

Project Report

**Title: Sudoku Solver and
Visualizer**

Computer science and Engineering

Submitted to

LOVELY PROFESSIONAL

UNIVERSITY PHAGWARA, PUNJAB



L OVELY
P ROFESSIONAL
U NIVERSITY

Introduction:

This project is a web-based Sudoku generator and solver. It uses Java with Spring Boot to create the backend, handling Sudoku puzzle generation and solving using a backtracking algorithm. The frontend is built with HTML, CSS, and JavaScript, featuring an interactive Sudoku grid. Users can select puzzle difficulty and solving speed, and the solution process is visualized in real-time. The grid cells are color-coded to indicate current solving steps, solved cells, and conflicts. The project also includes enhanced styling, such as rounded table borders and a background image, to improve user experience.

Backend:

- **Technology Stack: Java, Spring Boot**
- **Functionality:**
 - The backend is responsible for generating Sudoku puzzles and solving them.
 - **Sudoku Generation:**
 - Uses a random selection process to fill a 9x9 grid partially.
 - Implements a backtracking algorithm to ensure the grid forms a valid Sudoku puzzle.
 - **Sudoku Solving:**
 - Utilizes a backtracking algorithm to find solutions.
 - The algorithm checks for conflicts and ensures that each number placed in the grid adheres to Sudoku rules.

Frontend:

- **Technology Stack:** HTML, CSS, JavaScript
- **Interface:**
 - **Controls:**
 - Dropdowns for selecting puzzle difficulty (Easy, Medium, Hard).
 - Dropdown for selecting solving speed (Slow, Fast, Very Fast).
 - Buttons to generate a new Sudoku puzzle and to start the solving process.
 - **Visualization:**
 - The grid is dynamically generated based on the puzzle provided by the backend.
 - Real-time updates show the solving process, with color-coding for different cell states:
 - **Current Cell:** Highlighted in red to indicate the cell currently being processed.

- **Solved Cell:** Highlighted in green once a correct number is placed.
- **Conflict Cell:** Highlighted in red if a conflicting value is encountered.

- **Styling:**

- The interface features a cantered Sudoku grid with rounded borders for a polished look.
- A background image enhances the visual appeal.
- Controls and buttons are styled for ease of use and accessibility.

Real-Time Updates:

- **Mechanism:**

- JavaScript fetches solving steps from the backend.
- The solving process is visualized step-by-step at the selected speed.
- The grid updates in real-time, providing immediate feedback to the user.

- **User Experience:**

- The visual feedback helps users understand the solving algorithm.
- Color-coded updates enhance clarity and engagement.

Conclusion:

- . This project effectively demonstrates how to integrate backend logic with frontend visualization.
- . It provides an engaging and educational tool for Sudoku enthusiasts, showcasing the power of algorithms and real-time web technologies.
- . Future enhancements could include user input for custom puzzles, additional difficulty levels, and further UI/UX improvements.