**Smart Bridge Internship Generative AI With IBM Cloud**

**Project Title:**

**EduTutor AI: Personalized Learning with Generative AI and LMS Integration**

**Submitted By**

**Team ID :** LTVIP2025TMID32123

**Team Size :** 4

**Team Leader :** Mande Surya

**Team member :** Kuthadi Sai Kiran

**Team member :** Kola Raja

**Team member :** Lingam Siva Surya Pavan

**Introduction:**

EduTutor AI is an AI-powered personalized learning platform designed to transform how students learn and how educators monitor progress. By integrating generative AI (IBM Watsonx, Granite LLM), Google Classroom, and real-time performance analytics, the platform delivers adaptive, engaging educational experiences. This document outlines the six core development phases that structure the creation of EduTutor AI.

**Phase 1: Brainstorming & Ideation**

**Objective:**

Generate the core idea and envision how AI and LMS can combine to improve education.

**Key Points:**

- Identify the problem: limited personalization in traditional learning, delayed feedback, and static content.

- Ideate AI-powered features: dynamic quizzes, adaptive difficulty levels, real-time student feedback, performance dashboards.

- Visualize integration: IBM Watsonx for diagnostics, Granite LLM for personalized quiz generation.

- Discuss LMS sync: select Google Classroom as the hub for class materials and student data.

**Explanation:**

This phase unites educators, developers, and AI experts to conceptualize a modern digital-learning solution. We aim to blend AI-driven personalization with classroom alignment via LMS integration, crafting a smart ecosystem that adapts to each learner and empowers data-driven teaching.

**Phase 2: Requirement Analysis**

**Objective:**

Define detailed system requirements from all stakeholders.

**Key Points:**

Functional Requirements:

- Student dashboard with personalized quizzes

- Educator dashboard with performance insights

- Real-time feedback on quiz responses

Non-Functional Requirements:

- Modular, scalable architecture

- Low-latency, real-time AI processing

- Secure API integration with Google Classroom

**Technology Stack:**

Front-end:HTML, CSS, JavaScript

-Back-end:APIs for Watsonx, Granite LLM, Pinecone

LMS: Google Classroom for data sync

**Explanation:**

Here, inputs from students, teachers, and admins shape the precise functionality and behavior of EduTutor AI. The goal is a high-performance, secure, scalable web app that seamlessly syncs with existing systems and delivers real-time AI-powered learning.

**Phase 3: Project Design**

**Objective:**

Design the platform’s architecture and user interface.

**Key Points:**

Modular UI/UX design:

- Student interface: login, dashboard, quiz area

- Educator interface: performance analytics, student tracking

- Quiz engine: adaptive test generator

Wireframes for:

- Student quiz interface

- Educator performance dashboard

Integration plans:

- Watsonx (diagnostic testing)

- Granite LLM (quiz generation)

- Pinecone (progress data)

- Google Classroom (class/course sync)Responsive UI patterns: accessibility and mobile usability

**Explanation:**

Balancing functionality with intuitive design, this phase ensures that students and teachers find the platform natural to use. Backend services must communicate smoothly with AI engines, and the UI must adapt across devices while maintaining data security.

**Phase 4: Project Planning (Agile Methodologies)**

Objective:

Break the project into manageable sprints using Agile.

Key Points:

Sprint 1:Google Classroom Integration & Course Sync

Sprint 2:Student Dashboard + Quiz Generator (Granite LLM)

Sprint 3: Educator Dashboard + Performance Insights (Pinecone)

Sprint 4: Diagnostic Testing & Adaptive Quizzing (Watsonx)

Agile processes:daily stand-ups, sprint reviews & retrospectives, backlog grooming

Explanation:

Agile delivery lets us ship working features incrementally and adapt to feedback. Each sprint targets a specific module, enabling continuous refinement and stakeholder engagement.

**Phase 5: Project Development**

Objective:

Implement the features using the chosen technologies.

Key Points:

HTML:Structure for login forms, dashboards, quiz panels

CSS: Responsive, accessible styling across devices

JavaScript:

- API calls to sync Google Classroom data

- Communication with Watsonx and Granite LLM for quizzes and diagnostics

- Dynamic quiz rendering, answer validation, instant feedback

- Front-end state management for student performance

- Develop reusable components for quizzes and dashboards

Explanation:

In this build phase, HTML lays the foundation, CSS brings clarity and usability, and JavaScript powers interactivity. The front-end remains modular, while APIs handle AI and LMS integration in real time.

**Phase 6: Functional & Performance Testing**

Objective:

Ensure the application functions correctly and performs under load.

Key Points:

Unit Testing: validate individual JS components (quiz rendering, data sync)

Integration Testing:

- Google Classroom sync

- Retrieval of AI-generated quizzes from Granite LLM and Watsonx

Functional Testing:

- Student quiz workflows and feedback

- Educator performance analytics

Performance Testing: simulate concurrent users for smooth operation

Final Steps: bug fixes, UI optimization, deployment readiness

Explanation:

Combining automated and manual tests, we verify stability across scenarios, devices, and browsers. Once all criteria are met, the platform is primed for deployment.