***Multiplayer Chess Game using Socket Programming***

**Project Overview**

This project is a \*\*Multiplayer Chess Game\*\* built using \*\*Socket Programming\*\* principles. It demonstrates real-time two-player communication over a TCP/IP network, simulating a fully functional chess game across client-server architecture.

**Features**

- Real-time multiplayer chess using client-server sockets

- GUI-based chessboard for intuitive gameplay (if applicable)

- Robust connection handling for two players

- Turn-based game logic enforcement

- Clean shutdown and connection closing

**Technologies Used**

- Programming Language: `Python` / `C++` / `Java` \*(based on actual code in your project) \*

- Socket API: TCP sockets

- [Optional] GUI Library: Tkinter / Java Swing / C++ QT (if GUI exists)

**Project Structure**

```bash

Multiplayer\_Chess\_Socket/

├── server.py # Server-side socket handler

├── client.py # Client-side game interface

├── chess\_logic.py # Chess game logic module

├── README.md # Project documentation

└── assets/ # (Optional) images or GUI files

**How to Run**

Extract the .rar file:

bash

Copy

Edit

unzip "Multiplayer Chess Game using Socket Programming.rar"

cd Multiplayer\_Chess\_Socket/

Run the Server:

bash

Copy

Edit

python server.py

Run the Client on another terminal or system:

bash

Copy

Edit

python client.py

Play Chess

**Networking Concept**

This project uses socket communication based on:

TCP Protocol: Reliable, ordered, and error-checked delivery of a stream.

Multi-threading: To handle concurrent client connections.

**Learning Outcomes**

Hands-on experience with socket programming

Understanding of client-server communication in networking

Real-time data transmission and synchronization

Chess logic integration with communication models

**Authors**

Fida Hussain, Muhammad Abubakar, Arshad Ali

**License**

This project is for educational purposes only. Free to use with attribution.