

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

**Комп'ютерні інформаційні мережі**

**ЛАБОРАТОРНА РОБОТА №9**

Виконав:

Ст. Лук'янчук Денис

Група ПМі-33

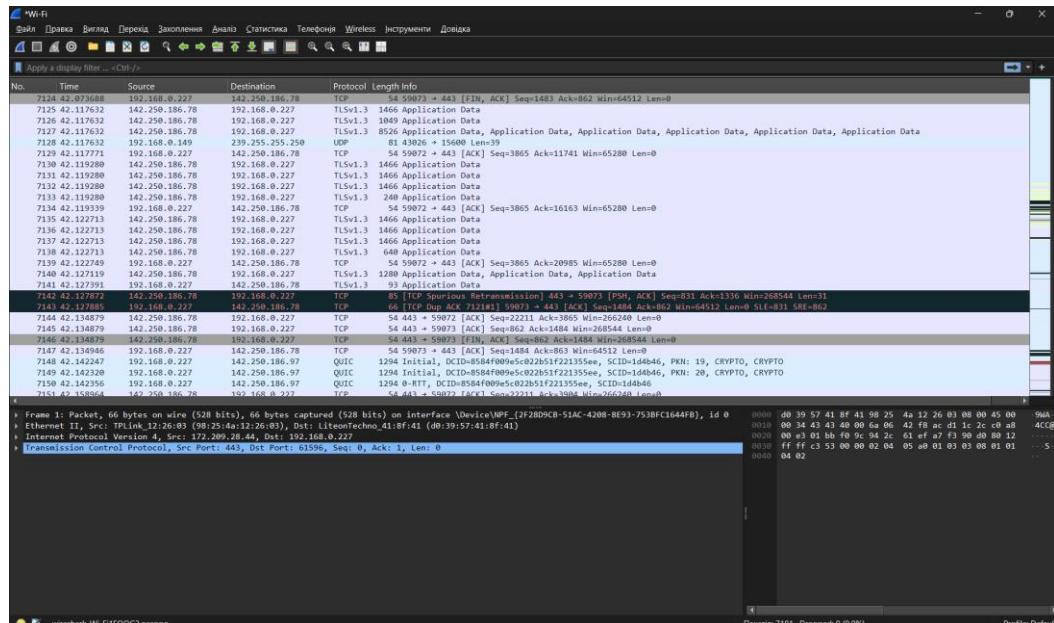
## Тема: “Протоколи прикладного рівня. Postman”

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

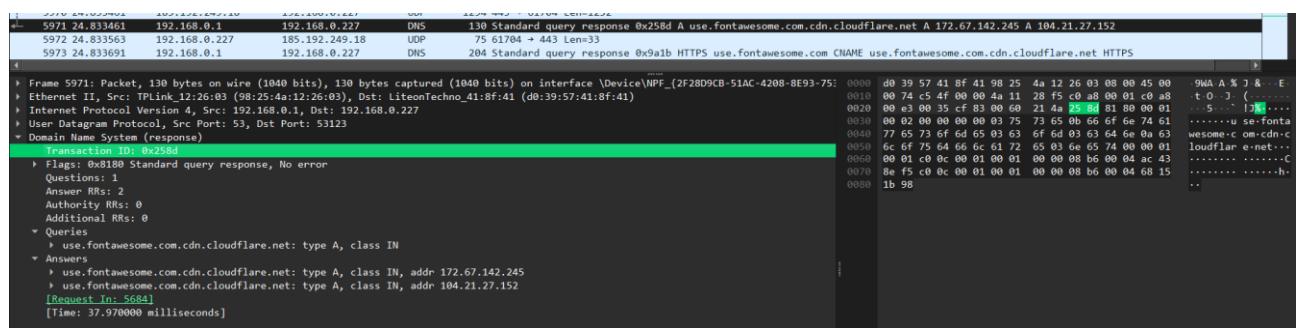
## Хід роботи

### Частина 1. Wireshark

1. Використовуючи Wireshark, захопив пакети, здійснюючи активність в браузері, а саме, на сайті, який працює за протоколом http.



2. Відшукав серед захоплених пакетів пару DNS-запит/відповідь:



**Transaction ID:** Унікальний ідентифікатор, який зв’язує цю відповідь із DNS-запитом (тобто сервер відповідає саме на той запит, який робив клієнт).

**Flags:** Вказує, що це відповідь на запит, і вона успішна (помилок немає).

**Questions:** Клієнт робив один DNS-запит.

**Answer RRs:** Сервер повернув дві відповіді.

**Query name:** Це домен, IP-адресу якого клієнт намагався отримати.

**Answers:** У відповіді містяться дві А-записи (IPv4-адреси)

3. За допомогою фільтру http відобразив відповідні пакети:

No.	http	e	Source	Destination	Protocol	Length info
5653	24, 72, 07695	192.168.0.227	69.43.111.82		HTTP	460 GET / (text/html)
5654	24, 411759	69.43.111.82	192.168.0.227		HTTP	758 HTTP/1.1 200 OK (text/html)
5642	24, 541676	192.168.0.227	69.43.111.82		HTTP	508 GET /grovesd/com224/notes/week2/links HTTP/1.1
5651	24, 686081	69.43.111.82	192.168.0.227		HTTP	1016 HTTP/1.1 200 OK (text/html)
5653	24, 72, 07695	192.168.0.227	69.43.111.82		HTTP	460 GET /grovesd/com224/css/notes.css HTTP/1.1
6197	24, 864128	192.168.0.227	69.43.111.82		HTTP	443 GET /grovesd/com224/js/prism.js HTTP/1.1
6208	24, 865061	69.43.111.82	192.168.0.227		HTTP	1482 HTTP/1.1 200 OK (text/css)
6262	24, 977995	69.43.111.82	192.168.0.227		HTTP	373 HTTP/1.1 200 OK (application/javascript)
6266	24, 978402	192.168.0.227	69.43.111.82		HTTP	524 GET /grovesd/com224/notes/week2/url-params.jpg HTTP/1.1
6267	24, 978627	192.168.0.227	69.43.111.82		HTTP	534 GET /grovesd/com224/notes/week2/rb-site-structure.png HTTP/1.1
6342	25, 105693	192.168.0.227	69.43.111.82		HTTP	542 GET /grovesd/com224/notes/week2/recipebook-site-structure.png HTTP/1.1
6350	25, 117727	69.43.111.82	192.168.0.227		HTTP	395 HTTP/1.1 200 OK (image/png)
6366	25, 117727	69.43.111.82	192.168.0.227		HTTP	395 HTTP/1.1 200 OK (JPEG image)
6379	25, 218985	69.43.111.82	192.168.0.227		HTTP	464 HTTP/1.1 200 OK (PNG)
6381	25, 226059	192.168.0.227	69.43.111.82		HTTP	492 GET /favicon.ico HTTP/1.1
6382	25, 343713	69.43.111.82	192.168.0.227		HTTP	458 HTTP/1.1 404 Not Found (text/html)
6841	36, 030166	192.168.0.227	149.154.167.216		HTTP	318 POST /api HTTP/1.1 (application/x-www-form-urlencoded)

4. Виділив текст HTTP-запиту та HTTP-відповіді та інтерпретував важливу інформацію, яка у них міститься:

No.	Time	Source	Destination	Protocol	Length Info
5634	24.267507	192.168.0.227	69.43.111.82	HTTP	390 GET / HTTP/1.1
5636	24.411759	69.43.111.82	192.168.0.227	HTTP	750 HTTP/1.1 200 OK (text/html)
5642	24.541676	192.168.0.227	69.43.111.82	HTTP	508 GET /~grovesd/comm244/notes/week2/links HTTP/1.1
5651	24.686081	69.43.111.82	192.168.0.227	HTTP	1016 HTTP/1.1 200 OK (text/html)
5653	24.720659	192.168.0.227	69.43.111.82	HTTP	460 GET /~grovesd/comm244/css/notes.css HTTP/1.1
6197	24.864128	192.168.0.227	69.43.111.82	HTTP	443 GET /~grovesd/comm244/js/prism.js HTTP/1.1
6281	24.865061	69.43.111.82	192.168.0.227	HTTP	1482 HTTP/1.1 200 OK (text/css)
6262	24.977990	69.43.111.82	192.168.0.227	HTTP	373 HTTP/1.1 200 OK (application/javascript)
6266	24.978402	192.168.0.227	69.43.111.82	HTTP	526 GET /~grovesd/comm244/notes/week2/url-parts.jpg HTTP/1.1
6267	24.978627	192.168.0.227	69.43.111.82	HTTP	534 GET /~grovesd/comm244/notes/week2/rb-site-structure.png HTTP/1.1
6342	25.105603	192.168.0.227	69.43.111.82	HTTP	542 GET /~grovesd/comm244/notes/week2/recipebook-site-structure.png HTTP/1.1
6350	25.117727	69.43.111.82	192.168.0.227	HTTP	628 HTTP/1.1 200 OK (PNG)
6366	25.117727	69.43.111.82	192.168.0.227	HTTP	395 HTTP/1.1 200 OK (JPEG/JIF image)
6379	25.218985	69.43.111.82	192.168.0.227	HTTP	464 HTTP/1.1 200 OK (PNG)
6381	25.226059	192.168.0.227	69.43.111.82	HTTP	492 GET /favicon.ico HTTP/1.1
6382	25.343713	69.43.111.82	192.168.0.227	HTTP	450 HTTP/1.1 404 Not Found (text/html)
6841	36.030166	192.168.0.227	149.154.167.216	HTTP	318 POST /api HTTP/1.1 (application/x-www-form-urlencoded)

## HTTP-запит (GET):

- Клієнт (браузер) виконує GET-запит на завантаження CSS-файлу.
  - Host: web.simmons.edu — домен, до якого виконується звернення.
  - User-Agent — інформація про браузер і ОС.
  - Accept: text/css — браузер очікує отримати CSS-контент.
  - Referer — сторінка, з якої йде завантаження стилів.
  - Connection: keep-alive — TCP-з'єднання буде збережене для наступних запитів.

No.	Time	Source	Destination	Protocol	Length Info
5634	24.267507	192.168.0.227	69.43.111.82	HTTP	390 GET / HTTP/1.1
5636	24.411759	69.43.111.82	192.168.0.227	HTTP	750 HTTP/1.1 200 OK (text/html)
5642	24.541676	192.168.0.227	69.43.111.82	HTTP	508 GET /~grovesd/comm244/notes/week2/links HTTP/1.1
5651	24.686081	69.43.111.82	192.168.0.227	HTTP	1016 HTTP/1.1 200 OK (text/html)
5653	24.720659	192.168.0.227	69.43.111.82	HTTP	460 GET /~grovesd/comm244/css/notes.css HTTP/1.1
6197	24.864128	192.168.0.227	69.43.111.82	HTTP	443 GET /~grovesd/comm244/js/prism.js HTTP/1.1
6201	24.865061	69.43.111.82	192.168.0.227	HTTP	1482 HTTP/1.1 200 OK (text/css)
6262	24.977990	69.43.111.82	192.168.0.227	HTTP	373 HTTP/1.1 200 OK (application/javascript)
6266	24.978402	192.168.0.227	69.43.111.82	HTTP	526 GET /~grovesd/comm244/notes/week2/url-parts.jpg HTTP/1.1
6267	24.978627	192.168.0.227	69.43.111.82	HTTP	534 GET /~grovesd/comm244/notes/week2/rb-site-structure.png HTTP/1.1
6342	25.105603	192.168.0.227	69.43.111.82	HTTP	542 GET /~grovesd/comm244/notes/week2/recipebook-site-structure.png HTTP/1.1
6350	25.117727	69.43.111.82	192.168.0.227	HTTP	628 HTTP/1.1 200 OK (PNG)
6366	25.117727	69.43.111.82	192.168.0.227	HTTP	395 HTTP/1.1 200 OK (JPEG JFIF image)
6379	25.218985	69.43.111.82	192.168.0.227	HTTP	464 HTTP/1.1 200 OK (PNG)
6381	25.226059	192.168.0.227	69.43.111.82	HTTP	492 GET /favicon.ico HTTP/1.1
6382	25.343713	69.43.111.82	192.168.0.227	HTTP	450 HTTP/1.1 404 Not Found (text/html)
6841	36.030166	192.168.0.227	149.154.167.216	HTTP	318 POST /api HTTP/1.1 (application/x-www-form-urlencoded)

```

Frame 6201: Packet, 1482 bytes on wire (11856 bits), 1482 bytes captured (11856 bits) on interface \Device\NPF_{2F28D9CB-51AC-4208-8E95-000000000000
Ethernet II, Src: TPLink_12:26:03 (98:25:4a:12:26:03), Dst: LiteonTechno_41:8f:41 (d0:39:57:41:8f:41)
Internet Protocol Version 4, Src: 69.43.111.82, Dst: 192.168.0.227
Transmission Control Protocol, Src Port: 80, Dst Port: 59055, Seq: 19588, Ack: 861, Len: 1428
[3 Reassembled TCP Segments (11648 bytes): #6193(1460), #6200(8760), #6201(1428)]
HyperText Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Date: Fri, 21 Nov 2025 18:28:43 GMT\r\n
  Server: Apache\r\n
  Last-Modified: Tue, 01 Oct 2019 21:50:40 GMT\r\n
  ETag: "2c72-593e057e2ec00"\r\n
  Accept-Ranges: bytes\r\n
  Content-Length: 11378\r\n
  Keep-Alive: timeout=5, max=99\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/css\r\n
  \r\n
[Request in frame: 5653]
[Time since request: 144.402000 milliseconds]
[Request URI: /~grovesd/comm244/css/notes.css]
[Full request URI: http://web.simmons.edu/~grovesd/comm244/css/notes.css]
File Data: 11378 bytes
Line-based text data: text/css (492 lines)

```

## HTTP-відповідь:

- 200 OK — файл успішно знайдено та передано.
- Content-Type: text/css — сервер повертає CSS-стилі.
- Content-Length: 11378 — розмір файлу у байтах.
- Last-Modified і ETag — дозволяють браузеру кешувати файл.
- Server: Apache — сервер, що обробив запит.

No.	Time	Source	Destination	Protocol	Length Info
5634	24.267507	192.168.0.227	69.43.111.82	HTTP	390 GET / HTTP/1.1
5636	24.411759	69.43.111.82	192.168.0.227	HTTP	750 HTTP/1.1 200 OK (text/html)
5642	24.541676	192.168.0.227	69.43.111.82	HTTP	508 GET /~grovesd/comm244/notes/week2/links HTTP/1.1
5651	24.686081	69.43.111.82	192.168.0.227	HTTP	1016 HTTP/1.1 200 OK (text/html)
5653	24.720659	192.168.0.227	69.43.111.82	HTTP	460 GET /~grovesd/comm244/css/notes.css HTTP/1.1
6197	24.864128	192.168.0.227	69.43.111.82	HTTP	443 GET /~grovesd/comm244/js/prism.js HTTP/1.1
6201	24.865061	69.43.111.82	192.168.0.227	HTTP	1482 HTTP/1.1 200 OK (text/css)
6262	24.977990	69.43.111.82	192.168.0.227	HTTP	373 HTTP/1.1 200 OK (application/javascript)
6266	24.978402	192.168.0.227	69.43.111.82	HTTP	526 GET /~grovesd/comm244/notes/week2/url-parts.jpg HTTP/1.1
6267	24.978627	192.168.0.227	69.43.111.82	HTTP	534 GET /~grovesd/comm244/notes/week2/rb-site-structure.png HTTP/1.1
6342	25.105603	192.168.0.227	69.43.111.82	HTTP	542 GET /~grovesd/comm244/notes/week2/recipebook-site-structure.png HTTP/1.1
6350	25.117727	69.43.111.82	192.168.0.227	HTTP	628 HTTP/1.1 200 OK (PNG)
6366	25.117727	69.43.111.82	192.168.0.227	HTTP	395 HTTP/1.1 200 OK (JPEG JFIF image)
6379	25.218985	69.43.111.82	192.168.0.227	HTTP	464 HTTP/1.1 200 OK (PNG)
6381	25.226059	192.168.0.227	69.43.111.82	HTTP	492 GET /favicon.ico HTTP/1.1
6382	25.343713	69.43.111.82	192.168.0.227	HTTP	450 HTTP/1.1 404 Not Found (text/html)
6841	36.030166	192.168.0.227	149.154.167.216	HTTP	318 POST /api HTTP/1.1 (application/x-www-form-urlencoded)

```

Frame 6841: Packet, 318 bytes on wire (2544 bits), 318 bytes captured (2544 bits) on interface \Device\NPF_{2F28D9CB-51AC-4208-8E95-000000000000
Ethernet II, Src: LiteonTechno_41:8f:41 (d0:39:57:41:8f:41), Dst: TPLink_12:26:03 (98:25:4a:12:26:03)
Internet Protocol Version 4, Src: 192.168.0.227, Dst: 149.154.167.216
Transmission Control Protocol, Src Port: 59067, Dst Port: 80, Seq: 229, Ack: 1, Len: 264
[2 Reassembled TCP Segments (492 bytes): #6837(228), #6841(264)]
HyperText Transfer Protocol
  POST /api HTTP/1.1\r\n
  Host: 149.154.167.216:80\r\n
  Content-Type: application/x-www-form-urlencoded\r\n
  Content-Length: 264\r\n
  Connection: Keep-Alive\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: uk-UA,en,*\r\n
  User-Agent: Mozilla/5.0\r\n
  \r\n
[Full request URI: http://149.154.167.216:80/api]
File Data: 264 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded

```

## HTTP-запит POST:

- POST — використовується для надсилання даних на сервер (форм, API-викликів).
- Content-Type: application/x-www-form-urlencoded — формат HTML-форми.
- Content-Length: 264 — показує обсяг переданих даних.

5. Які типи запитів, крім GET, були захоплені? Що вони означають? (Якщо інших типів запитів немає — поясніть їхню відсутність):

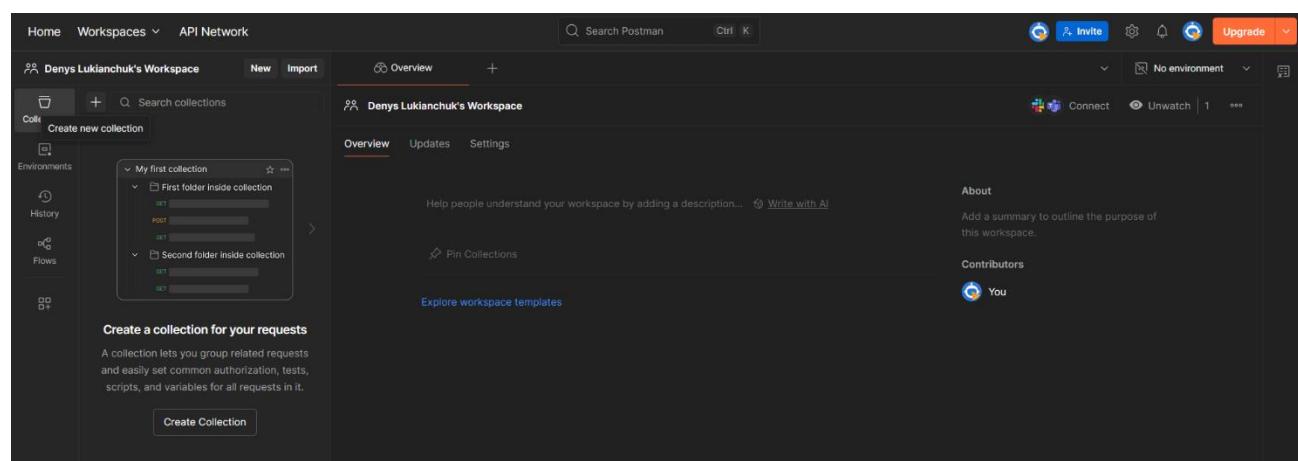
Крім GET був захоплений ще запит POST. Він використовується для надсилання інформації на сервер (дані форми або API-запиту).

Інші типи, такі як PUT, DELETE, OPTIONS, PATCH, не були захоплені, оскільки під час перегляду звичайних веб-сторінок браузер майже завжди використовує лише GET, а інколи POST. Інші методи застосовуються лише при роботі з API або при зміні даних на сервері.

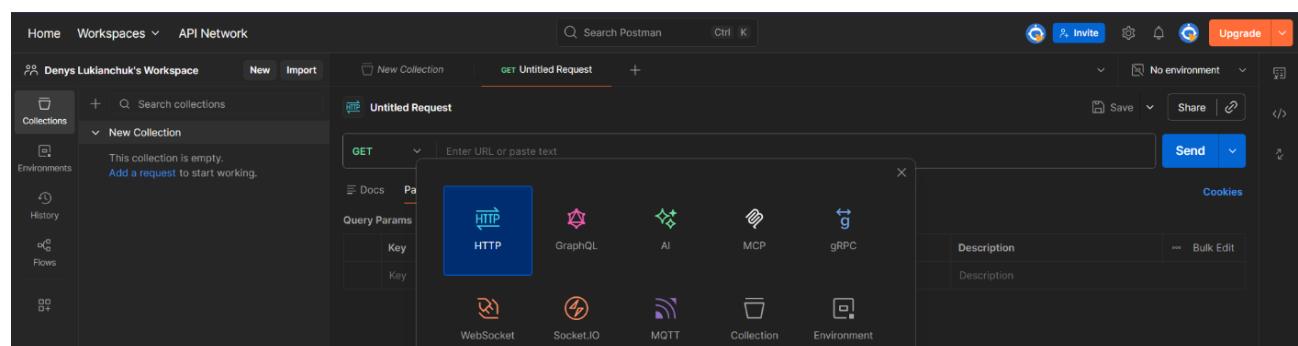
## Частина 2. Postman

6. Запустив веб-версію Postman та створив новий персональний робочий простір.

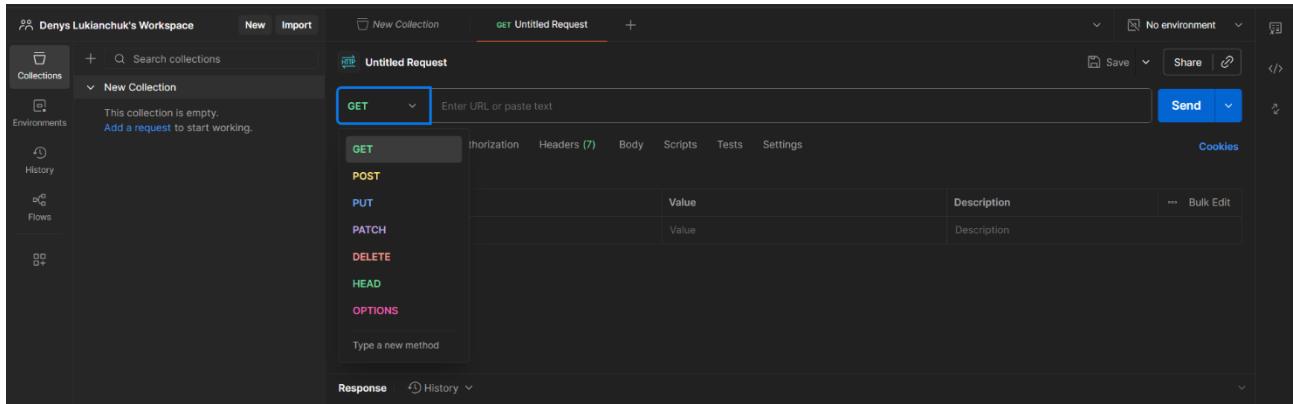
7. Створив колекцію:



8. Натиснув на кнопку New і оберав тип запиту HTTP:

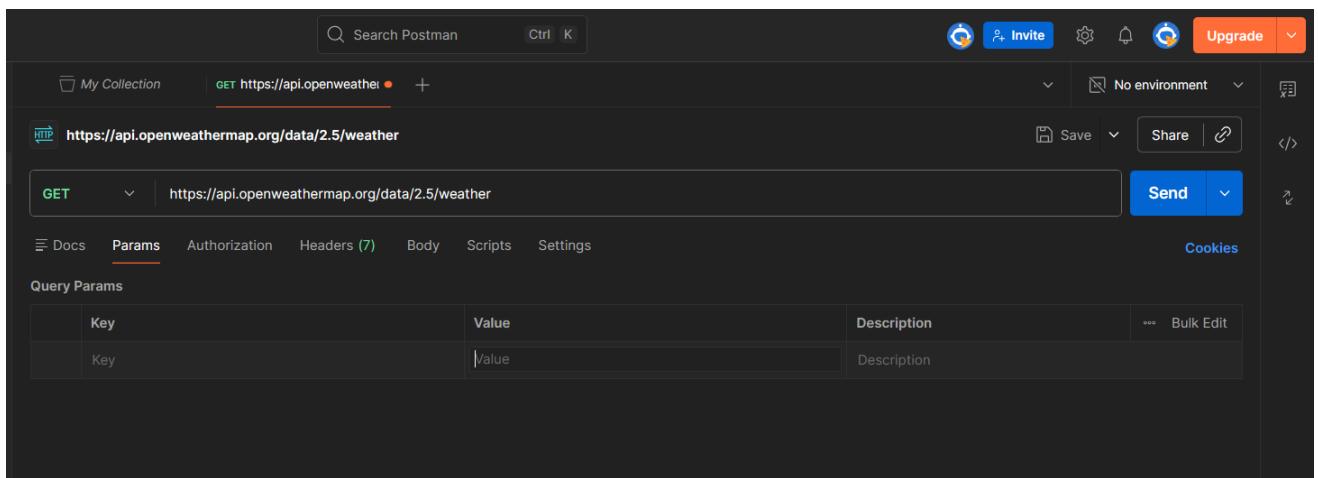


### 9. З випадаючого списку оберав метод GET:



The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections', 'Environments', 'History', and 'Flows'. The main area is titled 'Untitled Request' under 'New Collection'. A sidebar on the right lists HTTP methods: GET, POST, PUT, PATCH, DELETE, HEAD, and OPTIONS. The 'GET' method is selected. Below the methods is a text input field labeled 'Enter URL or paste text'. At the bottom, there are tabs for 'Response' and 'History'.

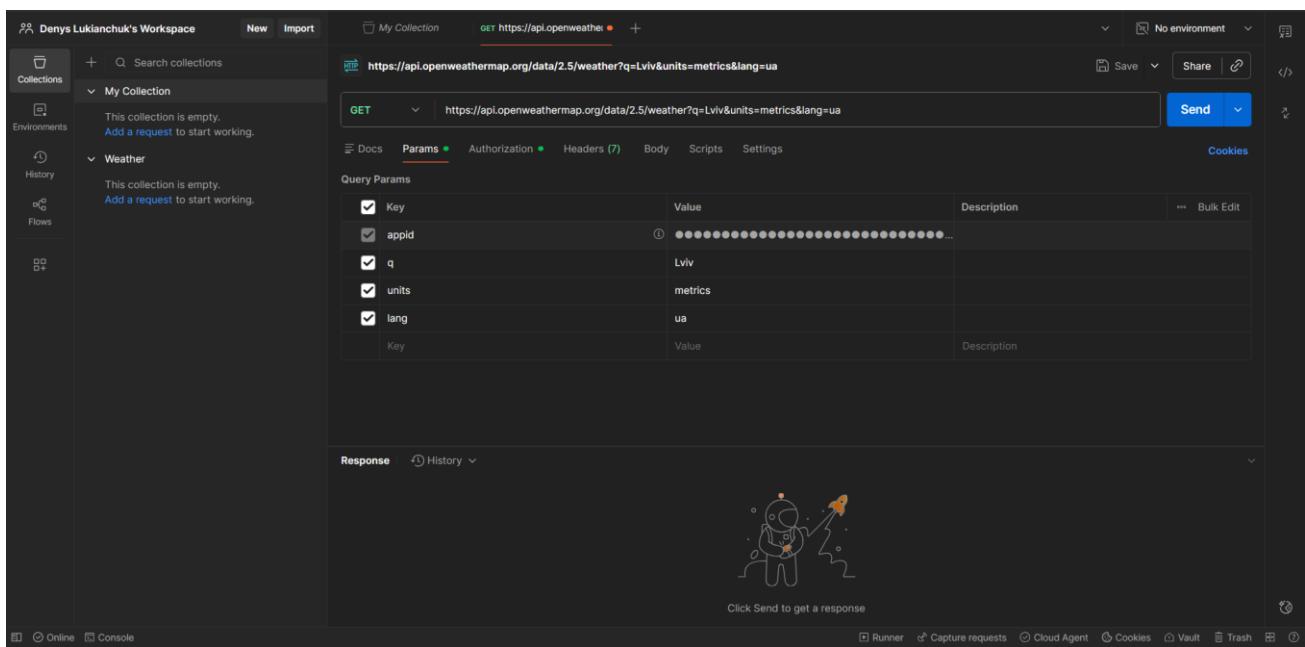
### 10. Ввів адресу кінцевої точки <https://api.openweathermap.org/data/2.5/weather> у поле Enter URL or paste text:



The screenshot shows the Postman application interface. The URL in the main input field is 'https://api.openweathermap.org/data/2.5/weather'. Below it, under 'Params', there is a table with one row: 'Key' (Value) and 'q' (Value 'Lviv'). Other tabs like 'Docs', 'Headers', 'Body', 'Scripts', and 'Settings' are visible at the top and bottom of the interface.

### 11. Визначився з типом API-запиту про поточну погоду, за назвою міста.

### 12. Перейшов на вкладку Params і задав імена та значення параметрів:



The screenshot shows the Postman application interface. The URL in the main input field is 'https://api.openweathermap.org/data/2.5/weather?q=Lviv&units=metrics&lang=ua'. Below it, under 'Params', there is a table with five rows: 'appid' (Value '1234567890'), 'q' (Value 'Lviv'), 'units' (Value 'metrics'), 'lang' (Value 'ua'), and 'Key' (Value 'Value'). The 'Params' tab is currently selected. Other tabs like 'Docs', 'Headers', 'Body', 'Scripts', and 'Settings' are visible at the top and bottom of the interface.

13. На вкладці Authorization задав параметри(тип авторизації, назву параметра у запиті, значення свого авторизаційного ключа і місце авторизаційної інформації у API-запиті):

The screenshot shows the Postman application interface. At the top, the URL is set to <https://api.openweathermap.org/data/2.5/weather?q=Lviv&units=metrics&lang=ua>. Below the URL, the method is selected as GET. The 'Authorization' tab is active, showing 'Auth Type' set to 'API Key'. The 'Key' field contains 'appid' and the 'Value' field contains a redacted API key. The 'Add to' dropdown is set to 'Query Params'. Other tabs like 'Docs', 'Params', 'Headers', 'Body', 'Scripts', and 'Settings' are visible. The bottom section shows 'Response' and 'History'.

14. Виконав запит, натиснувши кнопку Send і ознайомився з його результатами:

The screenshot shows the Postman interface with the 'Body' tab selected. The response body is displayed as JSON. The JSON structure includes coordinates, weather conditions, base information, and main weather parameters. The 'JSON' dropdown shows the raw JSON code, which is also pasted below:

```
1 {  
2     "coord": {  
3         "lon": 24.0232,  
4         "lat": 49.8383  
5     },  
6     "weather": [  
7         {  
8             "id": 804,  
9             "main": "Clouds",  
10            "description": "хмарно",  
11            "icon": "04n"  
12        }  
13    ],  
14    "base": "stations",  
15    "main": {  
16        "temp": 277.04,  
17        "feels_like": 274.17,  
18        "temp_min": 277.04,  
19        "temp_max": 277.04,  
20        "pressure": 1007,  
21        "humidity": 85,  
22        "sea_level": 1007,  
23        "grnd_level": 970  
24    },  
25    "visibility": 10000,  
26    "wind": {
```

```

27     "speed": 3.2,
28     "deg": 178,
29     "gust": 5.93
30   },
31   "clouds": {
32     "all": 100
33   },
34   "dt": 1764169793,
35   "sys": {
36     "country": "UA",
37     "sunrise": 1764136393,
38     "sunset": 1764167375
39   },
40   "timezone": 7200,
41   "id": 702550,
42   "name": "Lviv",
43   "cod": 200
44 }

```

Після натискання кнопки Send я отримав відповідь від API у форматі JSON. Вона містить повну інформацію про поточну погоду в місті Львів: координати міста, опис погоди (“хмарно”), температуру повітря, відчутну температуру, вологість, атмосферний тиск, швидкість та напрям вітру, рівень хмарності, дані про країну, час сходу та заходу сонця. Код відповіді 200 означає, що запит був виконаний успішно.

15. Зберіг запит у колекцію, натиснувши кнопку Save, давши йому змістовне ім’я:

The screenshot shows the Postman interface with a collection named 'Lviv\_Current\_Weather'. The collection has one item named 'Lviv\_Current\_Weather\_Vizual' which is a GET request to 'https://api.openweathermap.org/data/2.5/weather?q=Lviv&units=metrics&lang=ua'. There are buttons for 'Save' and 'Send' at the top right of the item card.

16. Створив копію запиту.

17. Додав до переліку параметрів параметр mode зі значенням html, виконав запит і порівняв його результат із результатом попереднього запиту:

The screenshot shows the Postman interface with a modified query. The URL is now 'https://api.openweathermap.org/data/2.5/weather?q=Lviv&units=metrics&lang=ua&mode=html'. The 'Params' tab is selected, showing the following parameters:

Key	Value	Description	Bulk Edit
appid	XXXXXXXXXXXXXX		
q	Lviv		
units	metrics		
lang	ua		
mode	html		
Key	Value	Description	

## Результат:

The screenshot shows the Postman interface with a successful API call to `https://api.openweathermap.org/data/2.5/weather?q=Lviv&units=metrics&lang=ua&mode=html`. The response status is 200 OK. The preview pane displays the HTML output for Lviv, which includes a weather icon, the temperature (3.89°C), cloudiness (100%), humidity (61%), wind (3.2 m/s), pressure (1007 hPa), and a link to more details.

Після додавання параметра mode=html API OpenWeatherMap повернув відповідь у форматі HTML замість JSON. HTML-версія містить візуальне оформлення: іконку погоди, температуру великим шрифтом, показники вологості, хмарності, тиску та вітру. Такий формат зручний для перегляду людиною.

Попередня відповідь у форматі JSON містила ті самі дані, але у вигляді структурованих полів, які призначені для подальшої обробки програмами. Таким чином, параметр mode=html змінює тип виводу зі структурованих даних на готову візуальну веб-сторінку.

18. Добав і виконав запит для отримання інформації про інший прогноз погоди з іншого погодного сервіс:

The screenshot shows the Postman interface with a successful API call to `https://api.met.no/weatherapi/locationforecast/2.0/compact?lat=49.8383&lon=24.0232`. The response status is 200 OK. The preview pane displays the JSON output, which includes a feature object with geometry, coordinates, properties (meta, updated\_at, units, air\_pressure\_at\_sea\_level, air\_temperature, cloud\_area\_fraction, and more), and a link to more details.

Для виконання додаткового запиту я використав інший погодний сервіс, який працює за координатами. У цьому API місто задається не через назву, як у OpenWeatherMap, а через географічні координати — широту (**lat**) і довготу (**lon**). Для Львова це lat=49.8383 та lon=24.0232.

У відповідь сервіс повернув актуальні погодні дані у форматі JSON: температуру, хмарність, швидкість вітру та інші параметри. API працює без авторизаційного ключа, що дозволяє використовувати його одразу.

## Висновок

У ході лабораторної роботи я навчився аналізувати мережевий трафік у Wireshark, визначати DNS-запити та відповіді, переглядати HTTP-запити й відповіді та розрізняти типи методів, що використовуються під час завантаження веб-сторінок. Я закріпив розуміння того, як браузер обмінюється даними з сервером.

У другій частині роботи я навчався формувати HTTP-запити в Postman, додавати параметри, використовувати API-ключі та отримувати дані від погодних сервісів у форматах JSON і HTML. Також я виконав додатковий запит за координатами й закріпив навички роботи з API та аналізу відповідей.