**CODTECH INTERNSHIP TASK-02**

**REAL-TIME CHAT SERVER**

**NAME :** VANTHANA.V

**INTERN ID:** CTO4DY468

**Email** : [poojavanthana5@gmail.com](mailto:poojavanthana5@gmail.com)

**Introduction**

Real-time communication is a crucial feature in many modern applications such as messaging apps, collaborative tools, and gaming platforms. This task focuses on developing a backend server that supports real-time chat functionality using WebSockets or Socket.IO. The server will enable multiple clients to communicate instantly in different chat rooms, enhancing user interaction and engagement.

### Objective

* To understand and implement real-time communication protocols like WebSockets or Socket.IO.
* To build a backend server capable of handling multiple clients and chat rooms simultaneously.
* To deliver a functional chat server using Node.js or Python that supports real-time message exchange.
* To gain practical experience in developing scalable and efficient communication systems for modern applications.

Program :

from fpdf import FPDF

# Create instance

pdf = FPDF()

pdf.add\_page()

pdf.set\_font("Arial", size=12)

# Title

pdf.set\_font("Arial", 'B', 16)

pdf.cell(200, 10, txt="Internship Task - 2", ln=True, align='C')

pdf.cell(200, 10, txt="Real-Time Chat Server", ln=True, align='C')

pdf.ln(10)

# Instructions

pdf.set\_font("Arial", 'B', 12)

pdf.cell(200, 10, txt="Instructions:", ln=True)

pdf.set\_font("Arial", size=11)

pdf.multi\_cell(0, 10, txt="""

• Develop a real-time chat backend using WebSockets or Socket.IO with support for multiple chat rooms.

• Deliverable: A Node.js or Python-based server handling real-time communication.

""")

pdf.ln(5)

# Sample Python Code

pdf.set\_font("Arial", 'B', 12)

pdf.cell(200, 10, txt="Sample Python Code (using WebSocket):", ln=True)

pdf.set\_font("Courier", size=10)

sample\_code = """

import asyncio

import websockets

clients = set()

async def handler(websocket):

clients.add(websocket)

try:

async for message in websocket:

for client in clients:

if client != websocket:

await client.send(message)

finally:

clients.remove(websocket)

start\_server = websockets.serve(handler, "localhost", 6789)

asyncio.get\_event\_loop().run\_until\_complete(start\_server)

asyncio.get\_event\_loop().run\_forever()

"""

pdf.multi\_cell(0, 5, sample\_code)

Sample Output:

pdf.set\_font("Arial", 'B', 12)

pdf.cell(200, 10, txt="Sample Output:", ln=True)

pdf.set\_font("Arial", size=11)

pdf.multi\_cell(0, 10, txt="""

Client 1: Hello

Client 2: Hi there!

Client 3: Welcome to the chat room.