

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Лабораторная работа

Выполнил:  
Олейникова Полина

Группа  
К33402

Проверил:  
Добряков Д. И.

Санкт-Петербург

2023 г.

## Задача

Разработка одностраничного веб-приложения (SPA) с использованием фреймворка Vue.JS

Мигрировать написанный сайт на фреймворк Vue.JS.

Минимальные требования:

- Должен быть подключён роутер
- Должна быть реализована работа с внешним API
- Разумное деление на компоненты

## Ход работы

Ссылка на проект <https://github.com/OleinikovaPolina/ProjectControl>

Подключен роутер:

```
import Vue from 'vue'
import VueRouter, { RouteConfig } from 'vue-router'
import { userModule } from '@/store'

Vue.use(VueRouter)

const routes: Array<RouteConfig> = [
  {
    path: '/register',
    name: 'register',
    meta: { guest: true },
    component: () => import('../views/RegisterView.vue'),
  },
  {
    path: '/',
    name: 'login',
    meta: { guest: true },
    component: () => import('../views/LoginView.vue'),
  },
  {
    path: '/profile/:id',
    name: 'profile',
    meta: { requiresAuth: true },
    component: () => import('../views/ProfileView.vue'),
  },
]
```

Обработка 404 страницы:

```

    {
      path: '/404',
      name: '404',
      component: () => import('../views/NotFoundView.vue'),
    },
    {
      path: '*',
      redirect: '/404',
    },
  ],
}

const router = new VueRouter({
  mode: 'hash',
  routes
})

```

## Разделение страниц для авторизованных пользователей и нет

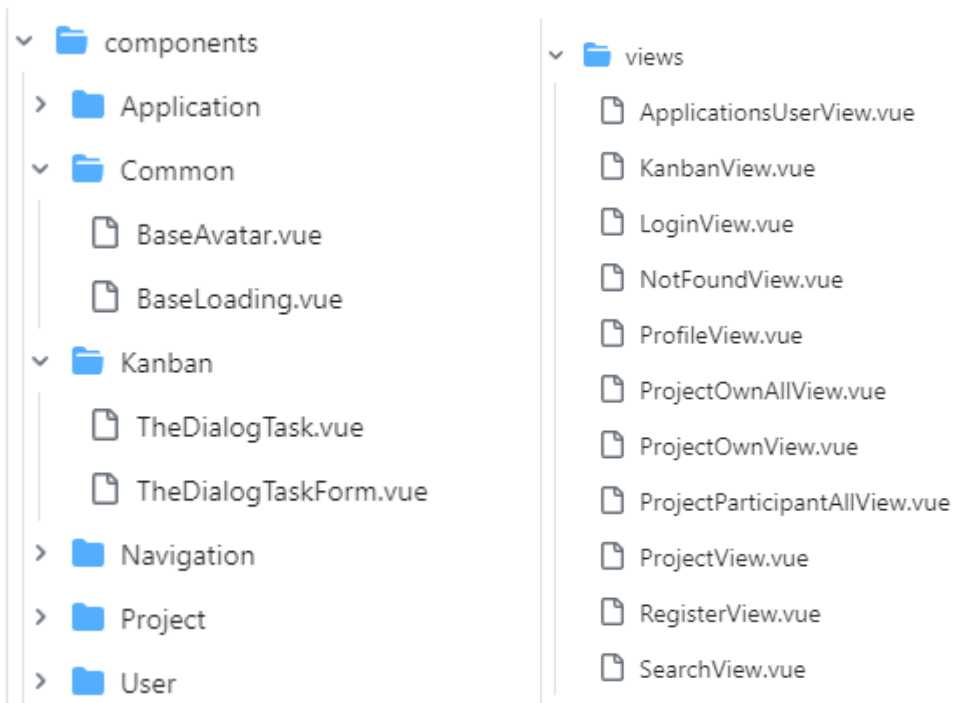
```

// запрет перехода на страницы пользователей для неавторизованных пользователей
router.beforeResolve(async (to, from, next) => {
  if (to.matched.some((record) => record.meta.requiresAuth)) {
    if(!localStorage.getItem('user')?.toString() && !userModule.state.isAuthenticated){
      await userModule.actions.userCheck({ id: localStorage.getItem('user')?.toString() || '' })
    }
    if (!userModule.state.isAuthenticated) {
      next('/')
      return
    }
    next()
  } else {
    next()
  }
})

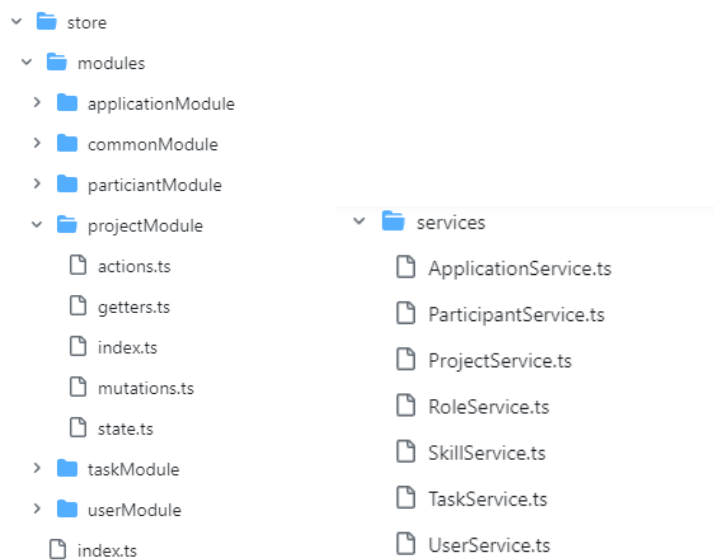
// запрет перехода на страницы входа и регистрации для авторизованных пользователей
router.beforeResolve(async (to, from, next) => {
  if (to.matched.some((record) => record.meta.guest)) {
    if(!localStorage.getItem('user')?.toString() && !userModule.state.isAuthenticated){
      await userModule.actions.userCheck({ id: localStorage.getItem('user')?.toString() || '' })
    }
    if (userModule.state.isAuthenticated) {
      next('/search')
      return
    }
    next()
  } else {
    next()
  }
})

```

Созданы страницы и есть разумное деление на компоненты, названные согласно style-guide



Работа с API: запросы описаны в предыдущей работе. Структура файлов выглядит так:



Внешний вид некоторых страниц:

Регистрация:

1

Introduce yourself

[Already have an account? Sign in](#)

[BACK](#)
[CLEAR](#)
[NEXT](#)

2

Tell about your skills

Biography

Normal

B

I

U

G

H<sub>1</sub>

H<sub>2</sub>

'''

'''

≡

≡

x<sub>2</sub>

x<sup>2</sup>

A

🖼️

🔗

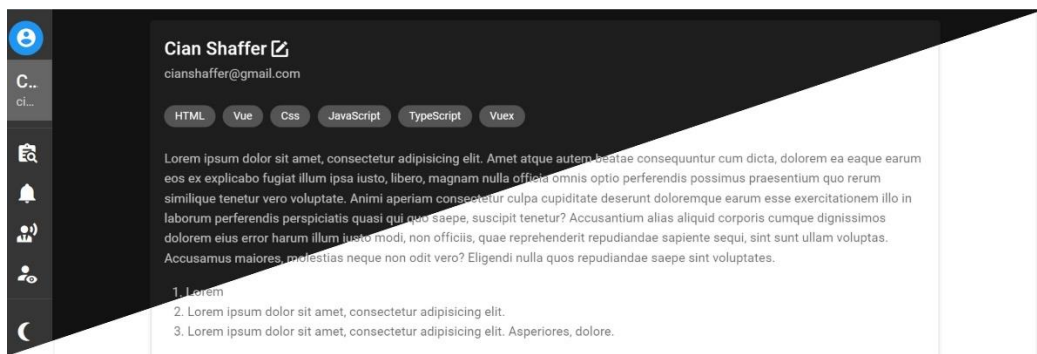
[BACK](#)
[CLEAR](#)
[NEXT](#)

3

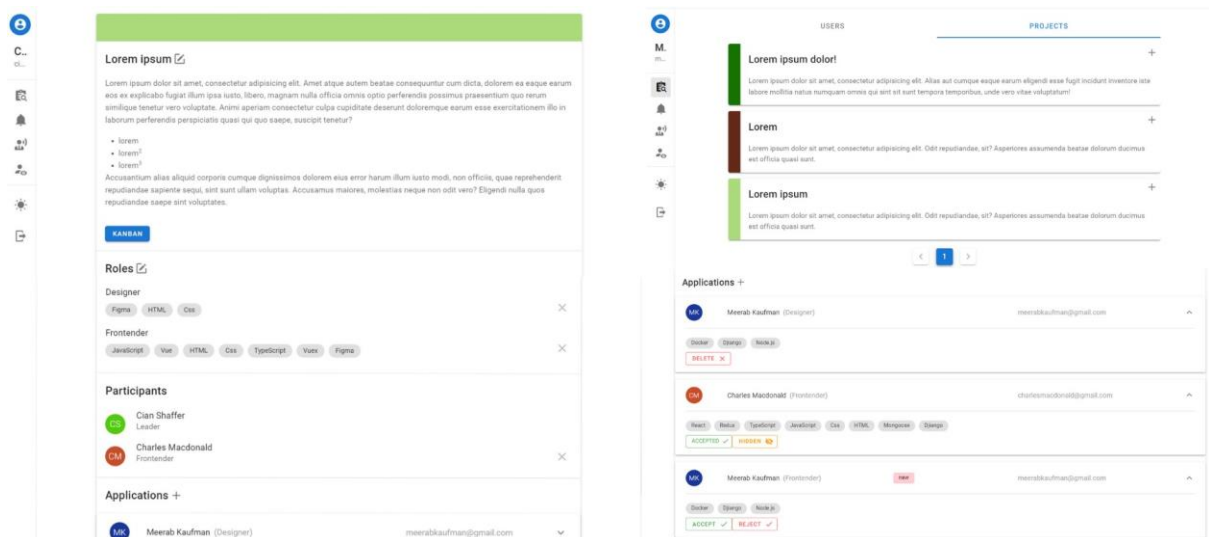
Create a password

[BACK](#)
[CLEAR](#)
[REGISTER](#)

## Темная\светлая тема на странице пользователя



## Внешний вид проектов с пагинацией; проекта с ролями, заявками, участниками



## Пример валидации при создании ролей в проекте:

Change roles +

Title  
Designer

Skills  
Figma HTML CSS

Title  
Frontender

Skills  
JavaScript Vue HTML CSS TypeScript Vuex Figma

Title  
This field is required

Skills  
This field is required

REVOKE CLOSE CHANGE

## Вывод

В ходе лабораторной работы была реализована клиентская часть приложения средствами Vue.js, Vuetify, Vuex, Vue-router и др. В процессе работы были изучены возможности и функционал данного фреймворка и библиотек.