Project Proposal

Project Title: 2D Chess with Adaptive AI Opponent

Submitted By: Shanza Noor

Course: AI

Instructor: Alishba Subhani

Submission Date: 11-05-2025

# 1. Project Overview

Project Topic:  
This project involves the development of a traditional 2D chess game enhanced with an AI opponent that uses the Minimax algorithm with Alpha-Beta pruning. The focus is on simulating intelligent gameplay using strategic evaluation functions.

Objective:  
The main goal is to develop a functional and competitive 2D chess game AI capable of making optimal decisions in real-time using Minimax with Alpha-Beta pruning, with custom heuristics to assess board positions and challenge human players.

# 2. Game Description

Original Game Background:  
Chess is a turn-based two-player strategy game played on an 8x8 board with 16 pieces per player. Each type of piece has specific movement rules. The game ends in checkmate, stalemate, or draw.

Innovations Introduced:  
- No rule changes or board modifications have been introduced.  
- The innovation lies in developing a competitive AI using efficient game tree search techniques and custom heuristics such as material value, positional advantage, and king safety.  
- These enhancements increase gameplay depth and simulate challenging AI behavior.

# 3. AI Approach and Methodology

AI Techniques to be Used:  
- Minimax Algorithm  
- Alpha-Beta Pruning  
- Heuristic-based Evaluation

Heuristic Design:  
- Material Value: Numerical scoring of pieces (e.g., queen = 9, pawn = 1).  
- Positional Advantage: Bonus points for controlling the center and active piece placement.  
- King Safety: Penalty for exposed or poorly defended kings.

Complexity Analysis:  
- Time complexity increases exponentially with depth, i.e., O(b^d), where b is the branching factor and d is the depth.  
- Alpha-Beta pruning reduces the number of nodes evaluated, significantly optimizing performance during deeper searches.  
- Main challenges include tuning heuristics and ensuring AI responsiveness within 2–3 seconds per move.

# 4. Game Rules and Mechanics

Modified Rules:  
- No modifications; standard chess rules are maintained.

Winning Conditions:  
- Checkmate, draw (stalemate, threefold repetition, insufficient material), or resignation.

Turn Sequence:  
- Turn-based. White plays first, and players alternate turns.

# 5. Implementation Plan

Programming Language:  
- Python (Backend & AI)  
- JavaScript (Frontend)

Libraries and Tools:  
- NumPy (AI logic)  
- React.js (User Interface)  
- Flask (Backend integration)  
- GitHub (Version Control)

Milestones and Timeline:  
- Week 1-2: Game rule and UI design  
- Week 3-4: Minimax and heuristic implementation  
- Week 5-6: Backend-frontend integration and game testing  
- Week 7: Final AI integration and debugging  
- Week 8: Performance evaluation and report compilation

# 6. References

- Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach  
- Chess Programming Wiki – Heuristics and Evaluation  
- Chess.com – Game Rules and Strategies  
- React.js Documentation  
- NumPy Documentation