# EduTutor AI – Project Documentation

## 1. Introduction

Project Title: EduTutor AI
Team Leader: Naresh R
Team Members:
- Muthukumar T
- Sanjay K
- Velmurugan P
- Janarthanan S

EduTutor AI is an AI-powered educational assistant designed to simplify learning through **concept explanations** and **quiz generation**.

## 2. Project Overview

### Purpose

EduTutor AI leverages **large language models (LLMs)** to help students and educators by providing detailed explanations of concepts and automatically generating quizzes for self-assessment.

### Features

- **Concept Explanation**: Provides detailed explanations of academic topics with examples.
- **Quiz Generator**: Creates quizzes with multiple types of questions (MCQs, True/False, Short Answer) along with an answer key.
- **Interactive UI**: Uses Gradio for a simple web-based interface.
- **AI-Powered Responses**: Built on IBM Granite model (`granite-3.2-2b-instruct`) for intelligent responses.

## 3. System Architecture

- **Frontend**: Built with **Gradio UI**, featuring tab-based navigation.
- **Backend**: Python-based, integrates Hugging Face Transformers for AI model inference.
- **LLM Integration**: Uses **IBM Granite 3.2 Instruct model** for generating explanations and quizzes.
- **Deployment**: Runs locally with GPU/CPU support and allows public sharing via Gradio.

## 4. Setup Instructions

### Prerequisites

- Python 3.9+

- pip package manager
- Internet connection (to fetch the model)

### Installation & Execution

```bash
# Clone the repository
git clone
cd edututor-ai

# Install dependencies
pip install -r requirements.txt

# Run the app
python edututorai.py
```

After running, access the **EduTutor AI dashboard** from the provided Gradio link.

## 5. Folder Structure

```
edututor-ai/
■■■ edututorai.py # Main application file
■■■ requirements.txt # Python dependencies
```

## 6. Running the Application

1. Start the application with `python edututorai.py`.
2. Gradio will generate a **local and public link**.
3. Navigate to the dashboard.
4. Use the **Concept Explanation** tab to get topic explanations.
5. Use the **Quiz Generator** tab to generate quizzes with answers.

## 7. API / Functionality

- `concept_explanation(concept)` → Explains a given concept in detail with examples.
- `quiz_generator(concept)` → Generates 5 quiz questions of different types and provides an answer key.

## 8. Authentication

Currently, the application does **not require authentication**.
(Future versions may add **role-based login** for students and teachers).

## 9. User Interface

- **Tabs** for switching between Concept Explanation & Quiz Generator.
- **Textbox Inputs** for entering a concept/topic.
- **Output Area** displaying explanations or generated quizzes.

## 10. Testing

- Unit tested with different concepts.
- Edge case handling: long prompts, uncommon topics.
- Verified model outputs for correctness.

## 11. Screenshots

(To be added after deployment UI screenshots are available)

## 12. Known Issues

- Requires stable internet for fetching model & running inference.
- Limited to English language.
- No offline model support yet.

## 13. Future Enhancements

- Voice-based tutoring.
- Multilingual support.
- Mobile app integration.
- Student progress tracking.
- Advanced reporting dashboard.