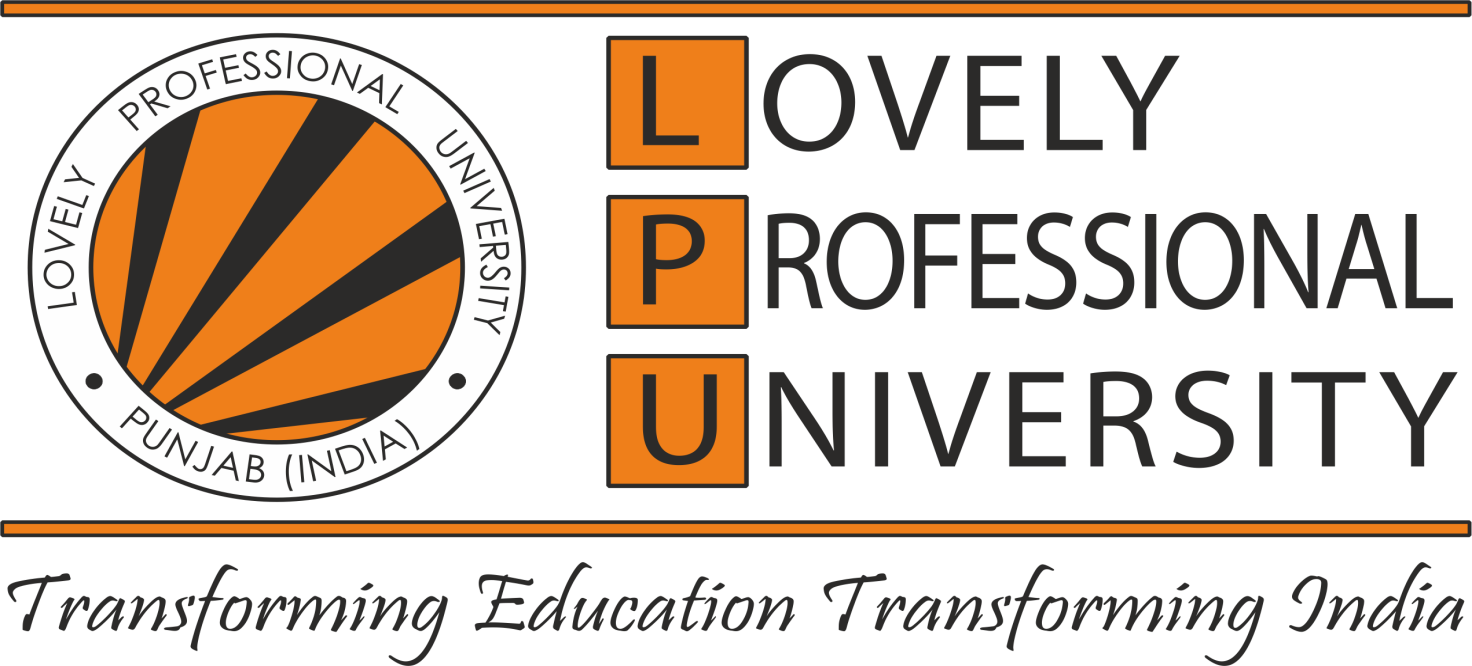
**PROJECT REPORT**



**Project Title - Sudoku Solver**

Name - Abhishek Kumar Maurya

Class – 9SK02

Registration No. – 12223228

Roll No. - 46

Submitted to - Rahul Rajput

Table of contents

1. **Introduction**
   1. Purpose
   2. Key Features
2. **SudokuSolver Class**
   1. isSafe
   2. solveSudokuWithGUI
3. **SudokuGUI Class**
   1. SudokuGUI Class Implementation
   2. SudokuSolver Class Implementation
4. **Conclusion**
5. **GitHub Repository Link**

1.**Introduction**

This report covers the implementation and functionality of a Sudoku solver with a graphical user interface (GUI) written in Java.

The project consists of two main classes: SudokuGUI and SudokuSolver.

**1.1. Purpose**

The purpose of this project is to visualize the process of solving Sudoku using Backtracking algorithm and use java gui library like javax, jpanel, etc.

**1.2. Key Features**

This project key feature is to visualize the process of solving sudoku using backtracking algorithms.

**2. SudokuSolver Class**

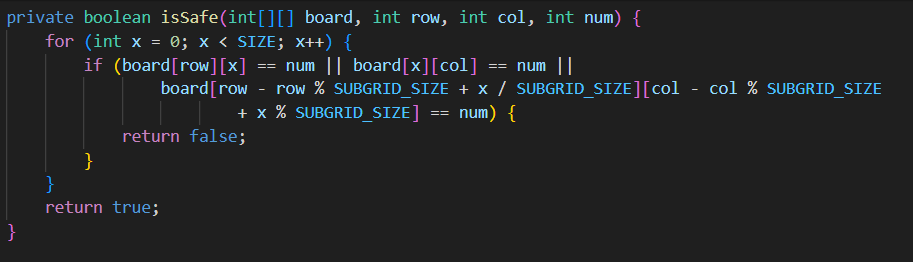
The SudokuSolver class contains the logic for solving the Sudoku puzzle. It uses a backtracking algorithm to find the solution and integrates with the GUI to visualize the solving process.

Key Methods:

* isSafe: Checks if placing a number in a specific cell is valid.
* solveSudokuWithGUI: Solves the puzzle with GUI updates to visualize the solving process.

**2.1 isSafe**

isSafe method checks whether it is safe to place a number from 1 to 9 on that cell by checking entire row, column and 3x3 subgrid.



**2.2 solveSudokuWithGUI**

This method solves the board by recursively checking every number from 1 to 9 on the board and calling isSafe function. This method implements Backtracking Algorithm to solve the board and implement gui part.



**3. SudokuGUI Class**

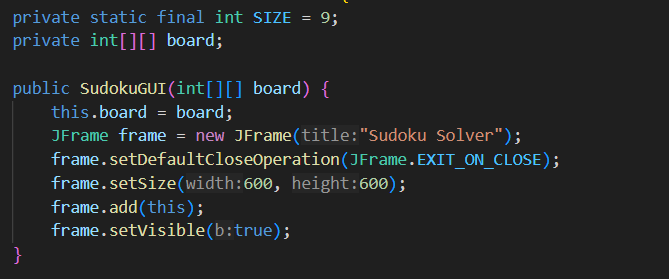
The SudokuGUI class extends JPanel and handles the graphical representation of the Sudoku board. It creates a window using JFrame and renders the Sudoku grid and numbers.

Key Methods:

* Constructor: Initializes the GUI with the provided Sudoku board.
* updateBoard: Updates the board and repaints the GUI.
* paintComponent: Draws the grid, numbers, and highlighted empty cells.

**3.1. SudokuGUI Class Implementation**

* The SudokuGUI class initializes the JFrame and sets the size and default close operation. The updateBoard method allows the board to be updated and the GUI to be repainted to reflect changes.



* A screen shot of a computer program

  Description automatically generatedThe paintComponent method is responsible for drawing the grid. It first fills empty cells with a light gray color and then draws the numbers in the cells. Grid lines are drawn, with thicker lines every three cells to highlight the 3x3 subgrids.

**3.2. SudokuSolver Class Implementation**

* A computer screen shot of text

  Description automatically generatedThe isSafe method checks if a number can be placed in a specific cell by ensuring the number does not already exist in the same row, column, or 3x3 sub grid.
* The solveSudokuWithGUI method implements backtracking to solve board and includes calls to gui.updateBoard and Thread.sleep to visualize the solving process in the GUI.

A screen shot of a computer program

Description automatically generated

**4. Conclusion**

This Sudoku solver with a GUI demonstrates the integration of a backtracking algorithm with a graphical interface, allowing users to visualize the solving process. The combination of SudokuGUI for rendering the board and SudokuSolver for solving the puzzle creates an interactive and educational tool for understanding how Sudoku puzzles can be solved programmatically.

**A screenshot of a sudoku game

Description automatically generated**

**5. GitHub Repository Link**

[Abhishek-Maurya2/sudokuSolver (github.com)](https://github.com/Abhishek-Maurya2/sudokuSolver)