

**Министерство науки и высшего образования Российской  
Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ITMO University**

**Отчет по домашней работе №5**

**По дисциплине** Фронтенд разработка

**Обучающийся** Грищук Елизавета Дмитриевна

**Факультет** факультет прикладной информатики

**Группа** К3343

**Направление подготовки** 45.03.04 Интеллектуальные системы в гуманитарной сфере

**Образовательная программа** Интеллектуальные системы в гуманитарной сфере

**Обучающийся**

Грищук Е.Д.

**Преподаватель**

Добряков Д.И.

## Цель работы

Освоить основы разработки одностраничных приложений (SPA) на Vue.js, научиться:

- инициализировать проект Vue с помощью Vite;
- настраивать роутинг с использованием vue-router;
- организовывать управление состоянием приложения через Pinia;
- выполнять HTTP-запросы к серверу с помощью Axios;
- работать с компонентами, формами и директивами Vue.

## Используемые технологии

- Vue.js 3 — фреймворк для построения пользовательского интерфейса;
- Vite — инструмент сборки проекта;
- Vue Router — маршрутизация в SPA;
- Pinia — менеджер состояния приложения;
- Axios — библиотека для HTTP-запросов;
- Bootstrap — CSS-фреймворк для стилизации интерфейса;
- json-server — эмуляция REST API.

## Ход выполнения работы

### 1. Инициализация проекта

Проект был создан с использованием команды:

```
npm init vue@latest
```

После инициализации были установлены необходимые зависимости:

```
npm install axios pinia pinia-persists vue-router bootstrap
```

Также были настроены скрипты запуска проекта и порт сервера (8080).

### 2. Настройка API

Для работы с сервером был создан общий экземпляр Axios (instance.js) с базовым URL `http://localhost:3000`. Для работы с заметками реализован класс `NotesApi`, содержащий методы получения и создания заметок.

### 3. Управление состоянием (Pinia)

Был настроен менеджер состояния Pinia. Создано хранилище notes, отвечающее за:

- загрузку списка заметок с сервера;
- создание новой заметки;
- хранение массива заметок в состоянии приложения.

### 4. Роутинг

С помощью vue-router был настроен один маршрут /, ведущий на представление NotesPage.vue, отображающее список заметок и форму их создания.

### 5. Компонентный подход

В проекте реализованы:

- компонент NoteCard.vue для отображения одной заметки;
- layout-компонент BaseLayout.vue, отвечающий за общий контейнер страницы;
- представление NotesPage.vue, объединяющее форму и список заметок.

### 6. Работа с формами и директивами Vue

В форме создания заметки использовались директивы:

- v-model — для двустороннего связывания данных формы;
- v-for — для отображения списка заметок;
- @submit.prevent — для обработки отправки формы без перезагрузки страницы.

После отправки формы данные передаются в Pinia Store, откуда отправляются на сервер.

### 7. Работа с backend

В качестве backend был использован json-server. Сервер запускался командой:

```
npx json-server --watch db.json --port 3000
```

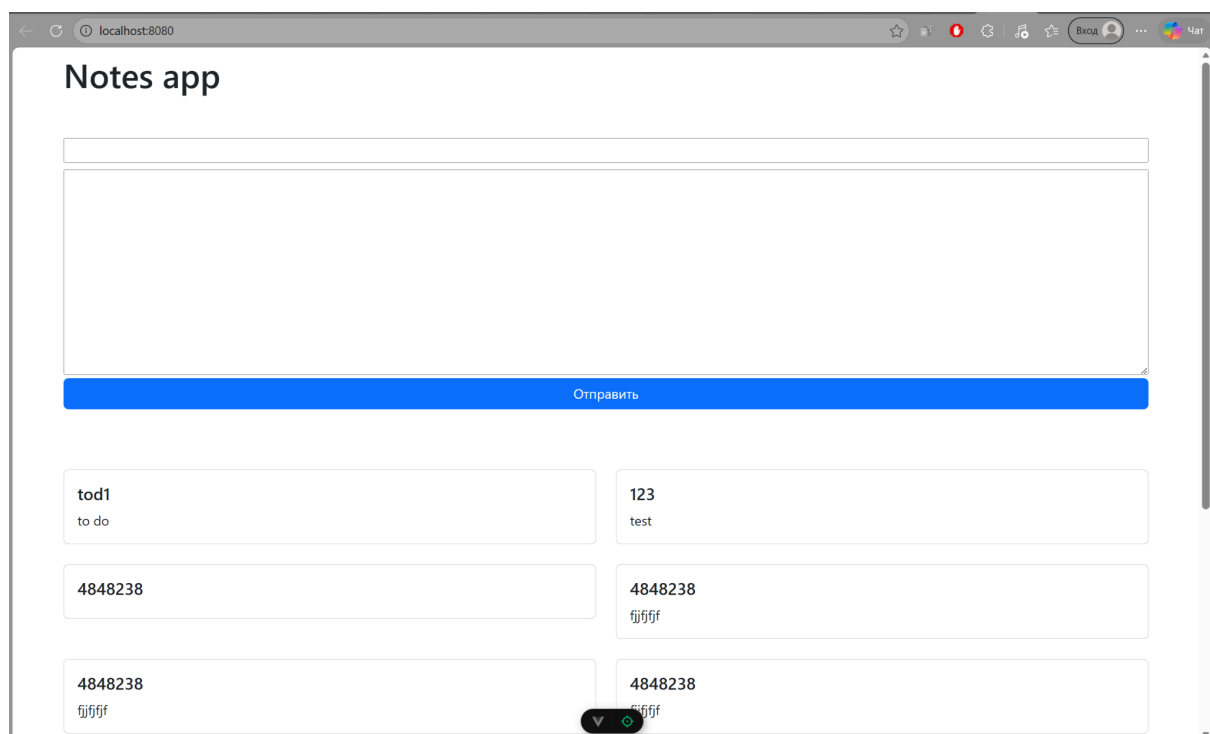
Данные о заметках хранятся в файле db.json.

## Результат работы

В результате выполнения лабораторной работы было разработано SPA-приложение «Notes App», позволяющее:

- просматривать список заметок;
- добавлять новые заметки через форму;
- получать и отправлять данные на сервер без перезагрузки страницы.

Приложение корректно взаимодействует с backend и использует современную архитектуру фронтенд-приложений.



## Вывод

В ходе лабораторной работы были получены практические навыки работы с Vue.js, Pinia и Axios. Было изучено взаимодействие клиента с REST API, организация структуры проекта и использование компонентного подхода при разработке SPA.

