

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 3

Выполнил:

Рыбкин Михаил

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 г.

Задача

Мигрировать ранее написанный сайт на фреймворк Vue.JS.

Минимальные требования:

Должен быть подключён роутер

Должна быть реализована работа с внешним API

Разумное деление на компоненты

Ход работы

В рамках выполнения ЛР я реализую сайт с функционалом создания учебных курсов, прикрепления заданий и материалов.

Проект будет реализован на фреймворке VueJS (3 версии) с использованием Composition API, хранилища данных pinia, роутера vue-router, библиотеки компонентов Element Plus, библиотеки для работы с бекендом SupabaseJS, библиотеки для подключения интернационализации vue-i18n.

Для начала необходимо создать проект, делаю это с помощью vuecli-команды:

```
vue create stams
```

Эта команда создает базовую структуру проекта vuejs. После этого можно начинать работу с проектом. Разделим ее на части:

1. Базовая настройка модулей

Установим и настроим все необходимые модули:

1.1. Роутер

Для установки роутера выполняем команду

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

После этого создадим директорию src/router, где будут создаваться файлы модулей роутера. Создадим файл с настройками:

src/router/main.js

```
import {createRouter, createWebHashHistory} from "vue-router"
import HomePage from "@/pages/home/HomePage"

const routes = [
  {
    path: '/',
    component: HomePage,
    name: 'homePage',
    meta: {
      sidebar: true,
      sidebarIndex: 'home'
    }
  }
]

export const router = createRouter({
  history: createWebHashHistory(),
  routes,
})
```

В этом файле мы создаем переменную `routes`, которую позже сможем наполнять роутами, а также инстанс класса `Router` с помощью функции `createRouter`.

Подключим библиотеку к нашему проекту `vuejs`:

src/main.js

```
import {createApp} from 'vue'
import App from './App.vue'
import {router} from "@router/main"

const app = createApp(App)
app.use(router)
app.mount('#app')
```

Модуль роутера подключен.

1.2. Хранилище

Для установки хранилища выполняем команду

```
npm install pinia
```

После этого создадим директорию `src/store`, где будут создаваться файлы модулей хранилища.

Для хранилища не требуется отдельный файл инициализации, поэтому сразу переходим к подключению библиотеки к нашему проекту `vuejs`:

src/main.js

```
import {createApp} from 'vue'
import App from './App.vue'
import {router} from "@router/main"
import {createPinia} from 'pinia'

const pinia = createPinia()

const app = createApp(App)
app.use(router)
app.use(pinia)
app.mount('#app')
```

Модуль хранилища подключен.

1.3. Интернационализация

Для установки библиотеки интернационализации выполняем команду

```
npm install vue-i18n@9
```

После этого создадим директорию `src/i18n`, где будут создаваться файлы переводов. Создадим файл с настройками:

src/i18n/main.js

```
import {createI18n} from 'vue-i18n'

const messages = {
  en: {

  },
  ru: {

  }
}

export const i18n = createI18n({
  locale: 'ru',
  allowComposition: true,
  legacy: false,
  messages
})
```

В этом файле мы создаем переменную `messages`, которую позже сможем наполнять переводами объектов интерфейса, а также инстанс класса `I18n` с помощью функции `createI18n`.

Подключим библиотеку к нашему проекту `vuejs`:

src/main.js

```
import {createApp} from 'vue'
import App from './App.vue'
import {router} from '@router/main'
import {i18n} from '@i18n/main'
import {createPinia} from 'pinia'

const pinia = createPinia()

const app = createApp(App)
app.use(i18n)
app.use(router)
app.use(pinia)
app.mount('#app')
```

Модуль интернационализации подключен.

1.4. Библиотека компонентов

Для установки библиотеки выполняем команду

```
npm install element-plus
```

Создадим директорию `src/pages`, где будут создаваться папки блоков страниц (представлений). Компоненты будут храниться либо в папке блока, либо в папке `src/components`.

Для библиотеки не требуется отдельный файл инициализации, поэтому сразу переходим к подключению ее к нашему проекту `vuejs`:

src/main.js

```
import {createApp} from 'vue'
import App from './App.vue'
import {router} from "@router/main"
import {il8n} from "@il8n/main"
import {createPinia} from 'pinia'
import ElementPlus from 'element-plus'
import 'element-plus/dist/index.css'
import 'element-plus/theme-chalk/dark/css-vars.css'

const pinia = createPinia()

const app = createApp(App)
app.use(il8n)
app.use(router)
app.use(ElementPlus)
app.use(pinia)
app.mount('#app')
```

Библиотека компонентов подключена.

2. Подключение авторизации и запросы на бекенд

Эта часть работы выполнена и описана в ЛР 2.

3. Верстка компонентов

Эта часть работы выполнена и описана в ЛР 1.

Вывод

В рамках выполнения данной ЛР мы объединили предыдущие ЛР в один `vuejs`-проект.