VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018

A Mini Project Report on

"Sudoku Game Solver"

Submitted in Partial Fulfillment of the Requirements for the IV Semester of the Degree of

Bachelor of Engineering in

Computer Science & Engineering

By

Aishwarya Sujan (1CR23CS021)

Under the Guidance of

Prof. Rajini Tiwari, Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with "A++" Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA,

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with “A++” Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CELEBRATING 30 YEARS

CMRIT

CMR INSTITUTE OF TECHNOLOGY, BENGALURU.

ACCREDITED WITH A++ GRADE BY NAAC

CERTIFICATE

This is to certify that the Database Management System Project work entitled **“Sudoku Game Solver”** has been carried out by **Aishwarya Sujan, 1CR23CS021** bonafide student of CMR Institute of Technology, Bengaluru in partial fulfillment for the award of the Degree of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2024-2025**. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the departmental library. This mini-project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Signature of Guide

Prof. Rajini Tiwari

Assistant Professor

Dept. of CSE, CMRIT

Signature of HOD

Dr. Kesavamoorthy R.

Professor & HoD

Dept. of CSE, CMRIT

DECLARATION

I, the student of IV semester of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the project work entitled "**Sudoku Game Solver**" has been successfully completed under the guidance of **Prof. Rajini Tiwari**, Assistant Professor, Dept. of Computer Science and Engineering, CMR Institute of Technology, Bengaluru. This project work is submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2024-2025. The matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date: 14th May, 2025

Aishwarya Sujan (1CR23CS021)

ABSTRACT

The Sudoku Solver project is a web-based application developed using HTML, CSS, and JavaScript, aimed at providing users with an interactive platform to solve 9×9 Sudoku puzzles efficiently. The application interface is designed to replicate the appearance of a traditional Sudoku grid, with distinct thick borders separating each 3×3 sub-grid and an alternating green background colour pattern to improve visual clarity and user experience. Users can manually enter numbers into the grid using input fields, where only values between 1 and 9 are allowed. Real-time input validation ensures that any invalid entry (such as numbers greater than 9 or empty values) is immediately cleared and an alert is shown to notify the user, maintaining the integrity of the puzzle before processing begins. The core of the application is a backtracking algorithm implemented in JavaScript, which recursively searches for valid numbers that can be placed in each empty cell without violating Sudoku rules. Once the "Solve" button is clicked, the current board is captured, validated, and passed to the solving algorithm. Upon successful completion, the solution is displayed directly within the existing input grid. The project demonstrates how front-end web development can be combined with classic algorithmic approaches to deliver a powerful and visually appealing utility. This project is not only useful for solving puzzles quickly but also serves as a practical demonstration of problem-solving, logic building, and web development integration. It is especially beneficial for users who enjoy Sudoku or are interested in understanding how algorithms can solve real-world problems.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out the Database Management System Project.

It gives me an immense pleasure to express my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bengaluru, for his constant encouragement.

I would like to extend my sincere gratitude to **Dr. Kesavamoorthy R.,** HOD, Department of Computer Science and Engineering, CMRIT, Bengaluru, who has been a constant support and encouragement throughout the course of this project.

I would like to thank my guide **Prof. Rajini Tiwari,** Asst. Professor**,** Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of the project work.

I would also like to thank all the faculty members of Department of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I thank my parents and friends for all the moral support they have given me during the completion of this work.

TABLE OF CONTENTS

Contents

Certificate

Declaration

Abstract

Acknowledgement

Table of contents

1.Introduction

2. Software Requirements

3. Implementation

4. Interpretation of Result

5. Conclusion and Future Scope

6. References

INTRODUCTION

Sudoku is a logic-based, combinatorial number-placement puzzle that has gained widespread popularity across the globe. Originating from a concept known as "Number Place," Sudoku challenges a player to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 subgrids (also called "boxes") contains all of the digits from 1 to 9. While it appears simple at first glance, solving a Sudoku puzzle can range from straightforward to extremely complex, making it a favorite among logic enthusiasts and puzzle solvers.

The motivation behind this project stems from the need to create an accessible, easy-to-use, and visually clear tool to solve Sudoku puzzles automatically. The traditional method of solving Sudoku involves manually identifying possible values for each cell through logical deduction. This process, while intellectually stimulating, can be time-consuming and frustrating—especially for more complex puzzles. This project seeks to automate the solving process by applying algorithmic logic, providing both a learning platform and a utility tool for users.

This Sudoku Solver is a web-based application built using core front-end technologies: **HTML**, **CSS**, and **JavaScript**. The interface is crafted to closely mimic the layout of a printed Sudoku puzzle, with thick borders separating each 3×3 box and alternating green background patterns for visual ease. Users can input their own puzzles using a grid of number input fields, ensuring flexibility in the types of puzzles that can be solved. Input validation is enforced in real-time—only numbers from 1 to 9 are allowed, and any invalid entry prompts an alert message and clears the incorrect value. This feature ensures that the input remains clean and suitable for the solving algorithm to process.

At the heart of the application lies a **backtracking algorithm**—a standard method used for solving constraint satisfaction problems like Sudoku. The algorithm works recursively, trying out numbers in empty cells and backtracking when a contradiction is found. This approach guarantees that if a valid solution exists, the algorithm will find it. Once the "Solve" button is clicked, the current grid is read, validated, and passed into the algorithm. The solution is then automatically filled into the grid, offering an immediate and accurate result.

SOFTWARE REQUIREMENTS

The following software components were used in the development and execution of the Sudoku Solver web application:

1. Frontend Technologies

- **HTML5**: Used for creating the structural layout of the Sudoku grid and input interface.
- **CSS3**: Applied for visual styling, including grid design, color patterns, and responsive layout.
- **JavaScript (ES6+)**: Handles the core logic, including dynamic grid creation, input validation, and the solving algorithm based on backtracking.

2. Web Browser

- **Microsoft Edge**
  The project is designed to run on modern web browsers, and development and testing were primarily conducted using Microsoft Edge, which supports all required web standards.

3. Code Editor

- **Notepad++**
  Used for writing and editing HTML, CSS, and JavaScript code due to its lightweight nature and support for syntax highlighting.

4. Operating System

- **Windows**
  The application was developed and executed on the Windows operating system, providing a stable and compatible environment for browser-based development.

IMPLEMENTATION

The implementation of the Sudoku Solver project involves a combination of HTML, CSS, and JavaScript. The following key steps describe the development process:

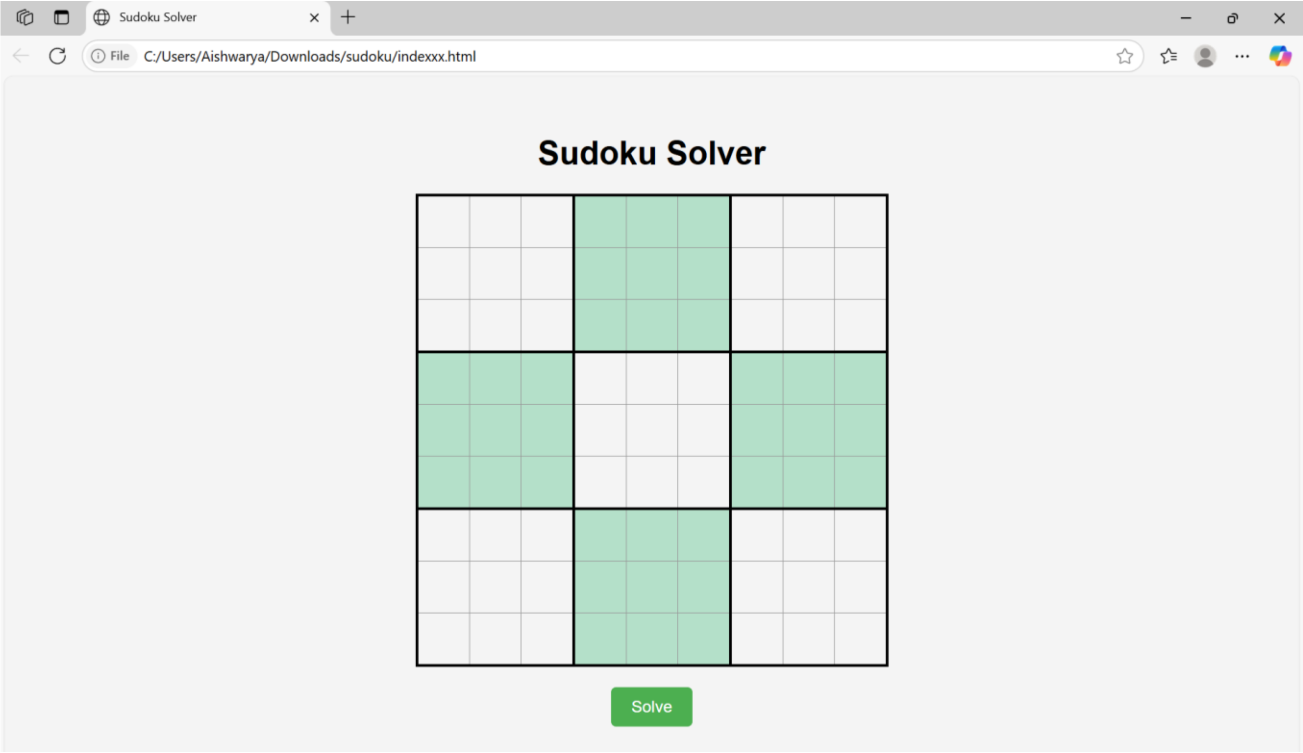
- Grid Creation
  - A 9×9 table is dynamically created using JavaScript.
  - Each cell contains an input field that accepts numbers from 1 to 9.
  - Every 3×3 box is styled with thick borders to visually separate Sudoku regions, enhancing user experience.
- Colour Highlighting
  - Alternate 3×3 blocks are given a green background using CSS to mimic a traditional Sudoku appearance and improve usability.
- Input Validation
  - JavaScript ensures that only numbers between 1 and 9 are accepted.
  - If a user enters a number outside this range, an alert is triggered, and the input is cleared automatically to prevent solving with invalid values.
- Backtracking Algorithm
  - A recursive backtracking function is implemented in JavaScript to solve the puzzle.
  - It scans for empty cells, checks for valid placements of numbers (row-wise, column-wise, and within the 3×3 box), and fills them recursively.
  - If no valid move is found, it backtracks and tries a different number.
- Solve Button Functionality
  - A “Solve” button is added to trigger the solving process.
  - On clicking, the current board values are extracted into a 2D array, passed to the solver, and then updated on the UI with the solution.
- Responsive Design
  - CSS is used to make the layout responsive and ensure that cells remain square-shaped across screen sizes.

SOURCE CODE:

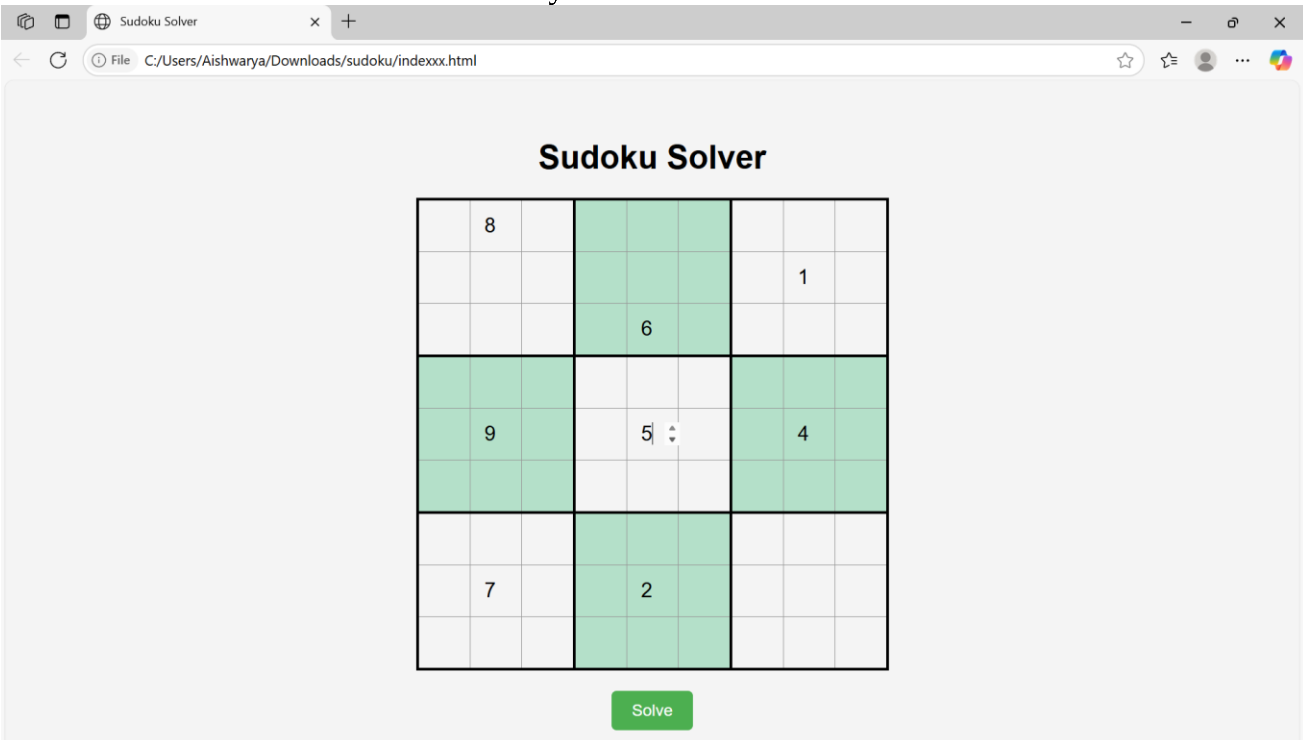
https://drive.google.com/drive/folders/1ofvzqkvtXwa_8Akms8esf4i3GzZujCc7?usp=drive_link

INTERPRETATION OF RESULTS (OUTPUT SCREENS)

Pre-entry

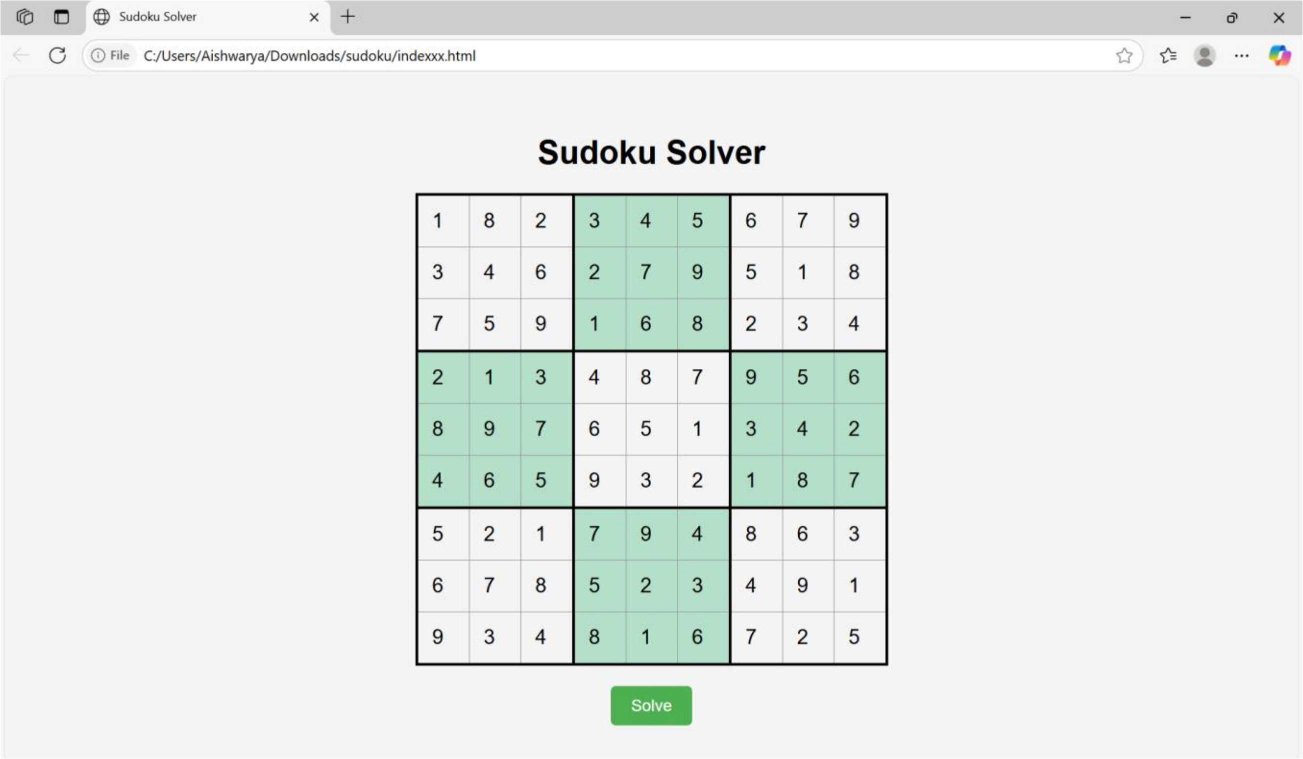


Entry of numbers



	8							
							1	
				6				
	9			5			4	
	7			2				

Solved Board



Sudoku Solver

1	8	2	3	4	5	6	7	9
3	4	6	2	7	9	5	1	8
7	5	9	1	6	8	2	3	4
2	1	3	4	8	7	9	5	6
8	9	7	6	5	1	3	4	2
4	6	5	9	3	2	1	8	7
5	2	1	7	9	4	8	6	3
6	7	8	5	2	3	4	9	1
9	3	4	8	1	6	7	2	5

CONCLUSION AND FUTURE SCOPE

Conclusion

- The Sudoku Solver project effectively demonstrates how a combination of front-end technologies such as HTML, CSS, and JavaScript can be used to build an interactive and functional web-based application.

- It offers users a simple interface to input Sudoku puzzles and instantly receive accurate solutions using a recursive backtracking algorithm.

- The grid-based structure mimics traditional Sudoku formats, with clear visual cues like colored 3×3 blocks and validation for correct inputs.

Future Scope

- **Mobile Compatibility:** Enhancing the design to be more responsive for mobile devices will expand usability across platforms.

- **Puzzle Generator:** Adding a feature that randomly generates new puzzles of varying difficulty levels can make the tool more engaging for users.

- **Step-by-Step Solving Visualization:** Implementing a feature that shows the solving process step-by-step can be educational for users learning Sudoku strategies.

- **Timer Integration:** Including a timer to track how long a user takes to solve a puzzle can turn the application into a competitive or self-improvement tool.

- **Difficulty Level Selection:** Allowing users to choose from easy, medium, or hard puzzles will make the solver more customizable and appealing to different skill levels.

- **Saving & Sharing Puzzles:** Features for saving current progress or sharing puzzles with friends can make the platform more interactive and social.

REFERENCES

- **Exact Method for Generating Strategy-Solvable Sudoku Clues**
  https://arxiv.org/abs/2005.14098
- **Memristive Oscillatory Circuits for Resolution of NP-Complete Logic Puzzles: Sudoku Case**
  https://arxiv.org/abs/2002.06339
- I**mage Detection and Digit Recognition to Solve Sudoku as a Constraint Satisfaction Problem**
  https://arxiv.org/abs/1905.10701
- **A Simple QUBO Formulation of Sudoku**
  https://arxiv.org/abs/2403.04816
- **Mathematics of Sudoku**
  https://en.wikipedia.org/wiki/Mathematics_of_Sudoku
- **Sudoku Solving Algorithms**
  https://en.wikipedia.org/wiki/Sudoku_solving_algorithms