Problem No: 07

Problem Name: Test HTML Code and a Web Page with Previous Lab 05 and Lab 06 HTTP Server

Objective

To test and display an external HTML file using the HTTP servers implemented in Lab 05 and Lab 06.

Theory

In previous labs, a basic HTTP server was created using Python's `http.server` module. In this experiment, instead of embedding HTML inside the Python script, an external HTML file is served.

This demonstrates:

- Separation of backend and frontend
- Serving static HTML files
- File handling in Python
- Real-world web server behavior

When the browser sends a GET request:

- The server reads the HTML file
- Sends HTTP response headers
- Sends HTML file content as response body

Application

Serving HTML files is the foundation of:

- Static website hosting
- Web development
- Backend frameworks
- Production web servers

Implementation

Step 1: Create HTML File (index.html)

```
<!DOCTYPE html>
<html>
<head>
<title>Lab 07 Test Page</title>
<style>
body{font-family:Arial;text-align:center;background:#eef;}
h1{color:#333;}
</style>
</head>
<body>
<h1>Testing External HTML Page</h1>
<p>This page is served using Python HTTP Server.</p>
</body>
</html>
```

Step 2: Modify Previous HTTP Server

```
from http.server import HTTPServer, BaseHTTPRequestHandler

HOST = "127.0.0.1"
PORT = 8080

class FileHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header("Content-type", "text/html")
        self.end_headers()

        with open("index.html", "r") as file:
            content = file.read()
        self.wfile.write(content.encode("utf-8"))

if __name__ == "__main__":
    print(f"Server␣running␣at␣http://{HOST}:{PORT}")
    server = HTTPServer((HOST, PORT), FileHandler)
    server.serve_forever()
```

Execution Steps

1. Create `index.html` in the same folder.
2. Save Python file.
3. Run: `python server.py`
4. Open browser and visit: `http://127.0.0.1:8080`

Result

The HTTP server successfully reads the external HTML file and displays it in the browser, confirming correct integration between HTML frontend and Python backend.

Sample Output

Browser Display:

Testing External HTML Page

This page is served using Python HTTP Server.

Discussion

This experiment demonstrates serving static HTML files using a Python HTTP server. It shows practical implementation of client-server communication and separation of content from server logic.

Conclusion

The HTML page was successfully tested and served using the previously implemented HTTP servers. This experiment strengthens understanding of web hosting fundamentals and file-based content delivery.