WebSocket Application

---

1. Introduction

This document describes a WebSocket-based application consisting of a Python WebSocket server, a Python client, and an HTML-based client. The purpose of the project is to demonstrate WebSocket communication both programmatically (using Python) and manually (using a browser interface).

---

2. Components Overview

2.1 WebSocket Server (`websocket_server.py`)

The WebSocket server listens for incoming connections on `localhost:8765` and handles messages from both Python and HTML clients. For every message received, it appends a random number and sends the modified message back to the client.

- **Responsibilities**:
  - Receive messages from connected clients.
  - Modify each message by appending a random integer.
  - Send the modified message back to the client.

2.2 Python WebSocket Client (`websocket_client.py`)

The Python WebSocket client is a script that connects to the WebSocket server and sends 10,000 messages programmatically. It receives and prints the server's responses, which include the original message and the appended random number.

- **Responsibilities**:
  - Connect to the WebSocket server on `localhost:8765`.
  - Send 10,000 messages in a loop.
  - Receive and print the server's responses.

2.3 HTML WebSocket Client (`index.html`)

The HTML WebSocket client provides a browser interface where users can manually send messages to the server. Messages typed into the input field are sent to the server, which responds with a modified message that is displayed in the browser.

- **Responsibilities**:
  - Provide a graphical interface for manual message input.
  - Display the server's response in real-time in the browser.

---

3. Application Flow

1. **Server**:
    - The server is started first and listens for WebSocket connections on `localhost:8765`.
2. **Python Client**:
    - The Python client connects to the server and sends 10,000 messages.
    - Each message is processed by the server, modified, and returned to the client, which prints the responses.
3. **HTML Client**:
    - The HTML file is opened in a browser, establishing a WebSocket connection.
    - Users can manually send messages from the browser and view the server's responses in real-time.

---

4. Running the Application

Step 1: Start the WebSocket Server

- Run `websocket_server.py` to initiate the server, which will listen for client connections.

Step 2: Run the Python Client (Optional)

- Execute `websocket_client.py` to send 10,000 programmatic messages to the server.

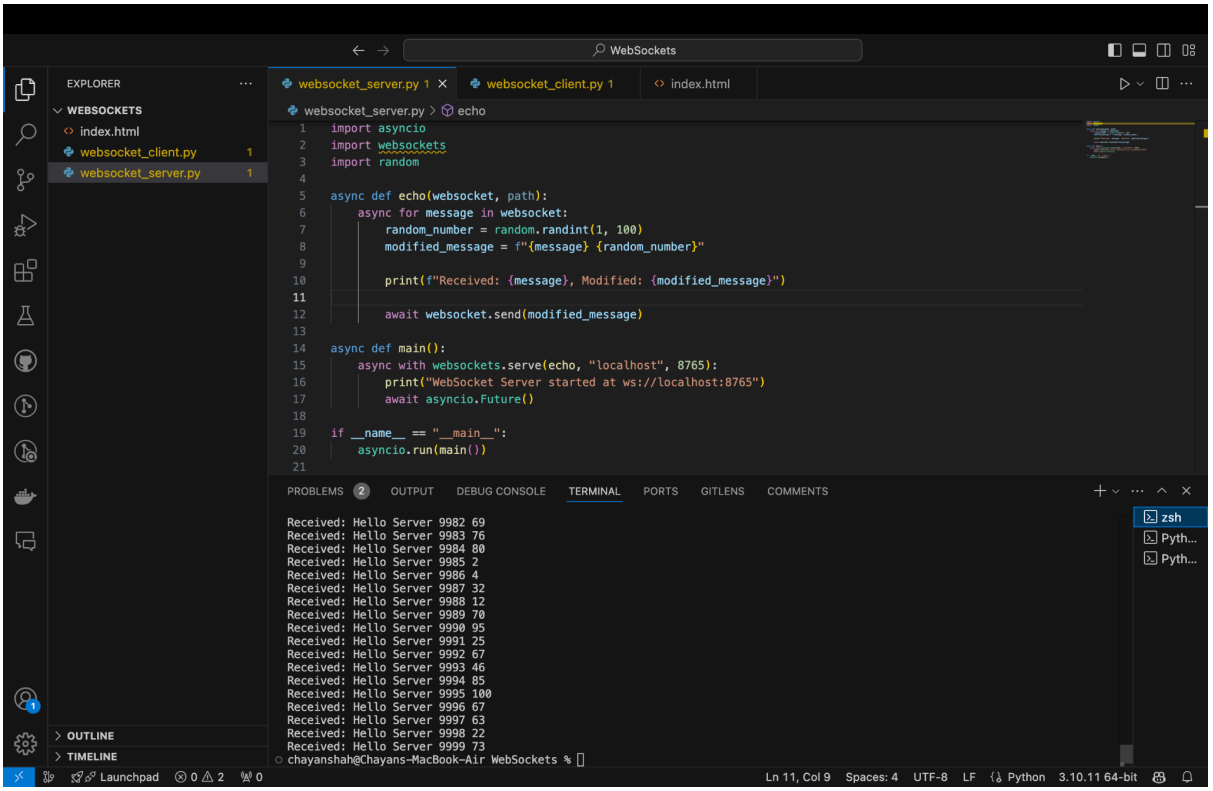
Step 3: Open the HTML Client

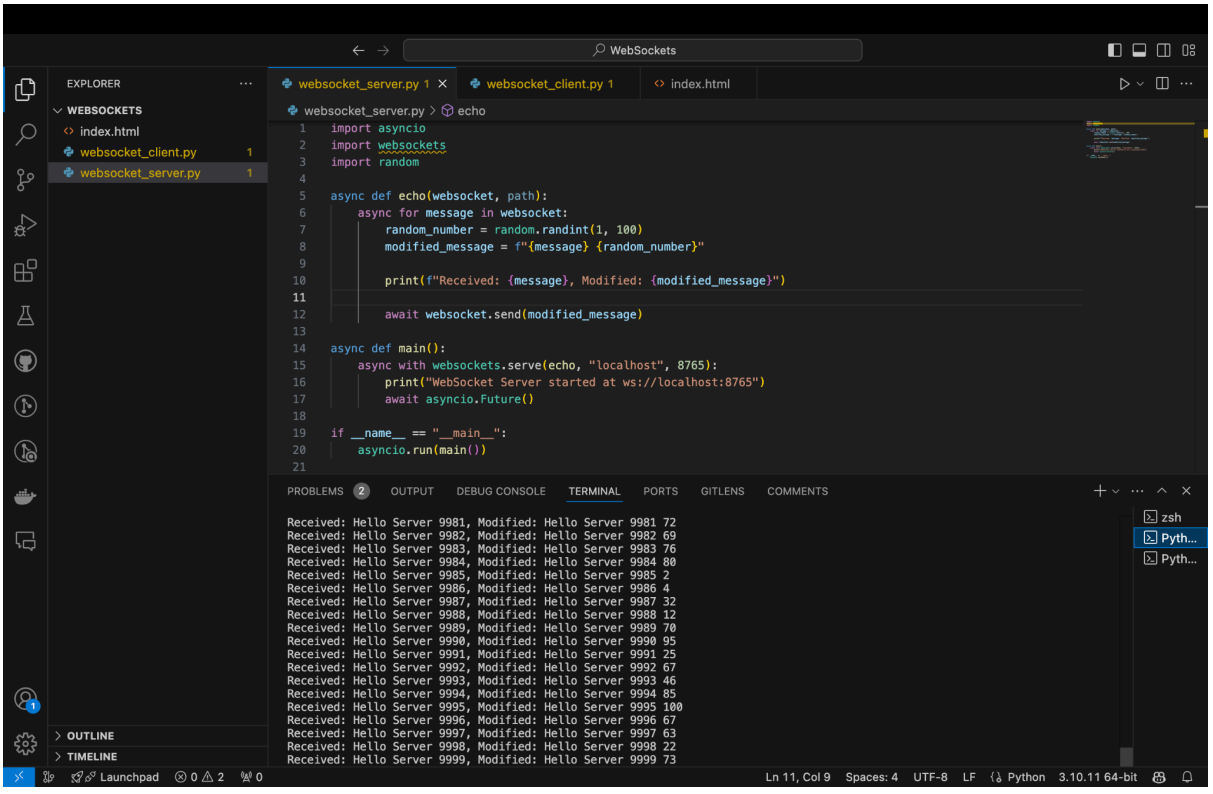
- Serve the `index.html` file using a local web server and open it in a browser to manually send messages to the WebSocket server.

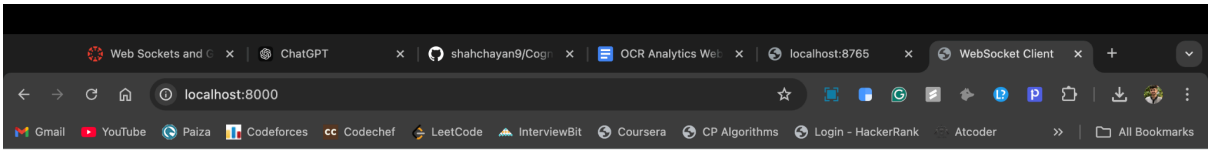
---

5. Conclusion

This WebSocket application demonstrates real-time communication using both a Python-based client for automated testing and an HTML-based client for manual interaction. The server efficiently handles multiple clients, modifying and responding to incoming messages. The application provides a hands-on experience with WebSocket technology, showcasing its flexibility for both automated and interactive use cases.


WebSockets
EXPLORER
WEBSOCKETS
index.html
websocket_client.py 1
websocket_server.py 1
websocket_server.py 1
websocket_client.py 1
index.html
websocket_server.py > echo
import asyncio
import websockets
import random
async def echo(websocket, path):
async for message in websocket:
random_number = random.randint(1, 100)
modified_message = f"{message} {random_number}"
print(f"Received: {message}, Modified: {modified_message}")
await websocket.send(modified_message)
async def main():
async with websockets.serve(echo, "localhost", 8765):
print("WebSocket Server started at ws://localhost:8765")
await asyncio.Future()
if __name__ == "__main__":
asyncio.run(main())
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
COMMENTS
Received: Hello Server 9982 69
Received: Hello Server 9983 76
Received: Hello Server 9984 80
Received: Hello Server 9985 2
Received: Hello Server 9986 4
Received: Hello Server 9987 32
Received: Hello Server 9988 12
Received: Hello Server 9989 70
Received: Hello Server 9990 95
Received: Hello Server 9991 25
Received: Hello Server 9992 67
Received: Hello Server 9993 46
Received: Hello Server 9994 85
Received: Hello Server 9995 100
Received: Hello Server 9996 67
Received: Hello Server 9997 63
Received: Hello Server 9998 22
Received: Hello Server 9999 73
chayanshah@Chayans-MacBook-Air WebSockets %
zsh
Pyth...
Pyth...
OUTLINE
TIMELINE
Launchpad
Ln 11, Col 9
Spaces: 4
UTF-8
LF
Python
3.10.11 64-bit

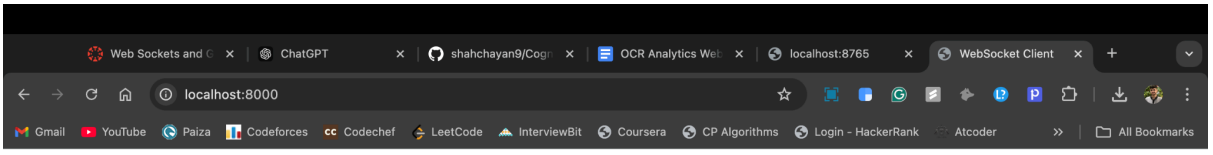

WebSockets
EXPLORER
WEBSOCKETS
index.html
websocket_client.py 1
websocket_server.py 1
websocket_server.py 1
websocket_client.py 1
index.html
websocket_server.py > echo
import asyncio
import websockets
import random
async def echo(websocket, path):
async for message in websocket:
random_number = random.randint(1, 100)
modified_message = f"{message} {random_number}"
print(f"Received: {message}, Modified: {modified_message}")
await websocket.send(modified_message)
async def main():
async with websockets.serve(echo, "localhost", 8765):
print("WebSocket Server started at ws://localhost:8765")
await asyncio.Future()
if __name__ == "__main__":
asyncio.run(main())
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
COMMENTS
Received: Hello Server 9981, Modified: Hello Server 9981 72
Received: Hello Server 9982, Modified: Hello Server 9982 69
Received: Hello Server 9983, Modified: Hello Server 9983 76
Received: Hello Server 9984, Modified: Hello Server 9984 80
Received: Hello Server 9985, Modified: Hello Server 9985 2
Received: Hello Server 9986, Modified: Hello Server 9986 4
Received: Hello Server 9987, Modified: Hello Server 9987 32
Received: Hello Server 9988, Modified: Hello Server 9988 12
Received: Hello Server 9989, Modified: Hello Server 9989 70
Received: Hello Server 9990, Modified: Hello Server 9990 95
Received: Hello Server 9991, Modified: Hello Server 9991 25
Received: Hello Server 9992, Modified: Hello Server 9992 67
Received: Hello Server 9993, Modified: Hello Server 9993 46
Received: Hello Server 9994, Modified: Hello Server 9994 85
Received: Hello Server 9995, Modified: Hello Server 9995 100
Received: Hello Server 9996, Modified: Hello Server 9996 67
Received: Hello Server 9997, Modified: Hello Server 9997 63
Received: Hello Server 9998, Modified: Hello Server 9998 22
Received: Hello Server 9999, Modified: Hello Server 9999 73
zsh
Pyth...
Pyth...
OUTLINE
TIMELINE
Launchpad
Ln 11, Col 9
Spaces: 4
UTF-8
LF
Python
3.10.11 64-bit



WebSocket Client

Enter your message he | Send Message

You: hi
Server: hi 22



WebSocket Client

Enter your message he | Send Message

You: hi
Server: hi 22
You: hello chayan
Server: hello chayan 66
You: Hello3333
Server: Hello3333 68
You: Enjoy Life
Server: Enjoy Life 28