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

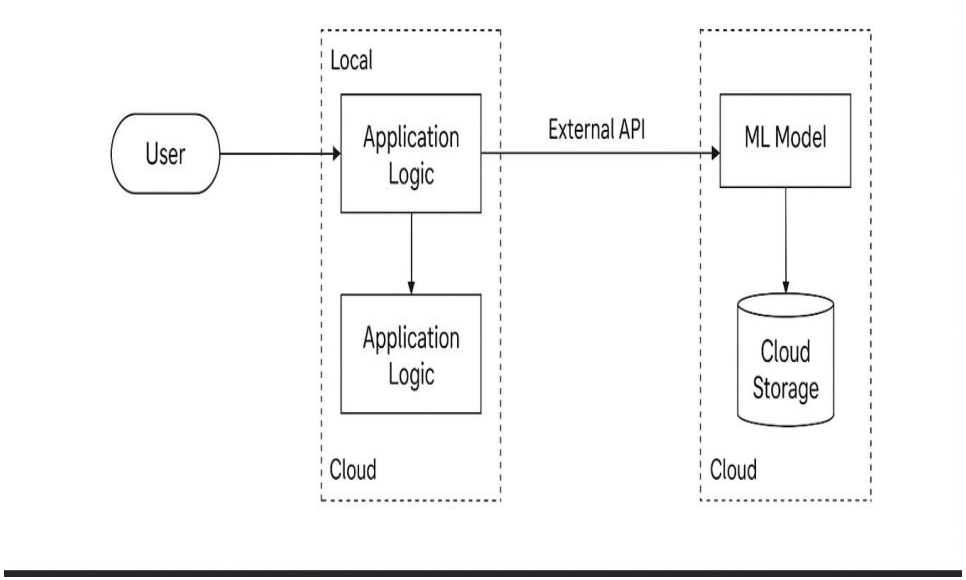
Date	27 June 2025
Team ID	LTVIP2025TMID30145
Project Name	EduTutor AI: Personalized Learning with Generative AI and LMS Integration
Maximum Marks	

Technical Architecture:

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

Example: Example: EduTutor AI - AI-powered personalized learning system

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/



Guidelines:

EduTutor AI – AI-powered personalized learning system

- Include all the processes (As an application logic / Technology Block)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third party API's etc.)
- Indicate Data Storage components / services
- Indicate interface to machine learning models(if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI for interaction	Gradio (Python), HTML/CSS (auto-generated by Gradio)
2.	Application Logic-1	AI educational Q&A system logic	Python
3.	Application Logic-2	Math Solver logic	Python, SymPy
4.	Application Logic-3	Quiz generator, study planner, and progress tracker	Python
5.	Database	Stores session data, quiz responses (currently in-memory)	(Extendable to NoSQL or SQLite/Cloud DB)
6.	Cloud Database	For scaling and storage	Hugging Face (model hosting), optional Cloud DB
7.	File Storage	Temporary data and caching	Local File System / Gradio state
8.	External API-1	Accessing generative language models	Hugging Face Hub API
9.	External API-2	Loading secrets for authentication (if on Colab)	Google Colab userdata API
10.	Machine Learning Model	AI model for generating educational responses	ibm-granite/granite-3.3-2b-instruct (Hugging Face)
11.	Infrastructure (Server / Cloud)	Development local; inference model via cloud model hub	Local (dev), Hugging Face Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Libraries and frameworks used are open source	Transformers, Gradio, SymPy, Python
2.	Security Implementations	Token-based access for Hugging Face models, secret loading via Colab	Hugging Face token auth, Google Colab secrets

S.No	Characteristics	Description	Technology
3.	Scalable Architecture	Modular components and fallback models allow vertical and horizontal scalability	Python OOP, Hugging Face model hosting
4.	Availability	AI models hosted on Hugging Face, available 24x7 for query handling	Hugging Face Hub
5.	Performance	Supports GPU acceleration, caching, and prompt optimization	PyTorch, Transformers pipeline

References:

https://huggingface.co/ibm-granite/granite-3.3-2b-instruct

https://github.com/gradio-app/gradio

https://huggingface.co/docs/transformers

https://www.sympy.org/en/index.html

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

https://c4model.com/