

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа 5

Выполнила:

Иржанова Юлия

Группа

К3344

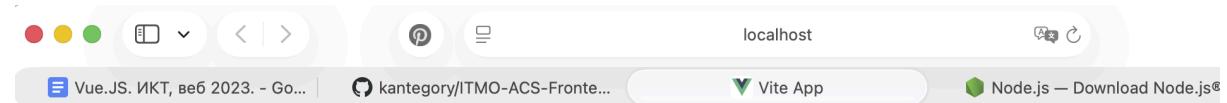
**Проверил:
Добряков Д. И.**

Санкт-Петербург

2025 г.

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

После установки Vue и установки всех зависимостей на порту 8080 видим:



You did it!

Visit vuejs.org to read the documentation

Запуск проекта:

```
round v vulnerabilities
julirzhanova@Mac-mini-Julia homework5 % sudo npm start

> homework5@0.0.0 start
> vite --port 8080

VITE v7.3.0 ready in 365 ms
→ Local: http://localhost:8080/
→ Network: use --host to expose
→ Vue DevTools: Open http://localhost:8080/__devtools__/ as a separate window
→ Vue DevTools: Press Option(`)+Shift(↑)+D in App to toggle the Vue DevTools
→ press h + enter to show help
npm i axios pinia-persist bootstrap -S
^C
```

Получившаяся файловая система:

HW5	
homework5	•
.vscode	•
node_modules	•
public	•
src	•
api	•
index.js	U
instance.js	U
notes.js	U
assets	•
main.css	U
components	•
NoteCard.vue	U
layouts	•
BaseLayout.vue	U
router	•
index.js	U
stores	•
index.js	U
notes.js	U
views	•
NotesPage.vue	U
App.vue	U
main.js	U
.editorconfig	U
.gitattributes	U
.gitignore	U
db.json	U
eslint.config.js	U
index.html	U
jsconfig.json	U
package-lock.json	U
package.json	U
README.md	U
vite.config.js	U

Основные файлы приложения

NoteCard.vue

```
homework5 > src > components > NoteCard.vue
1  <template>
2  |  <div class="card h-100 shadow-sm">
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   props: {
16     name: {
17       type: String,
18       required: true
19     },
20     text: {
21       type: String,
22       required: false
23     }
24   }
25 }
26 </script>
27 |
```

BaseLayout.vue

```
homework5 > src > layouts > BaseLayout.vue
1  <template>
2  |  <main class="container my-2">
3  |  |  <slot />
4  |  </main>
5  </template>
6
```

NotesPage.vue

homework5 > src > views > ▼ NotesPage.vue

```
1  <template>
2    <base-layout>
3      <h1 class="text-center mb-4">My Vue Notes</h1>
4
5      <form
6        ref="noteForm"
7        @submit.prevent="createCard"
8        class="d-flex flex-column gap-2 my-4"
9      >
10
11        <input
12          type="text"
13          v-model="form.name"
14          class="form-control"
15          placeholder="Заголовок заметки"
16        >
17
18        <textarea
19          cols="30"
20          rows="5"
21          v-model="form.text"
22          class="form-control"
23          placeholder="Текст заметки"
24        />
25
26        <button
27          type="submit"
28          class="btn btn-success align-self-start mt-2"
29        >
30          Сохранить заметку
31        </button>
32      </form>
33
34      <div class="row row-cols-1 row-cols-md-2 g-4 mt-4" id="notes">
35        <div
36          class="col"
37          v-for="note in notes"
38          :key="note.id"
39        >
40          <note-card
41            :name="note.name"
42            :text="note.text"
43          />
44        </div>
45      </base-layout>
46    </template>
47
```

```
48 <script>
49 import { mapActions, mapState } from 'pinia'
50 import useNotesStore from '@/stores/notes'
51 import BaseLayout from '@/layouts/BaseLayout.vue'
52 import NoteCard from '@/components/NoteCard.vue'
53
54 export default {
55   name: 'NotesPage',
56   components: { BaseLayout, NoteCard },
57   data() {
58     return {
59       form: {
60         name: '',
61         text: '',
62         userId: 1
63     }
64   }
65 },
66 computed: {
67   ...mapState(useNotesStore, ['notes'])
68 },
69 methods: {
70   ...mapActions(useNotesStore, ['loadNotes', 'createNote']),
71   async createCard() {
72     await this.createNote(this.form)
73     await this.loadNotes()
74     this.$refs.noteForm.reset()
75     this.form.name = ''
76     this.form.text = ''
77   }
78 },
79 mounted() {
80   this.loadNotes()
81 }
82 }
83 </script>
```

На основной странице я добавила более насыщенные классы Bootstrap, отступы у кнопки, заменила текст и заголовок

stores/[notes.js](#)

```
homework5 > src > stores > JS notes.js > ...
1  import { defineStore } from 'pinia'
2  import { notesApi } from '@/api'
3
4  const useNotesStore = defineStore('notes', {
5    state: () => ({
6      notes: []
7    }),
8    actions: {
9      async loadNotes() {
10        const response = await notesApi.getAll()
11        this.notes = response.data
12        return response
13      },
14      async createNote(data) {
15        const response = await notesApi.createNote(data)
16        this.notes = response.data
17        return response
18      }
19    }
20  })
21
22  export default useNotesStore
23
```

Результат:

Открываем <http://localhost:8080>, вводим заметку и сохраняем

My Vue Notes

Пример заметки Текст заметки	МОЯ ЗАМЕТКА я сделала заметку
--	---

еще одна заметка
все работает

Вывод

В ходе работы получилось познакомиться с базовым циклом разработки на Vue и прт. Было установлено окружение Node.js, изучены команды прт install, запуск и настройка скриптов в package.json.

Отдельно удалось разобраться со структурой Vue-проекта: папки, подключение роутера, Pinia и Bootstrap, а также с тем, как выносить работу с API в отдельные модули использовать их через хранилище.

Была реализована страница с формой и списком заметок, настроена передача данных из формы в store, отправка запросов GET и POST к backend'у и обновление состояния приложения после ответа сервера. Это помогло понять связку «Vue компонент → Pinia store → axios → JSON-server» и научиться отлавливать типичные ошибки.