Assignment 10

Implementation of TCP/UDP Socket Programming

NAME: Shirish Manoj Bobde

Reg. No.: 812

Roll No.: ECE/21152

Problem Statement

Implement a basic HTTP server in Python that listens on a specified port (e.g., port 8080). Create a HTML file (e.g., index.html, about.html) with some content. Implement the server to open the HTML file based on the request path. Test the server by accessing it through a web browser.  
**Additional Challenges  
(Optional):**

* Implement support for other HTTP methods (e.g., PUT, DELETE).
* Add support for serving static files (e.g., images, CSS).
* Implement basic authentication for accessing certain pages.
* Create a simple web application (e.g., a to-do list) that  
  interacts with the server using AJAX requests.

Codes

**Server**

from http.server import HTTPServer, SimpleHTTPRequestHandler

import base64

class CustomHandler(SimpleHTTPRequestHandler):

    def do\_GET(self):

        if self.path == "/":

            self.path = "/index.html"

        elif self.path == "/about.html":

            auth\_header = self.headers.get('Authorization')

            if auth\_header is None or 'Basic ' not in auth\_header:

                self.send\_response(401)

                self.send\_header('WWW-Authenticate', 'Basic realm="Secure Area"')

                self.end\_headers()

                return

            else:

                auth\_decoded = base64.b64decode(auth\_header.split(' ')[1]).decode('utf-8')

                username, password = auth\_decoded.split(':')

                if username =='admin' and password == '12345':

                    self.path = "/about.html"

                else:

                    self.send\_response(401)

                    self.send\_header('WWW-Authenticate', 'Basic realm="Secure Area"')

                    self.end\_headers()

                    return

        return super().do\_GET()

port = 8888

server\_address = ("", port)

httpd = HTTPServer(server\_address, CustomHandler)

print(f"Server is running on port {port}")

httpd.serve\_forever()

**Client**

import webbrowser

from http.server import HTTPServer, SimpleHTTPRequestHandler

class CustomHandler(SimpleHTTPRequestHandler):

    def do\_GET(self):

        if self.path == "/":

            self.path = "/index.html"

        return super().do\_GET()

port = 8888

server\_address = ("", port)

httpd = HTTPServer(server\_address, CustomHandler)

print(f"Server port: {port}")

# Continuously open pages

while True:

    # Ask the user for input

    page = input("Enter the page you want to open (or type 'quit' to exit): ")

    if page.lower() == 'quit':

        break

    # Open the browser to the specified page

    webbrowser.open(f"http://localhost:{port}/{page}")

httpd.server\_close()

**index.html**

import webbrowser

from http.server import HTTPServer, SimpleHTTPRequestHandler

class CustomHandler(SimpleHTTPRequestHandler):

    def do\_GET(self):

        if self.path == "/":

            self.path = "/index.html"

        return super().do\_GET()

port = 8888

server\_address = ("", port)

httpd = HTTPServer(server\_address, CustomHandler)

print(f"Server port: {port}")

# Continuously open pages

while True:

    # Ask the user for input

    page = input("Enter the page you want to open (or type 'quit' to exit): ")

    if page.lower() == 'quit':

        break

    # Open the browser to the specified page

    webbrowser.open(f"http://localhost:{port}/{page}")

httpd.server\_close()

**about.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>About Us</title>

</head>

<body>

    <h1>About Us</h1>

    <p>This is the about page of our website.</p>

    <p>The website is for demonstration purpose.</p>

</body>

</html>

**style.css**

body {

    font-family: Arial, sans-serif;

}

.container {

    max-width: 600px;

    margin: 0 auto;

}

input[type="text"] {

    width: 70%;

    padding: 8px;

    margin-bottom: 10px;

}

button {

    padding: 8px 20px;

    background-color: #4CAF50;

    color: white;

    border: none;

    cursor: pointer;

}

button:hover {

    background-color: #45a049;

}

ul {

    list-style-type: none;

}

li {

    padding: 8px;

    border-bottom: 1px solid #ddd;

}

li:last-child {

    border-bottom: none;

}

Outputs





