
📄 Project Documentation – Edu Tutor AI

1. Introduction

Project Title: Edu Tutor AI – Personalized Learning with Generative AI and LMS Integration

Team Members:

- Team Leader : Asifa Samrin N A
- Team Member 1 : Ayesha Safeeka K S
- Team Member 2 : Saliha Thabasum E K
- Team Member 3 : Sameera Faslin V M

---

2. Project Overview

Purpose:

Edu Tutor AI is designed to transform education by providing a personalized learning experience powered by Generative AI and seamless integration with Learning Management Systems (LMS). It helps students learn at their own pace, receive adaptive study materials, and interact with an intelligent tutor that generates content and provides real-time support based on their progress and learning needs.

Features:

Conversational Tutor Interface

Key Point: Interactive AI-powered chatbot

Functionality: Students can ask questions, receive explanations, and interact conversationally for a more engaging learning experience.

Generative AI Content Creation

Key Point: Dynamic learning materials

Functionality: Automatically generates quizzes, study notes, summaries, and practice exercises tailored to individual learning levels.

Adaptive Learning Paths

Key Point: Personalized progression

Functionality: Tracks student performance and suggests customized learning paths to optimize knowledge retention and growth.

LMS Integration

Key Point: Seamless platform compatibility

Functionality: Synchronizes student data, course content, grades, and progress reports with popular LMS platforms like Moodle, Blackboard, and Canvas.

Progress Analytics and Reporting

Key Point: Insightful dashboards

Functionality: Provides real-time analytics on student performance, engagement, and knowledge gaps for educators to adjust instruction accordingly.

Multimodal Content Support

Key Point: Flexible learning materials

Functionality: Supports text, video, interactive simulations, and PDFs to accommodate diverse learning styles.

Feedback and Recommendation System

Key Point: Continuous improvement

Functionality: Collects student feedback and suggests next steps to improve understanding and engagement.

Assessment Automation

Key Point: Efficient evaluation

Functionality: Automatically generates and grades assignments, quizzes, and tests based on learning objectives.

---

3. Architecture

Frontend (React or Streamlit):

Intuitive web interface allowing students and teachers to interact with the AI tutor, view progress, and access learning materials.

Backend (FastAPI or Node.js):

Handles API requests for content generation, student query handling, data synchronization with LMS, and analytics.

Generative AI Integration (OpenAI GPT or IBM Watsonx Granite):

Generates educational content such as summaries, quizzes, and study tips in real time.

LMS Integration (Moodle, Blackboard, Canvas APIs):

Syncs user profiles, course structures, grades, and learning materials with the selected LMS.

Database (PostgreSQL or MongoDB):

Stores student profiles, progress history, generated content, and feedback data.

---

4. Setup Instructions

Python 3.9+ or Node.js environment

Install dependencies from requirements.txt or package.json

API keys for OpenAI GPT or IBM Watsonx Granite

LMS API credentials

Configure .env with credentials

Launch backend server

Launch frontend web interface

Connect to LMS for data synchronization

---

5. Running the Application

Start the backend API server

Launch the frontend dashboard

Register users and sync courses with LMS

Generate personalized study materials

Monitor progress and assessments in real-time

Download reports or share via LMS

---

6. API Documentation

POST /ask-tutor: Ask a study-related question and receive an AI-generated answer

POST /generate-content: Generate quizzes, notes, or study guides dynamically

POST /sync-lms: Synchronize student data and course progress with LMS

GET /progress-report: Retrieve detailed performance analytics

POST /submit-feedback: Collect student feedback on content or tutor responses

---

7. Authentication

JWT-based Authentication for secure access

OAuth2 with LMS platform

Role-based access (Student, Educator, Admin)

---

8. User Interface

Clean and interactive design:

Sidebar navigation for courses, tutor chat, assessments, and reports

Interactive AI tutor chat window

Visual dashboards showing learning progress

Easy-to-use forms for feedback and assessment submissions

---

9. Testing

Unit Testing: For content generation and API endpoints

API Testing: Postman and Swagger

Manual Testing: UI interactions, content sync with LMS, quiz generation

Edge Case Handling: Empty inputs, API key errors, large content files

---