

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра моделювання та програмного забезпечення

**Лабораторна робота**

з дисципліни «Сучасні технології Internet-програмування»

Тема: «Дослідження роботи протоколу HTTP»

Виконала:

студентка групи ПЗ-22-2

Смаглюк К. А.

Перевірила

доцент кафедри моделювання та  
програмного забезпечення

Трачук А. А

Кривий Ріг  
2025р.

## Завдання 1

На данному сайті використовуються:

- типи HTTP-запитів: GET та POST;
- заголовки запитів:
  - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
  - Host: openweathermap.org
  - Content-Type: text/html; charset=UTF-8
- Відповідь сервера:
  - статусний код: Status Code: 200 OK
  - заголовки: Server: nginx/1.24.0

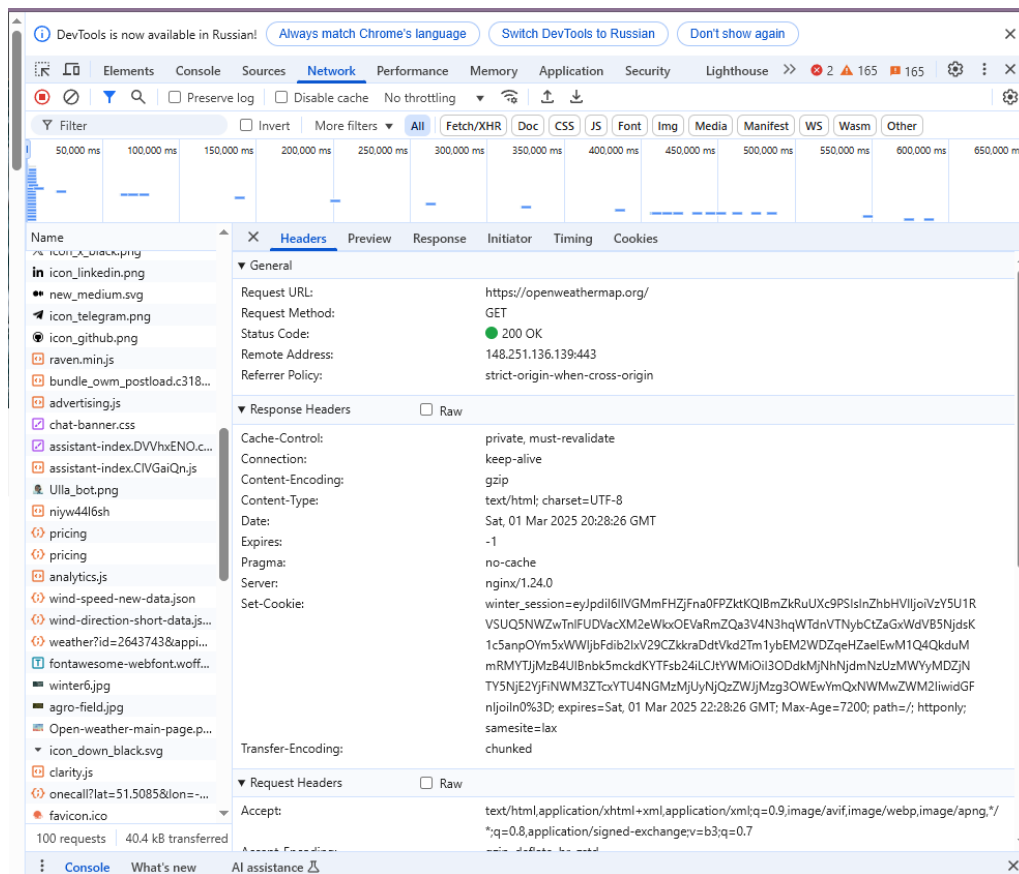


Рисунок 1 Інструменти розробника

## Завдання 2

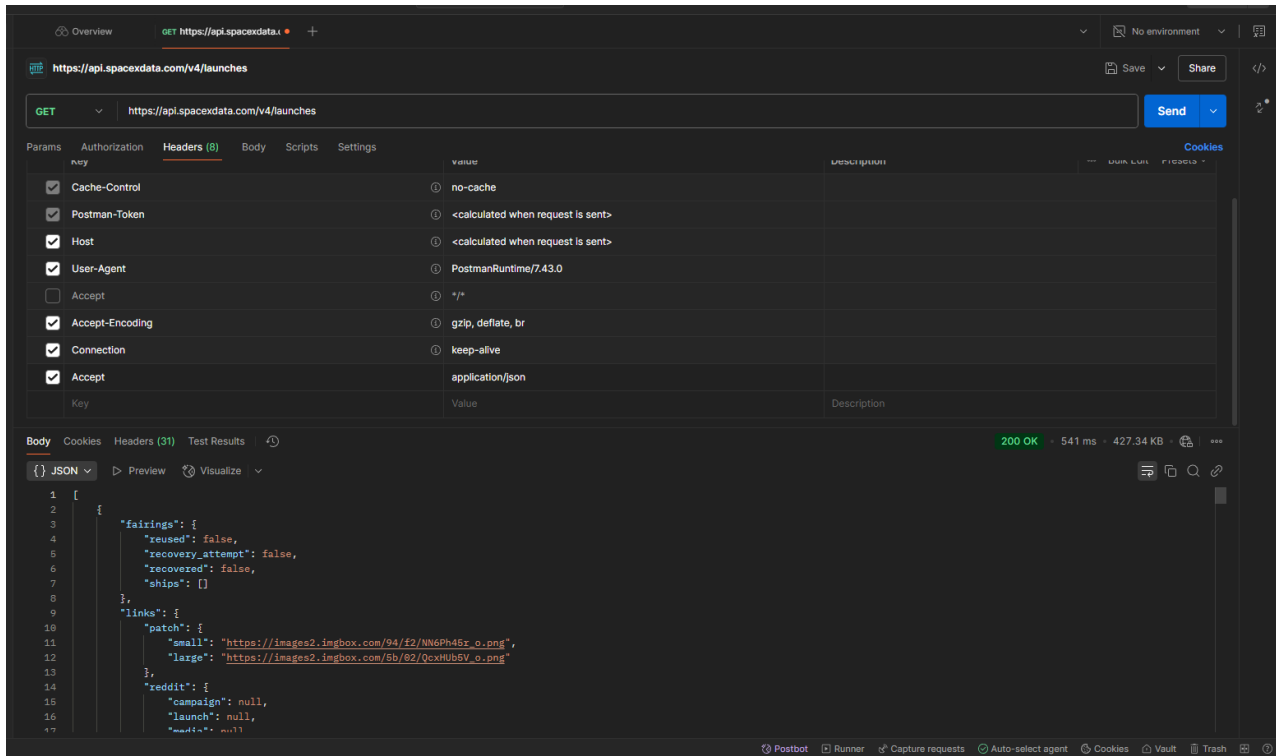


Рисунок 2 Скриншот запиту GET у Postman та відповіді сервера

## Завдання 3

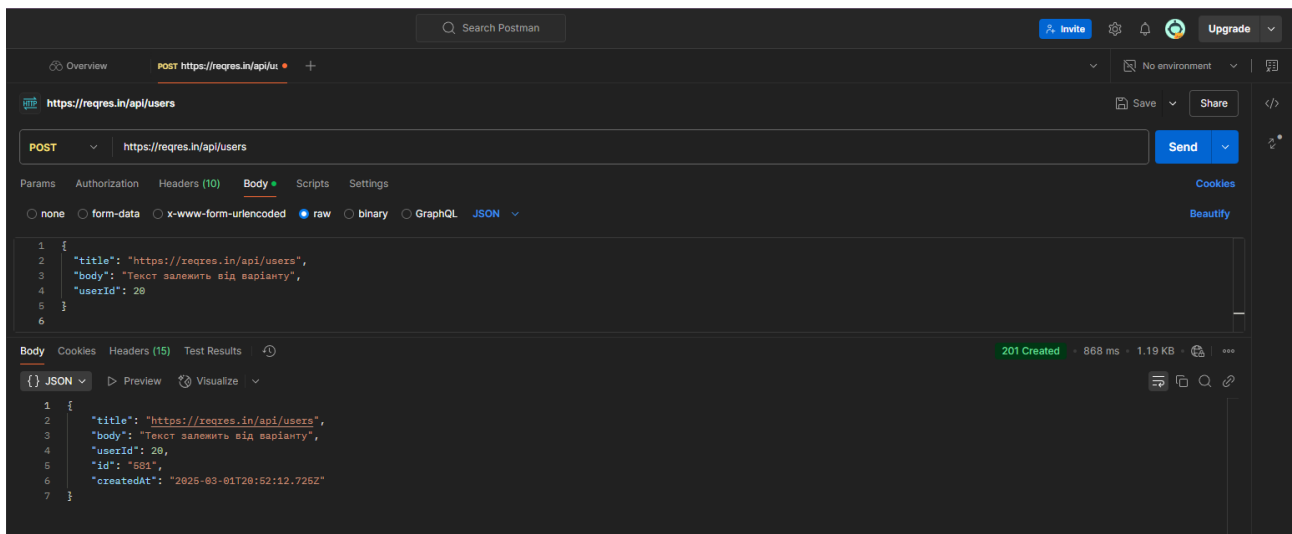


Рисунок 3 Скриншот запиту POST у Postman та відповіді сервера

## Завдання 4

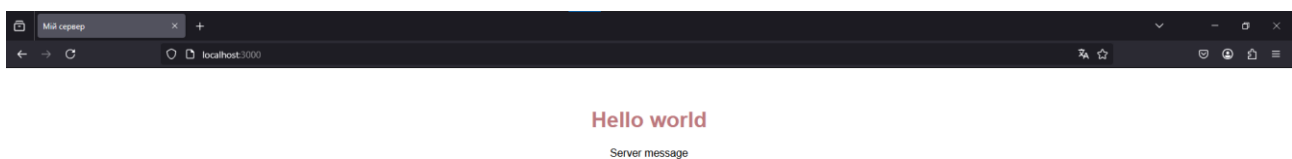


Рисунок 4 Відповідь сервера

## Код HTTP-сервера

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/html; charset=utf-8' });
  res.end(`
    <html>
    <head>
      <title>Мій сервер</title>
      <style>
        body { font-family: Arial, sans-serif; text-align: center; padding:
50px; }
        h1 { color: #bd787d; }
      </style>
    </head>
    <body>
      <h1>Hello world</h1>
      <p>Server message</p>
    </body>
    </html>
  `);
});

server.listen(3000, () => {
  console.log('Сервер запущено на порту 3000');
});
```

### Відповіді на питання

1. **HTTP (HyperText Transfer Protocol)** – це протокол передачі даних у мережі, що використовується для взаємодії клієнта (браузера) і сервера.

Основні методи:

- **GET** – отримання даних
- **POST** – надсилання нових даних
- **PUT** – оновлення даних
- **DELETE** – видалення даних

2. Заголовки передають додаткову інформацію між клієнтом і сервером.

Наприклад:

- **User-Agent** – інформує сервер про браузер/пристрій клієнта
- **Content-Type** – вказує тип переданих даних (наприклад, application/json)
- **Authorization** – використовується для автентифікації

3.

- **1xx** – Інформаційні (наприклад, 100 Continue)
- **2xx** – Успішні (наприклад, 200 OK)
- **3xx** – Перенаправлення (наприклад, 301 Moved Permanently)

- **4xx** – Помилки клієнта (наприклад, 404 Not Found)
- **5xx** – Помилки сервера (наприклад, 500 Internal Server Error)

4. **REST API (Representational State Transfer)** – це стиль архітектури для побудови веб-сервісів, що працюють через HTTP.

Основні принципи:

- **Клієнт-сервер** – розділення логіки клієнта і сервера
- **Безстанова взаємодія** – кожен запит містить всю необхідну інформацію
- **Кешування** – зменшення навантаження на сервер
- **Уніфікований інтерфейс** – використання стандартних HTTP-методів
- **Шарованість** – можливість мати проміжні сервери (проксі, балансувальники навантаження)

5. **HTTPS (HyperText Transfer Protocol Secure)** використовує шифрування через SSL/TLS, що забезпечує:

- **Конфіденційність** – передані дані шифруються
- **Цілісність** – дані не можуть бути змінені під час передачі
- **Автентифікацію** – перевіряється справжність сайту через SSL-сертифікат

### **Висновок**

На даній лабораторній роботі я навчилася працювати з HTTP-запитами та відповідями, аналізувати їхні заголовки та коди стану, а також здійснювати роботу з REST API.