-

**Semester Project Proposal**

---

Project Title:

Sudoku Solver: Intelligent Search and Interactive Gameplay

---

Group Members:

Zunaira – 23k0013

Emaan – 23i-2560

Tanisha – 23k0067

Dania – 23k0072

---

Short Description:

This project implements a Sudoku game that features an AI-powered solver using optimized search algorithms. It combines a clean, interactive user interface with algorithmic decision-making and efficient puzzle generation. The goal is to explore and demonstrate search space optimization, constraint satisfaction techniques, and AI-driven gameplay, while also analyzing performance and algorithm efficiency.

---

Project Components:

1. Search Algorithm:

Core algorithm: Backtracking with MRV (Minimum Remaining Values) heuristic.

Comparisons with:

Plain backtracking

Forward Checking (optional)

Emphasis on reducing the number of recursive calls and pruning the search tree.

2. Search Space Expansion:

Support for multiple puzzle difficulties.

Optional: Variants like 4x4 and 16x16 boards to explore larger/smaller search spaces.

Puzzle generator that ensures a unique solution and adjustable difficulty.

3. AI-Driven Functionality:

Solver can solve any valid puzzle input by the user.

AI can assist with hints based on current board state.

Real-time feedback on invalid inputs or rule violations.

4. User Interface:

Developed using Tkinter or Pygame.

Features:

Grid-based input for Sudoku puzzles

Visual solver demonstration (step-by-step solving or instant solution)

Controls for generating, solving, and resetting puzzles

Clean layout focused on usability and interaction clarity.

5. Comparative Analysis:

Evaluate and compare the implemented algorithms based on:

Number of nodes visited

Recursion depth

Time complexity

Performance on various difficulty levels

Include graphs or tables where helpful.

6. Documentation & Final Report:

Methodology: Puzzle structure, constraints, and search approach.

Description of each search algorithm used and its impact.

Evaluation section comparing performance with traditional methods.

Screenshots from the interface and dry run steps.

---

Tools & Technologies:

Language: Python

UI Framework: Tkinter or Pygame

Libraries: numpy for internal logic, time for performance analysis