

Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	27 June 2025
Team ID	LTVIP2025TMID59564
Project Name	EduTutor AI: personalized learning with generative AI and lms integration
Maximum Marks	4 Marks

Technical Architecture:

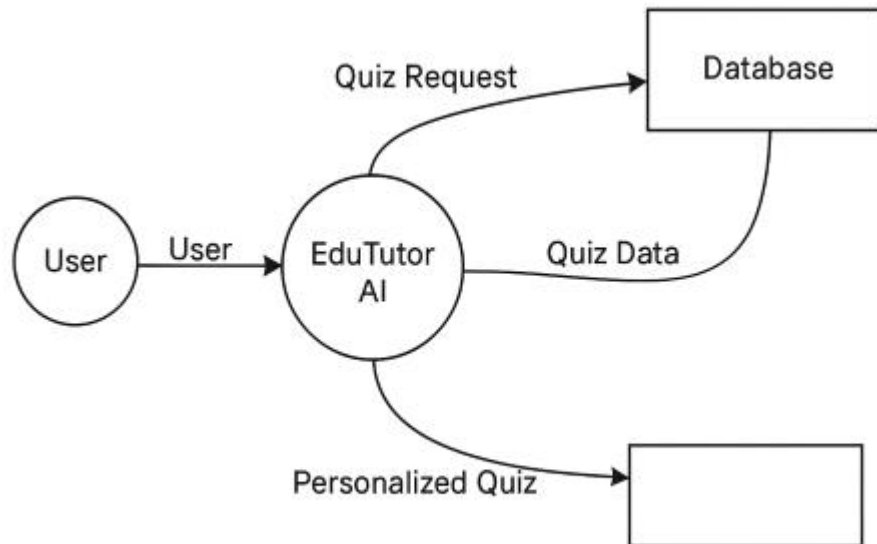


Table-1 : Components & Technologies:

S.No	Component	Description	Technology / Tools
1	User Interface	Web App & Mobile App (chat + dashboard)	Streamlit (Web), Flutter (Mobile), HTML, CSS
2	Application Logic-1	AI quiz generation from user topic & preferences	FastAPI, Python (LangChain, Prompt Templates, Moderation logic)
3	Application Logic-2	Text analysis + RAG for better quiz generation	IBM Watsonx.ai (Granite LLM), Pinecone (vector search), LangChain
4	Application Logic-3	Classroom management and quiz distribution	Google Classroom API integration
5	Database	Store quiz metadata, user details, history, and scores	PostgreSQL / SQLite
6	Cloud Database	Scalable remote storage of user data, logs, history	IBM Cloudant / Firebase Cloud Firestore

7	File Storage	Save reports, exports, model logs, and uploaded files	IBM Cloud Object Storage / Firebase Storage
8	External API-1	Fetch context for improved quiz generation	Wikipedia API / Hugging Face inference APIs
9	External API-2	User authentication & integration with LMS	Google OAuth2 / Classroom API
10	Machine Learning Model	Generate and evaluate quizzes from academic content	IBM Watsonx Granite (LLM), Text Moderation, RAG pipeline (LangChain)
11	Infrastructure	Backend deployment and app hosting	IBM Cloud / GCP / Streamlit Cloud / Firebase Hosting

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	UI and backend frameworks	Streamlit, FastAPI, Scikit-learn, Watsonx, LangChain
2	Security Implementations	Data encryption, role-based access, authentication	SHA-256, OAuth 2.0 (Google), .env Secrets, IAM Roles
3	Scalable Architecture	Microservice-ready, Firebase scaling, container deployment supported	FastAPI, Pinecone, Docker, Kubernetes-ready
4	Availability	99.9% uptime via managed cloud infrastructure	IBM Cloud Load Balancer, Google Firebase Hosting (Multi-region)
5	Performance	Use of CDN for static content, async API, model inference <3 sec response	Streamlit Caching, Firebase CDN, Watsonx Optimization, Async APIs

References:

<https://c4model.com/>

<https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>

<https://www.ibm.com/cloud/architecture>

<https://aws.amazon.com/architecture>

<https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>

