

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа

Выполнил:

Захарчук Александр

Группа К3441

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2025 г.

Задача

В рамках данной работы Вам предстоит изучить основные команды пакетного менеджера NPM и научиться стартовать проект на Vue. Научиться работать с прм и vue на основе мануала.

Ход работы

Для отображения одной заметки был реализован компонент NoteCard, который показан на рисунке 1.

```
1  <template>
2    <div class="card">
3      <div class="card-body">
4        <h5 class="card-title">{{ name }}</h5>
5        <p class="card-text">
6          {{ text }}
7        </p>
8      </div>
9    </div>
10   </template>
11
12  <script>
13  export default {
14    name: "NoteCard",
15
16    props: {
17      name: {
18        type: String,
19        required: true,
20      },
21      text: {
22        type: String,
23        required: false,
24      },
25    },
26  };
27  </script>
28
```

Рисунок 1 - Компонент NoteCard

Для хранения данных из API было реализовано хранилище с помощью библиотеки pinia. Код хранилища представлен на рисунке 2.

```
1 import { defineStore } from "pinia";
2 import { notesApi } from "@/api";
3
4 const useNotesStore = defineStore("notes", {
5   state: () => ({
6     notes: [],
7   }),
8
9   actions: {
10     async loadNotes() {
11       const response = await notesApi.getAll();
12
13       this.notes = response.data;
14
15       return response;
16     },
17
18     async createNote(data) {
19       const response = await notesApi.createNote(data);
20
21       this.notes = response.data;
22
23       return response;
24     },
25   },
26 });
27
28 export default useNotesStore;
```

Рисунок 2 - Хранилище

Маршрутизация в приложении выполняется с помощью vue-router. Настройка маршрутизации показана на рисунке 3.

```
1 import { createRouter, createWebHistory } from "vue-router";
2
3 const router = createRouter({
4   history: createWebHistory(import.meta.env.BASE_URL),
5   routes: [
6     {
7       path: "/",
8       name: "notes",
9       component: () => import("../views/NotesPage.vue"),
10      },
11    ],
12  });
13
14 export default router;
```

Рисунок 3 - Настройка маршрутизации

Код страницы заметок представлен на рисунках 4-5.

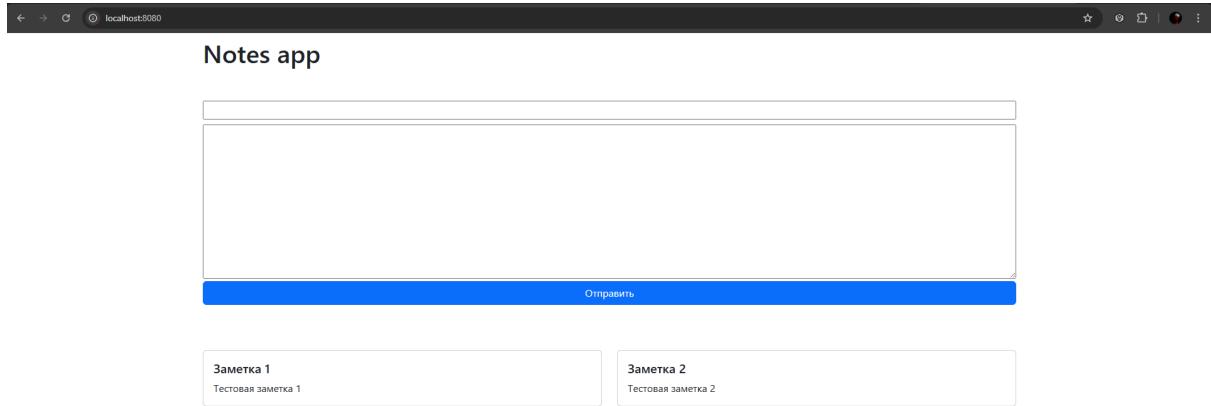
```
1 <template>
2   <base-layout>
3     <h1>Notes app</h1>
4
5   <form ref="noteForm" @submit.prevent="createCard" class="d-flex flex-column my-5">
6     <input type="text" v-model="form.name" class="my-1" />
7
8     <textarea cols="30" rows="10" v-model="form.text" class="my-1" />
9
10    <button type="submit" class="btn btn-primary">Отправить</button>
11  </form>
12
13  <div class="row row-cols-1 row-cols-md-2 g-4 mt-5" id="notes">
14    <div class="col" v-for="note in notes" :key="note.id">
15      <note-card :name="note.name" :text="note.text" />
16    </div>
17  </div>
18 </base-layout>
19 </template>
20
21 <script>
22 import BaseLayout from "@/layouts/BaseLayout.vue";
23 import NoteCard from "@/components/NoteCard.vue";
24 import { mapActions, mapState } from "pinia";
25
26 import useNotesStore from "@/stores/notes";
27
28 export default {
29   name: "NotesPage",
30 }
```

Рисунок 4 - Страница заметок

```
21 <script>
22 export default {
23   components: { BaseLayout, NoteCard },
24   data() {
25     return {
26       form: {
27         name: "",
28         text: "",
29         userId: 1,
30       },
31     };
32   },
33   computed: {
34     ...mapState(useNotesStore, ["notes"]),
35   },
36   mounted() {
37     this.loadNotes();
38   },
39   methods: {
40     ...mapActions(useNotesStore, ["loadNotes", "createNote"]),
41
42     async createCard() {
43       await this.createNote(this.form);
44       await this.loadNotes();
45
46       this.$refs.noteForm.reset();
47     },
48   },
49 }
50
51 
```

Рисунок 5 - Страница заметок

В результате получилось приложение, которое показано на рисунке 6.



Вывод

В ходе выполнения данной лабораторной работы были изучены основы работы с фреймворком Vue. Была создана страница для создания и просмотра заметок.