EduTutor AI: Personalized Learning with IBM Watsonx & Granite Models

Team ID : NM2025TMID12128

Team Size : 4

1.Team Leader : ROJA J

2.Team member : RITHIKA C

3.Team member : SOBIYASRI S

4.Team member : SURUTHI S

1. Introduction

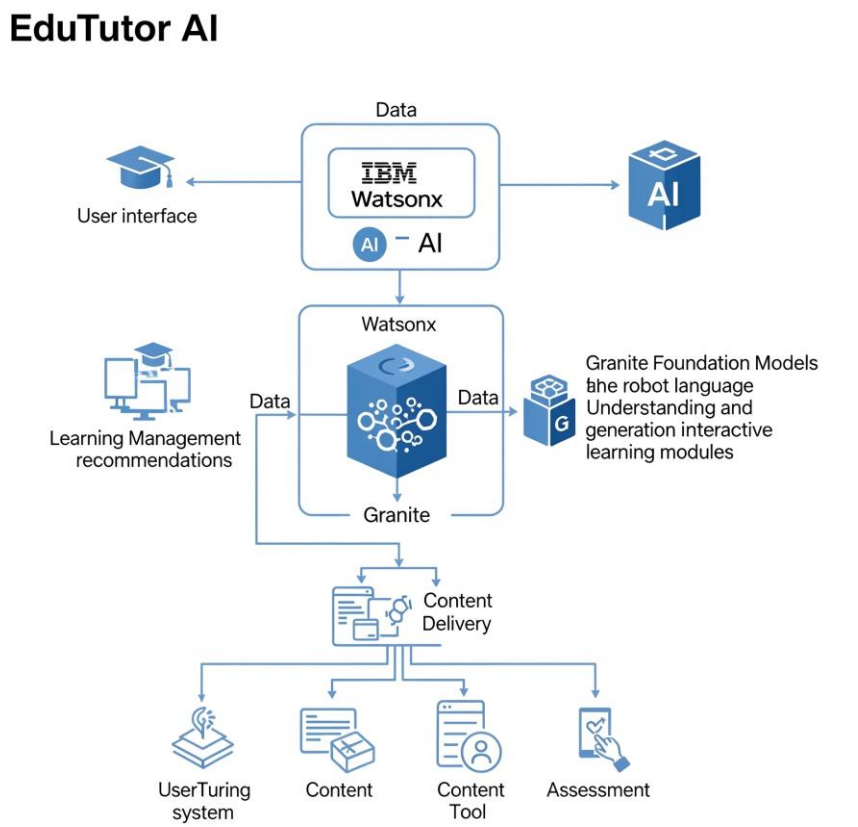
EduTutor AI is an AI-powered personalized education platform designed to transform learning and assessment.
It leverages IBM Watsonx and Granite foundation models to generate quizzes, provide instant feedback, integrate with Google Classroom,
and support adaptive learning experiences for students and educators.

2. Project Overview

Purpose:
The purpose of EduTutor AI is to provide an intelligent learning environment where students receive personalized quizzes,
real-time feedback, and adaptive learning recommendations while educators gain actionable insights into student performance.

Features:
- Dynamic quiz generation using IBM Watsonx Granite models.
- Google Classroom integration for course and student synchronization.
- Student dashboards with quiz history and progress tracking.
- Educator dashboards with analytics and insights.
- Adaptive testing with difficulty adjustments.
- Real-time feedback loops for personalized learning.

3. Architecture



The modular architecture includes:
- FastAPI Backend for login, quiz generation, evaluation, and classroom sync.
- Watsonx + Granite Models for quiz creation and adaptive testing.
- Pinecone Vector Database for storing user profiles, embeddings, and quiz history.
- Streamlit Frontend for role-based dashboards and quiz interaction.
- Google OAuth for authentication and integration.

4. Setup Instructions

1. Clone the repository from GitHub.
2. Install dependencies using: pip install -r requirements.txt
3. Create a .env file with the following variables:
   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
4. Run the backend: uvicorn main:app --reload
5. Launch the frontend using Streamlit or Gradio depending on deployment.

5. Folder Structure

Suggested folder structure:

```
/edututor-ai
  /backend
    main.py
    api/
    models/
  /frontend
    app.py
    pages/
  /config
    settings.py
  /tests
    test_quiz.py
  requirements.txt
  .env
```

6. Running the Applications

Option 1: Local Run
- Start backend with FastAPI (uvicorn).
- Start frontend with Streamlit: streamlit run app.py

Option 2: Google Colab Deployment
- Install required libraries (transformers, torch, gradio).
- Load IBM Granite model from Hugging Face.
- Launch Gradio interface in Colab.

Option 3: Cloud Deployment
- Deploy backend API on IBM Cloud or other hosting.
- Deploy frontend with Streamlit Cloud or Docker.

7. API Documentation

Key Endpoints:
- POST /login – Authenticate users (manual or Google OAuth).
- POST /generate_quiz – Generate quiz questions from a topic and difficulty.
- POST /submit_quiz – Submit answers and receive scores.
- GET /quiz_history – Fetch student's past quizzes.
- GET /educator_dashboard – Retrieve analytics for educators.

8. Authentication

Authentication is managed through:
- Manual login with username/password.
- Google OAuth 2.0 for secure Google Classroom integration.

9. User Interface

Student Panel:
- Dashboard (quiz history, progress tracking).
- Take Quiz (select topic and difficulty).
- Instant feedback.

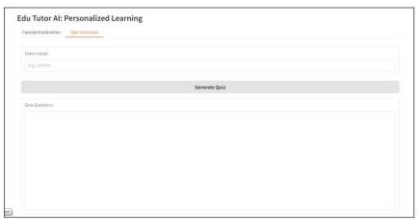
Educator Panel:
- Dashboard with student analytics.
- Access to quiz history and performance metrics.

10. Testing

Testing Process:
- Functional verification of API endpoints.
- Validation of AI quiz generation and scoring.
- Google Classroom sync testing.
- Frontend UI tests (Streamlit dashboards).

11. Output Screenshot

[Add screenshots of student and educator dashboards here]



12. Known Issues

- Frontend build errors in certain configurations.
- Limited scalability when running only on Colab.
- Integration errors during Google Classroom sync in some test cases.

13. Future Enhancement

- Improved adaptive learning using real-time embeddings.
- Mobile application for easier access.
- Advanced analytics for educators (learning trends, topic difficulty).
- Support for additional LMS platforms beyond Google Classroom.
- Voice-enabled quiz interaction and accessibility features.