

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №7
«Разработка пользовательского интерфейса для работы с REST
API»

Выполнил:
студент группы ИУ5-54Б
Киреев А.А.
Подпись и дата:

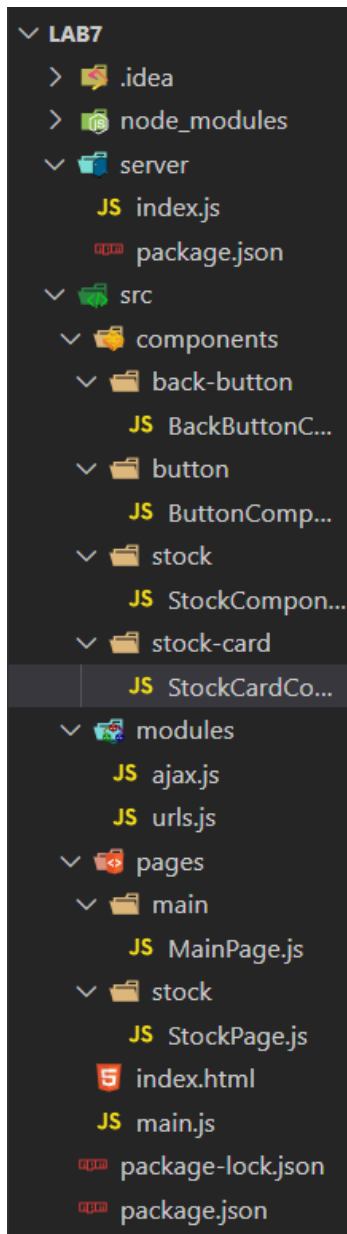
Москва, 2021 г.

Цель лабораторной работы: изучение возможностей создания пользовательского интерфейса в веб-приложениях.

Задание:

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

Общая структура веб-приложения:



Текст программы.

```
index.html X
src > index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Lab7</title>
6   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css">
7 </head>
8 <body>
9 <div id="root"></div>
10 <script src="main.js" type="module"></script>
11
12 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"></script>
13 </body>
14 </html>
```

```
JS main.js X
src > JS main.js > ...
1 import {MainPage} from "../pages/main/MainPage.js";
2
3 const root = document.getElementById('root');
4
5 const mainPage = new MainPage(root)
6 mainPage.render()
```

MainPage.js:

```
import {StockCardComponent} from "../../components/stock-card/StockCardComponents.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))
        })
    }
}

```

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

  const data = await this.getData()
  const stock = new StockComponent(this.page)
  stock.render(data.data)
}
}

```

```

JS urls.js  X
src > modules > JS urls.js > [0] urls
1  class Urls {
2    constructor() {
3      this.url = 'http://127.0.0.1:8000/';
4    }
5
6    stocks() {
7      return `${this.url}films/`
8    }
9
10   stock(id) {
11     return `${this.url}films/${id}/`
12   }
13 }
14
15 export const urls = new Urls()

```

```

JS ajax.js  X
src > modules > JS ajax.js > Ajax
1  class Ajax {
2    async get(url) {
3      const response = await fetch(url, { method: "GET" });
4      const responseData = await response.json();
5      return { status: response.status, data: responseData };
6    }
7  }
8
9  export const ajax = new Ajax();

```

```

JS index.js X
server > JS index.js > ...
1  'use strict';
2
3  const express = require('express');
4  const path = require('path');
5  const app = express();
6  const htmlPath = path.resolve(__dirname, '..', 'src/index.html')
7  const staticPath = path.resolve(__dirname, '..', 'src')
8  const port = 2000;
9
10 app.use('/', express.static(staticPath));
11
12 app.all('*', (req, res) => {
13   res.sendFile(htmlPath);
14 });
15
16 app.listen(port, () => {
17   console.log(`Server listening http://localhost:${port}`);
18 });

```

```

JS StockCardComponents.js X
src > components > stock-card > JS StockCardComponents.js > StockCardComponent
1  export class StockCardComponent {
2    constructor(parent) {
3      this.parent = parent;
4    }
5    getHTML(data) {
6      return (
7
8        <div class="card m-3" style="width: 300px;">
9          
11            <h5 class="card-title">${data.title_film}</h5>
12            <p class="card-text">Год: ${data.year}</p>
13            <button class="btn btn-primary" id="click-card-${data.id}" data-id="${data.id}">Нажми на меня</button>
14          </div>
15        </div>
16      )
17    }
18    addListeners(data, listener) {
19      document
20        .getElementById(`click-card-${data.id}`)
21        .addEventListener("click", listener)
22    }
23
24    render(data, listener) {
25      const html = this.getHTML(data)
26      this.parent.insertAdjacentHTML('beforeend', html)
27      this.addListeners(data, listener)
28    }
29  }
30

```

```

JS StockComponent.js X
src > components > stock > JS StockComponent.js > StockComponent > getHTML
1  export class StockComponent {
2    constructor(parent) {
3      this.parent = parent
4    }
5
6    getHTML(data) {
7      return (
8
9        <div class="card m-3" style="width: 540px;">
10          <div class="row g-0">
11            <div class="col-md-8">
12              <div class="card-body">
13                <h5 class="card-title">${data.title_film}</h5>
14                <p class="card-text">Год: ${data.year} г.</p>
15                <p class="card-text">Продолжительность: ${data.duration} мин.</p>
16              </div>
17            </div>
18          </div>
19        </div>
20      )
21    }
22
23    render(data) {
24      const html = this.getHTML(data)
25      this.parent.insertAdjacentHTML('beforeend', html)
26    }
27  }
28
29

```

```

JS BackButtonComponent.js
src > components > back-button > JS BackButtonComponent.js > BackButtonComponent
1  export class BackButtonComponent {
2      constructor(parent) {
3          this.parent = parent;
4      }
5
6      addListeners(listener) {
7          document
8              .getElementById("back-button")
9              .addEventListener("click", listener)
10     }
11
12     getHTML() {
13         return (
14             <button id="back-button" class="btn btn-primary" type="button">Назад</button>
15         )
16     }
17
18     render(listener) {
19         const html = this.getHTML()
20         this.parent.insertAdjacentHTML('beforeend', html)
21         this.addListeners(listener)
22     }
23 }
24
25

```

Результат работы веб-приложения.

