

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронтэнд-разработка

Отчет

Домашняя работа №5

Выполнил:

Гнеушев Владислав

К3439

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2025 г.

Задача

Изучить основные команды пакетного менеджера NPM и научиться стартовать проект на Vue. Научиться работать с npm и vue на основе мануала

Ход-работы

1. Компонента NoteCard.vue

```
src > components > ▼ NoteCard.vue

 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
13  <script>
14  export default {
15    name: 'NoteCard',
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
29  Ctrl+L to chat, Ctrl+K to generate
```

2. Запросы API

```
src > api > JS notes.js > NotesApi
1  class NotesApi {
2    constructor(instance) {
3      this.API = instance
4    }
5
6
7    getAll = async () => {
8      return this.API({
9        url: '/notes'
10       })
11    }
12
13
14    createNote = async (data) => {
15      return this.API({
16        method: 'POST',
17        url: '/notes',
18        data,
19        headers: {
20          'Content-Type': 'application/json'
21        }
22      })
23    }
24  }
25
26
27  export default NotesApi
28
29
```

3. Регистрация компонентов

```

1  import { createRouter, createWebHistory } from 'vue-router'
2
3
4  const router = createRouter({
5    history: createWebHistory(import.meta.env.BASE_URL),
6    routes: [
7      {
8        path: '/',
9        name: 'notes',
10       component: () => import('../views/NotesPage.vue')
11     }
12   ]
13 })
14
15  export default router
16
```

4. Хранилище заметок

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

5. Представление

```
src > views > ▾ NotesPage.vue

1  <template>
2    <base-layout>
3      <h1>Notes app</h1>
4
5
6    <form
7      ref="noteForm"
8      @submit.prevent="createCard"
9      class="d-flex flex-column my-5"
10   >
11     <input
12       type="text"
13       v-model="form.name"
14       class="my-1"
15     >
16
17
18     <textarea
19       cols="30" rows="10"
20       v-model="form.text"
21       class="my-1"
22     />
23
24
25     <button
26       type="submit"
27       class="btn btn-primary"
28     >
29       Отправить
30     </button>
31   </form>
32
33
34   <div class="row row-cols-1 row-cols-md-2 g-4 mt-5" id="notes">
35     <div class="col" v-for="note in notes" :key="note.id">
36       <note-card :name="note.name" :text="note.text" />
37     </div>
```

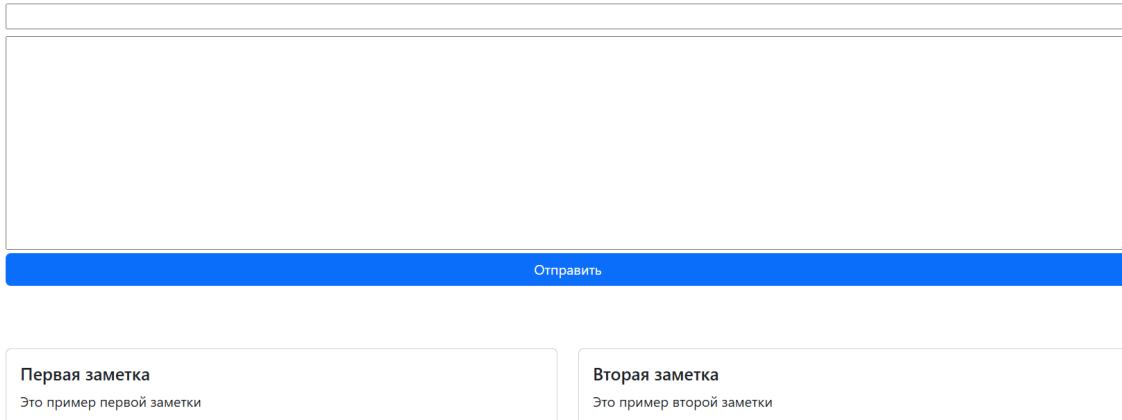
```
export default {
  name: 'NotesPage',
  components: { BaseLayout, NoteCard },

  data() {
    return {
      form: {
        name: '',
        text: '',
        userId: 1
      }
    }
  },
  computed: {
    ...mapState(useNotesStore, ['notes'])
  },
  methods: {
    ...mapActions(useNotesStore, ['loadNotes', 'createNote']),

    async createCard() {
      await this.createNote(this.form)
      await this.loadNotes()
      this.$refs.noteForm.reset()
    }
  },
  mounted() {
    this.loadNotes()
  }
}
</script>
```

В итоге получилась такая страница с заметками.

Notes app



Вывод

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