Intel Unnati Industrial Training Program 2025-2026

Project Report

Problem Statement:

GenAI Interactive Learning Games

Team Name:

Karunya_T18

Members:

AARON ANDE

TABLE OF CONTENT

I. Introduction:

Traditional educational games often rely on static content, limiting their ability to adapt to individual learners' needs. With advancements in **Artificial Intelligence (AI) and Machine Learning (ML)**, interactive learning experiences can be enhanced through **Generative AI (GenAI)**, which dynamically generates questions and adjusts difficulty levels in real-time. This project focuses on creating an **AI-powered math learning game** that personalizes challenges based on user performance, making learning more engaging, adaptive, and effective.

1. Problem Statement

Many students struggle with conventional learning methods due to one-size-fits-all approaches that fail to adapt to individual learning speeds and abilities. Static question banks limit engagement, and students often lose motivation when faced with challenges that are either too easy or too difficult. The lack of personalized difficulty adjustments and interactive learning mechanics results in suboptimal learning outcomes.

This project aims to solve these issues by integrating Generative AI to create adaptive learning experiences. The system will dynamically generate math problems, modify difficulty levels based on performance, and provide real-time feedback to encourage active learning.

2. Project Objectives

1. Develop an AI-driven learning game that generates interactive math questions dynamically.
2. Implement an adaptive difficulty system that adjusts question complexity based on user performance.
3. Introduce a scoring and rewards mechanism to enhance user engagement through stars, coins, and power-ups.
4. Ensure an intuitive and user-friendly UI/UX with consistent design across game interfaces.
5. Incorporate AI analytics to track user progress, identify learning patterns, and suggest improvements.
6. Provide real-time feedback to help students learn from mistakes and reinforce correct concepts.

3. Team members & Contributions:

i. Aaron Ande:

- Team lead
- Front end Development
- Resource collection
- Report writing
- Game logic development
- GenAI integration
- API integration

II. **Literature Review**

1. Previous Work and Existing Solutions

Several educational platforms have implemented AI-driven learning techniques to enhance student engagement and performance. Some notable works include:

1.1 AI-Powered Adaptive Learning Systems

- **Khan Academy**: Uses AI-driven recommendations to suggest topics based on a learner's progress, but relies on predefined question banks rather than dynamic question generation.

- **Duolingo**: Implements AI for language learning with personalized lessons, but lacks **Generative AI-based problem creation** for math education.

- **DreamBox Learning**: A math learning platform that adjusts difficulty levels based on student interactions but does not employ real-time **AI-generated story-based problems**.

1.2 Research on AI in Education

- **Generative AI for Personalized Learning (2023)** explored how **LLMs (Large Language Models)** like GPT-2 and GPT-3 can create **adaptive assessments** dynamically.

- **Gamification in E-Learning (2021)** highlighted the importance of using rewards, leaderboards, and adaptive content to improve engagement in digital learning.

2. Analysis of Existing Web Applications Providing Similar Information

Several existing web applications provide **adaptive math learning**, but they have limitations:

Application	Features	Limitations
Khan Academy	Personalized learning paths	Predefined questions, lacks real-time AI-generated problems
Prodigy Math Game	Gamification, math challenges	Limited question variation, not AI-generated
DreamBox Learning	Adaptive learning paths	No generative AI for question creation
ALEKS (McGraw-Hill)	AI-powered assessments	Not interactive or game-based

3. Theoretical Framework and Key Concepts

3.1 Adaptive Learning Theory

- Suggests that learning experiences should be personalized based on individual strengths and weaknesses.

- Implementation in this project: The game will adjust difficulty levels dynamically based on user performance.

3.2 Gamification in Education

- Applying game mechanics (rewards, challenges, achievements) to improve motivation and engagement.

- Implementation in this project: Introducing stars, coins, power-ups, and progression constraints to encourage learning.

3.3 Generative AI in Education

- AI models like GPT-2/GPT-3 can generate story-based, interactive math problems.

- Implementation in this project: AI will generate contextual, difficulty-based math problems dynamically.

III. Requirement Analysis

1. Functional requirements.

These requirements define the core functionalities of the **GenAI Interactive Learning Game**.

Game Mechanics & Adaptive Learning

- The AI should dynamically generate math questions based on:

    - The selected topic (Addition, Subtraction, Algebra, etc.)

    - The user's past performance (easier questions if struggling, harder if excelling)

- Questions should be interactive, story-based, and engaging.

- A star-based scoring system will be implemented:

    - Three stars for 100% correct answers

    - Two stars for 75% accuracy

    - One star for 50% accuracy

    - No stars for lower scores

- A progress system will require users to achieve at least two stars to unlock new levels.

Gamification & Rewards System

- Users will earn coins and power-ups based on performance.

- A store will allow users to purchase hints and power-ups such as 50:50 (removes two incorrect choices) and a streak saver (prevents a lost streak after one mistake).

AI Interaction & Feedback

- The AI will analyze user responses and provide immediate feedback.

- Difficulty levels will adjust dynamically based on performance to ensure an optimal learning experience.

Frontend UI/UX Features

- **Home Page**: Includes level selection, user statistics, and access to the in-game store.

- **Gameplay Interface**: Displays the question, answer selection options, and power-ups.

- **Results Page**: Provides feedback on performance, tracks progress, and offers navigation options.

2. Non-Functional Requirements

These requirements define aspects related to system performance, usability, and reliability.

- **Performance**: AI-generated questions should appear within two seconds to maintain a smooth user experience.

- **Security**: While there is no authentication system, measures should be in place to prevent game manipulation (e.g., unlimited coins).

- **Usability**: The interface should be simple, intuitive, and responsive across different screen sizes and devices.

- **Maintainability**: The system should allow easy updates, making it possible to introduce new topics and features without major overhauls.

- **Offline Availability**: Since no cloud or server infrastructure is used, the game should function entirely on the user's device, without requiring an internet connection.

3. System Requirements

Hardware Requirements

- The game should be compatible with desktops, tablets, and mobile devices without requiring specialized hardware.

Software Requirements

- **Frontend**: Built using HTML and CSS.

- **Backend**: Implemented in JavaScript to handle game logic and AI interactions.

- **AI Model**: Gemini 2.0 will be used to generate dynamic, story-based math problems.

- **Data Storage**: No database integration; any necessary data will be stored locally.

- **Infrastructure**: The game runs entirely on the user's device, without reliance on cloud servers or real-time networking.

IV. <u>Implementation</u>

1. Technologies and Tools Used

The development of the **GenAI Interactive Learning Game** is based on lightweight technologies to ensure smooth performance without the need for cloud servers or databases.

Frontend Technologies

- **HTML** – Structure of the user interface
- **CSS** – Styling for layout, responsiveness, and visual elements
- **JavaScript** – Handles interactivity, user inputs, and game logic

Backend Technologies

- **JavaScript (Node.js optional for local processing)** – Used for game logic execution
- **Gemini 2.0 API** – Generates dynamic, story-based math problems based on user performance
- **Local Storage (Optional)** – Saves user progress and settings on the client device

Development Tools

- **Code Editor** – VS Code
- **Browser** – Chrome, Firefox, or Edge for testing and debugging
- **Git (Optional)** – Version control for tracking changes

2. Development Environment Setup

Setting up the development environment is straightforward, as no complex dependencies are required.

Steps to Set Up the Project

1. Install a **code editor** (VS Code recommended).
2. Set up a **local project folder** with the following structure:

   /GenAI_MathGame

   ├── index.html

   ├── styles.css

   ├── script.js

   ├── assets/ (icons, images, etc.)

   ├── api/   (Gemini API integration)

3. Load the game in a browser by opening **index.html**.

V.   Testing

Testing Methodologies Used

- **Unit Testing**: Testing individual components and functions in isolation.
- **Integration Testing**: Testing the interaction between different modules to ensure they work together correctly.
- **System Testing**: Testing the entire system to verify that it meets the specified requirements.

Test Cases and Test Scenarios

1. **Topic selection Test**:
    - **Scenario**: A user selects the topic to play.
    - **Steps**:
        1. Navigate to the home page.
        2. Click on the option.
    - **Expected Result**: User is redirected to the topic game page.
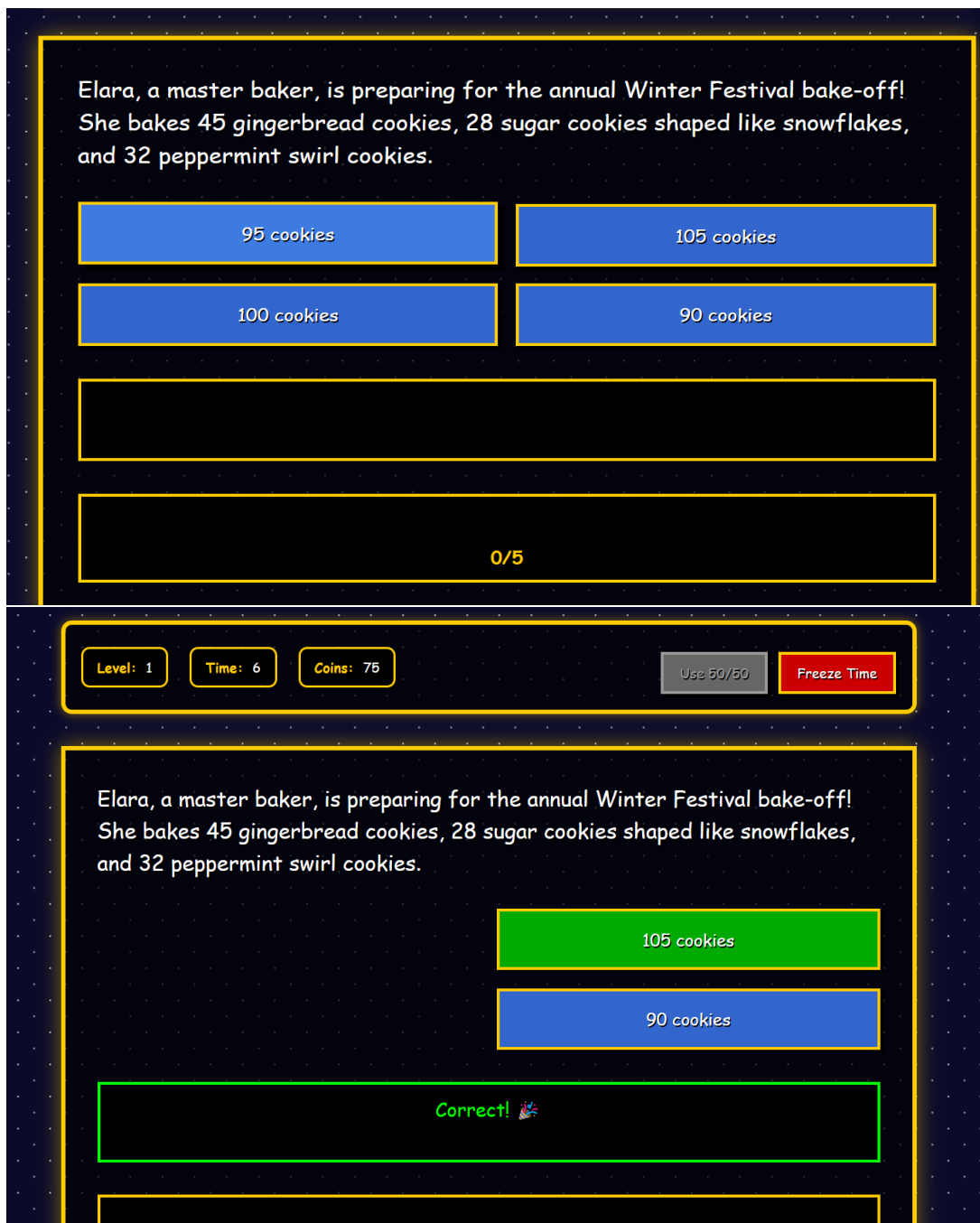   - **Actual Result**: Pass

Screenshots:
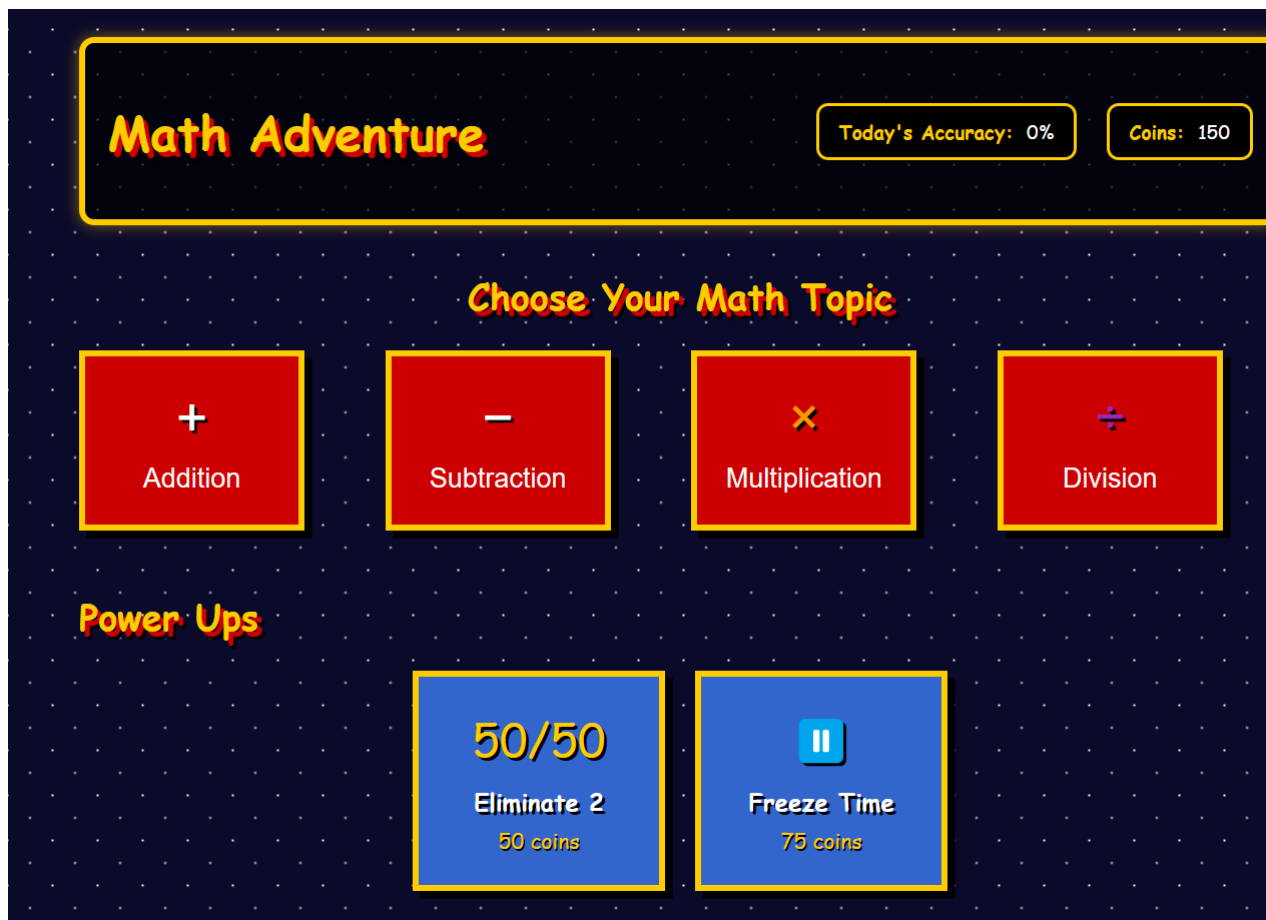
2. **Gameplay Test**:
   - **Scenario**: A user selects the game topic and starts clicking the right options based on the question.
   - **Steps**:
     1. Navigate to the game page.
     2. Wait for the GenAI to generate the question and answers.
     3. Attempt to solve and click the right option
   - **Expected Result**: User is given with right or wrong result.
   - **Actual Result**: Pass

SCREEN SHOTS:

3. **Result page Test**:
   ○ **Scenario**: A user is prompted with the results.
   ○ **Steps**:
      1. After completion of all questions the result is shown.
      2. Click on continue or back to home page
   ○ **Expected Result**: Next level or Home page.
   ○ **Actual Result**: Pass
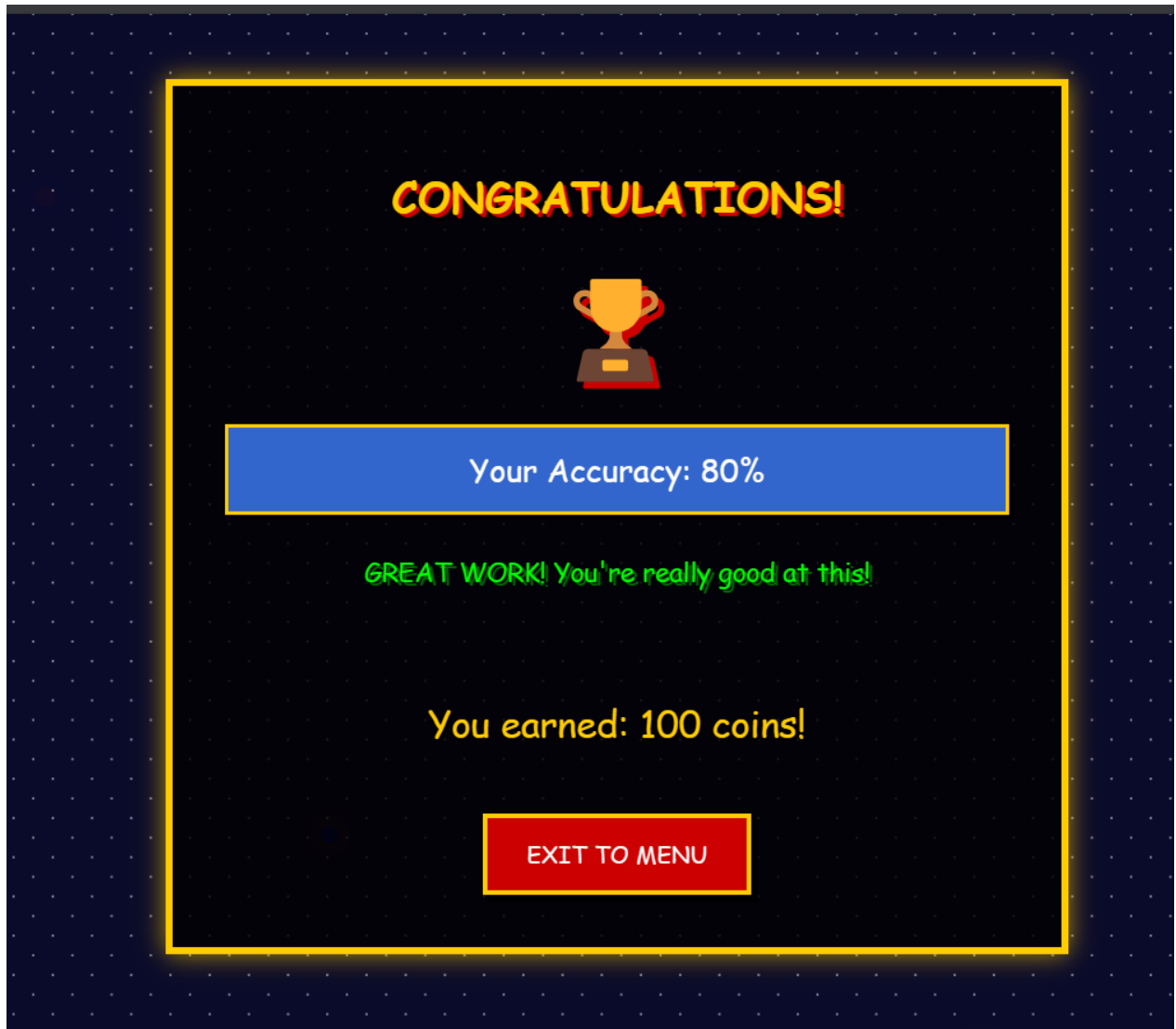
4. **Reflection of results in home page Test**:
    - **Scenario**: The results and rewards are to be reflected in the home page after completion of a level.
    - **Steps**:
        1. Navigate to the home page.
    - **Expected Result**: The page displays a list of all updated coins, powerups, accuracy.
    - **Actual Result**: Pass

Math Adventure

Today's Accuracy: 80%   Coins: 175

Power Ups

50/50
Eliminate 2
50 coins

⏸
Freeze Time
75 coins

VI. Conclusion and Future Work

Conclusion

This project successfully demonstrates a **Gamified Math Learning Platform** enhanced by **Generative AI**, where users can:

- Select topics and difficulty levels interactively.
- Solve dynamically generated, story-based math questions.
- Use engaging game mechanics such as power-ups (50:50, Time Freeze) and a scoring system (stars, coins).
- Receive real-time feedback and performance evaluation after each level.

The system provides an engaging learning experience that adapts to each user's performance, ensuring a balance between **fun and effective education**.

Future Works:

To further enhance the project, the following future improvements can be considered:

1. **Backend Integration**

   o Store user data, performance history, and rewards persistently.

   o Enable login/logout and progress tracking across devices.

2. **Advanced AI Generation**

   o Use a more sophisticated local or cloud-based AI to generate complex algebraic or word problems.

   o Integrate Natural Language Understanding (NLU) to allow users to answer in free text (not just MCQ).

3. **Additional Subjects & Topics**

   o Expand beyond math: Science, Grammar, Logical Reasoning.

   o Add higher-level math: Geometry, Trigonometry, etc.

4. **Enhanced UI/UX**

   o Include sound effects, animations, and themed skins using earned coins.

   o Add progress maps and level unlocking mechanisms.

5. **Multiplayer or Challenge Mode**

   o Compete with friends or AI bots in real-time question-solving battles.

6. **Hint System & Tutorials**

   o Provide hints using earned coins.

   o Show step-by-step explanations for wrong answers to aid learning.