Project Title: **Personalized Learning with Generative AI and LMS Integration.**

1.Introduction :

Team ID : NM2025TMID06827

Team Size : 4

Team Leader : MAHALAKSHMI R

Team member : KALPANA A

Team member : KAVIYA P

Team member : KIRUTHIKA I

Project Summary:

EduTutor AI is an intelligent, adaptive learning platform that leverages generative AI to deliver personalized educational experiences for students. The platform integrates seamlessly with existing Learning Management Systems (LMS) to enhance curriculum delivery, automate tutoring, and provide real-time feedback. By combining AI-driven content generation, student analytics, and dynamic learning pathways, EduTutor AI aims to revolutionize digital education and close learning gaps through tailored instruction.

Milestone 1: Requirements Specification

- **fastapi**
- **uvicorn**
- **langchain_ibm**
- **pinecone**
- **streamlit**
- **google-auth-oauthlib**
- **google-api-python-client**
- **python-dotenv**

Installation:

pip install -r requirements.txt

Requirements

- **fastapi**

- **uvicorn**
- **langchain_ibm**
- **pinecone**
- **streamlit**
- **google-auth-oauthlib**
- **google-api-python-client**
- **python-dotenv**

Installation:

pip install -r requirements.txt

Milestone 2: 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

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
```

Milestone 3: 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.**

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.**

Milestone 4: Google Classroom Sync

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

User logs in via Google.

Google Classroom Sync

- **Uses `google-auth-oauthlib` and `google-api-python-client`.**
- **User logs in via Google.**

Milestone 5: 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.**

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.**

Milestone 6: Streamlit Frontend UI

- **Student Panel:**
    - **Login (Manual or Google)**
    - **Dashboard**
    - **Take Quiz**
    - **Quiz History**

- **Educator Panel:**
    - **Dashboard (all student analytics)**

Project Goals:

1. **Personalized Learning:** Deliver customized learning experiences based on each student's performance, learning style, and pace.
2. **Generative AI Tutoring:** Use AI to generate explanations, quizzes, flashcards, and interactive content on demand.
3. **LMS Integration:** Ensure compatibility with major LMS platforms (e.g., Moodle, Canvas, Blackboard, Google Classroom).
4. **Real-Time Feedback & Assessment:** Provide immediate feedback to learners and actionable insights to educators.
5. **Scalability & Accessibility:** Make personalized education accessible to diverse student populations at scale.

Key Features:

- **AI-Powered Virtual Tutor:** Conversational agent that explains concepts, answers questions, and adapts responses based on student comprehension.
- **Content Generation Engine:** Automatically generates supplementary materials including quizzes, summaries, and interactive simulations.
- **Progress Tracking Dashboard:** Visual dashboards for students and teachers to monitor performance and engagement.
- **Smart Remediation:** Identifies learning gaps and automatically recommends targeted resources.
- **LMS Plug-In Modules:** Drop-in compatibility with popular LMS platforms using APIs or LTI (Learning Tools Interoperability) standards.

Target Users:

- K-12 Students
- Higher Education Students
- Educators and Instructional Designers
- Schools, Universities, and Online Learning Platforms

Technology Stack:

- **Frontend:** React.js / Vue.js (for interactive UI)

- **Backend:** Node.js / Python (API services)
- **AI/ML:** OpenAI GPT-4 (for generative responses), custom models for analytics
- **Database:** PostgreSQL / MongoDB
- **LMS Integration:** LTI 1.3, SCORM, REST APIs
- **Cloud Infrastructure:** AWS / Azure / Google Cloud

Expected Outcomes:

- Increased student engagement and academic performance.
- Reduced teacher workload through automation of routine tasks.
- Enhanced accessibility of personalized learning resources.
- Scalable AI tutoring across multiple subjects and grade levels.

Success Metrics:

- Student improvement rates (e.g., test scores, comprehension benchmarks)
- User adoption rates across LMS platforms
- Reduction in time educators spend on grading/content prep
- Engagement metrics (daily active users, session duration, etc.)

Project Timeline (High-Level):

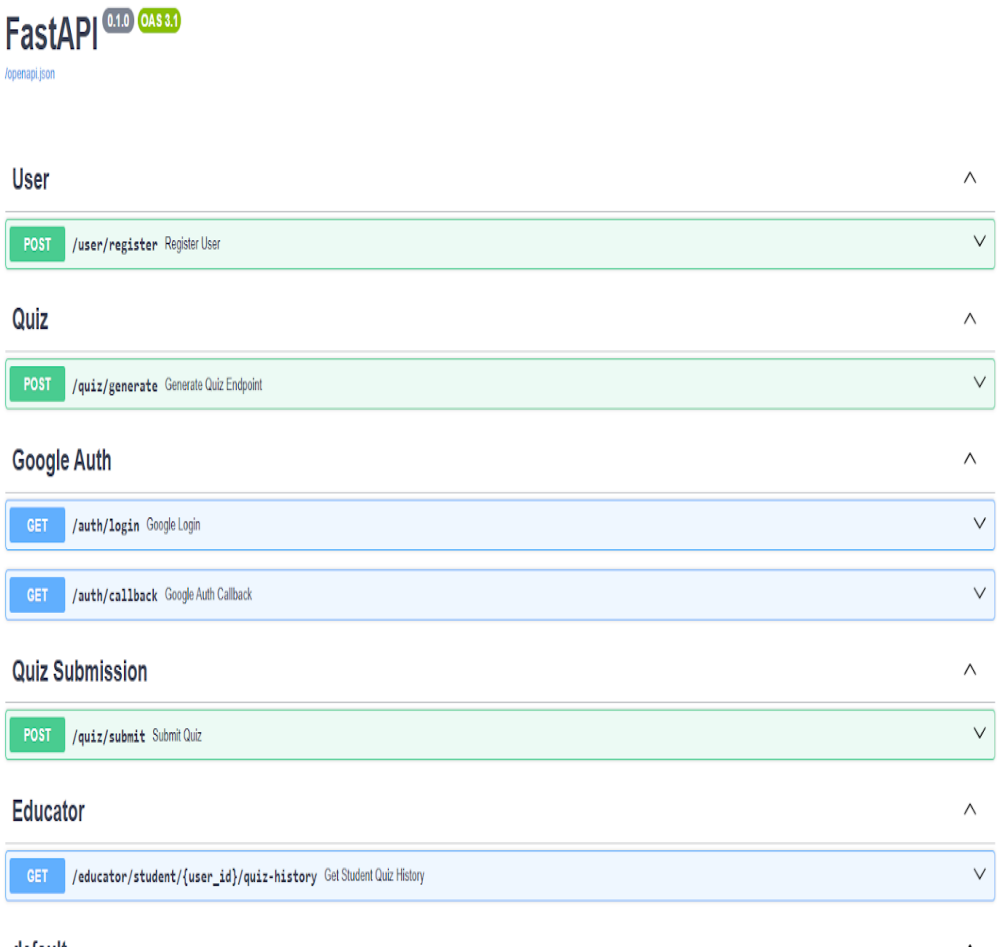
Phase	Duration	Key Milestones

Discovery & Planning	1 month	Requirement gathering, tech stack finalization
MVP Development	3 months	Core AI tutor, LMS integration, basic analytics
Beta Testing	2 months	Pilot with partner institutions
Full Launch	1 month	Public release, marketing rollout
Iteration & Expansion	Ongoing	Feedback loops, new features, scalability

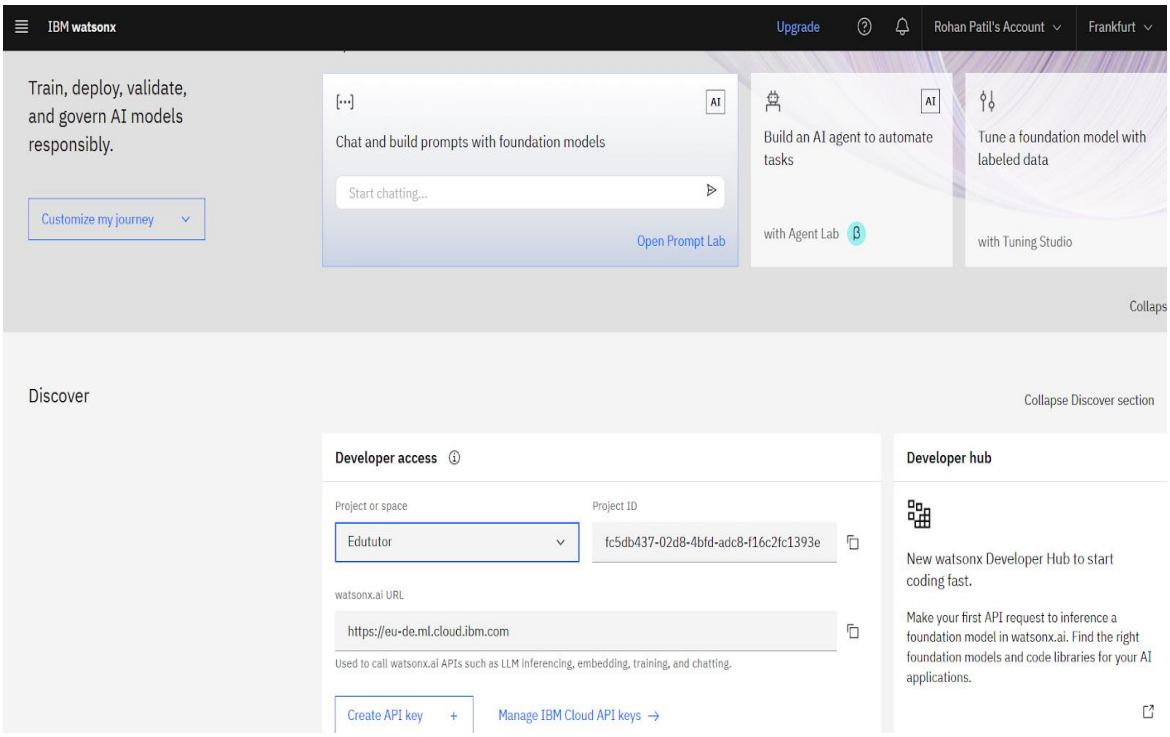
Milestone 7 : functional verification

Modular AI-Powered Architecture

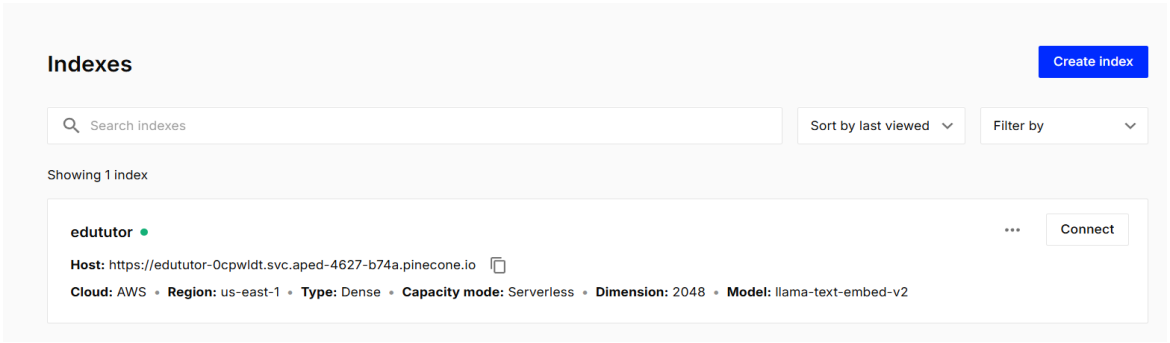
- **FastAPI Backend**
  Handles student/educator login, quiz generation, answer evaluation, classroom sync, and metadata updates.

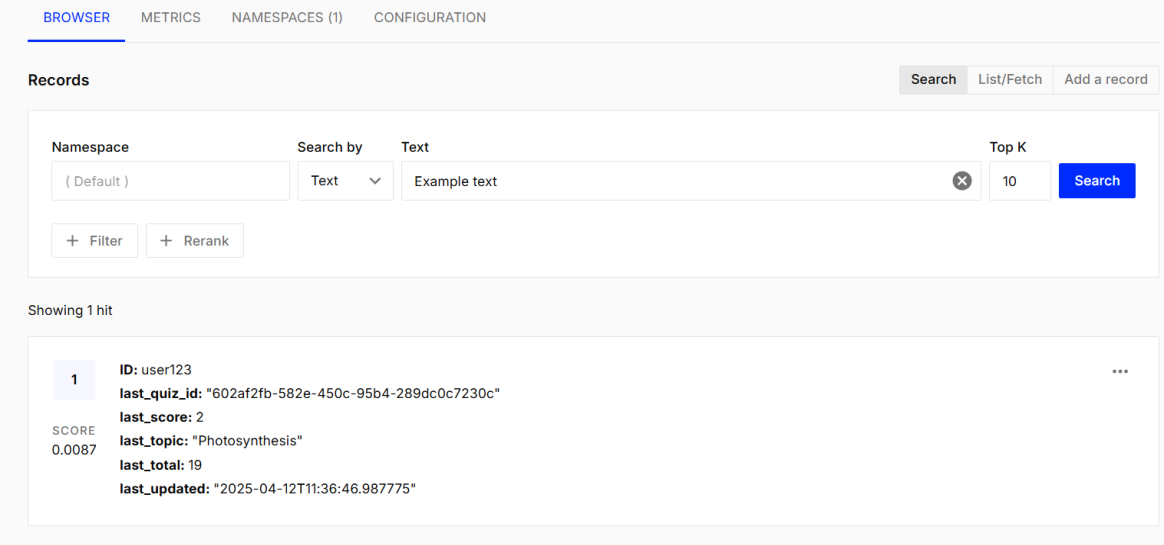


- **Watsonx + Granite Models**
  Generates MCQs dynamically from topic input. Provides instruction-tuned inference via LangChain Watsonx integration.

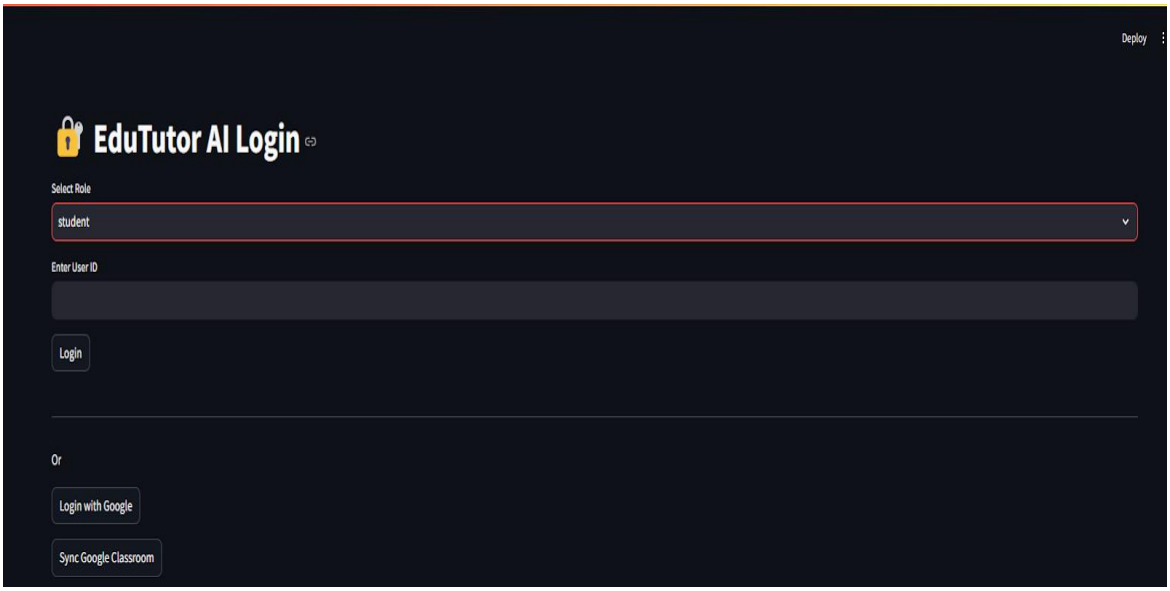


- **Pinecone Vector DB**
  stores user profile embeddings, quiz history metadata, and similarity search for adaptive learning.



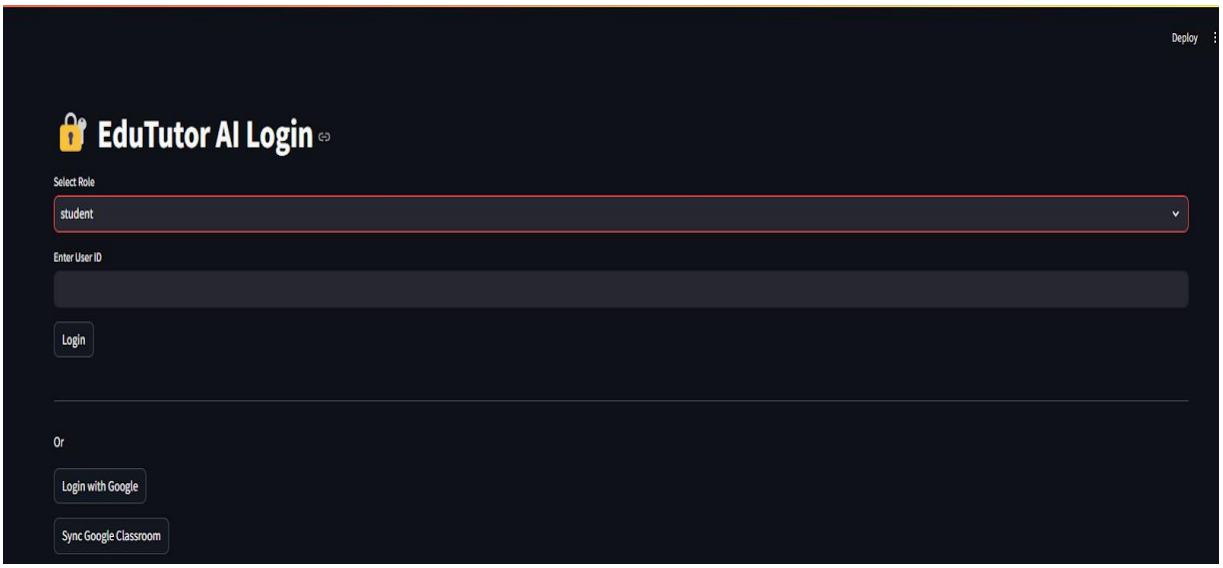


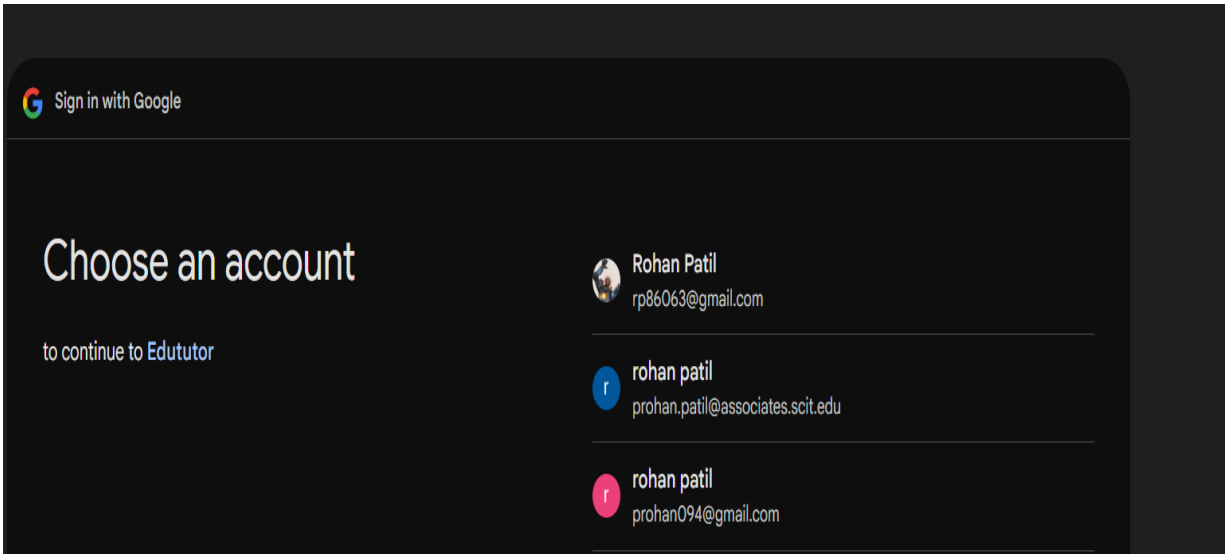
- **Streamlit Frontend**
  provides dashboards for both students and educators with role-based UI, quiz submission forms, and OAuth-based login**.**



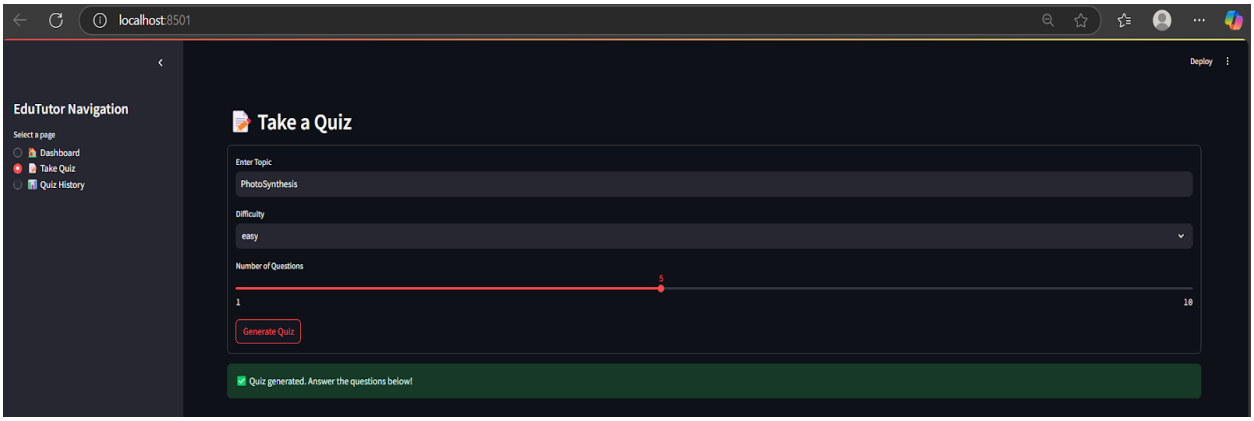
Project Flow:

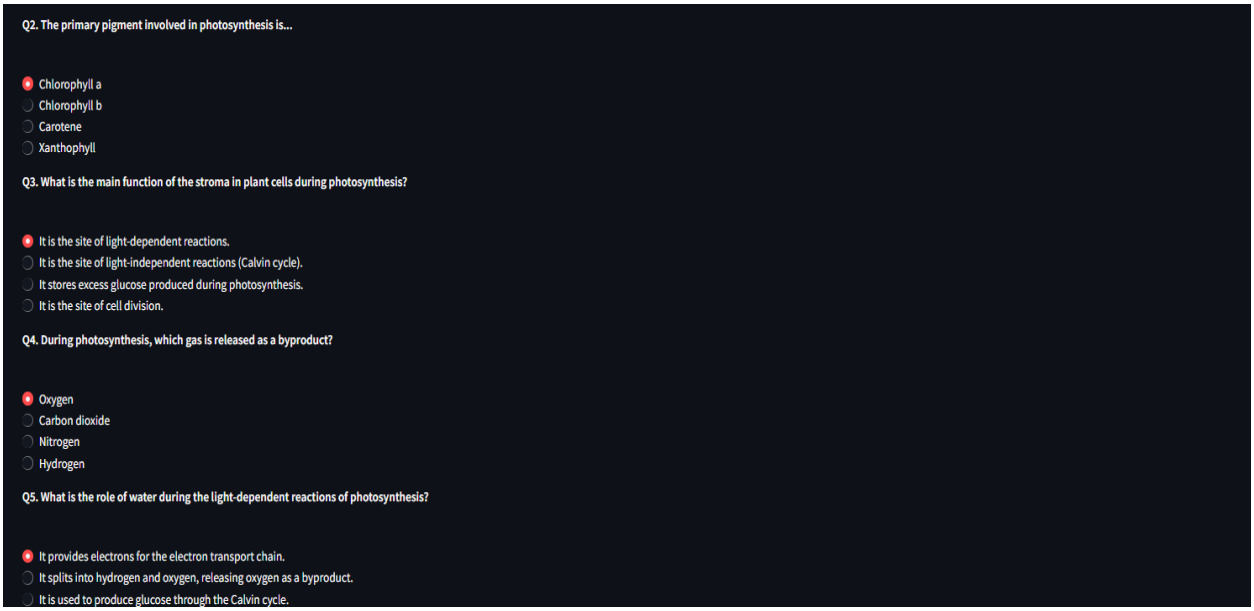
1. **User Input:**
   Students log in using credentials or Google Classroom and request a quiz by selecting topic and difficult**y.**



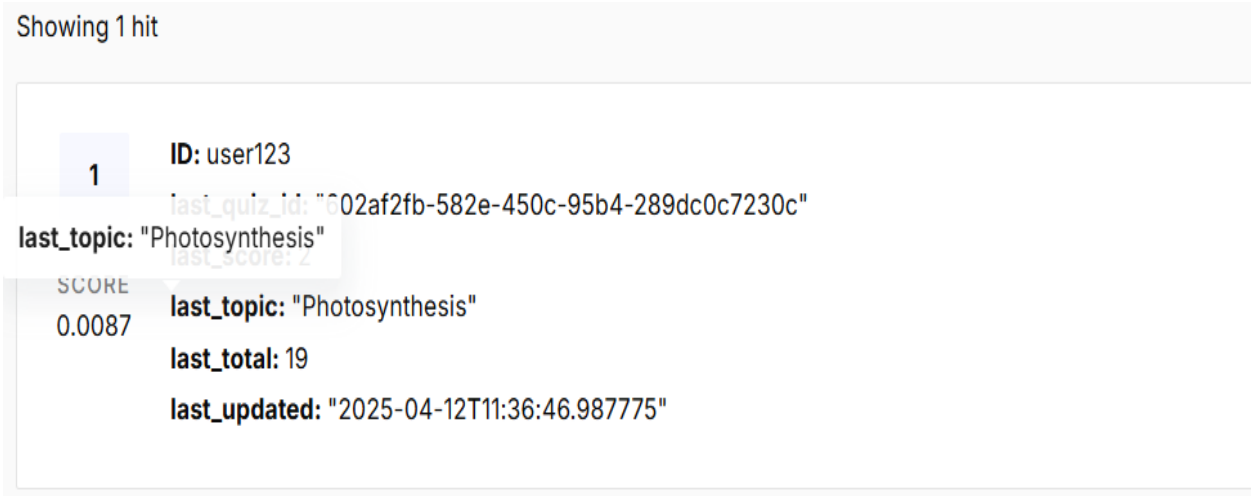


2. **AI Quiz Generation:**
   Watsonx+Granite models generate MCQs, stored temporarily without answers in the frontend and with answers in the backend.





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:**
   Educators access this data in the dashboard.