МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

**Комп’ютерні інформаційні мережі**

**Лабораторна робота №9**

Виконала:

студентка групи ПМі-31

Харитонова Анастасія

Львів 2024

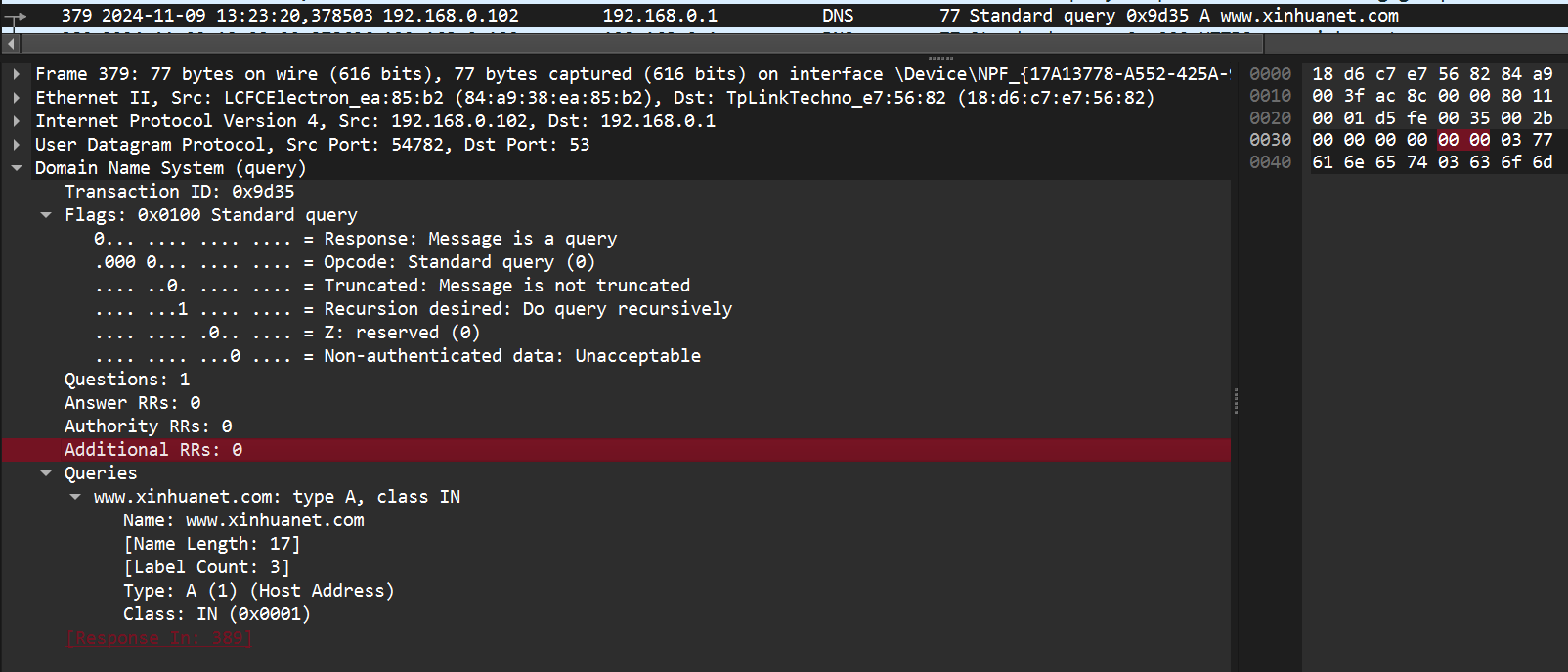
**Тема:** Протоколи прикладного рівня. Postman

**Мета:** Здобути практичні навички з інтерпретації повідомлень протоколів прикладного рівня та опанувати на базовому рівні роботу на платформі Postman.

**Хід роботи:**

**Частина 1. Wireshark**

**1-2.**



Transaction ID: 0x9d35 – унікальний ідентифікатор запиту.

Flags: 0x0100 Standard query

Response: Message is a query – вказує, що це саме запит.

Opcode: Standard query (0) – код операції, що вказує на тип запиту.

Truncated: Message is not truncated – повідомлення не урізане.

Recursion desired: Do query recursively – клієнт запитує рекурсивний пошук.

Z: reserved (0) – зареєстроване поле.

Non-authenticated data: Unacceptable – неавтентифіковані дані неприйняті.

Questions: 1 – к-ть запитів, поданих в повідомленні.

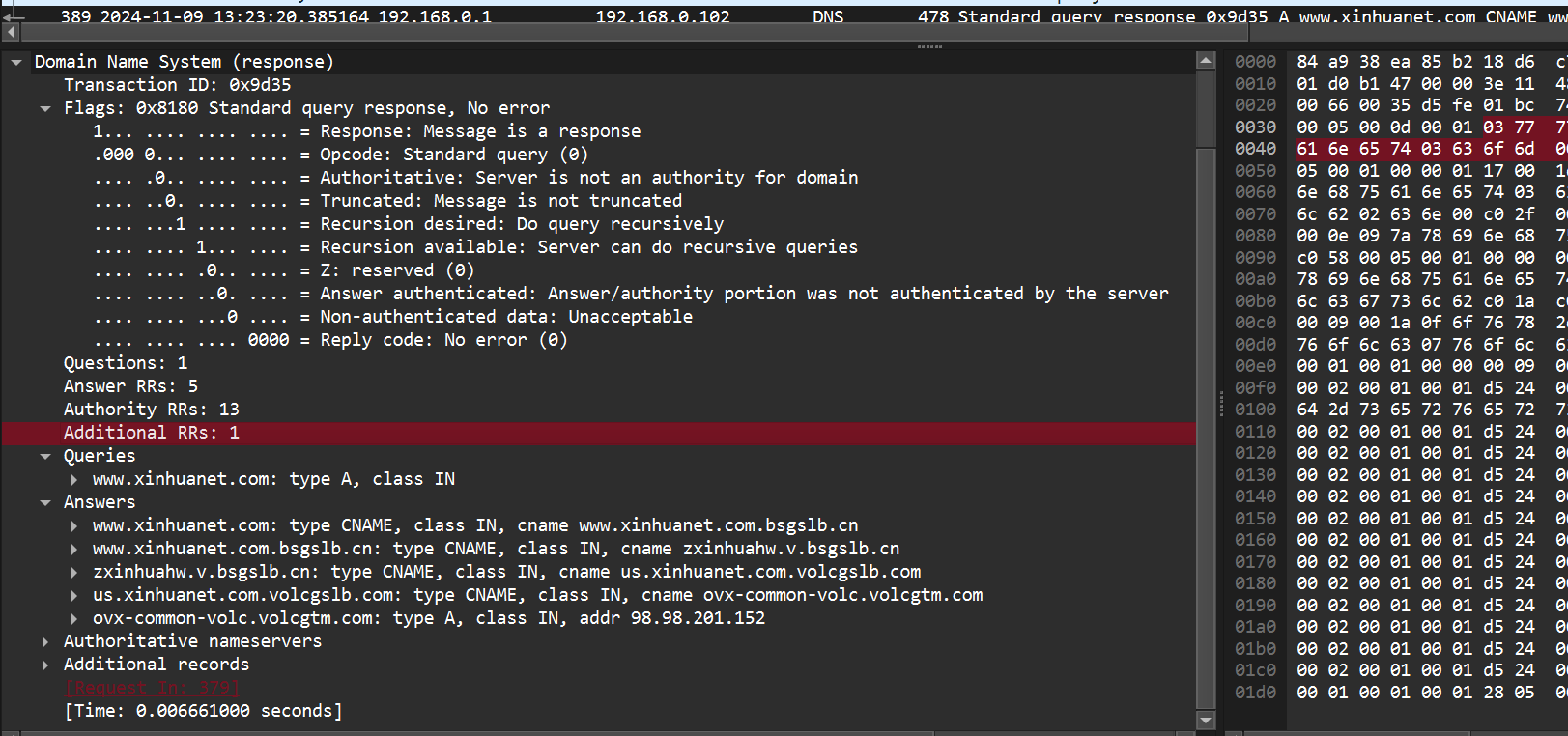
Answer RRs: 0 – відповідей нема.

Authority RRs: 0 – нема записів про авторитетні сервери.

Additional RRs: 0 – нема додаткових записів.

Queries

www.xinhuanet.com: type A, class IN – доменне ім’я, тип А (ІР4), клас вказує на те, що запит призначений для інтернету.



Transaction ID: 0x9d35 – ідентифікатор, поєднує запит і відповідь.

Flags: 0x8180 Standard query response, No error – стандартна відповідь без помилок.

Response: Message is a response – відповідь.

Recursion available: Server can do recursive queries – рекурсія доступна.

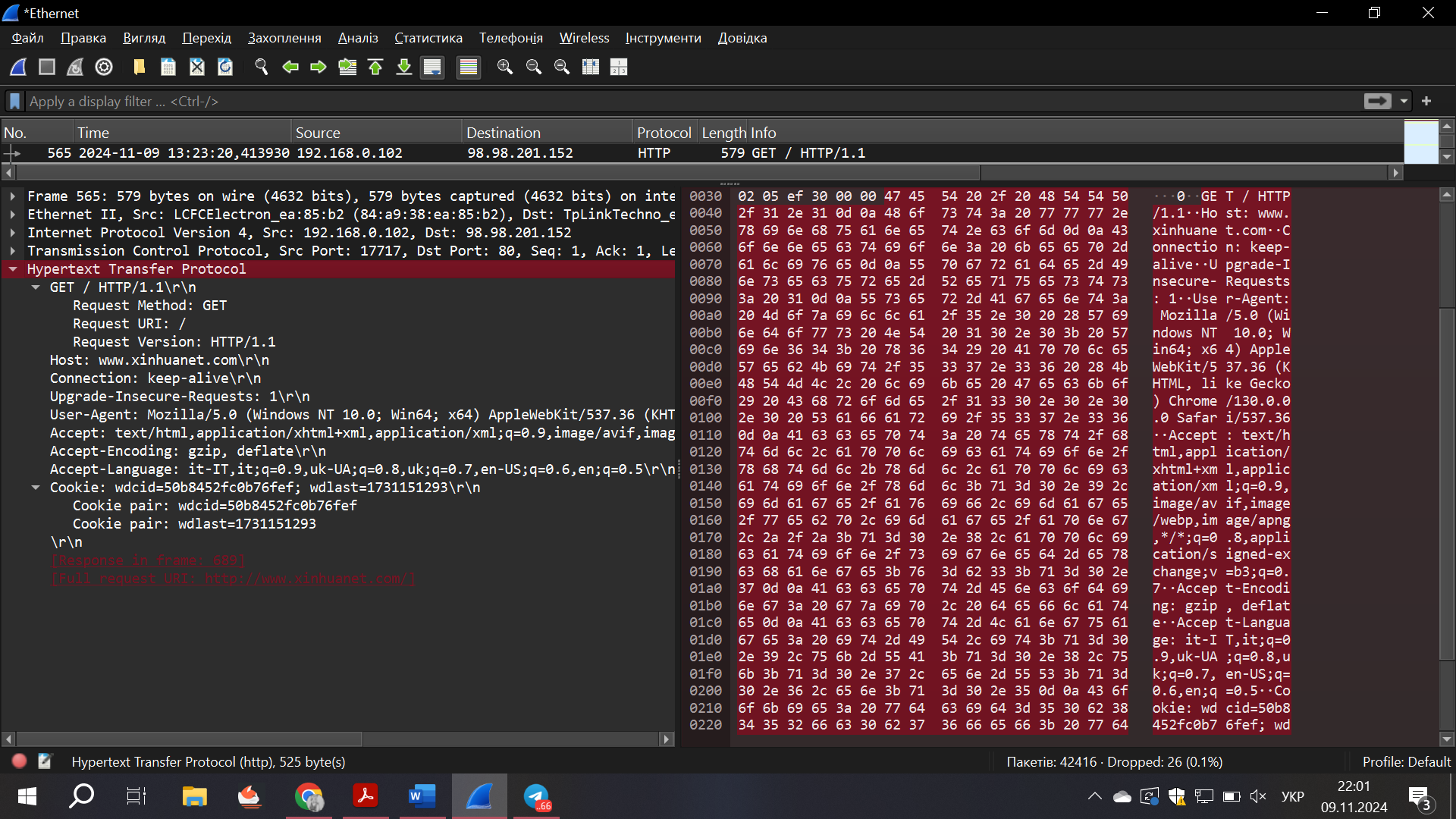
Answers – ланцюжок записів CNAME, що веде до остаточної ІР-адреси.

www.xinhuanet.com: type CNAME, class IN, cname …

ovx-common-volc.volcgtm.com: type A, class IN, addr 98.98.201.152

Additional recordsс – додаткові записи.

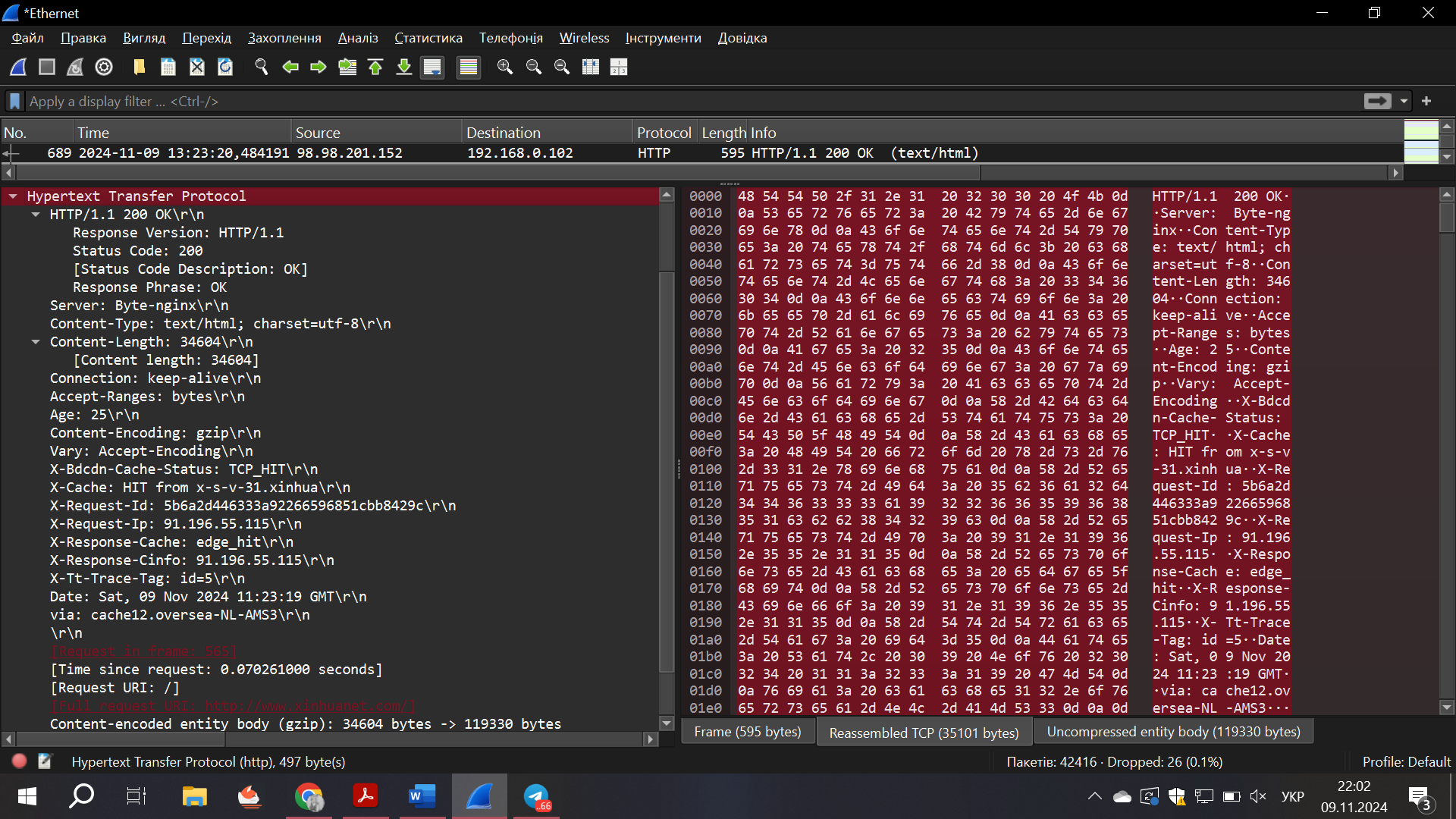
**3-4.**



**Request:**

Request method: GET – клієнт (браузер) запитує ресурс з сервера (в моєму випадку головну сторінку сайту).  
Request URI: / - шлях до ресурсу на сервері (/ корінь сайту).  
Request Version: HTTP/1.1 – версія протоколу HTTP.

Host: доменне ім’я сайту.  
Connection: вказує на тип з’єднання з сервером (keep-alive – відкрите для подальших запитів).  
Upgrade-Insecure-Requests: 1 – повідомлення для сервера, що клієнт готовий прийняти оновлення HTTP до HTTPS, якщо це можливо.  
User-Agent: ідентифікатор браузера та операційної системи.  
Accept: типи контенту, які браузер готовий прийняти.  
Accept-Encoding: які методи стиснення клієнт підтримує для відповіді від сервера.  
Accept-Language: які мови підтримує браузер і їх пріоритет.  
Cookie: передає cookies, які зберігаються в браузері клієнта.



**Response:**

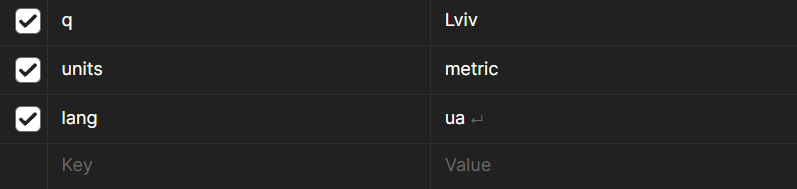
Response version: версія  
Status code: 200 [OK] – код вказує, що запит був успішно оброблений і відповідь надана.

Server: веб-сервер, який обробляє запит.  
Content-Type: тип контенту, який буде переданий в тілі відповіді.  
Content-Length: кількість байт у тілі відповіді.  
Connection: keep-alive з’єднання буде збережено.  
Accept-Ranges: сервер підтримує часткові запити до файлів.  
Age: скільки часу в секундах пройшло з моменту останнього кешування.  
Content-Encoding: вміст відповіді був стиснутий за допомогою алгоритму gzip.  
Vary: сервер змінює вміст в залежності від заголовка Accept-Encoding в запиті клієнта.  
**X-Bdcdn-Cache-Status**: TCP\_HIT – кеш на сервері використовувався для обслуговування цього запиту, ресурс був знайдений у кеші і передавався з нього, а не з сервера. **X-Cache**: з якого сервера були отримані кешовані дані. **X-Request-Id**: унікальний ідентифікатор для кожного запиту. **X-Request-Ip**: вказує на ІР-адресу клієнта. **X-Response-Cache**: відповідь була отримана безпосередньо з кешу. **X-Response-Cinfo**: інформація про сервер, що відповідав на запит. **X-Tt-Trace-Tag**: додаткові метадані.  
Date: дата і час.  
Via: відповідь пройшла через проміжний кеш-сервер.

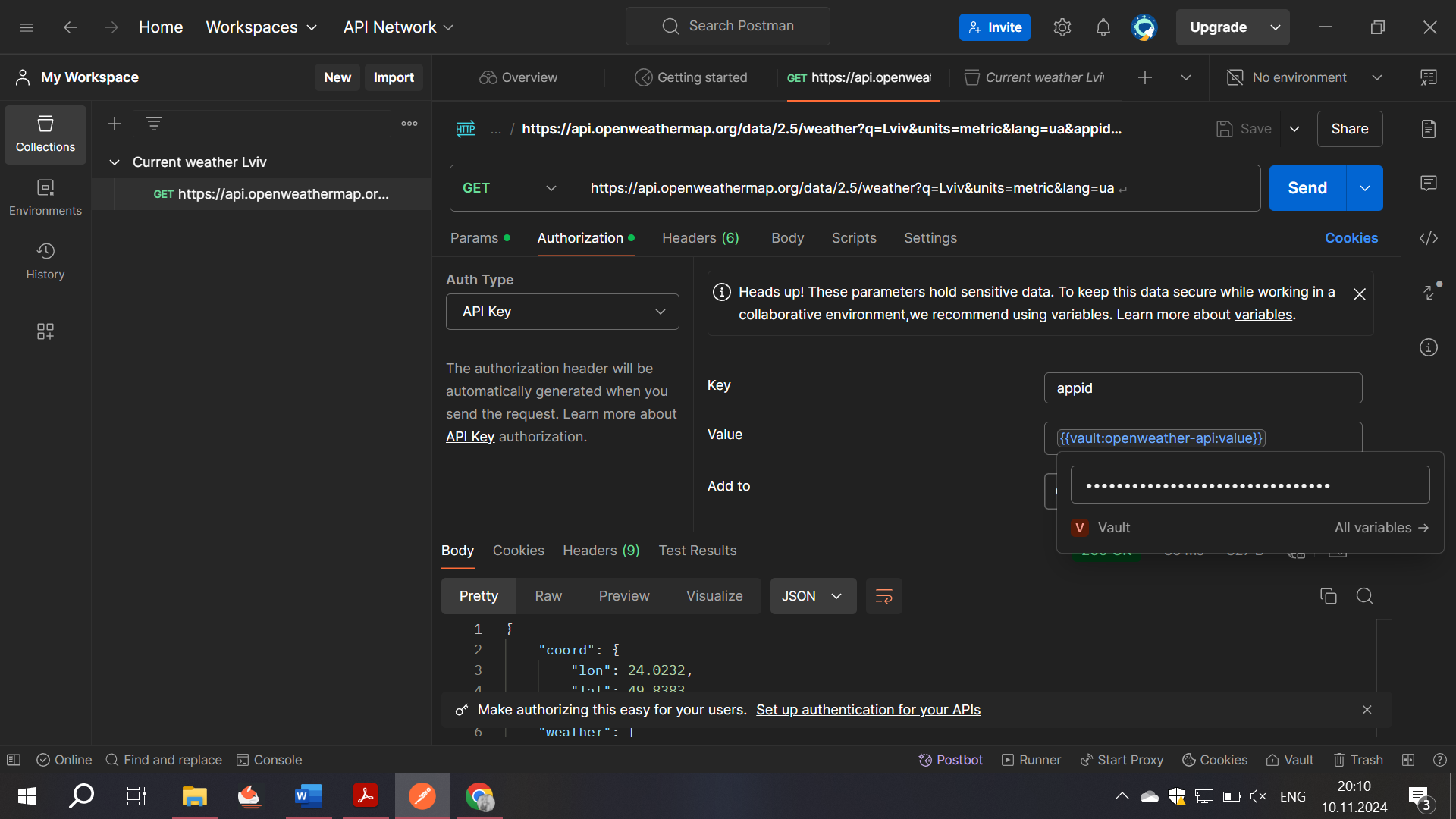
**5.** У моєму випадку були захоплені лише GET запити. Оскільки я не виконувала діяльність, наприклад: авторизація, реєстрація, надсилання форм і тд.  
POST – для відправки даних на сервер.  
PUT – зазвичай для оновлення існуючого ресурсу на сервері.  
DELETE – видалення ресурсу з сервера.

**Частина 2. Postman**

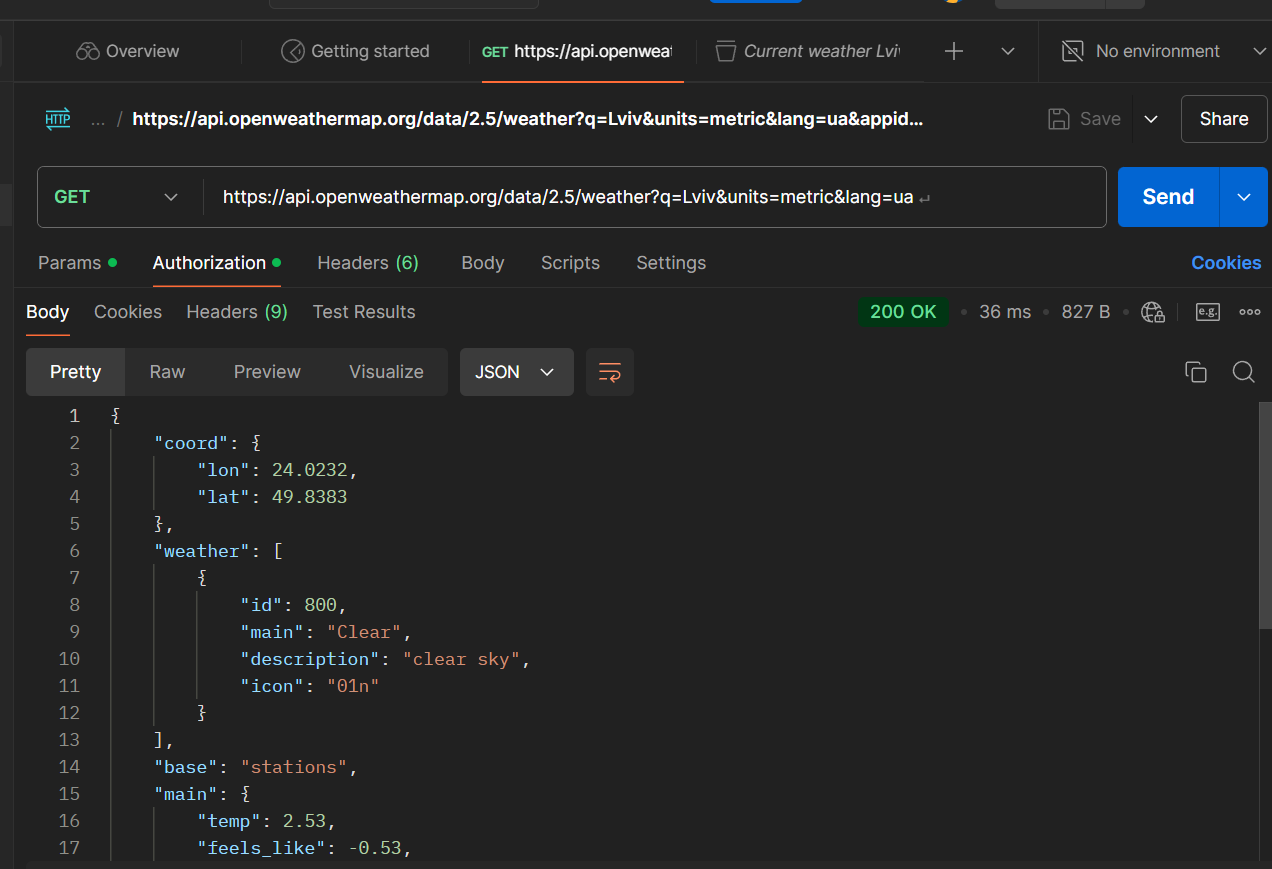
**10-12.** Створила колекцію, обрала тип запиту HTTP, метод GET, ввела адресу кінцевої точки, визначила тип АРІ-запиту про поточну погоду у Львові, у вкладці Params задала імена та значення параметрів.



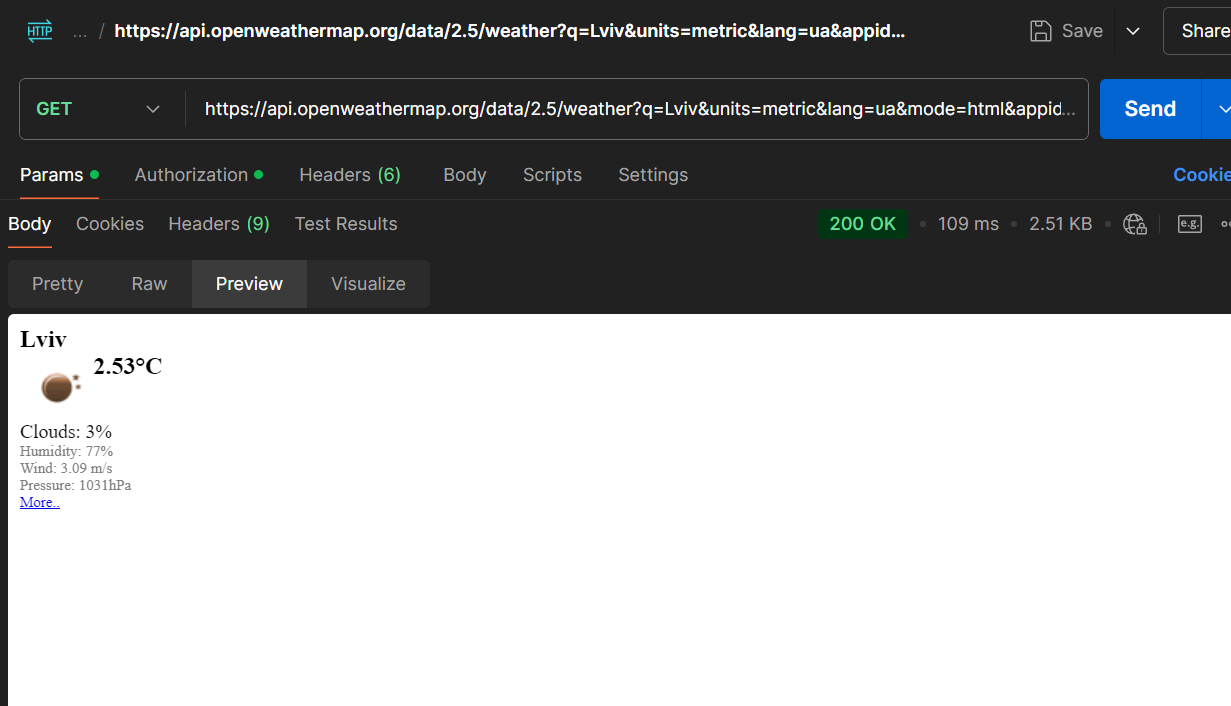
**13.** На вкладці Authorization задала параметри (значення – мій ключ).



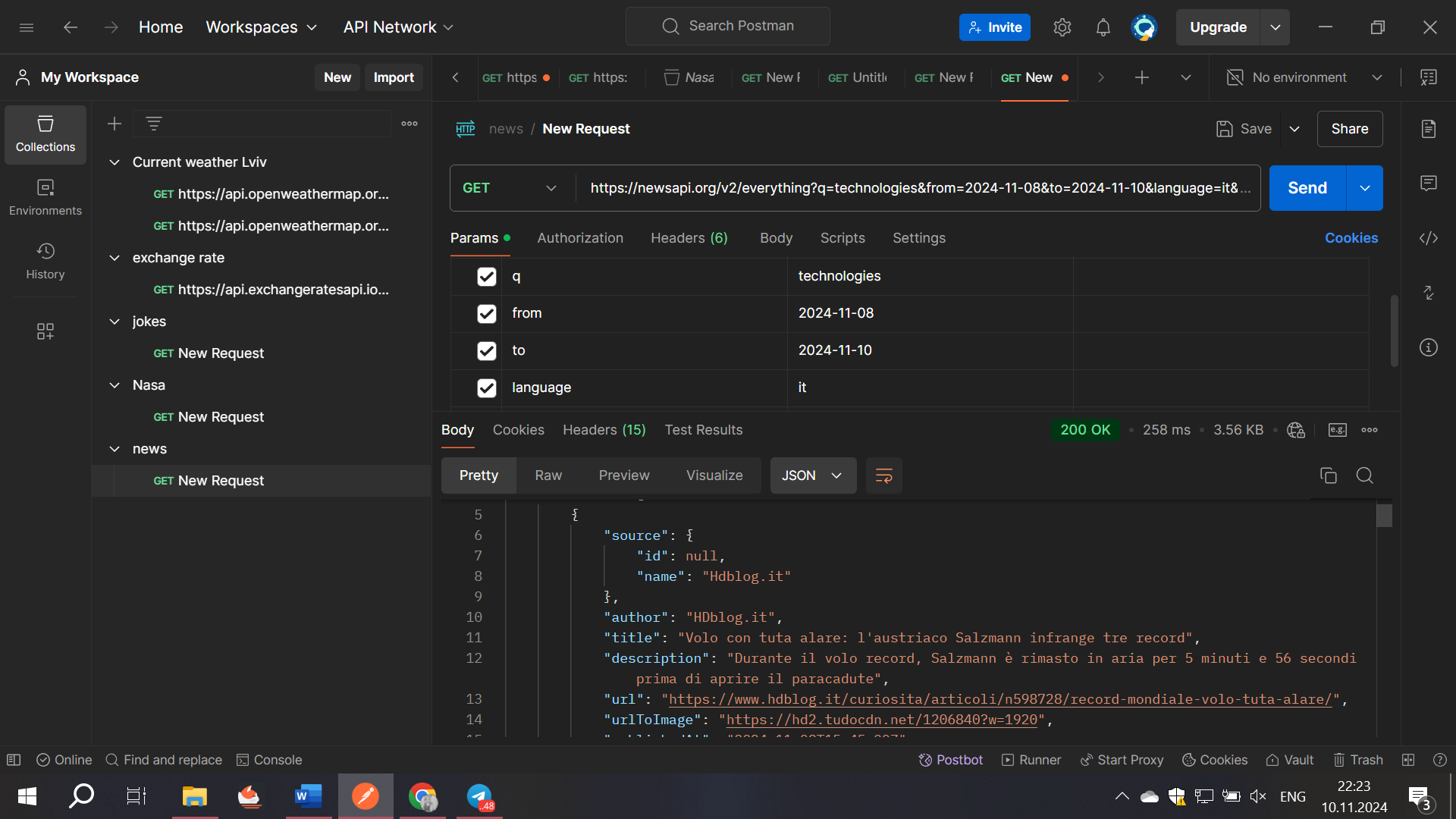
**14-15.** Виконала запит, ознайомилась з результатами, зберегла запит у колекцію.

****

**16-17.** Створила копію запиту, додала до параметрів mode=html і отримала такий результат. Формат відповіді – html.

****

**18.** Виконала запит до newsAPI, який відображає новини (вказала параметри q=тема, from – to = період часу, мова).



**Висновок:** Здобула практичні навички з інтерпретації повідомлень протоколів прикладного рівня та опанувала на базовому рівні роботу на платформі Postman.