# EduTutor AI Project Documentation

## 1. Introduction

Project Title: EduTutor AI – Personalized Education Platform Using IBM Watsonx & Granite LLM

Team ID: NM2025TMID00793

Team Members:  
- Maha Lakshmi G  
- Renuka Devi C  
-Priyanka L

## 2. Project Overview

Purpose:  
EduTutor AI is an AI-powered personalized education platform that revolutionizes the way students learn and educators assess progress. It provides dynamic quiz generation, student evaluation, Google Classroom integration, and real-time feedback—all powered by IBM Watsonx and Granite foundation models. Designed with modular architecture, this platform streamlines personalized education and enhances learning outcomes for students across academic levels.  
  
Features:  
- Dynamic Quiz Generation: AI-driven quizzes using Granite LLM.  
- Student Evaluation & Feedback: Real-time assessment and instant feedback.  
- Educator Dashboard: Insights, quiz history, and performance tracking.  
- Google Classroom Integration: Seamless synchronization with courses.  
- Adaptive Quizzing: Personalized difficulty based on diagnostic tests.

## 3. Use Case Scenarios

Scenario 1: Personalized Learning Experience  
Students sync courses via Google Classroom, receive AI-generated quizzes, and get instant feedback.  
  
Scenario 2: Educator Dashboard & Performance Insights  
Educators view quiz history, scores, last topics attempted, and insights via Pinecone vector database.  
  
Scenario 3: Diagnostic Testing and Adaptive Quizzing  
Students take a diagnostic test generated by IBM Watsonx; difficulty adapts accordingly.  
  
Scenario 4: Google Classroom Integration  
Seamless syncing of student data, subjects, and classes for auto quiz generation.

## 4. Architecture

- Frontend (React/Streamlit): Interactive UI with dashboards & quizzes.  
- Backend (FastAPI): RESTful APIs for quizzes, feedback, authentication.  
- LLM Integration (IBM Watsonx & Granite): Quiz generation & adaptive feedback.  
- Vector Database (Pinecone): Stores quiz data & performance insights.  
- Modular Architecture: Enables scalability & flexible updates.

## 5. Setup Instructions

Prerequisites:  
- Python 3.9+  
- FastAPI, Streamlit/React  
- IBM Watsonx & Pinecone API keys  
- Pandas, scikit-learn, matplotlib  
  
Installation Process:  
1. Clone repository  
2. Install dependencies from requirements.txt  
3. Configure API credentials in .env  
4. Run FastAPI backend  
5. Launch frontend  
6. Sync with Google Classroom

## 6. Folder Structure

- app/ – FastAPI backend logic  
- app/api/ – Routes for quizzes, feedback, diagnostics  
- ui/ – Frontend components (dashboard, quiz interfaces)  
- quiz\_generator.py – AI quiz generation functions  
- diagnostic\_engine.py – Diagnostic testing & adaptive quizzing  
- insight\_analyzer.py – Educator performance insights  
- classroom\_sync.py – Google Classroom integration  
- report\_generator.py – Creates progress reports

## 7. Running the Application

1. Start FastAPI backend  
2. Run frontend dashboard  
3. Log in as student/educator  
4. Sync Google Classroom  
5. Generate quizzes & view progress

## 8. API Documentation

- POST /quiz/generate – Generate quizzes  
- POST /quiz/submit – Submit answers for evaluation  
- GET /dashboard/insights – Fetch performance insights  
- POST /diagnostic/start – Start diagnostic test  
- POST /classroom/sync – Sync Google Classroom data

## 9. Authentication

- JWT-based authentication  
- OAuth2 with Google Classroom  
- Role-based access (student, educator, admin)

## 10. User Interface

- Student View: Quiz interface, instant feedback, progress tracking  
- Educator View: Dashboard with insights & analytics  
- Google Classroom Integration: One-click sync

## 11. Testing

- Unit Testing: For backend & quiz generation  
- API Testing: Swagger UI, Postman  
- Manual Testing: End-to-end flows  
- Edge Case Handling: Invalid inputs, missing data, connectivity issues

## 12. Screenshots

[Insert mockups of student quiz interface, educator dashboard, diagnostic test flow, Google Classroom sync]

## 13. Known Issues

- Dependent on API quota limits  
- Requires stable internet connectivity  
- Limited to English curriculum initially

## 14. Future Enhancements

- Multi-language support  
- Personalized study recommendations  
- Gamification elements  
- Mobile app  
- LMS integrations