Tribhuvan University

Faculty of Humanities and Social Sciences

8-PUZZLE GAME AND SOLVER

A PROJECT REPORT

Submitted to

Department of Computer Application

D.A.V. College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Himal Subedi

[6-2-469-161-2020]

November, 2024

Under the Supervision of

Er. Nihal Shrestha

Tribhuvan University

Faculty of Humanities and Social Sciences

D.A.V. College

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by HIMAL SUBEDI entitled "**8-PUZZLE GAME AND SOLVER**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

Er. Nihal Shrestha
SUPERVISOR
Lecturer
Bachelors in Computer Application
Jawalakhel, Lalitpur

Tribhuvan University

Faculty of Humanities and Social Sciences

D.A.V. College

LETTER OF APPROVAL

This is to certify that this project prepared by HIMAL SUBEDI entitled "**8-PUZZLE GAME AND SOLVER**" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Er. Nihal Shrestha Lecturer Bachelors in Computer Application Jawalakhel, Lalitpur	Mr. Sudip Adhikari Head Of Department Bachelors in Computer Application Jawalakhel, Lalitpur
Internal Examiner Ms. Shrilata Wagle Lecturer Bachelors in Computer Application Jawalakhel, Lalitpur	**External Examiner** Mr. Roshan Tandukar Asst. Professor Tribhuvan University

ACKNOWLEDGEMENT

In the pursuit of project completion, numerous individuals have extended their benevolent blessings and unwavering support. It is with sincere appreciation that I seize this moment to extend my heartfelt gratitude to each contributor.

I am profoundly indebted to Tribhuvan University for affording this remarkable opportunity. My gratitude extends to D.A.V. College for fostering an environment conducive to learning and providing the requisite infrastructure. I also wish to acknowledge Mr. Shashi Bhusan Chaturwedi, the BCA Department Coordinator, and Mr. Sudip Adhikari, our esteemed Department Head.

The indispensable guidance of Er. Nihal Shrestha has been a cornerstone, providing unwavering support and insightful direction throughout every project phase. This collaborative effort has undoubtedly shaped the project's success.

Lastly, I am compelled to express my heartfelt thanks to my parents and classmates whose unwavering support and assistance have been a constant throughout my journey in developing this project.

With Regards

Himal Subedi

TU Registration No: 6-2-469-161-2020

ABSTRACT

This project focuses on developing a web-based application for the 8-Puzzle Game and Solver. The system allows users to engage with the classic 8-puzzle game, where they must arrange numbered tiles in a 3x3 grid in the correct order by sliding them into an empty space. Additionally, the application incorporates an intelligent puzzle solver using the A* algorithm, which can solve user-input puzzles and display step-by-step solutions. The primary goal is to offer an interactive platform for users to play the 8-puzzle game and observe how algorithms solve such problems in real-time.

The project includes various functionalities such as randomizing solvable puzzles, checking for puzzle solvability, manually inputting puzzle configurations, and visualizing the solution path. Tools such as XAMPP, Visual Studio Code, and a web browser are employed to develop this system. This project aims to enhance user understanding of problem-solving techniques and algorithms through an engaging and educational interface.

Keywords: *Web-based Application, XAMPP, Visual Studio Code, 8-Puzzle Game and Solver, A* Algorithm*

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CSS	-	Cascading Style Sheet
DFD	-	Data Flow Diagram
ER	-	Entity Relationship Diagram
HTML	-	Hypertext Markup Language
IDE	-	Integrated Development Environment
JS	-	JavaScript
PHP	-	Hypertext Pre-Processor
SQL	-	Standard Query Language
VS	-	Visual Studio

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1 INTRODUCTION

1.1 Introduction

The 8-Puzzle Game and Solver is a dynamic web application designed to offer an engaging and interactive experience for enthusiasts of the classic 8-puzzle game. This application not only allows users to play the game by arranging the numbered tiles to reach the solved state but also provides a robust solver feature powered by the A* algorithm. With an intuitive interface, users can easily input their custom puzzle configurations and receive a step-by-step solution that demonstrates the optimal path to solve the puzzle. Whether for educational purposes or sheer enjoyment, this 8-Puzzle Game and Solver serves as a valuable tool for anyone interested in exploring problem-solving strategies and understanding the intricacies of the 8-puzzle challenge.

1.2 Problem Statement

Despite the enduring appeal of the classic 8-puzzle game, many enthusiasts find solving it manually to be both time-consuming and challenging. The traditional approach poses several difficulties: incorrect moves can lead to frustration, significant time and effort are required to manually find solutions, and the lack of a systematic strategy often results in suboptimal solutions, particularly for complex configurations.

The 8-Puzzle Game and Solver web application addresses these challenges by providing an interactive platform where users can play the game as well as solve it with ease and precision. Leveraging the powerful A* algorithm, the solver component of the application ensures that any puzzle configuration is solved efficiently and accurately. This eliminates the risk of frustration due to incorrect moves and offers an optimal solution every time. Additionally, the web-based nature of the application provides the convenience of solving puzzles from any device with a web browser and internet connection. The intuitive user interface and structured solving approach make it accessible and enjoyable for puzzle enthusiasts of all skill levels, enhancing both the gameplay and learning experience.

1.3 Objectives

The developed project helps users to play and solve 8-puzzle problems efficiently and accurately. It also eliminates the challenges associated with manual solving. So, the main objective of this project is:

- To develop a web-based 8-Puzzle Game and Solver application using A* algorithm

1.4 Scope and Limitations

The 8-Puzzle Game and Solver can be used in educational settings such as computer science departments, research institutes, and puzzle enthusiasts' communities. It serves as a tool for learning and teaching algorithms, problem-solving techniques, and artificial intelligence concepts. The system can be implemented in both academic environments and as a recreational web application, allowing users to either solve the puzzle themselves or observe the automated solver in action. Limitation of the project is:

- The system cannot provide hints or suggest optimal moves to help the user while playing the puzzle manually.

1.5 Report Organization

Chapter 1: This chapter provides an overview of the 8-Puzzle Game and Solver project, including its objectives and significance.

Chapter 2: This chapter explores existing systems and research related to 8-Puzzle Game and Solver to gain insights and knowledge.

Chapter 3: This chapter focuses on analyzing requirements and designing the 8-Puzzle Game and Solver.

Chapter 4: This chapter covers the implementation and testing phases of the 8-Puzzle Game and Solver.

Chapter 5: The final chapter summarizes the project's outcomes and presents recommendations for future enhancements.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

To ensure the development of an effective and user-friendly 8-puzzle game and solver application, it is important to consider the perspectives of both puzzle enthusiasts and researchers. Gathering insights from their experiences with traditional, manual puzzle-solving methods, including their expectations, preferences, and pain points, can inform the design and functionality of both the game and solver components. Factors such as ease of use, gameplay experience, algorithm efficiency, solution accuracy, and user interface design are taken into consideration to create a system that caters to the needs of all users involved.

Overall, the background study reveals the need for an 8-puzzle game and solver application that addresses the limitations of manual solving methods and aligns with the increasing demand for engaging and accurate puzzle solutions. By leveraging technological advancements and understanding the context of puzzle-solving, an effective and user-friendly solution can be developed to ensure optimal gameplay, reduce solving time, and enhance the overall puzzle-solving experience for users.

2.2 Literature Review

Solver of 8-Puzzle with Genetic Algorithm [1] published in the Journal of Applied Technology and Innovation, explores the application of Genetic Algorithms (GA) to solve the 8-puzzle problem, a well-known problem in artificial intelligence and mathematics. The authors provide a comprehensive approach by implementing genetic algorithms, inspired by Charles Darwin's theory of natural evolution, to generate, evaluate, and evolve populations of potential solutions. Through modifying parameters such as crossover rate, mutation chance, and population size, the study demonstrates that higher crossover rates and mutation chances generally lead to faster problem-solving, while larger population sizes tend to increase the time required to find solutions. The research highlights the efficacy of genetic algorithms in finding optimal solutions for the 8-puzzle problem,

although it acknowledges the trade-off in terms of computational time with varying parameter configurations.

Complete Solution of the Eight-Puzzle and the Benet of Node Ordering in IDA [2].This paper presents exhaustive statistical data from evaluating all possible configurations, providing insights into solution lengths, the easiest and hardest configurations, and the distribution of solution nodes. Additionally, the paper examines the benefits of node ordering in the IDA* algorithm. The findings highlight that common IDA* implementations with fixed node ordering perform worse compared to a simple random ordering of operators.

Performance Analysis of Various Uninformed and Informed Search Strategies on 8 Puzzle Problems [3].The research conducted by Vinay Menon and colleagues at Vellore Institute of Technology concluded that the A* algorithm is the most efficient for solving the 8-puzzle problem. By comparing various search strategies including Breadth First Search (BFS), Depth First Search (DFS), Hill Climbing, and Steepest Ascent Hill Climbing, they found that A* consistently required fewer node explorations and moves to reach the goal state. This efficiency is attributed to A*'s use of a combined heuristic that optimally balances the cost already incurred and the estimated cost to the goal. Their experiments demonstrated that while other algorithms sometimes failed to find a solution or took longer, A* reliably provided the best performance across different test cases.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

System analysis involves a comprehensive evaluation of the 8-Puzzle Game and Solver to identify its functional and non-functional requirements, analyze user needs, and ensure efficient system performance. Through systematic examination and modeling of both the gameplay and puzzle-solving processes, system analysis aims to understand the existing algorithms, data flow, and user interactions to design an optimal and user-friendly solution. This analysis will help identify essential features such as puzzle input methods, game mechanics, visualization of solving steps, and performance metrics, ensuring that the application meets the needs of both casual players and advanced puzzle enthusiasts while providing an intuitive interface for a seamless user experience.

3.1.1 Requirement Analysis

Requirement analysis encompasses the identification and documentation of both functional requirements, which define the system's desired functionality, and non-functional requirements, which specify the system's performance, usability, and other quality attributes. It also includes various diagrammatic figures that lets us understand the core backbone of the system and how it functions.

i. Functional Requirement

Functional requirements are crucial for a project as they define the specific features and behaviors the system must have to meet user needs and project goals. They serve as a roadmap for development, ensuring the system delivers the desired functionalities and outcomes. The functional requirements for this project are as follows:

i. User Login: To validate user and to track their progress.

ii. Puzzle Solving: To solve the given puzzle

iii. Solution Visualization: Showing step by step solution of the puzzle.

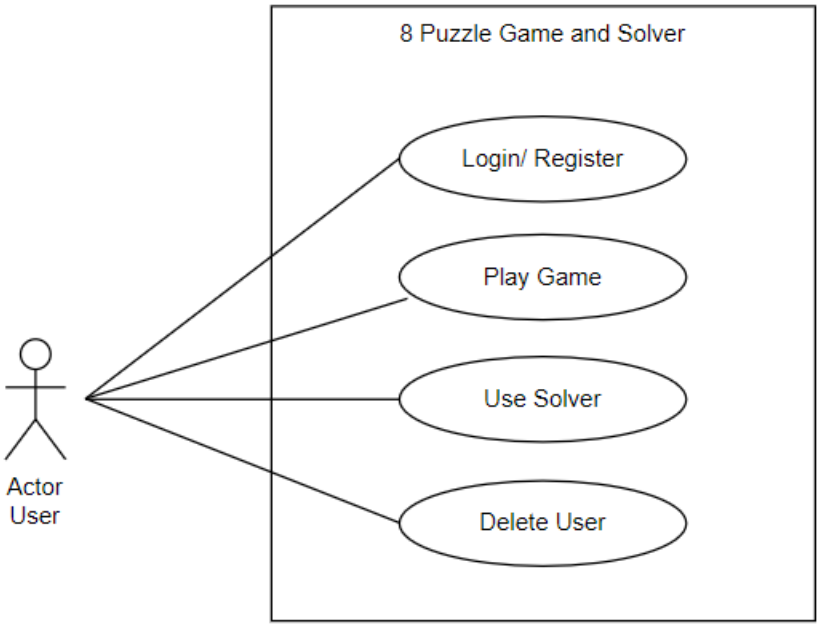


Figure 3. 1: Use Case Diagram of 8-puzzle Game and Solver

ii. Non-Functional Requirement

Non-Functional Requirement specifies criteria that can be used to judge the operation of the system. Unlike functional requirement these are non- lethal but having these requirements will make the project more successful. The non-functional requirements in this project are:

a. Usability: The system should have a user-friendly interface that is intuitive and easy to navigate, allowing users to input puzzle data conveniently.

b. Reliability: The system should be reliable and available for use at all times, minimizing any downtime or disruptions in service.

c. Performance: The system should be able to solve puzzle in reasonable time frame.

3.1.2 Feasibility Analysis

i. Technical Feasibility

The 8-Puzzle Game and Solver will be made using the following technically available resources:

A. For frontend use:

i. HTML

ii. CSS

B. For backend and data storage:

i. PHP

ii. JavaScript

iii. MySQL

As the above mention technology is easily available and are more than enough for the project's need. This project is technically feasible.

ii. Operational Feasibility

Operational feasibility is measured by how well the 8-puzzle game and solver will support both users and developers during its operational phases. This application enhances the puzzle-solving and gameplay experience by providing an engaging interface and efficient solutions to the 8-puzzle problem. By employing the A* algorithm, the solver component ensures optimal performance, meeting users' needs for effective puzzle-solving while allowing developers to maintain high standards of software quality and usability. The user-friendly interface facilitates easy input, interactive gameplay, and visualization of solutions, making it accessible for both casual players and researchers, ultimately fostering a supportive environment for continued engagement with the puzzle-solving community.

iii. Economic Feasibility

An 8-Puzzle Game and Solver application can yield significant economic benefits for both casual users and educational institutions. First, it can reduce the time and effort associated with manually solving puzzles, allowing users to enjoy the game while also learning and

applying problem-solving strategies without frustration from challenging configurations. Second, it can boost educational productivity by serving as a valuable teaching tool in mathematics and computer science courses, helping students grasp concepts related to algorithms. Third, the application increases accessibility and convenience, enabling users to play and solve the 8 Puzzle from anywhere with an internet connection, which is especially advantageous for remote learners and puzzle enthusiasts. Overall, implementing an effective 8-puzzle game and solver application can provide a strong return on investment for educational institutions and individual users alike, making it an economically viable solution for enhancing problem-solving skills, promoting engagement with computational thinking, and offering an enjoyable gaming experience.

iv. Scheduling Feasibility

This includes the project schedule and time allocated for their completion. The Gantt chart for the project is: -

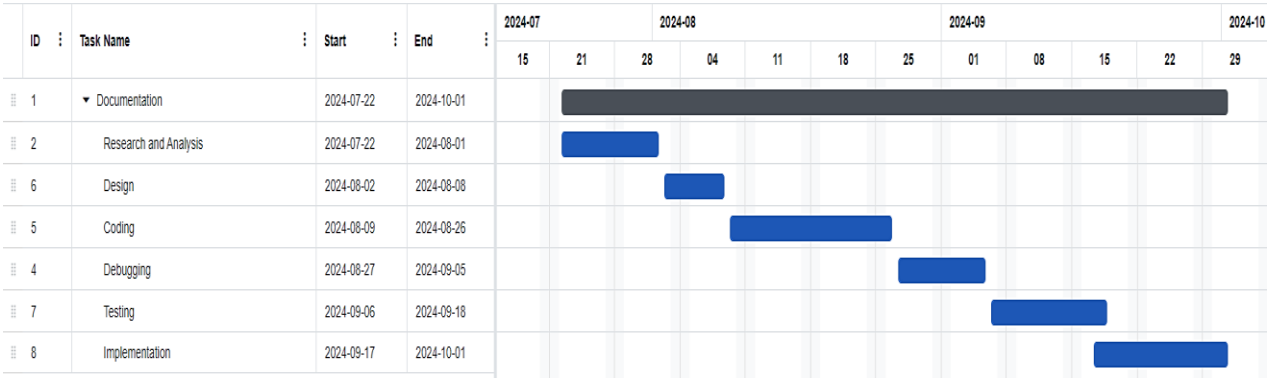


Figure 3. 2: Gantt Chart of 8-puzzle Game and Solver

3.1.3 Data Modeling

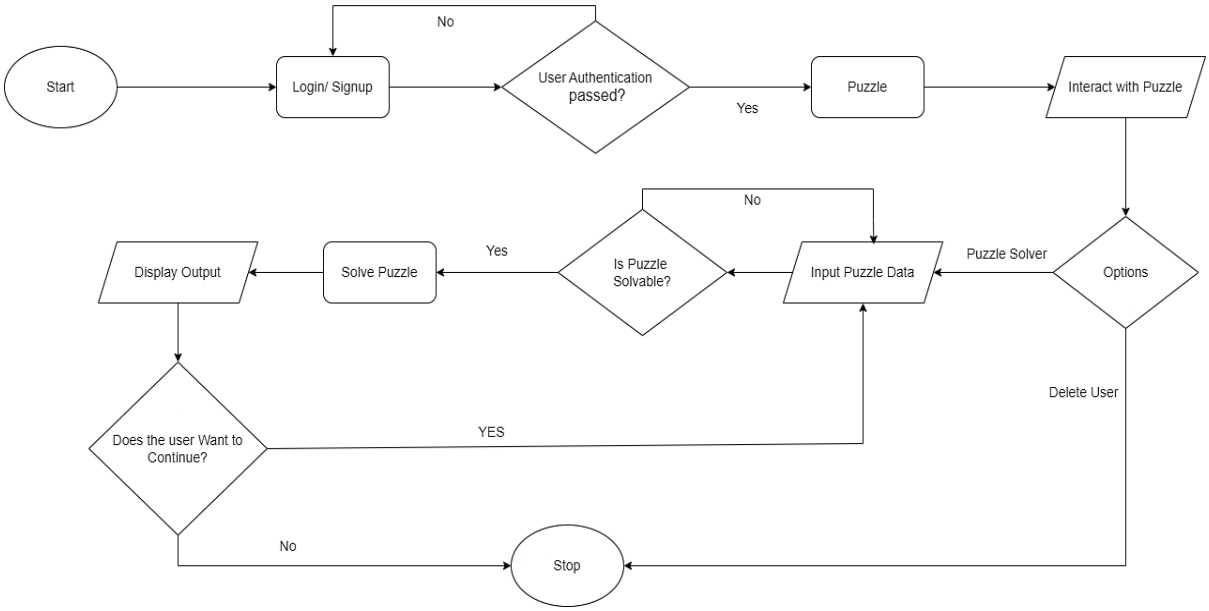


Figure 3. 3: Flow Chart of 8-puzzle Game and Solver

This flowchart provides a clear picture of how the flow of operations occurs in the 8-Puzzle Game and Solver and how the system functions. The system begins with a login, guiding users through the authentication process and various other processes until they decide to close the application. It visually outlines each step, including the puzzle game, puzzle input, solving algorithms, and the display of results, ensuring a seamless user experience. This comprehensive view enhances clarity and facilitates better understanding of the system's functionality.

3.1.4 Process Modeling

Process modeling is the practice of creating visual representations that depict the sequence of activities, decisions, and interactions within a system The level 0 and Level 1 Data Flow Diagram for the project is in figure 3.4 and figure 3.5 respectively.

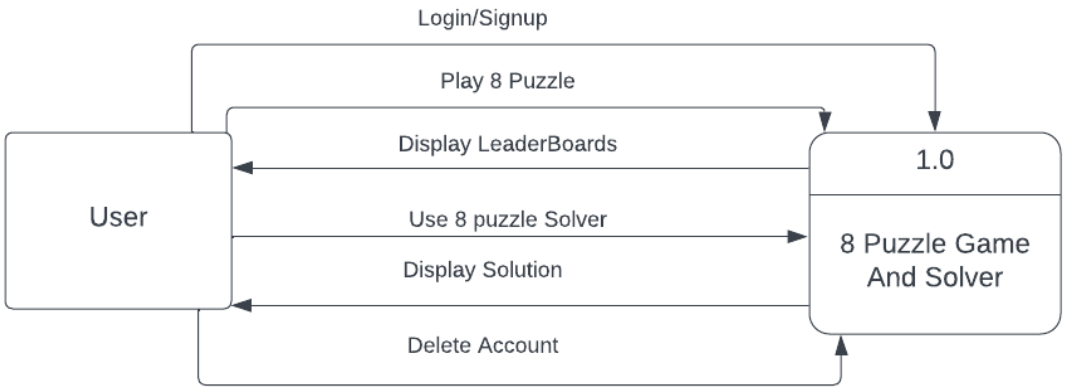


Figure 3. 4: Level-0 Data Flow Diagram of 8-puzzle Game and Solver

The Level 0 Data Flow Diagram shows how the user interacts with the 8 Puzzle Solver, the user is able to login/signup and after that is able to play the puzzle game or input puzzle data which the 8 Puzzle Solver process and gives an appropriate response back. 8 Puzzle Game also displays the data of top user that had completed the puzzle in the shortest of time. The user is also able to delete their account if they choose to do so.

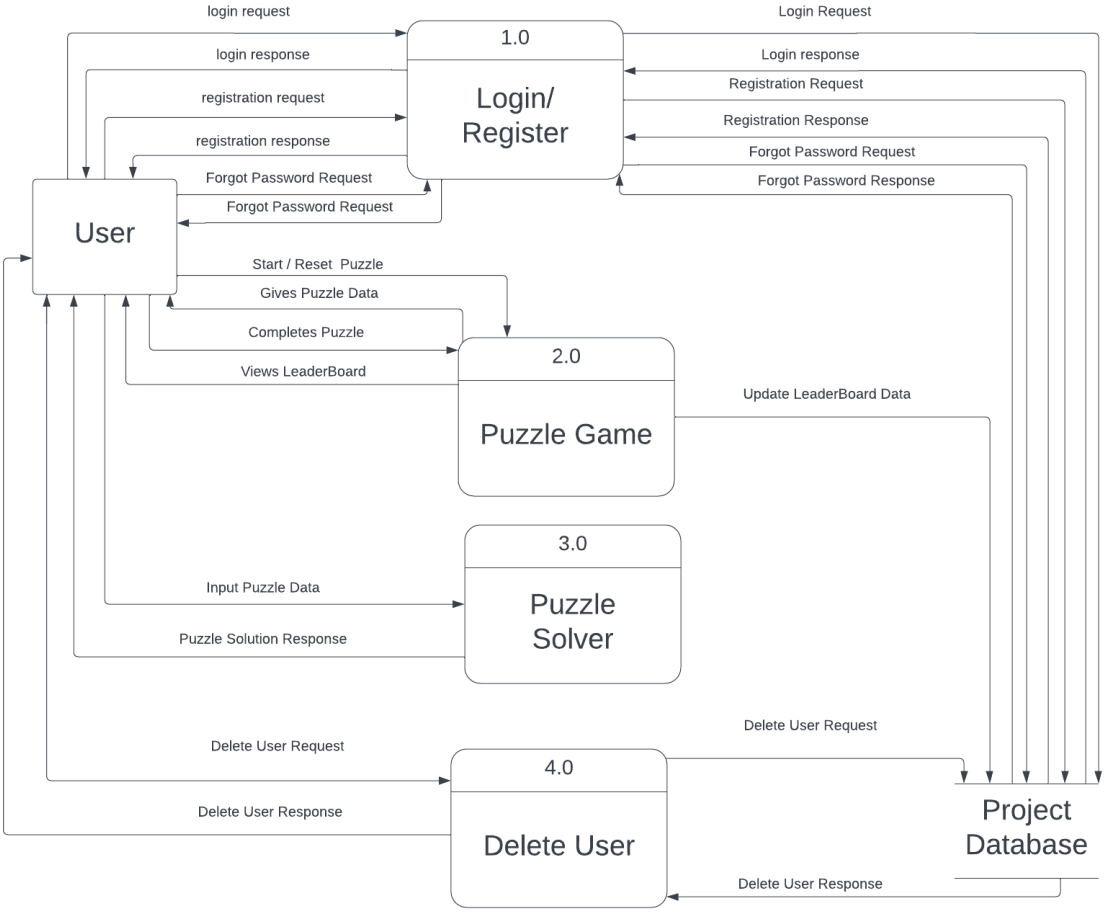


Figure 3. 5: Level-1 Data Flow Diagram of 8-puzzle Game and Solver

The level 1 DFD shows the various processes inside the 8 Puzzle Solver in detail. It shows how the data flows throughout the system and how each and every process communicate with each other and the database for the system as a whole to function properly and complete its intended use.

3.2 SYSTEM DESIGN

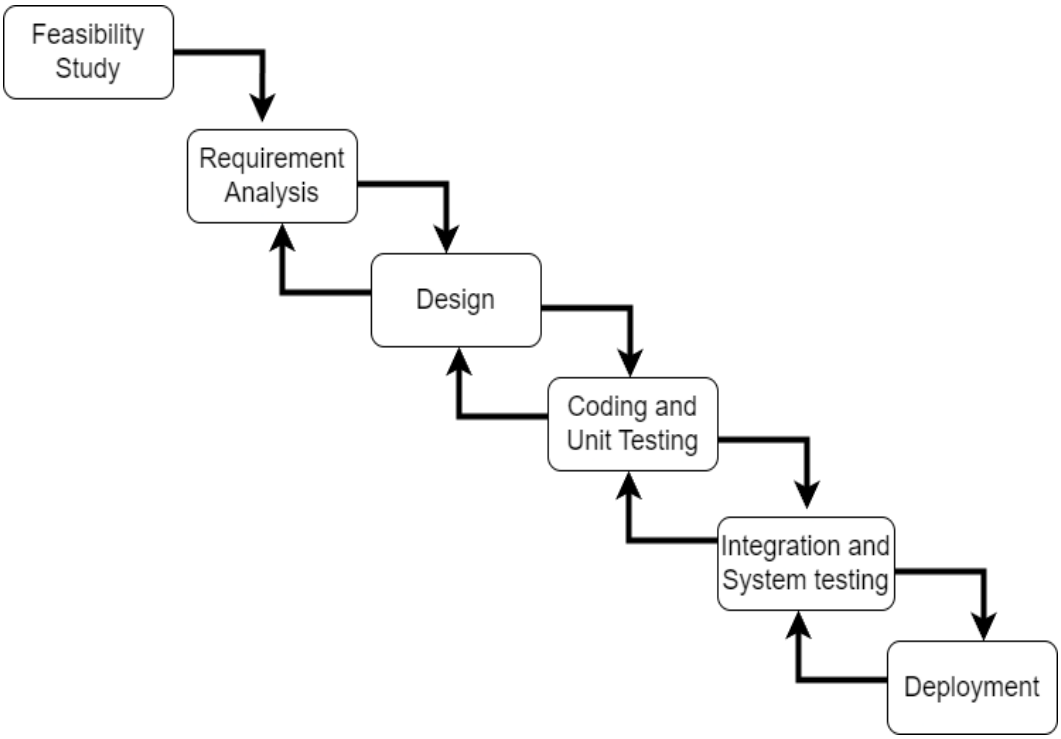


Figure 3. 6: Iterative Waterfall Model of 8-puzzle Game and Solver

The feasibility study phase ensured the project's viability by addressing the technical, operational, economic, and functional requirements of the 8-Puzzle Game and Solver. The requirement analysis phase comprehensively examines the necessary functionalities of both the game and solver and how they impact user interaction, gameplay experience, and problem-solving efficiency. During the design phase, decisions are made regarding the interactions between different modules, their functionalities, and the visual aspects of the application interface.

Coding will be carried out using Visual Studio Code, with unit testing conducted to ensure thorough testing of all project modules. The integration and system testing phase will evaluate inter-module interactions, gameplay mechanics, and the overall performance of the game and solver. As a practical environment often reveals errors in each development phase, the iterative waterfall model is employed, allowing for error correction in earlier stages and providing a structured approach for the project’s development. Consequently, the iterative waterfall model was selected for the 8-Puzzle Game and Solver project.

3.2.1 Architecture Design

Architecture design involves creating a high-level blueprint that defines the structure, components, and interactions of a system. The architecture design for this project is in figure 3.7:

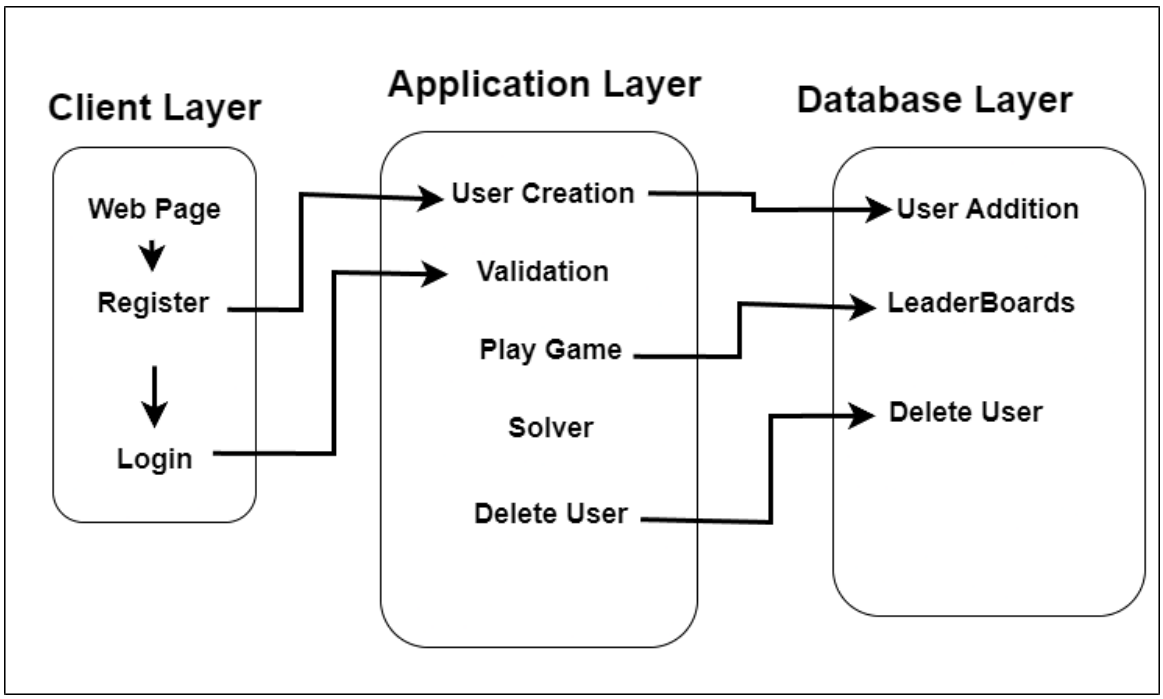


Figure 3. 7: Architecture Design of 8-puzzle Game and Solver

The Architectural Design Diagram illustrates the interaction between the different system layers, ensuring seamless functionality. The Client Layer represents the user interface, comprising various webpages through which users interact with the system. The Application Layer functions as a bridge, managing communication between the client interface and the underlying data processes. It handles the core logic and system operations. The Database Layer securely stores all critical information, such as user credentials and leaderboard data, ensuring efficient data retrieval and management. Together, these layers provide a cohesive and well-structured system.

3.2.2 Database Schema Design

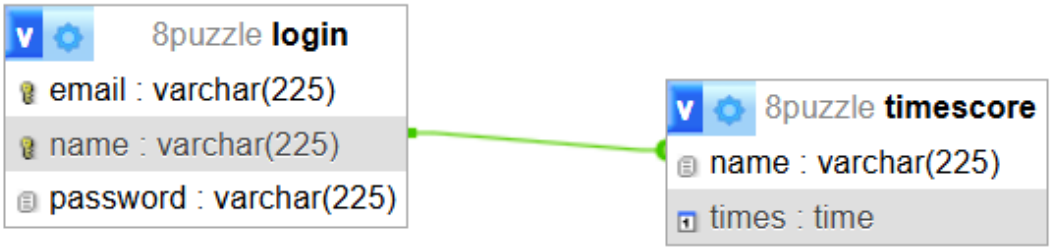


Figure 3. 8: Database Schema Diagram of 8-puzzle of Game and Solver

The Data Schema Diagram clearly illustrates the database structure for the 8-Puzzle Game and Solver Application. The database consists of two main tables: login and timescore. The login table stores user details, including email, name, and password, with the email serving as the primary key. The name field also has a unique constraint, ensuring that each user has a distinct display name. This unique name is then used as a foreign key in the timescore table, which tracks completion times for the leaderboard. The relationship between the two tables ensures that each user's scores are properly linked to their profile, enabling efficient management of leaderboard rankings.

3.2.3 Interface Design

Interface design involves creating intuitive and user-friendly interfaces for software applications, websites, or systems. It focuses on designing visually appealing layouts, organizing information in a logical manner, and incorporating interactive elements to enhance user experience. The various interface design for this project are:

Figure 3. 9: Login Page Design of 8-puzzle Game and Solver

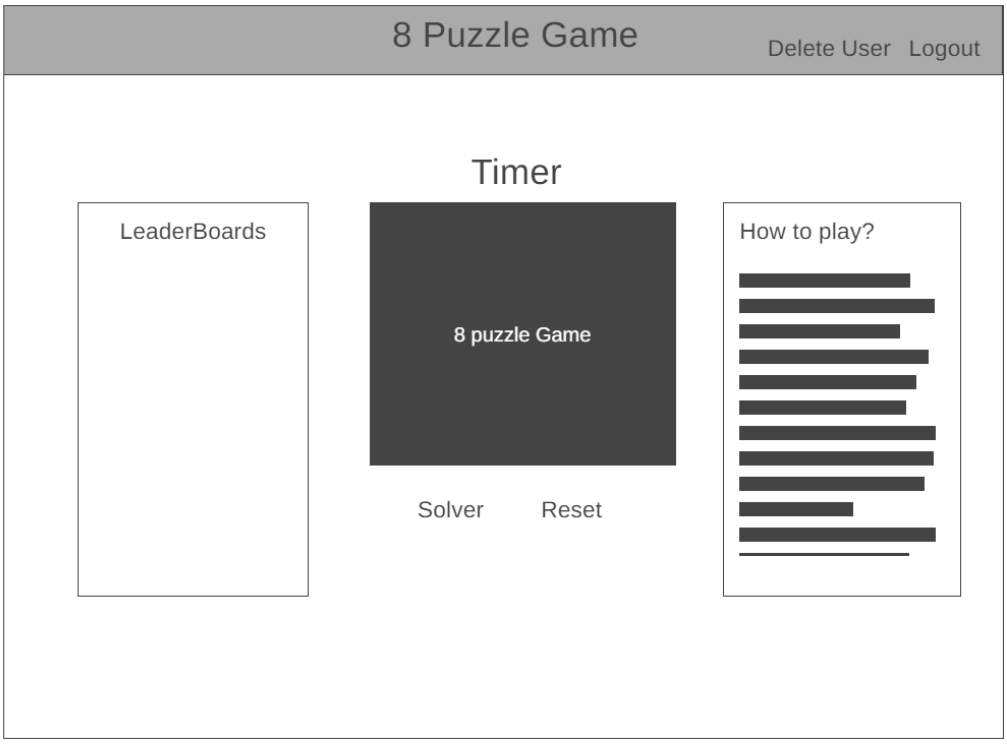


Figure 3. 10: 8-Puzzle Game Design

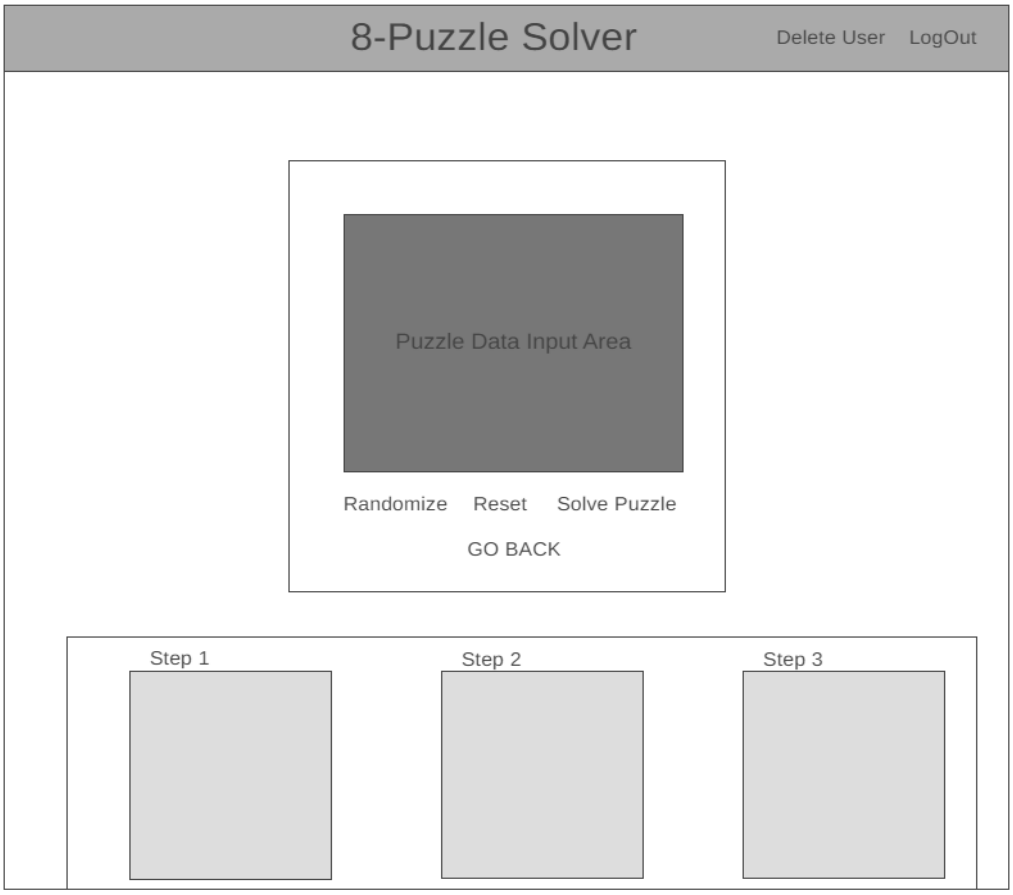


Figure 3. 11: 8-Puzzle Solver Design

3.3 Description of Algorithm

The algorithm for solving the 8-puzzle leverages the A* search strategy, combining the current state of the puzzle with a heuristic approach to find the optimal solution efficiently. Initially, the algorithm assesses the puzzle's current configuration and generates neighboring states by sliding tiles into the empty space. It calculates a score for each state based on the cost to reach that state and an estimated cost to reach the goal state, using the Manhattan distance as the heuristic. The algorithm maintains a priority queue of states to explore, selecting the one with the lowest score for further evaluation. This process continues iteratively until the algorithm finds the solution or exhausts the maximum iteration limit, ensuring an effective and systematic resolution of the puzzle. The step-by-step procedure of 8-puzzle solver are:

Step 1: The function existSolution calculates whether a given permutation is solvable based on the inversion count. An inversion occurs when a pair of tiles are in the wrong order relative to each other .i.e. in a pair of tile a, b such that a>b but a appears before b in the list. If the number of inversions is even, the puzzle is solvable.

Step 2: If the inversion count is odd, a new permutation is generated, and the process repeats until a solvable permutation is obtained.

Step 3: The A* algorithm uses the formula to calculate the f(n) value for each state. Formula for A* algorithm is f(n) = g(n) + h(n), where g(n) = depth of node and h(n) = Number of misplaced tiles.

Step 4: For every step the tiles are moved the value of f(n) is recalculated. And move which gives out the lowest value of f(n) is selected. If two state gives the same value of f(n) then both the solutions are explored.

Step 5: If the current state matches the goal state, the solution path is reconstructed, and the process ends.

Step 6: For each current state, its neighbors are generated by moving the blank tile i.e. 0 which is moved in valid directions.

Step 7: The heuristic function, h(n), is calculated using the Manhattan Distance. It measures the distance between the current position of each tile and its goal position by comparing the tiles that are not in their correct position.

Step 8: For each neighbor, if it offers a better path than previously known, it is added to the open set, and its g(n) and f(n) are updated. If a state is completely explored that it is added to closed set, after which no further exploration is done.

Step 9: The algorithm continues exploring states with lower f(n) score inside the openset until the goal state is found or the maximum iteration limit is reached.

Step 10: If a solution is found, the reconstructPath function traces the steps back to the initial state, and the solution is animated on the grid.

Step 11: If no solution is found within the iteration limit, a failure message is displayed.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools Used

Various tools that have been used in this project is listed below:

i) Microsoft Visual Studio:

Microsoft Visual Studio is a robust and feature-rich integrated development environment (IDE) that facilitates efficient software development across multiple platforms. It offers a comprehensive suite of tools and resources for coding, debugging, and testing applications. As visual studio is user-friendly and supports all programming language that is used in this project. It is used as an IDE for this project.

ii) XAMPP:

XAMPP is a widely used open-source software package that provides developers with a complete web development environment. It combines Apache, MySQL, PHP, and Perl to create a local server environment for building and testing web applications.

iii) Web Browser:

A web browser is a critical software application that enables users to access and interact with websites on the internet. It interprets HTML, CSS, and JavaScript code to render web pages and provides a user-friendly interface for browsing the internet. Microsoft Edge has been used in this project.

4.1.2 Implementation Details of Modules

Implementing modules in the 8-Puzzle Game and Solver involves developing and integrating specific features and functionalities to enhance efficiency, user experience, and operational aspects of the system. The following key modules are implemented:

1. 8-Puzzle Game

This is the primary module where users can engage in playing the 8-Puzzle. The module

provides an interactive interface for users to rearrange puzzle pieces to achieve the goal state. Key features within the game module include:

a. Timer: The timer runs continuously while users solve the puzzle. Upon successful completion of the puzzle, the elapsed time is recorded along with the user's username in the database. This feature not only adds a challenge to the game but also tracks performance for competitive rankings.
b. Leaderboard: The leaderboard displays the top 8 players who have completed the puzzle in the shortest time. This competitive feature encourages users to improve their performance, fostering a sense of achievement and community among players.
c. How to Play: A comprehensive guide provides detailed instructions on how to play the puzzle game. It explains the rules and objectives, making it easy for both new and experienced players to understand the gameplay mechanics. This feature ensures that users of all levels can enjoy the puzzle without confusion.
d. Logout: Users can easily log out from the game page at any time, ensuring that their session is terminated securely.
e. Delete Account: Users have the option to delete their account by providing the correct password. This action permanently removes their data from the system, giving users full control over their personal information.
f. Reset: he reset feature allows users to restart the puzzle with a new configuration of tiles. This is useful when users want to challenge themselves with different puzzle data or simply start over after being stuck.
g. Solver: For users who are unable to solve the puzzle or are interested in learning the solution process, the solver feature automatically solves the puzzle using the A* Algorithm. It provides a step-by-step explanation of the solution, making it an excellent learning tool.

2. 8-Puzzle Solver

The solver is a powerful feature that uses the A* Algorithm to solve the puzzle and display the steps involved in reaching the solution. This module helps users who are stuck or want to analyze the solution path for educational purposes. The solver module offers the following functionalities:

a. Reset: Clears the current puzzle data, allowing users to input their custom puzzle to be

solved. This feature is especially useful for users who wish to test specific configurations and see how the solver tackles different challenges.

b. Generates a random solvable puzzle configuration, allowing users to see the solver in action on a variety of puzzles. This feature adds variety and unpredictability to the puzzle-solving experience, making the game more engaging.
c. Solve Puzzle: The solver computes the solution using the A* Algorithm and presents the steps to the user. Each move is shown sequentially, demonstrating how the puzzle can be solved optimally. This feature is educational as it highlights the logic behind the solution.
d. Go Back: Allows users to return to the main puzzle game page after using the solver. This feature ensures smooth navigation between different modules, improving the user experience.
e. Logout: Users can log out from the solver page, ensuring their session is securely terminated.
f. Delete Account: Similar to the main game page, users can delete their account from the solver module by entering a valid password, thus giving them control over their personal data even while using the solver.

4.2 Testing

Software testing is a crucial phase in software development, guaranteeing that the application operates as envisioned, is devoid of glitches, and aligns with user demands. By methodically assessing and addressing concerns prior to rollout, testing ensures a smooth user journey and overall software dependability. This integral process elevates product quality and user contentment.

4.2.1 Test Case for Unit Testing

Test Case: 1

Table 4. 1 Login Testing

S.no	Test case	Input	Expected result	Actual result	Status
1.	Login	Entering correct login credential i.e., username, password	To enter the 8-Puzzle Page	Didn’t Enter into the page	Failed
2.	Login	Entering correct login credential i.e., username, password	To enter the Puzzle Page	Entered into Puzzle Page	Success

Test Case: 2

Table 4. 2 Register and Forgot Password Testing

S.no	Test case	Input	Expected result	Actual result	Status
1.	Register of user	Entering user credential i.e., email, username, password.	OTP verification and Adding of the user to the database along with its credentials	OTP wasn't sent to the given email.	Failed
2.	Register of user	Entering user credential i.e., email, username, password.	OTP verification and Adding of the user to the database along with its credentials	OTP was sent to the given email and the user data was added on the database when correct OTP was entered.	Success
3.	Forgot Password	Entering Registered Email, New password and Correct OTP	OTP sent to email address and change of password on the input of correct OTP.	Password Changed Successfully.	Success
4.	Forgot Password	Entering invalid email, which was not registered to the database.	Give error message that the email doesn't exist	Error Message was not shown.	Failed
5.	Forgot Password	Entering invalid email, which was not	Give error message that the	Error Message was shown.	Success

		registered to the database.	email doesn't exist		

Test Case: 3

Table 4. 3 Puzzle Game Testing

S.no	Test case	Input	Expected result	Actual result	Status
1.	Moving of puzzle tiles.	Clicking the corresponding tile to move to the empty space.	Movement of tile which was pressed.	The tile was moved.	Success
2.	Reset Functionality	Pressing the Reset Button	Solvable puzzle to be shown with even inversion.	Fatal Error is shown.	Failed
3.	Reset Functionality	Pressing the Reset Button	Solvable puzzle to be shown with even inversion	Unsolvable puzzle was shown with odd inversion.	Failed
4.	Reset Functionality	Pressing the Reset Button	Solvable puzzle to be shown with even inversion	Error with inversion calculation	Failed
5.	Reset Functionality	Pressing the Reset Button	Solvable puzzle to be shown with even inversion	Solvable puzzle was shown	Success
3.	Leaderboards	Viewing the leaderboards	List of the top 8 players with the least solve time.	Fatal Error is shown for foreign key constraint	Failed

				failed in timescore.	
4.	Leaderboards	Viewing the leaderboards	List of the top 8 players with the least solve time.	The list showed the players but not in ascending order of time.	Failed
5.	Leaderboards	Viewing the leaderboards	List of the top 8 players with the least solve time.	The List of players in proper order was shown.	Success
6.	Deletion of User	Entering valid password.	Deletion of user	Deletion of user.	Success
7.	Logout	Pressing the Logout Button.	Destruction of Session and redirect to Login	Destruction of Session and redirect to Login.	Success
8.	Solve Puzzle	Solving the puzzle	Alert Stating the puzzle was solved, and time taken to solve it	Alert didn't popup.	Failed
9.	Solve Puzzle	Solving the puzzle	Alert Stating the puzzle was solved, and time taken to solve it	Alert popup worked.	Success
10.	Timer	Viewing the timer	Timer gradually increasing accurately.	Time increase accurately.	Success

Test Case: 4

Table 4. 4 Solver Testing

S.no	Test case	Input	Expected result	Actual result	Status
1.	Data Validation Check	Entering duplicate and invalid numbers	Error about duplicate and invalid number	No Error was shown	Failed
2.	Data Validation Check	Entering duplicate and invalid numbers	Error about duplicate and invalid number	Error shown only about the invalid number	Failed
3.	Data Validation Check	Entering duplicate and invalid numbers	Error about duplicate and invalid number	Error about duplicate and invalid number	Success
4.	Reset Functionality	Click Reset Button	Clear the data inside the solver	Clear the data inside the solver	Success
5.	Randomize Functionality	Click Randomize Button	Give solvable set of random numbers with even inversion.	Numbers weren't generated	Failed
6.	Randomize Functionality	Click Randomize Button	Give solvable set of random numbers with even inversion.	Random number was generated but wasn't solvable	Failed
7.	Randomize Functionality	Click Randomize Button	Give solvable set of random numbers with even inversion.	Solvable set of random numbers were shown.	Success

8.	Solve Puzzle	Click on the Solve Button	Solve the puzzle	Error was displayed.	Failed
9.	Solve Puzzle	Click on the Solve Button	Solve the puzzle	The Puzzle was Solved	Success
10.	Solution Visualization	Click on the Solve Button	Solution Steps was shown	Solution Steps was shown	Success

4.2.2 Test Case For System Testing

Test Case: 5

Table 4. 5: System Testing

S.no	Test case	Input	Expected result	Actual result	Status
1.	Transfer of puzzle data from puzzle to solver.	Clicking the Solve Button	Puzzle values transfer puzzle to solver	The puzzle Array didn’t transfer.	Failed
2.	Transfer of puzzle data from puzzle to solver.	Clicking the Solve Button	Puzzle values transfer puzzle to solver	The puzzle Array got transferred.	Success
3.	Transfer of email from login to the puzzle via Session	Logged in using respective credentials	Email Passed via session	Email Passed via session	Success

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION

5.1 Conclusion

The 8-Puzzle Game and Solver is an online tool designed to challenge users with solving the classic 8-puzzle game while also providing an efficient solution through an algorithmic solver. The platform allows users to engage with the puzzle, and offers a step-by-step guide using the A* algorithm for when they require assistance. The main aim of this project is to provide both entertainment and education, helping users improve their problem-solving skills while understanding algorithmic approaches to puzzles, thereby making the process both engaging and informative.

5.2 Lesson Learnt

This project provided valuable insights into the development of the 8-Puzzle Game and Solver, as well as how each module interacts to form a cohesive system. It highlighted how various features complement one another, contributing to the overall functionality and success of the project. Some of the key lessons learned during the development of this project include:

- Understanding algorithms: Gained in-depth knowledge about the A* Algorithm and how it can be applied to solve complex problems like the 8-Puzzle.
- User interface design: Developed skills in creating intuitive and user-friendly interfaces for puzzle games.
- Learning about testing: Implemented various testing methods to ensure the game and solver modules function correctly under different conditions.
- Integration of various features: Discovered how different modules, such as the game, solver, and leaderboard, integrate seamlessly to enhance the user experience.
- Maintaining documentation: Learned the importance of maintaining comprehensive documentation to ensure future maintenance and scalability of the project

5.3 Future Recommendation

- Integration of more complicated puzzles like 15 Puzzle.
- Adding hint feature that can suggest optimal moves or strategies to help users improve their problem-solving skills without relying entirely on the solver.
- Adding responsive mobile view Feature.

REFERENCE

[1] V. P. S. H. L. T. C. w. Z. A. A. S. Cheng Min Yang, "Solver of 8-Puzzle with Genetic Algorithm," Asia Pacific University of Technology, 2023. [Online]. Available: https://jati.sites.apiit.edu.my/files/2022/12/Volume7_Issue1_Paper6_2023.pdf. [Accessed 1 August 2024].

[2] A. Reinefeld, Paderborn Center for Parallel Computing, [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f74460d8130a5e580e16f671d4fcf227d26790ab. [Accessed 1 August 2024].

[3] G. B. A. G. H. P. Vinay Menon, Vellore Institute of Technology, [Online]. Available: https://wwjmrd.com/upload/performance-analysis-of-various-uninformed-and-informed-search-strategies-on-8-puzzle-problems---a-case-study_1546243350.pdf. [Accessed 1 August 2024].

APPENDICES

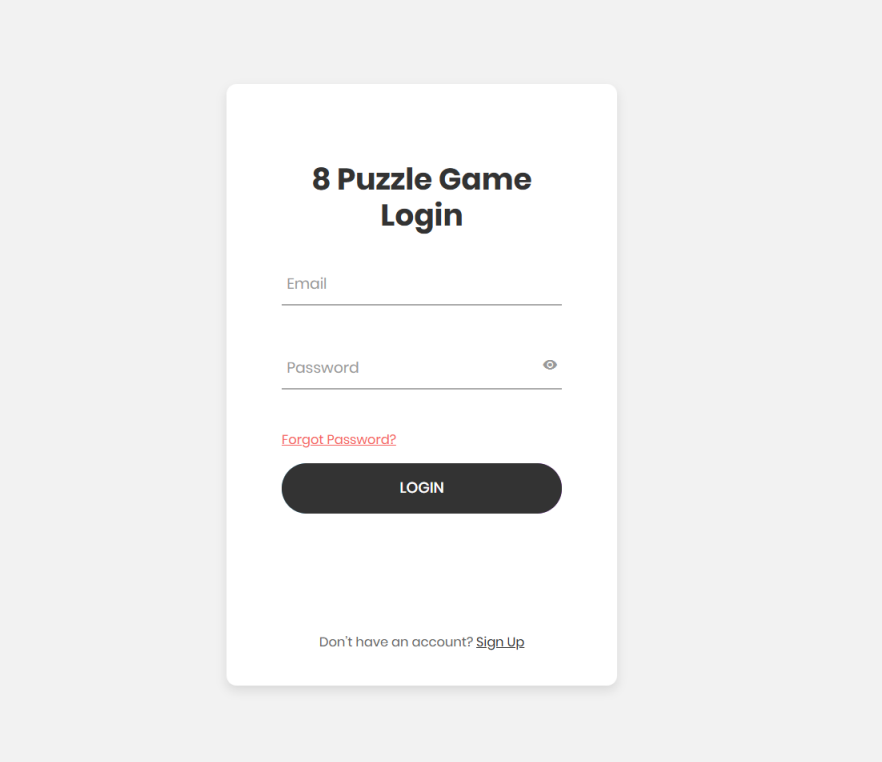


Figure 1: Login Page

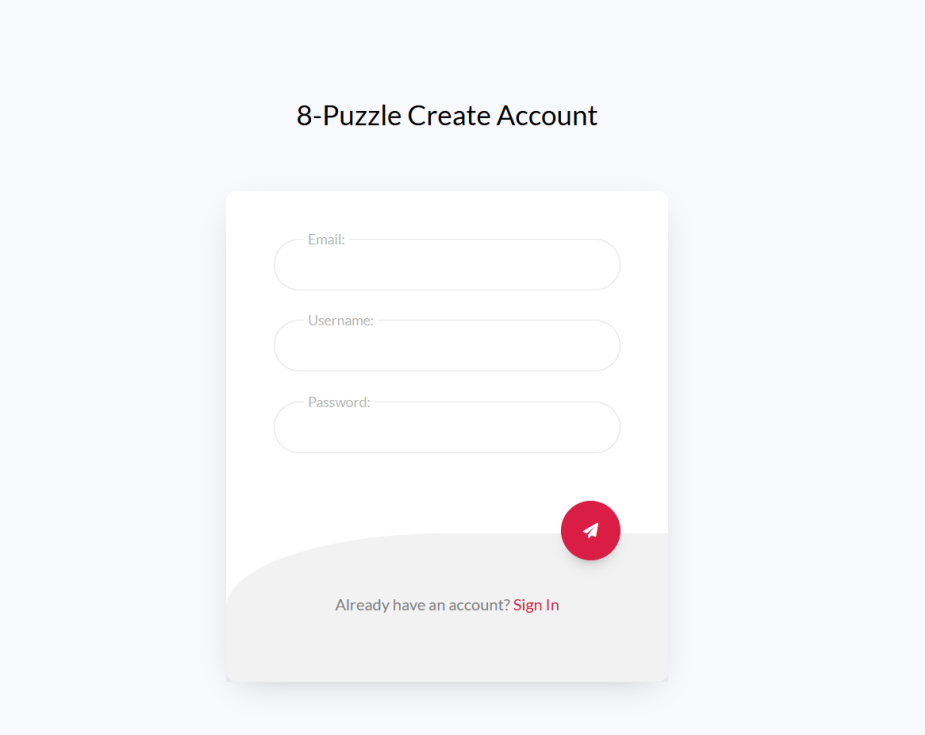


Figure 2: Register Page

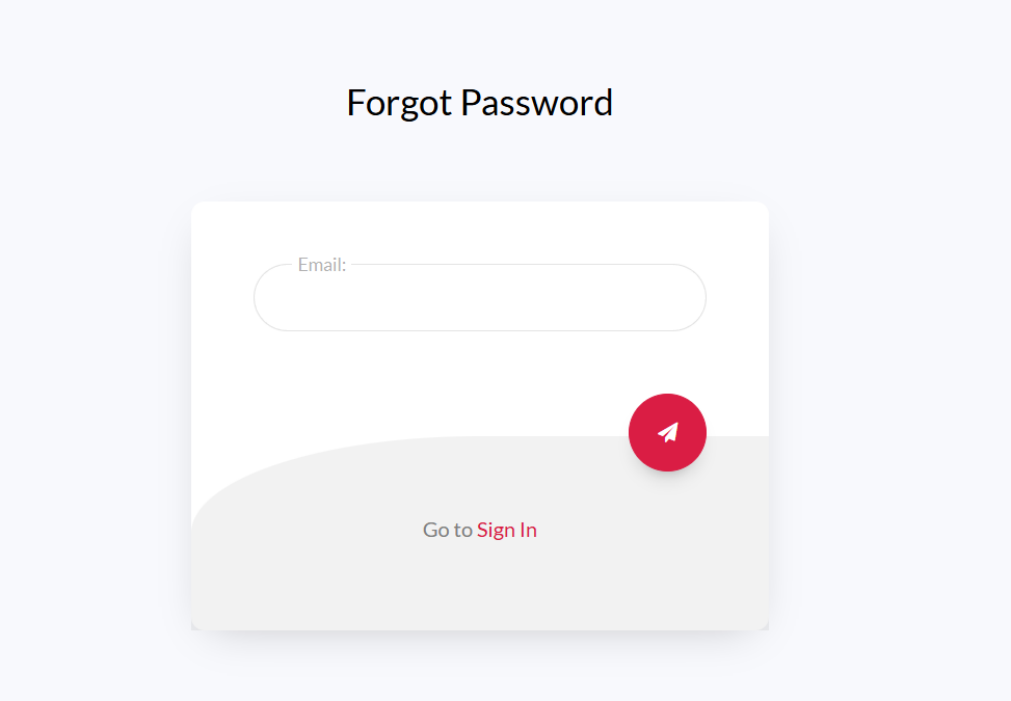


Figure 3: Forgot Password Page

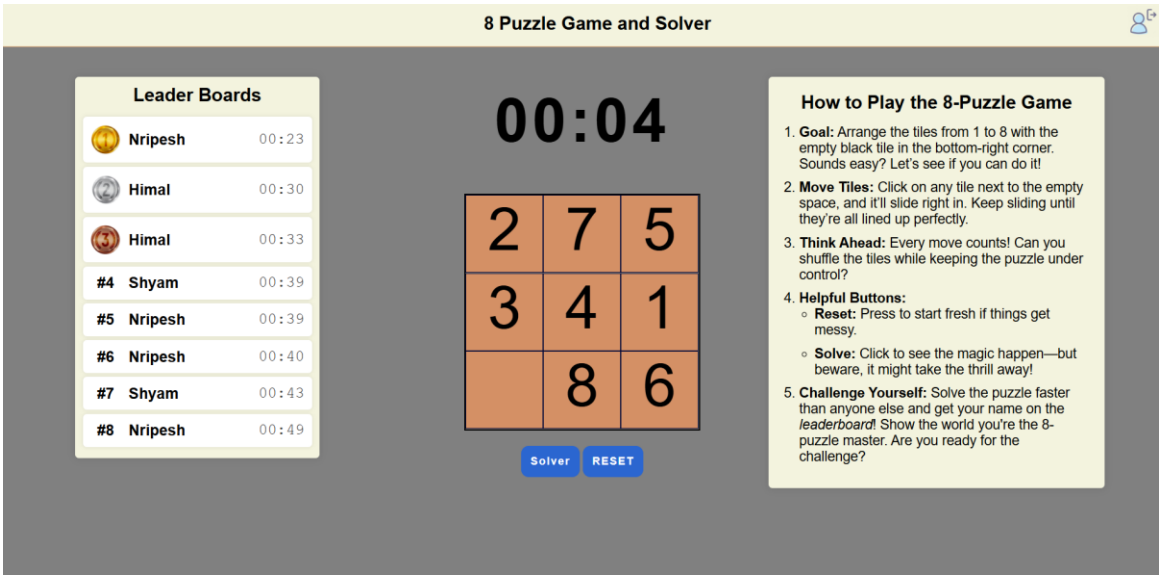


Figure 4: 8-Puzzle Page

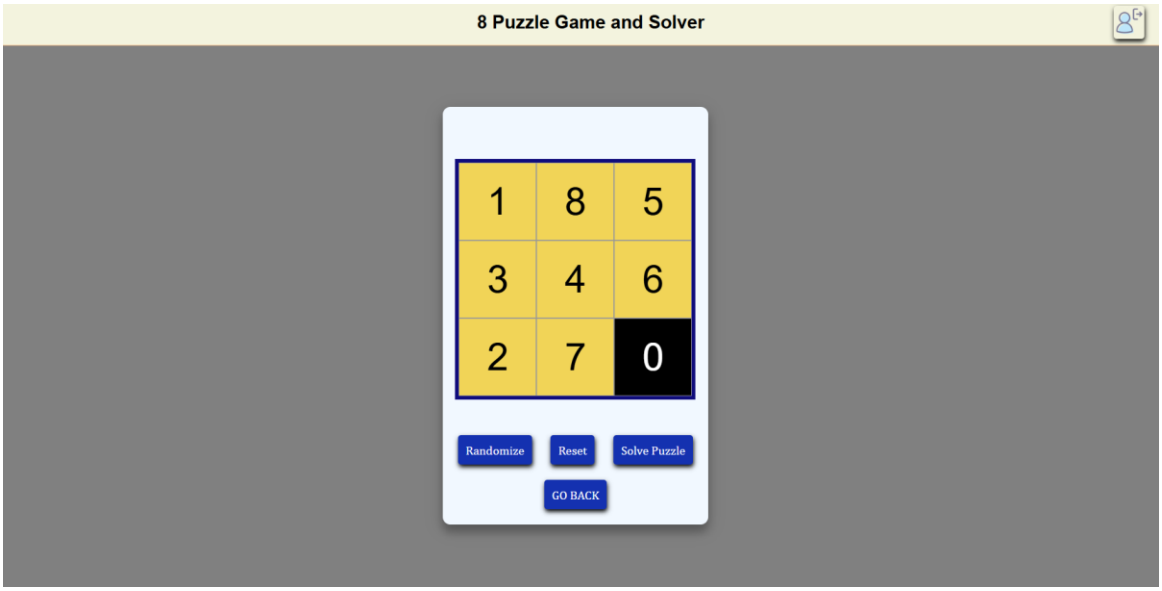


Figure 5: Puzzle Solver Page

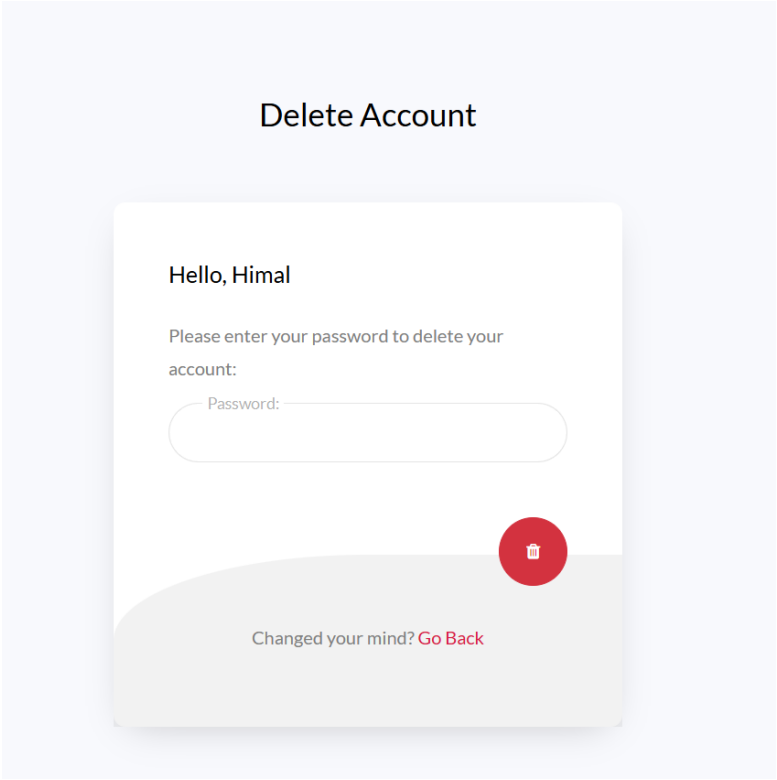
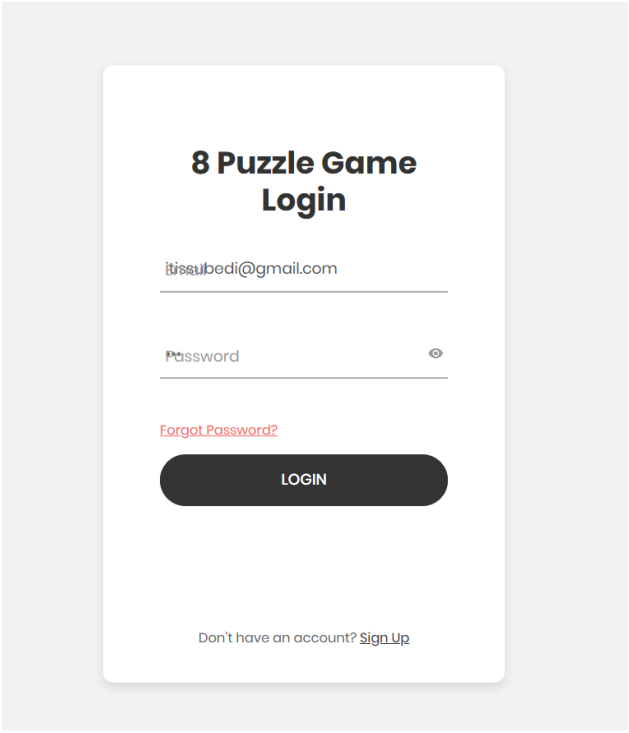


Figure 6: Puzzle Solver Page



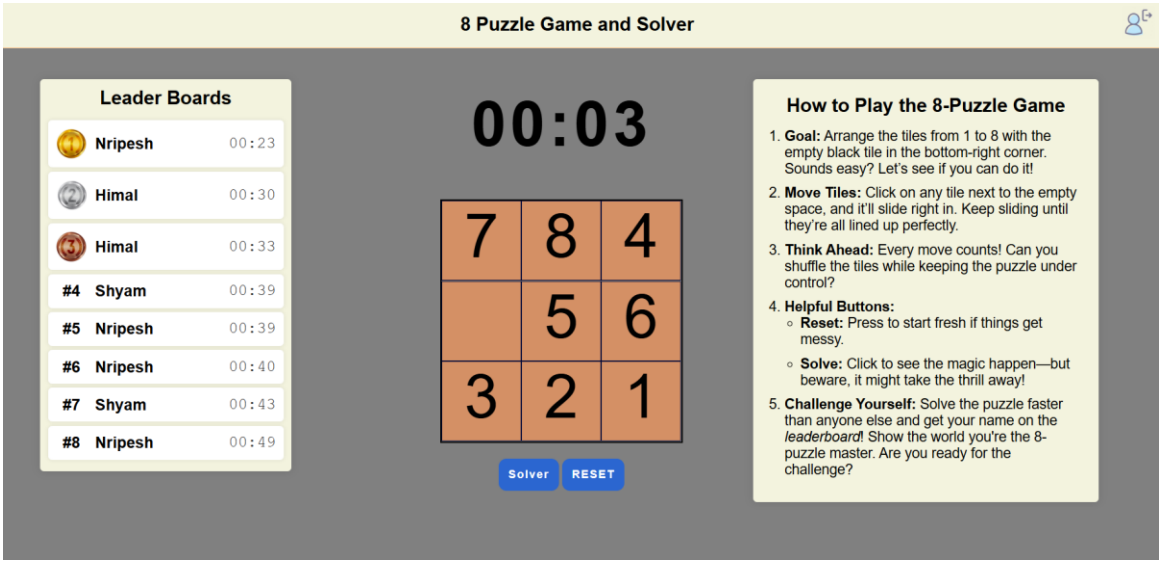


Figure 7: Test Case for Login

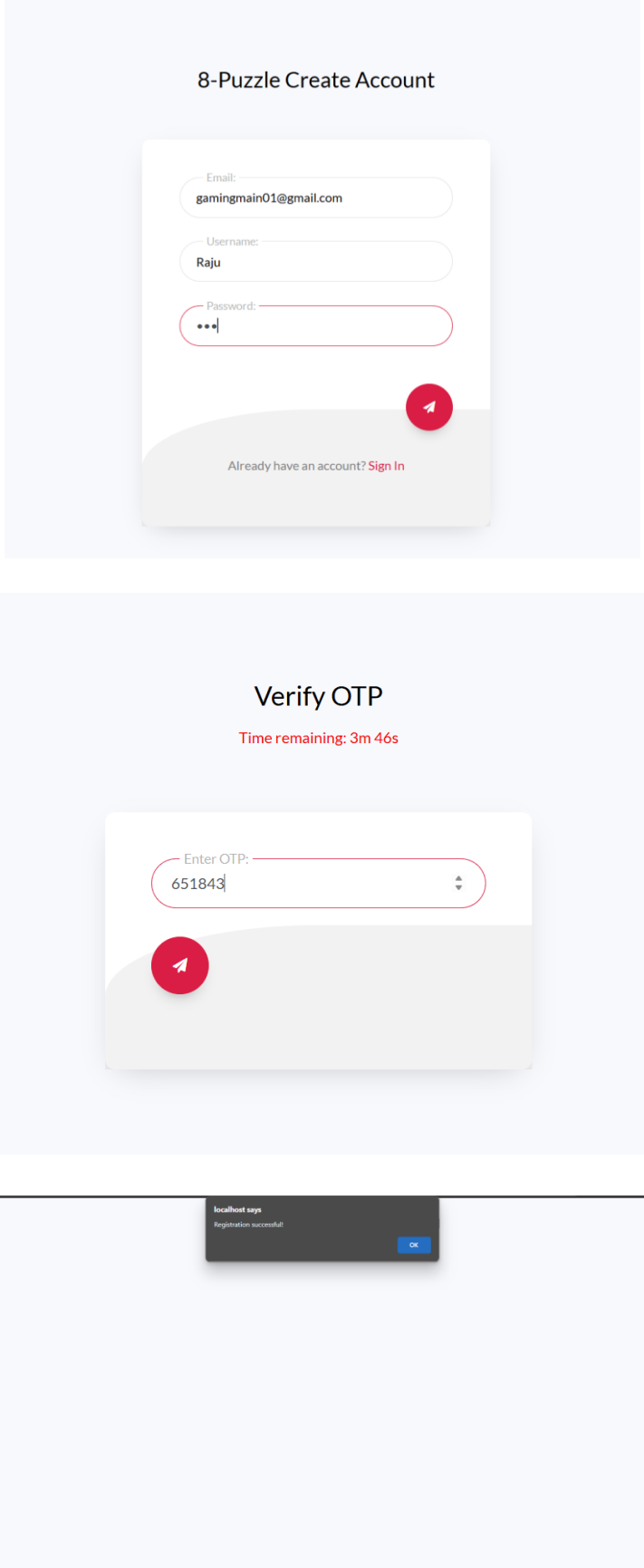


Figure 86: Test Case for Register User

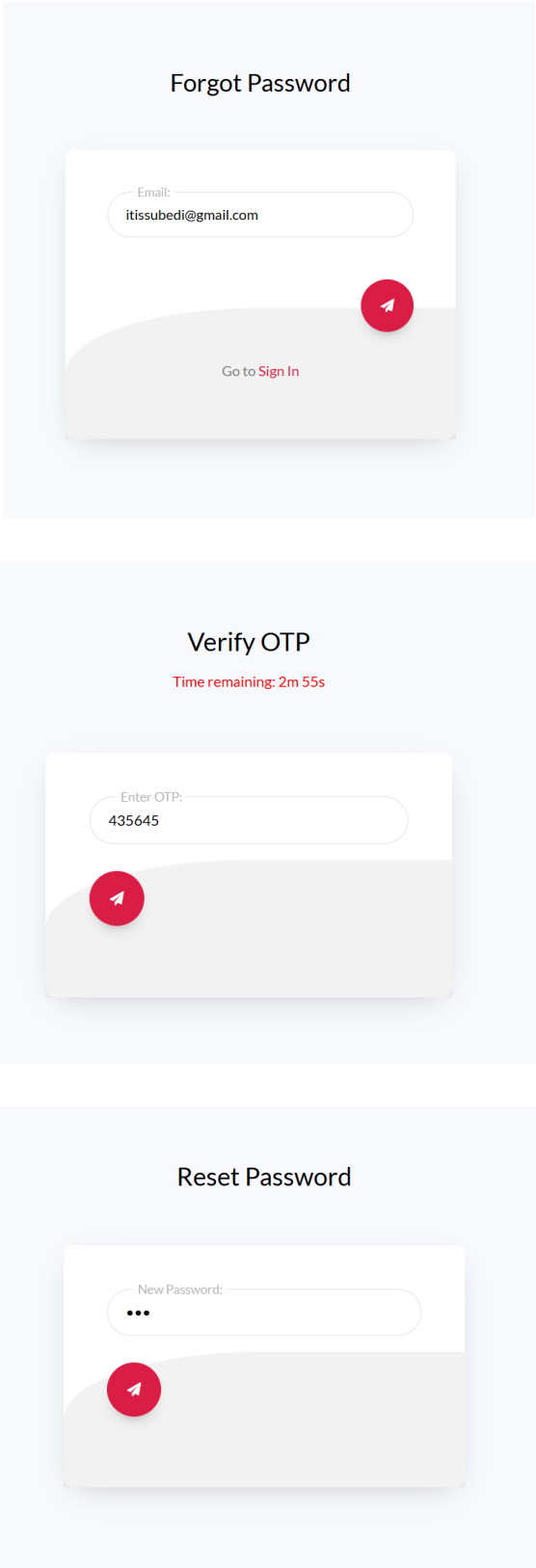


Figure 9: Test Case for Forgot Password

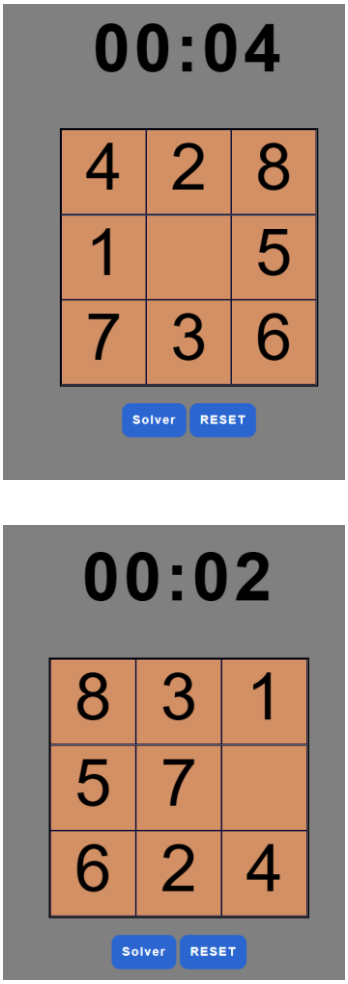


Figure 10: Test Case for Reset Functionality

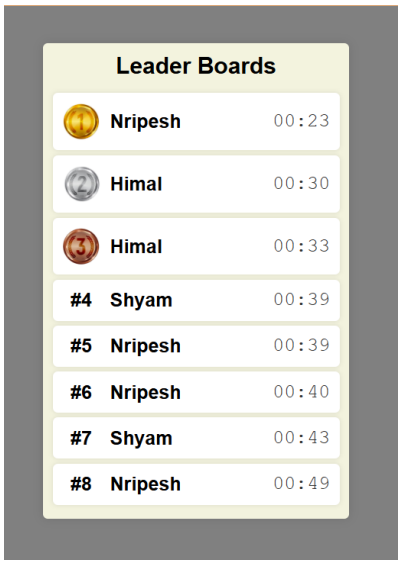


Figure 11: Test Case for Leaderboards

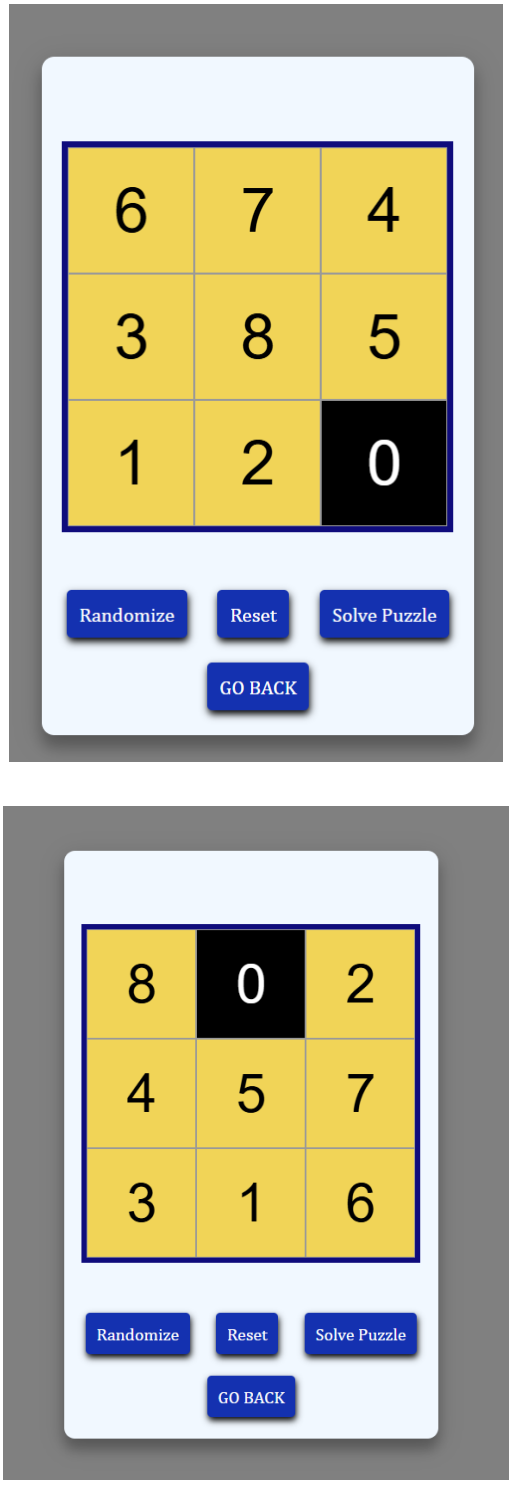


Figure 12: Test Case for Randomize Button

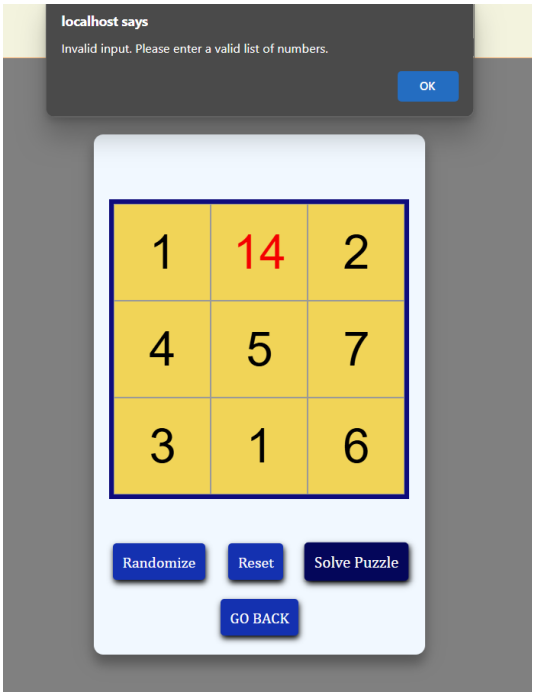


Figure 13: Test Case for Invalid and Duplicate input

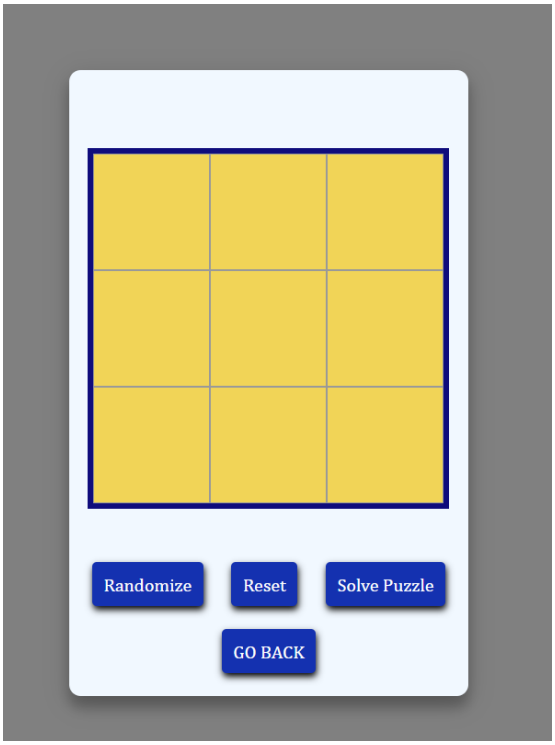
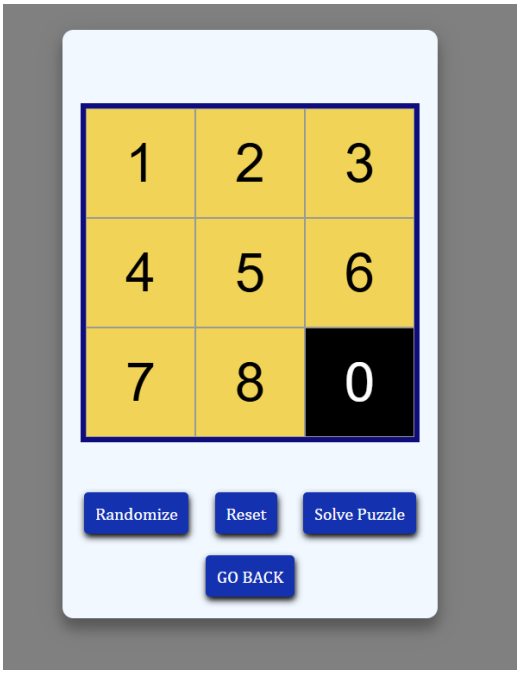
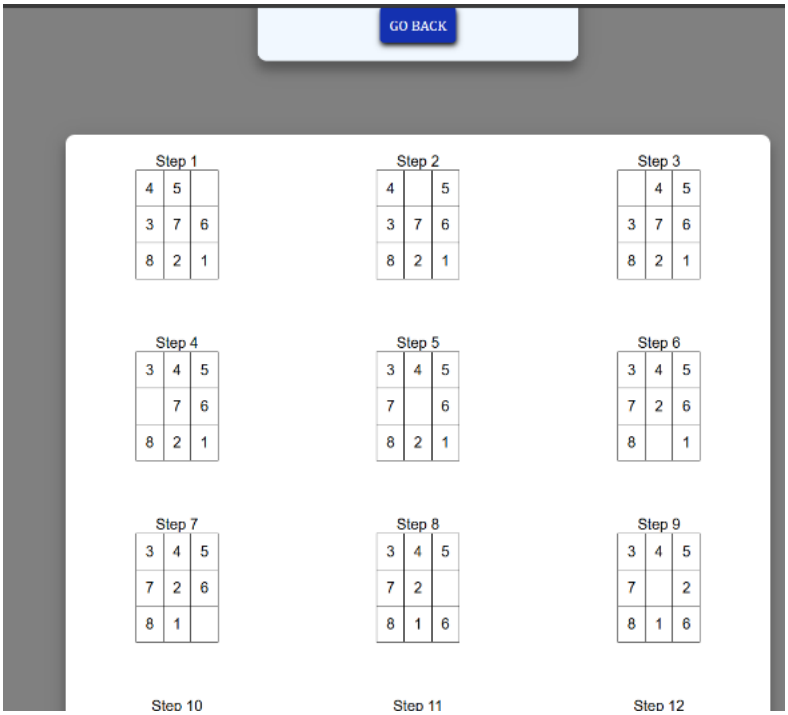


Figure 14: Test Case for Reset Button


1 2 3
4 5 6
7 8 0
Randomize
Reset
Solve Puzzle
GO BACK


GO BACK
Step 1
4 5
3 7 6
8 2 1
Step 2
4 5
3 7 6
8 2 1
Step 3
4 5
3 7 6
8 2 1
Step 4
3 4 5
7 6
8 2 1
Step 5
3 4 5
7 6
8 2 1
Step 6
3 4 5
7 2 6
8 1
Step 7
3 4 5
7 2 6
8 1
Step 8
3 4 5
7 2
8 1 6
Step 9
3 4 5
7 2
8 1 6
Step 10
Step 11
Step 12

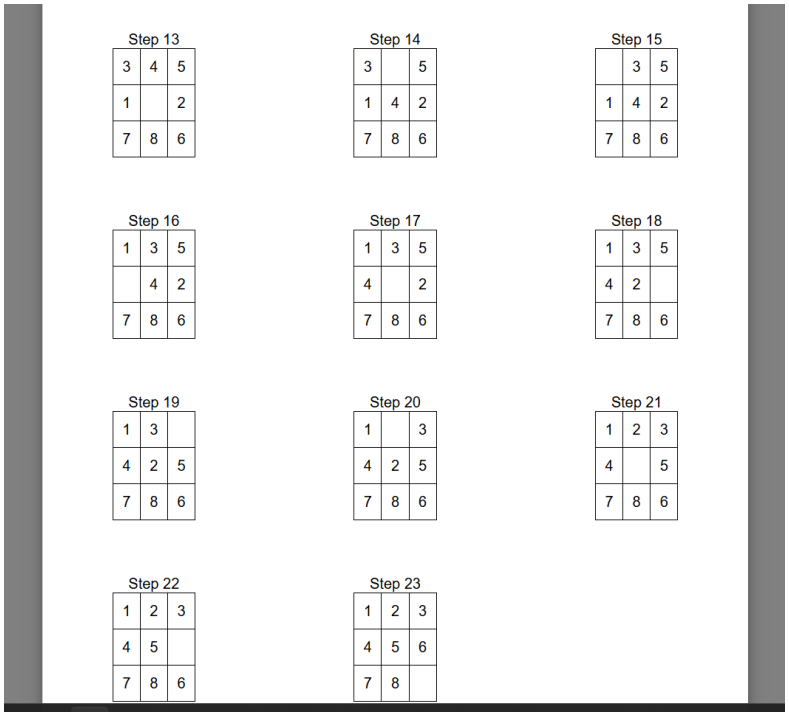
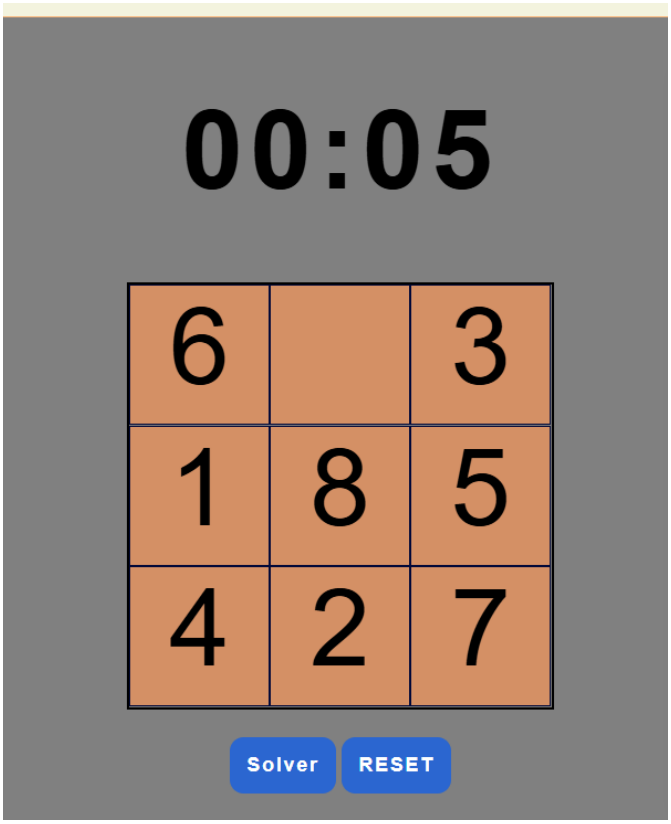


Figure 15: Test Case for Solution and Solution Visualization



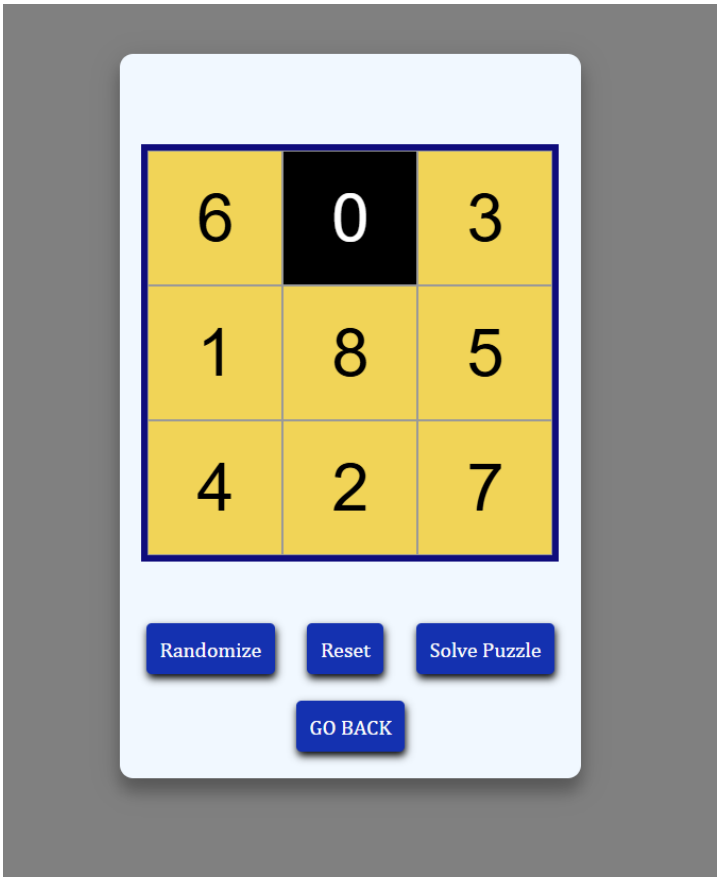


Figure 16: Test Case for Data Transfer from Puzzle to Solver