САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 3

Выполнил:

Григорян Самвел

Группа K3440

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Мигрировать ранее написанный сайт на фреймворк Vue.JS. Минимальные требования:

- Должен быть подключен роутер
- Должна быть реализована работа с внешним API
- Разумное деление на компоненты
- Использование composable

Ход работы

В рамках работы был выполнен перенос ранее созданного сайта на фреймворк Vue.JS. Сначала подготовлено окружение (npm, Vite, плагин Vue), затем создана папка src и базовые точки входа (main.js, App.vue).

Структура проекта после миграции стала более организованной. В корне находятся package.json, vue.config.js, vite.config.js и index.html, который используется как точка монтирования всего SPA-приложения. Внутри папки src лежат main.js и App.vue, а также несколько ключевых директорий.

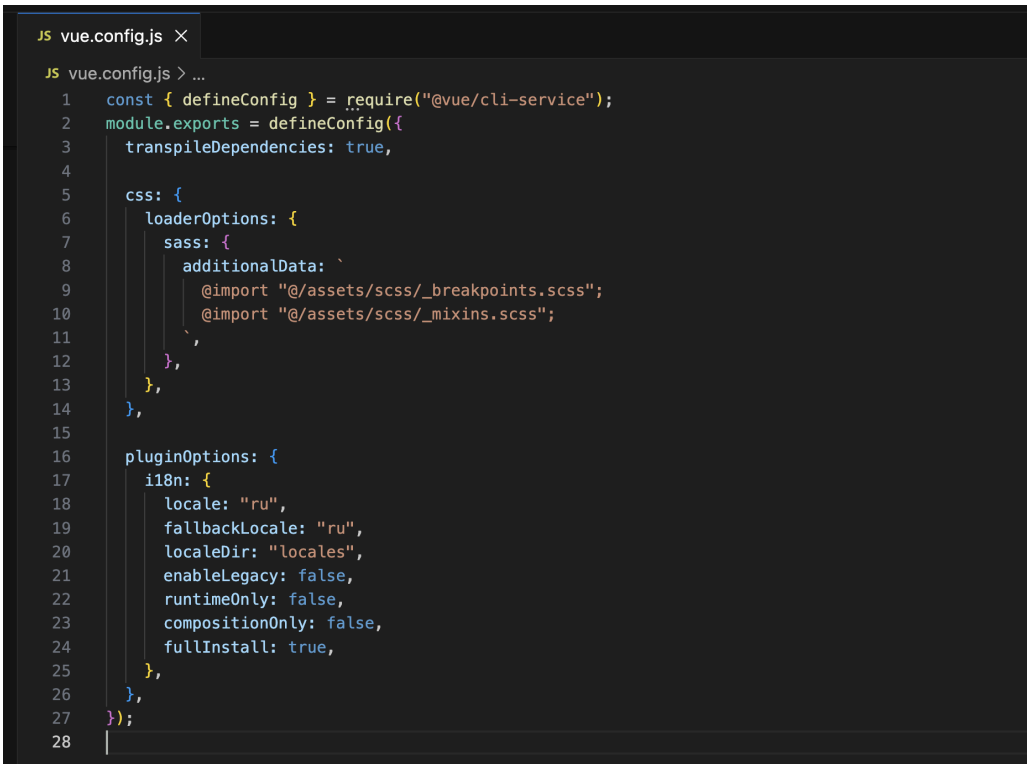

Рисунок 1 - Файл vue.config.js

Папка components содержит переиспользуемые элементы интерфейса: кнопки, инпуты, навигацию, карточки и прочее. Папка views хранит страницы, которые отображаются через роутер. Папка layouts используется для компоновок страниц и частей страниц. Папка data предназначена для различных методов работы с данными. Там находятся функции для работы с API, с localstorage и конфигурация глобального состояния Vuex. Папка router содержит конфигурацию маршрутов и проверку доступа. Папка services содержит вспомогательные функции. В папке assets находятся стили, в том числе переменные тем, а также статические ресурсы.

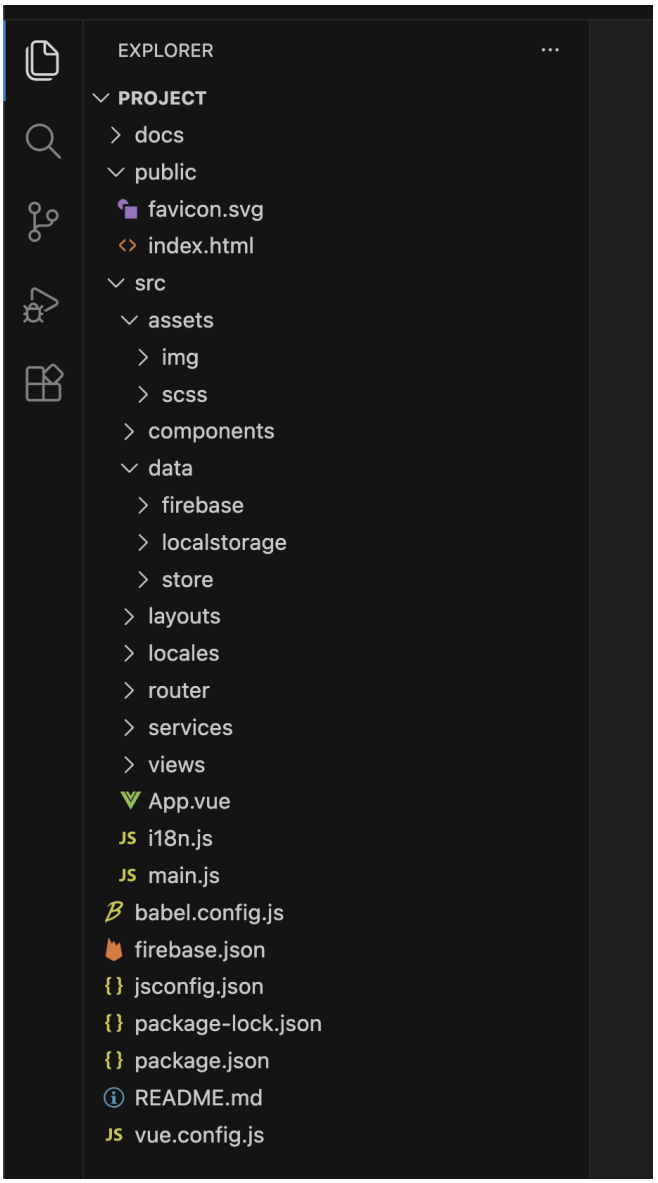

Рисунок 2 - Структура проекта

Работа страниц организована следующим образом. Например, при переходе пользователя на адрес наподобие /users/edit/:id роутер передает значение параметра id в компонент UserEditView. Компонент делает запрос к API и получает данные пользователя. Затем эти данные отображаются на странице.

Работа фильтрации происходит на странице общего списка. Пользователь указывает фильтры, форма связана с реактивными переменными. После отправки формы создаются параметры запроса, затем выполняется обращение к API. Сервер возвращает отфильтрованный список рецептов, который отображается на странице.

Роутер настроен так, что каждая страница подгружается по требованию. В нем также есть проверки авторизации. Если маршрут требует авторизации, а пользователь не вошел в систему, он будет перенаправлен на страницу входа. Компоненты отвечают за внешний вид и взаимодействие с пользователем, а повторяющиеся операции вынесены в отдельные функции. Такой подход делает код чище и удобнее.

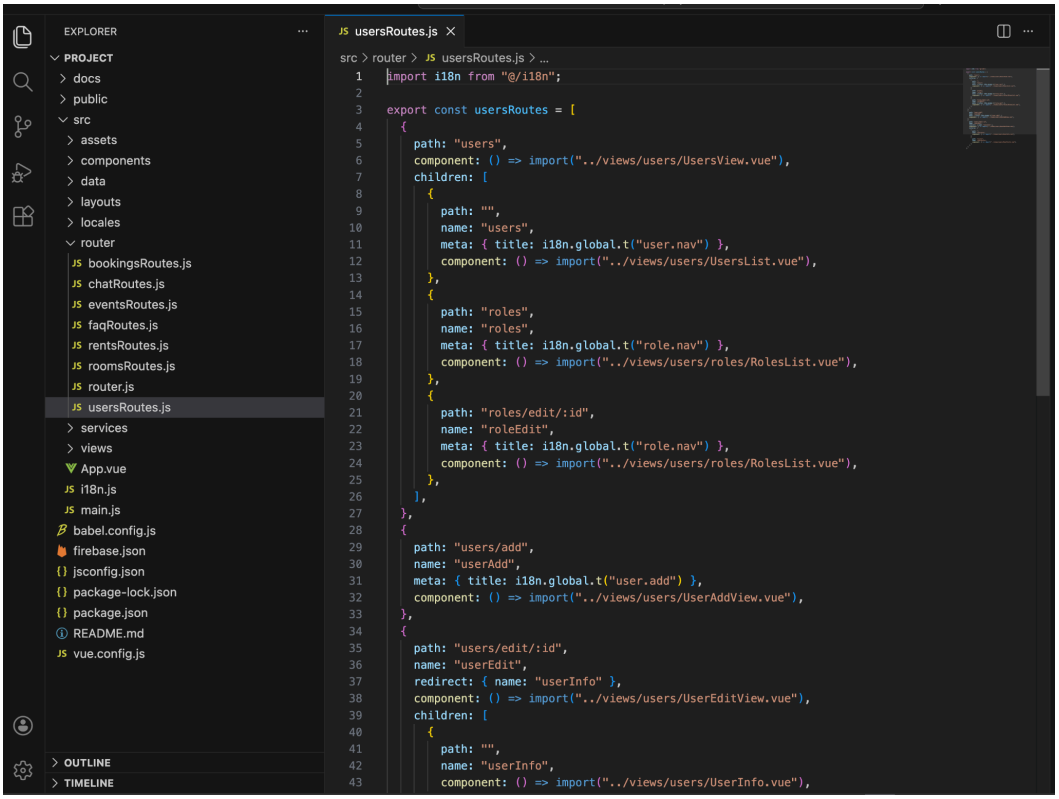

Рисунок 3 - Файл роутинга пользователей

Вывод

В результате работы было выполнено перенесение проекта на Vue, подключение и настройка Vue Router, реализация централизованной работы с внешним API, разделение интерфейса на небольшие переиспользуемые компоненты. Все требования лабораторной работы выполнены, а структура приложения стала более гибкой и удобной для дальнейшего развития.