Chat Application

Project Title

Chat Application

Objective of the Project

This project is a Java-based GUI Chat Application that demonstrates real-time text-based communication between a client and a server using TCP sockets and Swing GUI. The client also supports chat logging to a file with timestamps.

Files

File Name	Description
ChatServerX.java	Java GUI-based chat server using ServerSocket
ChatClientX.java	Java GUI-based chat client with logging and graceful shutdown

Features

- GUI: Clean, scrollable text area with message input and a "Send" button.
- Multithreaded: Keeps the GUI responsive with background threads.
- Logging: Logs all client messages with timestamps to a text file.
- Graceful Exit: Ends chat when "bye" is typed.
- Console-Free: Fully GUI-driven interface for both client and server.

Technologies Used

-Java SE (JDK 8 or higher)
- Java Sockets (TCP)
- Java I/O Streams
- Java Swing (GUI)
- Multithreading

Project Flow

1. **Compile both files:**
   javac ChatServerX.java
   javac ChatClientX.java

2. **Run Server First (in one terminal or window):**
   java ChatServerX

3. **Run Client (in a new window or terminal):**
   java ChatClientX

4. **Start chatting!**

How It Works

- **Server (ChatServerX)**
  - Starts a server socket on port 12345.
  - Waits for a client to connect using serverSocket.accept().
  - Displays incoming messages from the client in a JTextArea GUI.
  - Sends messages to the client via a GUI JTextField.

- **Client (ChatClientX)**
  - Connects to the server on localhost:12345.
  - Has a GUI to send and receive messages.
  - Saves all messages (incoming and outgoing) to a timestamped log file.
  - If either side sends "bye", the connection closes gracefully.

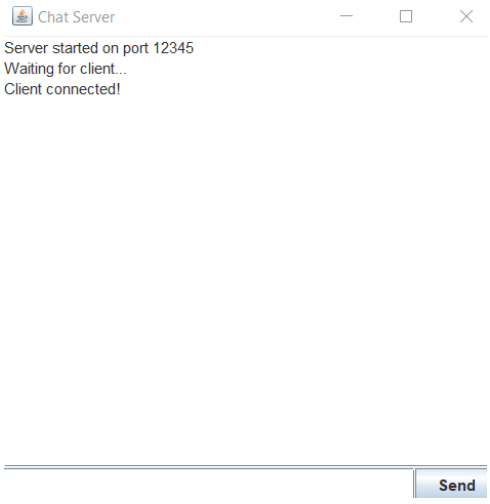
Code Highlights

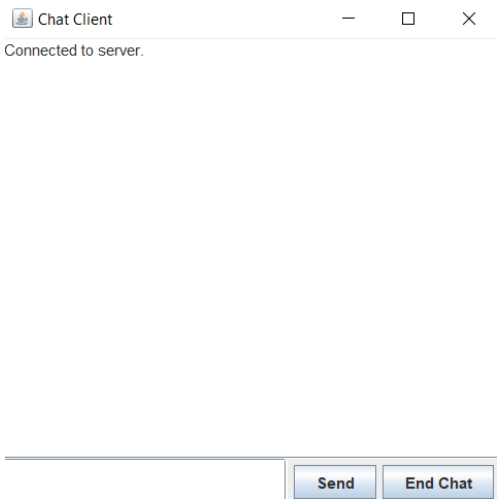
- **ChatClientX.java**
  - startClient() runs in a new thread to keep GUI responsive.
  - Uses SimpleDateFormat and BufferedWriter to log messages.
  - SwingUtilities.invokeLater() ensures thread-safe GUI updates.

- **ChatServerX.java**
  - Uses one thread to accept and handle a single client.
  - GUI for sending/receiving messages using JTextField and JTextArea.

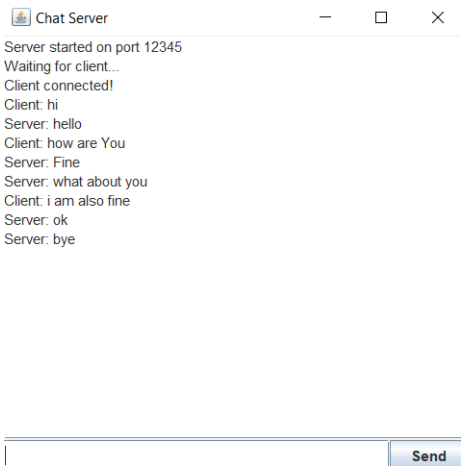
Sample Output

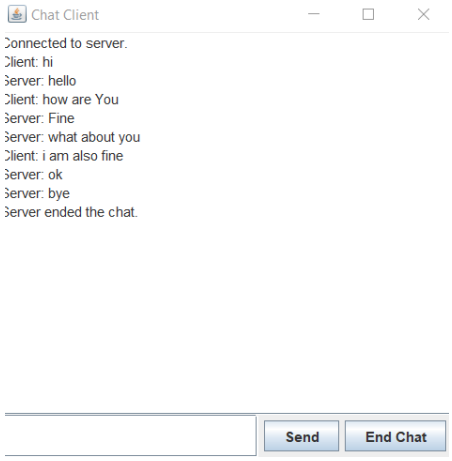
- After compiling and running the two java codes this two window occurs .





- On this two windowws server and client can chat with each other and anyone can send bye messege the chat will be ended gracefully.





- After chat ends one log file will generate with the history of client server chat with timestamp.

Future Enhancements

- Support multiple clients using List<ClientHandler> and multithreading.
- Add user login or nickname functionality.
- Enhance GUI with colors and chat bubbles.
- Add file transfer and image sharing.
- Store chat history in a database.
- Add a status bar or connection monitor.