Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



**Отчет**

**Лабораторная работа № 8**

**По курсу «Разработка интернет приложений»**

**«Создание одностраничного веб-приложения с использованием библиотеки Rect»**

**ИСПОЛНИТЕЛЬ:**

Пученков Дмитрий

Группа ИУ5-51Б

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2019 г.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2019 г.

Москва 2019

1. **Задание лабораторной работы**В этой лабораторной работе вы создадите React-приложение. Вам нужно будет создать одну страницу со списком репозиториев на гитхабе.  
   Требования к лабораторной работе:

* Настроенный конфиг webpack и babel
* React-приложение с одной страницей.
* На странице должны отображаться репозитории с github. Репозитории должны загружаться с помощью axios через открытый api гитхаба.
* Приложение должно успешно собираться вебпаком
* Приложение должно содержать css
* Приложение в dev режиме должно обновляться без перезагрузки страницы с помощью webpack-dev-server’а
* index.html должен добавляться с помощью HtmlWebpackPlugin

1. **package.json**  
   }
2. **webpack.config.js  
     
   const *path*** = *require*(**"path"**);  
   **const** HtmlWebpackPlugin = *require*(**"html-webpack-plugin"**);  
     
   module.**exports** = {  
    **entry**: **"./src/index.js"**,  
    **output**: {  
    **path**: ***path***.*join*(\_\_dirname, **"/dist"**),  
    **filename**: **"index-bundle.js"**  
   },  
    **module**: {  
    **rules**: [  
    {  
    **test**: /\.js$/,  
    **exclude**: /node\_modules/,  
    **use**: [**"babel-loader"**]  
    },  
    {  
    **test**: /\.css$/,  
    **use**: [**"style-loader"**, **"css-loader"**]  
    }  
    ]  
    },  
    **plugins**: [  
    **new** HtmlWebpackPlugin({  
    **template**: **"./src/index.html"**  
   })  
    ],  
    **resolve**: {  
    **alias**: {  
    **components**: ***path***.*resolve*(\_\_dirname, **'src/components/'**)  
    }  
    }  
   };
3. **.babelrc**{  
    **"presets"**: [**"@babel/preset-env"**, **"@babel/preset-react"**],  
    **"plugins"**: [**"@babel/plugin-proposal-export-default-from"**, **"@babel/plugin-proposal-class-properties"**]  
   }
4. **index.html**<!DOCTYPE **html**>  
   <**html lang="en"**>  
     
   <**head**>  
    <**meta charset="UTF-8"**>  
    <**meta name="viewport" content="width=device-width, initial-scale=1.0"**>  
    <**meta http-equiv="X-UA-Compatible" content="ie=edge"**>  
    <**title**>React Boilerplate</**title**>  
   </**head**>  
     
   <**body**>  
    <**div id="root"**>  
     
    </**div**>  
   </**body**>  
     
   </**html**>
5. **index.js  
   import React from "react"**;  
   **import ReactDOM from "react-dom"**;  
   **import** App **from "./App"**;  
     
   ReactDOM.render(<**App** />, ***document***.getElementById(**"root"**));
6. **App.js  
   import React**, { Component } **from "react"**;  
   **import** UserRepositories **from "components/UserRepositories"**;  
     
   **class** App **extends** Component {  
     
    render() {  
    **return** (  
    <**div**>  
    <**UserRepositories userName="crueltycute"** />  
    </**div**>  
    );  
    }  
   }  
     
   **export default** App;
7. **ListRender.js  
   import React**, { Component } **from "react"**;  
   **import './style.css'**;  
     
   **class** ListRender **extends** Component {  
     
    render() {  
    **const** {item} = **this**.**props**;  
    **return** (  
    <**li**>{item.**value**}</**li**>  
    );  
    }  
   }  
     
   **export default** ListRender;
8. **UserRepositories.js  
   import React**, { Component } **from "react"**;  
   **import *axios* from 'axios'**;  
   **import** ListRender **from "components/ListRender"**;  
     
   **class** UserRepositories **extends** Component {  
     
    **state** = {  
    **repNames**: []  
    }  
     
    componentDidMount() {  
    **let** url = **"https://api.github.com/users/"** + **this**.**props**.**userName** + **"/repos"**;  
    ***axios***.get(url).then(responce => {  
    **let** repNames = responce.**data**.map(item => {  
    **return** {**id**: item.**id**, **value**: item.**name**}  
    });  
    **this**.setState({repNames});  
    })  
    }  
     
    render() {  
     
    **return** (  
    <**div**>  
    <**h1**>{**this**.**props**.**userName**} repositories:</**h1**>  
    <**ul**>  
    {**this**.**state**.**repNames**.map(item =>  
    <**ListRender key=**{item.**id**} **item=**{item}/>  
    )}  
    </**ul**>  
    </**div**>  
    );  
    }  
   }  
     
   **export default** UserRepositories;
9. **style.css in listrender  
   li** {  
    **color**: **black**;  
    **font-size**: 20**px**;  
   }