Software Requirements Specification (SRS)

Project: Advanced Tic Tac Toe Game

1. Purpose

The purpose of this document is to define the functional and non-functional requirements for the Advanced Tic Tac Toe Game. This game provides users with a modern GUI, user authentication (registration and login), AI or human opponent options, and personal game history tracking.

2. Scope

Users: Anyone who wants to play Tic Tac Toe, either against another human or an AI.
Deployment: Desktop application built with C++ and Qt.
Persistence: User credentials and game history are stored in a local SQLite database.

3. Functional Requirements

3.1 Authentication

- User Registration: New users can register with a username and password.
- User Login: Existing users must log in to access the game.
- Session Handling: Users remain logged in until logout or application close.
- Error Handling: Appropriate error messages for wrong login credentials or registration errors.

3.2 Gameplay

- Start New Game: After login, users can start a new game.
- Opponent Choice: Select playing vs. AI or another human.
- Move Handling: Players click cells to place X or O. Invalid moves are not allowed.
- Game End Conditions: Detect wins, draws, and display result via popup.

3.3 AI Behavior

- AI is Always O.
- AI Move Strategy: Uses Minimax algorithm with Alpha-Beta Pruning.

3.4 Game History

- Save Results: Result (win/loss/draw) is saved if logged in.
- View History: Users can view a table of past game results.

3.5 GUI

- Modern Dark Themed Interface with Login/Register screens, Game Board, Reset Game button, Toggle Game Mode button, and Menu options.
- Status Updates: E.g., "X's turn", "O wins!" shown during gameplay.

4. Non-Functional Requirements

4.1 Performance

Quick response and lightweight application.

4.2 Reliability

Database persistence and crash resistance.

4.3 Usability

Simple and intuitive UI design with descriptive error messages.

4.4 Maintainability

Modular codebase with unit testing for AI, game logic, and user authentication.

4.5 Portability

Portable across major desktop platforms (Windows, macOS, Linux).

5. System Behavior

Event	System Behavior
User submits login form	Validates credentials and loads game screen
User submits registration form	Registers user if username is unique.
Player clicks cell	Marks X or O depending on turn; checks game status
Game ends	Displays result and stores it if logged in.
Reset button clicked	Resets the board for a new game.
Toggle mode clicked	Switches between Human vs Human and Human vs AI modes.
View history clicked	Opens a window showing previous game results.

6. Game Rules

Players: X always plays first.
Moves: Each player alternately places their mark (X or O) in an empty cell.
Winning Conditions: 3 marks in a row horizontally, vertically, or diagonally.
Draw: No empty cells and no winner.

7. Database Requirements

Tables:

- Users:
  - username (TEXT, primary key)
  - password (TEXT)
- Game History:
  - id (INTEGER, auto-increment)
  - username (TEXT, foreign key)
  - datetime (TEXT)
  - result (TEXT)
  - opponent (TEXT)