

Міністерство освіти і науки України

Національний технічний університет України „КПІ

імені Ігоря Сікорського ”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**ЗВІТ**

лабораторної работи №3

курсу «Основи WEB - технологій »

Тема: «:JavaScript. Формат JSON. API. Отримання даних із зовнішньої сторінки (fetch, promise)»

Перевірив: Виконав:

Доц. Голубєв Л. П. Гр. ІП-02

Трофимов Данило

Київ 2022

**Завдання 1.**

Створити js-скрипт для отримання інформації із зовнішньої сторінки <https://randomuser.me/api> за допомогою методів **fetch** та **promise** у форматі JSON. Розробити виведення інформації на сторінку у відповідності до свого варіанту:

Варіант 2. Picture, сell, city, country, postcode.

**Хід роботи:**

За допомогою текстового редактора VSCode, було створено структуру додаток з використанням бібліотеки React. Додаток має наступну структуру: index.js, app.ja, index.css, app.css. Мною було реалізовано компонент UserComponent.jsx та стилі до компоненту UserComponent.css, а також сервіс FetchUserService.js. Також було модифіковано App.js та App.css.

Розглянемо файл FetchUserService.js. В ньому описаний метод для отримання даних з API сервісу за допомогою бібліотеки Axios та її стандартного мметода Get.

Лістинг файлу FetchUserService.js

import axios from 'axios';

const BASE\_URL = 'https://randomuser.me/api';

const fetchRandomUser = async () => {

    try {

        const response = await axios.get(BASE\_URL);

        return response.data.results[0];

    } catch (error) {

        throw error;

    }

};

export default fetchRandomUser;

Тепер поглянемо на App.js, де ці дані отримуються та зберігаються в масив. Цей файл містить метод для отримання даних по кліку кнопки, та методом map ітерується по масиву та повертає масив компонентів UserComponent

Лістинг файлу App.js:

import React, { useState } from 'react';

import fetchRandomUser from './Services/FetchUserService';

import './App.css';

import UserComponent from './Components/UserComponent';

function App() {

  const [userData, setUserData] = useState([]);

  const fetchData = async () => {

    try {

      const data = await fetchRandomUser();

      setUserData([...userData, data]);

    } catch (error) {

      console.error('Error fetching data:', error);

    }

  };

  return (

    <div className="App">

      <div className='UserCardContainer'>

        {userData.map((user, index) => (

          <UserComponent userData={user} key={index} />

        ))}

      </div>

      <button onClick={fetchData} className='LoadMoreButton' > <i className="fa fa-download"></i></button>

    </div>

  );

}

export default App;

Далі переглянемо вміст файлу UserComponent.jsx. Цей комонент приймає масив даних для одного користуввача та будує html на основі цих даних, звертаючись до потрібних властивостей

Лістинг файлу UserComponent.jsx:

import React from 'react';

import './UserComponent.css';

const UserComponent = ({ userData }) => {

  return (

    <div className="UserCard">

      <img src={userData.picture.large} alt="User" />

      <h2>Random User Data:</h2>

      <p>Cell: {userData.cell}</p>

      <p>City: {userData.location.city}</p>

      <p>Country: {userData.location.country}</p>

      <p>Postcode: {userData.location.postcode}</p>

    </div>

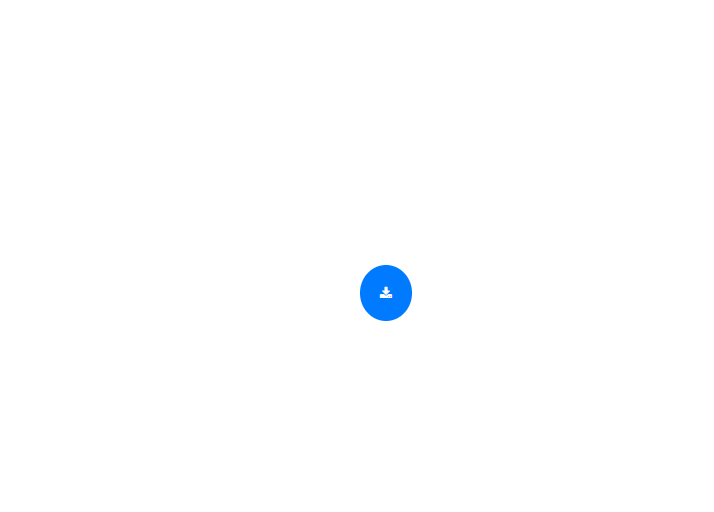
  );

};

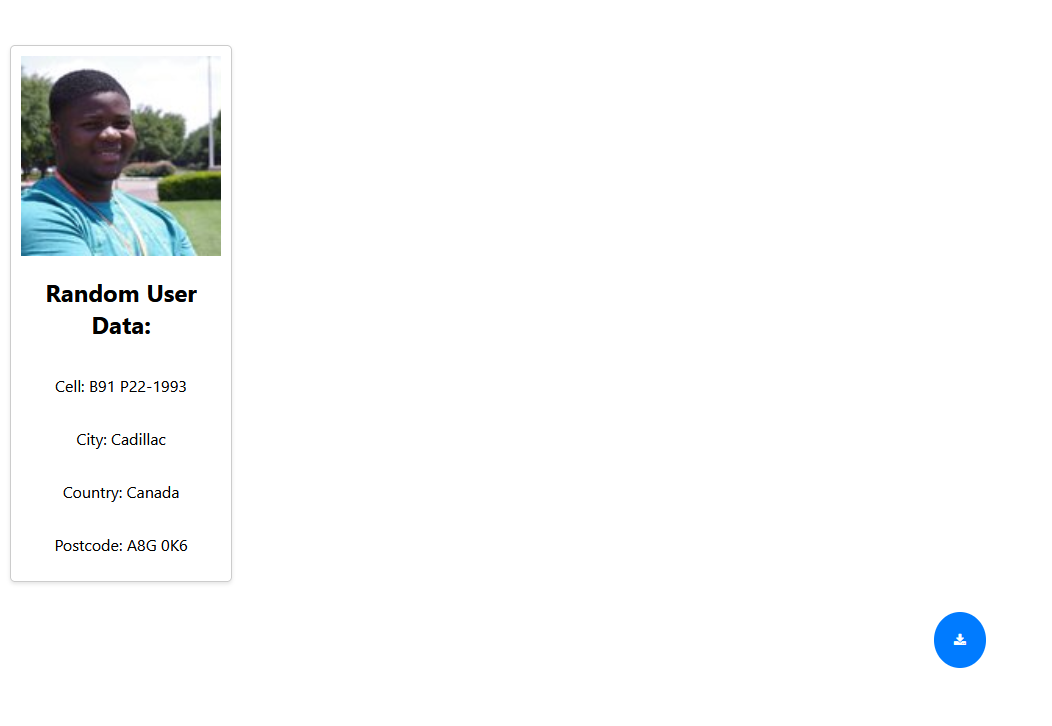
export default UserComponent;

Отриманий результат:

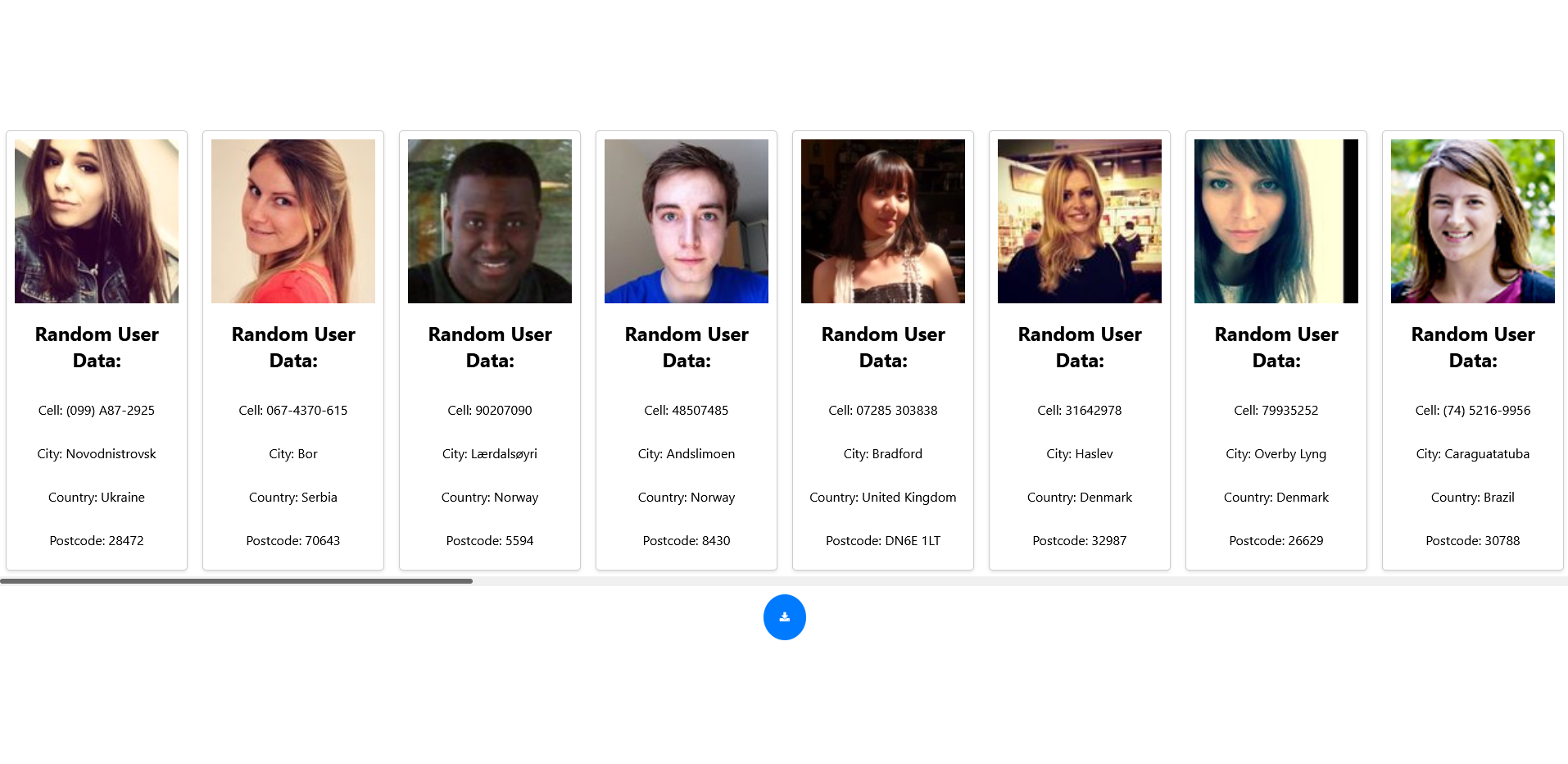
Сторінка при завантаженні:



Сторінка при одноразовому натисканні на кнопку:



Сторінка з декількома користувачами:



Висновок:

В результаті виконання завдання ми дослідили роботу з скриптовою мовою JavaScript. У ході виконання першої роботи ми дослідили роботу методів отримання даних з api-серверу у форматі json та навчитися використовувати такі дані. Було створено single page application, який при натисканні на кнопку завантажує дані одного нового користувача.

Посилання на репозиторій Git-hub: https://github.com/DanyloTrofymov/web-basics/tree/lab3