**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка Интернет-Приложений»

Отчет по лабораторной работе №8

«Разработка пользовательского интерфейса с использованием библиотеки React.»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-53 |  | Юрий Евгеньевич Гапанюк |
| Миронова Александра |  |  |
| Подпись и дата:  19.01.22 |  | Подпись и дата: |

Москва, 2022 г.

**Цель лабораторной работы:** изучение возможностей создания пользовательского интерфейса в веб-приложениях с использованием библиотеки React.

**Задание:**

На основе методических указаний разработайте React-приложение. Для создания приложения необходимо решить следующие задачи:

1. Создать стартовый React-проект. Удалить неиспользуемый код. Организовать директории для страниц, компонентов, утилит и работы с сетью.
2. Организовать роутниг в веб-приложении.
3. Разработать базовые страницы, на которых будут отображаться сущности из выбранной вами предметной области:
   1. Стартовая страница.
   2. Страница просмотра списка объектов.
   3. Страница просмотра конкретного объекта.
4. Вынести переиспользуемые компоненты в отдельные файлы:
   1. Для навигации по приложению можно добавить header.
   2. Для отображения дополнительной информации (данные о студенте и предметной области) можно использовать footer.
   3. Источники ввода-вывода (поля ввода (inputs)/формы/текстовые блоки).
   4. Переиспользуемые таблицы/гриды.
5. Добавить асинхронные запросы в разработанный API, чтобы страница получала данные с сервера.
6. Если в Вашем проекте реализована сложная логика работы с состоянием приложения, то рекомендуется добавить пользовательские хуки.
7. Страницы приложения должны хорошо отображаться как на больших, так и на маленьких экранах.

**Подготовка к выполнению программы:**

С помощью открытого апи GitHub https://docs.github.com/en , реализовано:

SPA приложение на react с поисковой строкой для поиска аккаунтов.

При введении корректного имени пользователя и нажатии на кнопку enter или же на поиск попадаем на страницу с данными об этом пользователе

Кнопка назад которая вернет нас на страницу поиска

Если ввели не существующий аккаунт пользователя обрабатываем ошибку.

**Выполнение задания лабораторной работы**

**Текст программы:**

**APP.js**

import React from 'react';

import './App.css';

import ProjectPage from './pages/project/index.js';

import HomePage from './pages/home/index.js';

import NotFound from './pages/home/notfound.js';

import { Switch, Route} from 'react-router-dom';

function App () {

return (

<div className="App">

<header className="App-header">

Github search users

</header>

<Switch>

<Route path='/lab9/build/project' component={ProjectPage} />

<Route exact path='/lab9/build/' component={HomePage} />

<Route component={NotFound} />

</Switch>

</div>

);

};

export default App;

**APP.css**

.App {

display: flex;

flex-direction: column;

align-items: center;

text-align: center;

background-color: #282c34;

min-height: 100vh;

}

.App-header {

background-color: #025483;

height: 100px;

width: 70vh;

min-width: 200px;

display: flex;

flex-direction: column;

border-radius: 0 0 30px 30px;

align-items: center;

justify-content: center;

font-size: calc(10px + 2vmin);

color: white;

}

@media screen and (max-device-width: 600px) {

.App-header {

width: 400px;

height: 100px;

font-size: 20px;

}

}

**index.js\home**

import React from 'react';

import './home.css';

import { useHistory } from 'react-router-dom';

function HomePage() {

console.log("HomePage");

const history = useHistory();

function enter\_todo(ev) { // ввод записи

if(ev.key === 'Enter') {

localStorage.setItem('login', ev.target.value);

history.push('/lab9/build/project')

}

}

function to\_project() {

let value = document.getElementById('input\_user');

console.log(value.value);

localStorage.setItem('login', value.value);

console.log("to\_project");

console.log(localStorage.login);

history.push('/lab9/build/project');

}

return (

<div className="home">

<input className="userame\_input" id="input\_user" placeholder="write here username..." onKeyDown={(ev)=>enter\_todo(ev)} />

<button onClick={()=>to\_project()}> Ok </button>

</div>

);

}

export default HomePage;

**home.css**

.home {

text-align: center;

display: flex;

flex-direction: row;

justify-content: center;

align-items: center;

}

.home-header {

background-color: #282c34;

height: 100px;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: calc(10px + 2vmin);

color: white;

}

.userame\_input {

padding-left: 10px;

display: flex;

flex-direction: row;

align-items: center;

width: 300px;

height: 42px;

margin-right: 0px;

margin: 20px;

border-radius: 5px;

font-size: 17px;

}

**index.js\project**

import React from 'react';

import './project.css';

import { useEffect } from 'react';

import { useHistory } from 'react-router-dom';

function ProjectPage() {

console.log("ProjectPage");

const history = useHistory();

const key = 'ghp\_crkqAwaGYxSOg36pKd4js7ccNeeZHu09SVQX';

function PutUserInfo(json) {

if (json === "" || json === undefined){

console.log("error username. stay default");

}

else{

console.log("the PutUserInfo func");

console.log(json);

document.getElementById("login").innerHTML = "login: " + json.items[0].login;

document.getElementById("id").innerHTML = "id: " + json.items[0].id;

document.getElementById("avatar").innerHTML = `<img height=100% width=100% src=${json.items[0].avatar\_url} alt=${json.items[0].login}/>`;

document.getElementById("node\_id").innerHTML = "node\_id: " + json.items[0].node\_id;

document.getElementById("gravatar\_id").innerHTML = "gravatar\_id: " + json.items[0].gravatar\_id;

document.getElementById("url").innerHTML = "url: " + json.items[0].url;

document.getElementById("html\_url").innerHTML = "html\_url: " + json.items[0].html\_url;

document.getElementById("followers\_url").innerHTML = "followers\_url: " + json.items[0].followers\_url;

document.getElementById("following\_url").innerHTML = "following\_url: " + json.items[0].following\_url;

document.getElementById("gists\_url").innerHTML = "gists\_url: " + json.items[0].gists\_url;

document.getElementById("starred\_url").innerHTML = "starred\_url: " + json.items[0].starred\_url;

document.getElementById("subscriptions\_url").innerHTML = "subscriptions\_url: " + json.items[0].subscriptions\_url;

document.getElementById("organizations\_url").innerHTML = "organizations\_url: " + json.items[0].organizations\_url;

document.getElementById("repos\_url").innerHTML = "repos\_url: " + json.items[0].repos\_url;

document.getElementById("events\_url").innerHTML = "events\_url: " + json.items[0].events\_url;

document.getElementById("received\_events\_url").innerHTML = "received\_events\_url: " + json.items[0].received\_events\_url;

document.getElementById("type").innerHTML = "type: " + json.items[0].type;

document.getElementById("site\_admin").innerHTML = "site\_admin: " + json.items[0].site\_admin;

document.getElementById("score").innerHTML = "score: " + json.items[0].score;

console.log("the PutUserInfo func yet");

}

};

function back\_to\_home() {

history.push('/lab9/build/');

}

useEffect(() => { // searching

let checkevent = localStorage.login;

console.log(localStorage.login);

let url = new URL(`https://api.github.com/search/users?q=${checkevent}`);

fetch(url, {

Authorization: 'Token' + key

}) // получение данных

.then(res => res.json())

.then(res => {

console.log(res);

if (res.message !== "Not Found"){

console.log('error');

PutUserInfo(res);

}

}, [])

.catch(er=>{

console.log('error');

document.getElementById("login").innerHTML ="404 user not found";

});

});

return (

<div className="project">

<header className="project-header">

<button onClick={()=>back\_to\_home()}> Home</button>

</header>

<div className="user-info">

<div className="pict-box">

<div className="icon-info" id="avatar"></div>

</div>

<div className="text-info" id="login"></div>

<div className="text-info" id="id"></div>

<div className="text-info" id="node\_id"></div>

<div className="text-info" id="gravatar\_id"></div>

<div className="text-info" id="url"></div>

<div className="text-info" id="html\_url"></div>

<div className="text-info" id="followers\_url"></div>

<div className="text-info" id="following\_url"></div>

<div className="text-info" id="gists\_url"></div>

<div className="text-info" id="starred\_url"></div>

<div className="text-info" id="subscriptions\_url"></div>

<div className="text-info" id="organizations\_url"></div>

<div className="text-info" id="repos\_url"></div>

<div className="text-info" id="events\_url"></div>

<div className="text-info" id="received\_events\_url"></div>

<div className="text-info" id="type"></div>

<div className="text-info" id="site\_admin"></div>

<div className="text-info" id="score"></div>

</div>

</div>

);

}

export default ProjectPage;

**project.css**

.project {

color: white;

}

.project-header {

background-color: #282c34;

height: 100px;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: 24px;

text-align: center;

}

.user-info {

display: flex;

flex-direction: column;

align-items: flex-start;

text-align: left;

background-color: #335070;

font-size: 14px;

min-height: 60vh;

min-width: 40vh;

padding: 20px;

margin: 10px;

border: 1px solid white;

}

@media screen and (max-device-width: 600px) {

.user-info {

width: 400px;

}

}

li {

text-align: center;

background-color: #dbf2f8;

font-size: calc(10px + 1vmin);

color: white;

margin: 17px;

width: 100px;

height: 42px;

border: 1px solid;

border-radius: 30px;

}

Link {

color: white;

}

.text-info {

font-size: 14px;

margin: 10px;

}

.icon-info {

height: 100%;

width: 100%;

}

.pict-box{

margin: 10px;

height: 100px;

width: 100px;

}

button {

width: 100px;

height: 42px;

color: #031d3a;

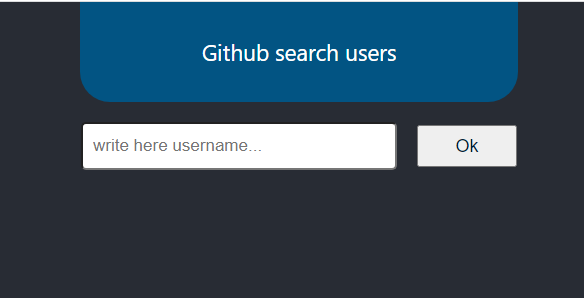
font-size: 18px;

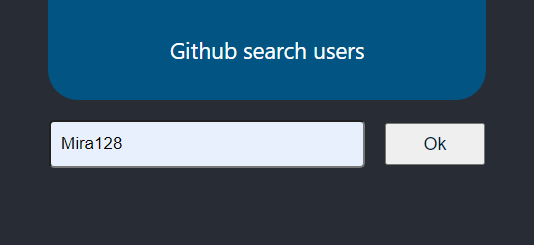
margin: 20px;

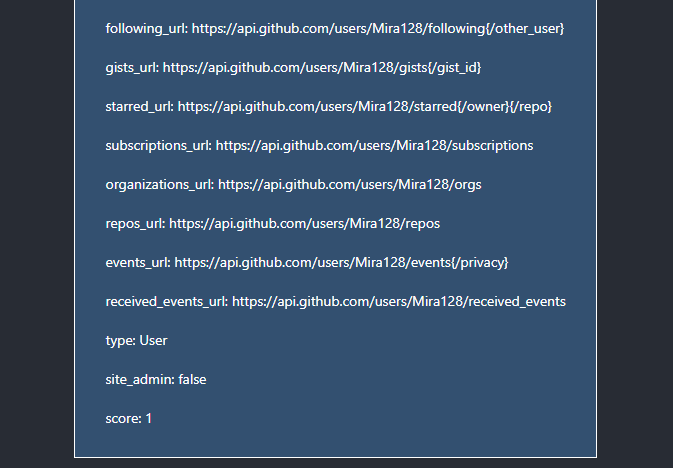
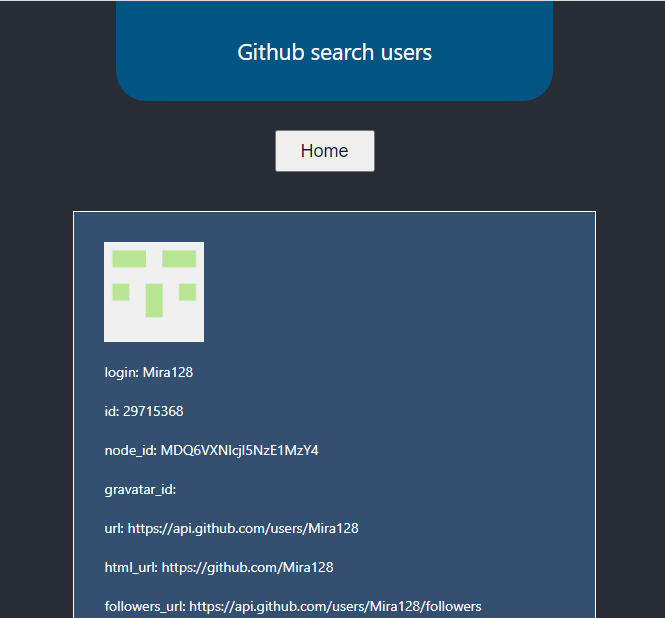
margin-left: 0px;

}

**Экранные формы с примерами выполнения программы:**

****

****

****