САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 3: Разработка одностраничного веб-приложения (SPA) с использованием фреймворка Vue.JS

Выполнил:

Кузнецов Артур

K3440

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Мигрировать ранее написанный сайт на фреймворк Vue.JS.

Минимальные требования:

– Должен быть подключён 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). После этого каждая страница старого сайта была превращена в view-компонент, а общие части вынесены в отдельные компоненты и composables.

Структура проекта после миграции стала более организованной. В корне находятся package.json, 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. Папка components содержит переиспользуемые элементы интерфейса: шапку, подвал, карточки рецептов, строки ингредиентов и шагов, спиннеры загрузки и элементы для отображения пустых состояний. Папка views хранит страницы, которые отображаются через роутер. Папка composables используется для логики, которую можно подключить в любом компоненте. Там находятся функции для работы с API, авторизацией, рецептами, социальными действиями и темой оформления. Папка router содержит конфигурацию маршрутов и проверку доступа. Папка config хранит константы с адресами API. Папка utils содержит вспомогательные функции. В папке assets находятся стили, в том числе переменные тем и файл со стилями темной темы, а также статические ресурсы.

Работа страниц организована следующим образом. Например, при переходе пользователя на адрес наподобие /recipe/:id роутер передает значение параметра id в компонент RecipeView. Компонент вызывает функцию loadRecipe из composable useRecipe, которая делает запрос к API и получает данные рецепта. Затем эти данные отображаются на странице. Работа фильтрации происходит на странице поиска. 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 с помощью компонентов RecipeCard.

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. Работа с API объединена в модуль useApi, где находятся функции для запросов, обработки ошибок и работы с токенами. Composables используются для разделения логики. Компоненты отвечают за внешний вид и взаимодействие с пользователем, а повторяющиеся операции вынесены в отдельные функции. Такой подход делает код чище и удобнее.

Вывод

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