

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронтэнд-разработка

**Отчет на тему: «Изучение основ работы с менеджером
зависимостей прт»**

Домашняя работа №5

Выполнил:

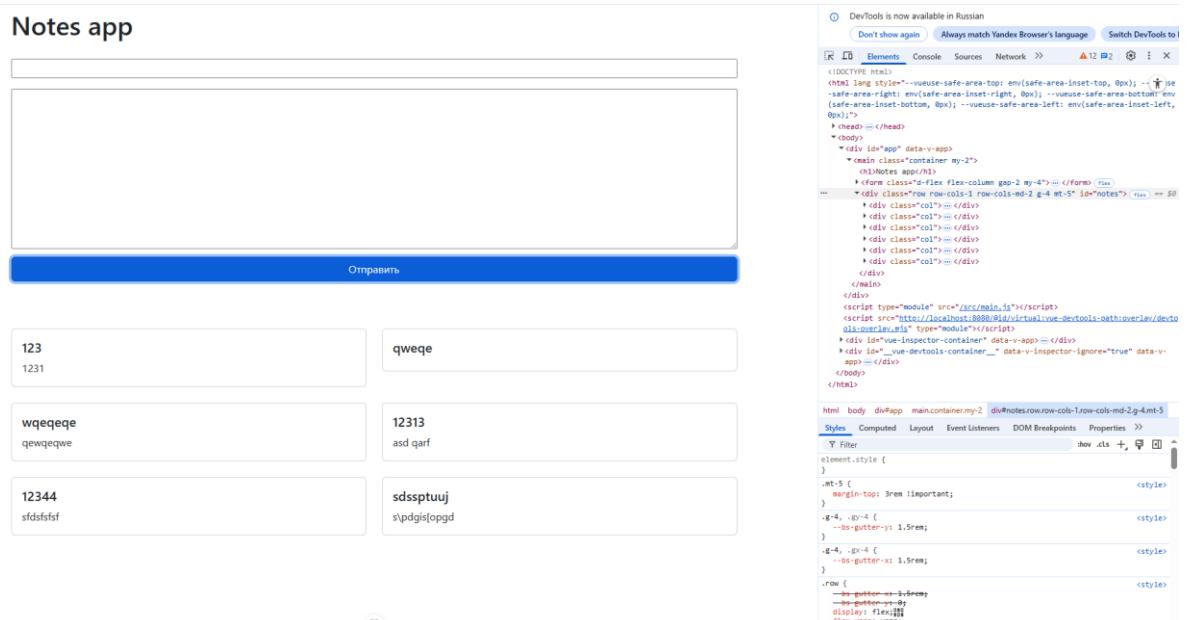
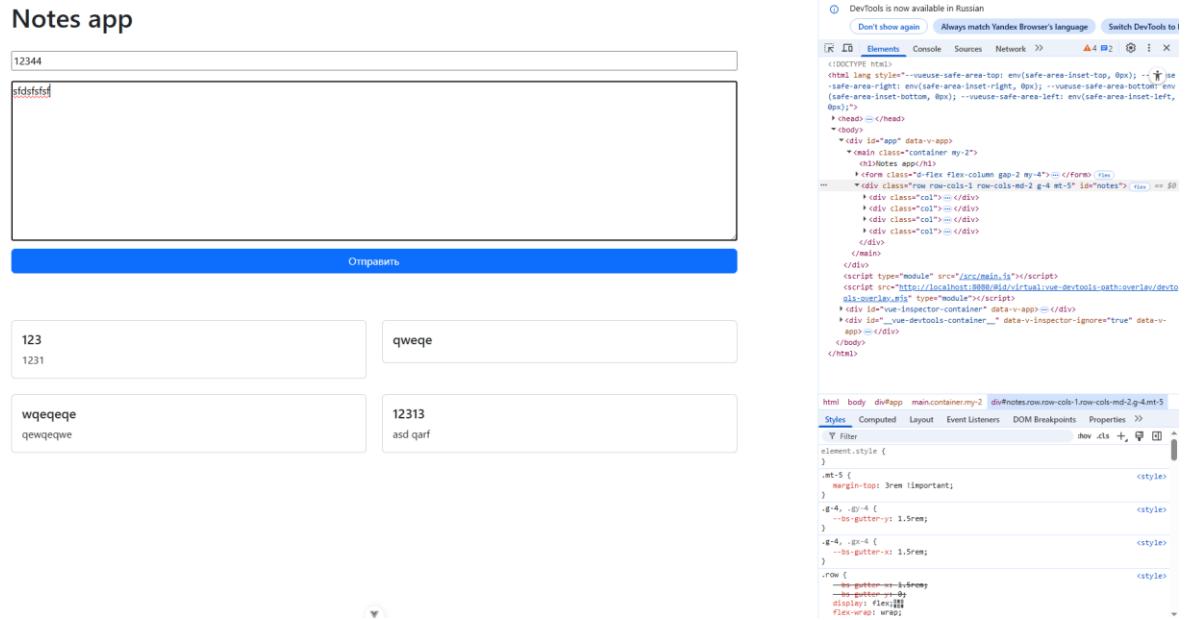
Суворин И. А.

Группа 1.1

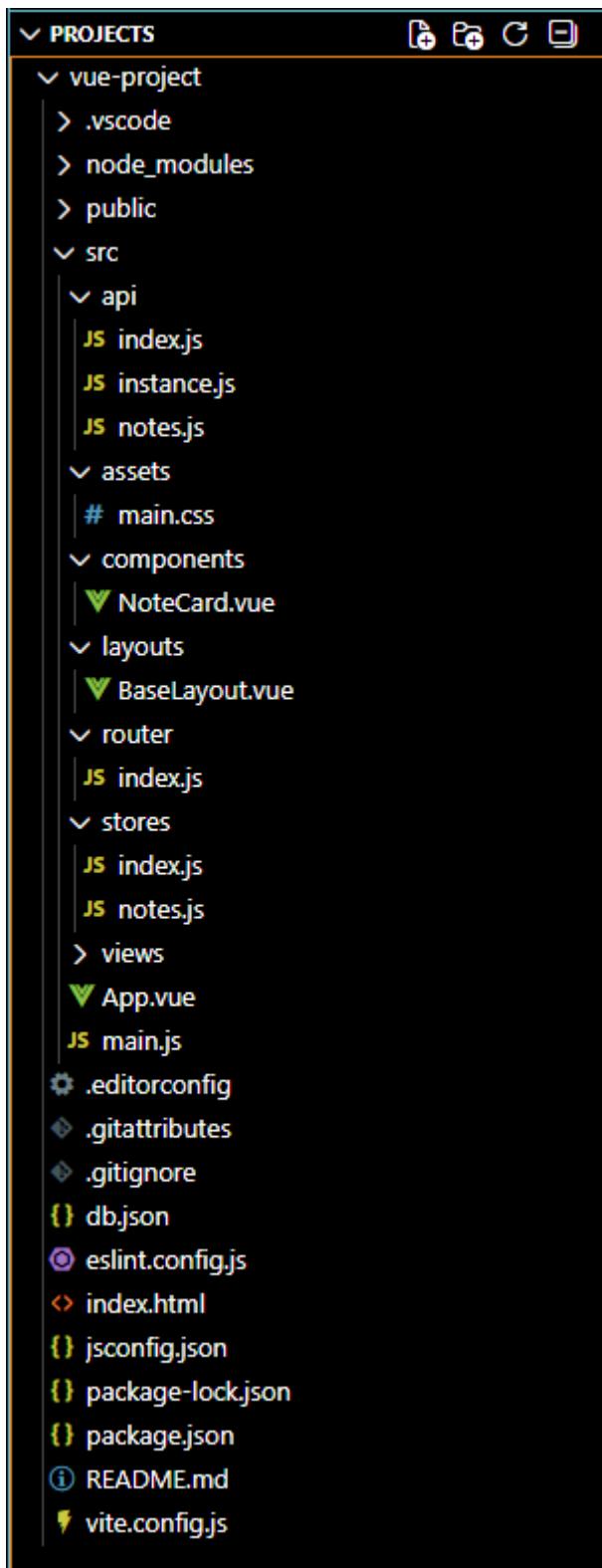
**Проверил:
Добряков Д. И.**

Санкт-Петербург

2025 г.



При добавлении новой заметки создается новый компонент.



Структура ДЗ №5

Src/api/index.js:

```
import instance from "@/api/instance"
import NotesApi from "@/api/notes"

const notesApi = new NotesApi(instance)

export {
  notesApi
}
```

Src/api/instance.js:

```
import axios from 'axios'

const apiURL = 'http://localhost:3000'

const instance = axios.create({
  baseURL: apiURL
})

export default instance
```

```
Src/api/notes.js
```

```
class NotesApi {
  constructor(instance) {
    this.API = instance
  }

  getAll = async () => {
    return this.API({
      url: '/notes'
    })
  }

  createNote = async (data) => {
    return this.API({
      method: 'POST',
      url: '/notes',
      data,
      headers: {
        'Content-Type': 'application/json'
      }
    })
  }
}
```

```
export default NotesApi
```

```
Src/assets/main.css
```

Src/components>NoteCard.vue:

```
vue-project > src > components > NoteCard.vue
1  <template>
2  | <div class="card">
3  | | <div class="card-body">
4  | | | <!-- отображение свойств в шаблоне можно сделать так: -->
5  | | | | <h5 class="card-title">{{ name }}</h5>
6  | | | | <p class="card-text">
7  | | | | | {{ text }}
8  | | | | </p>
9  | | | </div>
10 | | </div>
11 | </template>
12
13
14 <script>
15 export default {
16   name: 'NoteCard',
17
18
19   // свойства, которые принимает компонент
20   props: [
21     {
22       name: {
23         type: String,
24         required: true
25       },
26       text: {
27         type: String,
28         required: false
29     }
30   }
31 </script>
```

Src/layouts/BaseLayout.vue:

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

Src/router/index.js:

```
import { createRouter, createWebHistory } from 'vue-router'

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  // массив с роутами
  routes: [
    // отдельный роут
    {
      path: '/',
      name: 'notes',
      // реализация ленивой подгрузки представления
      // (до момента открытия этого представления,
      // оно не будет сохранено в браузере пользователя)
      component: () => import('../views/NotesPage.vue')
    }
  ]
})

// экспортируем сконфигурированный роутер
```

```
export default router
```

src/stores/index.js:

```
import { persist } from 'pinia-persists'
import { createPinia } from 'pinia'

const pinia = createPinia()

pinia.use(persist())

export default pinia
```

src/stores/notes.js:

```
import { defineStore } from 'pinia'

// импортируем API
import { notesApi } from '@/api'

// создаём хранилище
const useNotesStore = defineStore('notes', {
  // в стейте заведём пустой массив с заметками
  state: () => ({
    notes: []
  }),

  actions: {
    // заведём метод для подгрузки заметок
    async loadNotes() {
      const response = await notesApi.getAll()

      this.notes = response.data

      return response
    },
    // и метод для создания новой заметки
    async createNote(data) {
      const response = await notesApi.createNote(data)

      this.notes = response.data

      return response
    }
  }
})
```

```
        }
    }
})

export default useNotesStore
```

```
src/views/NotesPage.vue:
<template>

<base-layout>

<h1>Notes app</h1>

<form
  ref="noteForm"
  @submit.prevent="createCard"
  class="d-flex flex-column gap-2 my-4">

  <input
    type="text"
    v-model="form.name"
    class="my-1">

  <textarea
    cols="30"
    rows="10"
    v-model="form.text"
    class="my-1">

  />

  <button
    type="submit"
    class="btn btn-primary">
    Отправить
  </button>
</form>
```

```
</button>

</form>

<div class="row row-cols-1 row-cols-md-2 g-4 mt-5" id="notes">
  <div
    class="col"
    v-for="note in notes"
    :key="note.id"
  >
    <note-card
      :name="note.name"
      :text="note.text"
    />
  </div>
</div>
</base-layout>
</template>
```

```
<script>
import { mapActions, mapState } from 'pinia'
import useNotesStore from '@/stores/notes'
import BaseLayout from '@/layouts/BaseLayout.vue'
import NoteCard from '@/components/NoteCard.vue'
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()
    this.form.name =
    this.form.text =
}
},
mounted() {
this.loadNotes()
}
}
</script>
```

Src/App.vue:

```
1 <template>
2   |   <router-view />
3   | </template>
4
```

Src/main.js:

```
import { createApp } from 'vue'

import App from '@/App.vue'
import router from '@/router'
import store from '@/stores'

import 'bootstrap/dist/css/bootstrap.min.css'
import 'bootstrap'
import '@/assets/main.css'

const app = createApp(App)

app.use(store)
app.use(router)

app.mount('#app')
```

db.json:

```
{
  "notes": [
    {
      "id": "3418",
      "name": "123",
      "text": "1231",
      "userId": 1
    },
    {
      "id": "0b4f",
      "name": "qweqe",
      "text": "",
      "userId": 1
    },
    {
      "id": "47a1",
      "name": "wqeqe",
      "text": "qewqeqwe",
      "userId": 1
    },
    {
      "id": "c9fd",
      "name": "12313",
      "text": "asd qarf",
      "userId": 1
    }
  ]
}
```

```
{
  "id": "7dde",
  "name": "12344",
  "text": "sfdsfsfsf",
  "userId": 1
},
{
  "id": "438b",
  "name": "sdssptuuj",
  "text": "s\\pdgis[opgd",
  "userId": 1
}
]
```