroy(Request $request, Visit $visit) {

        $user = $request->user();

```

```

    if (!$user->visits->contains($visit)) {
        return;
    }

    $visit->exited_on = Carbon::now('Europe/Berlin');
    $visit->save();

    return $this->response($user);
}

public function store(VisitRequest $request) {
    $user = $request->user();
    $validator = Validator::make([], []);

    if ($user->unique_id && $user->visits()->count() > 0) {
        return response()->json(['message' => trans('auth.notactivated')], 500);
    }

    $rooms = RoomsHelper::getRooms();

    $room = $rooms->first(function($room) use ($request) {
        $room_without_name = explode(' ', $request->room)[0];

        return $room === $request->room
            || explode(' ', $room)[0] === $room_without_name
            || Str::contains($room, $room_without_name);
    });

    if (!$room) {
        $validator->errors()->add('room', __('validation.not_in', ['attribute' =>
        __('validation.attributes.room')]));
        throw new ValidationException($validator);
    }

    $last_visit = $user->visits->first();

    // Carbon parse uses the current time as default when parsing
    $visited_on = Carbon::parse($request->input('visited_on'))->timezone('Europe/Berlin');
    $exited_on = $request->input('exited_on');

    if ($exited_on) {
        $exited_on = Carbon::parse($exited_on)->timezone('Europe/Berlin');
    }

    if ($request->filled(['visited_on', 'exited_on'])) {

        // Check if there is an overlap
        if (Visit::isOverlapping($user, $visited_on, $exited_on)) {
            $validator->errors()->add('visited_on', __('visit.date_conflict'));
            throw new ValidationException($validator);
        }
    }
    else if ($request->filled('visited_on')) {
        if (!$visited_on->isToday()) {

```

```

        $validator->errors()->add('visited_on', __('validation.required', ['attribute' =>
__('validation.attributes.visited_on')]));
        throw new ValidationException($validator);
    }

    if ($last_visit && $last_visit->visited_on->startOfMinute()->greaterThan($visited_on)) {
        $validator->errors()->add('visited_on', __('visit.active'));
        throw new ValidationException($validator);
    }
}

if (!$request->filled('exited_on') && $last_visit && !$last_visit->exited_on) {
    $last_exited_on = $last_visit->visited_on->isToday()
        ? $visited_on
        : $last_visit->visited_on->endOfDay();

    $last_visit->exited_on = $last_exited_on->startOfMinute();
    $last_visit->save();
}

$user->visits()->create([
    'room' => $room,
    'visited_on' => $visited_on->startOfMinute(),
    'exited_on' => $exited_on ? $exited_on->startOfMinute() : null,
]);

return $this->response($user);
}

private function response(User $user) {
    // Visits could have been loaded in memory.
    $user->refresh();

    return new UserResource($user);
}
}

```

Datei: VisitItem.vue

```

<template>
    <v-list-item-content
        class="pl-6"
        style="position: relative; overflow: visible"
    >
        <v-list-item-title class="mb-1">
            {{ item.room }}
        </v-list-item-title>
        <v-list-item-subtitle class="text-caption">
            <v-icon x-small class="mr-1">schedule</v-icon>
            {{ formatTime(item.visited_on) }}
            <v-icon x-small class="ml-2 mr-1">people_outline</v-icon>
            {{ item.total_visits.count }}
        </v-list-item-subtitle>
    </v-list-item-content>

```



```

        </v-list-item-subtitle>
    </v-list-item-content>
</template>

<script>
import { format, parse } from "date-fns";

export default {
  props: {
    item: Object
  },

  methods: {
    formatTime(datetime) {
      return format(
        parse(datetime, "yyyy-MM-dd HH:mm:ss", new Date()),
        "HH:mm"
      );
    },
  }
};
</script>

<style>
</style>

```

Datei: VisitRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class VisitRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()

```

```

    {
        return [
            'room' => ['required'],
            'visited_on' => ['sometimes', 'required', 'date'],
            'exited_on' => ['nullable', 'required_with_all:visited_on,exited_on', 'after_or_equal:visited_on',
'date']
        ];
    }
}

```

Datei: VisitResource.php

```
<?php
```

```
namespace App\Http\Resources;
```

```
use App\Helpers\RoomsHelper;
```

```
use Illuminate\Http\Resources\Json\JsonResource;
```

```
class VisitResource extends JsonResource
```

```

{
    /**
     * Transform the resource into an array.
     *
     * @param \Illuminate\Http\Request $request
     * @return array
     */
    public function toArray($request)
    {
        return [
            'id' => $this->scrambled_id,
            'room' => $this->room,
            $this->merge(RoomsHelper::getRoomData($this->room)),
            'total_visits' => $this->total_visits,
            'visited_on' => $this->visited_on->format('Y-m-d H:i:s'),
            'exited_on' => $this->exited_on ? $this->exited_on->format('Y-m-d H:i:s') : null
        ];
    }
}

```

Datei: VisitsEmpty.vue

```
<template>
```

```
<v-card>
```

```
<div class="overline px-4 pt-2">Zusammen gegen Corona</div>
```

```
<div class="d-flex">
```

```
<div class="body-2 px-4 text--secondary flex-grow-1 pb-4">
```

```
<span>Helfen Sie uns, Ansteckungsketten nachzuerfolgen.</span>
```

```
</div>
```

```

      <div class="mr-4 text-h4 font-weight-light d-flex align-center justify-center" style="min-width:
48px">
        <v-icon size="48" class="mb-3">coronavirus</v-icon>
      </div>
    </div>
  </v-card>
</template>

```

Datei: VisitsLoading.vue

```

<template>
  <v-card class="mb-4 py-3">
    <v-skeleton-loader width="120px" height="32px" type="list-item" />
    <v-skeleton-loader v-for="i in 3" :key="i" height="64px" type="list-item-two-line" />
  </v-card>
</template>

<script>
export default {};
</script>

<style>
</style>

```

Datei: api.js

```

import { camelCase } from "lodash-es";

import connection from '@requests/connection'

import store from '@store';

// declare a response interceptor
connection.interceptors.response.use((response) => {

  const data = response?.data?.data || response?.data || response;

  const expected_values = ['token', 'visits', 'user', 'suggested', 'total_encounters',
'direct_contact_consent'];

  expected_values.forEach(val => {
    const event = camelCase(`set ${val}`);
    if (val in data) store.dispatch(event, data[val])
  });

  return data;
}, error => {
  // 419 CSRF Token mismatch
  if ([419, 401].includes(error?.response?.status)) {
    store.dispatch('setError', {
      message: 'Token ungültig. Bitte neuladen.',
      action: 'reload'
    });
  }
});

```

```

    });

    return Promise.reject(error);
  }

  const error_bag = error?.response?.data?.errors || [];
  let message = Object.values(error_bag).map(v => v.join(' ')).join('<br />');

  if (error_bag.length === 0) {
    message = error?.response?.data?.message || error?.message;
  }

  store.commit('setError', message);

  return Promise.reject(error);
});

export function showVisits() {
  return connection.get(`/api/visit`);
}

export function setVisit(room, visited_on, exited_on) {
  const payload = { room, visited_on, exited_on };

  return connection.put(`/api/visit`, payload);
}

export function exitVisit(id) {
  return connection.delete(`/api/visit/${id}`);
}

export function init() {
  return connection.post(`/account`, { token: store.state.token });
}

export function login({ username, password }) {
  return connection.post(`/api/account/login`, { username, password });
}

export function logout() {
  return connection.post(`/api/account/logout`);
}

export function register(user_data) {
  return connection.post(`/api/account/register`, user_data);
}

export function setDirectContact(gives_consent) {
  return connection.post(`/api/account/contact`, { gives_consent });
}

export function loadRooms() {
  return connection.get(`/api/rooms`);
}

```

```
export default connection;
```

Datei: api.php

```
<?php
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Route;
```

```
/*
```

```
|-----
```

```
| API Routes
```

```
|-----
```

```
|
```

```
| Here is where you can register API routes for your application. These
```

```
| routes are loaded by the RouteServiceProvider within a group which
```

```
| is assigned the "api" middleware group. Enjoy building your API!
```

```
|
```

```
*/
```

```
// Route::middleware('auth:api')->get('/user', function (Request $request) {
```

```
//     return $request->user();
```

```
// });
```

```
Route::middleware('auth:sanctum')->group(function () {
```

```
    Route::get('rooms', 'RoomController@show');
```

```
    Route::get('visit', 'VisitController@show');
```

```
    Route::put('visit', 'VisitController@store');
```

```
    Route::delete('visit/{visit}', 'VisitController@destroy');
```

```
    Route::post('account/login', 'UserSyncController@store');
```

```
    Route::post('account/register', 'AccountController@register');
```

```
    Route::post('account/contact', 'AccountController@directContactConsent');
```

```
});
```

Datei: app.js

```
/**
```

```
 * First we will load all of this project's JavaScript dependencies which
```

```
 * includes Vue and other libraries. It is a great starting point when
```

```
 * building robust, powerful web applications using Vue and Laravel.
```

```
*/
```

```
require('./bootstrap');
```

```
window.Vue = require('vue');
```

```
/**
```

```
 * The following block of code may be used to automatically register your
```

```

* Vue components. It will recursively scan this directory for the Vue
* components and automatically register them with their "basename".
*
* Eg. ./components/ExampleComponent.vue -> <example-component></example-component>
*/

const files = require.context('./', true, /\.vue$/i);
files.keys().map(key => Vue.component(key.split('/').pop().split('.')[0], files(key).default));

// Vue.component('example-component', require('./components/ExampleComponent.vue').default);

/**
 * Next, we will create a fresh Vue application instance and attach it to
 * the page. Then, you may begin adding components to this application
 * or customize the JavaScript scaffolding to fit your unique needs.
 */

const app = new Vue({
  el: '#app'
});

```

Datei: app.php

```
<?php
```

```
return [
```

```

/*
|-----
| Application Name
|-----
|
| This value is the name of your application. This value is used when the
| framework needs to place the application's name in a notification or
| any other location as required by the application or its packages.
|
*/

'name' => env('APP_NAME', 'Laravel'),

/*
|-----
| Application Environment
|-----
|
| This value determines the "environment" your application is currently
| running in. This may determine how you prefer to configure various
| services the application utilizes. Set this in your ".env" file.
|
*/

'env' => env('APP_ENV', 'production'),

```

```
/*
|-----
| Application Debug Mode
|-----
|
| When your application is in debug mode, detailed error messages with
| stack traces will be shown on every error that occurs within your
| application. If disabled, a simple generic error page is shown.
|
*/

'debug' => (bool) env('APP_DEBUG', false),

/*
|-----
| Application URL
|-----
|
| This URL is used by the console to properly generate URLs when using
| the Artisan command line tool. You should set this to the root of
| your application so that it is used when running Artisan tasks.
|
*/

'url' => env('APP_URL', 'http://localhost'),

'asset_url' => env('ASSET_URL', null),

/*
|-----
| Application Timezone
|-----
|
| Here you may specify the default timezone for your application, which
| will be used by the PHP date and date-time functions. We have gone
| ahead and set this to a sensible default for you out of the box.
|
*/

'timezone' => 'Europe/Berlin',

/*
|-----
| Application Locale Configuration
|-----
|
| The application locale determines the default locale that will be used
| by the translation service provider. You are free to set this value
| to any of the locales which will be supported by the application.
|
*/

'locale' => 'de',
```

```
/*
|-----
| Application Fallback Locale
|-----
|
| The fallback locale determines the locale to use when the current one
| is not available. You may change the value to correspond to any of
| the language folders that are provided through your application.
|
*/

'fallback_locale' => 'de',

/*
|-----
| Faker Locale
|-----
|
| This locale will be used by the Faker PHP library when generating fake
| data for your database seeds. For example, this will be used to get
| localized telephone numbers, street address information and more.
|
*/

'faker_locale' => 'de_DE',

/*
|-----
| Encryption Key
|-----
|
| This key is used by the Illuminate encrypter service and should be set
| to a random, 32 character string, otherwise these encrypted strings
| will not be safe. Please do this before deploying an application!
|
*/

'key' => env('APP_KEY'),

'cipher' => 'AES-256-CBC',

/*
|-----
| Autoloaded Service Providers
|-----
|
| The service providers listed here will be automatically loaded on the
| request to your application. Feel free to add your own services to
| this array to grant expanded functionality to your applications.
|
*/

'providers' => [
```



```

/*
 * Laravel Framework Service Providers...
 */
Illuminate\Auth\AuthServiceProvider::class,
Illuminate\Broadcasting\BroadcastServiceProvider::class,
Illuminate\Bus\BusServiceProvider::class,
Illuminate\Cache\CacheServiceProvider::class,
Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
Illuminate\Cookie\CookieServiceProvider::class,
Illuminate\Database\DatabaseServiceProvider::class,
Illuminate\Encryption\EncryptionServiceProvider::class,
Illuminate\Filesystem\FilesystemServiceProvider::class,
Illuminate\Foundation\Providers\FoundationServiceProvider::class,
Illuminate\Hashing\HashServiceProvider::class,
Illuminate\Mail\MailServiceProvider::class,
Illuminate\Notifications\NotificationServiceProvider::class,
Illuminate\Pagination\PaginationServiceProvider::class,
Illuminate\Pipeline\PipelineServiceProvider::class,
Illuminate\Queue\QueueServiceProvider::class,
Illuminate\Redis\RedisServiceProvider::class,
Illuminate\Auth\Passwords>PasswordResetServiceProvider::class,
Illuminate\Session\SessionServiceProvider::class,
Illuminate\Translation\TranslationServiceProvider::class,
Illuminate\Validation\ValidationServiceProvider::class,
Illuminate\View\ViewServiceProvider::class,

/*
 * Package Service Providers...
 */

/*
 * Application Service Providers...
 */
App\Providers\AppServiceProvider::class,
App\Providers\AuthServiceProvider::class,
// App\Providers\BroadcastServiceProvider::class,
App\Providers\EventServiceProvider::class,
App\Providers\RouteServiceProvider::class,

],

/*
|-----
| Class Aliases
|-----
|
| This array of class aliases will be registered when this application
| is started. However, feel free to register as many as you wish as
| the aliases are "lazy" loaded so they don't hinder performance.
|
*/

'aliases' => [

```

```
'App' => Illuminate\Support\Facades\App::class,  
'Arr' => Illuminate\Support\Arr::class,  
'Artisan' => Illuminate\Support\Facades\Artisan::class,  
'Auth' => Illuminate\Support\Facades\Auth::class,  
'Blade' => Illuminate\Support\Facades\Blade::class,  
'Broadcast' => Illuminate\Support\Facades\Broadcast::class,  
'Bus' => Illuminate\Support\Facades\Bus::class,  
'Cache' => Illuminate\Support\Facades\Cache::class,  
'Config' => Illuminate\Support\Facades\Config::class,  
'Cookie' => Illuminate\Support\Facades\Cookie::class,  
'Crypt' => Illuminate\Support\Facades\Crypt::class,  
'DB' => Illuminate\Support\Facades\DB::class,  
'Eloquent' => Illuminate\Database\Eloquent\Model::class,  
'Event' => Illuminate\Support\Facades\Event::class,  
'File' => Illuminate\Support\Facades\File::class,  
'Gate' => Illuminate\Support\Facades\Gate::class,  
'Hash' => Illuminate\Support\Facades\Hash::class,  
'Http' => Illuminate\Support\Facades\Http::class,  
'Lang' => Illuminate\Support\Facades\Lang::class,  
'Log' => Illuminate\Support\Facades\Log::class,  
'Mail' => Illuminate\Support\Facades\Mail::class,  
'Notification' => Illuminate\Support\Facades\Notification::class,  
'Password' => Illuminate\Support\Facades>Password::class,  
'Queue' => Illuminate\Support\Facades\Queue::class,  
'Redirect' => Illuminate\Support\Facades\Redirect::class,  
'Redis' => Illuminate\Support\Facades\Redis::class,  
'Request' => Illuminate\Support\Facades\Request::class,  
'Response' => Illuminate\Support\Facades\Response::class,  
'Route' => Illuminate\Support\Facades\Route::class,  
'Schema' => Illuminate\Support\Facades\Schema::class,  
'Session' => Illuminate\Support\Facades\Session::class,  
'Storage' => Illuminate\Support\Facades\Storage::class,  
'Str' => Illuminate\Support\Str::class,  
'URL' => Illuminate\Support\Facades\URL::class,  
'Validator' => Illuminate\Support\Facades\Validator::class,  
'View' => Illuminate\Support\Facades\View::class,
```

```
],
```

```
];
```

Datei: auth.php

```
<?php
```

```
return [
```

```
    /*
```

```
    |-----
```

```
    | Authentication Language Lines
```

```
    |-----
```

```
    |
```

| The following language lines are used during authentication for various  
| messages that we need to display to the user. You are free to modify  
| these language lines according to your application's requirements.

|  
\*/

'failed' => 'These credentials do not match our records.',  
'throttle' => 'Too many login attempts. Please try again in :seconds seconds.',  
'notactivated' => 'This account has not been activated yet.'

];

Datei: backgroundRecolorMixin.js

```
export default {  
  
  data: () => ({  
    htmlTag: null,  
    startPosition: null,  
    supportsStartPosition: null,  
  }),  
  
  created() {  
    /**  
     * @todo if there is nothing to scroll, turn off overscroll  
     */  
  
    this.htmlTag = document.querySelector('html');  
    this.addWheelEventListener();  
  },  
  
  methods: {  
    // addTouchEventListeners() {  
    //   let _startY;  
  
    //   this.htmlTag.addEventListener('touchstart', e => _startY = e.touches[0].pageY, {passive:  
true});  
  
    //   this.htmlTag.addEventListener('touchmove', e => {  
    //     const y = e.touches[0].pageY;  
  
    //     this.setOverscroll(y < _startY);  
    //   }, {passive: true});  
  
    //   this.htmlTag.addEventListener('touchend', () => setTimeout(() => {  
    //     this.setOverscroll(false);  
    //   }, 250), {passive: true});  
    // },  
  
    addWheelEventListener() {  
      window.addEventListener('scroll', (e) => {
```

```

    this.htmlTag.removeEventListener('wheel', this.handleWheel);

    if (document.querySelector('html').classList.contains('overflow-y-hidden')) {
        e.preventDefault();
        // window.scrollTo(0, 0);
    }

    let size_difference = (document.body.scrollHeight - document.documentElement.clientHeight)
/ 4;

    this.setOverscroll(window.scrollY > size_difference);

    }, { passive: true });

    this.htmlTag.addEventListener('wheel', this.handleWheel, {passive: true});
},

handleWheel(e) {

    if (!this.startPosition) this.startPosition = e.screenY - e.clientY;
    else if (this.startPosition !== e.screenY - e.clientY) this.supportsStartPosition = true;

    if (this.supportsStartPosition) {
        this.setOverscroll(this.startPosition > (e.screenY - e.clientY));
    }
},

setOverscroll(is_overscrolled) {
    this.htmlTag.classList.toggle('scrolled', is_overscrolled);
},
},
}

```

Datei: bootstrap.js

```

window._ = require('lodash');

/**
 * We'll load jQuery and the Bootstrap jQuery plugin which provides support
 * for JavaScript based Bootstrap features such as modals and tabs. This
 * code may be modified to fit the specific needs of your application.
 */

try {
    // window.Popper = require('popper.js').default;
    // window.$ = window.jQuery = require('jquery');

    require('bootstrap');
} catch (e) {}

/**
 * We'll load the axios HTTP library which allows us to easily issue requests
 * to our Laravel back-end. This library automatically handles sending the

```

```

* CSRF token as a header based on the value of the "XSRF" token cookie.
*/

window.axios = require('axios');

window.axios.defaults.headers.common['X-Requested-With'] = 'XMLHttpRequest';

/**
 * Echo exposes an expressive API for subscribing to channels and listening
 * for events that are broadcast by Laravel. Echo and event broadcasting
 * allows your team to easily build robust real-time web applications.
 */

// import Echo from 'laravel-echo';

// window.Pusher = require('pusher-js');

// window.Echo = new Echo({
//   broadcaster: 'pusher',
//   key: process.env.MIX_PUSHER_APP_KEY,
//   cluster: process.env.MIX_PUSHER_APP_CLUSTER,
//   forceTLS: true
// });

```

Datei: broadcasting.php

```
<?php
```

```
return [
```

```

    /*
    |-----
    | Default Broadcaster
    |-----
    |
    | This option controls the default broadcaster that will be used by the
    | framework when an event needs to be broadcast. You may set this to
    | any of the connections defined in the "connections" array below.
    |
    | Supported: "pusher", "redis", "log", "null"
    |
    */

    'default' => env('BROADCAST_DRIVER', 'null'),

    /*
    |-----
    | Broadcast Connections
    |-----
    |
    | Here you may define all of the broadcast connections that will be used
    | to broadcast events to other systems or over websockets. Samples of

```

| each available type of connection are provided inside this array.

|  
\*/

```
'connections' => [  
  
    'pusher' => [  
        'driver' => 'pusher',  
        'key' => env('PUSHER_APP_KEY'),  
        'secret' => env('PUSHER_APP_SECRET'),  
        'app_id' => env('PUSHER_APP_ID'),  
        'options' => [  
            'cluster' => env('PUSHER_APP_CLUSTER'),  
            'useTLS' => true,  
        ],  
    ],  
  
    'redis' => [  
        'driver' => 'redis',  
        'connection' => 'default',  
    ],  
  
    'log' => [  
        'driver' => 'log',  
    ],  
  
    'null' => [  
        'driver' => 'null',  
    ],  
  
],  
];
```

Datei: cache.php

```
<?php
```

```
use Illuminate\Support\Str;
```

```
return [  
  
    /*  
    |-----  
    | Default Cache Store  
    |-----  
    |  
    | This option controls the default cache connection that gets used while  
    | using this caching library. This connection is used when another is  
    | not explicitly specified when executing a given caching function.  
    |  
    | Supported: "apc", "array", "database", "file",
```

```

|     "memcached", "redis", "dynamodb"
|
*/

'default' => env('CACHE_DRIVER', 'file'),

/*
|-----|
| Cache Stores
|-----|
|
| Here you may define all of the cache "stores" for your application as
| well as their drivers. You may even define multiple stores for the
| same cache driver to group types of items stored in your caches.
|
*/

'stores' => [

    'apc' => [
        'driver' => 'apc',
    ],

    'array' => [
        'driver' => 'array',
        'serialize' => false,
    ],

    'database' => [
        'driver' => 'database',
        'table' => 'cache',
        'connection' => null,
    ],

    'file' => [
        'driver' => 'file',
        'path' => storage_path('framework/cache/data'),
    ],

    'memcached' => [
        'driver' => 'memcached',
        'persistent_id' => env('MEMCACHED_PERSISTENT_ID'),
        'sas' => [
            env('MEMCACHED_USERNAME'),
            env('MEMCACHED_PASSWORD'),
        ],
        'options' => [
            // Memcached::OPT_CONNECT_TIMEOUT => 2000,
        ],
        'servers' => [
            [
                'host' => env('MEMCACHED_HOST', '127.0.0.1'),
                'port' => env('MEMCACHED_PORT', 11211),
                'weight' => 100,
            ]
        ]
    ]
]

```

```

        ],
    ],
],

'redis' => [
    'driver' => 'redis',
    'connection' => 'cache',
],

'dynamodb' => [
    'driver' => 'dynamodb',
    'key' => env('AWS_ACCESS_KEY_ID'),
    'secret' => env('AWS_SECRET_ACCESS_KEY'),
    'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),
    'table' => env('DYNAMODB_CACHE_TABLE', 'cache'),
    'endpoint' => env('DYNAMODB_ENDPOINT'),
],

],

/*
|-----
| Cache Key Prefix
|-----
|
| When utilizing a RAM based store such as APC or Memcached, there might
| be other applications utilizing the same cache. So, we'll specify a
| value to get prefixed to all our keys so we can avoid collisions.
|
*/

'prefix' => env('CACHE_PREFIX', Str::slug(env('APP_NAME', 'laravel'), '_').'_cache'),

];

```

Date: channels.php

```
<?php
```

```
use Illuminate\Support\Facades\Broadcast;
```

```

/*
|-----
| Broadcast Channels
|-----
|
| Here you may register all of the event broadcasting channels that your
| application supports. The given channel authorization callbacks are
| used to check if an authenticated user can listen to the channel.
|
*/

```



```
Broadcast::channel('App.User.{id}', function ($user, $id) {
    return (int) $user->id === (int) $id;
});
```

Datei: connection.js

```
import axios from 'axios';

const axiosInstance = axios.create({
  baseURL: process.env.VUE_APP_BACKEND,
  headers: {
    common: {
      'X-Requested-With': 'XMLHttpRequest',
      'Content-Type': 'application/json'
    }
  },
  withCredentials: true
});

export default axiosInstance;
```

Datei: console.php

```
<?php

use Illuminate\Foundation\Inspiring;
use Illuminate\Support\Facades\Artisan;

/*
|-----
| Console Routes
|-----
|
| This file is where you may define all of your Closure based console
| commands. Each Closure is bound to a command instance allowing a
| simple approach to interacting with each command's IO methods.
|
*/

Artisan::command('inspire', function () {
    $this->comment(Inspiring::quote());
})->describe('Display an inspiring quote');
```

Datei: cors.php

```
<?php
```

```
return [
```

```

/*
|-----|
| Cross-Origin Resource Sharing (CORS) Configuration
|-----|
|
| Here you may configure your settings for cross-origin resource sharing
| or "CORS". This determines what cross-origin operations may execute
| in web browsers. You are free to adjust these settings as needed.
|
| To learn more: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
|
*/

'paths' => [
    'api/*',
    '/account',
    '/sanctum/csrf-cookie'
],

'allowed_methods' => ['*'],

'allowed_origins' => explode(',', env('ALLOWED_ORIGINS', '*')),

'allowed_origins_patterns' => [],

'allowed_headers' => ['*'],

'exposed_headers' => [],

'max_age' => 0,

'supports_credentials' => true
];

```

Date: database.php

```
<?php
```

```
use Illuminate\Support\Str;
```

```
return [
```

```

/*
|-----|
| Default Database Connection Name
|-----|
|
| Here you may specify which of the database connections below you wish
| to use as your default connection for all database work. Of course
| you may use many connections at once using the Database library.
|

```

```

*/

'default' => env('DB_CONNECTION', 'mysql'),

/*
|-----
| Database Connections
|-----
|
| Here are each of the database connections setup for your application.
| Of course, examples of configuring each database platform that is
| supported by Laravel is shown below to make development simple.
|
|
| All database work in Laravel is done through the PHP PDO facilities
| so make sure you have the driver for your particular database of
| choice installed on your machine before you begin development.
|
*/

'connections' => [

    'sqlite' => [
        'driver' => 'sqlite',
        'url' => env('DATABASE_URL'),
        'database' => env('DB_DATABASE', database_path('database.sqlite')),
        'prefix' => "",
        'foreign_key_constraints' => env('DB_FOREIGN_KEYS', true),
    ],

    'mysql' => [
        'driver' => 'mysql',
        'url' => env('DATABASE_URL'),
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => 'utf8mb4',
        'collation' => 'utf8mb4_unicode_ci',
        'prefix' => "",
        'prefix_indexes' => true,
        'strict' => true,
        'engine' => null,
        'options' => extension_loaded('pdo_mysql') ? array_filter([
            PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
        ]) : [],
    ],

    'pgsql' => [
        'driver' => 'pgsql',
        'url' => env('DATABASE_URL'),
        'host' => env('DB_HOST', '127.0.0.1'),

```

```

        'port' => env('DB_PORT', '5432'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'charset' => 'utf8',
        'prefix' => '',
        'prefix_indexes' => true,
        'schema' => 'public',
        'sslmode' => 'prefer',
    ],

    'sqlsrv' => [
        'driver' => 'sqlsrv',
        'url' => env('DATABASE_URL'),
        'host' => env('DB_HOST', 'localhost'),
        'port' => env('DB_PORT', '1433'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'charset' => 'utf8',
        'prefix' => '',
        'prefix_indexes' => true,
    ],
],

/*
|-----
| Migration Repository Table
|-----
|
| This table keeps track of all the migrations that have already run for
| your application. Using this information, we can determine which of
| the migrations on disk haven't actually been run in the database.
|
*/

'migrations' => 'migrations',

/*
|-----
| Redis Databases
|-----
|
| Redis is an open source, fast, and advanced key-value store that also
| provides a richer body of commands than a typical key-value system
| such as APC or Memcached. Laravel makes it easy to dig right in.
|
*/

'redis' => [
    'client' => env('REDIS_CLIENT', 'phpredis'),

```

```

'options' => [
    'cluster' => env('REDIS_CLUSTER', 'redis'),
    'prefix' => env('REDIS_PREFIX', Str::slug(env('APP_NAME', 'laravel'), '_').'_database_'),
],

'default' => [
    'url' => env('REDIS_URL'),
    'host' => env('REDIS_HOST', '127.0.0.1'),
    'password' => env('REDIS_PASSWORD', null),
    'port' => env('REDIS_PORT', '6379'),
    'database' => env('REDIS_DB', '0'),
],

'cache' => [
    'url' => env('REDIS_URL'),
    'host' => env('REDIS_HOST', '127.0.0.1'),
    'password' => env('REDIS_PASSWORD', null),
    'port' => env('REDIS_PORT', '6379'),
    'database' => env('REDIS_CACHE_DB', '1'),
],

],

];

```

Datei: disposable-email.php

```
<?php
```

```
return [
```

```

/*
|-----
| JSON Source URL
|-----
|
| The source URL yielding a list of disposable email domains. Change this
| to whatever source you like. Just make sure it returns a JSON array.
|
| A sensible default is provided using jsDelivr's services. jsDelivr is
| a free service, so there are no uptime or support guarantees.
|
*/

```

```
'source' => 'https://th-wildau.de/security/disposable_domains.json',
```

```

/*
|-----
| Storage Path
|-----
|
| The location where the retrieved domains list should be stored locally.

```

```

| The path should be accessible and writable by the web server. A good
| place for storing the list is in the framework's own storage path.
|
*/

'storage' => storage_path('framework/disposable_domains.json'),

/*
|-----
| Cache Configuration
|-----
|
| Here you may define whether the disposable domains list should be cached.
| If you disable caching or when the cache is empty, the list will be
| fetched from local storage instead.
|
| You can optionally specify an alternate cache connection or modify the
| cache key as desired.
|
*/

'cache' => [
    'enabled' => true,
    'store' => 'default',
    'key' => 'disposable_email.domains',
],
];

```

Datei: filesystems.php

```
<?php
```

```

return [

    /*
    |-----
    | Default Filesystem Disk
    |-----
    |
    | Here you may specify the default filesystem disk that should be used
    | by the framework. The "local" disk, as well as a variety of cloud
    | based disks are available to your application. Just store away!
    |
    */

    'default' => env('FILESYSTEM_DRIVER', 'local'),

    /*
    |-----
    | Default Cloud Filesystem Disk
    |-----

```

```

|
| Many applications store files both locally and in the cloud. For this
| reason, you may specify a default "cloud" driver here. This driver
| will be bound as the Cloud disk implementation in the container.
|
|*/

'cloud' => env('FILESYSTEM_CLOUD', 's3'),

/*
|-----
| Filesystem Disks
|-----
|
| Here you may configure as many filesystem "disks" as you wish, and you
| may even configure multiple disks of the same driver. Defaults have
| been setup for each driver as an example of the required options.
|
| Supported Drivers: "local", "ftp", "sftp", "s3"
|
|*/

'disks' => [

  'local' => [
    'driver' => 'local',
    'root' => storage_path('app'),
  ],

  'public' => [
    'driver' => 'local',
    'root' => storage_path('app/public'),
    'url' => env('APP_URL').'/storage',
    'visibility' => 'public',
  ],

  's3' => [
    'driver' => 's3',
    'key' => env('AWS_ACCESS_KEY_ID'),
    'secret' => env('AWS_SECRET_ACCESS_KEY'),
    'region' => env('AWS_DEFAULT_REGION'),
    'bucket' => env('AWS_BUCKET'),
    'url' => env('AWS_URL'),
    'endpoint' => env('AWS_ENDPOINT'),
  ],

],

/*
|-----
| Symbolic Links
|-----
|
| Here you may configure the symbolic links that will be created when the

```

| `storage:link` Artisan command is executed. The array keys should be  
| the locations of the links and the values should be their targets.

```
|  
*/  
  
'links' => [  
    public_path('storage') => storage_path('app/public'),  
],  
  
];
```

Datei: hashing.php

```
<?php
```

```
return [  
  
    /*  
    |-----  
    | Default Hash Driver  
    |-----  
    |  
    | This option controls the default hash driver that will be used to hash  
    | passwords for your application. By default, the bcrypt algorithm is  
    | used; however, you remain free to modify this option if you wish.  
    |  
    | Supported: "bcrypt", "argon", "argon2id"  
    |  
    */  
  
    'driver' => 'bcrypt',  
  
    /*  
    |-----  
    | Bcrypt Options  
    |-----  
    |  
    | Here you may specify the configuration options that should be used when  
    | passwords are hashed using the Bcrypt algorithm. This will allow you  
    | to control the amount of time it takes to hash the given password.  
    |  
    */  
  
    'bcrypt' => [  
        'rounds' => env('BCRYPT_ROUNDS', 10),  
    ],  
  
    /*  
    |-----  
    | Argon Options  
    |-----  
    |
```



| Here you may specify the configuration options that should be used when  
| passwords are hashed using the Argon algorithm. These will allow you  
| to control the amount of time it takes to hash the given password.

```
|  
*/  
  
'argon' => [  
  'memory' => 1024,  
  'threads' => 2,  
  'time' => 2,  
],  
  
];
```

Datei: index.js

```
const state = {  
  token: null,  
  total_encounters: 0,  
  user: {}  
}  
  
const getters = {  
  
}  
  
const mutations = {  
  setToken: (state, token) => {  
    state.token = token;  
  },  
  setTotalEncounters: (state, total_encounters) => {  
    state.total_encounters = total_encounters;  
  },  
  setUser: (state, user) => {  
    state.user = user;  
  }  
}  
  
const actions = {  
  setToken({ commit }, token) {  
    commit('setToken', token);  
  },  
  setTotalEncounters({ commit }, total_encounters) {  
    commit('setTotalEncounters', total_encounters);  
  },  
  setUser({ commit }, user) {  
    commit('setUser', user);  
  }  
}  
  
export default {  
  // namespace: true,
```

```
    mutations,  
    state,  
    actions,  
    getters  
  }  
}
```

Datei: installMixin.js

```
import Vue from 'vue';  
import Browser from "browser";  
import { parse, format, isPast, addMinutes } from "date-fns"  
// import { parse, format, isValid, isPast, addMinutes } from "date-fns"  
import { mapState } from 'vuex';
```

```
let observable = Vue.observable({  
  deferred_prompt: null  
})
```

```
export default {  
  data: () => ({  
    browser: null,  
  
    installing: false,  
    already_installed: false,  
    // deferred_prompt: null  
  }),
```

```
  created() {  
    this.browser = Browser.getParser(window.navigator.userAgent);  
  
    // if (!this.isListeningToInstallEvents()) {  
    this.listenToInstallEvents();  
    // }  
  },
```

```
  methods: {  
    // isListeningToInstallEvents() {  
    //   return 'beforeinstallprompt' in getEventListeners(window);  
    // },  
    listenToInstallEvents() {  
      window.addEventListener('beforeinstallprompt', (e) => {  
        console.log("beforeinstallprompt", e);  
        this.installing = false;  
        // Prevent the mini-infobar from appearing on mobile  
        e.preventDefault();  
  
        if (this.already_installed) {  
          console.log("Ignore beforeinstallprompt");  
          return;  
        }  
      })
```

```
      // Stash the event so it can be triggered later.  
      this.deferred_prompt = e;
```

```

// Update UI notify the user they can install the PWA

// Wait for the user to respond to the prompt
this.deferred_prompt.userChoice.then((choiceResult) => {
  if (choiceResult.outcome === 'accepted') {
    console.log('User accepted the A2HS prompt');
  } else {
    console.log('User dismissed the A2HS prompt');
  }

  this.installing = false;
  this.deferred_prompt = null;
});
});

window.addEventListener("appinstalled", () => {
  console.log("App is installed already");
  this.installing = false;
  this.deferred_prompt = null;
  this.already_installed = true;
});
},

deferInstall() {
  const later = format(addMinutes(new Date(), 1), 'yyyy-MM-dd HH:mm');
  // const later = format(addDays(new Date(), 1), 'yyyy-MM-dd HH:mm');

  this.setPromptInstallDate(later);
},
},

computed: {
  ...mapState({
    prompt_install_date: (state) => parse(state.prompt_install_date, 'yyyy-MM-dd HH:mm', new
Date())
  }),
  deferred_prompt: {
    get(){
      return observable.deferred_prompt
    },
    set(val){
      observable.deferred_prompt = val
    }
  },
  isSafari() {
    return this.bowser.satisfies({
      mobile: {
        safari: '>=11.3',
      }
    });
  },
  isFirefox() {
    return this.bowser.satisfies({

```

```

        android: {
            firefox: '>=58',
        },
    });
},
browser() {
    if (this.isFirefox) return 'Firefox';
    if (this.isSafari) return 'Safari';
    return 'Chrome';
},
showInstall() {
    return false;
    // console.log("Show install?", this.prompt_install_date);
    // if (!isValid(this.prompt_install_date)) {
    //     const today = format(new Date(), 'yyyy-MM-dd HH:mm');
    //     this.setPromptInstallDate(today);
    //     return false;
    // }

    // console.log("Is in the past?", isPast(this.prompt_install_date));

    // if (!isPast(this.prompt_install_date)) {
    //     return false;
    // }

    // return this.canBeInstalled;
},
installWasDeferred() {
    return this.canBeInstalled
        && this.prompt_install_date
        && !isPast(this.prompt_install_date);
},
canBeInstalled() {
    /**
     * @todo TEMP
     */
    return (!this.isStandalone && (this.isFirefox || this.isSafari)) || this.deferred_prompt;
},
isStandalone() {
    const urlParams = new URLSearchParams(window.location.search);
    const source = urlParams.get('source');

    return navigator.standalone
        || source === 'pwa'
        || window.matchMedia('(display-mode: standalone)').matches;
},
}
}
}

```

Datei: logging.php

<?php

```
use Monolog\Handler\NullHandler;
use Monolog\Handler\StreamHandler;
use Monolog\Handler\SyslogUdpHandler;
```

```
return [
```

```
    /*
    |-----
    | Default Log Channel
    |-----
    |
    | This option defines the default log channel that gets used when writing
    | messages to the logs. The name specified in this option should match
    | one of the channels defined in the "channels" configuration array.
    |
    */
```

```
    'default' => env('LOG_CHANNEL', 'stack'),
```

```
    /*
    |-----
    | Log Channels
    |-----
    |
    | Here you may configure the log channels for your application. Out of
    | the box, Laravel uses the Monolog PHP logging library. This gives
    | you a variety of powerful log handlers / formatters to utilize.
    |
    | Available Drivers: "single", "daily", "slack", "syslog",
    |                   "errorlog", "monolog",
    |                   "custom", "stack"
    |
    */
```

```
    'channels' => [
        'stack' => [
            'driver' => 'stack',
            'channels' => ['single'],
            'ignore_exceptions' => false,
        ],

        'single' => [
            'driver' => 'single',
            'path' => storage_path('logs/laravel.log'),
            'level' => 'debug',
        ],

        'daily' => [
            'driver' => 'daily',
            'path' => storage_path('logs/laravel.log'),
            'level' => 'debug',
            'days' => 14,
        ],
    ],
```

```
'slack' => [
    'driver' => 'slack',
    'url' => env('LOG_SLACK_WEBHOOK_URL'),
    'username' => 'Laravel Log',
    'emoji' => ':boom:',
    'level' => 'critical',
],

'papertrail' => [
    'driver' => 'monolog',
    'level' => 'debug',
    'handler' => SyslogUdpHandler::class,
    'handler_with' => [
        'host' => env('PAPERTRAIL_URL'),
        'port' => env('PAPERTRAIL_PORT'),
    ],
],

'stderr' => [
    'driver' => 'monolog',
    'handler' => StreamHandler::class,
    'formatter' => env('LOG_STDERR_FORMATTER'),
    'with' => [
        'stream' => 'php://stderr',
    ],
],

'syslog' => [
    'driver' => 'syslog',
    'level' => 'debug',
],

'errorlog' => [
    'driver' => 'errorlog',
    'level' => 'debug',
],

'null' => [
    'driver' => 'monolog',
    'handler' => NullHandler::class,
],

'emergency' => [
    'path' => storage_path('logs/laravel.log'),
],
],

];
```

Datei: mail.php

<?php

```

return [

    /*
    |-----
    | Default Mailer
    |-----
    |
    | This option controls the default mailer that is used to send any email
    | messages sent by your application. Alternative mailers may be setup
    | and used as needed; however, this mailer will be used by default.
    |
    */

    'default' => env('MAIL_MAILER', 'smtp'),

    /*
    |-----
    | Mailer Configurations
    |-----
    |
    | Here you may configure all of the mailers used by your application plus
    | their respective settings. Several examples have been configured for
    | you and you are free to add your own as your application requires.
    |
    | Laravel supports a variety of mail "transport" drivers to be used while
    | sending an e-mail. You will specify which one you are using for your
    | mailers below. You are free to add additional mailers as required.
    |
    | Supported: "smtp", "sendmail", "mailgun", "ses",
    |             "postmark", "log", "array"
    |
    */

    'mailers' => [
        'smtp' => [
            'transport' => 'smtp',
            'host' => env('MAIL_HOST', 'smtp.mailgun.org'),
            'port' => env('MAIL_PORT', 587),
            'encryption' => env('MAIL_ENCRYPTION', 'tls'),
            'username' => env('MAIL_USERNAME'),
            'password' => env('MAIL_PASSWORD'),
            'timeout' => null,
            'auth_mode' => null,
        ],

        'ses' => [
            'transport' => 'ses',
        ],

        'mailgun' => [
            'transport' => 'mailgun',
        ],
    ],

```

```

    'postmark' => [
        'transport' => 'postmark',
    ],

    'sendmail' => [
        'transport' => 'sendmail',
        'path' => '/usr/sbin/sendmail -bs',
    ],

    'log' => [
        'transport' => 'log',
        'channel' => env('MAIL_LOG_CHANNEL'),
    ],

    'array' => [
        'transport' => 'array',
    ],
],

/*
|-----
| Global "From" Address
|-----
|
| You may wish for all e-mails sent by your application to be sent from
| the same address. Here, you may specify a name and address that is
| used globally for all e-mails that are sent by your application.
|
*/

'from' => [
    'address' => env('MAIL_FROM_ADDRESS', 'hello@example.com'),
    'name' => env('MAIL_FROM_NAME', 'Example'),
],

/*
|-----
| Markdown Mail Settings
|-----
|
| If you are using Markdown based email rendering, you may configure your
| theme and component paths here, allowing you to customize the design
| of the emails. Or, you may simply stick with the Laravel defaults!
|
*/

'markdown' => [
    'theme' => 'default',

    'paths' => [
        resource_path('views/vendor/mail'),
    ],
],

```



```
};
```

Datei: main.js

```
import Vue from 'vue'
import App from './App.vue'
import './registerServiceWorker'
import store from './store';
import router from './router'
import vuetify from './plugins/vuetify';
import storageMixin from '@mixins/storageMixin';

Vue.mixin(storageMixin);

Vue.config.productionTip = false;

const requireComponent = require.context(
  // The relative path of the components folder
  './components/common',
  // Whether or not to look in subfolders
  false,
  // The regular expression used to match base component filenames
  /[A-Z]\w+\.(vue|js)$/
)

requireComponent.keys().forEach(fileName => {
  // Get component config
  const componentConfig = requireComponent(fileName)

  // Get PascalCase name of component
  const componentName = fileName
    .split('/')
    .pop()
    .replace(/^\.\w+$/, "");

  // Register component globally
  Vue.component(
    componentName,
    // Look for the component options on `default`, which will
    // exist if the component was exported with `export default`,
    // otherwise fall back to module's root.
    componentConfig.default || componentConfig
  )
});

const v = new Vue({
  router,
  store,
  vuetify,
  render: h => h(App)
}).$mount('#app')
```

```
export default v;
```

Datei: pagination.php

```
<?php
```

```
return [
```

```
    /*
    |-----
    | Pagination Language Lines
    |-----
    |
    | The following language lines are used by the paginator library to build
    | the simple pagination links. You are free to change them to anything
    | you want to customize your views to better match your application.
    |
    */

    'previous' => '&laquo; Previous',
    'next' => 'Next &raquo;',
```

```
];
```

Datei: passwords.php

```
<?php
```

```
return [
```

```
    /*
    |-----
    | Password Reset Language Lines
    |-----
    |
    | The following language lines are the default lines which match reasons
    | that are given by the password broker for a password update attempt
    | has failed, such as for an invalid token or invalid new password.
    |
    */

    'reset' => 'Your password has been reset!',
    'sent' => 'We have emailed your password reset link!',
    'throttled' => 'Please wait before retrying.',
    'token' => 'This password reset token is invalid.',
    'user' => "We can't find a user with that email address.",
```

```
];
```

Datei: queue.php

```
<?php
```

```
return [
```

```
    /*
```

```
    |-----  
    | Default Queue Connection Name  
    |-----  
    |
```

```
    | Laravel's queue API supports an assortment of back-ends via a single  
    | API, giving you convenient access to each back-end using the same  
    | syntax for every one. Here you may define a default connection.  
    |
```

```
    */
```

```
    'default' => env('QUEUE_CONNECTION', 'sync'),
```

```
    /*
```

```
    |-----  
    | Queue Connections  
    |-----  
    |
```

```
    | Here you may configure the connection information for each server that  
    | is used by your application. A default configuration has been added  
    | for each back-end shipped with Laravel. You are free to add more.  
    |
```

```
    | Drivers: "sync", "database", "beanstalkd", "sqs", "redis", "null"  
    |
```

```
    */
```

```
    'connections' => [
```

```
        'sync' => [  
            'driver' => 'sync',  
        ],
```

```
        'database' => [  
            'driver' => 'database',  
            'table' => 'jobs',  
            'queue' => 'default',  
            'retry_after' => 90,  
        ],
```

```
        'beanstalkd' => [  
            'driver' => 'beanstalkd',  
            'host' => 'localhost',  
            'queue' => 'default',  
            'retry_after' => 90,  
            'block_for' => 0,  
        ],
```

```
        'sqs' => [
```

```

    'driver' => 'sqs',
    'key' => env('AWS_ACCESS_KEY_ID'),
    'secret' => env('AWS_SECRET_ACCESS_KEY'),
    'prefix' => env('SQS_PREFIX', 'https://sqs.us-east-1.amazonaws.com/your-account-id'),
    'queue' => env('SQS_QUEUE', 'your-queue-name'),
    'suffix' => env('SQS_SUFFIX'),
    'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),
  ],

  'redis' => [
    'driver' => 'redis',
    'connection' => 'default',
    'queue' => env('REDIS_QUEUE', 'default'),
    'retry_after' => 90,
    'block_for' => null,
  ],

],

/*
|-----
| Failed Queue Jobs
|-----
|
| These options configure the behavior of failed queue job logging so you
| can control which database and table are used to store the jobs that
| have failed. You may change them to any database / table you wish.
|
*/

'failed' => [
  'driver' => env('QUEUE_FAILED_DRIVER', 'database'),
  'database' => env('DB_CONNECTION', 'mysql'),
  'table' => 'failed_jobs',
],

];

```

Datei: registerServiceWorker.js

```

/* eslint-disable no-console */

import { register } from 'register-service-worker'

if (process.env.NODE_ENV === 'production') {
  register(`${process.env.BASE_URL}service-worker.js`, {
    ready () {
      console.log(
        'App is being served from cache by a service worker.\n' +
        'For more details, visit https://goo.gl/AFskqB'
      )
    },
  },

```

```

registered () {
  console.log('Service worker has been registered.')
},
cached () {
  console.log('Content has been cached for offline use.')
},
updatefound () {
  console.log('New content is downloading.')
},
updated() {
  console.log('New content is available; please refresh.')
},
offline () {
  console.log('No internet connection found. App is running in offline mode.')
},
error (error) {
  console.error('Error during service worker registration:', error)
}
});
}

```

Datei: router.js

```

import Vue from 'vue'
import VueRouter from 'vue-router'

import store from '@/store';

Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'home',
    component: () => import(/* webpackChunkName: 'home' */ '@/views/Home'),
    children: [
      {
        path: 'new',
        name: 'add',
        props: route => ({ query: route.query.room }),
        component: () => import(/* webpackChunkName: 'add' */ '@/views/Add'),
      },
      {
        path: 'settings',
        name: 'settings',
        component: () => import(/* webpackChunkName: 'settings' */ '@/views/Settings'),
      },
      {
        path: 'suggested',
        name: 'suggested',
        component: () => import(/* webpackChunkName: 'suggested' */ '@/views/Suggested'),
      },
      {

```

```

    path: 'account',
    name: 'account',
    component: () => import(/* webpackChunkName: 'user' */ '@/views/Register'),
  },
  {
    path: 'scanner',
    name: 'scanner',
    component: () => import(/* webpackChunkName: 'scanner' */ '@/views/QrScanner'),
  }
]
},
];

```

```

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
});

```

```

router.beforeEach((to, from, next) => {
  if (to.name === 'account' && store.getters.isLoggedIn) {
    next({ name: 'home' });
  }
  else if (to.name === 'home' && store.getters.mustLogin) {
    next();
    router.push({ name: 'account' });
  }
  else if (to.name !== 'account' && store.getters.mustLogin) {
    next({ name: 'account' });
  }
  else {
    next();
  }
});

```

```
export default router
```

Datei: sanctum.php

```
<?php
```

```
return [
```

```

/*
|-----
| Stateful Domains
|-----
|
| Requests from the following domains / hosts will receive stateful API
| authentication cookies. Typically, these should include your local
| and production domains which access your API via a frontend SPA.
|

```

```

*/

'stateful' => explode(',', env('SANCTUM_STATEFUL_DOMAINS',
'localhost,127.0.0.1,127.0.0.1:8000,::1,localhost:443')),

/*
|-----|
| Expiration Minutes
|-----|
|
| This value controls the number of minutes until an issued token will be
| considered expired. If this value is null, personal access tokens do
| not expire. This won't tweak the lifetime of first-party sessions.
|
*/

'expiration' => null,

/*
|-----|
| Sanctum Middleware
|-----|
|
| When authenticating your first-party SPA with Sanctum you may need to
| customize some of the middleware Sanctum uses while processing the
| request. You may change the middleware listed below as required.
|
*/

'middleware' => [
    'verify_csrf_token' => App\Http\Middleware\VerifyCsrfToken::class,
    'encrypt_cookies' => App\Http\Middleware\EncryptCookies::class,
],
];

```

Date: server.php

```
<?php
```

```

/**
 * Laravel - A PHP Framework For Web Artisans
 *
 * @package Laravel
 * @author Taylor Otwell <taylor@laravel.com>
 */

$uri = urldecode(
    parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH)
);

// This file allows us to emulate Apache's "mod_rewrite" functionality from the

```

```
// built-in PHP web server. This provides a convenient way to test a Laravel
// application without having installed a "real" web server software here.
if ($Suri !== '/' && file_exists(__DIR__.'/public'.$Suri)) {
    return false;
}

require_once __DIR__.'/public/index.php';
```

Datei: service-worker.js

```
/* eslint-disable */

workbox.setConfig({
    debug: false,
});

workbox.core.skipWaiting();
workbox.core.clientsClaim();

workbox.precaching.precacheAndRoute(self.__WB_MANIFEST || []);

workbox.routing.registerRoute(
    /\.(?:png|gif|jpg|jpeg|svg)$/,
    workbox.strategies.staleWhileRevalidate({
        cacheName: 'images',
        plugins: [
            new workbox.expiration.Plugin({
                maxEntries: 60,
                maxAgeSeconds: 30 * 24 * 60 * 60, // 30 Days
            }),
        ],
    }),
);

workbox.routing.registerRoute(
    new RegExp('/api/rooms'),
    workbox.strategies.cacheFirst({
        cacheName: 'corona-app-rooms',
        plugins: [
            new workbox.cacheableResponse.Plugin({
                statuses: [0, 200],
            }),
            new workbox.expiration.Plugin({
                // Cache for 30 minutes
                maxAgeSeconds: 30 * 60,
                maxEntries: 1,
            }),
        ],
    })
);
```



Date: services.php

```
<?php
```

```
return [
```

```
    /*
    |-----
    | Third Party Services
    |-----
    |
    | This file is for storing the credentials for third party services such
    | as Mailgun, Postmark, AWS and more. This file provides the de facto
    | location for this type of information, allowing packages to have
    | a conventional file to locate the various service credentials.
    |
    */

    'mailgun' => [
        'domain' => env('MAILGUN_DOMAIN'),
        'secret' => env('MAILGUN_SECRET'),
        'endpoint' => env('MAILGUN_ENDPOINT', 'api.mailgun.net'),
    ],

    'postmark' => [
        'token' => env('POSTMARK_TOKEN'),
    ],

    'ses' => [
        'key' => env('AWS_ACCESS_KEY_ID'),
        'secret' => env('AWS_SECRET_ACCESS_KEY'),
        'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),
    ],
];
```

Date: session.php

```
<?php
```

```
use Illuminate\Support\Str;
```

```
return [
```

```
    /*
    |-----
    | Default Session Driver
    |-----
    |
    | This option controls the default session "driver" that will be used on
    | requests. By default, we will use the lightweight native driver but
    | you may specify any of the other wonderful drivers provided here.
```

```

|
| Supported: "file", "cookie", "database", "apc",
|           "memcached", "redis", "dynamodb", "array"
|
|*/

'driver' => env('SESSION_DRIVER', 'file'),

/*
|-----|
| Session Lifetime
|-----|
|
| Here you may specify the number of minutes that you wish the session
| to be allowed to remain idle before it expires. If you want them
| to immediately expire on the browser closing, set that option.
|
|*/

'lifetime' => env('SESSION_LIFETIME', 120),

'expire_on_close' => false,

/*
|-----|
| Session Encryption
|-----|
|
| This option allows you to easily specify that all of your session data
| should be encrypted before it is stored. All encryption will be run
| automatically by Laravel and you can use the Session like normal.
|
|*/

'encrypt' => false,

/*
|-----|
| Session File Location
|-----|
|
| When using the native session driver, we need a location where session
| files may be stored. A default has been set for you but a different
| location may be specified. This is only needed for file sessions.
|
|*/

'files' => storage_path('framework/sessions'),

/*
|-----|
| Session Database Connection
|-----|
|

```

```
| When using the "database" or "redis" session drivers, you may specify a  
| connection that should be used to manage these sessions. This should  
| correspond to a connection in your database configuration options.
```

```
|  
*/
```

```
'connection' => env('SESSION_CONNECTION', null),
```

```
/*
```

```
|-----  
| Session Database Table
```

```
|-----  
|
```

```
| When using the "database" session driver, you may specify the table we  
| should use to manage the sessions. Of course, a sensible default is  
| provided for you; however, you are free to change this as needed.
```

```
|  
*/
```

```
'table' => 'sessions',
```

```
/*
```

```
|-----  
| Session Cache Store
```

```
|-----  
|
```

```
| While using one of the framework's cache driven session backends you may  
| list a cache store that should be used for these sessions. This value  
| must match with one of the application's configured cache "stores".
```

```
|  
| Affects: "apc", "dynamodb", "memcached", "redis"
```

```
|  
*/
```

```
'store' => env('SESSION_STORE', null),
```

```
/*
```

```
|-----  
| Session Sweeping Lottery
```

```
|-----  
|
```

```
| Some session drivers must manually sweep their storage location to get  
| rid of old sessions from storage. Here are the chances that it will  
| happen on a given request. By default, the odds are 2 out of 100.
```

```
|  
*/
```

```
'lottery' => [2, 100],
```

```
/*
```

```
|-----  
| Session Cookie Name
```

```
|-----  
|
```

```
| Here you may change the name of the cookie used to identify a session
| instance by ID. The name specified here will get used every time a
| new session cookie is created by the framework for every driver.
```

```
|
| */
```

```
'cookie' => env(
    'SESSION_COOKIE',
    Str::slug(env('APP_NAME', 'laravel'), '_').'_session'
),
```

```
/*
```

```
-----
| Session Cookie Path
```

```
-----
|
```

```
| The session cookie path determines the path for which the cookie will
| be regarded as available. Typically, this will be the root path of
| your application but you are free to change this when necessary.
```

```
|
| */
```

```
'path' => env('APP_FRONTEND_URL', '/'),
```

```
/*
```

```
-----
| Session Cookie Domain
```

```
-----
|
```

```
| Here you may change the domain of the cookie used to identify a session
| in your application. This will determine which domains the cookie is
| available to in your application. A sensible default has been set.
```

```
|
| */
```

```
'domain' => env('SESSION_DOMAIN', null),
```

```
/*
```

```
-----
| HTTPS Only Cookies
```

```
-----
|
```

```
| By setting this option to true, session cookies will only be sent back
| to the server if the browser has a HTTPS connection. This will keep
| the cookie from being sent to you if it can not be done securely.
```

```
|
| */
```

```
'secure' => env('SESSION_SECURE_COOKIE'),
```

```
/*
```

```
-----
| HTTP Access Only
```

```
-----
|
```

```

|
| Setting this value to true will prevent JavaScript from accessing the
| value of the cookie and the cookie will only be accessible through
| the HTTP protocol. You are free to modify this option if needed.
|
| */
'http_only' => true,

/*
|-----
| Same-Site Cookies
|-----
|
| This option determines how your cookies behave when cross-site requests
| take place, and can be used to mitigate CSRF attacks. By default, we
| will set this value to "lax" since this is a secure default value.
|
| Supported: "lax", "strict", "none", null
|
| */

'same_site' => 'strict',
];

```

Datei: statesMixin.js

```

export default {
  data: () => ({
    state: 0,
    state_count: 0
  }),

  methods: {
    setStateCount(count) {
      this.state_count = count;
    },
    setState(state) {
      this.state = state;
    },
    cycleState() {
      this.state = (this.state + 1) % this.state_count;
    },
  }
};

```

Datei: storageMixin.js

```

import { mapActions, mapGetters, mapState } from "vuex";

```

```

export default {
  data: () => ({

  }),
  computed: {
    ...mapState([
      'user',
      'rooms',
      'token',
      'visits',
      'suggested',
      'total_encounters',
      'globalError',
      'direct_contact_consent',
    ]),

    ...mapGetters([
      'mustLogin',
      'isLoggedIn',
    ]),

    isMobile() {
      return this.$vuetify.breakpoint.smAndDown;
    },

    hasVisits() {
      return this.visits.length > 0;
    }
  },

  methods: {
    ...mapActions([
      'setUser',
      'setError',
      'setToken',
      'setTheme',
      'setRooms',
      'getRooms',
      'setTotalEncounters',
      'setPromptInstallDate',
      'setDirectContactConsent',
    ])
  },
};

```

Datei: store.js

```

import Vue from 'vue'
import Vuex from 'vuex'

import createPersistedState from 'vuex-persistedstate'

import { loadRooms } from "@/requests/api";

```

```

Vue.use(Vuex);

const persistedstate = createPersistedState({
  paths: ['token', 'user', 'prompt_install_date', 'theme', 'rooms'],
  key: 'corona-app'
});

const state = {
  token: null,
  total_encounters: 0,
  user: {},
  visits: [],
  direct_contact_consent: null,
  prompt_install_date: null,
  suggested: null,
  globalError: null,
  theme: 'light',
  rooms: null,
  rooms_loading: false,
  rooms_promise: null
}

const getters = {
  isLoggedIn: state => {
    return state.user && (state.user.ldap_id || state.user.first_name);
  },
  mustLogin: (state, getters) => {
    return !getters.isLoggedIn && state.visits.length > 0;
  }
}

const mutations = {
  setToken: (state, token) => {
    state.token = token;
  },
  setTotalEncounters: (state, total_encounters) => {
    state.total_encounters = total_encounters;
  },
  setUser: (state, user) => {
    state.user = user;
  },
  setSuggested: (state, suggested) => {
    state.suggested = suggested;
  },
  setError: (state, error) => {
    state.globalError = error;
  },
  setDirectContactConsent: (state, direct_contact_consent) => {
    state.direct_contact_consent = direct_contact_consent;
  },
  setVisits: (state, visits) => {
    state.visits = visits;
  }
}

```

```

    },
    mutate: (state, payload) => {
      state[payload.property] = payload.value;
    }
  }
}

const actions = {
  setToken({ commit }, token) {
    commit('setToken', token);
  },
  setTotalEncounters({ commit }, total_encounters) {
    commit('setTotalEncounters', total_encounters);
  },
  setUser({ commit }, user) {
    commit('setUser', user);
  },
  setSuggested({ commit }, suggested) {
    commit('setSuggested', suggested);
  },
  setError({ commit }, error) {
    commit('setError', error);
  },
  setDirectContactConsent({ commit }, direct_contact_consent) {
    commit('setDirectContactConsent', direct_contact_consent);
  },
  setTheme({ commit }, theme) {
    commit('mutate', {
      property: 'theme',
      value: theme
    });
  },
  setPromptInstallDate({ commit }, date) {
    commit('mutate', {
      property: 'prompt_install_date',
      value: date
    });
  },
  setVisits({ commit, state }, visits) {
    // Dirty solution that helps the setting of visits remain global,
    // but allows for success animations (e.g. Leave room button)
    // to complete before updating the UI
    if (state.visits.length === 0) {
      commit('setVisits', visits);
    } else {
      setTimeout(() => commit('setVisits', visits), 1000);
    }
  },
  setRooms({ commit }, rooms) {
    commit('mutate', {
      property: 'rooms',
      value: rooms
    });
  },
  getRooms({ commit, state, dispatch }) {

```



```

    if (state.rooms_promise) return state.rooms_promise;

    const rooms_promise = loadRooms();

    rooms_promise
      .then(rooms => {
        dispatch('setRooms', rooms);
        console.log("Rooms promise setRooms", rooms_promise);
      })
      .catch(() => {
        dispatch('setError', 'Raumliste konnte nicht aktualisiert werden. ');
        console.log("Rooms promise setRooms", rooms_promise);
      })
      .finally(() =>
        commit('mutate', {
          property: 'rooms_promise',
          value: null
        })
      );

    commit('mutate', {
      property: 'rooms_promise',
      value: rooms_promise
    });

    return rooms_promise;
  }
}

export default new Vuex.Store({
  mutations,
  state,
  actions,
  getters,
  // modules,
  plugins: [persistedstate]
})

```

Datei: themeMixin.js

```

import { mapState } from "vuex";

export default {
  data: () => ({
    mq: null,
    unsupported: false
  }),

  created() {

    try {
      window.matchMedia('(prefers-color-scheme: dark)')
    }
  }
}

```

```

    .addEventListener('change', this.preferedColorSchemeListener);
  } catch (error) {
    try {
      window.matchMedia('(prefers-color-scheme: dark)')
        .addListener('change', this.preferedColorSchemeListener);
    } catch (error) {
      this.unsupported = true;
    }
  }
},

computed: {
  ...mapState([
    'theme'
  ]),
  supportsAutoTheme() {
    return window.matchMedia('(prefers-color-scheme: dark)').media !== 'not all'
  },
  options() {
    let values = [
      {
        value: 'light',
        text: 'Hell',
        icon: 'brightness_7'
      },
      {
        value: 'dark',
        text: 'Dunkel',
        icon: 'bedtime'
      }
    ];

    if (this.supportsAutoTheme) {
      values.push({
        value: 'auto',
        text: 'Automatisch',
        icon: 'brightness_auto'
      })
    }

    return values;
  }
},

methods: {
  preferedColorSchemeListener(e) {
    console.log("Changing theme", this.theme, e.matches, e.matches);
    if (this.theme !== 'auto') return;

    this.$vuetify.theme.dark = e.matches;
  }
},

watch: {

```

```

    theme: {
      handler() {
        if (this.theme === 'auto') {
          this.$vuetify.theme.dark = window.matchMedia && window.matchMedia('(prefers-color-
scheme: dark)').matches;
        } else {
          this.$vuetify.theme.dark = this.theme === 'dark';
        }
      },
      immediate: true
    }
  }
}

```

Datei: updateSwMixin.js

```

export default {
  data() {
    return {
      refreshing: false,
      registration: null,
      updateExists: false,
    }
  },
  created() {
    document.addEventListener('swUpdated', this.updateAvailable, { once: true });

    navigator.serviceWorker.addEventListener('controllerchange', () => {
      // We'll also need to add 'refreshing' to our data originally set to false.
      if (this.refreshing) return
      this.refreshing = true
      // Here the actual reload of the page occurs
      window.location.reload()
    })
  },
  methods: {
    updateAvailable(event) {
      this.registration = event.detail
      this.updateExists = true
    },
    refreshApp() {
      this.updateExists = false
      // Make sure we only send a 'skip waiting' message if the SW is waiting
      if (!this.registration || !this.registration.waiting) return
      // Send message to SW to skip the waiting and activate the new SW
      this.registration.waiting.postMessage({ type: 'SKIP_WAITING' })
    }
  }
}

```

Datei: validation.php

```
<?php
```

```
return [
```

```
/*
```

```
|-----
```

```
| Validation Language Lines
```

```
|-----
```

```
|
```

```
| The following language lines contain the default error messages used by
```

```
| the validator class. Some of these rules have multiple versions such
```

```
| as the size rules. Feel free to tweak each of these messages here.
```

```
|
```

```
*/
```

```
'accepted' => 'The :attribute must be accepted.',
```

```
'active_url' => 'The :attribute is not a valid URL.',
```

```
'after' => 'The :attribute must be a date after :date.',
```

```
'after_or_equal' => 'The :attribute must be a date after or equal to :date.',
```

```
'alpha' => 'The :attribute may only contain letters.',
```

```
'alpha_dash' => 'The :attribute may only contain letters, numbers, dashes and underscores.',
```

```
'alpha_num' => 'The :attribute may only contain letters and numbers.',
```

```
'array' => 'The :attribute must be an array.',
```

```
'before' => 'The :attribute must be a date before :date.',
```

```
'before_or_equal' => 'The :attribute must be a date before or equal to :date.',
```

```
'between' => [
```

```
    'numeric' => 'The :attribute must be between :min and :max.',
```

```
    'file' => 'The :attribute must be between :min and :max kilobytes.',
```

```
    'string' => 'The :attribute must be between :min and :max characters.',
```

```
    'array' => 'The :attribute must have between :min and :max items.',
```

```
],
```

```
'boolean' => 'The :attribute field must be true or false.',
```

```
'confirmed' => 'The :attribute confirmation does not match.',
```

```
'date' => 'The :attribute is not a valid date.',
```

```
'date_equals' => 'The :attribute must be a date equal to :date.',
```

```
'date_format' => 'The :attribute does not match the format :format.',
```

```
'different' => 'The :attribute and :other must be different.',
```

```
'digits' => 'The :attribute must be :digits digits.',
```

```
'digits_between' => 'The :attribute must be between :min and :max digits.',
```

```
'dimensions' => 'The :attribute has invalid image dimensions.',
```

```
'distinct' => 'The :attribute field has a duplicate value.',
```

```
'email' => 'The :attribute must be a valid email address.',
```

```
'ends_with' => 'The :attribute must end with one of the following: :values.',
```

```
'exists' => 'The selected :attribute is invalid.',
```

```
'file' => 'The :attribute must be a file.',
```

```
'filled' => 'The :attribute field must have a value.',
```

```
'gt' => [
```

```
    'numeric' => 'The :attribute must be greater than :value.',
```

```
    'file' => 'The :attribute must be greater than :value kilobytes.',
```

```
    'string' => 'The :attribute must be greater than :value characters.',
```

```
    'array' => 'The :attribute must have more than :value items.',
```

```
],
```

```
'gte' => [
```

```
    'numeric' => 'The :attribute must be greater than or equal :value.',
```

```

'file' => 'The :attribute must be greater than or equal :value kilobytes.',
'string' => 'The :attribute must be greater than or equal :value characters.',
'array' => 'The :attribute must have :value items or more.',
],
'image' => 'The :attribute must be an image.',
'in' => 'The selected :attribute is invalid.',
'in_array' => 'The :attribute field does not exist in :other.',
'integer' => 'The :attribute must be an integer.',
'ip' => 'The :attribute must be a valid IP address.',
'ipv4' => 'The :attribute must be a valid IPv4 address.',
'ipv6' => 'The :attribute must be a valid IPv6 address.',
'json' => 'The :attribute must be a valid JSON string.',
'lt' => [
    'numeric' => 'The :attribute must be less than :value.',
    'file' => 'The :attribute must be less than :value kilobytes.',
    'string' => 'The :attribute must be less than :value characters.',
    'array' => 'The :attribute must have less than :value items.',
],
'lte' => [
    'numeric' => 'The :attribute must be less than or equal :value.',
    'file' => 'The :attribute must be less than or equal :value kilobytes.',
    'string' => 'The :attribute must be less than or equal :value characters.',
    'array' => 'The :attribute must not have more than :value items.',
],
'max' => [
    'numeric' => 'The :attribute may not be greater than :max.',
    'file' => 'The :attribute may not be greater than :max kilobytes.',
    'string' => 'The :attribute may not be greater than :max characters.',
    'array' => 'The :attribute may not have more than :max items.',
],
'mimes' => 'The :attribute must be a file of type: :values.',
'mimetypes' => 'The :attribute must be a file of type: :values.',
'min' => [
    'numeric' => 'The :attribute must be at least :min.',
    'file' => 'The :attribute must be at least :min kilobytes.',
    'string' => 'The :attribute must be at least :min characters.',
    'array' => 'The :attribute must have at least :min items.',
],
'not_in' => 'The selected :attribute is invalid.',
'not_regex' => 'The :attribute format is invalid.',
'numeric' => 'The :attribute must be a number.',
'password' => 'The password is incorrect.',
'present' => 'The :attribute field must be present.',
'regex' => 'The :attribute format is invalid.',
'required' => 'The :attribute field is required.',
'required_if' => 'The :attribute field is required when :other is :value.',
'required_unless' => 'The :attribute field is required unless :other is in :values.',
'required_with' => 'The :attribute field is required when :values is present.',
'required_with_all' => 'The :attribute field is required when :values are present.',
'required_without' => 'The :attribute field is required when :values is not present.',
'required_without_all' => 'The :attribute field is required when none of :values are present.',
'same' => 'The :attribute and :other must match.',
'size' => [
    'numeric' => 'The :attribute must be :size.',

```

```

    'file' => 'The :attribute must be :size kilobytes.',
    'string' => 'The :attribute must be :size characters.',
    'array' => 'The :attribute must contain :size items.',
],
'starts_with' => 'The :attribute must start with one of the following: :values.',
'string' => 'The :attribute must be a string.',
'timezone' => 'The :attribute must be a valid zone.',
'unique' => 'The :attribute has already been taken.',
'uploaded' => 'The :attribute failed to upload.',
'url' => 'The :attribute format is invalid.',
'uuid' => 'The :attribute must be a valid UUID.',

'phone' => 'Telephone number is invalid.',
'indisposable' => 'Disposable email addresses are not allowed.',

/*
|-----
| Custom Validation Language Lines
|-----
|
| Here you may specify custom validation messages for attributes using the
| convention "attribute.rule" to name the lines. This makes it quick to
| specify a specific custom language line for a given attribute rule.
|
*/
'custom' => [
    'attribute-name' => [
        'rule-name' => 'custom-message',
    ],
],

/*
|-----
| Custom Validation Attributes
|-----
|
| The following language lines are used to swap our attribute placeholder
| with something more reader friendly such as "E-Mail Address" instead
| of "email". This simply helps us make our message more expressive.
|
*/

'attributes' => [
    'room' => 'Room',
    'visited_on' => 'Visited on',
    'telephone' => 'Telephone'
],

];

```

Date: view.php

```

<?php
return [

    /*
    |-----
    | View Storage Paths
    |-----
    |
    | Most templating systems load templates from disk. Here you may specify
    | an array of paths that should be checked for your views. Of course
    | the usual Laravel view path has already been registered for you.
    |
    */

    'paths' => [
        resource_path('views'),
    ],

    /*
    |-----
    | Compiled View Path
    |-----
    |
    | This option determines where all the compiled Blade templates will be
    | stored for your application. Typically, this is within the storage
    | directory. However, as usual, you are free to change this value.
    |
    */

    'compiled' => env(
        'VIEW_COMPILED_PATH',
        realpath(storage_path('framework/views'))
    ),

];

```

Datei: vuetify.js

```

import Vue from 'vue';
import Vuetify from 'vuetify/lib';
import minifyTheme from 'minify-css-string';
import 'material-design-icons-iconfont/dist/material-design-icons.css' // Ensure you are using css-loader

```

```
Vue.use(Vuetify);
```

```

const vuetify = new Vuetify({
  theme: {
    options: {
      minifyTheme,
      customProperties: true,
    },
  },
});

```

```

themes: {
  light: {
    // Change 3
    primary: '#0069B4',
    accent: '#fc9000',

    info: {
      base: '#007cd6',
      lighten1: '#4196f3', // yes
    },

    background: {
      base: '#f7f7f7',
      darken1: '#F3F4F3',
      darken2: '#e8e8e8', // yes
      darken3: '#cccccc', // yes
      darken4: '#b3b3b3',
      darken5: '#707070', // yes
    },

    secondary: {
      base: '#424242',
      lighten5: '#c1c1c1', // yes
      lighten4: '#a6a6a6', // yes
      lighten3: '#8b8b8b', // yes
      lighten2: '#727272', // yes
      lighten1: '#595959'
    },

    error: '#FF5252',
    success: '#4CAF50',
    warning: '#FFC107',
  },
  dark: {
    // primary: '#0d47a1',
    primary: '#1d3042', // '#14202c' // '#131920', // '#205381'
    accent: '#c88732', // '#fc9000',
    info: '#1b6fb2', // '#2196f3',

    background: '#252525',

    secondary: {
      base: '#d8d8d8',
      lighten1: '#696969',
      lighten2: '#868686',
      lighten3: '#9a9a9a',
      lighten4: '#afafaf',
      lighten5: '#bfbfbf',
    },
    error: '#CF6679',
    // success: '#007E33',
    success: '#4CAF50',
    warning: '#FFC107',
  }
}

```



```
    }
  },
  icons: {
    iconfont: 'md',
  },
});

export default vuetify;
```

Datei: web.php

```
<?php
```

```
use Illuminate\Support\Facades\Route;
```

```
/*
```

```
|-----
```

```
| Web Routes
```

```
|-----
```

```
|
```

```
| Here is where you can register web routes for your application. These  
| routes are loaded by the RouteServiceProvider within a group which  
| contains the "web" middleware group. Now create something great!
```

```
|
```

```
*/
```

```
Route::put('user/visit', 'VisitController@store');
```

```
Route::post('account', 'AccountController@init');
```

```
Route::get('search', 'SearchController@show')->withoutMiddleware(['csrf', 'auth'])-  
>middleware('auth.basic.search.api');
```

Datei: webpack.mix.js

```
const mix = require('laravel-mix');
```

```
/*
```

```
|-----
```

```
| Mix Asset Management
```

```
|-----
```

```
|
```

```
| Mix provides a clean, fluent API for defining some Webpack build steps  
| for your Laravel application. By default, we are compiling the Sass  
| file for the application as well as bundling up all the JS files.
```

```
|
```

```
*/
```

```
mix.js('resources/js/app.js', 'public/js')
```

```
    .sass('resources/sass/app.scss', 'public/css');
```

Datei: welcome.blade.php

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Laravel</title>

    <!-- Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">

    <!-- Styles -->
    <style>
      html, body {
        background-color: #fff;
        color: #636b6f;
        font-family: 'Nunito', sans-serif;
        font-weight: 200;
        height: 100vh;
        margin: 0;
      }

      .full-height {
        height: 100vh;
      }

      .flex-center {
        align-items: center;
        display: flex;
        justify-content: center;
      }

      .position-ref {
        position: relative;
      }

      .top-right {
        position: absolute;
        right: 10px;
        top: 18px;
      }

      .content {
        text-align: center;
      }

      .title {
        font-size: 84px;
      }

      .links > a {
        color: #636b6f;
      }
    </style>
  </head>
  <body>
    <div class="flex-center text-align-center">
      <h1>Laravel</h1>
    </div>
  </body>
</html>
```

```

padding: 0 25px;
font-size: 13px;
font-weight: 600;
letter-spacing: .1rem;
text-decoration: none;
text-transform: uppercase;
}

.m-b-md {
margin-bottom: 30px;
}
</style>
</head>
<body>
<div class="flex-center position-ref full-height">
  @if (Route::has('login'))
    <div class="top-right links">
      @auth
        <a href="{{ url('/home') }}">Home</a>
      @else
        <a href="{{ route('login') }}">Login</a>

        @if (Route::has('register'))
          <a href="{{ route('register') }}">Register</a>
        @endif
      @endauth
    </div>
  @endif

  <div class="content">
    <div class="title m-b-md">
      Laravel
    </div>

    <div class="links">
      <a href="https://laravel.com/docs">Docs</a>
      <a href="https://laracasts.com">Laracasts</a>
      <a href="https://laravel-news.com">News</a>
      <a href="https://blog.laravel.com">Blog</a>
      <a href="https://nova.laravel.com">Nova</a>
      <a href="https://forge.laravel.com">Forge</a>
      <a href="https://vapor.laravel.com">Vapor</a>
      <a href="https://github.com/laravel/laravel">GitHub</a>
    </div>
  </div>
</div>
</body>
</html>

```