### Project Tittle: Edu Tutor AI: Personalized Learning with Generative AI and LMS Integration

Team ID : NM2025TMID11780

Team Size : 4

Team Leader : LAKSHMAN R

Team member : DEEPAK V

Team member : DHARMENDRAN R

Team member : KALAIVANAN A

**EduTutor AI: Personalized Learning with Generative AI and LMS Integration**

**Category: Generative AI with IBM**

**Skills Required:**  
IBM Cloud

**Project Description:**

**Project description:**

EduTutor AI is an AI-powered personalized education platform that revolutionizes the way students learn, and educators assess progress. It provides dynamic quiz generation, student evaluation, Google Classroom integration, and real-time feedback—all powered by IBM Watsonx and Granite foundation models. Designed with modular architecture, this platform streamlines personalized education and enhances learning outcomes for students across academic levels.

# Scenario 1: Personalized Learning Experience

A student logs into EduTutor AI and synchronizes their courses using their Google Classroom credentials. The platform analyses course data, generates quizzes on key topics using the Granite LLM, and assesses responses for instant feedback—creating a highly personalized and engaging learning journey**.**

# Scenario 2: Educator Dashboard & Performance Insights

Educators can log in to view real-time quiz performance of all students. The dashboard highlights quiz history, scores, last topics attempted, and insights fetched from the Pinecone vector database. This empowers teachers to monitor learning progress and personalize their instruction based on data.

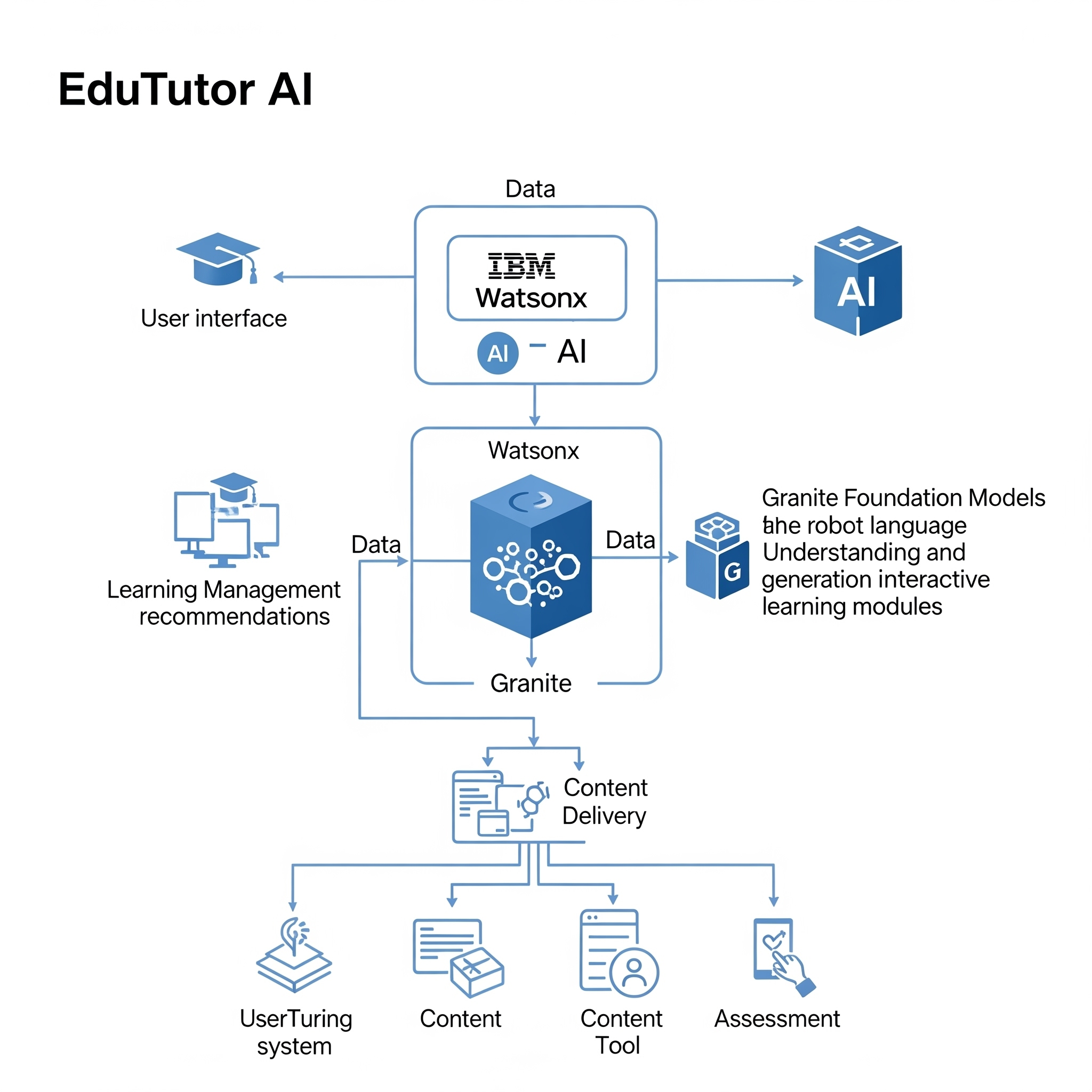
# Scenario 3: Diagnostic Testing and Adaptive Quizzing

Upon registration, students undergo a diagnostic test generated by IBM Watsonx models. Based on the results, the platform adapts quiz difficulty and topic relevance, ensuring students are challenged at the right level.

# Scenario 4: Google Classroom Integration

EduTutor AI syncs courses directly from Google Classroom, allowing seamless access to student data, class names, and subjects. This enables automatic quiz topic generation and helps maintain consistent alignment with the academic curriculum

### Architecture



### project flow

1. **User Input:**  
    Students log in using credentials or Google Classroom and request a quiz by selecting topic and difficult**y.**
2. **AI Quiz Generation:**  
    Watsonx+Granite models generate MCQs, stored temporarily without answers in the frontend and with answers in the backend
3. **User Quiz Submission:**  
    **Students submit answers via UI. The backend evaluates the answers, scores the quiz, and stores it in Pinecone DB.But Some errors couldnt built the frontend part of it**
4. **Feedback Loop:**
5. **1.Educators access this data in the dashboard.**

### Requirements Specification

* **fastapi**
* **uvicorn**
* **langchain\_ibm**
* **pinecone**
* **streamlit**
* **google-auth-oauthlib**
* **google-api-python-client**
* **python-dotenv**

**Installation:**

**pip install -r requirements.txt**

### Initialization of Environment Variables

**Create a .env file with:**

**env**

**WATSONX\_MODEL\_ID=granite-13b-instruct-v2**

**WATSONX\_API\_KEY=your\_ibm\_watsonx\_api\_key**

**WATSONX\_ENDPOINT=https://us-south.ml.cloud.ibm.com**

**WATSONX\_PROJECT\_ID=your\_project\_id**

**PINECONE\_API\_KEY=your\_pinecone\_api\_key**

**PINECONE\_INDEX\_NAME=edututor**

### AI Integration with IBM Watsonx

* **Model Setup: Granite model is loaded via langchain\_ibm.WatsonxLLM.**
* **Prompt Template: Dynamically generates quiz questions using LangChain’s PromptTemplate.**
* **Quiz Parsing: Watsonx output is parsed into structured JSON for display and evaluation.**

### Google Classroom Sync

* **Uses google-auth-oauthlib and google-api-python-client.**

**User logs in via Google.**

### Pinecone Vector DB Integration

* **Stores each user's profile with embeddings.**
* **Quiz metadata (score, topic, date) is updated after submission.**
* **Allows educators to fetch student data and analyze progress.**

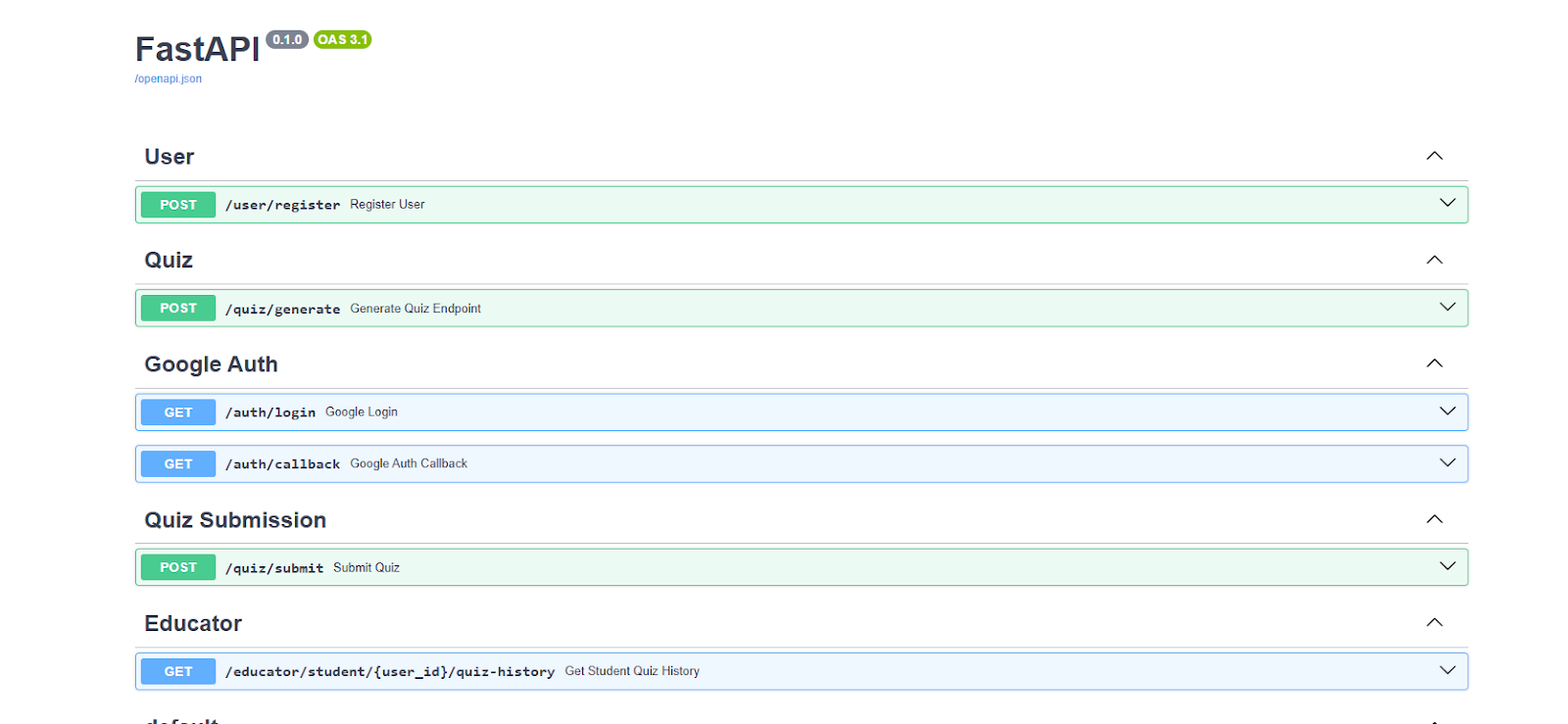
### Streamlit Frontend UI

* **Student Panel:**  
    
  + **Login (Manual or Google)**
  + **Dashboard**
  + **Take Quiz**
  + **Quiz History**
* **Educator Panel:**  
    
  + **Dashboard (all student analytics)**

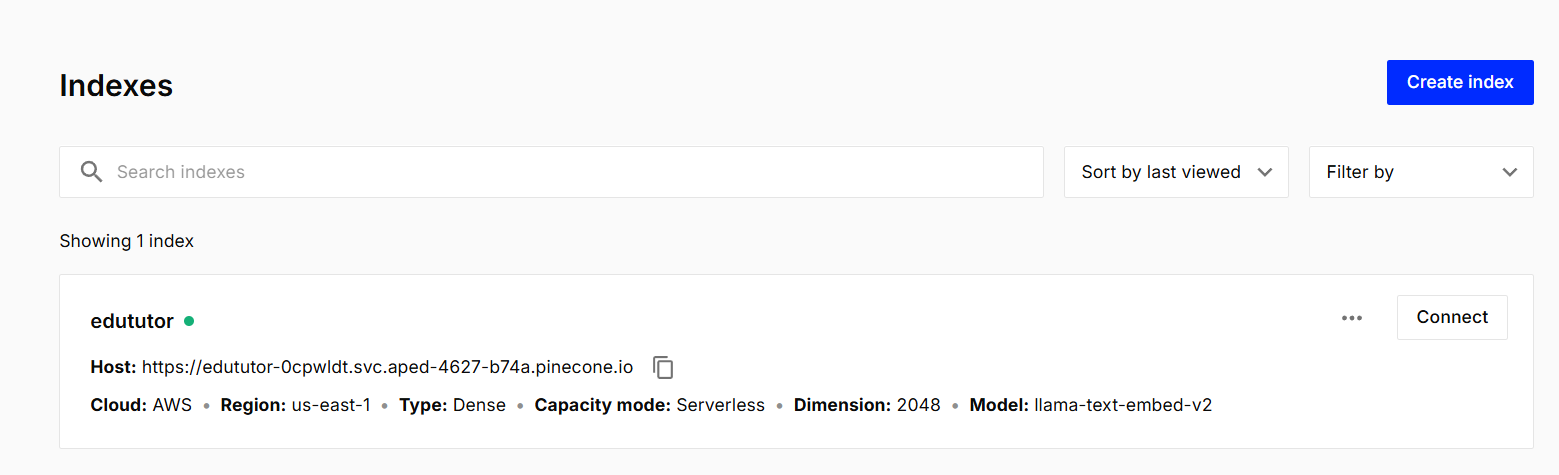
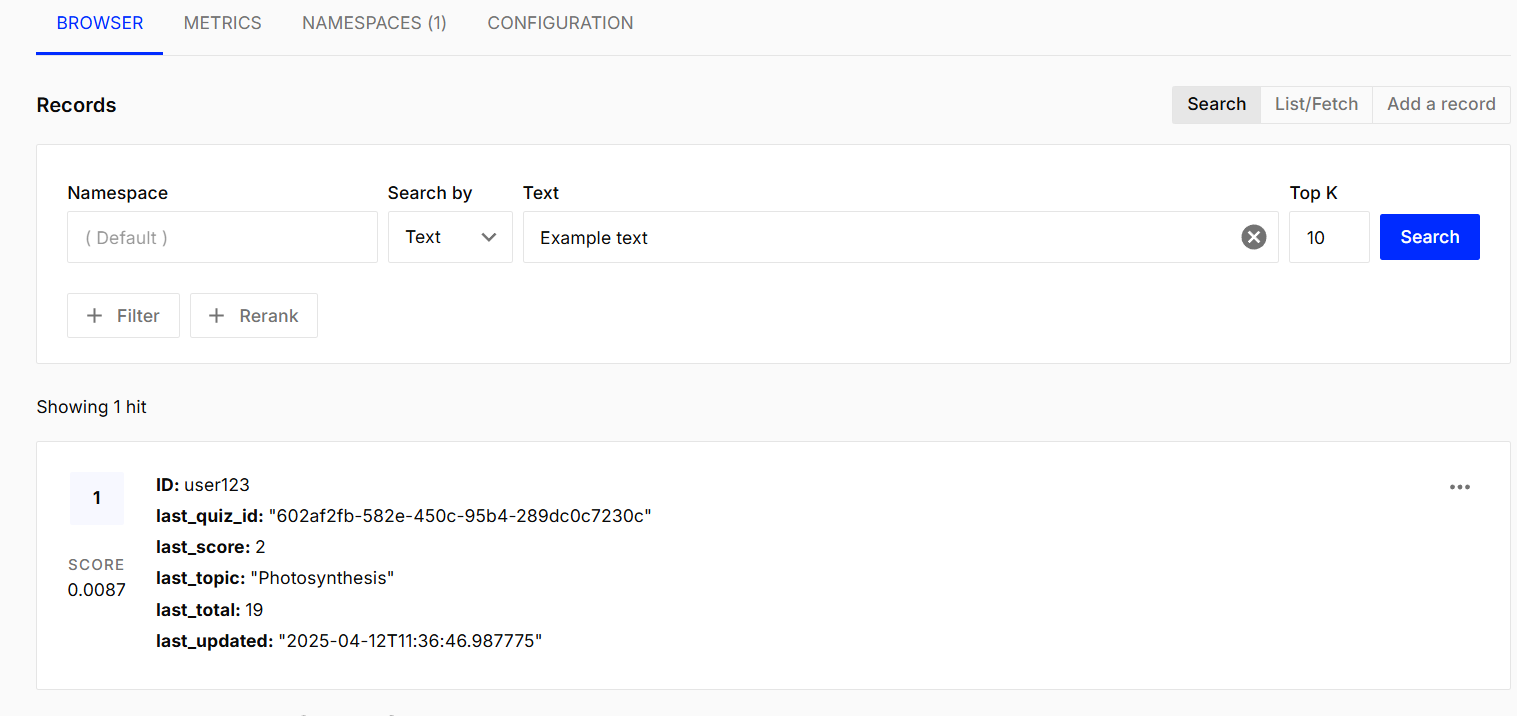
### functional verification

### **Modular AI-Powered Architecture**

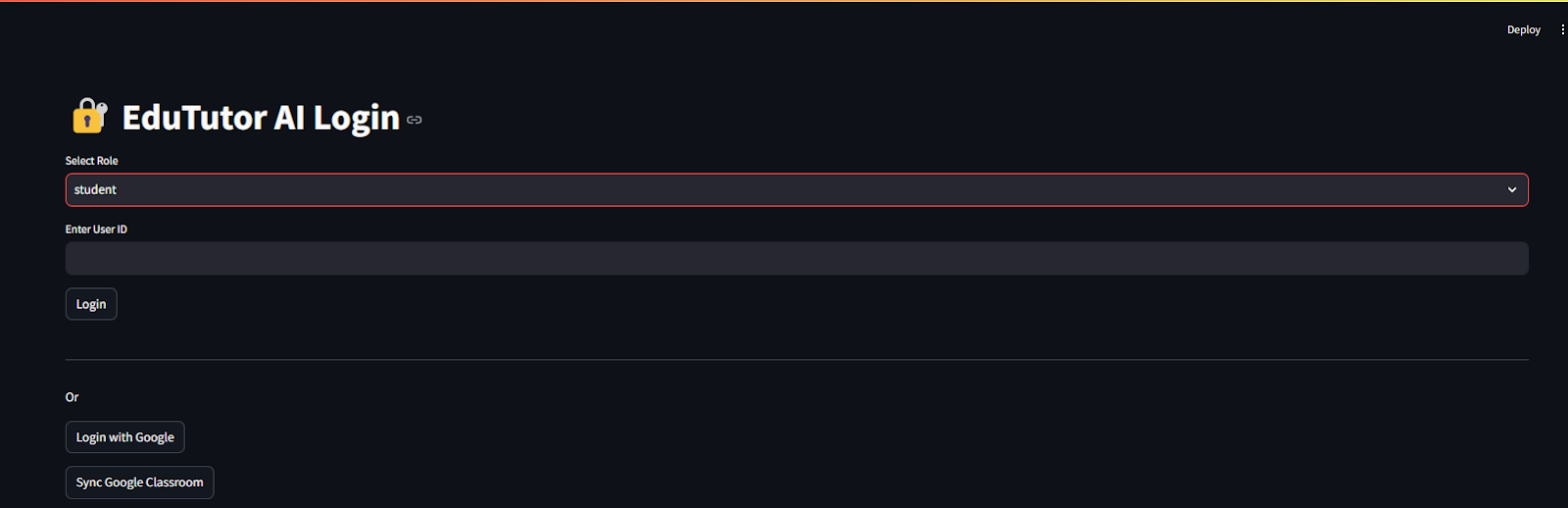
* **FastAPI Backend**  
   Handles student/educator login, quiz generation, answer evaluation, classroom sync, and metadata updates.



**Pinecone Vector DB**  
stores user profile embeddings, quiz history metadata, and similarity search for adaptive learning.

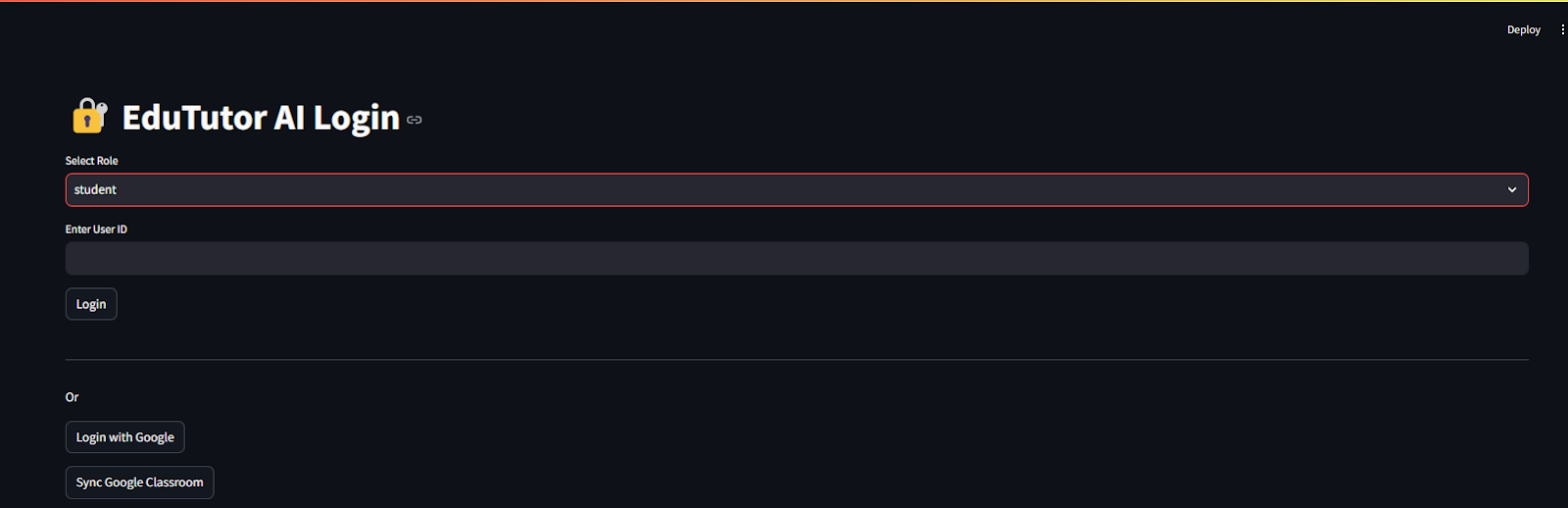


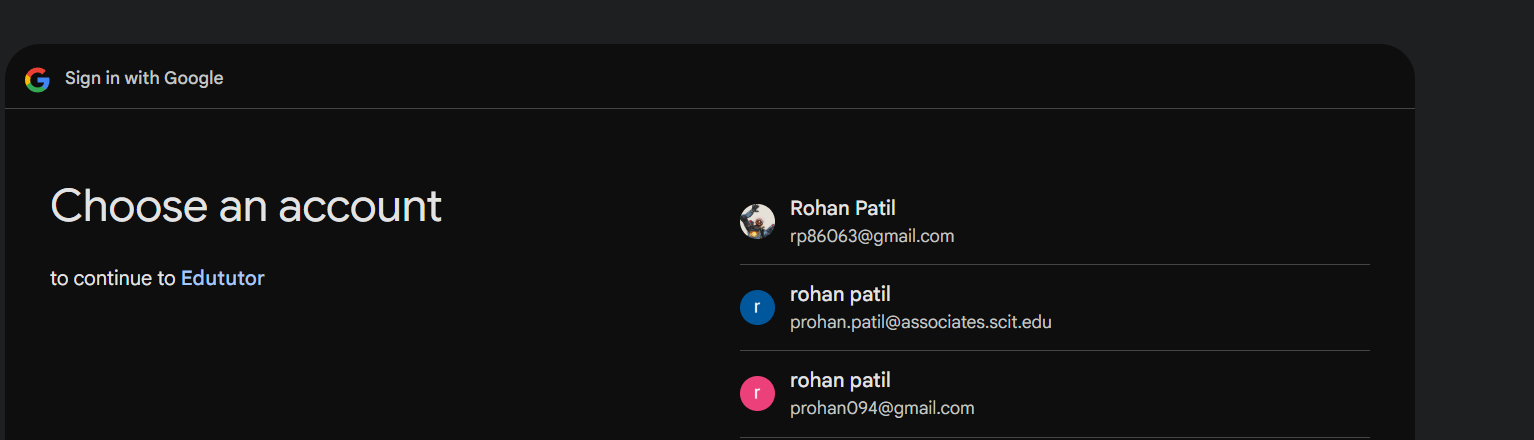
**Streamlit Frontend**  
provides dashboards for both students and educators with role-based UI, quiz submission forms, and OAuth-based login



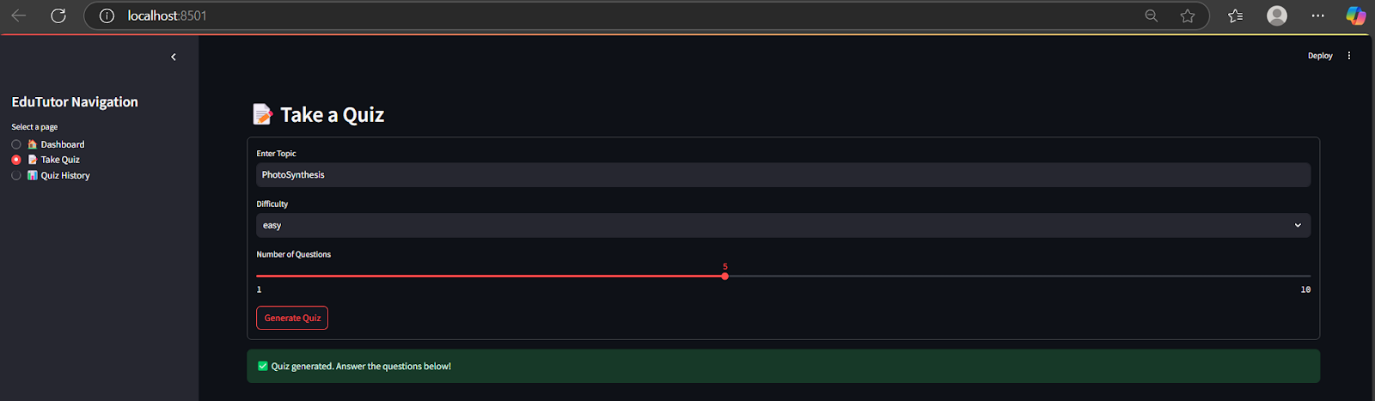
### **Project Flow:**

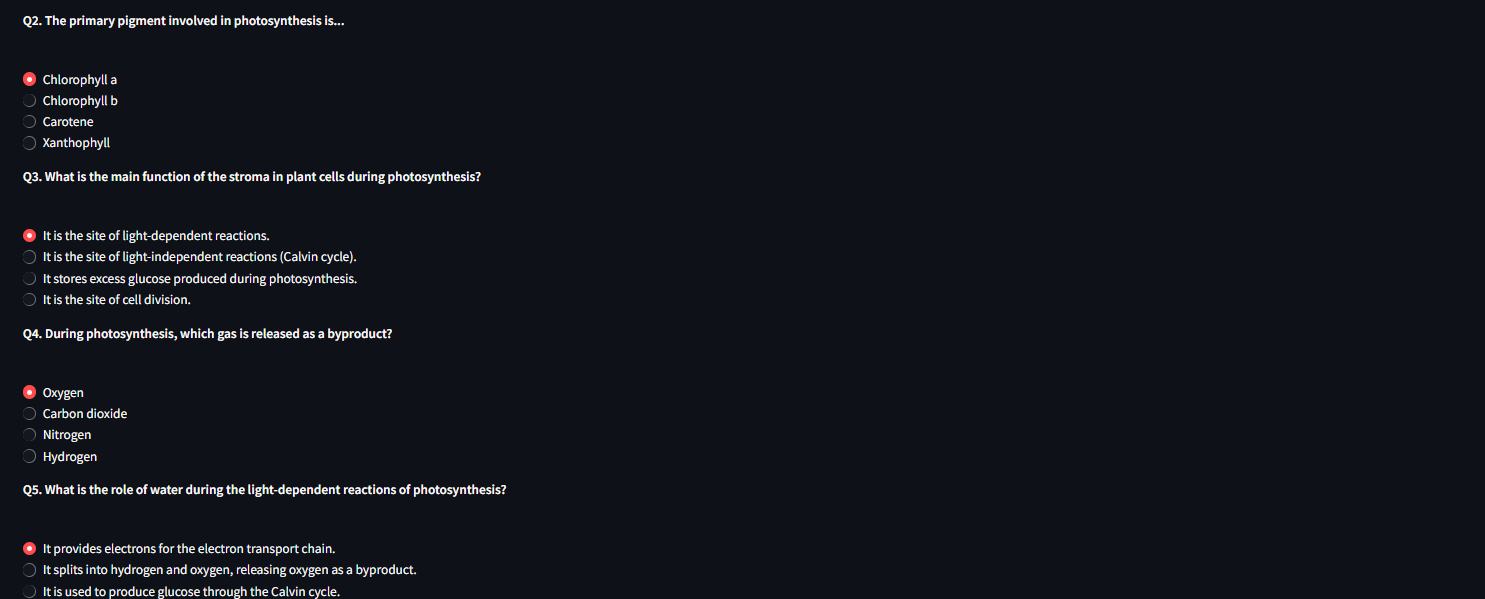
1. **User Input:**  
    Students log in using credentials or Google Classroom and request a quiz by selecting topic and difficult





1. **AI Quiz Generation:**  
    Watsonx+Granite models generate MCQs, stored temporarily without answers in the frontend and with answers in the backend.





1. **User Quiz Submission:**  
    **Students submit answers via UI. The backend evaluates the answers, scores the quiz, and stores it in Pinecone DB.But Some errors couldnt built the frontend part of it**

