

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Практикум №6

з курсу «Основи Front-end технологій»

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

Викладач:  
Жереб К.А.

Виконав:  
Хільчук А.В.  
студент 3 курсу  
групи ІП-14 ФІОТ

Київ-2023

## Практична робота №6

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

**Завдання:**

Створити js-скрипт для отримання інформації із зовнішньої сторінки <https://randomuser.me/api> за допомогою методів fetch та promise у форматі JSON.

№ вар.	№ рядків в табл. 2				
1	1	2	3	4	5
2	1	3	4	5	6
3	1	4	5	6	7
4	1	2	4	6	8
5	1	3	5	7	9
6	1	2	4	5	6
7	1	3	4	6	7
8	1	2	5	6	8
9	1	4	6	3	2
10	1	3	4	7	9
11	1	4	5	2	6
12	1	5	7	8	9
13	1	8	9	6	5
14	1	6	4	2	7
15	1	3	5	6	6
16	1	7	2	8	4
17	1	4	3	5	7

1	picture	1
2	name	4
3	cell	
4	city	3
5	country	
6	postcode	2
7	email	5
8	phone	
9	coordinates	

**Виконання:**

*Опис використаних технологій:*

Отож, перш за все необхідно буде застосовувати метод fetch.

Fetch() – це глобальний метод інтерфейсу Fetch API для витяжки ресурсів мережею. Даний метод приймає у якості параметрів стрічку – адресу запитуваних ресурсів- а також у деяких перевантаженнях налаштування запиту. Працює він асинхронно, отож, повертає значення класу promise, яке по виконанню запиту отримає Response. Детальніше дослідити даний метод можна за посиланням на документацію:

<https://developer.mozilla.org/en-US/docs/Web/API/fetch>

Promise – об’єкт, що використовується для проведення відкладених та асинхронних обчислень. Він дозволяє оброблювати результати даних операцій так, ніби вони були синхронними, “обіцяючи” отримати результат в майбутньому. Кожен promise має свій статус, що може набувати значень “pending”, “fulfilled” та “rejected” та відображає статус виконання операції, а також властивість, де зберігається результат обчислень. Більш детально з promise можна ознайомитися в документації

за посиланням: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise)

Такі об'єкти мають низку методів, однак у рамках даної лабораторної роботи було застосовано два: `then` та `catch()`

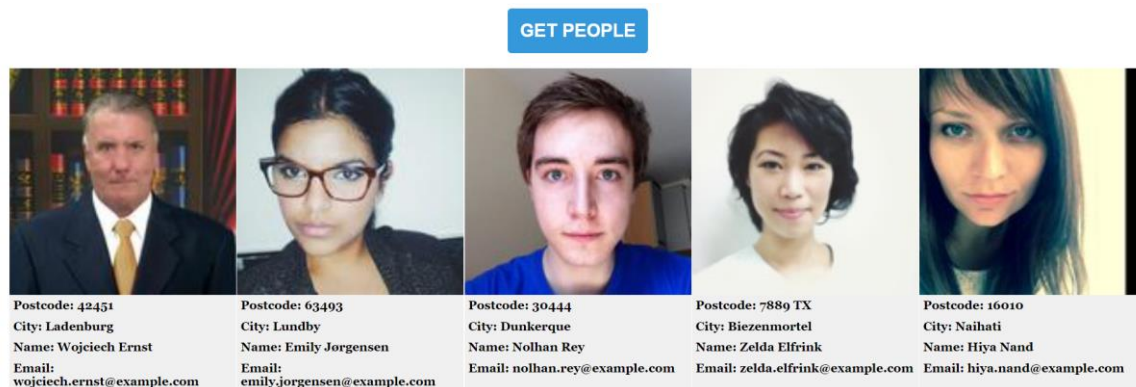
Метод `then` об'єктів `Promise` використовується для обробки результатів інших промісів та приймає до двох аргументів: функцію для обробки промісу зі статусом `fulfilled` і опціонально функцію на випадок `rejected`. Він аналогічно повертає об'єкт `Promise`, що дозволяє вибудовувати ланцюги викликів інших методів. Більш детально можна ознайомитись за документацією: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Promise/then](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/then)

Метод `catch()` схожий на `then()` у тому, що вони використовуються для обробки результатів промісу, однак даний метод у свою чергу спрацьовує лише у випадку, коли проміс має статус `Rejected`.

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Promise/catch](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/catch)

*Результуюча сторінка:*

Результатом виконання лабораторної роботи є дана веб-сторінка:



Сторінку було розгорнуто на Github Pages за даним посиланням: <https://khilchukartemip-14.github.io/web-front-end-basics/Lab6/Index.html>

Із кодом можна ознайомитися в даному репозиторії:

<https://github.com/KhilchukArtemIP-14/web-front-end-basics/tree/main/Lab6/Index.html>

## **Висновок**

Отож, у ході виконання лабораторної роботи було зверстано сторінку, що посилає запит ресурсів на віддалений API, а згодом динамічно відображає отримані дані. Реалізацію даної функціональності було досягнуто за допомогою методу `fetch()`, а також `catch()` і `then()` об'єкту `Promise` мовою JavaScript, які забезпечували комунікацію із віддаленими сервісами мережею. Урешті-решт, сторінку було розгорнуто на хостингу Github Pages. Набуто практичних навичок застосування JavaScript для отримання віддалених ресурсів мережею, а також роботи з асинхронними обчисленнями даною мовою.