

# 2322 Project Report

Name: Cao Kaihan

NetID: 23097881D

## I . General introduction

This program implements multi-threaded HTTP server in Python that handles GET and HEAD requests with support for keep-alive connections and conditional requests (If-Modified-Since). The server follows basic HTTP/1.1 protocol standards and includes proper error handling.

## II. Key features and Design

### 1. Multi-threaded Architecture:

- Creates a new thread for each client connection
- Maintains a list of active connections
- Handles concurrent requests efficiently

### 2. HTTP Method Support:

- Fully implements GET and HEAD methods
- Returns appropriate status codes (200, 304, 400, 404)
- Handles conditional requests with If-Modified-Since header

### 3. Connection Management:

- Supports both keep-alive and close connection modes
- Implements timeout for keep-alive connections (10 seconds)
- Properly cleans up resources on shutdown

### 4. Content Handling:

- Serves both text (HTML) and binary (images) content
- Automatically detects content type from file extensions
- Includes proper Content-Length and Last-Modified headers

## 5. Logging:

- Maintains a server log (server\_log.txt) with timestamped records
- Logs client address, requested file, and response status

## III. Execution and Screen Capture

### 1. Start listening to ports

```
C:\Users\LEGION\PycharmProjects\COMP2322Project>python webserver.py

The HTTP server is now running on port 8000
```

After executing the program, you can see that our server is running on the port 8000.

### 2. Entering the link in the browser



After entering this, you can see the index.html page, which is the default page.

---

## Index.html page

```
2025-04-27 23:09:27.415363: client('127.0.0.1', 49584) requested file /test.html with response 200 OK
```

The record now is generated by the program, into the log file.

### 3. Testing not exist (404) case:

Entering <http://127.0.0.1:8000/not-exist.html>

You can see the server gives msg:

```
Server response:  
  
HTTP/1.1 404 Not Found  
Content-Length : 0  
Content-Type: text/plain
```

2025-04-28 12:36:05.013010: client('127.0.0.1', 61548) requested file /not-exist.html with response 404 Not Found

(This is what in the log file)

### 4. Testing other cases

Let's now enter other cases by entering

<http://127.0.0.1:8000/index2.html> in the browser. You will see a page like this.

---

#### To test Request command

- 1. Click link below to test **GET** request on **text and image file**:  
[test.html](http://127.0.0.1:8000/test.html)
- 2. Click button below to test **HEAD** request and show received header fields:

response message:

#### To test Response messages

- 3. Click button below to test **400 Bad Request** response:
- 4. Click button below to test **304 Not Modified** response:

response message:

After clicking the buttons, you can see their status code and msg.

---

#### To test Request command

- 1. Click link below to test **GET** request on **text and image file**:  
[test.html](#)
  - 2. Click button below to test **HEAD** request and show received header fields:
- response message: **Status: 200 OK; Type: text/html; Length: 654**

#### To test Response messages

- 3. Click button below to test **400 Bad Request** response:
  - 4. Click button below to test **304 Not Modified** response:
- response message: **Status: 400 Bad Request**
- response message: **Status: 304 Not Modified**