

**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.

The screenshot shows the RecipeApp user profile interface. At the top, there's a navigation bar with the app logo, 'RecipeApp', and links for 'Главная', 'Поиск', 'Создать рецепт'. On the right are icons for 'Мой профиль' (My Profile) and 'Выход' (Logout). Below the navigation is a section titled 'Профиль пользователя' (User Profile) with the name 'developer'. It displays '7 рецептов' (7 recipes) and '1 лайков' (1 like). A 'Создать рецепт' (Create Recipe) button is in the top right. Below this, there are two tabs: 'Рецепты' (Recipes) and 'Лайки' (Likes), with 'Likes' being the active tab. The main content area shows four recipe cards in a grid:

- string2**
string
Сложность: 5
[Открыть](#)
- string3**
string
Сложность: 5
[Открыть](#)
- string_UPDATED**
string
Сложность: 5
[Открыть](#)
- pivo**
pivo
Сложность: 5
[Открыть](#)

Рисунок 1 - Страница профиля

The screenshot shows the RecipeApp recipe detail page for a recipe named 'string_UPDATED'. At the top, it has the 'RecipeApp' logo, 'Главная', 'Поиск', 'Создать рецепт', and profile/logout icons. The recipe title is 'string_UPDATED'. It shows a complexity of '5' and a cooking time of '100 мин'. The author is 'developer'. The left side of the page contains the recipe steps:

- Шаг 1
asdfgfhg
- Шаг 2
adfsqgn
- Шаг 3
asdfghmn

The right side contains sections for ingredients, tags, and actions:

- Ингредиенты** (Ingredients)
tomato 500 g
potato 500 g
carrot 200 string
- Теги** (Tags)
tag2
- ДЕЙСТВИЯ** (Actions)
Лайков: 2
[Убрать лайк](#) (Remove like)
[Редактировать](#) (Edit)

At the bottom, there's a 'Комментарии' (Comments) section with a count of 'Всего: 3'.

Рисунок 2 - Страница рецепта

The screenshot shows the 'Create Recipe' page of the RecipeApp. At the top, there's a navigation bar with the app logo, 'RecipeApp', and links for 'Главная', 'Поиск', and 'Создать рецепт'. On the right are 'Мой профиль' and 'Выход' buttons. The main form has a title 'Создать рецепт' and a subtitle 'Заполните форму.' It's divided into sections: 'Основная информация' (Name: ' ', Complexity: '1', Description: ' ', Cooking time: '0'), 'Медиа' (Photo link: 'https://...', Video link: 'https://...'), and 'Ингредиенты' (Ingredient dropdown: 'Выберите', Quantity: '0', Unit: ' '). A blue 'Добавить' button is located at the top right of the ingredients section.

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.

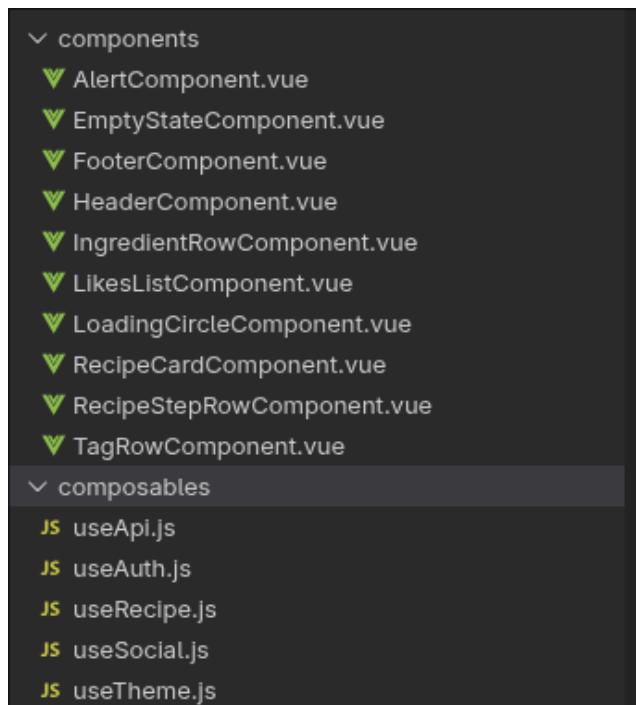


Рисунок 4 - Используемые Components и Composables

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.

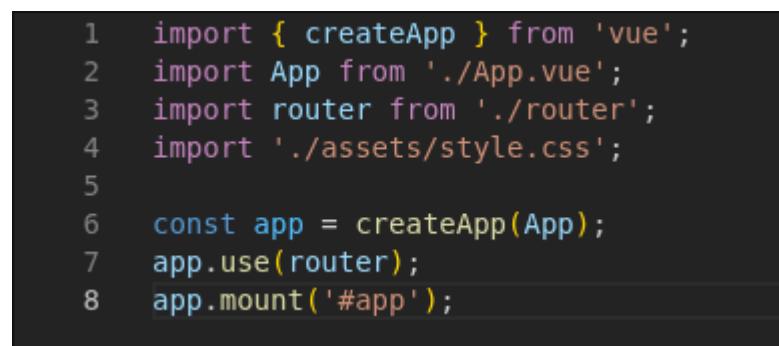
Файл App.vue является основным компонентом приложения. Здесь инициализируются шапка, роутер и подвал сайта, инициализируются функции из composables и стили. На рисунке 5 представлен код файла.



```
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.