EduTutor AI: Personalized Learning with Generative AI and LMS Integration

# 1.Introduction

Project Title: EduTutor AI – Personalized Learning with Generative AI

**Team Members:**  
 - CHALLA RAGHAVENDRA-Full-stack Development AI development, coordination  
 - ULLIKANTI KHALEEL AHMAD – Backend DeveloperQuiz logic, prompt design  
 - SIRASANI JANARDHAN REDDY – UI/UX Designer Interface design using Gradio  
 - GAJAVALLI MANIDHAR– Tester/Validator App testing, user feedback integration

# 2.Project Overview

* Purpose:  
  EduTutor AI is designed to help students learn more effectively by using AI to summarize PDFs, generate quizzes, and track their academic progress interactively.
* Features:
  + Secure Login & Registration
  + Upload and summarize academic PDFs
  + AI-based tutor answering concept questions
  + Topic-based and document-based quizzes
  + Personalized Progress Tracker
  + Simple and interactive UI (Gradio)

# 3. Architecture

# \*Frontend: Developed using Gradio (Python-based UI library), served on Hugging Face Spaces. Acts like a React-like frontend for fast prototyping.

# \*Backend: Built using Python, with API-based integration to AI services (OpenAI, Transformers). Logic for summarization, quiz generation, login management, and progress tracking is handled in Python functions.

# \*Database: Uses Python dictionaries or JSON (can be extended to MongoDB) to store user details, quiz data, and progress metrics.

# 4. Setup Instructions

# Prerequisites:

# Python 3.8+

# Gradio

# Hugging Face Transformers

# OpenAI or similar LLM API key

# Installation:

# git clone <https://huggingface.co/spaces/Anoosha-12/EdututorAi> cd EdututorAi

# pip install -r requirements.txt python app.py

# 

# 5. Folder Structure

# Aap.py

# requirements.txt

# readme.md

# 6. Running the Application

# Frontend (Gradio):

# python app.py # Automatically starts UI with Gradio

# Backend:

# Included in the same Gradio interface. No separate server.

# 7. Authentication

| Endpoint | Method | Description | Parameters |
| --- | --- | --- | --- |
| /login | POST | Login with username & password | username, password |
| /register | POST | Register new user | username, password |
| /upload\_pdf | POST | Uploads a PDF | pdf\_file |
| /summarize\_pdf | GET | Generates summary from PDF |  |
| /generate\_quiz | POST | Creates quiz based on topic/pdf | topic or pdf\_text |
| /track\_progress | GET | Shows progress summary | user\_id |

# 8. Folder Structure

# Authentication is handled using a simple username-password system.

# Data is stored temporarily in memory or JSON (extendable to MongoDB).

# Sessions are not persistent unless implemented with a database.

# 9. User Interface

# Screenshot from project: ![UI Screenshot](attachment from earlier)

# Clean Gradio tabbed interface

# Sections: Register, Login, AI Tutor, PDF Quiz, Summary, Progress Tracker

# Login status shown dynamically

# Real-time quiz and summary interaction

# 10. Testing

# Manual testing done for each feature (login, upload, quiz, tracker)

# Suggested tools:

# Postman (for API)

# Pytest (for backend function testing)

# 11. Screenshots or Demo

# Hugging Face Live Demo: https://huggingface.co/spaces/Anoosha-12/EdututorAi

# Screenshots:

# Login screen (Thara – Student)

# AI Tutor

# 

# Quiz & Summary view

# 

# 11. Known Issues

# No database persistence for progress tracking (can be lost on restart)

# Quiz generation depends on context accuracy from LLM

# No admin dashboard for teacher role

# 12. Future Enhancements

* Add **MongoDB** or Firebase for real-time storage
* Build **Teacher/Admin Panel**
* Add **adaptive learning path**
* Export progress as PDF reports
* Notification/reminder system for users