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()
```

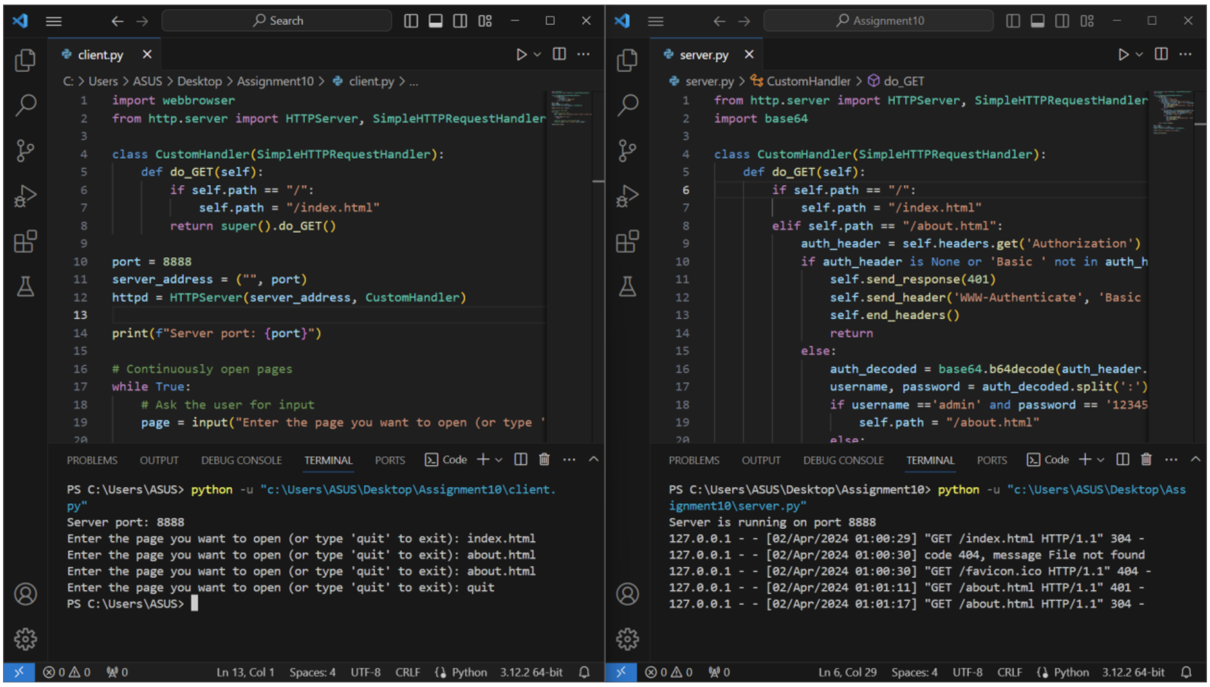
about.html

```
<!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

```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



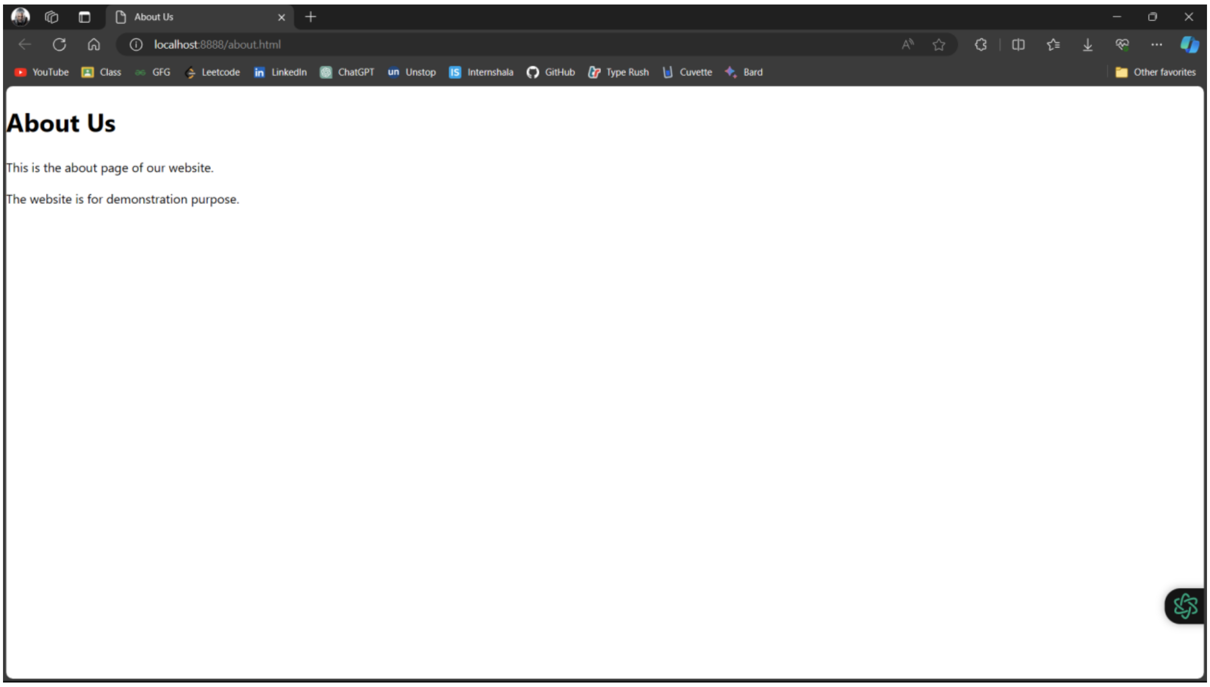
Search
client.py
C: > Users > ASUS > Desktop > Assignment10 > client.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code
Ln 13, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit
Assignment10
server.py
server.py > CustomHandler > do_GET
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code
Ln 6, Col 29 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit

```python
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 '
```

```
PS C:\Users\ASUS> python -u "c:\Users\ASUS\Desktop\Assignment10\client.
py"
Server port: 8888
Enter the page you want to open (or type 'quit' to exit): index.html
Enter the page you want to open (or type 'quit' to exit): about.html
Enter the page you want to open (or type 'quit' to exit): about.html
Enter the page you want to open (or type 'quit' to exit): quit
PS C:\Users\ASUS>
```

```python
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_h
                self.send_response(401)
                self.send_header('WWW-Authenticate', 'Basic
                self.end_headers()
                return
            else:
                auth_decoded = base64.b64decode(auth_header.
                username, password = auth_decoded.split(':')
                if username =='admin' and password == '12345
                    self.path = "/about.html"
                else:
```

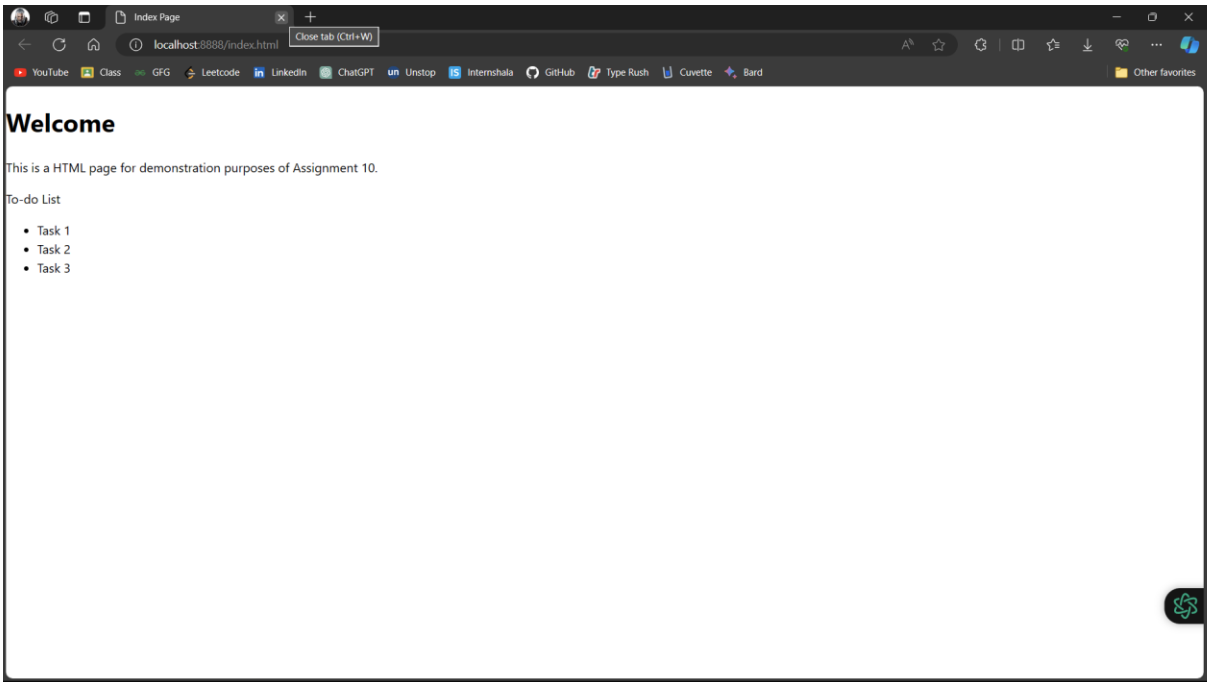
```
PS C:\Users\ASUS\Desktop\Assignment10> python -u "c:\Users\ASUS\Desktop\Ass
ignment10\server.py"
Server is running on port 8888
127.0.0.1 - - [02/Apr/2024 01:00:29] "GET /index.html HTTP/1.1" 304 -
127.0.0.1 - - [02/Apr/2024 01:00:30] code 404, message File not found
127.0.0.1 - - [02/Apr/2024 01:00:30] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [02/Apr/2024 01:01:11] "GET /about.html HTTP/1.1" 401 -
127.0.0.1 - - [02/Apr/2024 01:01:17] "GET /about.html HTTP/1.1" 304 -
```



About Us

This is the about page of our website.

The website is for demonstration purpose.



Welcome

This is a HTML page for demonstration purposes of Assignment 10.

To-do List

- Task 1
- Task 2
- Task 3