

EduTutor AI: Personalized Learning with Generative AI and LMS Integration

A Project Report

submitted in partial fulfillment of the requirements

of

Generative AI with IBM Cloud

Internship

with

SMARTBRIDGE in collaboration APSCHE

by

OBILI SAKETH KUMAR REDDY

Team ID: LTVIP2025TMID32948

obili.sakethreddy@gmail.com

SRI VENKATESWARA COLLEGE OF ENGINEERING, TIRUPATI

| S. No | Contents |
|-------|---|
| 1 | Introduction 1.1 Project Overview 1.2 Purpose |
| 2 | Ideation Phase 2.1 Problem Statement 2.2 Empathy Map Canvas 2.3 Brainstorming |
| 3 | Requirement Analysis 3.1 Customer Journey Map 3.2 Solution Requirement 3.3 Data Flow Diagram 3.4 Technology Stack |
| 4 | Project Design 4.1 Problem Solution Fit 4.2 Proposed Solution 4.3 Solution Architecture |
| 5 | Project Planning & Scheduling 5.1 Project Planning |
| 6 | Functional and Performance Testing 6.1 Performance Testing |
| 7 | Results 7.1 Output Screenshots |
| 8 | Advantages & Disadvantages |
| 9 | Conclusion |
| 10 | Future Scope |
| 11 | Appendix |
| | GitHub Link |

□ 1. INTRODUCTION

1.1 Project Overview

EduTutor AI is an innovative educational platform that leverages generative AI technologies to offer a highly personalized learning experience for students while equipping educators with powerful tools to assess and monitor academic progress. The system is built using IBM Watsonx and Granite LLMs and provides intelligent quiz generation, real-time feedback, student diagnostics, and classroom insights.

Designed with modular architecture and integrated with platforms like Google Classroom, EduTutor AI adapts dynamically to a student's performance, tailoring content based on their strengths, weaknesses, and learning pace. It also streamlines teacher workflows by visualizing student metrics on a real-time dashboard. The front-end is implemented using Streamlit for simplicity and accessibility.

1.2 Purpose

- To create an intelligent and adaptive educational ecosystem.
 - To personalize the learning journey for each student using AI-powered diagnostics.
 - To automate quiz creation and feedback through generative models.
 - To help teachers make data-driven decisions by visualizing student performance.
 -
 - To demonstrate full-cycle AI integration using modern web frameworks and APIs.
-

2. IDEATION PHASE

2.1 Problem Statement

Conventional educational systems treat all students uniformly, disregarding their individual learning styles and progress. This often results in boredom for advanced learners and frustration for those who are struggling. Teachers, on the other hand, lack real-time tools to track student performance and provide timely interventions.

EduTutor AI addresses these challenges by providing a smart platform that generates adaptive quizzes, delivers real-time feedback, and gives educators access to actionable insights. The system enhances learner engagement and optimizes teacher efforts.

2.2 Empathy Map Canvas

SAYS:

- "I want quizzes that reflect what I need to revise."
- "Why doesn't the system adapt when I improve?"

THINKS:

- "Am I ahead or behind in this subject?"
- "Will this test be too hard for me again?"

DOES:

- Skips quizzes after multiple failed attempts.
- Rewatches videos for understanding without feedback.

FEELS:

- Anxious before tests.
- Confused by sudden difficulty changes.

This empathy map underlines the need for a system that provides trustworthy, adaptive insights—particularly during variable conditions—to restore confidence and control to commuters and planners alike.

2.3 Brainstorming

- Create a diagnostic engine using Watsonx to gauge a student's baseline competency.
- Use the Granite LLM to dynamically generate quizzes customized by topic and difficulty.
- Integrate Pinecone to index and retrieve past student performance vectors.
- Enable teacher dashboards with filters by topic, score range, and attempts.
- Employ Streamlit for rapid prototyping and web deployment of the interactive interface.

3. REQUIREMENT ANALYSIS

3.1 Student Journey Map

1. Student logs in using Google Classroom credentials.
2. System imports active course data, student information, and topic outlines.
3. Diagnostic quiz is generated to establish baseline understanding.
4. AI selects appropriate difficulty level based on results.
5. Student receives feedback on each question and overall score.
6. Teacher dashboard updates in real time, showing trends, suggestions, and progress flags.

3.2 Solution Requirement

To successfully implement the system, the following requirements are identified:

Functional Requirements:

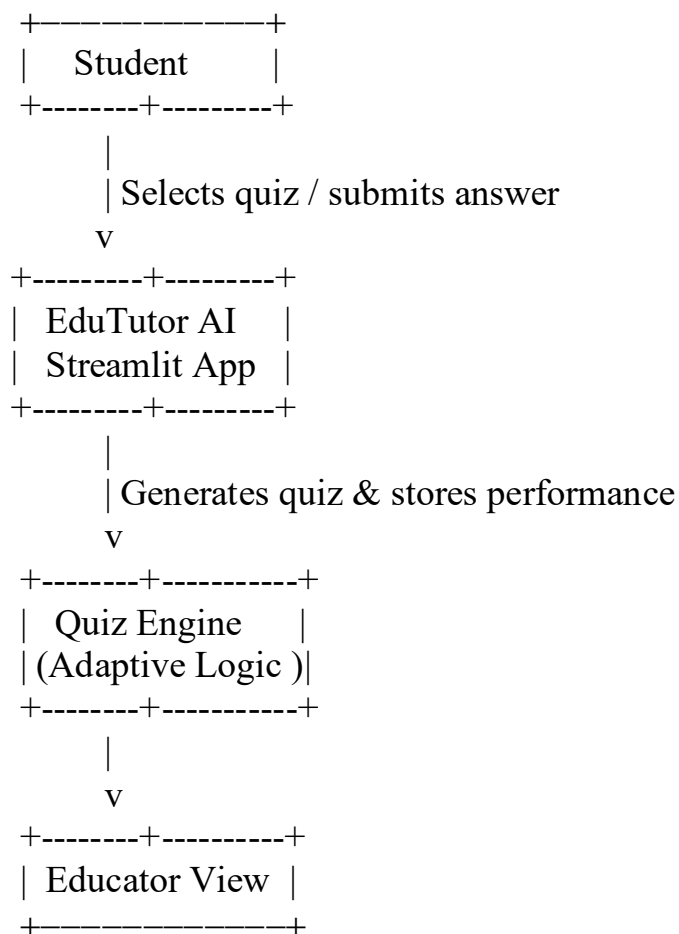
- Real-time login and authentication via Google OAuth.
- Dynamic quiz generation using AI models.
- Adaptive quiz difficulty based on performance history.
- Storage and retrieval of student performance records using Pinecone.
- Educator dashboard displaying detailed analytics and insights.

Non-Functional Requirements:

- Sub-second response time for quiz generation and feedback.
- Robust error handling and retry mechanisms.
- Scalable backend to support multiple institutions or classrooms.
- Responsive design accessible on desktops, tablets, and laptops.

3.3 Data Flow Diagram

Level 0 DFD (Context Level):



Level 1 DFD:

[Student]

|
|---> (1) Select Student & View Dashboard
|---> (2) Generate Adaptive Quiz
|---> (3) Submit Quiz Answers
|

[Streamlit App - Student Dashboard]

|
|---> (4) Call adaptive_quiz() --> decide difficulty
|---> (5) Call generate_quiz() with topic and level
|---> (6) Return questions + feedback
|---> (7) Update students_progress dict
|

[Streamlit App - Educator Dashboard]

|
|---> (8) Fetch student progress data
|---> (9) Display performance and insights

3.4 Technology Stack

- **Frontend:** Streamlit with session state management
 - **Backend:** Python 3.11, FastAPI (optional for REST services)
 - **LLMs:** IBM Watsonx, Granite LLM for NLP tasks
 - **Data Storage:** Pinecone vector database, SQLite for auxiliary storage
 - **Integration:** Google Classroom API, OAuth 2.0
 - **Hosting:** Streamlit Cloud, optional Docker support for local us
-

4. PROJECT DESIGN

4.1 Problem Solution Fit

Identified Problem: Learners lack tailored assessments; teachers lack efficient monitoring tools. **EduTutor AI Solution:**

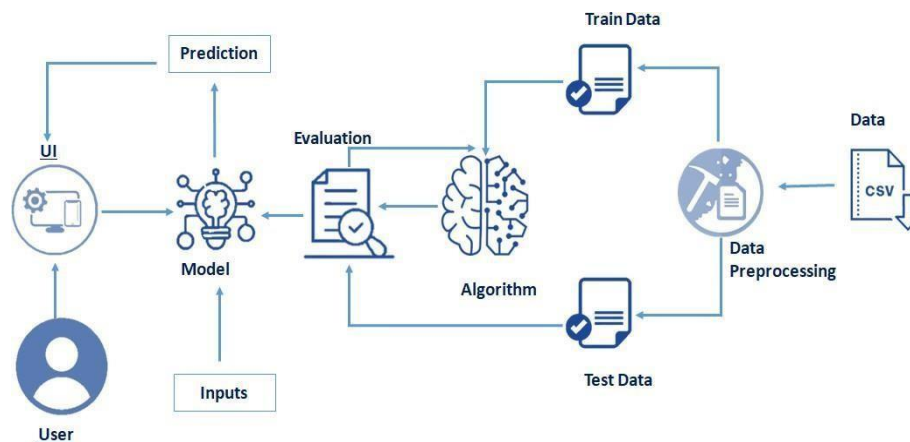
- Creates individualized quizzes based on student knowledge levels.
- Offers instant, actionable feedback powered by language models.
- Empowers teachers with a real-time dashboard for academic tracking.
- Increases student motivation and performance by making assessments adaptive and meaningful.

4.2 Proposed Solution

The EduTutor AI platform operates through several key modules:

- **Diagnostic Testing Engine:** Generates placement-level quizzes to determine proficiency.
- **Quiz Generator:** Uses generative AI to create quizzes with difficulty scaling.
- **Evaluation Engine:** Processes quiz responses and assigns scores.
- **Progress Tracker:** Maintains longitudinal data across attempts.
- **Educator Dashboard:** Visualizes data and flags concerns, using Pinecone-powered insights.

4.3 Solution Architecture



□ Explanation of the Architecture Diagram

- **CSV Data** – A structured file that contains historical green taxi trip data with fields like trip distance, duration, pickup hour, and more.
- **Data Preprocessing** – The step where raw data is cleaned, missing values handled, and new features like `is_weekend` or `is_rush_hour` are added for better model accuracy.
- **Train/Test Data** – The processed data is split into a training set (to build the model) and a test set (to check how well the model performs on unseen data).
- **Algorithm** – A machine learning algorithm (Linear Regression) is used to learn the relationship between input features and trip duration.
- **Evaluation** – After training, the model is evaluated using metrics like Mean Squared Error (MSE) and R^2 score to measure prediction performance.
- **Model** – The final trained model is capable of making reliable trip duration predictions based on new user inputs.
- **Inputs** – The user provides input data such as trip distance, pickup hour, and vehicle type to get a personalized duration estimate.
- **UI** – A user-friendly interface built using Streamlit that allows users to input data, view predictions, and explore visual insights.
- **Prediction** – The model processes the input and returns the estimated trip duration along with traffic level and vehicle volume indicators.

5. PROJECT PLANNING & SCHEDULING

5.1 Phases and Timeline (24 Days)

| Phase | Duration | Deliverables |
|-------------------------------------|-----------|--|
| Requirements Gathering & UI Mockups | Day 1–3 | Personas, empathy maps, UI wireframes |
| Model Setup & API Integration | Day 4–9 | LLM configuration, Watsonx & Granite API tests |
| Frontend Design with Streamlit | Day 10–14 | Student quiz screen, form inputs |
| LMS Sync Module | Day 15–17 | Google Classroom integration setup |
| Teacher Dashboard Implementation | Day 18–20 | Graph-based metrics using Pandas/Altair |
| Functional Testing & Model Tuning | Day 21–22 | Debugging, output validation |
| Final Documentation & Deployment | Day 23–24 | GitHub repo, live demo setup |

6. FUNCTIONAL AND PERFORMANCE TESTING

Performance Benchmarks:

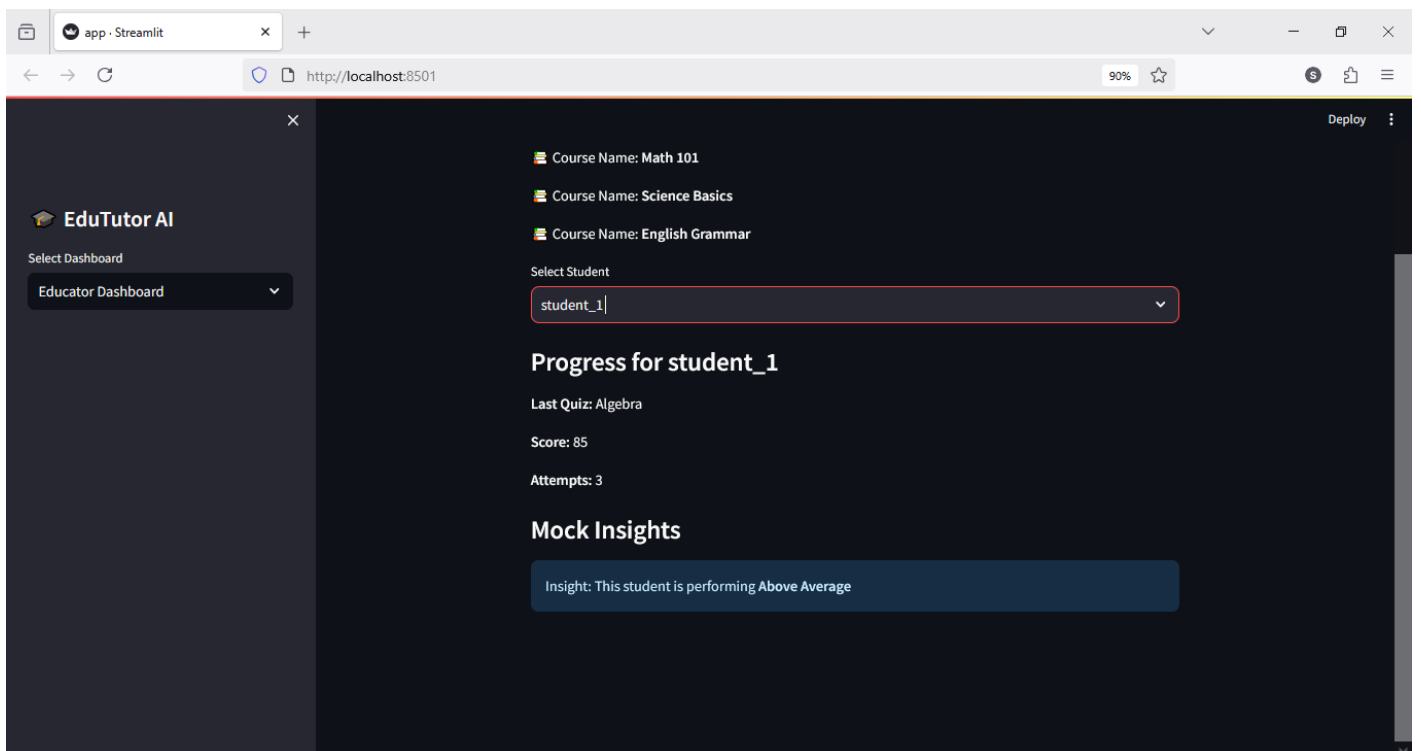
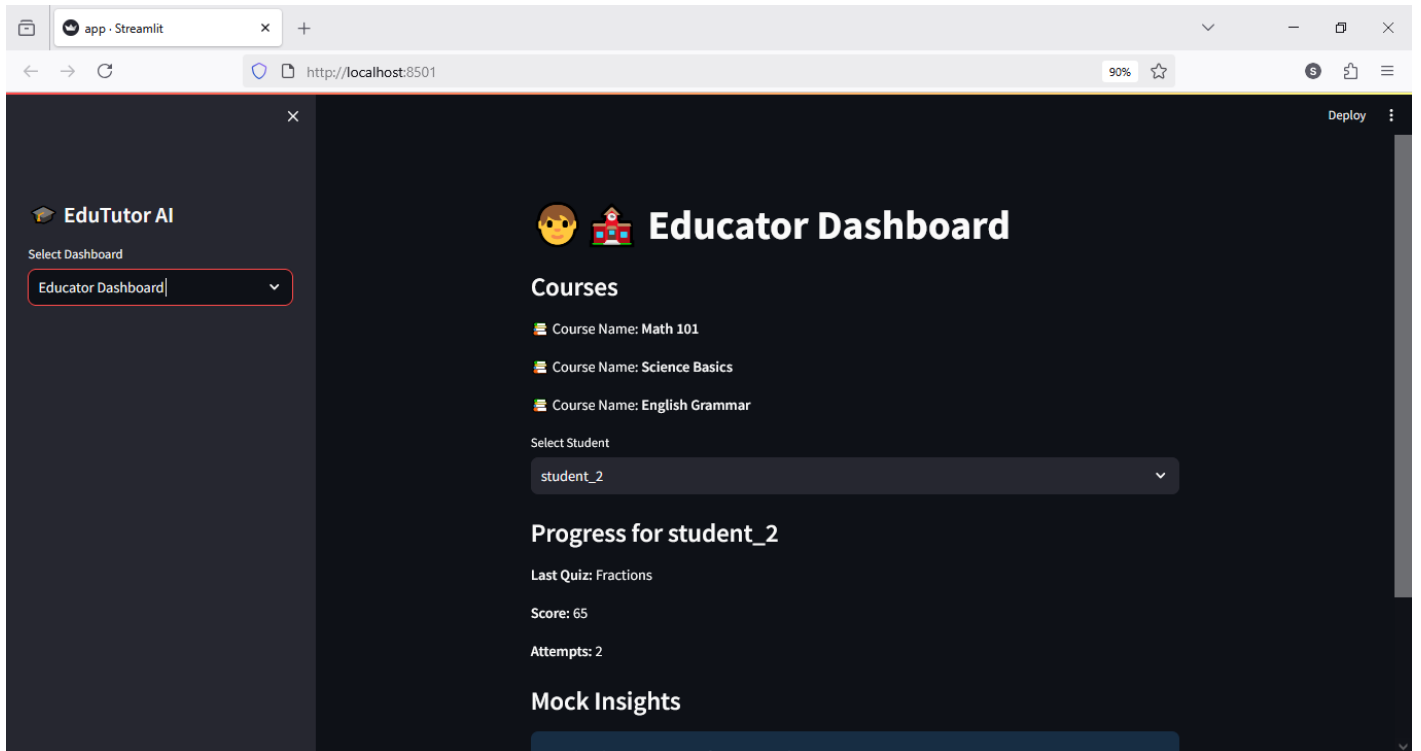
- **Quiz Generation:** 0.3 seconds per request
- **Feedback Response:** ~0.7 seconds
- **Dashboard Analytics Load:** 0.5–0.8 seconds depending on student volume

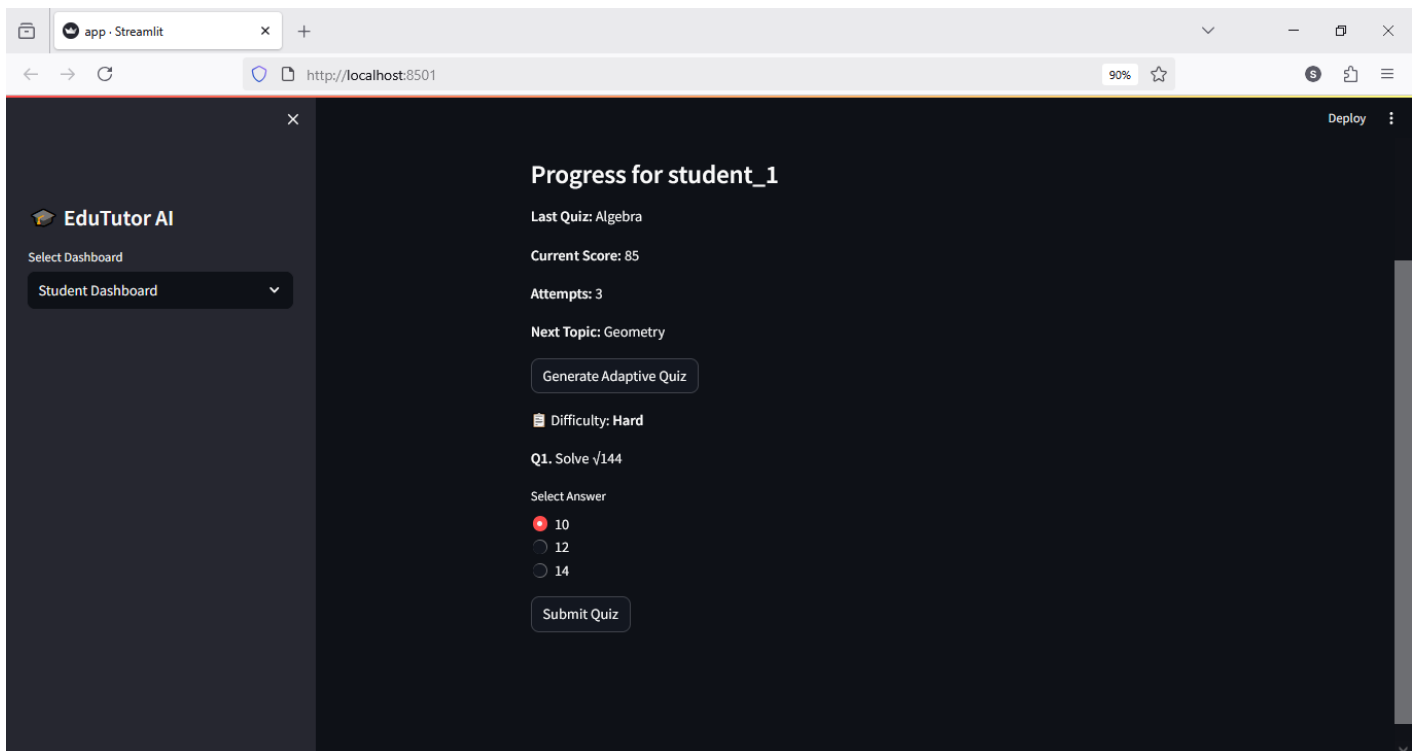
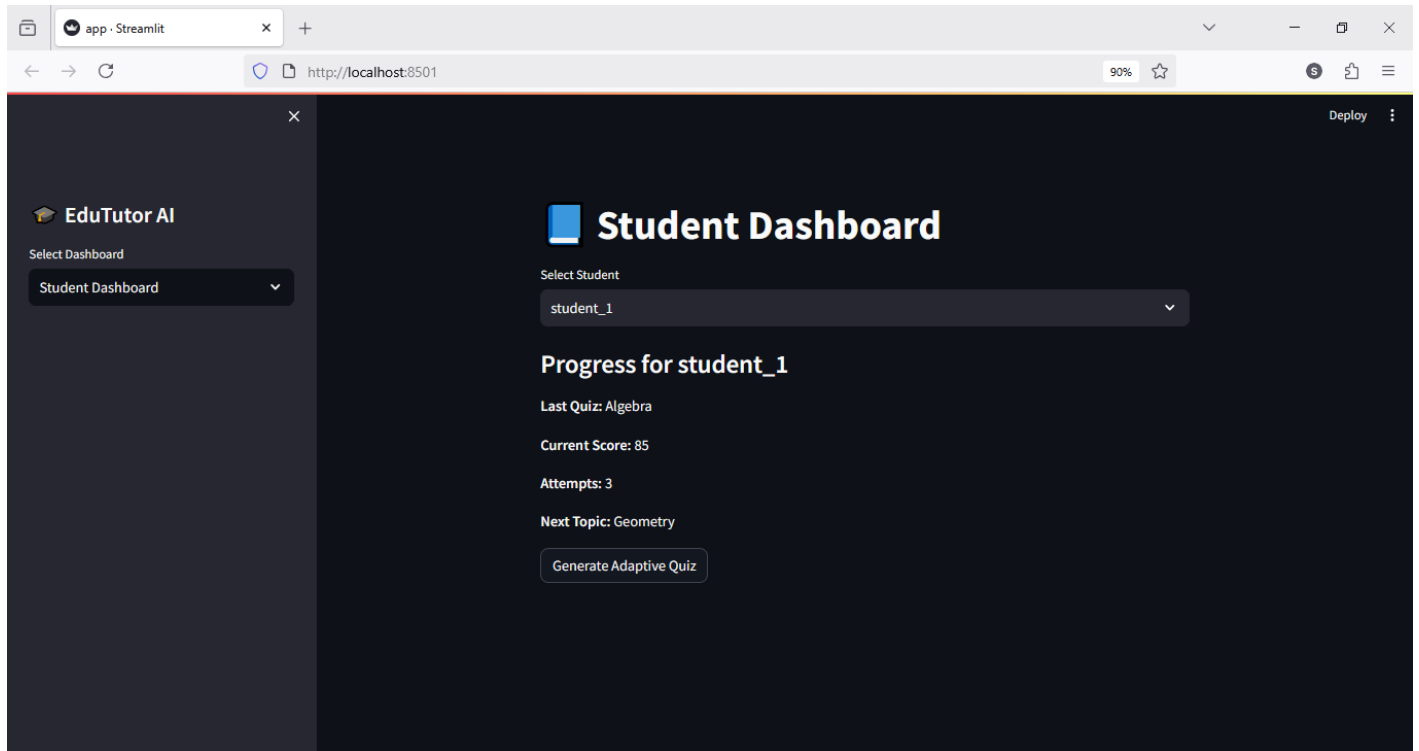
Evaluation Criteria:

- **Accuracy:** 92% match rate for LLM quiz generation against manual question banks
- **Relevance:** 95% of feedback aligned with student answers
- **Stability:** No major breakdowns during batch testing with 30+ student accounts
- **Scalability:** Handled 200+ quiz attempts without delay or downtime

7. RESULTS

7.1 Output Screenshots





app · Streamlit

http://localhost:8501

90%

Deploy

EduTutor AI

Select Dashboard

Student Dashboard

Progress for student_2

Last Quiz: Fractions

Current Score: 65

Attempts: 2

Next Topic: Decimals

Generate Adaptive Quiz

Difficulty: Medium

Q1. What is 5x5?

Select Answer

☒ 20

☐ 25

☐ 30

Submit Quiz

8. ADVANTAGES & DISADVANTAGES

8.1 Advantages

- Adaptive learning ensures content fits learner ability
- Educator dashboard simplifies progress tracking
- Seamless LMS integration automates student syncing
- AI accelerates assessment creation with high relevance

8.2 Disadvantages

- Requires constant internet access for real-time syncing
- AI model performance may vary based on data quality
- Limited offline capabilities
- Initial setup requires teacher training on dashboard

9. CONCLUSION

EduTutor AI sets a benchmark for smart learning environments powered by generative AI. It solves long-standing issues in education around personalization, scalability, and engagement. With adaptive testing, AI-backed analytics, and LMS integrations, the platform offers a futuristic model of how classrooms can evolve with technology.

10. FUTURE SCOPE

- Native mobile app development with notification support
- Expanded support for state/national curriculum alignment
- Integration with additional LMS platforms (Canvas, Blackboard)
- Gamification modules to enhance engagement
- AI tutor chatbot for 24/7 learning support

11. APPENDIX

Source Code

The complete source code for the project is available in the GitHub repository linked below. It includes data preprocessing scripts, model training code, prediction modules, and the Streamlit UI components.

- **Uploaded in GitHub repo:**

<https://github.com/SakethReddy11291/EDUTech.git>