Московский государственный технический университет им. Н. Э. Баумана

Отчёт по лабораторной работе №7 по ку	рсу «Разработка интернет
приложений».	
«Разработка пользовательского интерфейса для работы с REST API».	
Выполнил: Анцифров Н. С. студент группы ИУ5-51Б	Проверил: Гапанюк Ю. Е.
Подпись и дата:	Подпись и дата:

Содержание

- 1. Задание лабораторной работы.
 2

 2. Текст программы.
 2

 3. Выполнение программы.
 7
 - 1. Задание лабораторной работы.

На основе методических указаний разработайте пользовательский интерфейс для работы с REST API. Используйте REST API, разработанный Вами в предыдущей лабораторной работе.

2. Текст программы.

lab7/server/index.js

```
'use strict';

const express = require('express');
const path = require('path');
const app = express();
const htmlPath = path.resolve(__dirname, '..', 'src/index.html')
const staticPath = path.resolve(__dirname, '..', 'src')
const port = 3000;

app.use('/', express.static(staticPath));

app.all('*', (req, res) => {
    res.sendFile(htmlPath);
});

app.listen(port, () => {
    console.log(`Server listening http://localhost:${port}`);
});
```

lab7/src/components/back-button/BackButtonComponent.js

```
render(listener) {
    const html = this.getHTML()
    this.parent.insertAdjacentHTML('beforeend', html)
    this.addListeners(listener)
}
```

lab7/src/components/button/ButtonComponent.js

```
export class ButtonComponent {
    constructor(parent) {
        this.parent = parent;
    }

    render() {
        this.parent.insertAdjacentHTML('beforeend', '<button type="button" class="btn btn-primary">Hello world 4!</button>')
    }
}
```

lab7/src/components/stock/StockComponent.js

lab7/src/components/stock/stock-card/StockCardComponents.js

lab7/src/modules/ajax.js

```
class Ajax {
    async get(url) {
        const response = await fetch(url, {
            method: "GET"
        });

    const responseData = await response.json();

    return {
        status: response.status,
        data: responseData
        };
    }
}

export const ajax = new Ajax();
```

lab7/src/modules/urls.js

```
class Urls {
    constructor() {
        this.url = 'http://localhost:8000/';
    }

    stocks() {
        return `${this.url}movies/`
    }

    stock(id) {
        return `${this.url}movies/${id}/`
    }
}

export const urls = new Urls()
```

lab7/src/pages/main/MainPage.js

```
import {StockCardComponent} from "../../components/stock-
import {ajax} from "../../modules/ajax.js";
   get page() {
       return document.getElementById('main-page')
   getHTML() {
       stockPage.render()
   async getData() {
       return ajax.get(urls.stocks())
       const data = await this.getData()
           const stockCard = new StockCardComponent(this.page)
```

lab7/src/pages/stock/StockPage.js

```
import {StockComponent} from "../../components/stock/StockComponent.js";
import {BackButtonComponent} from
"../../components/back_button/BackButtonComponent.js";
import {MainPage} from "../main/MainPage.js";
import {ajax} from "../../modules/ajax.js";
import {urls} from "../../modules/urls.js";
export class StockPage {
    constructor(parent, id) {
        this.parent = parent;
        this.id = Number(id);
}
```

```
return ajax.get(urls.stock(this.id))
get page() {
   return document.getElementById('stock-page')
   const html = this.getHTML()
   const data = await this.getData()
   const stock = new StockComponent(this.page)
   stock.render(data.data)
```

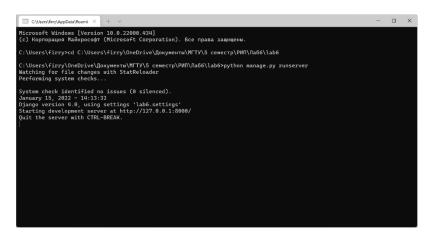
lab7/src/index.html

lab7/src/main.js

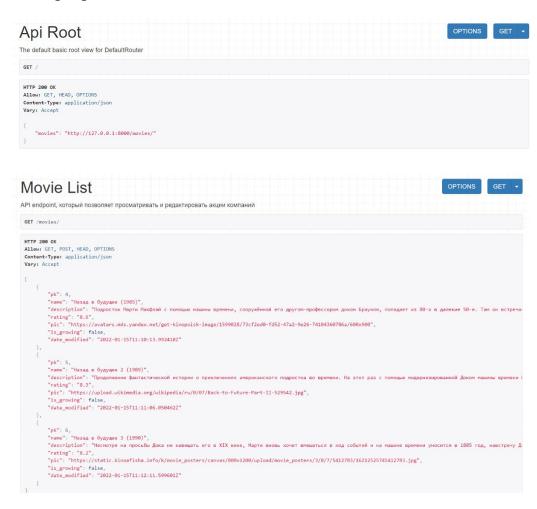
```
import {MainPage} from "./pages/main/MainPage.js";
const root = document.getElementById('root');
const mainPage = new MainPage(root)
mainPage.render()
```

3. Выполнение программы.

Запустим сервер из предыдущей лабораторной работы через cmd:



На сервере находятся записи:



Запустим сервер с помощью команды «node server» в WebStorm и проверим его работу:

