**Readme markdown code**

**# server-client-system-in-java**

**A Java-based client-server chat app with a Swing GUI supporting real-time messaging, file transfer, and command execution using JSON over sockets. It features multithreaded server handling and an interactive client interface for smooth communication and file exchange.**

**This is a multithreaded Java Client-Server Chat Application using \*\*Swing GUI\*\* on the client side and a \*\*JSON-based protocol\*\* for communication. It supports:**

**- ✅ Text messaging between users**

**- 📁 File requests and downloads**

**- 🛠️ Command execution (like `mkdir`, `ls`, etc.)**

**- 🚫 User management (connect/disconnect)**

**- 💡 Real-time feedback via GUI**

**---**

**## 📦 Features**

**| Feature | Description |**

**|------------------------|----------------------------------------------------------|**

**| 🧑‍💻 GUI Client | Built using Java Swing for easy message/file interaction |**

**| 🌐 Multithreaded Server | Handles multiple clients simultaneously |**

**| 📩 JSON Messaging | Uses `org.json` for structured communication |**

**| 📁 File Transfer | Clients can request and receive files from the server |**

**| ⚙️ Command Execution | Simple commands like `mkdir`, `ls`, etc. are supported |**

**| 🔌 Connect/Disconnect | Real-time status and clean disconnection handling |**

**---**

**## 🧰 Requirements**

**- Java JDK 8 or higher**

**- `json.jar` (included)**

**- A modern IDE like IntelliJ IDEA, Eclipse, or VS Code**

**---**

**## 🔧 Setup Instructions**

**### 💡 Server Setup**

**1. Open your IDE and load the project.**

**2. Compile and run `Server.java`.**

**3. Ensure `shared\_files/` directory exists for file sharing.**

**### 💬 Client Setup**

**1. Open `ClientAppGUI.java`.**

**2. Run the program.**

**3. Enter the server IP and connect.**

**4. Start chatting or download shared files.**

**---**

**## 📎 How It Works**

**- The \*\*Server\*\* listens on a specified port and spawns a `ClientHandler` thread for each connecting client.**

**- Each \*\*Client\*\* sends/receives messages through socket communication.**

**- File sharing is handled by allowing clients to download files from the `shared\_files/` directory.**

**- Messages and commands are exchanged using the \*\*JSON\*\* format for consistency.**

**---**

**In terminal:**

**```bash**

**javac -cp .;json.jar Server.java ClientHandler.java ClientAppGUI.java**

**```**

**---**

**## 🔒 Security Note**

**This is a basic chat app intended for learning purposes. For production use:**

**- Add encryption (SSL/TLS)**

**- Authenticate users**

**- Implement message logging & rate limiting**

**---**

**## 📄 License**

**This project is open-source and free to use under the MIT License.**

**---**

**## 🤝 Contributing**

**Pull requests are welcome! For major changes, please open an issue first to discuss what you would like to change.**

**---**

**## 📧 Contact**

**Made by \*\*haseeb Ullah\*\***

**Email: [haseebullahkhilji0987@gmail.com](mailto: haseebullahkhilji0987@gmail.com)**

**---**