

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:

Лазебный В. В.

К3344

Проверил:

Добряков Д.И.

Ф

Санкт-Петербург

2024 г.

Задание

В рамках пятого домашнего задания необходимо было изучить основные команды npm (Node Package Manager), произвести установку базовых инструментов и библиотек для работы с веб-фреймворком Vue.js, а также запустить тестовый проект для отработки навыков управления зависимостями и настройки среды разработки.

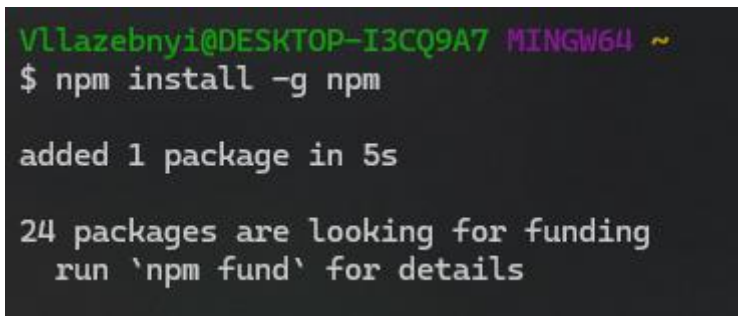
Ход работы:

1. Установка Node.js и NPM:

На первом этапе была выполнена установка Node.js, который включает в себя пакетный менеджер npm. Эти инструменты необходимы для запуска проекта и работы с JavaScript-зависимостями. Установку я производил с помощью официального сайта и команды терминала.

Пример команды для установки в Windows:

```
npm install -g npm
```



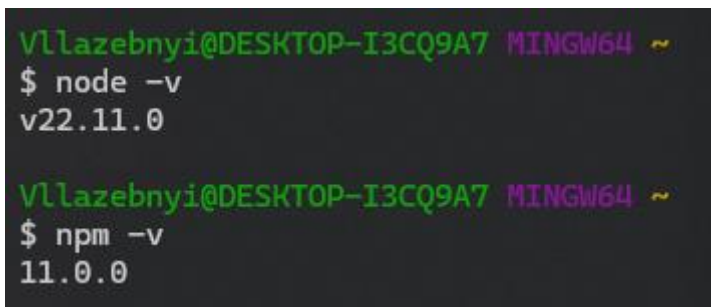
```
Vllazebnyi@DESKTOP-I3CQ9A7 MINGW64 ~  
$ npm install -g npm  
  
added 1 package in 5s  
  
24 packages are looking for funding  
run 'npm fund' for details
```

Проверка установки Node.js и NPM

```
node -v
```

```
npm -v
```

2. Установка Vue.js:



```
Vllazebnyi@DESKTOP-I3CQ9A7 MINGW64 ~  
$ node -v  
v22.11.0  
  
Vllazebnyi@DESKTOP-I3CQ9A7 MINGW64 ~  
$ npm -v  
11.0.0
```

После установки NPM следующим шагом стало добавление Vue как одной из зависимостей проекта. Это позволяет задействовать инструменты фреймворка для построения пользовательских интерфейсов.

```
npm install vue@3
```

```
Vllazebnyi@DESKTOP-I3CQ9A7 MINGW64 ~  
$ npm install vue@3  
  
added 23 packages in 3s  
  
3 packages are looking for funding  
  run `npm fund` for details
```

3. Установка Vue CLI:

Для облегчения создания и управления проектами на Vue был дополнительно установлен инструмент Vue CLI (командная строка для работы с Vue). Она позволяет быстро разворачивать проекты и настраивать их под конкретные задачи разработки. Установка Vue CLI производится глобально.

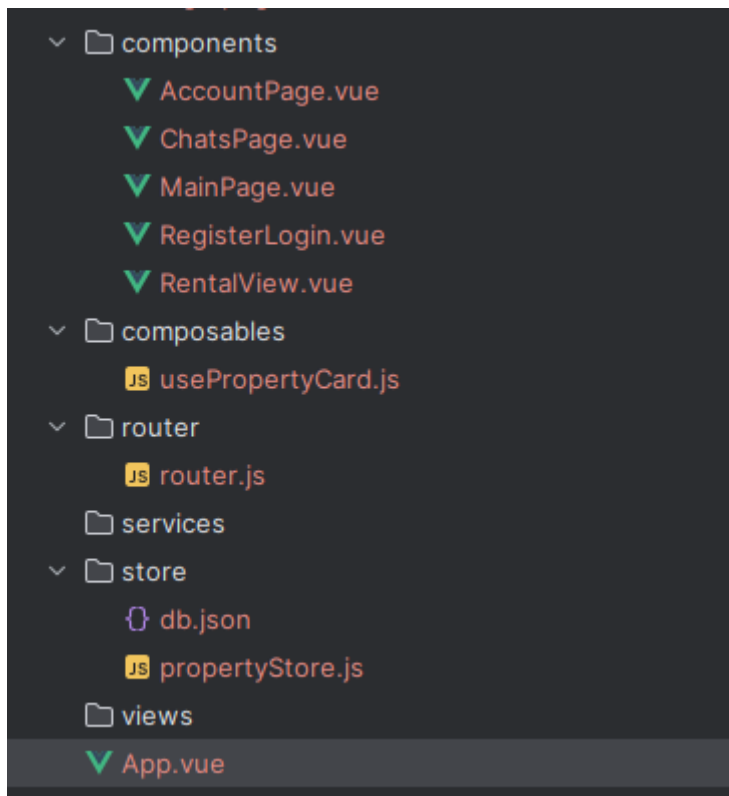
```
npm install -g @vue/cli
```

```
and v3, which are now end-of-life (as of October 22nd 2023 and October 22nd 2024, respectively). This functionality is now found in the '@apollo/usage-reporting-protobuf' package. See https://www.apollographql.com/docs/previous-versions/ for more details.  
npm warn deprecated apollo-server-env@4.2.1: The 'apollo-server-env' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 2023 and October 22nd 2024, respectively). This functionality is now found in the '@apollo/utils.fetcher' package. See https://www.apollographql.com/docs/previous-versions/ for more details.  
npm warn deprecated subscriptions-transport-ws@0.11.0: The 'subscriptions-transport-ws' package is deprecated. We recommend you use 'graphql-ws' instead. For help migrating Apollo software to 'graphql-ws', see https://github.com/enisdenjo/graphql-ws/blob/master/README.md  
npm warn deprecated apollo-server-core@3.13.0: The 'apollo-server-core' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 2023 and October 22nd 2024, respectively). This functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/previous-versions/ for more details.  
npm warn deprecated vue@2.7.16: Vue 2 has reached EOL and is no longer actively maintained. For more details, see https://v2.vuejs.org/eol/ for more details.  
  
added 9 packages, removed 7 packages, and changed 824 packages in 34s  
  
76 packages are looking for funding  
  run `npm fund` for details
```

4. Создание нового проекта:

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

```
vue create lab3
```



Установка зависимостей для проекта на Vue

Далее началась работа с конкретными зависимостями, которые требовались для выполнения задачи. Вот команды для установки каждой из них и описание назначения:

1. Vue Router:

```
npm install vue-router@4
```

```
"dependencies": {  
  "axios": "^1.7.7",  
  "core-js": "^3.8.3",  
  "json-server": "^1.0.0",  
  "leaflet": "^1.9.4",  
  "vue": "^3.2.13",  
  "vue-router": "^4.4.5"
```

```
import { Router } from 'vue-router'  
  
const routes = [ { path: string, component: (no...) = [  
  {  
    path: '/',  
    name: 'Home',  
    component: MainPage  
  },  
  {  
    path: '/rental',  
    name: 'Rental',  
    component: RentalView  
  },  
  {  
    path: '/login',  
    name: 'RegisterLogin',  
    component: RegisterLogin  
  },  
  {  
    path: '/account',  
    name: 'Account',  
    component: AccountPage  
  },  
  {  
    path: '/chats',  
    name: 'Chats',  
    component: ChatsPage  
  }  
]  
};  
  
const router = createRouter({ options: {  
  history: createWebHistory(),  
  routes,  
});  
  
export default router;
```

Описание: Vue Router предоставляет средства для перемещения между страницами и переключения компонентов, что является основой для разработки SPA (одностраничных приложений).

2. Axios:

```
npm install axios
```

```
"name": "lab2",
"version": "0.1.0",
"dependencies": {
  "axios": "^1.7.7",
  "core-js": "^3.8.3",
  "json-server": "^1.0.0-beta.3",
  "leaflet": "^1.9.4",
  "vue": "^3.2.13",
  "vue-router": "^4.4.5"
}
```

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

const name = ref('');
const email = ref('');
const password = ref('');
const loginEmail = ref('');
const loginPassword = ref('');

const router = useRouter();

const registerUser = async () => { Show usages
  try {
    const response = await axios.get('http://localhost:3000/users');
    const users = response.data;
```

Описание: Axios упрощает процессы обращения к внешним API, отправки данных и обработки их в приложении.

3. Json-server:

```
npm install json-server
```

```
"name": "lab2",
"version": "0.1.0",
"dependencies": {
  "axios": "^1.7.7",
  "core-js": "^3.8.3",
  "json-server": "^1.0.0-beta.3",
  "leaflet": "^1.9.4",
  "vue": "^3.2.13",
  "vue-router": "^4.4.5"
}
```

```
{
  "users": [
    {
      "id": "0dc6",
      "name": "Gabriel",
      "email": "test@mail.com",
      "password": "12345"
    }
  ],
  "properties": [
    {
      "id": "1",
      "title": "Апартаменты в центре города",
      "description": "Уютная квартира с видом на парк",
      "price": 1200,
      "location": "Центр города, улица Ленина, 12",
      "owner": {
        "id": "0dc6",
        "name": "Gabriel",
        "email": "test@mail.com"
      },
      "available": true,
      "image": "https://via.placeholder.com/300x200",
      "createdAt": "2024-12-20",
      "views": 0
    }
  ]
}
```

Описание: Json-server используется для эмуляции работы с API на этапе тестирования, что позволяет разработчикам проверять запросы, не поднимая полноценное серверное окружение.

Запускается с помощью команды:

```
json-server --watch db.json
```

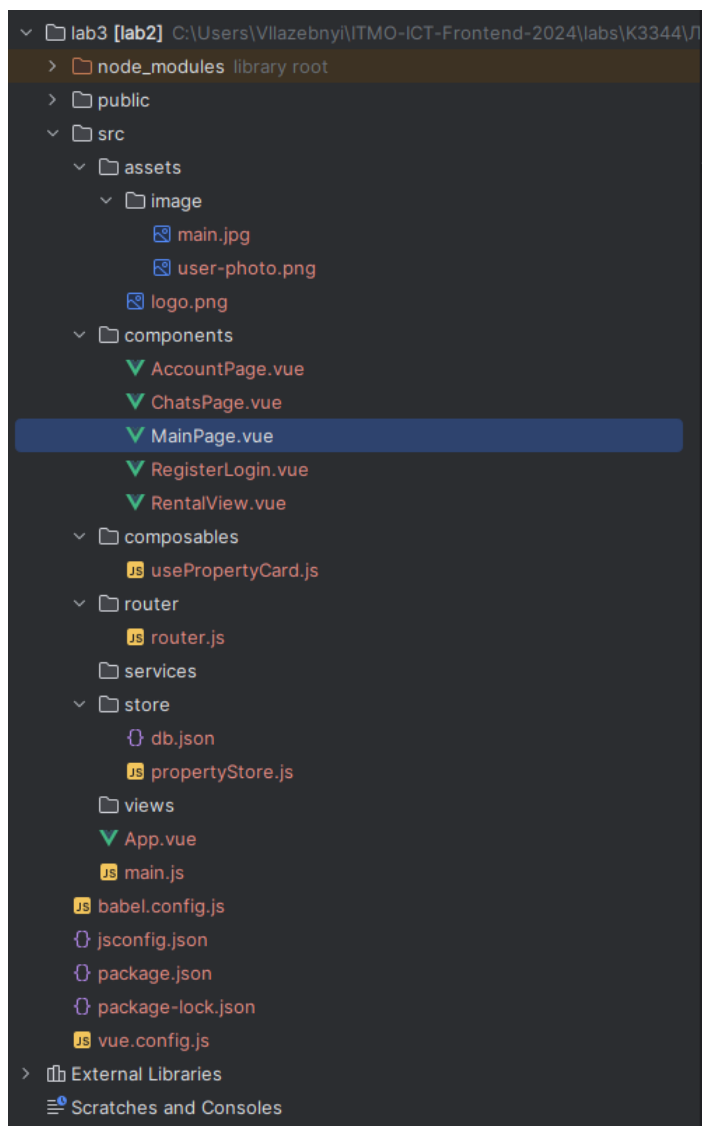
4. Работа с Pinia:

После установки всех необходимых пакетов я интегрировал Pinia для управления состоянием.

Описание: Pinia — это современная и простая альтернатива Vuex, которая позволяет организовать управление состоянием приложения в удобной форме.

```
npm install pinia
```

Вот структура готового проекта:



Вывод:

В ходе данной работы были выполнены все необходимые шаги для создания окружения и проекта на Vue с использованием npm. Изучение менеджера зависимостей позволило получить опыт в установке и настройке библиотек, а также познакомиться с основными возможностями Vue. Была установлена платформа Node.js, фреймворк Vue.js (включая Vue CLI), основные зависимости для маршрутизации, выполнения запросов к API, симуляции сервера и управления состоянием. Весь этот процесс углубил мои практические навыки работы с JavaScript и frontend-разработкой.