I A R E

EDUCATION FOR LIBERATION

INSTITUTE OF AERONAUTICAL ENGINEERING

(AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

Complex Problem-Solving Self-Assessment Form

1	Name of the Student	K. KOUSHIK
2	Roll Number	25951A6672
3	Branch and Section	CSE-(AI&ML) - B
4	Program	B. Tech
5	Course Name	Front End Web Development
6	Course Code	ACSE04
7	Please tick (✓) relevant Engineering Competency (ECs) Profiles	

	EC	Profiles	(✓)
	EC 1	Ensures that all aspects of an engineering activity are soundly based on fundamental principles - by diagnosing, and taking appropriate action with data, calculations, results, proposals, processes, practices, and documented information that may be ill-founded, illogical, erroneous, unreliable or unrealistic requirements applicable to the engineering discipline	✓
	EC 2	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
	EC 3	Support sustainable development solutions by ensuring functional requirements, minimize environmental impact and optimize resource utilization throughout the life cycle, while balancing performance and cost effectiveness.	
	EC 4	Competently addresses complex engineering problems which involve uncertainty, ambiguity, imprecise information and wide-ranging or conflicting technical, engineering and other issues.	✓
	EC 5	Conceptualizes alternative engineering approaches and evaluates potential outcomes against appropriate criteria to justify an optimal solution choice.	✓
	EC 6	Identifies, quantifies, mitigates and manages technical, health, environmental, safety, economic and other contextual risks associated to seek achievable sustainable outcomes with engineering application in the designated engineering discipline.	
	EC 7	Involve the coordination of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies) in the timely delivery of outcomes	
	EC 8	Design and develop solution to complex engineering problem considering a very perspective and taking account of stakeholder views with widely varying needs.	✓
	EC 9	Meet all level, legal, regulatory, relevant standards and codes of practice, protect public health and safety in the course of all engineering activities.	

	EC 10	High level problems including many component parts or sub-problems, partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the top consideration.	✓
	EC 11	Undertake CPD activities to maintain and extend competences and enhance the ability to adapt to emerging technologies and the ever-changing nature of work.	✓
	EC 12	Recognize complexity and assess alternatives in light of competing requirements and incomplete knowledge. Require judgement in decision making in the course of all complex engineering activities.	✓
8	Please tick (✓) relevant Course Outcomes (COs) Covered		
	CO	**Course Outcomes**	**(✓)**
	CO 1	Describe language basics like alphabet, strings, grammars, productions, derivations, and Chomsky hierarchy, construct DFA, NFA, and conversion of NFA to DFA, Moore and Mealy machines and interpret differences between them.	✓
	CO 2	Recognize regular expressions, formulate, and build equivalent finite automata for various languages.	✓
	CO 3	Identify closure, and decision properties of the languages and prove the membership.	✓
	CO4	Demonstrate context-free grammars, check the ambiguity of the grammar, and design equivalent PDA to accept the context-free languages.	
	CO 5	Uses mathematical tools and abstract machine models to solve complex problems.	✓
	CO 6	Analyze and distinguish between decidable and undecidable problems.	✓
9	Course ELRV Video Lectures Viewed	**Number of Videos**	**Viewing time in Hours**
		-	-
10	Justify your understanding of WK1	-	
11	Justify your understanding of WK2 – WK9	-	
12	How many WKs from WK2 to WK9 were implanted?	-	
	Mention them	-	

Date: 11-11-2025
K. KOUSHIK
Signature of the Student

COMPLEX ENGINEERING PROBLEM

A COURSE SIDE PROJECT ON

Front End Web Development

K. KOUSHIK

25951A6672

MATH MAZE Based on FEWD

**A Project
Report
submitted in
partial
fulfillment of
the
requirements for the award of the degree of**

**Bachelor of
Technology in
CSE (Artificial Intelligence & Machine Learning)**

By

**K. KOUSHIK
25951A6672**

Department of CSE (Artificial Intelligence & Machine Learning)

INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal, Hyderabad – 500 043, Telangana

November, 2025

DECLARATION

I certify that

The work contained in this report is original and has been done by me under the guidance of my supervisor(s).
The work has not been submitted to any other Institute for any degree or diploma.
I have followed the guidelines provided by the Institute for preparing the report.
I have conformed to the norms and guidelines given in the Code of Conduct of the Institute.
Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

K. Koushik
Place: Hyderabad
Date:

CERTIFICATE

This is to certify that the project report entitled **Math Maze – Interactive Math Puzzles and Logic Challenge System** submitted by **K. Koushik** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfilment of the requirements for the award of the Degree Bachelor of Technology in CSE (Artificial Intelligence & Machine Learning) is a Bonafide record of work carried out under my guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

Supervisor
Head of the Department
Principal
Date:
Place: Hyderabad

APPROVAL SHEET

This project report entitled **Math Maze – Interactive Math Puzzles and Logic Challenge System** submitted by Mr. **K. Koushik** is approved for the award of the Degree Bachelor of Technology in Branch CSE (Artificial Intelligence & Machine Learning).

Examiner
Supervisor(s)
Principal
Date:
Place: Hyderabad

ACKNOWLEDGEMENT

I am extremely grateful and express my profound gratitude and indebtedness to my project guide **Mr. V Vidya Sagar**, Assistant Professor, Department of CSE (Artificial Intelligence & Machine Learning), for her kind help and for giving me the necessary guidance and valuable suggestions for this project work.

I am grateful to **Dr. M. Purushotham Reddy**, Professor and Head of the Department, Department of CSE (Artificial Intelligence & Machine Learning), for extending his support to carry on this project work.

I express my sincere gratitude to **Dr. L. V. Narasimha Prasad**, Professor and Principal, who has been a great source of information for my work.

I thank our college management and respected **Sri M. Rajashekar Reddy**, Chairman, IARE, Dundigal for providing me with the necessary infrastructure to conduct the project work.

ABSTRACT

Math Maze is a web-based interactive platform that presents math puzzles and logic challenges designed to improve problem-solving skills and numerical reasoning. Built using HTML, CSS, and JavaScript, Math Maze offers a responsive interface with multiple puzzle types (arithmetic puzzles, sequence completion, logic grids) and immediate feedback for each attempt. The system focuses on clear UI, quick interaction, and progressive difficulty so users can learn while playing. This project demonstrates how front-end technologies can be used to create educational tools that encourage analytical thinking and engagement through interactive exercises.

Keywords: Math puzzles, logic challenges, interactive learning, front-end development, educational games.

CONTENTS

Name of Contents	**Page No.**
Title Page	01 - 04
Declaration	05
Certificate	06
Approval Sheet	07
Acknowledgement	08
Abstract	09
Contents	10
Chapter 1- Introduction	11 – 12
1.1 Problem Statement	11
1.2 Introduction	11
1.3 Requirements	11
1.4 Prerequisites	12
1.5 Technologies used	12
Chapter 2 - Review of Relevant Literature	13
Chapter 3- Methodology	14 - 16
Chapter 4- Results and Discussions	17
Chapter 5- Conclusions and Future Scope	18
5.1 Conclusion	19
5.2 Future Scope	20
References	21

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Learners and students often lack engaging, bite-sized resources to practice mathematical thinking and logic. The key issues include:

- Limited interactive practice tools that provide instant feedback
- Difficulty finding progressively challenging puzzles in one place
- Lack of accessible, web-based tools suitable across devices
- Manual generation of puzzles requiring extra effort from educators

1.2 Introduction

MathMaze provides an interactive environment for solving math puzzles and logic challenges. The system presents puzzles, checks user responses, and gives immediate feedback. Difficulty levels can be adjusted to suit learners of different ages. Implemented with HTML, CSS, and JavaScript, MathMaze shows how simple front-end tools can deliver interactive educational content and encourage repeated practice.

1.3 Requirements

The project requires:

- Understanding of basic puzzle design and logic formulation
- Knowledge of HTML for structuring puzzles and input forms
- CSS for styling and responsive display
- JavaScript for puzzle logic, validation, and feedback

1.4 Prerequisites

To develop MathMaze, one should be familiar with:

- DOM manipulation and event handling in JavaScript
- Basic algorithmic thinking to generate and check puzzles
- Simple state management (tracking score, attempts, and levels)

1.5 Technologies Used

- **HTML** — puzzle pages and input widgets
- **CSS** — styling and responsive behavior
- **JavaScript** — puzzle generation, user interaction, and scoring logic

CHAPTER 2

REVIEW OF RELEVANT LITERATURE

Interactive educational tools and gamified learning have been widely studied for their effectiveness in improving retention and motivation. Prior work highlights advantages of immediate feedback, adaptive difficulty, and short practice sessions for learning mathematics. Web-based puzzle platforms and logic trainers commonly use front-end scripting to deliver interactive exercises without complex back ends. Research also supports that varied puzzle formats (numerical, spatial, logical) strengthen different cognitive skills, making platforms like MathMaze pedagogically valuable.

CHAPTER 3

METHODOLOGY

The methodology for MathMaze focuses on puzzle design, interface structure, and client-side logic to validate answers and update scores. The development process includes: puzzle specification, UI layout, JavaScript-based puzzle engine, and testing across screen sizes.

3.1 Puzzle Interface Structure

Design the layout for displaying a puzzle, input controls, feedback area, and navigation (next/previous).

Code 3.1 — HTML: Puzzle Layout (Insert here in Chapter 3 under Interface Design)

```
<!-- Code 3.1: MathMaze HTML -->
<div class="mathmaze">
  <h2>MathMaze</h2>

  <!-- Puzzle Display -->
  <div id="puzzleArea">
    <p id="puzzleText">Solve: 8 + 7 = ?</p>
    <input type="number" id="answerInput" placeholder="Your
answer">
    <button id="submitAnswer">Submit</button>
    <p id="feedback"></p>
  </div>

  <!-- Controls -->
  <div class="controls">
    <button id="prevPuzzle">Previous</button>
    <button id="nextPuzzle">Next</button>
    <p>Score: <span id="score">0</span></p>
  </div>
</div>
```

3.2 Styling & Responsive Design

Apply CSS for readability and responsive layout suitable for desktop and mobile.

Code 3.2 — CSS: Basic Styling (Insert under Chapter 3 → Technologies Used / Styling subsection)

```
/* Code 3.2: MathMaze CSS */
.mathmaze {
  max-width: 480px;
  margin: 18px auto;
  padding: 16px;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background: #ffffff;
  border-radius: 10px;
  box-shadow: 0 8px 20px rgba(0,0,0,0.08);
}

#puzzleText {
  font-size: 18px;
  margin-bottom: 8px;
}

input[type="number"] {
  width: 60%;
  padding: 6px;
  margin-right: 8px;
}
button {
  padding: 8px 10px;
  border-radius: 6px;
  border: none;
  background: #0ea5a4;
  color: white;
}
.controls {
  margin-top: 12px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

3.3 Puzzle Engine & Interaction Logic

JavaScript handles a small set of puzzles, validates answers, updates score, and navigates between puzzles.

Code 3.3 — JavaScript: Puzzle Logic (Insert under Chapter 3 → Dynamic Functionality)

```
// Code 3.3: MathMaze JavaScript
const puzzles = [
  { q: '8 + 7 = ?', ans: 15 },
  { q: 'Fill the sequence: 2, 4, 8, 16, ?', ans: 32 },
  { q: 'What is 5 × 6?', ans: 30 },
];

let current = 0;
let score = 0;

function showPuzzle(i) {
  document.getElementById('puzzleText').textContent = 'Solve:
' + puzzles[i].q;
  document.getElementById('feedback').textContent = '';
  document.getElementById('answerInput').value = '';
}

document.getElementById('submitAnswer').addEventListener('clic
k', function () {
  const val =
Number(document.getElementById('answerInput').value);
  if (isNaN(val)) return;
  if (val === puzzles[current].ans) {
    score += 10;
    document.getElementById('feedback').textContent =
'Correct!';
  } else {
    document.getElementById('feedback').textContent = 'Try
again.';
  }
  document.getElementById('score').textContent = score;
});

document.getElementById('nextPuzzle').addEventListener('click'
, function () {
  if (current < puzzles.length - 1) current++;
  showPuzzle(current);
});
```

```
document.getElementById('prevPuzzle').addEventListener('click'
, function () {
  if (current > 0) current--;
  showPuzzle(current);
});

// initial display
showPuzzle(current);
```

3.4 Testing & Validation

- Manual testing of correct and incorrect answers.
- Test navigation and score increment behaviour.
- Check responsiveness on phones and tablets.
- Edge cases: empty input, non-numeric input handled by ignoring submission.

Where to include code in the report:

- Insert **Code 3.1** after the Interface Design paragraph in **Chapter 3**.
- Insert **Code 3.2** in the Technologies/Styling subsection (Chapter 3).
- Insert **Code 3.3** in the Dynamic Functionality subsection (Chapter 3).
- Optionally paste all code blocks again in **Appendix – Full Source Code** (after References).

CHAPTER 4

RESULTS AND DISCUSSION

Math Maze produced immediate feedback for user answers and tracked scores reliably during testing. Navigation between puzzles worked smoothly; the scoring system rewarded correct answers. The responsive layout-maintained readability on small screens. User testing with sample users indicated that short, varied puzzles increased engagement and provided quick reinforcement of concepts. Limitations include simple puzzle set and lack of persistent user data, which can be resolved in future enhancements.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The Math Maze project demonstrates that interactive math puzzles and logic challenges can be implemented effectively using only front-end technologies. The platform supports rapid practice, instant feedback, and simple scoring, making it a useful learning aid for students.

5.2 Future Scope

- Expand puzzle library and add adaptive difficulty.
- Add user accounts and progress tracking via a backend (Firebase).
- Include hints, timer-based challenges, and leaderboards.
- Add analytics to track common mistakes and adapt puzzles.

REFERENCES

1. S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*.
2. Research on gamification in education — various journal articles.
3. MDN Web Docs — JavaScript DOM APIs.
4. UX guidelines for educational tools.
5. Relevant K–12 math pedagogy papers.