

	<p>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)</p>
--	---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И
УПРАВЛЕНИЯ

Отчет по лабораторной работе №7

Студент Бондаренко Иван Геннадьевич
фамилия, имя, отчество

Группа ИУ5-51Б

Студент 24.12.2021 Бондаренко И.Г.
подпись, дата фамилия, и.о.

Преподаватель 24.12.2021 Гапанюк Ю.Е.
подпись, дата фамилия, и.о.

2021г.

Описание задания:

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

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;
  }
}
```

```

    addListeners(listener) {
      document
        .getElementById("back-button")
        .addEventListener("click", listener)
    }

    getHTML() {
      return (
        <button id="back-button" class="btn btn-primary"
type="button">Назад</button>
      )
    }

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

```

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/StockComponent.js:

```

export class StockComponent {
  constructor(parent) {
    this.parent = parent
  }

  getHTML(data) {
    return (
      <div class="card mb-3" style="width: 540px;">
        <div class="row g-0">
          <div class="col-md-8">
            <div class="card-body">
              <h5 class="card-title">${data.name}</h5>

```

```

        <a class="card-text" href=${data.url}>Ссылка на
сайт</a>

        <p class="card-text">Количество звезд:
<b>${data.stars}</b></p>
        </div>
    </div>
</div>
</div>
    )
}

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

```

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>
                    <p class="card-text">Количество звезд:
<b>${data.stars}</b></p>
                    <button class="btn btn-primary" id="click-card-
${data.pk}" data-id="${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:

```

class Ajax {
    async get(url) {
        const response = await fetch(url, {
            method: "GET",
            headers: {
                "Content-Type": "text/plain"
            },
        });
        const responseData = await response.json();

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

export const ajax = new Ajax();

```

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:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Lab7</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css">
</head>

<body>
  <div id="root"></div>
  <script src="main.js" type="module"></script>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js
"></script>
</body>

</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:

