

# Lahore Garrison

# University

## DHA Phase 6 Lahore

### Lab Task 2



<b>Student Name</b>	Ateeb Qaiser
<b>Student Roll No</b>	Fa23/BSCS/279
<b>Section</b>	G
<b>Subject</b>	Web Technologies Lab
<b>Instructor Name</b>	Mr. M Yousaf

## Lab 2: HTTP, URLs and Request/Response Basics Objectives

- Understand how HTTP works by observing request/response cycles.
- Use browser DevTools and Postman/curl to inspect headers and status codes.
- Simulate caching and status code behavior.

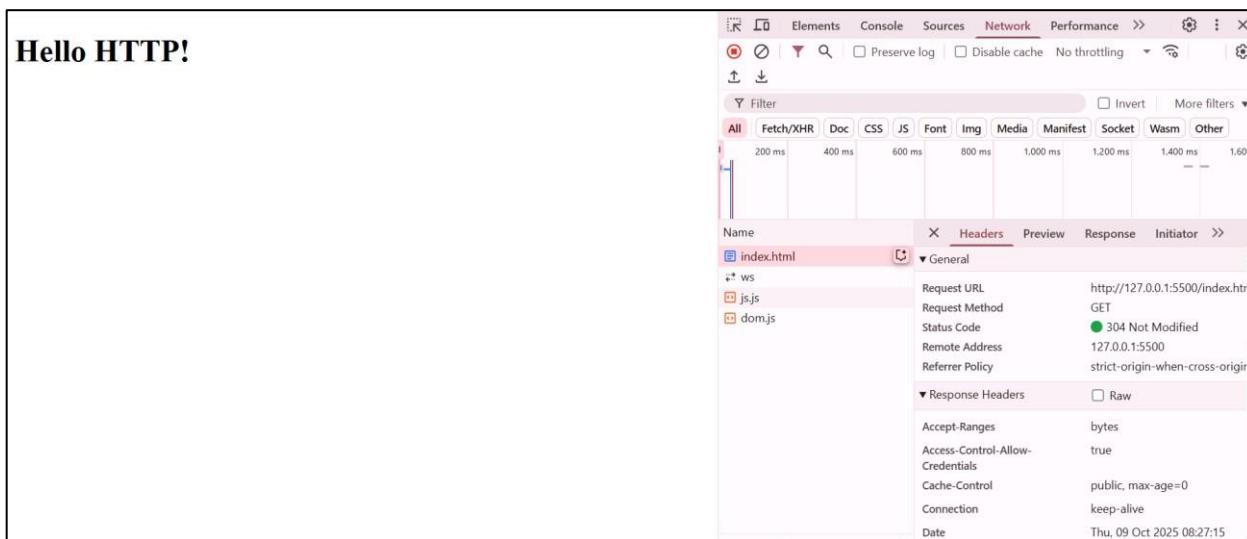
### Tools Required:

- Browser DevTools (Chrome, Firefox, Edge)
- Postman or curl (command-line tool)
- Live Server (VS Code extension or any static server)

### Tasks

#### Step 1: Inspect HTTP Requests with DevTools

1. Open any simple HTML page (e.g., index.html) using Live Server.
2. Right-click → Inspect → go to Network tab.
3. Reload the page (F5) to capture network activity.
4. Click on the main request (e.g., index.html) and observe:
  - Request Method (GET) o Status Code (e.g., 200 OK)
  - Request Headers (Host, User-Agent)
  - Response Headers (Content-Type, Cache-Control)



## Step 2: Make GET/POST Requests with Postman or curl Using Postman

1. Open Postman and create a GET request to your local server (e.g., <http://localhost:5500/index.html>).
2. Observe the status code, response body, and headers.
3. Create a POST request to a test endpoint (e.g., <https://httpbin.org/post>).

### Add a JSON body:

- { "name": "Student", "lab": "HTTP Basics" }
- Send the request and inspect the response.

### Using curl (optional)

```
curl -X POST https://httpbin.org/post -H "Content-Type: application/json" -d
'{"name":"Student","lab":"HTTP Basics"}
```

The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:5500/index.html`. The response status is `200 OK` with a duration of `12 ms` and a size of `2.04 KB`. The response body is displayed as HTML, showing the structure of the `index.html` file.

The screenshot shows the Postman interface for a POST request to `https://httpbin.org/post`. The response status is `200 OK` with a duration of `8.45 s` and a size of `898 B`. The response body is displayed as JSON, showing the original payload and the server's response.

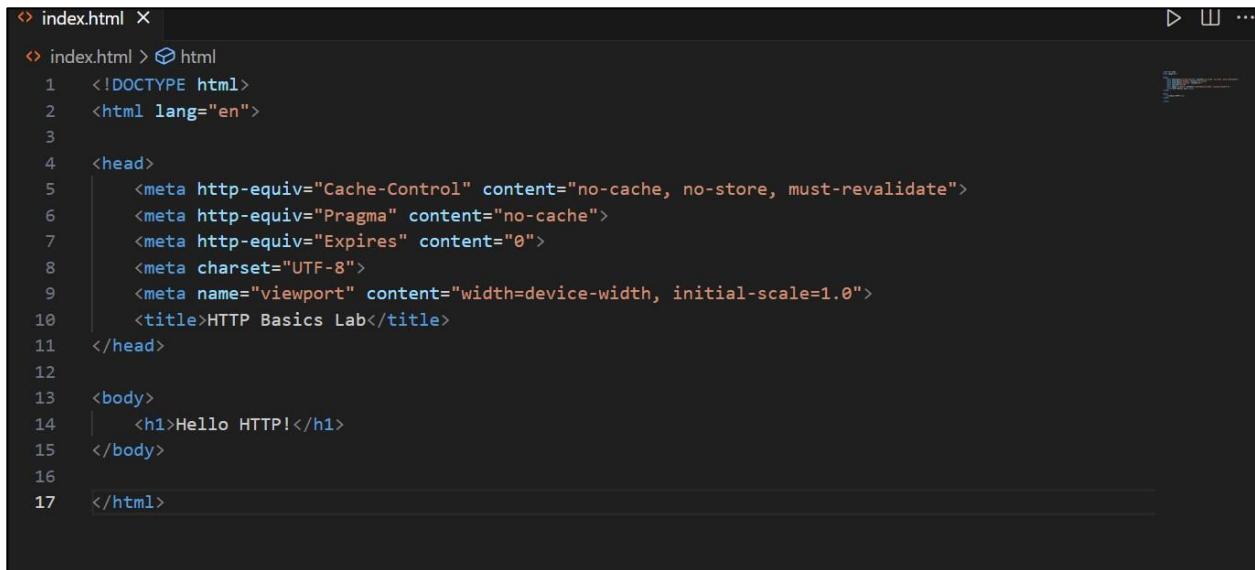
## Step 3: Simulate Caching and Status Codes

### Modify HTML to simulate caching

1. Add meta tags to your HTML <head>:
2. <meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">
3. <meta http-equiv="Pragma" content="no-cache">
4. <meta http-equiv="Expires" content="0">
5. Reload the page and observe changes in **response headers**. **Simulate Status Codes** Use these test URLs:

- **200 OK:** <https://httpbin.org/status/200>
- **301 Redirect:** <https://httpbin.org/status/301>
- **404 Not Found:** <https://httpbin.org/status/404>

**Screenshot Tip:** Capture each status code response in DevTools or Postman.



```
index.html X
index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">
6    <meta http-equiv="Pragma" content="no-cache">
7    <meta http-equiv="Expires" content="0">
8    <meta charset="UTF-8">
9    <meta name="viewport" content="width=device-width, initial-scale=1.0">
10   <title>HTTP Basics Lab</title>
11 </head>
12
13 <body>
14   <h1>Hello HTTP!</h1>
15 </body>
16
17 </html>
```

The screenshot shows the Postman application interface. At the top, there's a toolbar with icons for HTTP methods (Get, Post, Put, Delete), a URL input field containing "https://httpbin.org/status/200", and a "Save" button. Below the toolbar is a navigation bar with "GET" selected, followed by a dropdown menu and the URL again. To the right of the URL is a "Send" button. The main workspace has tabs for "Params", "Authorization", "Headers (7)", "Body", "Scripts", "Tests", and "Settings", with "Params" currently active. A "Query Params" section is present. Below it is a table with columns "Key", "Value", and "Description". Under the "Body" tab, the response status is shown as "200 OK" with a green background, along with response time (6.87 s) and size (236 B). Below the status are buttons for "Raw", "Preview", and "Visualize". The response body itself is a simple number "1".

The screenshot shows the Postman application interface. At the top, there are tabs for 'Overview' (selected), 'GET https://httpbin.org/status/404', 'Post https://httpbin.org/gpu', and 'No environments'. Below the tabs, the URL 'https://httpbin.org/status/404' is entered in the address bar, along with 'Save' and 'Share' buttons.

The main workspace shows a 'GET' request with the URL 'https://httpbin.org/status/404'. Below the request, tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Scripts', 'Tests', 'Settings', and 'Cookies' are visible. The 'Cookies' tab is selected.

A 'Query Params' table is present, with columns 'Key', 'Value', and 'Description'. It contains one row with 'Key' as 'key' and 'Value' as 'value'.

At the bottom, there are tabs for 'Body', 'Cookies', 'Headers (7)', and 'Test Results'. The 'Test Results' tab is selected, displaying the response status '404 NOT FOUND', execution time '11.70 s', size '243 B', and a copy icon. Below this, there are buttons for 'Raw', 'Preview', 'Debug with AI', and a search/filter icon.

The screenshot shows the Postman interface with a GET request to <https://httpbin.org/status/301>. The response is a 200 OK status with the following details:

- Body:** Raw, Preview, Visualize
- Headers:** Content-Type: application/json; charset=utf-8, Date: Mon, 12 Dec 2022 10:45:21 GMT, Connection: keep-alive, Transfer-Encoding: chunked
- Test Results:** 943 ms, 686 B

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

## **Deliverables**

Create a Markdown file or short report with:

### **Required Sections**

#### **Screenshots of:**

- DevTools request inspection
- One GET and one POST request (Postman or curl)
- Status code simulations (2xx, 3xx, 4xx)

#### **Explanations:**

- Identify parts of request/response (method, headers, status)
- Describe what caching headers do
- Explain the meaning of each status code

## **CONCLUSION:**

**This lab helped me understand how HTTP works by observing real-time request and response cycles using DevTools and Postman. I also learned how caching headers and different status codes affect web communication between client and server.**

## **RUBRICS:**

Performance			Lab Report		
Description	Total Marks	Marks Obtained	Description	Total Marks	Marks Obtained
Ability to Conduct practical	5		Structure	5	
Data Analysis & Interpretation	5		Efficiency	5	
Total Marks obtained			Total Marks Obtained		

Instructor Signature \_\_\_\_\_