

Project Report

EduTutor AI: Personalized Learning with Generative AI and LMS Integration

1. Introduction

- Project Title: EduTutor AI: Personalized Learning with Generative AI and LMS Integration
- Team ID: NM2025TMID03354
- Team Member: DHANASRI A
- Team Member: HARINI B
- Team Member: HARINI P
- Team Member: PADMA PRIYA T

2. Project Overview

➤ Purpose:

The Educational AI Assistant is designed to be an intelligent, user-friendly, and interactive learning tool that helps students and teachers better grasp academic concepts. As the need for personalized and engaging education grows, students often utilize a variety of resources that aren't always organized, trustworthy, or easy to understand. This project fills that gap by utilizing AI to provide explanations that are context-aware, create quizzes for practice, and generate flashcards for quick review. The goal of the system is to make learning more effective, interactive, and student-friendly.

➤ Conversational Interface:

Key Point: AI-powered natural language learning support

Functionality: Students and teachers can type questions in plain language, get detailed explanations of ideas, make quizzes with answers, and make flashcards for quick review, all in a simple interactive interface. It makes learning easier for everyone by lowering the need for technical knowledge. The interface encourages active participation by making it feel like a tutor, allowing students to ask questions, explore, and make changes without any problems.

➤ **Concept Explanation:**

Key Point: Simplified learning of academic topics

Functionality:

- Takes the user's input concept (e.g., *machine learning*, *data structures*).
- Passes the input as a prompt to the AI model (granite-3.2-2b-instruct).
- The model generates a **detailed explanation** with examples, ensuring clarity for students.
- The response is displayed in the Gradio textbox in an easy-to-read format.
- Helps students avoid spending extra time searching multiple online sources.

➤ **Quiz Generator:**

Key Point: Interactive self-assessment

Functionality:

- Takes a subject from the user, like physics or networking.
- The AI makes five quiz questions in different styles, such as multiple choice, true/false, and short answer.
- Using a structured format (ANSWERS: keyword), it automatically separates the questions from their answers.
- Shows questions in one box and answers in another, so students can take the quiz first and then check their answers.

➤ **Flashcards**

Key Point: Quick revision support

Functionality:

- Takes an idea or topic as input (like operating system ideas).
- Makes five flashcards in this style:
 Q: <question>
 A: <answer>
- Gives them to students in a small, easy-to-use format.
- Helps you remember things by letting you review them quickly and often.
- Very helpful for studying for tests and making last-minute changes.

3. Architecture

The architecture of the Educational AI Assistant is designed with a modular, layered approach to provide scalability, clarity, and intelligent learning support. It combines a user-friendly frontend, a robust backend, and advanced AI-powered natural language processing to deliver explanations, quizzes, and flashcards in real time. The system is divided into four main layers:

i. **AI Assistant Layer (IBM Granite LLM Integration)**

This is the **core intelligence layer** of the system. It is responsible for understanding user input and generating meaningful, structured educational content. The integration of **IBM Granite 3.2B Instruct LLM** ensures that responses are not only accurate but also context-aware and tailored for academic purposes.

- **Natural Language Understanding:** The model interprets student questions, academic terms, and the contextual meaning behind prompts.
- **Intelligent Response Generation:** It produces detailed concept explanations, multiple types of quiz questions with answers, and concise flashcards.
- **Adaptive Learning:** By modifying prompts, the system can scale to new subjects, making it flexible for future enhancements.
- **Reliability:** Unlike random online searches, the model generates structured and curated educational responses.

Tools Used:

- **IBM Granite LLM** – Core large language model for academic query processing and content generation.
- **Hugging Face Transformers** – Framework for model loading, tokenization, and text generation.
- **Python** – Programming backbone for integration, feature implementation, and backend workflows.
- **Gradio** – UI framework providing an interactive, multi-tab interface (Concept Explanation, Quiz Generator, Flashcards).

ii. **Backend Layer (Python + Transformers)**

The backend acts as the **processing engine** of the system. It manages all interactions between the frontend interface and the AI model.

- **Prompt Handling:** Takes user input, reformats it into prompts suitable for the LLM, and handles tokenization.

- **Execution Optimization:** Utilizes GPU acceleration when available (`torch.cuda.is_available()`), making the system efficient.
- **Function Modules:** Implements dedicated functions such as `concept_explanation()`, `quiz_generator()`, and `flashcard_generator()` to keep the code modular and maintainable.
- **Output Processing:** Separates raw model output into user-friendly sections, such as splitting quiz questions from their answers.

This layer ensures smooth communication between the user interface and the AI assistant, making the system reliable and easy to extend.

iii. Frontend Layer (Gradio User Interface)

The frontend provides a **simple and interactive interface** for students and educators to use the platform without technical expertise. Built using **Gradio**, it offers an accessible design with tab-based navigation.

- **Tabs for Multiple Features:**
 - Concept Explanation Tab – Generates detailed, example-based explanations.
 - Quiz Generator Tab – Produces quizzes with a separate section for answers.
 - Flashcards Tab – Creates short Q/A cards for revision.
- **Custom Theme:** Red and green gradient buttons and layouts enhance visual appeal, making learning more engaging.
- **Ease of Use:** The frontend is lightweight, browser-based, and requires no installation from the user side.

This layer focuses on usability, ensuring the system is friendly to both beginners and advanced learners.

iv. Output Layer (Learning Support Modules)

The final layer is the **educational output** delivered to the user. This is where AI-powered results are transformed into **practical learning aids**.

- **Concept Explanation:** Provides detailed, example-rich explanations that break down complex topics into easy-to-understand content.

- **Quiz Generator:** Creates five quiz questions in varied formats (MCQs, true/false, short answer) along with an **answers section** to help learners self-assess.
- **Flashcards:** Produces concise Q/A pairs for **quick memorization and revision**, simulating real flashcards used in study practices.

The output layer ensures that learners receive not just raw responses, but structured educational content that can directly improve study efficiency.

4. Setup Instructions

Prerequisites:

- I. Python 3.9 or later
- II. pip (Python package installer) and virtual environment tools (optional but recommended)
- III. Hugging Face Transformers and PyTorch for model integration
- IV. IBM Watsonx Granite access (if using the official Granite API; for local demo, you can use Hugging Face model hub)
- V. Stable internet connection to download models and dependencies.

5. Project Workflow

1. **User Input** → Student/teacher enters a query in the Gradio interface.
2. **Backend Processing** → Python functions prepare the prompt and send it to the AI model.
3. **AI Model (Granite LLM)** → Understands query, generates explanations, quizzes, or flashcards.
4. **Post-processing** → Output is cleaned and structured.
5. **User Output** → Results displayed in Gradio (concepts, quizzes, flashcards).

1. Frontend (Gradio UI)

- Built with Gradio Blocks + Tabs for structured interaction.
- Three main modules:
 - Concept Explanation – Detailed topic explanations.
 - Quiz Generator – 5 quiz questions + separate answers.
 - Flashcards – Quick Q/A style revision.

- Real-time interaction without page refresh.

2. Backend (Python + Transformers)

- Written in Python, powered by Hugging Face Transformers.
- Functions: `concept_explanation()`, `quiz_generator()`, `flashcard_generator()`.
- Handles input preprocessing, LLM prompts, tokenization, and output formatting.
- Optimized for CPU/GPU execution.

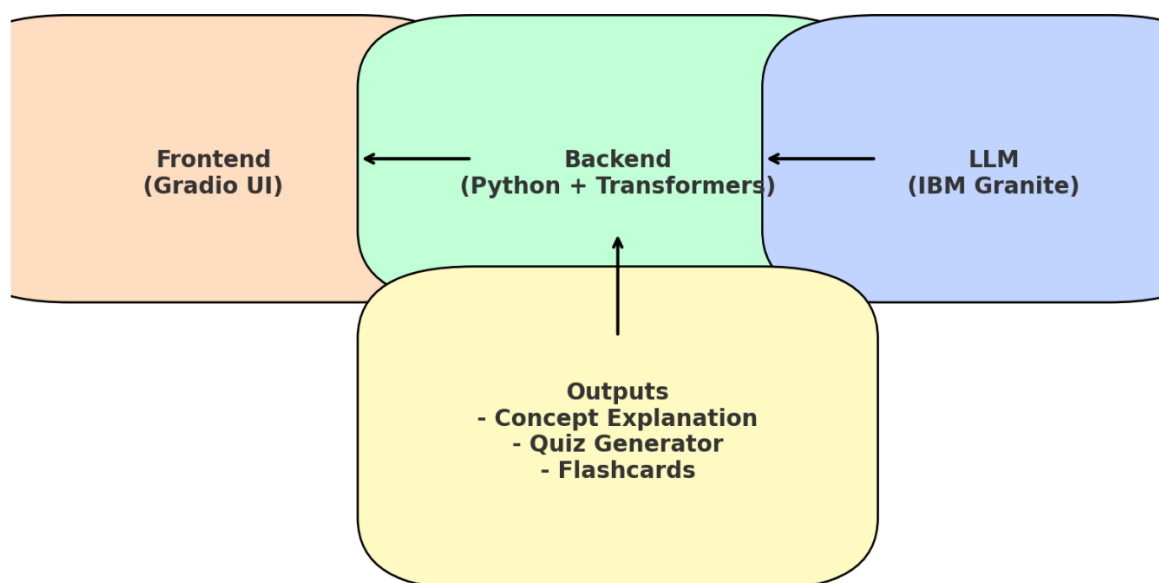
3. LLM Integration (IBM Granite 3.2B Instruct)

- Core AI model for natural language understanding & generation.
- Produces structured outputs (explanations, MCQs, T/F, descriptive answers).
- Guided by prompt engineering for accuracy and consistency.

4. Output Modules

- Concept Explanation – Simple explanations with examples.
- Quiz Generator – 5 diverse questions + answers.
- Flashcards – Short Q/A cards for revision.

Architecture of Educational AI Assistant



6. Authentication & User Interface

Authentication:

- For this demo version, the system runs in an **open environment** without login requirements.
- In a secure deployment, authentication can be integrated using:
 - **Token-based Authentication (JWT or API Keys)** – to protect API endpoints.
 - **OAuth2 with IBM Cloud Credentials** – for enterprise-level access control.
 - **Role-based Access** – different roles (e.g., *student*, *teacher*, *admin*) to control permissions.
 - **Planned Enhancements** – user session management, activity logging, and personalized history tracking.

User Interface:

- The interface is designed to be **minimalist, clear, and highly accessible** for non-technical users (students & educators).
- **Features include:**
 - **Sidebar Navigation** – easy movement across modules.
 - **Tabbed Layouts** – separate tabs for Concept Explanation, Quiz Generator, and Flashcards.
 - **Real-Time Interaction** – user queries processed instantly without page refresh.
 - **Visual Enhancements** – custom theme with engaging red-green gradients.
 - **Output Download Support (Planned)** – future support for exporting quizzes, explanations, or flashcards as **PDFs** for offline use.
 - **Help Texts & Guided Flows** – ensures smooth usage for first-time users.

7. Testing

- **Unit Testing:** Verified core functions (concept explanation, quiz generator, flashcard generator) for correct input/output handling.
- **Model Testing:** Checked IBM Granite responses for accuracy, structure, and consistency.
- **Manual Testing:** Tested Gradio UI tabs for real-time responses, usability, and theme design.
- **Edge Cases:** Handled empty inputs, long prompts, missing answers, and both CPU/GPU execution.
- **Integration:** Ensured smooth flow from frontend → backend → LLM → output modules.

8. Screen shorts

Educational AI Assistant

Concept Explanation **Quiz Generator** Flashcards

Enter a topic

gen ai

Generate Quiz

Quiz Questions

1. Multiple Choice: Which of the following is NOT a primary use case for generative AI in content creation?
 - a) Text generation
 - b) Image synthesis
 - c) Programming code
 - d) Traditional painting
2. True/False: Generative AI models can learn and improve from unstructured data, such as text or images, without explicit programming.
3. Short Answer: Describe a scenario where a generative AI model could be utilized to enhance human creativity in graphic design.
 - a) Generating color palettes based on mood
 - b) Automating the placement of elements in a layout
 - c) Creating original illustrations from scratch
4. Multiple Choice: Which of the following is a potential ethical concern when using generative AI for deepfakes or manipulated media?
 - a) Privacy invasion

Show desktop

- c) Programming code
d) Traditional painting

2. True/False: Generative AI models can learn and improve from unstructured data, such as text or images, without explicit programming.

3. Short Answer: Describe a scenario where a generative AI model could be utilized to enhance human creativity in graphic design.

- a) Generating color palettes based on mood
b) Automating the placement of elements in a layout
c) Creating original illustrations from scratch




4. Multiple Choice: Which of the following is a potential ethical concern when using generative AI for deepfakes or manipulated media?

- a) Privacy invasion
b) Misinformation propagation
c) Job displacement in creative industries
d) All of the above

5. True/False: Generative AI models can effectively generate high-quality, copyright-free music compositions, given enough training data and parameters.

Answers

1. c) Programming code
2. False: While generative AI models can learn from structured data, they generally require explicit programming for tasks like generating programming code.
3. c) Creating original illustrations from scratch: A generative AI model could be trained to generate unique illustrations based on a designer's style or brief, helping them explore new creative directions without starting from scratch.
4. d) All of the above: Ethical concerns related to generative AI in deepfakes or manipulated media include privacy invasion, misinformation propagation, and potential job displacement in creative industries.
5. False: While generative AI can create music, the quality and copyright-free status depend on the training data and models' capabilities. Current AI-generated music often involves copyrighted materials, and achieving truly original, copyright-free music with high quality is a challenging research area.

Use via API  · Built with Gradio  · Settings 

Educational AI Assistant

Concept Explanation Quiz Generator **Flashcards**

Enter a topic

operating system

Generate Flashcards

Flashcards (Q/A pairs)

1.
Q: What is the primary function of an operating system?
A: The primary function of an operating system (OS) is to manage computer hardware and software resources, providing common services for computer programs. It acts as an intermediary between the user and the computer hardware, ensuring efficient and effective use of system resources.
2.
Q: Which key components are managed by the operating system?
A: The operating system manages key components such as:
- Process management: Scheduling, prioritization, and synchronization of processes.
- Memory management: Allocation, deallocation, and protection of memory.
- File system management: Creation, organization, and access to files and directories.
- Device management: Communication and control of input/output devices.
- Security management: Authentication, authorization, and encryption of data.
- 3.

Show desktop

9. Known Issues

1. Response Generation Delay

- When using large topics or long prompts, the AI response can take several seconds to generate, especially on CPU-only systems.
- Reason: The model (granite-3.2-2b-instruct) is resource-intensive and performs best on a GPU.

2. Incomplete or Repetitive Responses

- Sometimes, explanations or flashcards may be cut off or contain repeated sentences.

- Reason: The `max_length` parameter limits output tokens; large or complex prompts may exceed this limit.

3. Tokenization Errors

- Occasionally, some special characters or very long inputs may cause tokenization issues.
- Reason: The tokenizer may not handle unusual symbols or extremely long text well.

4. Quiz Answer Mismatch

- If the AI does not format the “ANSWERS:” section correctly, the app may fail to split questions and answers.
- Workaround: Manually adjust or verify the output if the split fails.

5. UI Display Limitations

- Very long text in the output boxes (especially explanations or quiz questions) may not display fully in Gradio Textboxes.
- Workaround: Scroll within the textbox or reduce the `max_length` of output for smoother display.

6. Model Download & Memory Issues

- On first run, the model must be downloaded (~4–6 GB), which can fail on slow networks.
- Running the model on systems with <8GB GPU memory may result in CUDA out-of-memory errors.
- Workaround: Use smaller models like `distilgpt2` for testing.

10. Advantages & Disadvantages

Advantages:

- AI-generated explanations, quizzes, and flashcards with real-time responses.
- Simple and intuitive UI using Gradio for easy interaction.
- Supports concept explanation, quiz generation, and flashcards in one platform.
- Customizable themes enhance user experience.

Disadvantages:

- Requires internet for model download or API use.
- No persistent storage, so sessions or generated content are not saved.
- Quiz evaluation feature is not implemented; users must check answers manually.
- Resource-intensive, may be slow on CPU or low-memory systems.

11.Conclusion

The Educational AI Assistant is a versatile tool designed to aid learning by providing concept explanations, quiz generation, and flashcards in a single platform. It leverages AI to deliver real-time, detailed responses and simplifies studying with an intuitive Gradio-based interface. While it currently requires internet access and lacks persistent storage or automatic quiz evaluation, it demonstrates the potential of AI to enhance education. With future enhancements like interactive quizzes, multilingual support, and progress tracking, this tool can become a comprehensive learning assistant for students and educators.

12.Future Enhancements

1. **Support for Multiple Languages:** Enable explanations, quizzes, and flashcards in languages other than English.
2. **Interactive Quiz Mode:** Allow users to **answer quizzes within the app** and get instant feedback.
3. **Audio Output:** Convert explanations and flashcards into **speech** for auditory learning.
4. **Improved Flashcard Formatting:** Include images or diagrams in flashcards for better understanding.
5. **User Profile & Progress Tracking:** Save user history, track learning progress, and personalize content.
6. **Smaller, Faster Model Options:** Allow lightweight models for users with limited hardware or slow internet.