EC3022D

COMPUTER NETWORKS

Socket Programming

Salvin Saji
B200930EC
EC-04

Aim :

Create a web server using socket programming. Use a web browser to access an HTML file on the server.

Theory :

Socket programming is a method for communicating between two computers across a network. It makes use of sockets, which are endpoints used to send and receive data between two computers.

There are two sorts of sockets in socket programming: client sockets and server sockets. TCP and UDP sockets are the two types of sockets used in socket programming. TCP (Transmission Control Protocol) is a dependable protocol that ensures data is transferred correctly and without error.
It connects the client and server via a connection-oriented communication channel. UDP (User Datagram Protocol), on the other hand, is an untrustworthy protocol that does not ensure that data will be transferred without error or in the correct order.
It connects the client and server via a connectionless communication channel.

Server-side and client-side involve different steps in socket programming.

Server-side:

1. Create a socket: The server creates a socket that will listen for incoming connections.
2. Bind the socket: The server binds the socket to a specific port number and IP address.
3. Listen for connections: The server starts listening for incoming connections.
4. Accept connections: When a client sends a connection request, the server accepts it and establishes a connection.

5. Receive data: The server receives data from the client over the established connection.
6. Process data: The server processes the received data and sends a response if necessary.
7. Close connection: The server closes the connection after all data has been exchanged.

Client-side:

1. Create a socket: The client creates a socket that will be used to connect to the server.
2. Connect to the server: The client connects to the server using the server's IP address and port number.
3. Send data: The client sends data to the server over the established connection.
4. Receive data: The client receives data from the server.
5. Process data: The client processes the received data and sends another request if necessary.
6. Close connection: The client closes the connection after all data has been exchanged.

Code :

Python code :

```python
import socket
import threading

IP = socket.gethostbyname(socket.gethostname())
PORT = 5050
FORMAT = 'utf-8'

server = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
server.bind((IP,PORT))

def handle_client(conn,addr):
    print(f'{addr} connected \n')
    req = conn.recv(1024).decode(FORMAT)
    reqPath = req.split('\n')[0].split()[1]
    if reqPath == '/':
        reqPath = '/index.html'
    print(reqPath)

    try:
        with open('resource' + reqPath) as f:
            cont = f.read()
            res = f'HTTP/1.1 200 OK\n\n{cont}'
    except FileNotFoundError:
        try:
            with open('resource/404.html') as f:
                cont = f.read()
                res = f'HTTP/1.1 404 NOT FOUND\n\n{cont}'
        except FileNotFoundError:
            cont = 'File not found'
            res = f'HTTP/1.1 404 NOT FOUND\n\n{cont}'

    conn.send(res.encode(FORMAT))
    conn.close()

def start():
```

```
    server.listen()
    print(f"Server running at {IP}:{PORT}")

    while True:
        conn,addr = server.accept()
        thread = threading.Thread(target=handle_client,args=(conn,addr))
        thread.start()
        print(f'Active Clients = {threading.active_count()-1}')

start()
```

index.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome to my website!</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            background-color: pink;
            font-family: Arial, sans-serif;
        }

        .container {
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }

        h1 {
            font-size: 3rem;
            margin-bottom: 1rem;
        }
```

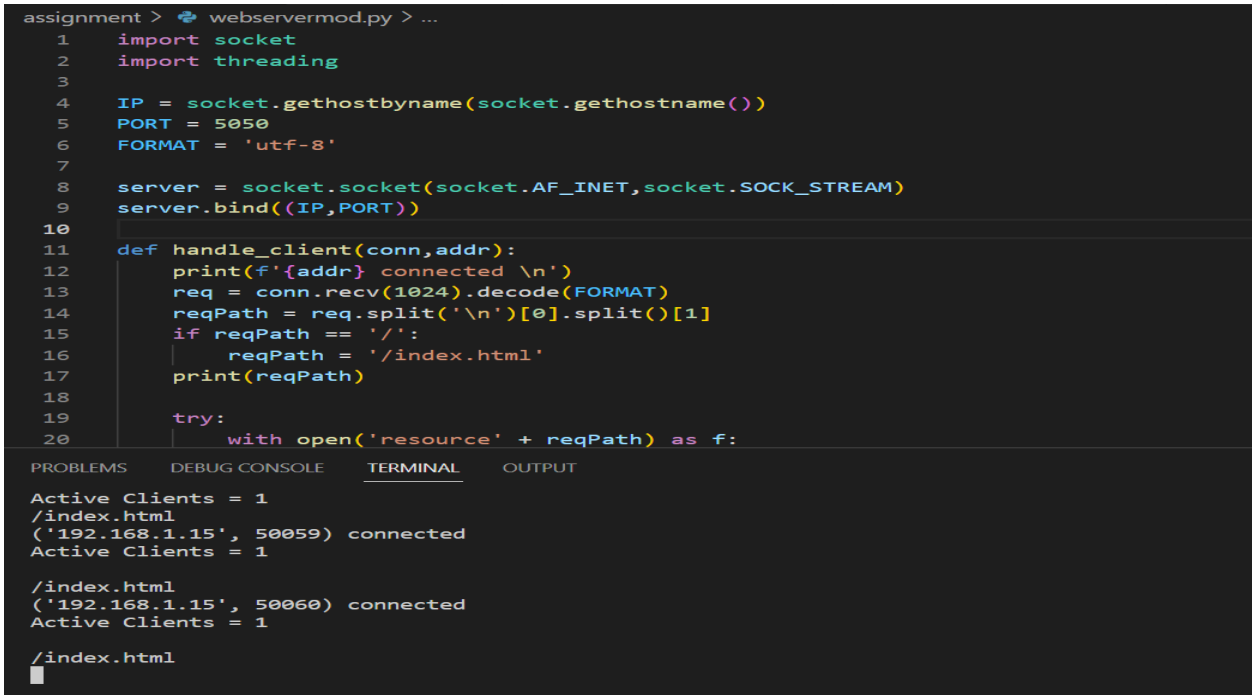
```
        p {
            font-size: 1.5rem;
            margin-bottom: 2rem;
            text-align: center;
            max-width: 80%;
        }

        button {
            background-color: #4CAF50;
            border: none;
            color: white;
            padding: 1rem;
            font-size: 1.2rem;
            cursor: pointer;
            border-radius: 0.5rem;
        }

        button:hover {
            background-color: #3e8e41;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome to my website!</h1>
        <p>Hi I'm Salvin. Thanks for being here. Feel free to leave any
suggetions</p>
        <button>Get Started</button>
    </div>
</body>
</html>
```

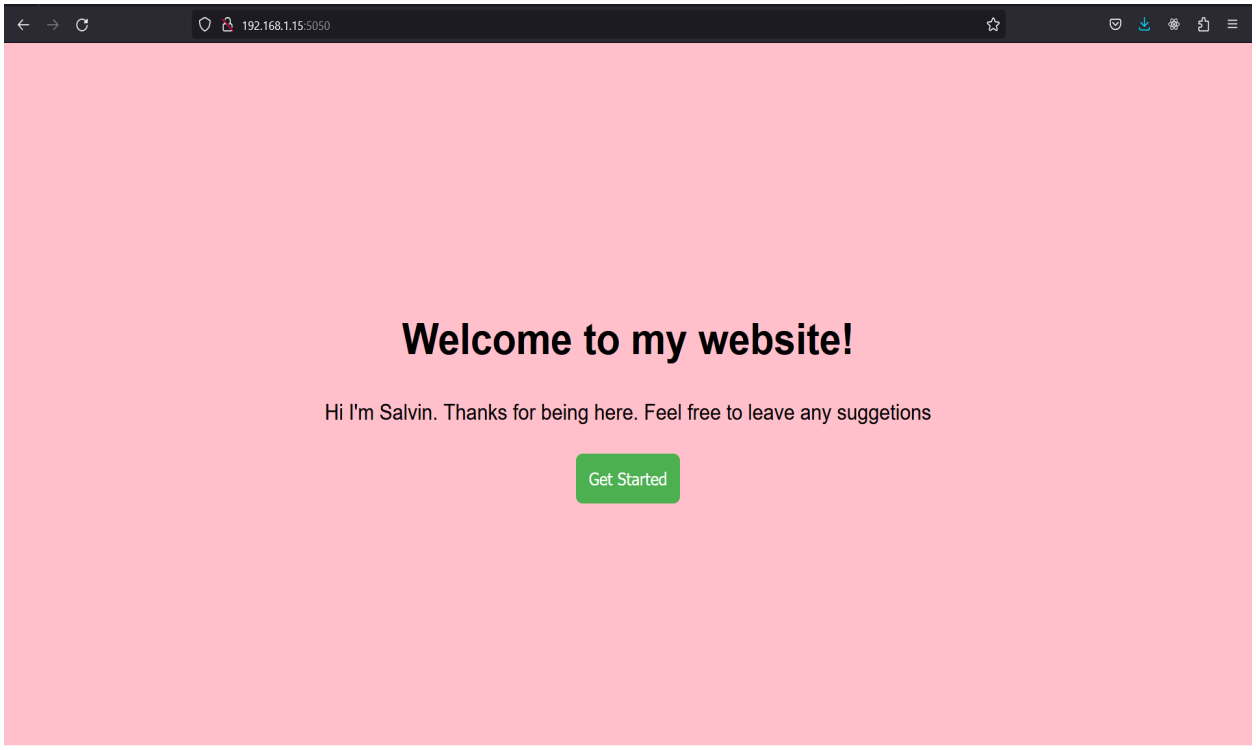
404.html :

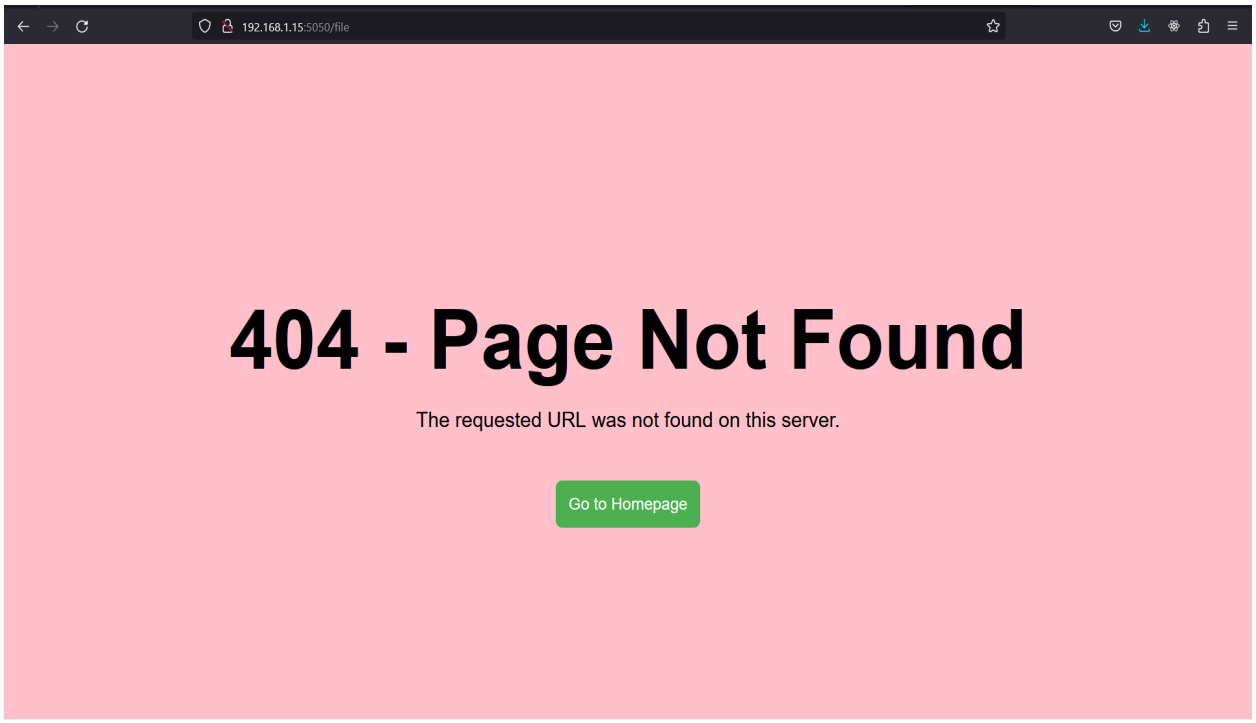
```
<!DOCTYPE html>
<html>
<head>
    <title>404 - Page Not Found</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            background-color: pink;
            font-family: Arial, sans-serif;
        }

        .container {
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }

        h1 {
            font-size: 6rem;
            margin-bottom: 0;
        }

        p {
            font-size: 1.5rem;
            margin-top: 1.5rem;
        }

        a{
            text-decoration: none;
        }

        .btn {
            background-color: #4CAF50;
            border: none;
            color: white;
```

```
            padding: 1rem;
            font-size: 1.2rem;
            cursor: pointer;
            border-radius: 0.5rem;
            margin-top: 2rem;
        }

        .btn:hover{
            background-color: #3e8e41;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>404 - Page Not Found</h1>
        <p>The requested URL was not found on this server.</p>
        <a href="/" class="btn">Go to Homepage</a>
    </div>
</body>
</html>
```

Output :




192.168.1.15:5050
Welcome to my website!
Hi I'm Salvin. Thanks for being here. Feel free to leave any suggetions
Get Started


192.168.1.15:5050/file
404 - Page Not Found
The requested URL was not found on this server.
Go to Homepage

Conclusion :

A socket-based web server is a reliable and efficient way to serve web content that can handle a large number of simultaneous connections and is flexible and scalable. However, it requires careful configuration and management to ensure optimal performance and security. The socket module is used to build and maintain the server. The threading module is used to handle multiple requests simultaneously.