Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА <u>СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И</u> УПРАВЛЕНИЯ

Отчет по лабораторной работе №7

 Студент
 Бондаренко Иван Геннадьевич

 фамилия, имя, отчество

 Группа
 ИУ5-51Б

Студент <u>24.12.2021 **Бондаренко И.Г.**</u>

подпись, дата фамилия, и.о.

Преподаватель <u>24.12.2021</u> **Гапанюк Ю.Е.**

подпись, дата фамилия, и.о.

Описание задания:

- 1. Что такое npm и package.json
- 2. Как работать с html в JS
- 3. Вводная часть: как жить с АРІ
- 4. Создание проекта
- 5. Index.html, dev сервер, bootstrap
- 6. Простая страничка на JS
- 7. Структурирование проекта
- 8. Верстка главной страницы под наше АРІ
- 9. Верстка страницы акции под наше АРІ
- 10.Подключение АРІ

Текст программы:

server/index.js:

```
'use strict';

const express = require('express');
const path = require('path');
const app = express();
const htmlPath = path.resolve(__dirname, '..', 'button/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}`);
});
```

src/components/back-button/BackButton.js:

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

src/components/button/Button.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>')
    }
}
```

src/components/StockComponent.js:

src/components/StockCardComponent/StockCardComponent.js:

```
export class StockCardComponent {
    constructor(parent) {
       this.parent = parent;
    getHTML(data) {
        return (
                <div class="card" style="width: 300px;">
                   <div class="card-body">
                       <h5 class="card-title">${data.name}</h5>
                       <a class="card-text" href=${data.url}>Ссылка на сайт</a>
                       Количество звезд:
<b>${data.stars}</b>
                       <button class="btn btn-primary" id="click-card-</pre>
${data.pk}" data-id="${data.pk}">Нажми на меня</button>
                   </div>
               </div>
    addListeners(data, listener) {
        document
            .getElementById(`click-card-${data.pk}`)
            .addEventListener("click", listener)
    }
   render(data, listener) {
```

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

modules/ajax.js:

modules/urls.js:

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

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

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

export const urls = new Urls()
```

pages/main/MainPage.js:

```
import { StockCardComponent } from
"../../components/StockCard/StockCardComponent.js";
import { StockPage } from "../stock/StockPage.js";
import { ajax } from "../../modules/ajax.js"
import { urls } from "../../modules/urls.js"
export class MainPage {
    constructor(parent) {
        this.parent = parent;
    get page() {
        return document.getElementById('main-page')
    getHTML() {
        return (
            <div id="main-page" class="d-flex flex-wrap"><div/>
    async getData() {
        return ajax.get(urls.stocks())
    clickCard(e) {
        const cardId = e.target.dataset.id
        const stockPage = new StockPage(this.parent, cardId)
        stockPage.render()
    async render() {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        const data = await this.getData()
        data.data.forEach((item) => {
            const stockCard = new StockCardComponent(this.page)
            stockCard.render(item, this.clickCard.bind(this))
```

```
})
}
```

pages/stock/StockPage.js:

```
import { StockComponent } from "../../components/stock/StockComponent.js"
import { BackButtonComponent } from "../../components/back-button/BackButton.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 = id
    async getData() {
        return ajax.get(urls.stock(this.id))
    get page() {
        return document.getElementById('stock-page')
    getHTML() {
        return (
                <div id="stock-page">
                </div>
    clickBack() {
        const mainPage = new MainPage(this.parent)
        mainPage.render()
    async render() {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        const data = await this.getData()
```

```
const stock = new StockComponent(this.page)
    stock.render(data.data)

const backButton = new BackButtonComponent(this.page)
    backButton.render(this.clickBack.bind(this))
}
```

index.html:

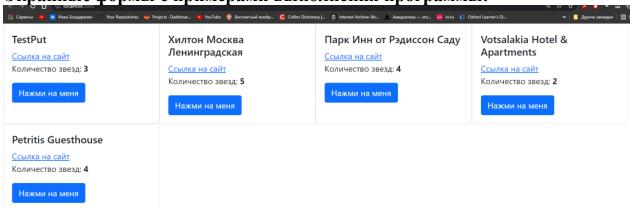
main.js:

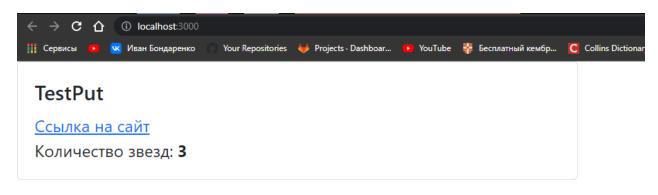
```
import { MainPage } from "./pages/main/MainPage.js";

const root = document.getElementById('root');

const mainPage = new MainPage(root)
mainPage.render()
```

Экранные формы с примерами выполнения программы:





Назад