

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

Факультет «Информатика и системы управления»

Отчёт по лабораторной работе №8
по курсу «Разработка интернет-приложений»

Выполнил:

студент группы ИУ5-51Б
Головацкий А. Д.

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю. Е.

Подпись и дата:
14.12.2021

Подпись и дата:

Москва, 2021 г.

Описание задания:

С помощью открытого api GitHub “<https://docs.github.com/en>”, реализовать:

1. SPA приложение с поисковой строкой для поиска аккаунтов
2. При введении корректного имени пользователя и нажатии на кнопку enter или же на поиск попадаем на страницу с данными об этом пользователе
3. Кнопка назад которая вернет нас на страницу поиска
4. Если ввели не существующий аккаунт пользователя обрабатываем нормально ошибку.

Требования к дизайну:

1. Красивый user friendly дизайн
2. Реализовать верстку для всех типов устройств

Требования к технологиям:

1. Все реализовать на react
2. Развернуть все на vercel

Текст программы:

src/App.jsx:

```
import './App.css';
import Routes from './Routes';

function App() {
  return (
    <div className="App">
      <Routes />
    </div>
  );
}

export default App;
```

src/Routes.jsx:

```
import React from 'react';
import { BrowserRouter, Route, Switch } from 'react-router-dom';
import Finder from './components/Finder/Finder'
import UserInfo from './components/UserInfo/UserInfo'
import UserNotFound from './components/UserNotFound/UserNotFound'

const Routes = () => (
  <BrowserRouter basename="/lab9/build/index.html">
    <Switch>
      <Route exact path="/" component={Finder} />
      <Route exact path="/userInfo/:user" component={UserInfo} />
      <Route exact path="/userNotFound" component={UserNotFound} />
    </Switch>
  </BrowserRouter>
);

export default Routes;
```

src/utls.js:

```
export default function normalizeUrl(url) {  
  if ((url.split(''))[url.length - 1] === '/')  
    return (url.slice(0, -1));  
  return (url);  
}
```

src/App.css:

```
.App {  
  min-width: 320px;  
  padding: 20px;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  flex-direction: column;  
}
```

src/components/Finder/Finder.jsx:

```
import React from 'react'  
import {useRouteMatch, useHistory} from 'react-router-dom';  
import './Finder.css'  
import normalizeUrl from '../../utls'  
  
function Finder() {  
  const history = useHistory();  
  const match = useRouteMatch();  
  const input = React.useRef(null);  
  const cur_url = normalizeUrl(match.url);  
  
  const findUser = () => {  
    if (input.current.value.trim().length > 0)  
      history.push(`${cur_url}/userInfo/${input.current.value.trim()}`);  
  };  
  
  const enterPress = (e) => {  
    if(e.key === 'Enter')  
      findUser();  
  };  
  
  return (  
    <div className="finder">  
      <span className="finder-header">Finder</span>  
      <div className="finderBox">  
        <span className="finderBox-text">Enter github login:</span>  
        <input className="finderBox-input" type="text" ref={input}  
onKeyPress={enterPress}></input>  
        <button className="finderBox-btn" onClick={findUser}>Find</button>  
      </div>  
    </div>  
  );  
}  
  
export default Finder;
```

src/components/Finder/Finder.css:

```
@font-face {  
  font-family: Wave-Regular;  
  src: url(../../fonts/Wave-Regular.ttf);  
}  
  
.finder {  
  color: rgb(161, 83, 172);  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;
```

```

    width: 100%;
}

.finderBox {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    margin: 100px 0px 0px 0px;
    border: 2px solid rgb(76, 89, 145);
    border-radius: 10px;
    padding: 10px;
    width: 60%;
    height: 200px;
}

.finderBox-text {
    font-size: 30px;
    font-family: sans-serif;
}

.finderBox-input {
    margin-top: 20px;
    border: none;
    outline: none;
    border: 1px solid white;
    border-radius: 5px;
    height: 40px;
    width: 50%;
    background-color: rgb(19, 23, 27);
    font-size: 20px;
    color: white;
}

.finderBox-input:hover, .finderBox-input:focus {
    border: 1px solid rgb(76, 89, 145);
}

.finderBox-btn {
    margin-top: 20px;
    width: 150px;
    height: 50px;
    color: rgb(161, 83, 172);
    font-size: 30px;
    background-color: rgb(32, 36, 39);
}

.finder-header {
    font-size: 60px;
    font-family: Wave-Regular;
}

@media screen and (max-width: 500px) {
    .finderBox-text {
        font-size: 25px;
    }
    .finderBox {
        width: 80%;
        height: 200px;
    }
    .finderBox-input {
        width: 80%;
    }
}

```

src/components/ReturnBtn/ReturnBtn.jsx:

```
import React from 'react'
import './ReturnBtn.css'

function ReturnBtn( {history, url} ) {
  const cur_url = url.slice(0, url.split('/').lastIndexOf('/'));
  const goBack = () => {
    history.push(`/${cur_url}/`);
  }
  return (
    <button className="returnBtn" onClick={goBack}>
      Go Back
    </button>
  );
}

export default ReturnBtn;
```

src/components/ReturnBtn/ReturnBtn.css:

```
.returnBtn {
  margin-top: 30px;
  width: 150px;
  height: 50px;
  color:rgb(161, 83, 172);
  font-size: 30px;
  background-color: rgb(32, 36, 39);
}
```

src/components/UserInfo/UserInfo.jsx:

```
import React, { useEffect, useState } from 'react'
import { useHistory, useParams, useRouteMatch } from 'react-router';
import './UserInfo.css'
import normalizeUrl from '../../utils'
import ReturnBtn from '../../ReturnBtn/ReturnBtn'
import loading from '../../images/Rainbow.gif'

function UserInfo() {
  const history = useHistory();
  const match = useRouteMatch();
  const userName = useParams();
  const cur_url = normalizeUrl(match.url);
  const [user, setUser] = useState();
  useEffect(() => {
    fetch(`https://api.github.com/users/${userName.user}`)
      .then((data) => {
        if(data.status === 404) {
          history.push(`/userNotFound`);
          return null;
        }
        return (data.json());
      })
      .then((data) => {
        console.log(data);
        if (data)
          setUser(data);
      })
  }, [cur_url, history, userName])
  return user ? (
    <div className="container">
      <span className="container-header">Finder</span>
      <div className="container-content">
        <div className="container-content-mainInfo">
          <img className="container-content-mainInfo-avatar"
            src={user.avatar_url} alt="avatar"/>
          <div className="container-content-mainInfo-text">
```

```

        <a href={user.html_url} className="container-content-
mainInfo-text-login">{user.login}</a>
        <span className="container-content-mainInfo-text-
name">{user.name}</span>
        <span className="container-content-mainInfo-text-name">id:
{user.id}</span>
    </div>
</div>
<div className="container-content-additionalInfo">
    <span className="container-content-additionalInfo-
header">Info:</span>
    <span className="container-content-additionalInfo-
info">Followers: {user.followers}</span>
    <span className="container-content-additionalInfo-
info">Following: {user.following}</span>
    <span className="container-content-additionalInfo-info">Created
at: {user.created_at.replace('T', ' ').replace('Z', ' ')}</span>
    <span className="container-content-additionalInfo-info">Public
repos: {user.public_repos}</span>
    <span className="container-content-additionalInfo-info">Type:
{user.type}</span>
</div>
</div>
<ReturnBtn history={history} url={cur_url.slice(0,
cur_url.split('').lastIndexOf('/'))} />
</div>
) : (
    <div className="container">
        <div className="container-header">
            Loading...
        </div>
        <img className="loadingGif" src={loading} alt="loading" />
    </div>
);
};

export default UserInfo;

```

src/components/UserInfo/UserInfo.css:

```

.container {
    padding: 0px;
    color: rgb(161, 83, 172);
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    width: 100%;
}

.container-header {
    font-size: 60px;
    color: rgb(161, 83, 172);
    font-family: Wave-Regular;
}

.container-content {
    border: 2px solid rgb(76, 89, 145);
    border-radius: 10px;
    padding: 20px;
    margin-top: 50px;
    width: 50%;
    max-height: 620px;
}

.container-content-mainInfo {
    display: flex;
    width: 100%;
}

```

```
}

.container-content-additionalInfo {
  display: flex;
  width: 100%;
  flex-direction: column;
}

.container-content-additionalInfo-header {
  color: white;
  font-weight: 500;
  margin-top: 20px;
  font-size: 40px;
}

.container-content-additionalInfo-info {
  color: rgb(218, 214, 214);
  font-size: 30px;
  margin-left: 80px;
  margin-top: 5px;
}

.container-content-mainInfo-text {
  display: flex;
  flex-direction: column;
  justify-content: flex-start;
  margin-left: 60px;
  margin-top: 10px;
  color: white;
}

.container-content-mainInfo-text-login {
  font-size: 50px;
  text-decoration: none;
  color: rgb(161, 83, 172);
}

.container-content-mainInfo-text-login:hover {
  font-size: 50px;
  text-decoration: none;
  color: white;
}

.container-content-mainInfo-text-name {
  font-size: 30px;
  color: rgb(131, 125, 125);
}

.container-content-mainInfo-avatar {
  margin: 10px;
  width: 40%;
  max-width: 300px;
  min-width: 150px;
  height: 40%;
  border-radius: 50%;
}

.loadingGif {
  width: 100px;
  height: 100px;
  margin-top: 150px;
}

@media screen and (max-width: 1000px) {
  .container-content {
    width: 70%;
  }
}
```

```

@media screen and (max-width: 600px) {
  .container-content {
    width: 80%;
  }

  .container-content-mainInfo-text {
    margin-left: 40px;
  }

  .container-content-mainInfo-text-login {
    font-size: 40px;
  }

  .container-content-mainInfo-text-login:hover {
    font-size: 40px;
    text-decoration: none;
    color: white;
  }

  .container-content-mainInfo-text-name {
    font-size: 20px;
    color:rgb(131, 125, 125);
  }

  .container-content-additionalInfo-header {
    color: white;
    font-weight: 500;
    margin-top: 20px;
    font-size: 30px;
  }

  .container-content-additionalInfo-info {
    color: rgb(218, 214, 214);
    font-size: 20px;
    margin-left: 80px;
    margin-top: 5px;
  }
}

@media screen and (max-width: 600px) {
  .container-content-mainInfo-avatar {
    min-width: 100px;
  }

  .container-content {
    width: 90%;
  }
}

```

src/components/UserNotFound/UserNotFound.jsx:

```

import './UserNotFound.css'
import ReturnBtn from '../ReturnBtn/ReturnBtn'
import { useHistory, useRouteMatch } from 'react-router';
import normalizeUrl from '../../utils'

function UserNotFound() {
  const history = useHistory();
  const match = useRouteMatch();
  const cur_url = normalizeUrl(match.url);
  return(
    <div className="page">
      <span className="page-header">Finder</span>
      <div className="page-content">
        User does not exist =<br></br>      Please try again
      </div>
      <ReturnBtn history={history} url={cur_url} />
    </div>
  )
}

```



```

    </div>
  );
}

export default UserNotFound;

```

src/components/UserNotFound/UserNotFound.css:

```

@font-face {
  font-family: Wave-Regular;
  src: url(../../fonts/Wave-Regular.ttf);
}

.page {
  padding: 0px;
  color: rgb(161, 83, 172);
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  width: 100%;
}

.page-header {
  font-size: 60px;
  font-family: Wave-Regular;
}

.page-content {
  margin-top: 100px;
  font-size: 50px;
  color: white;
  text-align: center;
}

@media screen and (max-width: 530px) {
  .page-content {
    margin-top: 100px;
    font-size: 40px;
    color: white;
    text-align: center;
  }
}

```

src/index.css:

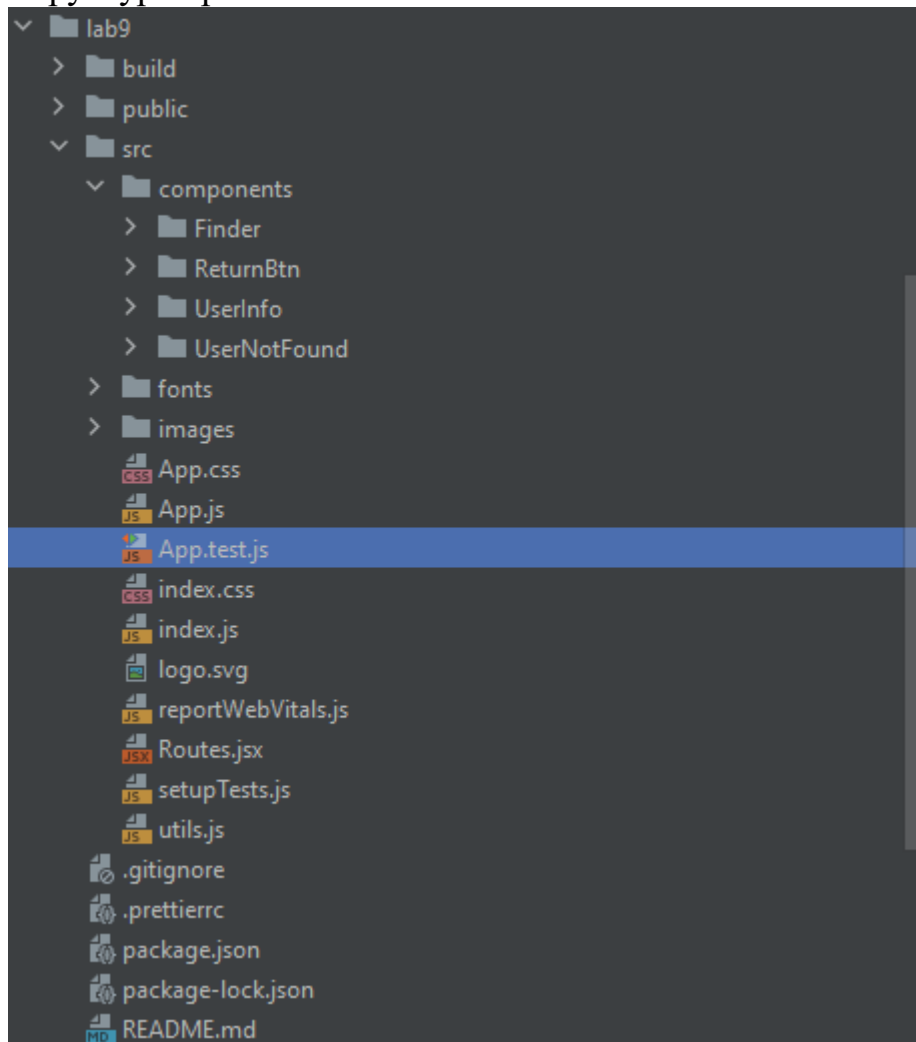
```

body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  background-color: rgb(19, 23, 27);
}

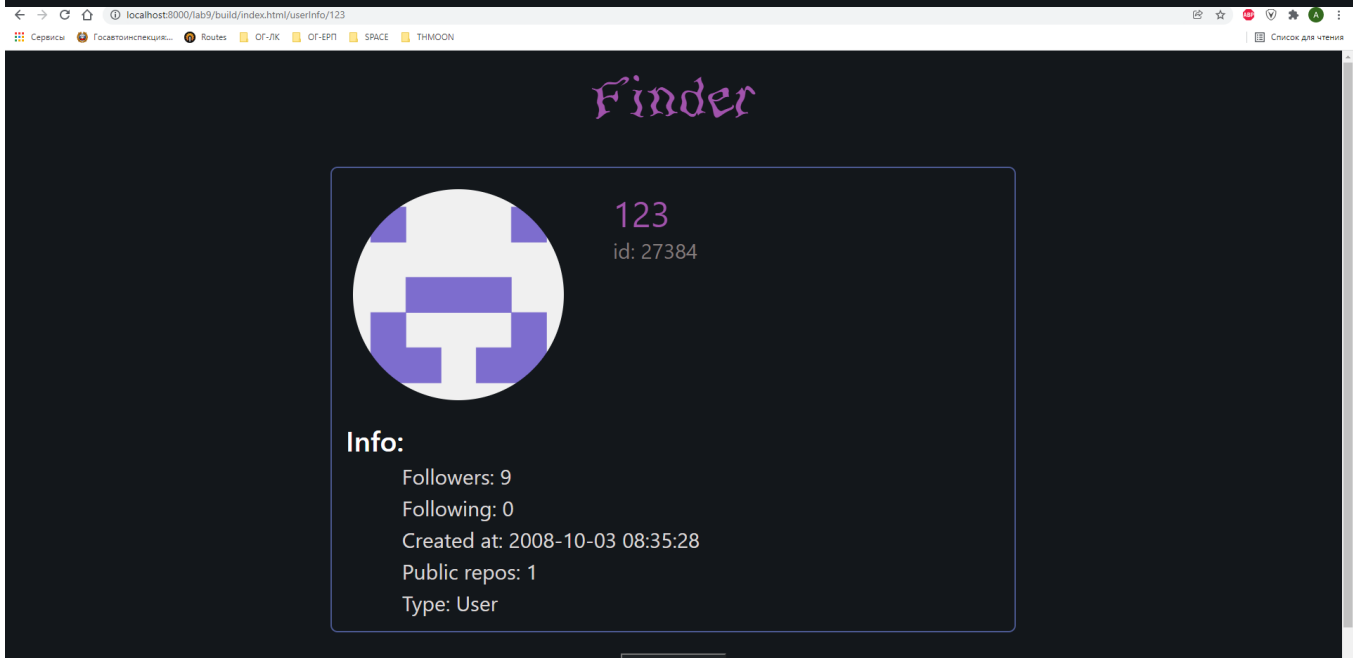
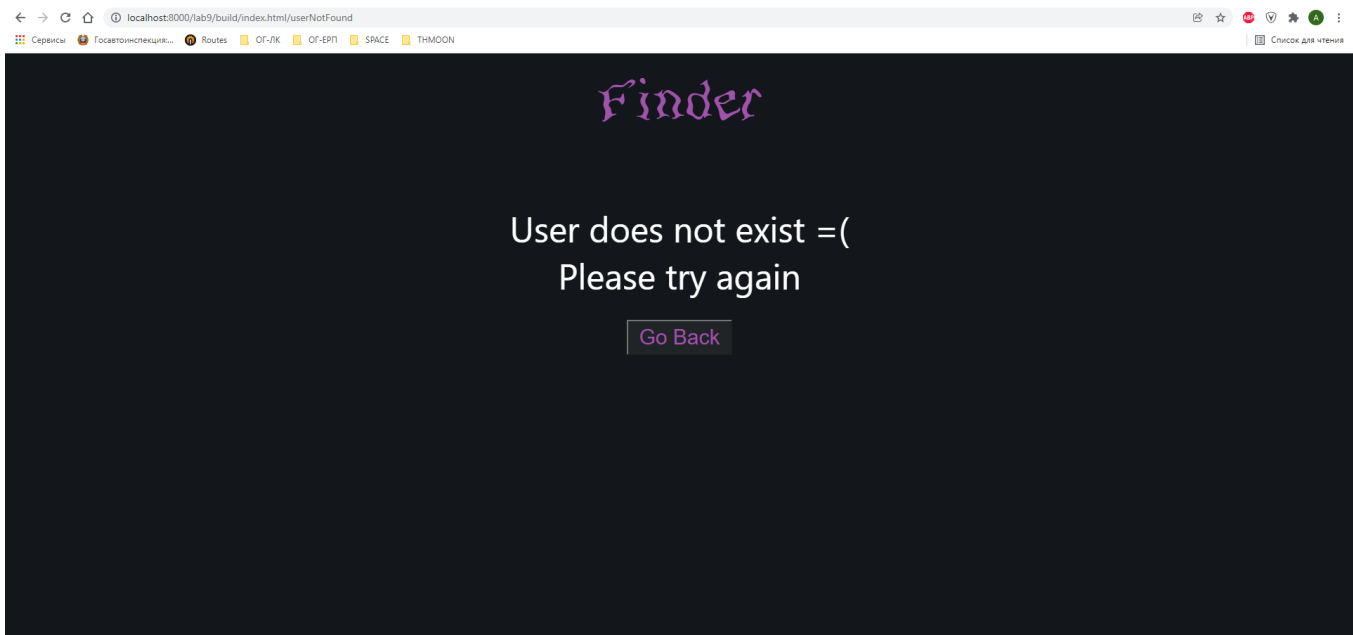
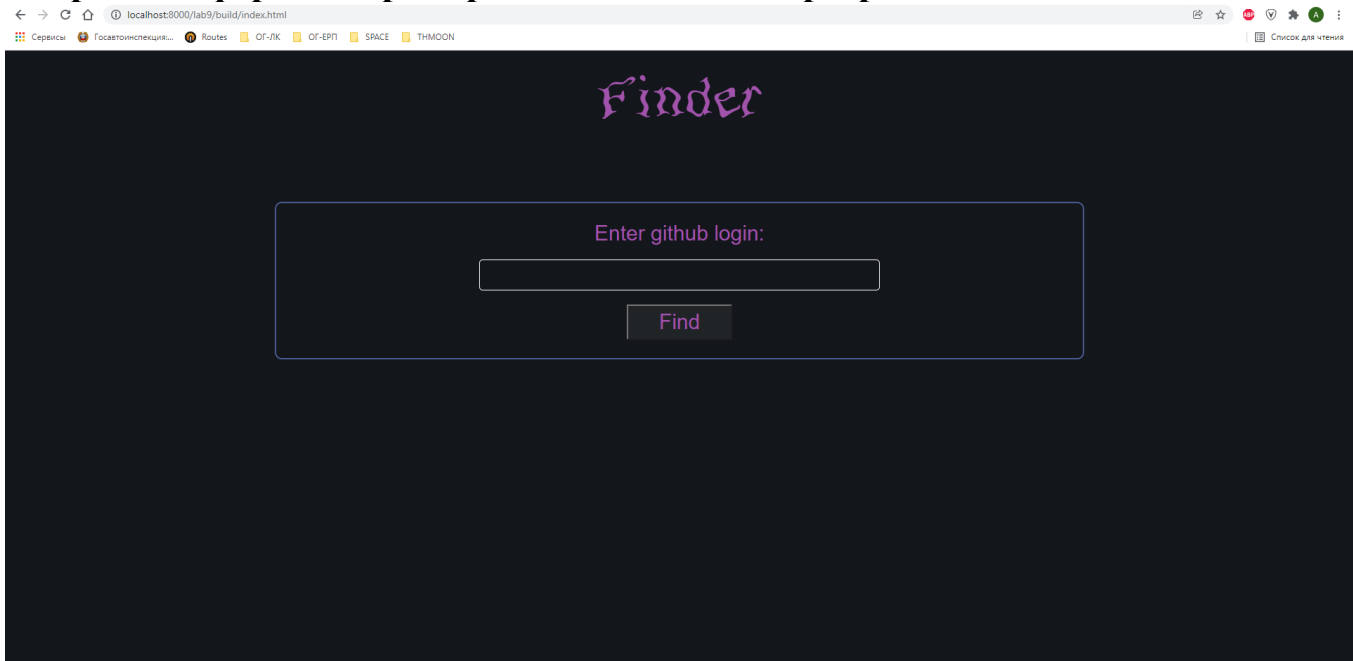
code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}

```

Структура проекта:



Экранные формы с примерами выполнения программы:



Finder



TheLavv

Andrey Golovatskiy

id: 56406879

Info:

Followers: 0

Following: 0

Created at: 2019-10-10 16:59:14

Public repos: 11

Type: User

[Go Back](#)