

EduTutor AI: Personalized Learning with IBM Watsonx & Granite Models

Team ID : NM2025TMID12092

Team Size : 4

Team Leader : SHAHUL HAMEED J

Team member : SATHEESH KUMAR M

Team member : SHEIK SUHAIL AHAMED B

Team member : SHEIK THAMEEM 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

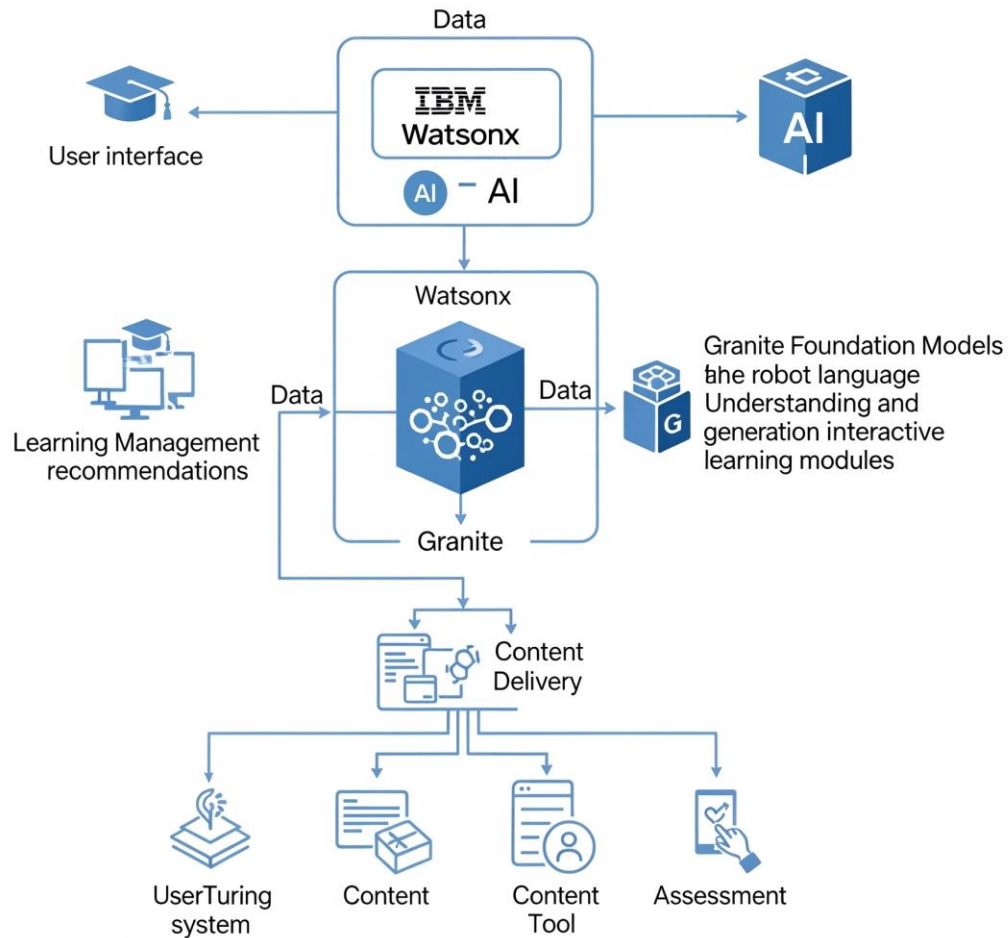
Purpose:

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.

EduTutor AI



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]

Edu Tutor AI: Personalized Learning

[Concept Explanation](#)[Quiz Generator](#)

Enter a concept

g., machine learning

Explain

Explanation

Edu Tutor AI: Personalized Learning

[Concept Explanation](#)[Quiz Generator](#)

Enter a concept

DATA MINING

Explain

Explanation

- **Association Rule Learning**: Identifies rules that describe large portions of data (e.g., market basket analysis, recommendation systems). Techniques such as Apriori and Eclat are commonly used.

4. **Evaluation**: After building models, they are evaluated for accuracy, precision, recall, F1-score, area under the ROC curve (AUC-ROC), or other relevant metrics to ensure they generalize well on unseen data.

5. **Post-processing & Deployment**: The best performing models are optimized, and their results are visualized for stakeholders. The final step involves integrating these models into operational systems or decision-making processes, enabling actionable insights and predictions.

Examples of Data Mining Applications:

- Marketing and Customer Relationship Management (CRM)**: Data mining helps in understanding customer behavior, preferences, and churn risks. For example, mining transactional data can identify patterns leading to purchases, like 'customer A bought milk and bread last Tuesday,' which can be used to recommend relevant products.
- Fraud Detection**: Banks and financial institutions use data mining to detect anomalous financial transactions that could indicate fraudulent activities. By analyzing patterns in historical transaction data, machine learning models can flag suspicious activities in real-time.

Edu Tutor AI: Personalized Learning

Concept Explanation Quiz Generator

Enter a topic
e.g., physics

Generate Quiz

Quiz Questions

Edu Tutor AI: Personalized Learning

Concept Explanation Quiz Generator

Enter a topic
COMPUTER

Generate Quiz

Quiz Questions

4. Multiple Choice: Which of the following is a secondary storage device?
a) Solid State Drive (SSD)
b) Cache Memory
c) Hard Disk Drive (HDD)
d) Random Access Memory (RAM)

5. True/False: A computer's processor speed is measured in bytes.

ANSWERS

1. Multiple Choice: b) Web browsing

2. True/False: False. RAM does not store data permanently; it temporarily stores data for quick access by the CPU.

3. Short Answer: In computer science, a bit is the basic unit of information. It can have one of two values: 0 or 1. Bits combine to form bytes, which are groups of eight bits, and are the fundamental units of digital information. Bits are used for data storage, transmission, and processing in computers.

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.