

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронтенд разработка

Отчет

ЛР 3 - Разработка одностраничного веб-приложения
(SPA) с использованием фреймворка Vue.JS

Выполнил:
Сергеев Виктор
К3441

Проверил:
Добряков Д. И.

Санкт-Петербург

2026 г.

Задача

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

Каким требованиям ваше приложение должно удовлетворять:

- Должен быть подключен роутер;
- Должна быть реализована работа с внешним API;
- Разумное деление на компоненты;
- Использование `composable` для выделения повторяющегося функционала в отдельные файлы

Ход работы

Ранее реализованное фронтенд приложение содержит следующие страницы, которые требуется также перенести в новое приложение на фреймворк Vue.JS:

- Страница регистрации;
- Страница логина;
- Страница профиля пользователя;
- Главная страница с рецептами;
- Страница рецепта;
- Страница создания и изменения рецепта;
- Страница поиска по фильтрам.

Эти страницы были реализованы в соответствующих View. На рисунках 1-3 показаны примеры того, как выглядят страницы после отрисовки движком Vue.

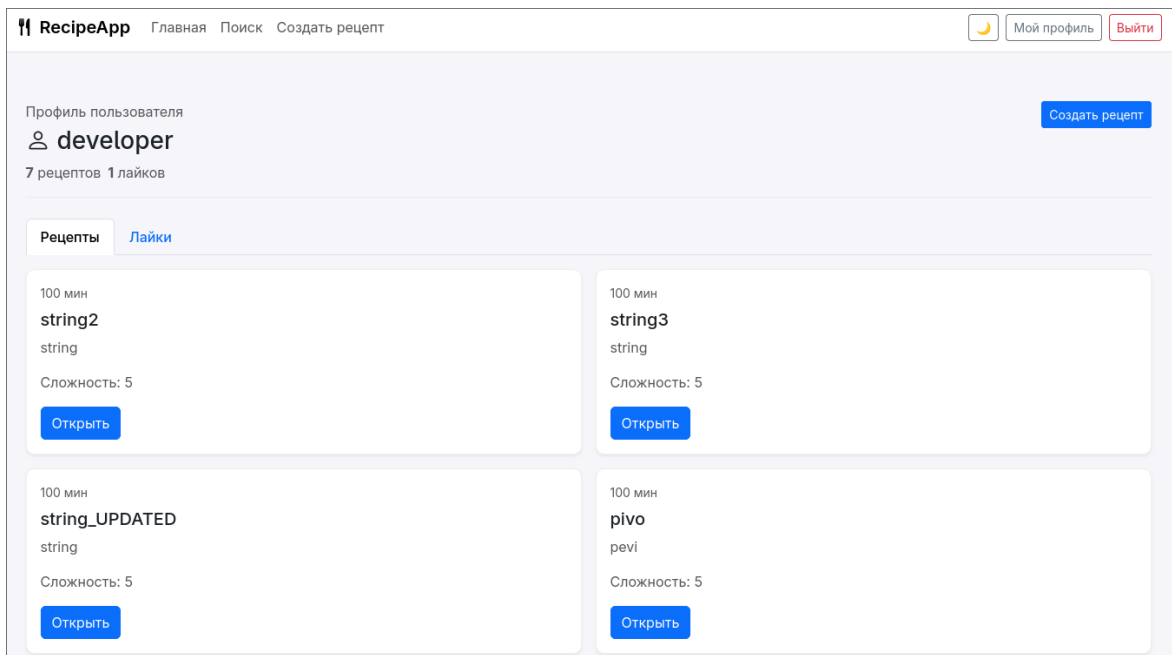


Рисунок 1 - Страница профиля

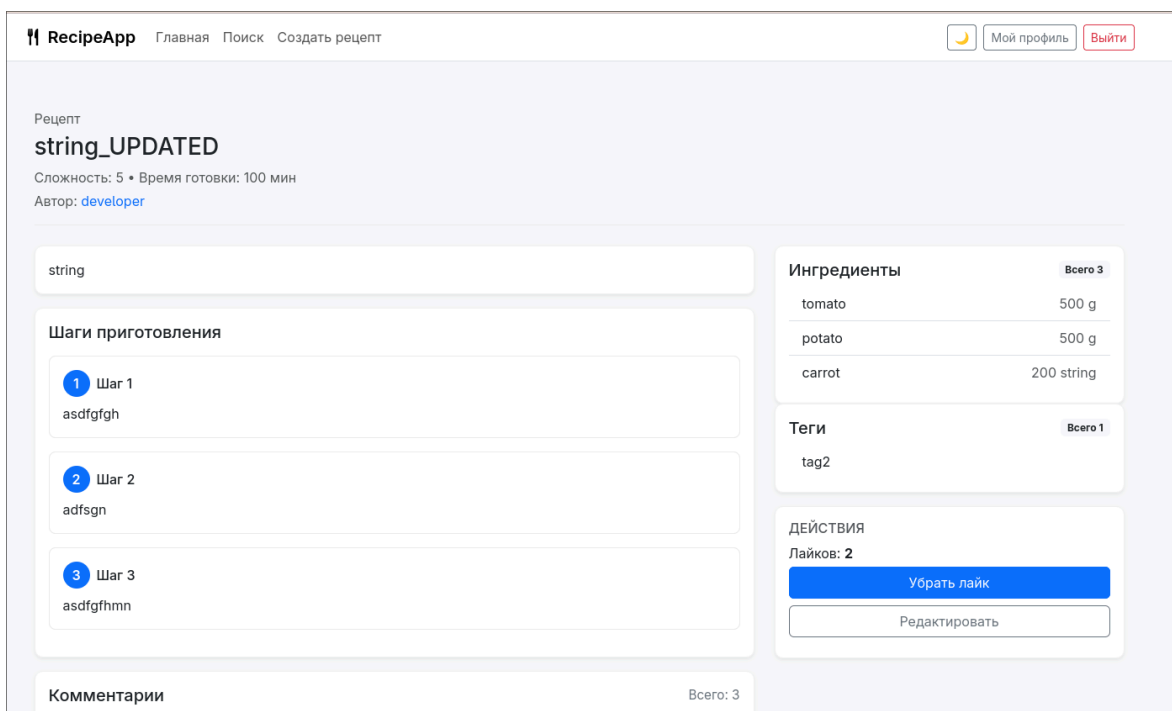


Рисунок 2 - Страница рецепта

RecipeApp Главная Поиск Создать рецепт

Мой профиль Выйти

Создать рецепт

Заполните форму.

Основная информация

Название*

Описание

Сложность*

Готовка (мин)*

Медиа

Ссылка на фото*

Ссылка на видео

Ингредиенты

Ингредиент*	Количество*	Единица измерения*
Выберите	0	

+ Добавить

Рисунок 3 - Страница создания рецепта

При работе приложения есть постоянные куски кода, которые требуется вынести в отдельные сущности: html-вёрстку в Components, js-скрипты в Composables. На рисунке 4 представлен набор Components и Composables, используемых в приложении.

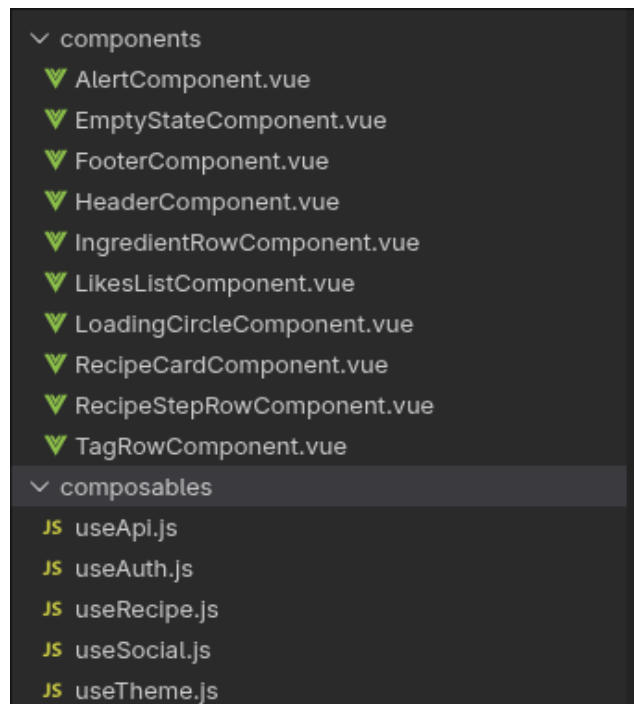


Рисунок 4 - Используемые Components и Composables

Эти сущности используются напрямую во Views. Практически вся вёрстка была перенесена с предыдущего проекта и добавлен конструкции фреймворка Vue.JS: для итерации по массиву объектов и условные операторы для создания динамики сайта. Также были перенесены скрипты, стили, SVG-иконки и доступность.

Для маршрутизации по страницам используется роутер из модуля vue-router. В index.js файле адреса URI привязываются к соответствующим Views, после чего создаётся объект router.

Файл App.vue является основным компонентом приложения. Здесь инициализируются шапка, роутер и подвал сайта, инициализируются функции из composables и стили. На рисунке 5 представлен код файла.

```

1  <template>
2    <div id="app">
3      <HeaderComponent />
4      <router-view />
5      <FooterComponent />
6    </div>
7  </template>
8
9  <script setup>
10   import { RouterView } from 'vue-router'
11   import { onMounted } from 'vue';
12   import { useAuth } from '@/composables/useAuth';
13   import { useTheme } from '@/composables/useTheme';
14   import HeaderComponent from '@/components/HeaderComponent.vue';
15   import FooterComponent from '@/components/FooterComponent.vue';
16
17   const { init } = useAuth();
18   const { theme } = useTheme();
19   onMounted(async () => {
20     await init();
21   });
22 </script>
23
24 <style scoped>
25 #app {
26   min-height: 100vh;
27   display: flex;
28   flex-direction: column;
29 }
30 main {
31   flex: 1;
32 }
33 </style>
34

```

Рисунок 5 - App.vue

Наконец, точкой входа является скрипт main.js. Здесь создаётся App и в него добавляется роутер. На рисунке 6 представлена точка входа.

```

1  import { createApp } from 'vue';
2  import App from './App.vue';
3  import router from './router';
4  import './assets/style.css';
5
6  const app = createApp(App);
7  app.use(router);
8  app.mount('#app');

```

Рисунок 6 - main.js

Вывод

Ранее разработанное приложение в рамках предыдущих лабораторных работ было мигрированного на фреймворк Vue.JS.