



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)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

## Отчет по лабораторной работе №8

Студент Вардумян Арсен Тигранович  
*фамилия, имя, отчество*

Группа ИУ5-51Б

Студент 18.12.2021 Вардумян А.Т.  
*подпись, дата фамилия, и.о.*

Преподаватель 18.12.2021 Гапанюк Ю.Е.  
*подпись, дата фамилия, и.о.*

## 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 React, {useState} from 'react'
import {BrowserRouter as Router, Switch, Route, Redirect} from 'react-router-dom'
import HomePage from './components/HomePage'
import UserPage from './components/UserPage'

function App() {
  const [userName, setUserName] = useState("")

  return (
    <>
      <Router>
        <Switch>
          <Route path="/" exact>
            <HomePage onSearch={value => setUserName(value)}>
          </Route>
          <Route path="/user" exact>
            <UserPage userName={userName}>
          </Route>
          <Redirect to="/" />
        </Switch>
      </Router>
    </>
  )
}
```

```
}  
  
export default App;
```

“src/index.js”:

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import './index.css';  
import App from './App';  
import reportWebVitals from './reportWebVitals';  
  
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById('root')  
)  
;  
  
// If you want to start measuring performance in your app, pass a function  
// to log results (for example: reportWebVitals(console.log))  
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals  
reportWebVitals();
```

“src/components/HomePage.js”:

```
import {React, useState} from 'react'  
import { useHistory, useRouteMatch } from 'react-router'  
  
function HomePage({onSearch}) {  
  const [value, setValue] = useState("")  
  const history = useHistory()  
  
  function removeSlash(path) {  
    return path.charAt(path.length - 1) === "/" ? path.slice(0, -1) : path  
  }  
  
  const url = removeSlash(useRouteMatch().url)  
  
  function handleSubmit(value) {  
    onSearch(value)  
    history.push(`${url}/user`)  
  }  
  
  return (  
    <div className="container center-align">  
      <div className="input-field-container s12">  
        <div className="input-field">  
          <i className="material-icons prefix">account_circle</i>  
          <input
```

```

        id="icon_prefix"
        type="text"
        className="validate"
        onChange={(e) => setValue(e.target.value)}
        onKeyPress={(e) => {
            if(e.key === "Enter") {
                handleSubmit(value)
            }
        }} />
        <label htmlFor="icon_prefix">Github Account</label>
    </div>
</div>
    <button className="waves-effect waves-light btn-large" onClick={() =>
        handleSubmit(value)}>Search</button>
    </div>
)
}

export default HomePage

```

“src/components/UserCard.js”:

```

import {React} from 'react'

function UserCard({userData}) {
    function isNull(text) {
        return text && text
    }

    return (
        <div className="card col z-depth-4">
            <div className="waves-effect waves-block waves-light center-align">
                <img className="circle card-img" alt="user_foto" src={userData.avatar_url} />
                <div className="card-title">{userData.login}</div>
                <div className="card-title">{userData.updated_at}</div>
            </div>
            <div className="card-content col m7">
                <span className="card-title">{isNull(userData.name)}</span>
                <span className="card-title">{isNull(userData.location)}</span>
                <span className="card-title">{isNull(userData.followers)} followers</span>
                <span className="card-title">{isNull(userData.following)} following</span>
                <span className="card-title">Repositories: {isNull(userData.public_repos)}</span>
                <span className="card-title">Created: {userData.created_at}</span>
                <span className="card-title"><a href={userData.html_url}>{userData.html_url}</a></span>
            </div>
        </div>
    )
}

export default UserCard

```

“src/components/UserPage.js”:

```

import {React, useState, useEffect} from 'react'
import UserCard from './UserCard'

function UserPage({userName}) {
  const TOKEN = process.env.REACT_APP_API_KEY
  const [userData, setUserData] = useState({})
  const [error, setError] = useState(true)

  useEffect(async () => {
    if (userName) {
      const data = await fetch(
        `https://api.github.com/users/${userName}`
      )
      // {
      //   headers: {
      //     authorization: `token ${TOKEN}`
      //   }
      // }
      // })
      .then(res => {
        if (res.ok) {
          setError(false)
          return res.json()
        }
      })
      data && setUserData(data)
    }
  }, [userName])

  return error ? (
    <div>There is no such a github account...</div>
  ) : (
    <UserCard userData={userData} />
  )
}

export default UserPage

```

“public/index.html”:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
  <!--
  manifest.json provides metadata used when your web app is installed on a

```

```

    user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css"
    />
    <link
      href="https://fonts.googleapis.com/icon?family=Material+Icons"
      rel="stylesheet"
    />
    <!--
    Notice the use of %PUBLIC_URL% in the tags above.
    It will be replaced with the URL of the `public` folder during the build.
    Only files inside the `public` folder can be referenced from the HTML.

    Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
    work correctly both with client-side routing and a non-root public URL.
    Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.

    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn build`.
    -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
  </body>
</html>

```

“src/index.css”:

```

:root{
  --main-color: #8e24aa;
}

body, #root {
  width: 100vw;
  height: 100vh;
}

```

```
#root, .input-field-container {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-weight: 500 !important;
}

.card-img {
  width: 12rem;
  margin: 2rem auto;
}

.card-title {
  font-size: 22px !important;
  font-weight: 400 !important;
}

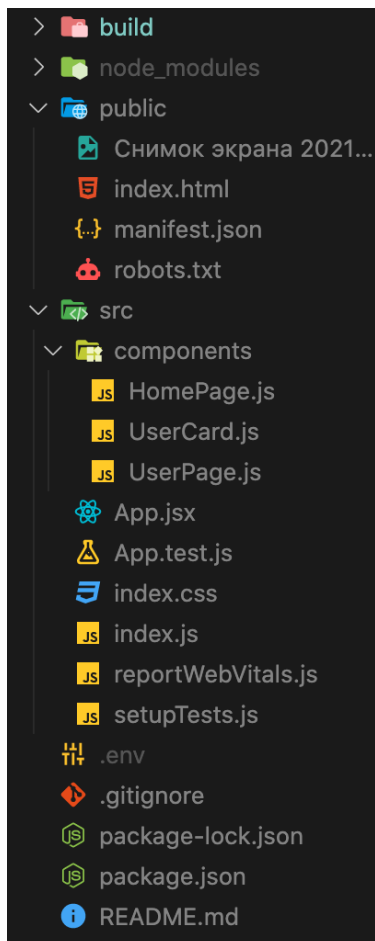
@media screen and (max-width:600px){
  .card-title {
    font-size: 18px !important;
  }

  .img-err {
    font-size: 1.2rem !important;
  }
}

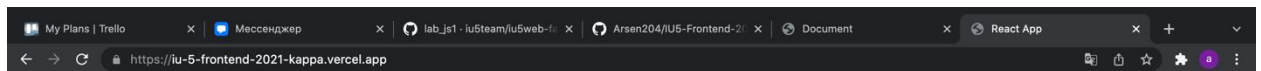
@media screen and (max-width:320px){
  .card-title {
    font-size: 16px !important;
    margin: 0 !important;
  }

  .img-err {
    font-size: 1rem !important;
  }
}
```

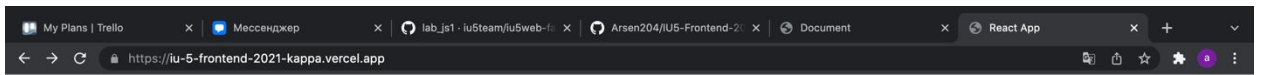
Структура проекта:



## Экранные формы с примерами выполнения программы:



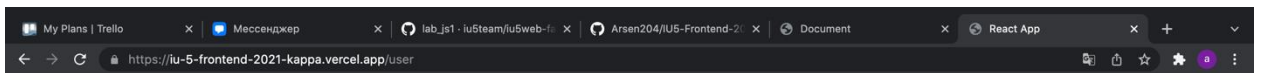





Github Account



SEARCH





Yerlan

2021-11-12T14:16:21Z

Yerlan Mangazbayev

Astana, Kazakhstan

3 followers

0 following

Repositories: 20

Created: 2013-03-20T01:21:23Z

<https://github.com/Yerlan>

My Plans | Trello x Мессенджер x lab.js1 · lu5team/lu5 x Arsen204/lu5-Front x Document x React App x Yerlan (Yerlan Manga x

https://github.com/Yerlan

Search or jump to... Pull requests Issues Marketplace Explore

Yerlan Mangazbayev

Yerlan

Follow

Software Engineer

3 followers · 0 following

Astana, Kazakhstan  
mangazbayev@gmail.com

Achievements

Block or Report

Overview

Repositories 20

Projects

Packages

Stars 1

Popular repositories

scrollreveal.js

Forked from jlmakes/scrollreveal

Easy scroll animations for web and mobile browsers.

JavaScript

Public

jquery.pin

Forked from webpop/jquery.pin

Pin any element within a container

HTML

Public

yerlan.github.com

HTML 1

Public

adidas\_weather\_api

Node Rest API App for Adidas

JavaScript

Public

adidas-wishlist

Adidas Software Engineer take-home challenge

JavaScript

Public

thisplace\_challenge

JavaScript

Public

24 contributions in the last year

	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Mon													
Wed													
Fri													

Learn how we count contributions