2022-2023

COMP 308 Final Project

Title: A Simple Real-Time Chat Application using Socket Programming

Description: In this project, you will build a simple real-time chat application using socket programming in Python. The chat application will allow multiple clients to connect to a server and communicate with each other in real-time. The application will have a graphical user interface (GUI) for the chat window and message input.

Requirements:

- Use socket programming to establish a connection between the server and clients (i.e. you can use your **programming assignment code** for this)
- Implement the server-side functionality to manage the chat sessions and handle incoming messages from clients.
- Implement the client-side functionality to connect to the server and send and receive messages in real-time.
- Design a GUI for the chat window and message input using a Python GUI library such as Tkinter.
- Ensure that the chat application can handle multiple clients connecting and disconnecting from the server.

Process:

1. **Planning and Design:**
    - Define the chat application requirements and features.
    - Design the user interface for the chat window and message input. User
    - Plan the implementation strategy for the server-side and client-side functionality.
2. **Server-Side Implementation**:
    - Create a Python script to implement the server-side functionality.
    - Use socket programming to establish a connection between the server and clients.

- Implement the logic to manage chat sessions and handle incoming messages from clients.
- Ensure that the server can handle multiple clients connecting and disconnecting.

3. **Client-Side Implementation**:
    - Create a Python script to implement the client-side functionality.
    - Use socket programming to connect to the server and send and receive messages in real-time.
    - Implement the logic to handle user input and display messages in the chat window.
4. **GUI Implementation:**
    - Use a Python GUI library such as tkinter to design the chat window and message input interface. Chat should support at least two buttons for <**Send**> and <**Exit chat**> functions.
    - Integrate the GUI with the server-side and client-side functionality.
    - Ensure that the GUI can handle multiple clients connecting and disconnecting.
5. **Documentation and Submision:**
    - Write a brief report documenting the design and implementation of the chat application.
    - Include screenshots of the GUI and a description of the features implemented.
    - Provide the Python code for the server-side, client-side, and GUI implementation.
    - Submit a demo video (max. 3 min.) which showcase your chat application in action.
    - Send both zipped and unzipped versions of your documents.