

**Group Details:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Roll Number** | **Section** |
| **Muhammad Hasnain** | **22F-3718** | **BS(SE)-6A** |
| Asfand Ahmed | 22F-3727 | BS(SE)-6A |
| Marsad Saqib | 22F-3733 | BS(SE)-6A |

**Project Details:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name** | **Submitted to** | **Submitted By** | **Submission Date** |
| Multiplayer Chess Game | Sir Saqib Ameer | Squadron Leaders | 05/11/2k25 |

**Team Logo:**



# 1. Introduction

This project is a real-time multiplayer chess game built using Python and socket programming. It allows two players to play chess over a network, with a graphical interface powered by Pygame and Tkinter. The project demonstrates concepts of computer networking, concurrency, and GUI programming.

# 2. Features

* Real-time chess gameplay between two players over a network.
* Graphical chessboard and pieces using Pygame.
* In-game chat functionality.
* Move validation and game state management.
* User-friendly connection and help dialogs.
* Support for chess rules: castling, en passant, pawn promotion, check, and checkmate.
* Chess clock: Each player has a timer; lose on time is supported.

# 3. Project Structure

CN\_LAB Project\_Multiplayer Chess Game/

├── client/

│ ├── client\_main.py

│ └── assets/

│ └── (chess piece images)

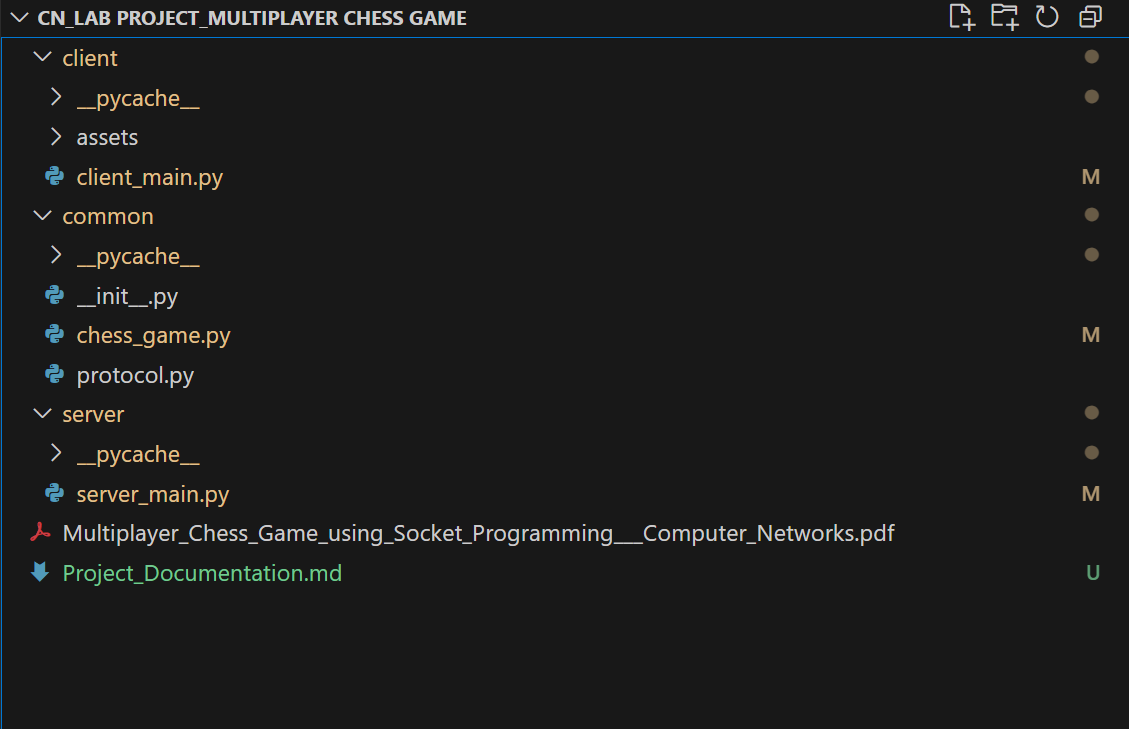
├── server/

│ └── server\_main.py

├── common/

│ ├── chess\_game.py

│ └── protocol.py



# 4. How It Works

* The server listens for incoming connections and manages the game state.
* Each client connects to the server, sends their name, and receives their assigned color (white/black).
* Players interact with a graphical chessboard, make moves, and chat.
* The server validates moves, updates the game state, and broadcasts updates to all clients.
* The game ends when checkmate, stalemate, draw, or time-out is detected.

# 5. Installation & Setup

**Prerequisites**

* Python 3.10+
* Required Python packages: `pygame`, `python-chess`, `tkinter` (standard with Python)

**Install Dependencies**

Open a terminal and run:

**pip install pygame python-chess**

# 6. Usage

1. Start the Server

python -m server.server\_main

**The server will listen on `0.0.0.0:5555` by default.**

2. Start the Client(s)

python -m client.client\_main

* Enter the server IP, port, and your name in the GUI dialog.
* Wait for another player to join.
* Play chess and chat in real-time.

# 7. Protocol

All communication between client and server uses JSON-encoded messages. Each message has a `type` and a `content` dictionary.

Example:

```json

{

"type": "move",

"content": {

"move": "e2e4"

}

}

```

Message Types:

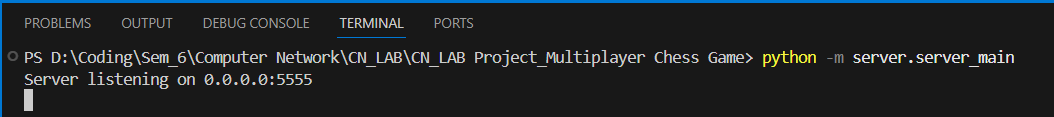
* `join`: Sent by client to join the game.
* `color`: Sent by server to assign color.
* `move`: Sent by client to make a move.
* `board`: Sent by server to update board state and timers.
* `chat`: Chat messages.
* `error`: Error messages.

# 8. Time Control Logic

* Each player starts with a fixed amount of time (e.g., 5 minutes).
* The timer for the player whose turn it is counts down; it pauses when their move is sent and the other player's timer starts.
* If a player's time runs out, the other player is declared the winner and a win message is shown.
* Both timers are displayed side by side at the bottom of the screen in the white area.

# 9. Screenshots

**Server Runs**

****

**Client Joins**

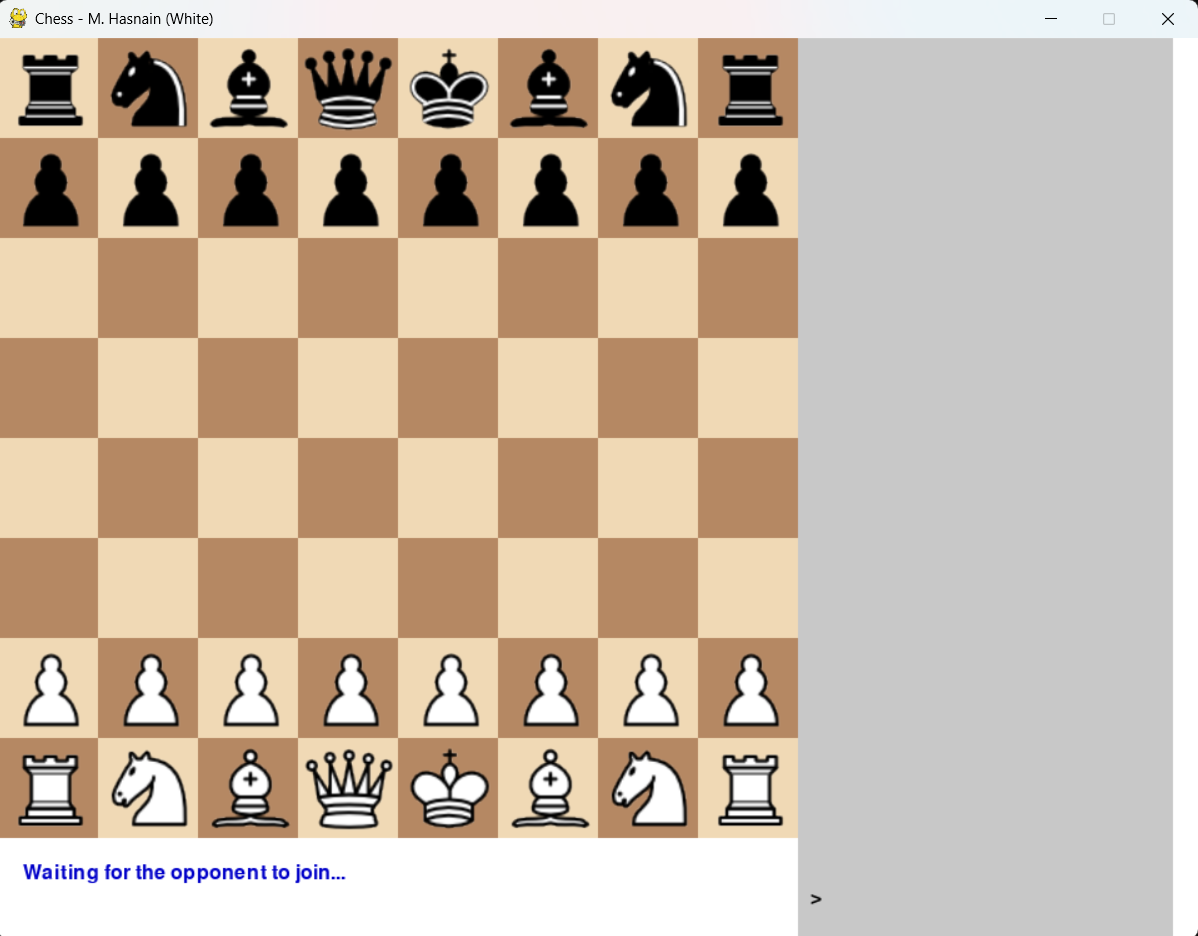
**A black screen with white text

AI-generated content may be incorrect.**

**A screenshot of a computer

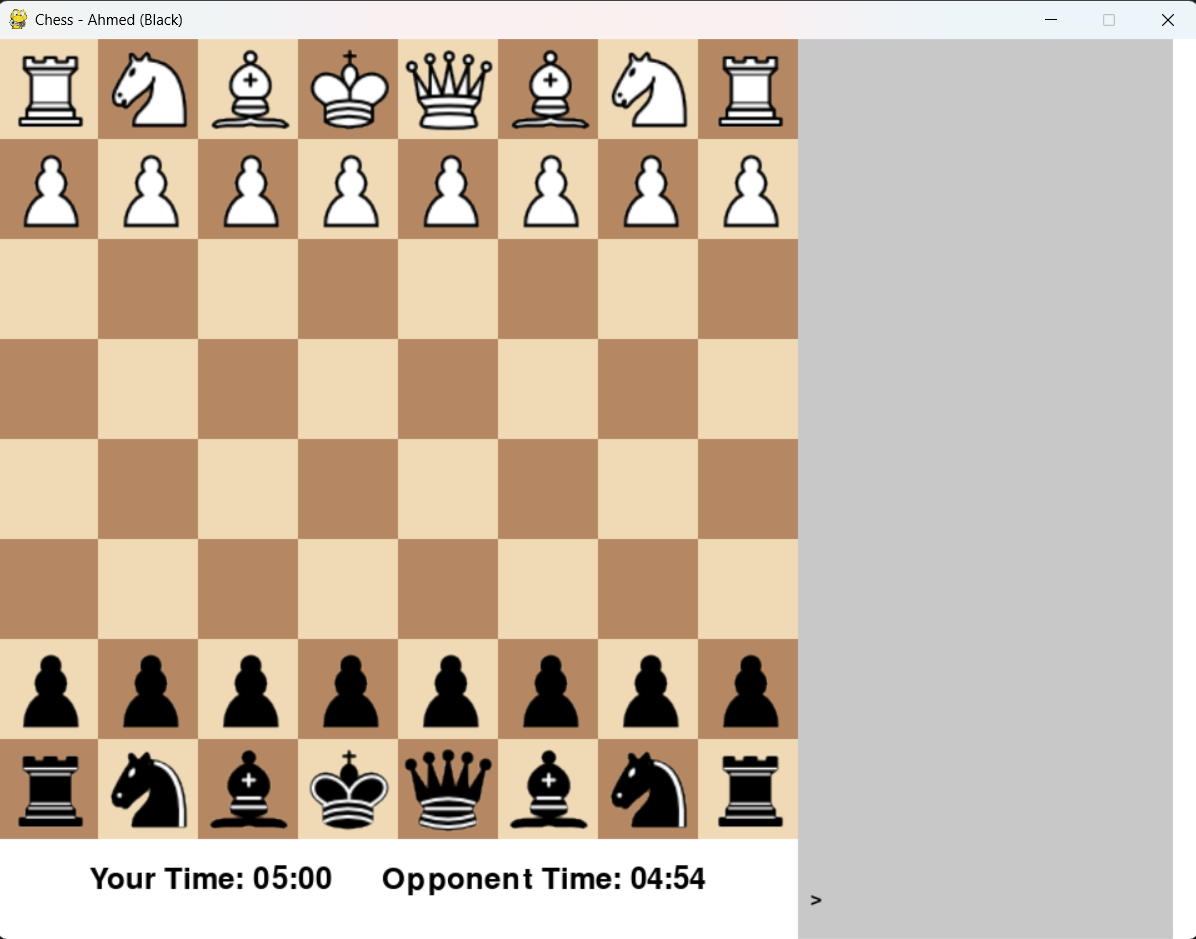
AI-generated content may be incorrect.**

* **As the First player joins the game he will be shown the side of the “white moves”**
* **And He waits for the second player to join the game.**

****

**Second Player Joins.**

* **As soon as the second player joins the game the timer of the first player starts and will be shown to both of the players.**

****

**Both Side View (Two Boards).**

****

**Shows the Movable Places where a user can move.**

**A screenshot of a game

AI-generated content may be incorrect.**

**Spectator Mode:**

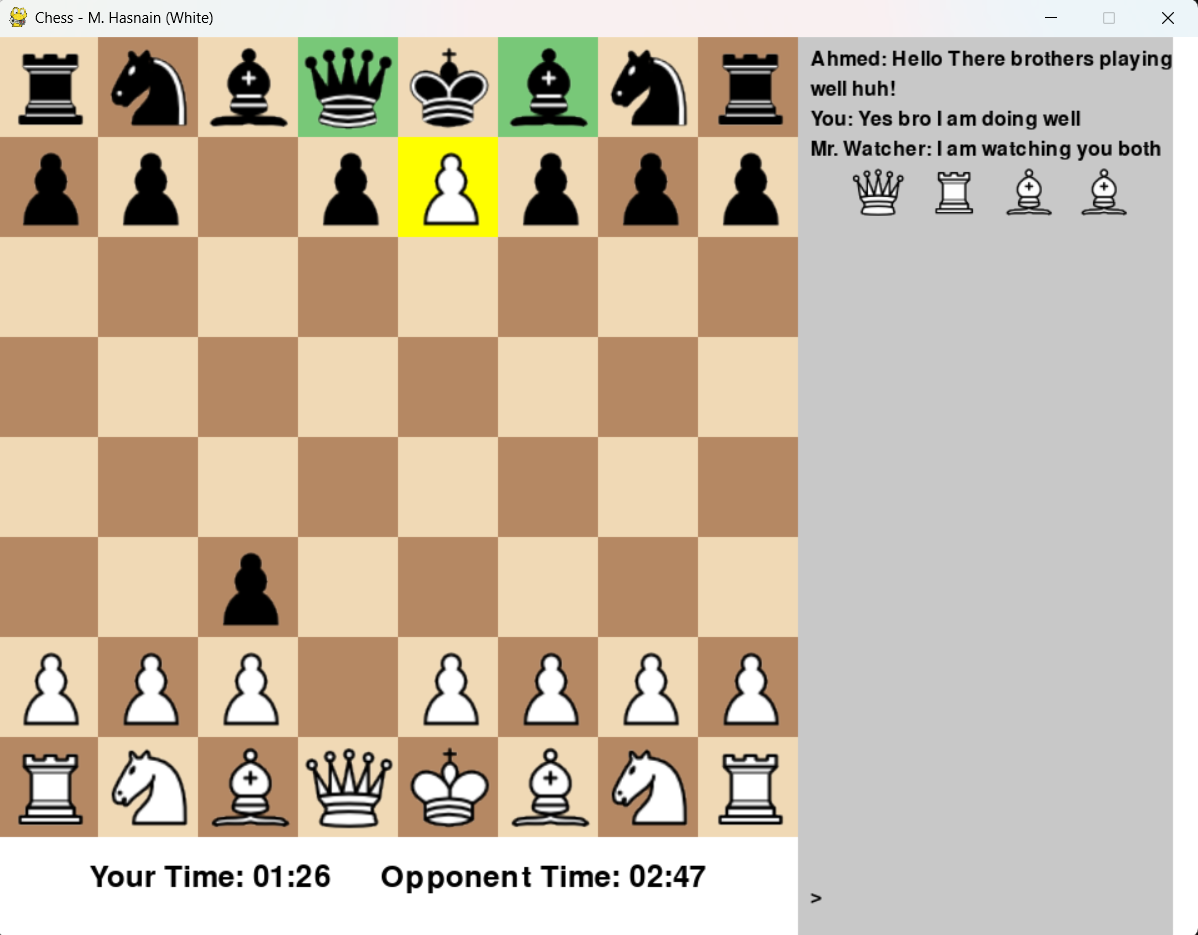
**A screenshot of a computer game

AI-generated content may be incorrect.**

**Shared Chat Message between all the players and Spectators.**

****

**Showing user to make his pawn special to Other Special Piece of Chess.**

****

**On Check the King turns to Red.**

**A screenshot of a game

AI-generated content may be incorrect.**

**Time ends, player winning state and reset game functionalities.**

****

**“Help” Button to make user learn the chess rules and about the pieces of chess.**

**Screens screenshot of a computer

AI-generated content may be incorrect.**