Національний технічний університет України «КПІ»

Факультет інформатики та обчислювальної техніки

Кафедра Інформаційних систем та технологій

Лабораторна робота №5

з дисципліни «Проектування інформаційних систем»

на тему: «Використання і створення API»

Виконала:

студентка гр. ІС-34

Ященко Олександра

Викладач:

Ковальов М. О.

2025 рік

**Мета**: ознайомитися з використанням та свторенням API

**Постановка завдання**

Підключити зовнішній сервіс до свого рішення. Сервіс можна використовувати існуючий або створити власний. Приклади зовнішніх сервісів наведені в Додатку.

Вправа 6.1. Обрати зовнішній сервіс. і описати сценарій його використання. Створити в GitHub репозіторій з описом сценарію використання API зовнішнього застосування.

Вправа 6.2. Реалізувати на будь-якій мові програмування виклик API зовнішнього застосування і візуалізацію відповіді. Додати в GitHub код і опис прикладу застосування API.

Вправа 6.3. Реалізувати на будь-якій мові програмування http-сервер, організувати звертання до нього і отримати відповідь у браузері. Звертання має містити Ваш логін у Moodle. Відповідь має містити Ваші особисті дані (прізвище, ім’я, курс, група).

**Виконання роботи:**

Посилання на GitHub репозиторій:

<https://github.com/kitchensink20/PIS-labs/tree/main/REST-API-Test>

***Вправа 6.1***

# **Опис сценаріїв використання NewsAPI**

## **1. Загальний сценарій**

**Мета:** Створення консольного клієнта для отримання та відображення актуальних новин з України через NewsAPI.

## **2. Сценарій вашої роботи**

### **Функціональність:**

* **Підключення до API:** Використання https модуля Node.js для виклику NewsAPI
* **Автентифікація:** Передача API ключа в параметрах запиту
* **Пошук новин:** Фільтрація за ключовим словом "Ukraine"
* **Обмеження результатів:** Отримання 3 найсвіжіших новин
* **Відображення даних:** Консольний вивід з форматуванням:
* Номер новини
* Заголовок
* Джерело новин
* Короткий опис
* URL статті

### **Технічна реалізація:**

1. **HTTP запит** з обов'язковим User-Agent заголовком
2. **Обробка JSON відповіді** з перевіркою статусів
3. **Обробка помилок** - мережевих, парсингу, API помилок
4. **Структурований вивід** даних у читабельному форматі

### **Практичне застосування:**

Демонстрація базових навичок роботи з REST API, HTTP запитами та обробкою JSON даних в Node.js. Код готовий для розширення до повноцінного новинного агрегатора з веб-інтерфейсом.

***Вправа 6.2***

Реалізація за допомогою NodeJS:

const https = require('https');

const { URL } = require('url');

const API\_KEY = '68bbe5540abf4c98a11ad18cce21cc79';

function makeRequest(*url*, *callback*) {

const urlObj = new URL(*url*);

const options = {

hostname: urlObj.hostname,

path: urlObj.pathname + urlObj.search,

method: 'GET',

headers: {

'User-Agent': 'Node.js News Client 1.0'

}

};

https.request(options, (*res*) => {

let data = '';

*res*.on('data', (*chunk*) => {

data += *chunk*;

});

*res*.on('end', () => {

callback(null, *res*.statusCode, data);

});

}).on('error', (*error*) => {

callback(*error*, null, null);

}).end();

}

function testNewsAPI() {

if (API\_KEY === 'YOUR\_API\_KEY') {

return;

}

const testUrl = `https://newsapi.org/v2/everything?q=Ukraine&pageSize=3&apiKey=${API\_KEY}`;

console.log('Тестуємо підключення до NewsAPI...');

makeRequest(testUrl, (*error*, *statusCode*, *data*) => {

if (*error*) {

console.error('Помилка мережі:', *error*.message);

return;

}

console.log(`Статус код: ${*statusCode*}`);

if (*statusCode* !== 200) {

console.log('Відповідь сервера:', *data*.substring(0, 200) + '...');

return;

}

try {

const jsonData = JSON.parse(*data*);

if (jsonData.status === 'error') {

console.error('Помилка API:', jsonData.message);

return;

}

if (jsonData.articles && jsonData.articles.length > 0) {

console.log('\n=== НОВИНИ З NEWSAPI ===');

jsonData.articles.forEach((article, index) => {

console.log(`\n${index + 1}. ${article.title}`);

console.log(` Джерело: ${article.source.name}`);

console.log(` Опис: ${article.description || 'Опис відсутній'}`);

console.log(` URL: ${article.url}`);

});

} else {

console.log('Новини не знайдено');

}

} catch (parseError) {

console.error('Помилка обробки JSON:', parseError.message);

console.log('Відповідь:', data.substring(0, 200) + '...');

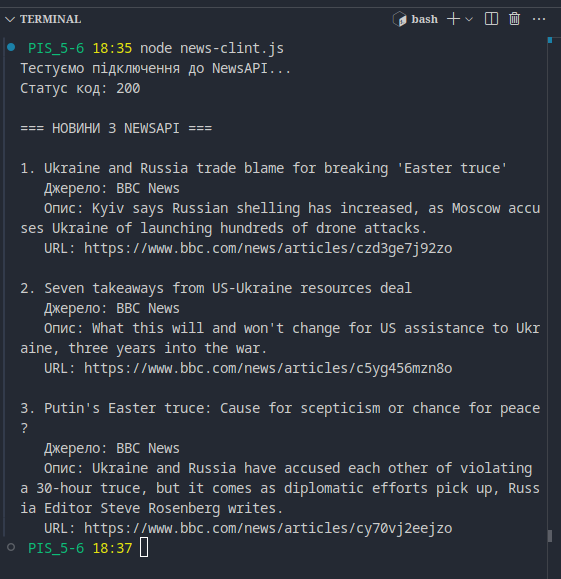
}

});

}

testNewsAPI();

Результат:



***Вправа 6.3***

Створення серверу засобами Node.js:

const http = require('http');

const url = require('url');

const personalData = {

login: 'is-34fiot-23-184',

surname: 'Ященко',

name: 'Олександра',

course: '2 курс',

group: 'ІС-34'

};

const server = http.createServer((*req*, *res*) => {

const parsedUrl = url.parse(*req*.url, true);

const pathname = parsedUrl.pathname;

const query = parsedUrl.query;

*res*.setHeader('Content-Type', 'text/html; charset=utf-8');

if (pathname === '/') {

*res*.writeHead(200);

*res*.end(`

<!DOCTYPE html>

<html lang="uk">

<head>

<meta charset="UTF-8">

<title>Особисті дані студента</title>

<style>

body {

font-family: Arial, sans-serif;

max-width: 600px;

margin: 100px auto;

padding: 20px;

background: #f0f0f0;

}

.container {

background: white;

padding: 40px;

border-radius: 10px;

box-shadow: 0 2px 10px rgba(0,0,0,0.1);

}

h1 { color: #333; text-align: center; }

input {

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ddd;

border-radius: 5px;

}

button {

width: 100%;

padding: 12px;

background:rgb(46, 46, 46);

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

button:hover { background:rgb(0, 0, 0); }

</style>

</head>

<body>

<div class="container">

<h1>Інформація про студента</h1>

<form action="/info" method="GET">

<label>Введіть ваш логін з Moodle:</label>

<input type="text" name="login" required>

<button type="submit">Отримати інформацію</button>

</form>

</div>

</body>

</html>

`);

}

else if (pathname === '/info') {

const login = query.login;

if (!login) {

*res*.writeHead(400);

*res*.end('<h1>Помилка: логін не вказано!</h1><a href="/">Назад</a>');

return;

}

if(personalData.login !== login) {

*res*.writeHead(400);

*res*.end('<h1>Помилка: Користувача не знайдено в системі!</h1><a href="/">Назад</a>');

return;

}

*res*.writeHead(200);

*res*.end(`

<!DOCTYPE html>

<html lang="uk">

<head>

<meta charset="UTF-8">

<title>Особисті дані</title>

<style>

body {

font-family: Arial, sans-serif;

max-width: 600px;

margin: 100px auto;

padding: 20px;

background: #f0f0f0;

}

.container {

background: white;

padding: 40px;

border-radius: 10px;

box-shadow: 0 2px 10px rgba(0,0,0,0.1);

}

h1 { color: #333; text-align: center; }

.info {

background: #f8f9fa;

padding: 20px;

border-radius: 5px;

margin: 20px 0;

}

.field {

margin: 10px 0;

display: flex;

justify-content: space-between;

}

.label { font-weight: bold; }

a {

display: block;

text-align: center;

margin-top: 20px;

color:rgb(0, 0, 0);

text-decoration: none;

}

</style>

</head>

<body>

<div class="container">

<h1>Особисті дані студента</h1>

<div class="info">

<div class="field">

<span class="label">Логін у Moodle:</span>

<span>${personalData.login}</span>

</div>

<div class="field">

<span class="label">Прізвище:</span>

<span>${personalData.surname}</span>

</div>

<div class="field">

<span class="label">Ім'я:</span>

<span>${personalData.name}</span>

</div>

<div class="field">

<span class="label">Курс:</span>

<span>${personalData.course}</span>

</div>

<div class="field">

<span class="label">Група:</span>

<span>${personalData.group}</span>

</div>

</div>

<a href="/">← Повернутися</a>

</div>

</body>

</html>

`);

}

else {

*res*.writeHead(404);

*res*.end(`

<!DOCTYPE html>

<html lang="uk">

<head>

<meta charset="UTF-8">

<title>Сторінка не знайдена</title>

<style>

body {

font-family: Arial, sans-serif;

text-align: center;

padding: 100px;

background: #f0f0f0;

}

h1 { color: #dc3545; }

a { color: #007bff; text-decoration: none; }

</style>

</head>

<body>

<h1>404 - Сторінка не знайдена</h1>

<a href="/">На головну</a>

</body>

</html>

`);

}

});

const PORT = 3000;

server.listen(PORT, () => {

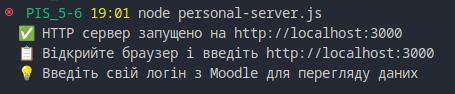
console.log(`✅ HTTP сервер запущено на http://localhost:${PORT}`);

console.log(`📋 Відкрийте браузер і введіть http://localhost:${PORT}`);

console.log(`💡 Введіть свій логін з Moodle для перегляду даних\n`);

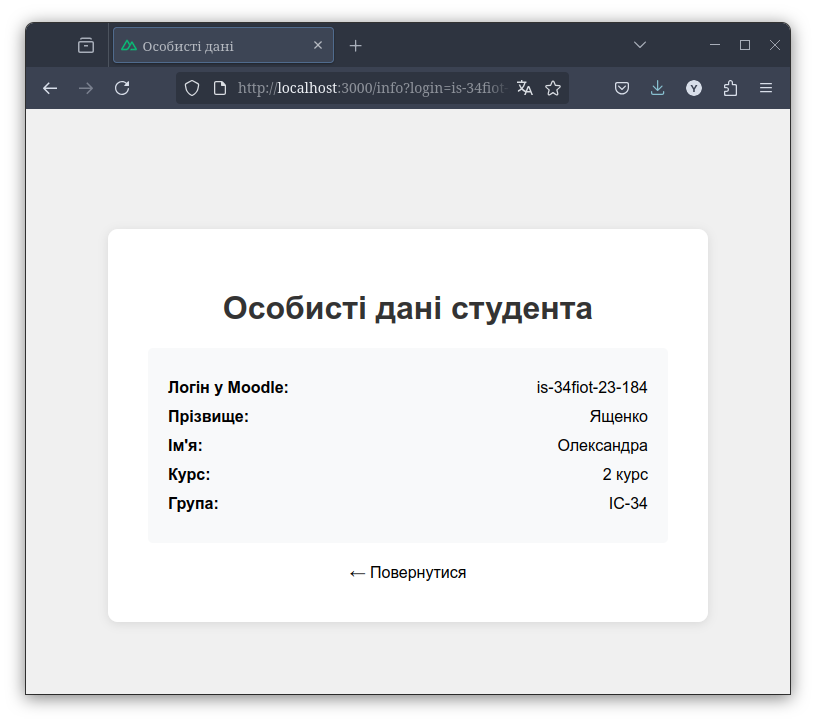
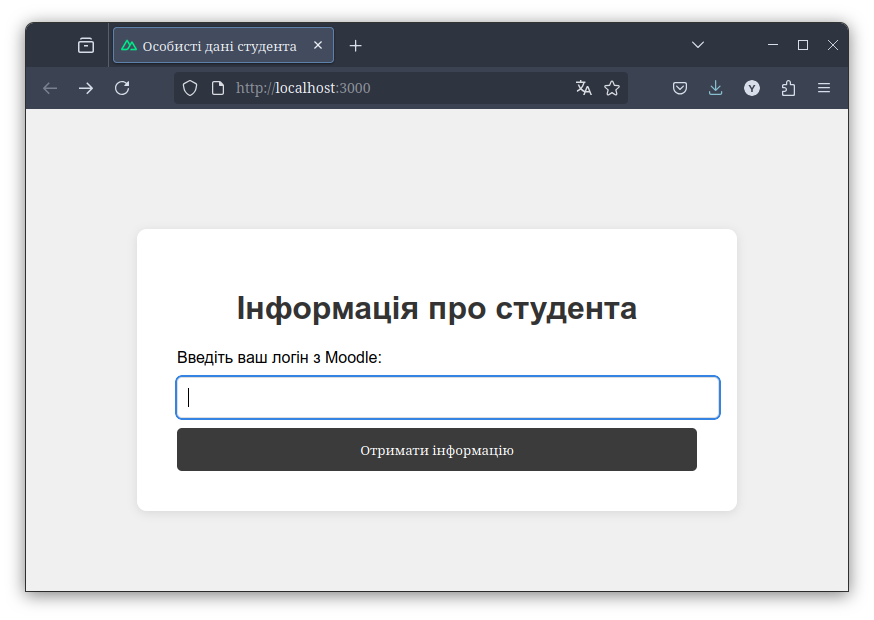
});

Запуск серверу, результат в терміналі VS code:



Запуск серверу, результат на веб-сторінці:

іва



**Висновки:**

У ході роботи я ознайомилася з використанням та створенням API та

підключила зовнішній сервіс до свого рішення.