# Internship Tasks

Task 02

**SUBMITTED BY:**

**Komal Akbar**

**SUBMITTED TO:**

**CodSoft-Software Company**

**Task:02**

****Project Documentation: Tic-Tac-Toe AI with Minimax Algorithm****

**Overview:**

This project introduces a Tic-Tac-Toe game implemented in Python, featuring an AI opponent powered by the Minimax algorithm with Alpha-Beta Pruning. The game enables users to play against an intelligent AI that employs strategic moves to compete.

**Objectives:**

The project's primary objectives were to:

- Develop a functional Tic-Tac-Toe game with a graphical representation of the board.

- Implement an AI opponent capable of making optimal moves using the Minimax algorithm.

- Provide an engaging user experience through interactive gameplay.

- Offer players the chance to compete against an intelligent AI opponent.

**Key Features:**

1. ****Graphical Game Board:****

   The game showcases the Tic-Tac-Toe board as a graphical layout, with numbered positions for user input.

2. ****Player Interaction:****

   Players can make their moves by entering a number corresponding to the desired position on the board.

3. ****AI Opponent:****

   The AI opponent is driven by the Minimax algorithm with Alpha-Beta Pruning, making strategic moves to compete effectively.

4. ****Win and Draw Detection:****

   The game includes mechanisms to detect winning moves and draws, ensuring an accurate outcome.

5. ****Dynamic Responses:****

Players receive real-time feedback on their moves and the game's outcome, enhancing engagement.

**Benefits:**

- Provides an engaging platform for playing Tic-Tac-Toe against an intelligent AI opponent.

- Demonstrates proficiency in Python programming, game design, and algorithm implementation.

- Showcases the application of Minimax algorithm for optimizing decision-making in games.

**User Experience:**

Players interact with the game by selecting positions on the board, while the AI opponent employs the Minimax algorithm to make strategic decisions. Real-time feedback informs players of their progress and the game's outcome.

**Usage:**

1. Players take turns to make moves by entering a number between 0 and 8.

2. The game detects winning combinations and announces the winner or a draw.

3. The AI opponent leverages the Minimax algorithm to make strategic moves, providing a challenging gameplay experience.

**Acknowledgments:**

This project was made possible by the integration of the Minimax algorithm and the Python programming language. It showcases the potential of AI-driven decision-making in interactive games.

**Future Enhancements:**

- Implementation of a graphical user interface (GUI) for enhanced visual appeal.

- Integration of more advanced AI algorithms for increased strategic complexity.

- Inclusion of a two-player mode for human vs. human gameplay.

**Conclusion:**

The Tic-Tac-Toe AI project demonstrates the synergy between Python programming and strategic decision-making algorithms. By offering a platform for playing against an AI opponent, the project underscores the value of combining technology and gaming for engaging user experiences.