

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 3

Выполнил:

Власов Владислав

Группа
К3439

Проверил:

Добряков Д. И.

Санкт-Петербург

2026 г.

Задача

Мигрировать ранее разработанное приложение (в рамках ЛР1 и ЛР2) на фреймворк Vue.JS.

- Должен быть подключён роутер
- Должна быть реализована работа с внешним API (желательно посредством axios)
- Разумное деление на компоненты (продемонстрируйте понимание компонентного подхода)
- Использование composable для выделения повторяющегося функционала в отдельные файлы

Ход работы

Для лабораторной работы было настроено сущности взаимодействия с внешним API с помощью axios.

```
import axios from 'axios'

const authURL = 'http://localhost:3001/users/api/auth'
const usersURL = 'http://localhost:3001/users/api/users'
const skillsURL = 'http://localhost:3003/characters/api/skills'

const authInstance = axios.create({
  baseURL: authURL
})
const userInstance = axios.create({
  baseURL: usersURL
})
const skillsInstance = axios.create({
  baseURL: skillsURL
})

export {
  authInstance,
  userInstance,
  skillsInstance
}
```

```
import { authInstance, userInstance, skillsInstance } from "@/api/instances"
import AuthApi from "@/api/auth"
```

```
import UsersApi from "@api/users"
import SkillsApi from "@api/skills"

const authApi = new AuthApi(authInstance)
const usersApi = new UsersApi(userInstance)
const skillsApi = new SkillsApi(skillsInstance)

export {
  authApi,
  usersApi,
  skillsApi
}
```

Также была настроена маршрутизация по сайту и элементы сайта разбиты на компоненты. Всего на сайте было реализовано 4 страницы и 9 компонентов.

А именно, страница персонажей, на которой расположена заглушка, реализованная сразу на странице, а не как компонент (хотя и могла бы быть им).

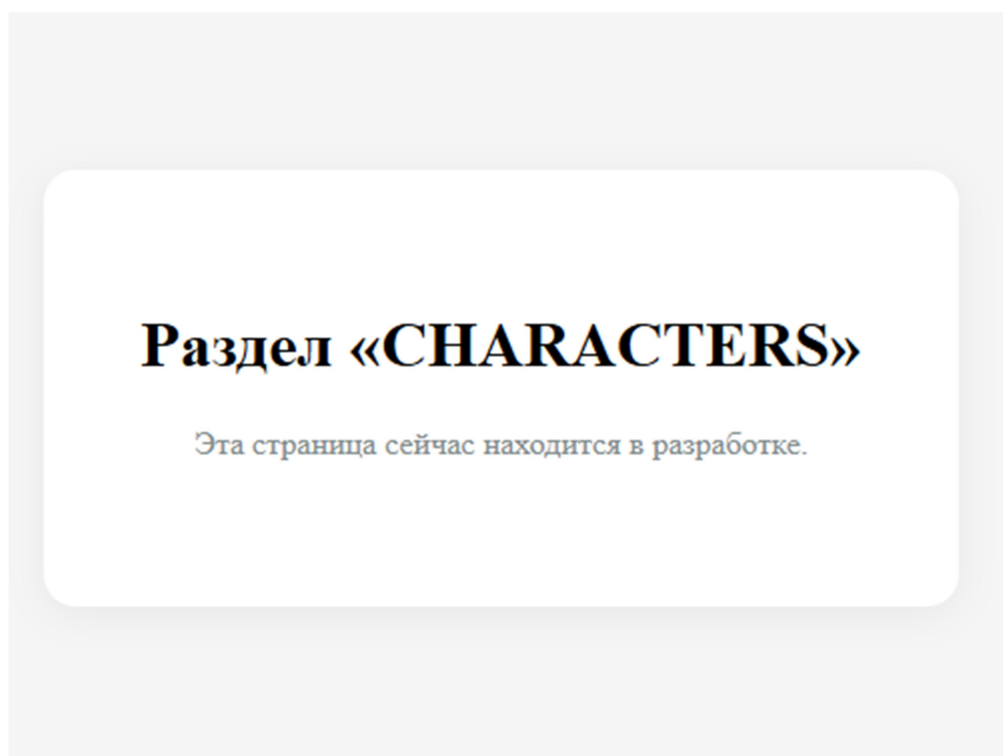


Рисунок 1 – страница персонажей.

На странице навыков (см. рисунок 2) расположено 3 компонента. Кнопка с названием навыка, карточка навыка и карточка с формой создания навыка.

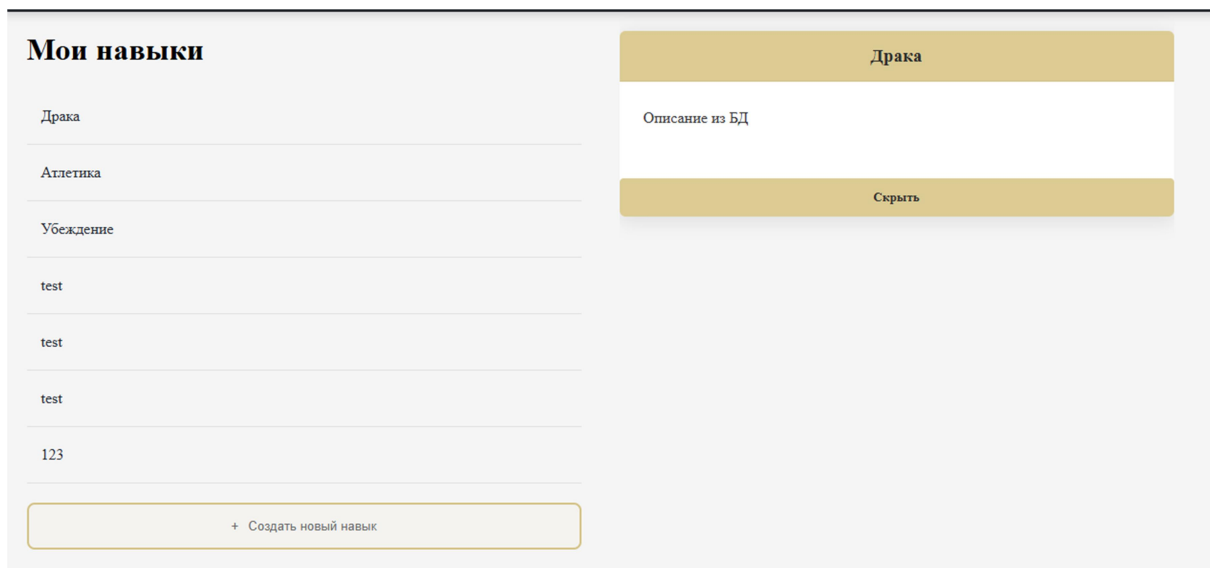


Рисунок 2 – страница навыка.

The screenshot shows a 'Новый навык' (New Skill) form. It has a gold header with the title 'Новый навык'. The form contains three input fields: 'Название' (Name) with placeholder text 'Наименование', 'Атрибут' (Attribute) with a dropdown menu showing 'Выберите атрибут', and 'Описание' (Description) with placeholder text 'Зона ответственности навыка'. At the bottom are two buttons: 'Отмена' (Cancel) and 'Создать' (Create).

Рисунок 3 – карточка создания нового навыка.

На странице профиля также выделено 3 компонента. Карточка пользователя, доп. Информация и форма редактирования профиля.

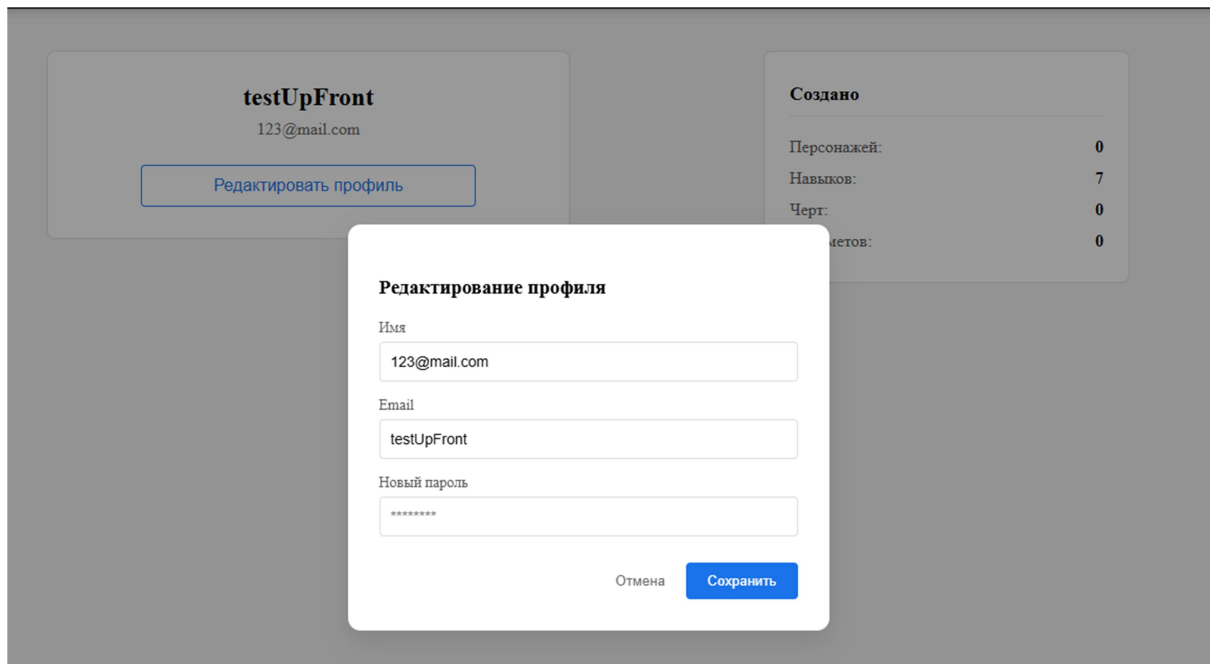


Рисунок 4 – страница профиля.

Наконец, на странице авторизации расположены ещё 2 компонента, которые сменяют друг друга – окна входа и регистрации.

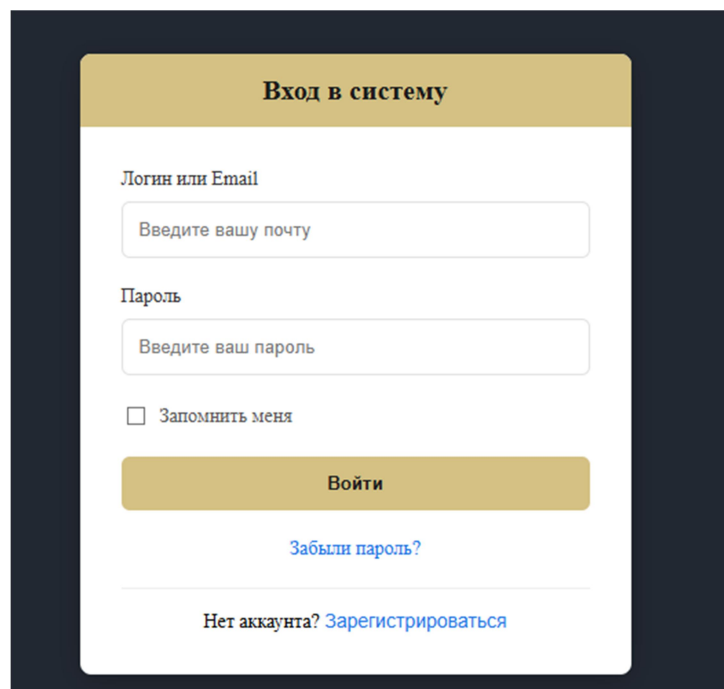


Рисунок 5 – окно входа.

Регистрация

Логин

Email

Пароль

Пароль должен содержать не менее 8 символов

Повторите пароль

Зарегистрироваться

Уже есть аккаунт? [Авторизоваться](#)

Рисунок 6 – окно регистрации.

Последний из 9-ти компонентов – кнопка в окне навигации, которое является частью Layout. Эта кнопка имеет 3 состояния, регулируемых классами CSS.

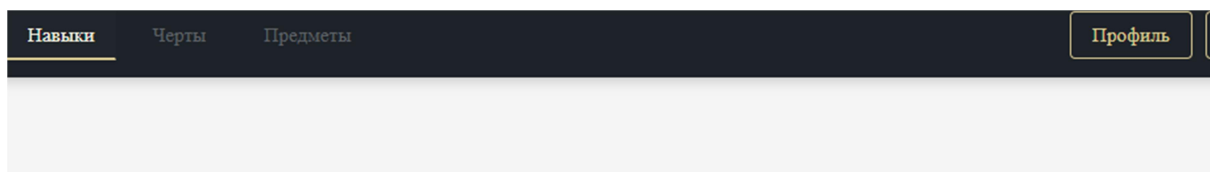


Рисунок 7 – кнопки навигации.

```
const buttonClasses = {
  'nav-btn': true,
  'nav-btn--disabled': props.isDisabled,
  'nav-btn--action': props.isAction
};
</script>

<template>
  <component
    :is="isDisabled ? 'span' : 'RouterLink'"
    :to="to"
  >
```

```

      :class="buttonClasses"
    >
      <span class="nav-btn__label">{{ label }}</span>
    </component>
  </template>

```

Также в проекте использованы composables. А именно для отчистки формы.

```

const useFormReset = () => {

  const resetFields = (formData) => {

    for (const property in formData) {
      formData[property] = '';
    }
    console.log('Форма очищена:');
  };

  return { resetFields };
};

export default useFormReset;

```

Для переключения по страницам реализован роутер

```

import { createRouter, createWebHistory } from 'vue-router'

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes: [
    {
      path: '/characters',
      name: 'characters',
      component: () => import('../views/CharacterPage.vue')
    },
    {
      path: '/skills',
      name: 'skills',
      component: () => import('../views/SkillsPage.vue')
    },
    {
      path: '/profile',
      name: 'profile',
      component: () => import('../views/ProfilePage.vue')
    },
    {
      path: '/auth',
      name: 'auth',
      component: () => import('../views/AuthView.vue')
    }
  ]
})

```

```
    },  
  ]  
})  
  
export default router
```

API с сайтом связано через pinia хранилище, настроенное на сохранение данных в localstore.

А для того, чтобы постоянно не загружать информацию о навыках, реализована проверка на наличие данных и их время жизни

```
state: () => ({  
  skills: [],  
  lastUpdated: null,  
  ttl: 15 * 60 * 1000  
}),  
  
actions: {  
  
  async getSkills(token) {  
  
    const now = Date.now()  
  
    if (this.skills.length > 0 && this.lastUpdated && (now -  
this.lastUpdated < this.ttl)) {  
      console.log('Навыки уже есть в сторе, пропускаем загрузку');  
      return;  
    }  
  }  
}
```

Вывод

Сайт, написанный на чистом html + css + js был переписан на Vue и паттерн одностраничного приложения. Это гораздо удобнее с точки зрения организации кода.