**EduTutor-AI: Intelligent Educational Assistant**

**Project Documentation**

**Team ID: NM2025TMID03685**

**Team Size: 4**

**Team Leader: DHARSHINI M**

**Team member: DHIKSHA P**

**Team member: DILLIRAJI G**

**Team member: DEVASHRI R**

**Category:** AI-Powered Educational Application Development

**Required Skills: Python, Hugging Face Transformers, PyTorch, Gradio**

**Project Overview**

**EduTutor-AI** is a **sophisticated learning assistant** designed to provide students with an interactive and **tailored educational experience**. Powered by **IBM Granite LLM models**, the system supports learners by offering detailed **explanations** of **complex topics** and **automatically** creating practice **quizzes** **for** **assessment** and **revision**.

Platform includes:

* **Comprehensive Concept Explanation:** **Delivers well-structured** and thorough explanations of **academic topics**, supported by **relevant** **examples**.
* **Quiz Generation Module:** Produces a wide range of question types including **MCQs, True/False**, and **short-answer** questions, complete with solutions.
* **Conversational Learning Support:** Engages learners through AI-driven dialogue, offering personalized responses and guidance.
* **User-Friendly Interface:** Built with Gradio, ensuring a smooth, accessible, and intuitive user experience.

Use Case Scenarios:

**Concept Learning**

* **Action:** The learner enters a topic (e.g., Machine Learning).
* **Result:** EduTutor-AI responds with a detailed explanation and examples to improve understanding.

**Quiz Practice**

* **Action:** The learner requests quiz questions on a particular subject (e.g., Physics).
* **Result:** The system produces a variety of questions with their correct answers.

**Interactive Assistance**

* **Action:** The learner asks subject-related questions through the interface.
* **Result:** EduTutor-AI delivers detailed, context-aware, and customized responses.

**Technical Architecture**

**Prerequisites**

* **Python (3.8 or later)**
* **Hugging Face Transformers**
* **PyTorch**
* **Gradio** (for front-end interface)
* **GPU** (recommended for faster performance)
* **Internet connectivity** (for model downloads)

**Project Structure**

* **edututor.py:** Main application file
* **models/:** Directory containing model checkpoints
* **static/:** Optional folder for styling and image assets

**Project Setup & System Design**

**Model & Library Selection**

* **IBM Granite 3.2 2B Instruct Model**
* **Hugging Face Transformers** & **PyTorch** for model inference
* **Gradio** for creating an interactive interface

**System Flow**

Input → AI Model Inference → Processed Output → Display on UI

**Core Functionalities**

* **generate\_response():** Performs core AI inference and returns results
* **concept\_explanation():** Generates detailed topic explanations with examples
* **quiz\_generator():** Produces quizzes with multiple question types and answers

**Data Handling & Logic:**

* **User inputs** are processed through the model and returned in a well-**formatted output.**
* Gradio supports **session-based interactions**, and **lightweight storage** can be implemented to track quiz history or learner progress.

**Frontend Development (Gradio)**

* Separate **tabs** for Concept Explanation and Quiz Generation
* **Text input fields** for student queries
* **Interactive buttons** to trigger actions
* **Output areas** displaying explanations or quiz results

**Testing & Integration**

* **Local Execution:** Run edututor.py to launch the Gradio interface
* **Test Scenarios:**
* Input concept → **receive explanation**
* Input topic → **generate quiz**
* **Debugging focuses on refining prompts** and improving UI interaction

**Deployment**

* **Containerization:** Deploy using Docker with a Python base image
* **Hosting:** Cloud deployment options include AWS, Azure, or Hugging Face Spaces
* **Security:** Enforce HTTPS and secure API key usage
* **Monitoring:** Track system performance, errors, and usage analytics

**Documentation & Deliverables**

* **README File:** Setup instructions, usage details, API references
* **User Guide:** Screenshots and feature descriptions
* **Demo Video:** Step-by-step walkthrough of EduTutor-AI’s features

**EduTutor-AI provides an end-to-end intelligent learning assistant, enabling students to grasp complex concepts easily and practice with AI-generated quizzes, fostering a deeper and more engaging educational experience. In addition, it promotes personalized learning by adapting explanations and quizzes to different subject areas and complexity levels. It encourages self-paced study, allowing learners to revisit concepts and practice until mastery is achieved. The system also enhances critical thinking and problem-solving skills through diverse quiz formats, while its interactive design keeps students motivated and engaged. Furthermore, EduTutor-AI can serve as a valuable support tool for teachers, helping them create supplementary learning materials quickly and efficiently**

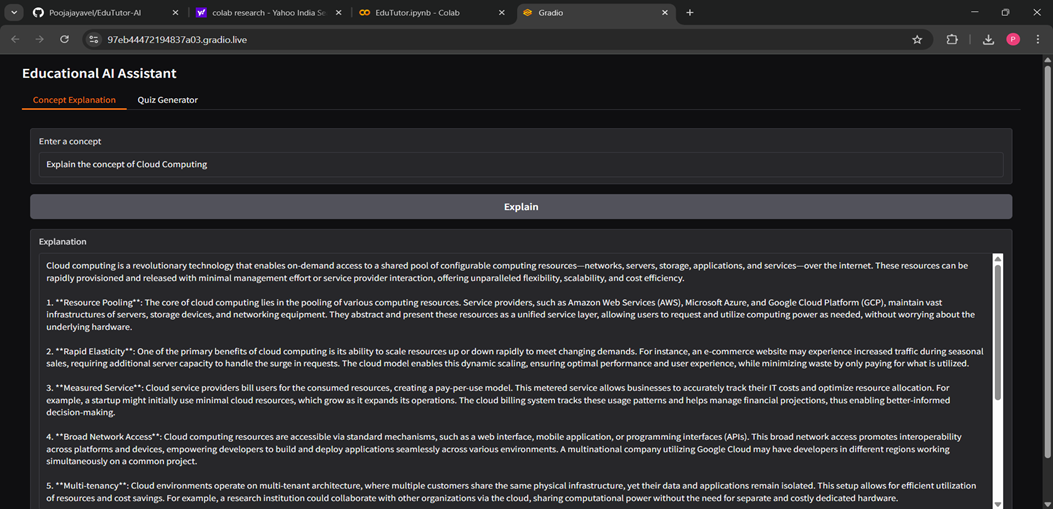
**Project Output**

**1. Concept Explanation Example**

**Input:** *Explain Cloud Computing*

**Output:**

**A detailed description covering key principles like Resource Pooling, Rapid Elasticity, Measured Service, Broad Network Access, and Multi-tenancy. It also explains On-Demand Self-Service for automatic resource provisioning and highlights cost efficiency achieved through pay-as-you-go pricing.**

****

**2. Quiz Generator Tab**

**Input:** Generate a quiz on Python Programming

**Output:** The system generates **multiple question** **types** (MCQs, True/False, Short answers). For Example

* **Multiple Choice:**

Which of the following is NOT a valid Python data type?

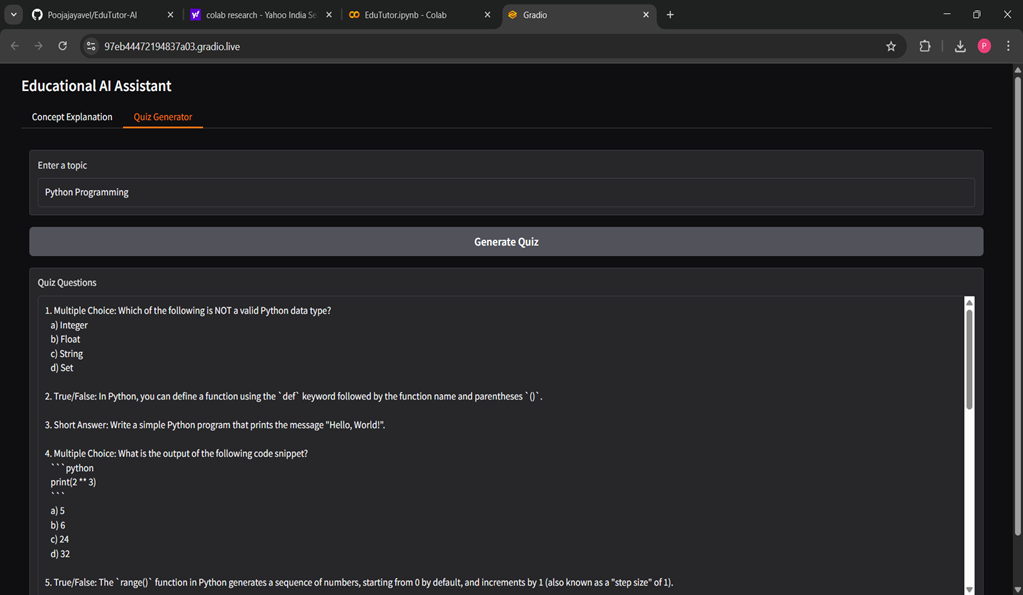
a) Integer b) Float c) String d) Set

**Answer: d) Set**

* **True/False:**

In Python, you can define a function using the def keyword.

**Answer: True**

****

**Conclusion**

Developing **EduTutor-AI** has been an **invaluable experience**, combining **AI** **model integration**, **Python programming**, and **UI development** into a single project. Leveraging **IBM Granite LLM** and **Hugging Face** **Transformers**, the system successfully provides learners with detailed explanations and **diverse quiz generation**, significantly **enhancing** **the learning process.**

This project demonstrates how **artificial intelligence** can be effectively applied to create **engaging, interactive, and personalized educational tools** that benefit both students and educators. **EduTutor**-**AI** ultimately contributes to the vision of **AI-driven learning, offering a scalable, reliable,** and **user-friendly** solution to support **knowledge acquisition** and **skill development.**