Edututor AI – Personalized Learning

1. Introduction

Project Title: Edututor AI – Personalized Learning

Team ID : NM2025TMID03855

Team Leader : MOHAMMED RISWAN A

Team member : MUKHIL A

Team member : NIRANJAN P

Team member : PRASANTH S

2. Project Overview

Purpose

Edututor AI is designed to transform education through **personalized and adaptive learning**. It leverages AI models to **explain concepts in simple terms** and **generate interactive quizzes**. Learners can study at their own pace while teachers can use the tool to enhance lessons.

Existing Features (as per code):

- **Concept Explanation** – Learners can input any topic and get a **detailed explanation with examples**.
- **Quiz Generator** – Automatically generates **5 quiz questions** (MCQs, True/False, Short Answer) and provides the **answers section**.

Planned Features:

- Learning progress tracking and analytics.
- Personalized recommendations for further study.
- Multilingual support for diverse learners.
- AI-powered doubt resolution chatbot.

---

3. Architecture

Frontend

- Built using **Gradio UI** with **two tabs**:
    1. Concept Explanation
    2. Quiz Generator

Backend

- **Model**: IBM Granite 3.2-2B Instruct (via Hugging Face).
- **Frameworks**: PyTorch, Transformers.
- **Functions in Code**:
  - concept_explanation(concept) → Generates detailed concept explanation.
  - quiz_generator(concept) → Generates 5 questions with answers.

Planned Enhancements

- FastAPI backend for scalability.
- Database for storing student progress & results.
- Adaptive learning paths using ML.

---

4. Setup Instructions

Run in Colab or local environment:

```
!pip install transformers torch gradio -q
```

Then launch the EdututorAI.ipynb code. Gradio will provide a link to access the app.

---

5. Folder Structure (Current + Planned)

```
├── edututor_project.py      # Main file (Gradio app, same as notebook)
├── app/                     # Backend logic (future)
│   ├── granite_llm.py       # Model wrapper (planned)
│   ├── document_tools.py    # Helpers (planned)
│   ├── forecast.py          # Learning analytics (planned)
│   ├── anomaly.py           # Performance anomaly detection (planned)
├── ui/                      # Frontend components (future split)
│   ├── concept_tab.py
│   ├── quiz_tab.py
├── requirements.txt
```

---

6. Running the Application

1. Install dependencies.
2. Run edututor_project.py (or Colab notebook).

3. Open the **Gradio link** in a browser.
4. Use:
   - **Concept Explanation tab** → enter topic → get explanation.
   - **Quiz Generator tab** → enter topic → get 5 quiz questions + answers.
5. Planned: Save results, track learning, and download reports.

---

7. API Documentation (Planned with FastAPI)

- POST /explain-concept – Generate concept explanation.
- POST /generate-quiz – Create quiz with answers.
- POST /track-progress – Save student progress.
- POST /recommend-topics – Suggest next learning topics.

---

8. Authentication

Future secure deployment will support:

- Token-based authentication (JWT).
- OAuth2 (Google/Microsoft login).
- Role-based access (Student/Teacher/Admin).

---

9. User Interface

- **Concept Explanation tab** → input topic, get detailed explanation.
- **Quiz Generator tab** → input topic, get 5 questions + answers.
- Planned tabs: Progress tracking, recommendations, reports.

---

10. Testing

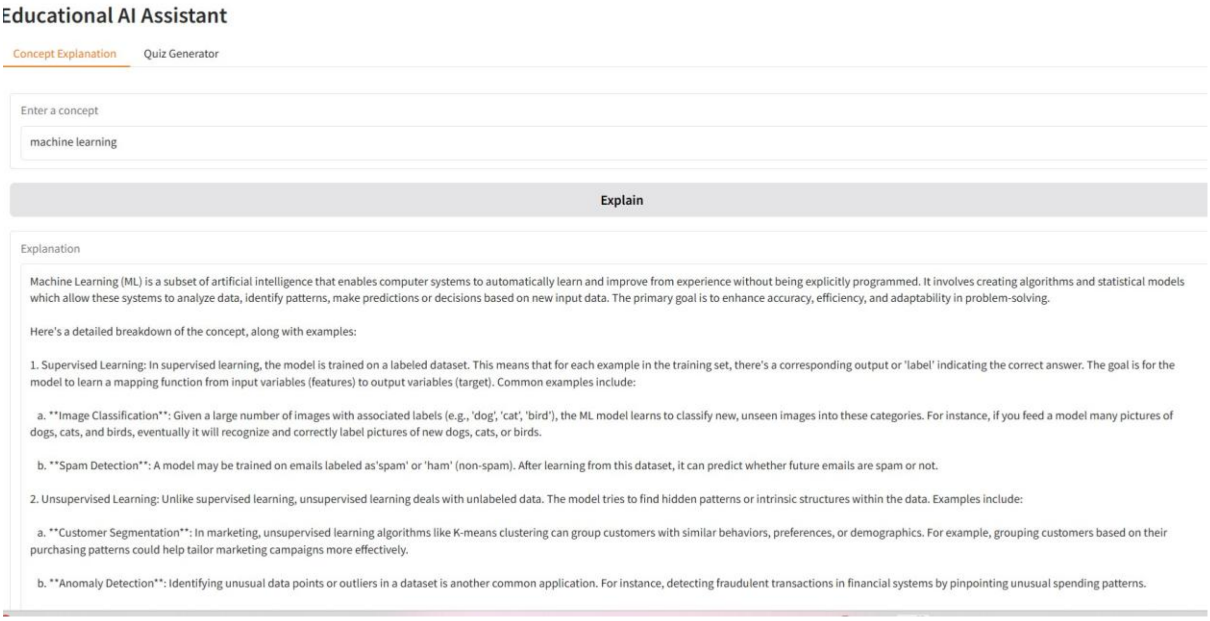
- **Unit Testing**: Model prompts, response formatting.
- **Manual Testing**: Check concept explanations & quiz quality.
- **Planned**: Automated API testing with FastAPI + Pytest.

---

11. Known Issues

- Limited to **text input only**.
- No tracking or multilingual support yet.

- No authentication in demo version.



Enter a concept: machine learning

Explanation

```
Machine Learning (ML) is a subset of artificial intelligence that enables computer systems to automatically learn and improve from experience without being explicitly programmed. It involves creating algorithms and statistical models which allow these systems to analyze data, identify patterns, make predictions or decisions based on new input data. The primary goal is to enhance accuracy, efficiency, and adaptability in problem-solving.

Here's a detailed breakdown of the concept, along with examples:

1. Supervised Learning: In supervised learning, the model is trained on a labeled dataset. This means that for each example in the training set, there's a corresponding output or 'label' indicating the correct answer. The goal is for the model to learn a mapping function from input variables (features) to output variables (target). Common examples include:

  a. **Image Classification**: Given a large number of images with associated labels (e.g., 'dog', 'cat', 'bird'), the ML model learns to classify new, unseen images into these categories. For instance, if you feed a model many pictures of dogs, cats, and birds, eventually it will recognize and correctly label pictures of new dogs, cats, or birds.

  b. **Spam Detection**: A model may be trained on emails labeled as'spam' or 'ham' (non-spam). After learning from this dataset, it can predict whether future emails are spam or not.

2. Unsupervised Learning: Unlike supervised learning, unsupervised learning deals with unlabeled data. The model tries to find hidden patterns or intrinsic structures within the data. Examples include:

  a. **Customer Segmentation**: In marketing, unsupervised learning algorithms like K-means clustering can group customers with similar behaviors, preferences, or demographics. For example, grouping customers based on their purchasing patterns could help tailor marketing campaigns more effectively.

  b. **Anomaly Detection**: Identifying unusual data points or outliers in a dataset is another common application. For instance, detecting fraudulent transactions in financial systems by pinpointing unusual spending patterns.
```

---

13. Future Enhancements

- Add **progress tracking & analytics**.
- Support CSV/Excel for bulk quiz generation.
- AI-based doubt resolution.
- Generate **PDF learning reports**.
- Voice-based input/output for accessibility.

Project Demo video link:

https://drive.google.com/file/d/1kl5FGJzT8EoYLtfZ1xyiKmY8Agmap9m-/view?usp=drivesdk