# Sudoku puzzle based on ASP

Bhargavi Chadalawada, Dhanwanth Batini, Divya Komati, Noufa Haneefa,
Manish Doddamanne Nagaraju.

January 10, 2025

## 1 Introduction

The Sudoku game is designed to engage players of all ages and skill levels through a classic number-placement puzzle that challenges logic and critical thinking. With three distinct difficulty levels such as Easy, Medium, and Hard, the game caters to both beginners and experienced enthusiasts alike. The easy mode serves as an accessible introduction to the game's mechanism, allowing new players to familiarize themselves with the basic rules of Sudoku without feeling overwhelmed, while the medium and hard mode present increasingly complex puzzles that require advanced strategies and analytical skills, enhancing players problem-solving abilities [2].

This application offers an immersive gaming experience but also features an intuitive user interface that allows for seamless interaction. Players can easily navigate through the game, select their desired difficulty level, and fill in their answers on visually appealing grid. To further enhance the learning experience, the game incorporates validation rules that provide immediate feedback on the correctness of the players inputs, helping them learn from their mistakes in real time.

This Sudoku Puzzle is a web-based application that combines Answer Set Programming (ASP) and Clingo to efficiently process Sudoku rules and constraints.It enables users to generate puzzles, solve grids, validate inputs, and request hints, showcasing ASP power in solving complex logic-driven problems. By fusing declarative programming with modern web technologies, this project provides a seamless and interactive Sudoku-solving experience, demonstrating how ASP simplifies intricate logic while delivering an engaging and educational platform.

# 2 Application Description

## 2.1 Application System

The Sudoku Puzzle is built with two primary components: the frontend and the backend. The frontend is built using HTML, CSS, and JavaScript, creating a visually appealing and interactive user interface. It enables users to interact with a dynamic Sudoku grid where they can generate puzzles, input numbers, solve puzzles, and request hints.The backend is powered by Clingo which is a powerful answer set programming system that facilitates complex problem-solving through logic programming[5].

The ASP component delivers a responsive and intuitive interface, enabling players to select difficulty levels, fill in their answers, and receive real-time feedback on their entries, ensuring smooth navigation and user engagement. Clingo efficiently manages puzzles data and game states, allowing players to request hints and apply logical reasoning for puzzle-solving .It processes requests sent from the frontend, handles logic-solving tasks, and generates or solves Sudoku puzzles based on the user's inputs. The communication between the frontend and backend is facilitated by the Fetch API, ensuring smooth data exchange and real-time updates.

The interaction begins when a user selects a difficulty level or inputs a partially completed puzzle. These actions trigger requests from the frontend to the backend, where Clingo processes the logic and constraints to generate and solve the puzzle. The results are then returned to the frontend in JSON format, enabling the grid to update dynamically. The interactions between the front end and the back end are fluid and dynamic , utilizing structured queries that uphold data integrity while updating game progress and statistics.

## 2.2 Technical Details

The application's functionality is accomplished by combining declarative and imperative programming techniques. The backend's primary function is to encode Sudoku rules using ASP, which makes sure that every row, column, and 3x3 subgrid has distinct integers between 1 and 9. Clingo uses these rules as input to create puzzles and solve grids. As the foundation of the reasoning, the rules.lp file establishes the linkages and constraints inside the Sudoku grid.

JavaScript is used in the frontend for real-time validation and dynamic modifications. It checks against backend precomputed solutions to make sure user inputs follow Sudoku rules. Furthermore, responsive design concepts and CSS styles improve the user experience by making the interface aesthetically pleasing and device-compatible.

The Fetch API is used to accomplish the integration, enabling the frontend to communicate user actions and puzzle states to the backend and receive the outcomes. The system will continue to be responsive and engaged owing to this smooth connection.

## 2.3 Application Usage

**Choosing of Difficulty level:**

The below figure depicts the process of choosing the level of difficulty by drop down menu which consists three options such as 'Easy, Medium, and Hard'. After choosing the difficulty, the user needs to click on 'Generate Sudoku' button to start the game.

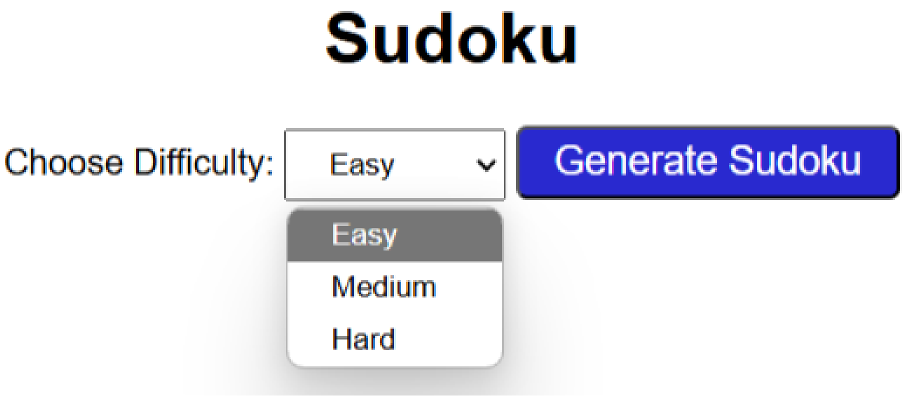
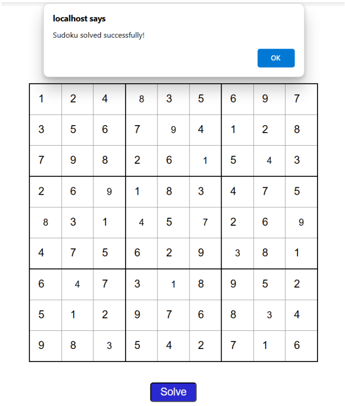


Figure 1: Difficulty Levels.

**Solving a Sudoku Puzzle:**

After manually entering the numbers on the grid, users can ask the system to solve the problem for them. The current grid state is sent to the backend via the Sudoku function, where Clingo analyses it to identify a workable solution. The user sees the finished grid when it has been returned to the frontend.



**localhost says**

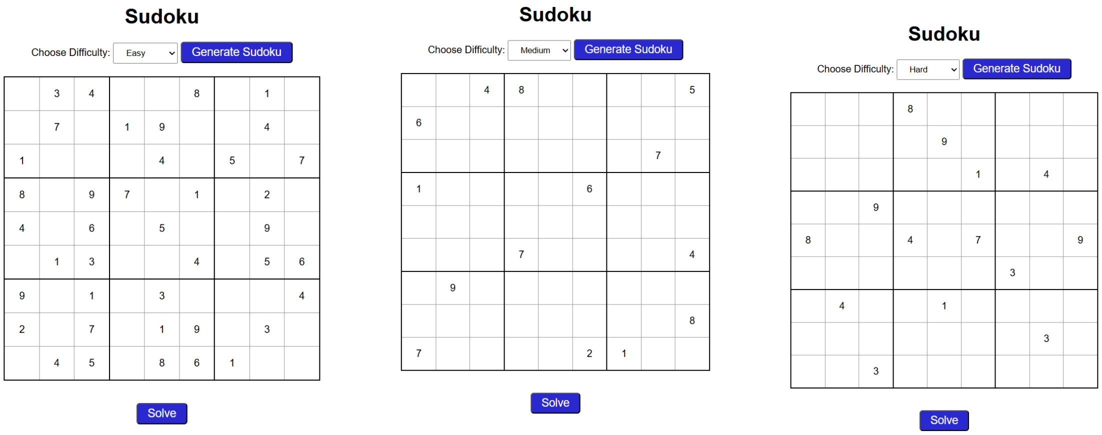
Sudoku solved successfully!

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

Figure 2: Automatic Puzzle Solver.

### Generating a Sudoku Puzzle:

To generate a new Sudoku puzzle, users can choose from three different difficulty levels: Easy, Medium, and Hard. A valid grid is returned by the backend after processing the request according to ASP rules, and the frontend dynamically displays it. The number of pre-filled cells and the complexity of the applied constraints are determined by the level of difficulty.



**Sudoku**

Choose Difficulty: Easy

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

(a) Easy Level

**Sudoku**

Choose Difficulty: Medium

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 4 | 8 | | | | | 5 |
| 6 | | | | | | | | |
| | | | | | | | 7 | |
| 1 | | | | | 6 | | | |
| | | | | | | | | |
| | | | 7 | | | | | 4 |
| | 9 | | | | | | | |
| | | | | | | | | 8 |
| 7 | | | | | 2 | 1 | | |

(b) Medium Level

**Sudoku**

Choose Difficulty: Hard

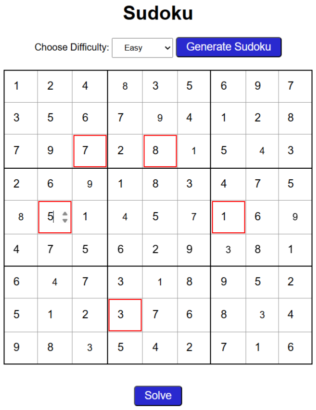
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 8 | | | | | |
| | | | | 9 | | | | |
| | | | | | 1 | | 4 | |
| | | 9 | | | | | | |
| 8 | | | 4 | | 7 | | | 9 |
| | | | | | | 3 | | |
| | 4 | | | 1 | | | | |
| | | | | | | | 3 | |
| | | 3 | | | | | | |

(c) Hard Level

Figure 3: Difficulty Levels of Sudoku Puzzle

**Validating Inputs:**

The system dynamically compares user input to the solution recorded in the backend as users complete the grid. Red borders are used to indicate incorrect entries, giving users immediate notice so they can fix their errors.



Sudoku

Choose Difficulty: Easy

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

Figure 4: Puzzle Validation.

.

## 3 Discussion

The Sudoku game project served as an engaging and educational application designed to enhance players' logical and critical thinking skills through a classic number placement puzzle. It was developed using HTML,CSS,JavaScript for the front-end interface , ASP and clingo for back-end management, aiming to create an enjoyable gaming experience that appealed to users across various age groups and skill levels. Players could choose from three distinct difficulty levels such as Easy, Medium, and Hard, each tailored to challenge different levels of expertise while fostering cognitive development. The interactive nature of the application was strengthened by a user-friendly interface that allowed for seamless navigation and real-time feedback on player inputs. This combination of thorough planning, intuitive design, and dynamic data handling encapsulated the essence of the project.

Throughout the development of this application, frequent lessons were learned that extended beyond just programming. One key insight was the significance of user experience in application design. the necessity for intuitive navigation and responsive feedback became paramount, emphasizing that an engaging interface could significantly impact user retention and satisfaction.the integration of the clingo database underscored the importance of efficient data management, highlighting how well-structured schemas could improve both retreival speed and data integrity.

Despite its strengths, the project also encountered limitations. Even though ASP and Clingo are strong, their dependence presents scalability and portability issues. More complicated limitations or larger grids could result in more computing cost.Furthermore, Clingo prioritises overcoming constraints over adding diversity, so randomisation in puzzle development necessitates additional reasoning. One notable constraint was the reliance on pre-defined difficulty levels, which did not fully accommodate the vast range of player skill and learning curves. A potential enhancement could have involved adaptive difficulty settings, allowing the game to respond dynamically to players' performances in real time. Also, while clingo served as an effective lightweight database, it did not scale well for larger applications with numerous simultaneous users, revealing a not scale well for larger applications with numerous simultaneous users, revealing a limitation in handling extensive player data over time. Moreover, the static nature of Sudoku puzzles generated by algorithms could some times lead to repetitive game play experiences , highlighting the need for diverse puzzle generation techniques to enhance player engagement.

The project also affords an opportunity to explore and apply various theories and technologies learned from the course.The importance of knowledge representation played a significant role in formulating the game logic. Additionally, delving into the principles of the Semantic Web illuminated data could be shared and integrated across different applications, enhancing interoperability and collaboration.

## 4 Conclusion

This is a fun and effective way to solve Sudoku puzzles by fusing the strength of declarative programming with contemporary online technology. Through the use of Clingo and Answer Set Programming (ASP), the project shows how intricate logic-driven issues can be precisely and simply resolved. The system offers features including puzzle generation, automatic solving, real-time input validation, and hint requests, allowing users to engage with Sudoku puzzles in a dynamic manner.

This project demonstrates how adaptable ASP is for practical applications and how it can be integrated with frontend technologies to provide a smooth user experience. It is a useful platform for education and entertainment because of its user-friendly interface and strong backend logic, which enable users to explore the possibilities of logic-based programming.

Future developments could add more sophisticated features to the system, like bigger grids, more customizable difficulty settings, and cloud platform distribution for wider accessibility. Overall, the sudoku puzzle is a demonstration of declarative programming's strength and usefulness in building interactive, intelligent systems.

## References

[1] Sheng-Wei Wang .A Dataset of Sudoku Puzzles With Difficulty Metrics Experienced by Human Players.

[2] Desk,T.T.(2024). A Step-by-Step Guide to Playing Sudoku on Times of India. [online] The Times of India. Available at: https://timesofindia.indiatimes.com/technology/gaming/a-step-by-step-guide-to-playing-sudoku-on-times-of-india/articleshow/111783754.

[3] Yıldız Güven1, Cihat Gültekin1, A. Beyzanur Dedeoğlu1:Comparison of Sudoku Solving Skills of Preschool Children Enrolled in the Montessori Approach and the National Education Programs.

[4] Ahmad Khudzairi (2019).SOLVING DIAGONAL SUDOKU PUZZLE 9X9 GRID BY JAVA PROGRAMMING ALGORITHM.

[5] Sean (2023). How I built the backend for my Sudoku app - SLTC—Sean Learns To Code - Medium. [online] Medium. Available at: https://medium.com/sltc-sean-learns-to-code/how-i-built-the-backend-for-my-sudoku-app-843e0de81094.

[6] Ushir Yogesh(2019) : A Review on Different Strategies of Solving Sudoku Puzzles https://www.researchgate.net/publication/353118456 A Review on Different Strategie of Solving Sudoku Puzzles

[7] HollisWilliams (2024):The Mathematics of Sudoku https://www.researchgate.net/publication/386106991 The Mathematics of Sudoku November 2024.