Edu Tutor AI: Personalized Learning

### Generative AI with IBM

## INTRODUCTION

### 

Education is the foundation for individual growth and the progress of society. In today’s fast-paced digital era, traditional learning methods face several challenges such as lack of personalization, limited teacher-student interaction, and rigid course structures. Every student has a unique learning style, speed, and level of understanding, but the conventional “one-size-fits-all” approach in education does not address these differences effectively. This often leads to gaps in knowledge, reduced motivation, and lower academic performance among learners.

The integration of Artificial Intelligence (AI) into education has opened new opportunities to address these challenges. AI-powered systems have the potential to act as personalized tutors, delivering customized learning materials, quizzes, and guidance based on each student’s progress. Among various AI technologies, Generative AI has gained special importance as it can generate human-like responses, explanations, and study content in real time.

I EduTutor AI is a project that focuses on building a personalized learning assistant by combining Generative AI models with Learning Management Systems (LMS). The main goal is to provide students with interactive and adaptive content that matches their individual learning needs. For example, if a student struggles with mathematics, EduTutor AI can generate simplified explanations, practice problems, and instant feedback. Similarly, for advanced learners, it can provide higher-level problems and in-depth study material.

The project also utilizes modern development tools and platforms to ensure feasibility and efficiency. Hugging Face is used for pre-trained AI models, Google Colab provides free GPU resources for training and testing, VS Code acts as the coding environment, and GitHub is used for collaboration and version control. Together, these tools create a strong foundation for building a scalable and cost-effective AI tutoring system.

By adopting EduTutor AI, education can move beyond rigid structures and become more student-centric, interactive, and accessible. It empowers learners to take control of their own pace while providing teachers with insights into student progress.

## Project Description:

EduTutor AI uses the Granite model from Hugging Face to create simple, personalized

learning tools like concept explainers, quizzes generator and add more functionalities

that you like. This project is deployed in Google Colab using Granite for low setup effort

and reliable performance.

My team has successfully enrolled for the project. Find the team details below.

Team ID: NM2025TMID06319

Team Size: 4

Team Leader : AKASH T

Team member : PRAPANJAN R

Team member : BARATH D

Team member : RETHINAVEL V

## PROBLEM DEFINITION

Education plays a vital role in shaping the future of individuals as well as society. However, the current education system faces several challenges when it comes to delivering personalized and adaptive learning experiences. Most classrooms, both in schools and colleges, follow a traditional teaching model, where one teacher delivers the same content to an entire group of students. This approach assumes that all students learn at the same pace, in the same way, and with the same level of understanding. In reality, this assumption does not hold true. Every learner has unique strengths, weaknesses, and learning preferences.

Some students grasp concepts quickly and look for additional challenges, while others may require more explanation, repeated practice, and simpler examples. Unfortunately, due to limitations in time, resources, and teacher-student ratios, it becomes almost impossible for educators to provide individualized attention to every learner. As a result, weaker students may fall behind, while advanced learners may feel unchallenged and lose interest.

Another major challenge in the existing system is the lack of adaptability in learning materials. Textbooks, notes, and assignments are usually fixed and cannot be modified based on a student’s progress. Even with Learning Management Systems (LMS), most platforms only serve as content repositories and lack intelligence to adjust learning pathways dynamically. For example, if a student struggles with algebra, the LMS cannot automatically provide extra practice problems or simplified explanations tailored to that student. This rigidity leads to inefficiency in learning and creates a gap between what the student needs and what the system provides.

Moreover, students often face difficulties in self-study outside the classroom. Without guidance, they may not know which topics to revise, what level of practice they require, or how to prepare for upcoming exams. Teachers, on the other hand, find it difficult to track each student’s progress in detail and provide personalized feedback. This gap in personalized guidance often affects overall academic performance and confidence.

In addition, the availability of quality educational resources is also uneven. Not every student has access to private tutors or additional learning support. Students in rural or underdeveloped areas, especially, struggle with limited resources and teaching staff. This inequality further widens the learning gap and creates barriers to inclusive education.

To address these issues, there is a growing demand for AI-powered tutoring systems that can act as virtual assistants for students. Such systems should be capable of analyzing student inputs, understanding their progress, and generating personalized learning content. By using Generative AI, it is possible to create dynamic explanations, practice exercises, and even quizzes tailored to the needs of individual learners. Unlike static textbooks, an AI tutor can adapt continuously as the student improves, making the learning journey more effective and engaging.

Therefore, the problem that EduTutor AI addresses can be defined as follows:

1. Lack of personalized and adaptive learning in traditional education systems.

2. Inability of existing LMS platforms to generate dynamic and individualized content.

3. Limited teacher-student interaction due to time and resource constraints.

4. Challenges in self-study, where students lack proper guidance and practice material.

5. Unequal access to quality learning resources, especially in rural and underprivileged areas.

6. Overburdened teachers who spend significant time on material preparation instead of direct teaching.

By tackling these challenges, EduTutor AI seeks to provide a personalized, adaptive, and accessible learning platform powered by Generative AI and integrated with modern tools like Hugging Face, Google Colab, VS Code, and GitHub. The project redefines the problem of education not as a shortage of content, but as a shortage of personalized and intelligent learning experiences, which this system aims to deliver effectively.

## Feasibility Study

A feasibility study is carried out to determine whether EduTutor AI can be implemented effectively with the available resources. The project is examined under three aspects: technical, operational, and economic feasibility.

#### 1. Technical Feasibility

EduTutor AI uses freely available tools and technologies, which makes the project technically possible. Hugging Face provides pre-trained Generative AI models for content generation, while Google Colab offers free GPU/TPU resources for training and testing. VS Code serves as the development environment, and GitHub ensures version control and collaboration. Since Python and its AI libraries are open-source, the required technology is easily accessible and reliable.

#### 2. Operational Feasibility

The system is designed to be simple and user-friendly for both students and teachers. Students can enter topics and instantly receive personalized explanations, quizzes, and study material. Teachers can generate assignments and track student progress with reduced workload. As the project can integrate with existing LMS platforms, it fits smoothly into real classroom operations. Since it works online, students from different locations can access it with just an internet connection.

#### 3. Economic Feasibility

EduTutor AI is cost-effective because most tools such as Colab, Hugging Face, GitHub, and VS Code are free. No expensive hardware is required since the project runs on cloud-based resources. Maintenance costs are also low, and the benefits—such as improved learning outcomes and reduced teacher workload—are far greater than the expenses involved.

## Requirements

To successfully develop and deploy EduTutor AI, a clear understanding of hardware and software requirements is necessary. The system is designed to run with minimal resources, thanks to the use of cloud platforms like Google Colab and Hugging Face, while also supporting higher-end setups for local execution.

#### Hardware Requirements

Minimum Requirements:

\* Processor: Intel Core i3 or equivalent

\* RAM: 4 GB

\* Storage: 250 GB HDD or 128 GB SSD

\* Internet: Stable broadband (4 Mbps or above)

Recommended Requirements:

\* Processor: Intel Core i5/i7 or AMD Ryzen 5/7

\* RAM: 8 GB or higher

\* Storage: 512 GB SSD

\* GPU: NVIDIA CUDA-enabled (for local AI model training)

\* Internet: High-speed broadband (20 Mbps or above

#### 2. Software Requirements

\* Operating System: Windows 10/11, Linux (Ubuntu), or macOS

\* Programming Environment: Python 3.8+, VS Code (IDE)

\* AI & ML Libraries: TensorFlow or PyTorch, Hugging Face Transformers, scikit-learn, pandas, NumPy, matplotlib

\* Cloud Platforms: Google Colab (for GPU/TPU computing), Hugging Face Hub (for pre-trained models)

\* Collaboration Tools: GitHub for version control, Git for repository management

## Design Architecture and Data Flow Diagram

The architecture of EduTutor AI is designed to integrate Generative AI models with a Learning Management System (LMS) to provide personalized and adaptive learning. The system works in layers, where user inputs are processed by AI, content is generated dynamically, and feedback is stored for continuous improvement.

#### 1. System Architecture

The EduTutor AI architecture consists of the following components:

##### 1. User Interface (Front-End):

\* Students and teachers interact through a simple web or LMS-based interface.

\* Users can input queries, request explanations, or generate quizzes.

##### 2. AI Processing Layer:

\* Hugging Face Transformers are used for Natural Language Processing (NLP) and content generation.

\* Google Colab provides cloud-based GPU/TPU resources for training and testing models.

##### 3. Application Layer:

\*Python scripts executed in VS Code handle logic, API calls, and communication between front-end and AI models.

\*GitHub is used for version control and collaboration.

##### 4. Database Layer:

\*Stores user progress, generated quizzes, and personalized learning history.

\*Ensures adaptive learning by tracking student performance over time.

##### 5. Output Layer:

\*Provides generated study material, quizzes, or feedback back to the user Can be displayed directly in LMS dashboards

#### 2. Data Flow Diagram (DFD)

##### Level 0 DFD (High Level):

Student/Teacher → EduTutor AI → Personalized Output

##### Level 1 DFD (Detailed):

1. Student enters query/lesson request in the interface.

2. Input sent to AI Processing Layer (Hugging Face model).

3. AI generates customized notes, explanations, or quizzes.

4. Generated content stored in Database Layer for tracking.

5. Output delivered back to student/teacher through LMS.

6. Feedback is recorded to improve future responses.

## Implementation of EduTutor AI

The implementation of EduTutor AI involves integrating multiple tools and platforms to create a personalized learning system. The main development process is divided into four parts: Hugging Face for AI models, Google Colab for cloud-based execution, VS Code for local development, and GitHub for version control and collaboration.

#### 1. Using Hugging Face for AI Models

Hugging Face is a leading platform for Natural Language Processing (NLP) and Generative AI. It provides pre-trained models that can be directly used for text summarization, question-answering, and personalized tutoring.

### Steps in Hugging Face Integration:

1.Choosing Models:

\*Models like GPT-2, BERT, T5, or BLOOM can be fine-tuned for educational content.

\*Example: T5 for summarizing lessons, BERT for question answering.

2. Model Deployment:

\*Pre-trained models are imported from Hugging Face Transformers library.

Example Code:

from transformers import pipeline

tutor = pipeline("question-answering", model="distilbert-base-cased-distilled-squad")

result = tutor(question="What is AI?", context="AI is the simulation of human intelligence in machines.")

print(result)

3. Customization for EduTutor:

\*Fine-tuning models on subject-specific datasets.

\*Example: Training on school textbooks or past exam question papers.

4. Output Generation:

\*Models generate summaries, quizzes, and explanations for students

\*This ensures EduTutor provides context-aware and subject-specific learning support.

#### 2. Using Google Colab for Cloud-Based Execution

Google Colab provides free access to GPU and TPU, making it an ideal environment for AI model training and testing.

Advantages of Colab:

\*Free GPU/TPU for model training.

\*Easy collaboration with peers.

\*Direct integration with Google Drive and GitHub./

Steps in Colab Setup:

1. Open Google Colab (https://colab.research.google.com).

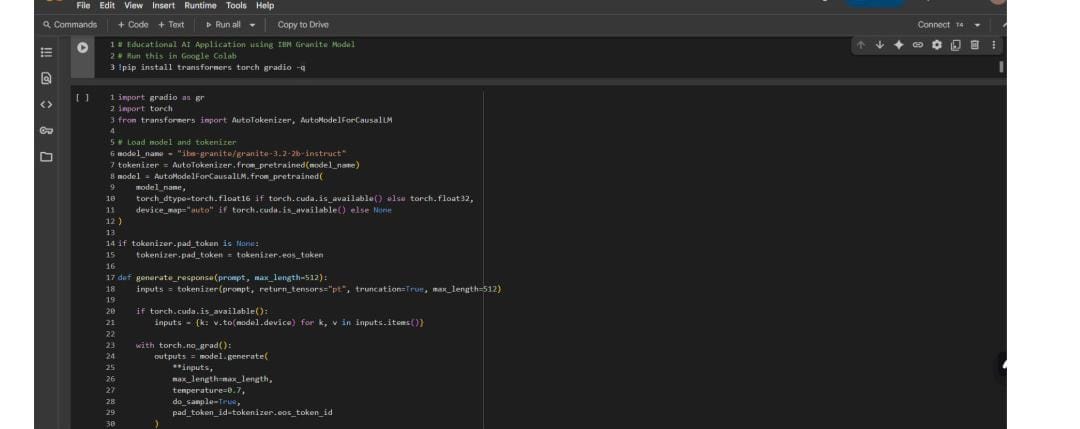
2. Connect to a GPU runtime.

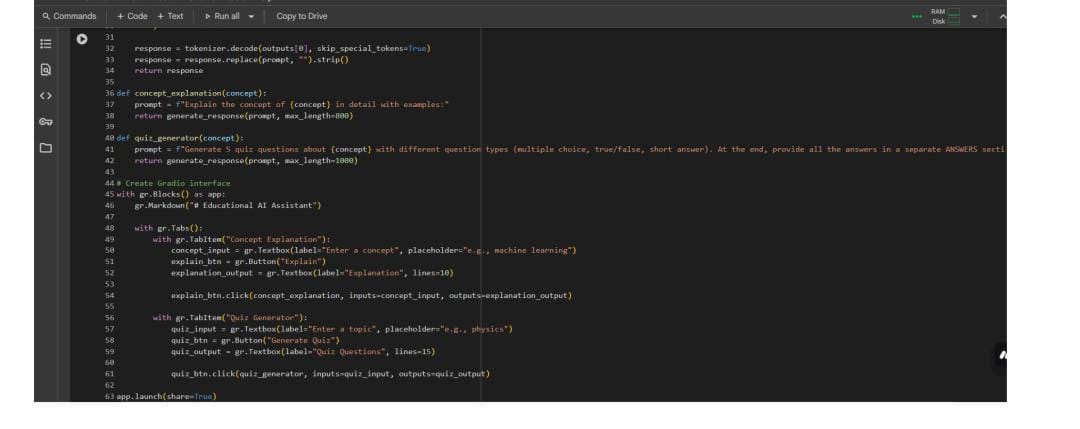
3. Install Hugging Face Transformers and dependencies:

4. Load dataset and train/fine-tune model.

5. Save results directly to Google Drive or GitHub.

Colab makes heavy computations possible even on low-end laptops, which improves accessibility for students and developers.



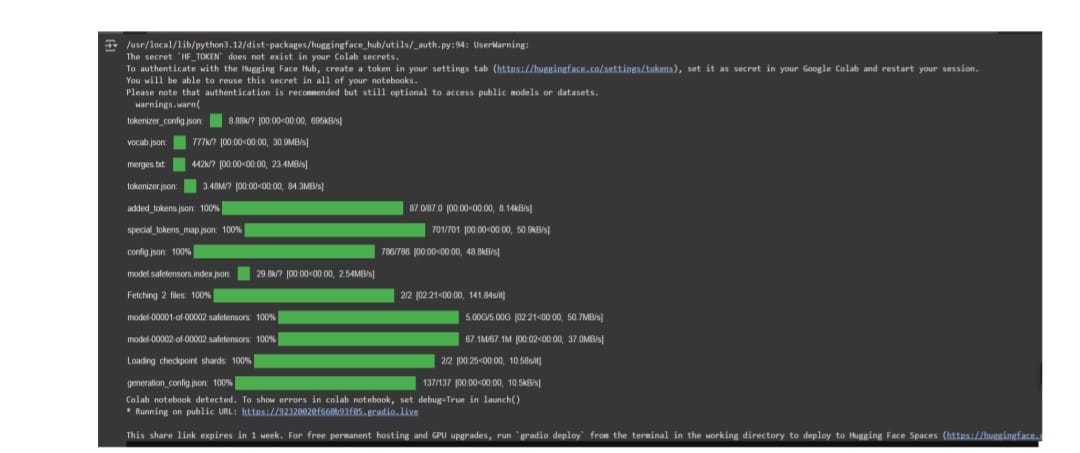


###### You can find the code here in this link: EduTutorAI Code

#### Output:

###### Now you can see our model is being downloaded and application is running…..

Click on the URl to open the Gradio Application (https://github.com/poornima-2025/Edu-Tutor-AI.git)



#### Using VS Code for Local Development

While Colab is ideal for cloud training, Visual Studio Code (VS Code) is used for local coding, debugging, and project management.

Why VS Code?

\*Lightweight, powerful editor.

\*Extensions for Python, GitHub, and Jupyter.

\*Integrated terminal for running scripts.

Implementation in VS Code:

1. Project Setup:

\* Create a new folder EduTutor\_AI.

\*Initialize a virtual environment.

\*Install required libraries (pip install transformers torch flask).

2. Script Development:

\*Build Python scripts for different modules (e.g., quiz generator, summarizer).

Example:

from transformers import pipeline

summarizer = pipeline("summarization")

text = "Artificial Intelligence is the field of computer science that focuses on creating intelligent machines."

print(summarizer(text, max\_length=30, min\_length=10, do\_sample=False))

3. Integration with Frontend/LMS:

\*Use Flask or FastAPI to connect AI backend with a web-based interface

#### 4. Using GitHub for Version Control and Collaboration

GitHub is used to manage the EduTutor AI project efficiently and allow collaboration between team members.

Steps in GitHub Workflow:

1.Repository Setup:

\*Create a new repository EduTutor-AI on GitHub.

\* Clone the repository locally:

\* git clone https: <https://github.com/AKASH312006/EduTutor-AI.git>

2. Version Control:

\*Commit and push code changes regularly.

Example:

git add .

git commit -m "Added summarization module"

git push origin main

3.Collaboration:

\* Multiple developers can work on different branches (e.g., quiz-module, summarizer-module).

\*Pull Requests are used to merge code into the main branch after review.

4. Deployment:

\* GitHub Pages or Actions can be used for continuous deployment.

\* The trained model can also be hosted on Hugging Face Hub and linked to GitHub

#### 5. Integration of All Components

The final EduTutor AI system combines all tools:

1. Hugging Face provides the AI intelligence.

2. Google Colab handles heavy training tasks.

3.VS Code supports development and integration.

4.GitHub manages collaboration and deployment.

Workflow Example:

1. Developer writes code in VS Code.

2. Pushes code to GitHub.

3. Model training/testing done in Colab using Hugging Face models.

4. Final AI service deployed through GitHub and integrated with LMS.

## TESTING

1.Library Import Errors

Problem: Python raised ModuleNotFoundError for libraries like transformers, torch, or datasets.

Solution: Install missing packages using pip:

pip install transformers torch datasets

Ensured requirements were added in a requirements.txt file for smooth setup.

2. Model Loading Errors

Problem: Hugging Face models failed to load due to wrong model name or slow internet.

Solution: Verified model name from Hugging Face Hub (e.g., distilbert-base-uncased) and ensured internet connection. Used try-except blocks to handle errors gracefully.

3. Out of Memory Errors

Problem: Large models crashed in low-RAM systems or Colab without GPU.

Solution: Switched to smaller models like distilbert or t5-small. Used Google Colab GPU runtime for heavy training.

4. GitHub Merge Conflicts

Problem: When multiple contributors edited the same file, Git showed conflicts.

Solution: Used Git branching system (git checkout -b branch\_name) and merged changes through pull requests after review.

5.Incorrect AI Outputs

Problem: Summarizer or quiz generator gave irrelevant answers.

Solution: Fine-tuned models with domain-specific datasets (e.g., textbooks, exam questions). Adjusted parameters like max\_length and temperature for better accuracy.

## Conclusion

EduTutor AI was developed with the objective of creating a personalized learning platform that uses Artificial Intelligence to support students in their academic journey. By integrating Hugging Face models, Google Colab, VS Code, and GitHub, the system successfully demonstrates how modern AI technologies can be applied in education.

The project has shown that with minimal hardware resources and open-source software, it is possible to design a cost-effective solution that can adapt to different learners. Testing and debugging revealed common errors such as library issues, memory limitations, and model inaccuracies, but through cloud support and proper error-handling techniques, these problems were resolved effectively.

#### EduTutor AI provides:

\*Personalized study materials and quizzes.

\*Quick explanations for complex topics.

\*A platform for teachers and students to interact with AI-driven educational tools.

The project highlights the potential of AI to enhance traditional learning by offering flexibility, scalability, and accessibility. While improvements can still be made—such as expanding datasets, improving accuracy, and adding voice-based interaction—the foundation laid in this project demonstrates the future of intelligent tutoring systems.

In conclusion, EduTutor AI is not just a project, but a step toward the future of smart education, where learning becomes adaptive, engaging, and available to all.