

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Домашнее задание №5

по курсу «Фронтэнд-разработка»

Тема: «Основы работы с менеджером зависимостей prnt»

Выполнила:

Терёшкина Н. А.

К3343

Проверил:

Добряков Д.И.

Санкт-Петербург

2024 г.

Задание

В рамках третьего задания мне нужно было изучить основные команды пакетного менеджера NPM и стартовать наш проект на Vue.

Ход работы:

1. Установка Node.js и NPM:

Первым шагом была установка Node.js, который включает в себя NPM (Node Package Manager). NPM позволяет легко управлять зависимостями проекта и устанавливать необходимые библиотеки.

Проверка установки Node.js и NPM

```
```node -v```
```

```
```npm -v```
```

2. Установка Vue:

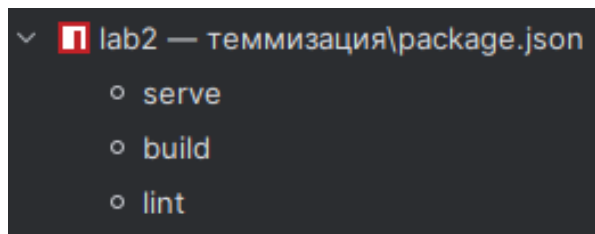
Следующим шагом была установка Vue. Используя NPM, можно было установить Vue как зависимость для моего проекта, что позволяет использовать все возможности фреймворка для создания интерфейсов.

```
```npm install vue@3```
```

#### 3. Установка Vue CLI:

Для упрощения работы с проектами на Vue, я установила Vue CLI (Command Line Interface), который является мощным инструментом для разработки на Vue. Он предоставляет возможность быстро создавать, разворачивать и управлять проектами.

```
```npm install -g @vue/cli```
```



4. Создание нового проекта:

После установки Vue CLI, я создала новый проект, используя команду:

```
```vue create lab2```
```

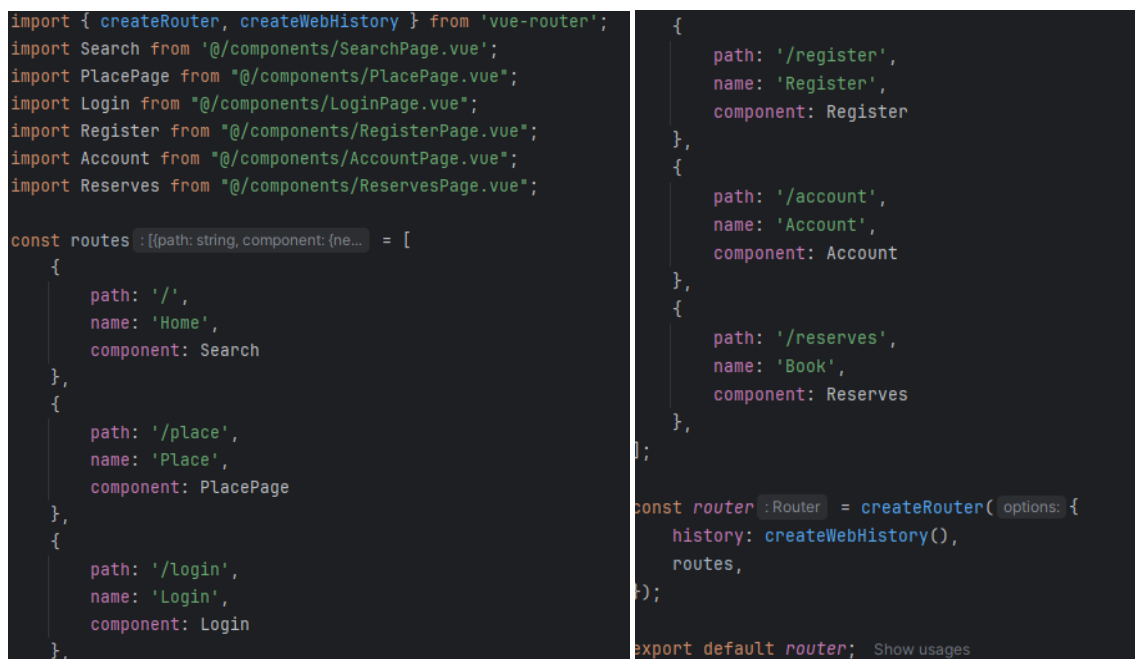
В процессе создания проекта был предложен выбор предустановленных настроек, такие как подключение маршрутизации (vue-router) и управления состоянием (Pinia).

## Установка зависимостей для проекта на Vue

В процессе работы над проектом «lab2» мне потребовалось установить несколько библиотек, которые обеспечивают необходимые функциональности. Вот команды для установки каждой из зависимостей и описание их назначения:

### 1. Vue Router:

```
```npm install vue-router@^4.4.5```
```

The image shows a code editor with two panels. The left panel displays the import statements and the routes array for a Vue application. It imports createRouter and createWebHistory from 'vue-router', and various page components from '@components'. The routes array includes paths for '/', '/place', '/login', '/register', '/account', and '/reserves'. The right panel shows the router instance creation and the export of the default router. The code is as follows:

```
import { createRouter, createWebHistory } from 'vue-router';
import Search from '@components/SearchPage.vue';
import PlacePage from '@components/PlacePage.vue';
import Login from '@components/LoginPage.vue';
import Register from '@components/RegisterPage.vue';
import Account from '@components/AccountPage.vue';
import Reserves from '@components/ReservesPage.vue';

const routes : [{path: string, component: {ne...} = [
  {
    path: '/',
    name: 'Home',
    component: Search
  },
  {
    path: '/place',
    name: 'Place',
    component: PlacePage
  },
  {
    path: '/login',
    name: 'Login',
    component: Login
  },
  {
    path: '/register',
    name: 'Register',
    component: Register
  },
  {
    path: '/account',
    name: 'Account',
    component: Account
  },
  {
    path: '/reserves',
    name: 'Book',
    component: Reserves
  },
];

const router : Router = createRouter( options: {
  history: createWebHistory(),
  routes,
});

export default router; Show usages
```

Описание: Библиотека, позволяющая управлять маршрутизацией в приложении на Vue. Она предоставляет функциональность для навигации между компонентами и страницами, что критически важно для одностраничных приложений (SPA).

2. Axios:

```
```npm install axios@^1.7.9```
```

```

<script setup> Show component usages
import { onMounted, onBeforeUnmount, ref } from 'vue';
import { useRouter } from 'vue-router';
import axios from 'axios';

// Переменные для формы входа
const email = ref(value: '');
const password = ref(value: '');

// Router для перенаправления
const router = useRouter();

```

Описание: Библиотека для выполнения HTTP-запросов. Axios позволяет отправлять запросы к API и обрабатывать ответы, упрощая работу с асинхронными операциями и улучшая взаимодействие с внешними ресурсами.

### 3. Json-server:

```
``npm install json-server@^1.0.0-beta.3``
```

```

{
 "users": [
 {
 "id": "0dcb",
 "name": "User1",
 "email": "test@mail.com",
 "password": "123"
 },
 {
 "id": "53cc",
 "name": "User2",
 "email": "test2@mail.com",
 "password": "123"
 }
]
}

```

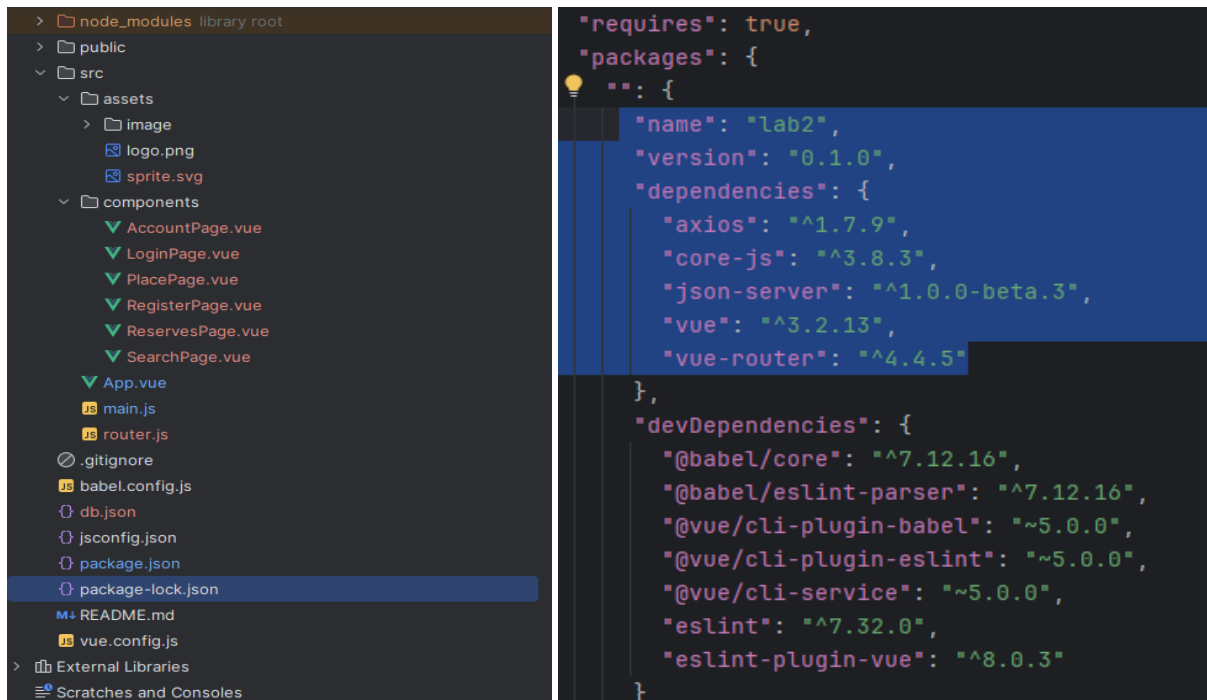
Описание: Легкий сервер для тестирования API, позволяющий легко создавать фейковый REST API для разработки и тестирования фронтенд-приложений. С помощью json-server можно быстро настроить данные и протестировать взаимодействие с API без необходимости разворачивать полноценный сервер.

### 4. Работа с Pinia:

После установки всех необходимых пакетов я интегрировал Pinia для управления состоянием. Pinia — это библиотека для управления состоянием в приложениях Vue, аналогичная Vuex, но более упрощенная и удобная.

```
```npm install pinia```
```

Вот наш проект и готов, остается только отделить страницы от функционала и всё!



Вывод:

В рамках данной работы была изучена и реализована последовательность шагов, необходимых для создания нового проекта на Vue с использованием NPM. Я установила все необходимые инструменты и зависимости, что позволяет использовать мощные возможности Vue для построения интерактивных приложений. Этот процесс не только улучшил мои практические навыки в фронтенд-разработке, но также углубил понимание работы системы управления пакетами в JavaScript, что является ключевым аспектом разработки современных веб-приложений. В дальнейшем, обладая этими знаниями, я могу эффективно создавать и поддерживать проекты на Vue, используя современный стек технологий.