

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

Date	27 June 2025
Team ID	LTVIP2025TMID29500
Project Name	EduTutor-AI_personalized-learning-withgenerative-ai-and-lms-integration
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

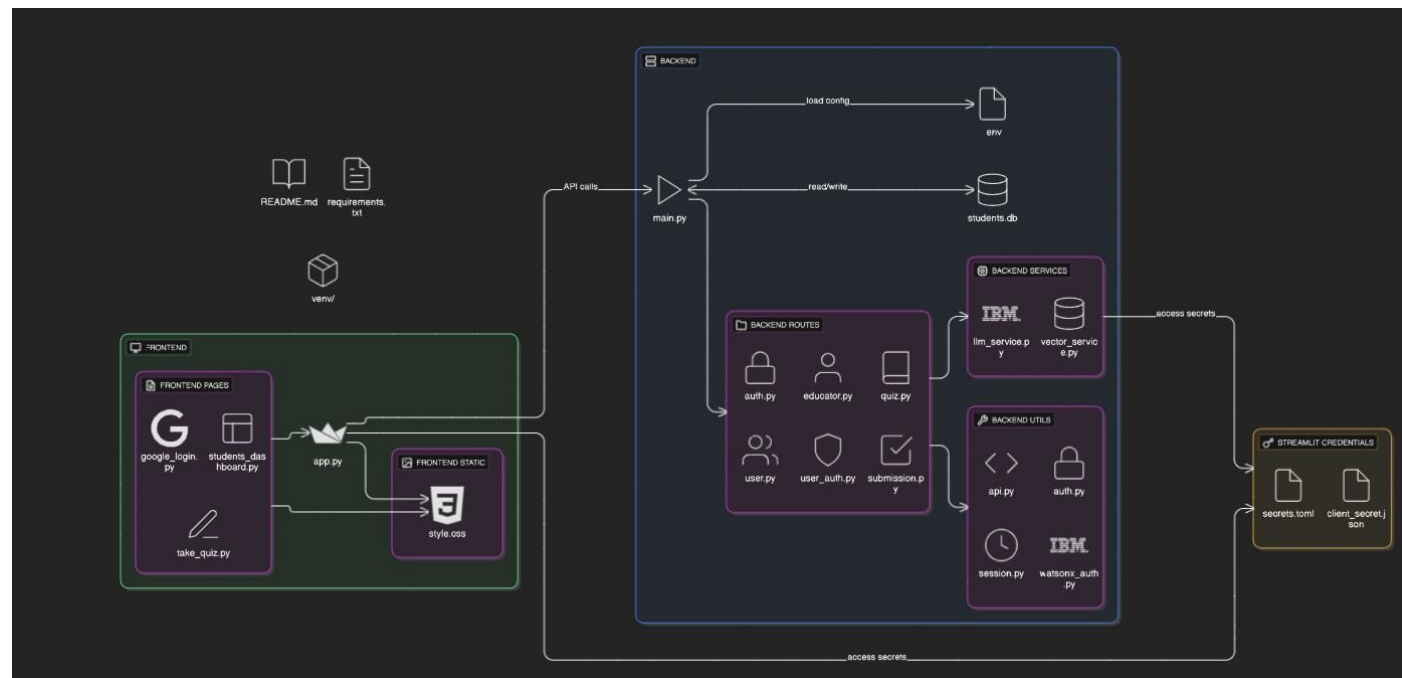


Table-1 : Components & Technologies:

S.No	Component	Description	Technology/Tools Used
1.	User Interface	How users interact with the application — for quiz access and results view	Streamlit (Student/Admin UI), React JS
2.	Application Logic-1	Backend logic for quiz generation, student evaluation	Python (Flask/FastAPI)
3.	Application Logic-2	Voice-to-text conversion for quiz questions (if used)	IBM Watson Speech to Text (STT)
4.	Application Logic-3	Chatbot for user support or quiz guidance	IBM Watson Assistant
5.	Database	Stores student records, quiz questions, scores	MySQL, Firebase Realtime DB, NoSQL
6.	Cloud Database	Cloud-hosted database for scalability	IBM Cloudant, IBM DB2 on Cloud
7.	File Storage	Stores result exports, user-uploaded files, etc.	IBM Cloud Object Storage, Local Filesystem
8.	External API-1	For quiz content enrichment (e.g., weather quiz, if applicable)	IBM Weather API, Open Trivia DB
9.	External API-2	For secure user verification or personalization	Aadhar API, Google OAuth API
10.	Machine Learning Model	AI-based model for adaptive quiz difficulty or score prediction	Scikit-learn, Custom ML Model
11.	Infrastructure	Where the app is hosted and deployed	IBM Cloud Foundry, Kubernetes, Localhost

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology Used
1.	Open-Source Frameworks	Frameworks used for frontend, backend, and ML	Streamlit, React JS, Flask, Scikit-learn, Pandas, NumPy

2.	Security Implementations	User data protection, secure login, and API access	SHA-256 password hashing, OAuth2.0, JWT, IAM Controls, HTTPS, OWASP Guidelines
S.No	Characteristics	Description	Technology Used
3.	Scalable Architecture	Designed using a 3-tier structure with potential for microservices (modularized quiz engine, auth)	3-Tier Architecture, Docker, IBM Cloud Kubernetes, FastAPI/Flask APIs
4.	Availability	Deployed with high availability through cloud services, backups, and horizontal scaling	IBM Cloud Load Balancer, Redundancy on Kubernetes, Auto-Scaling, Cloud Foundry
5.	Performance	Designed for fast quiz access and high concurrency; uses caching and stateless APIs	Redis Cache, CDNs, Optimized DB Queries, Async APIs, Uvicorn (FastAPI)