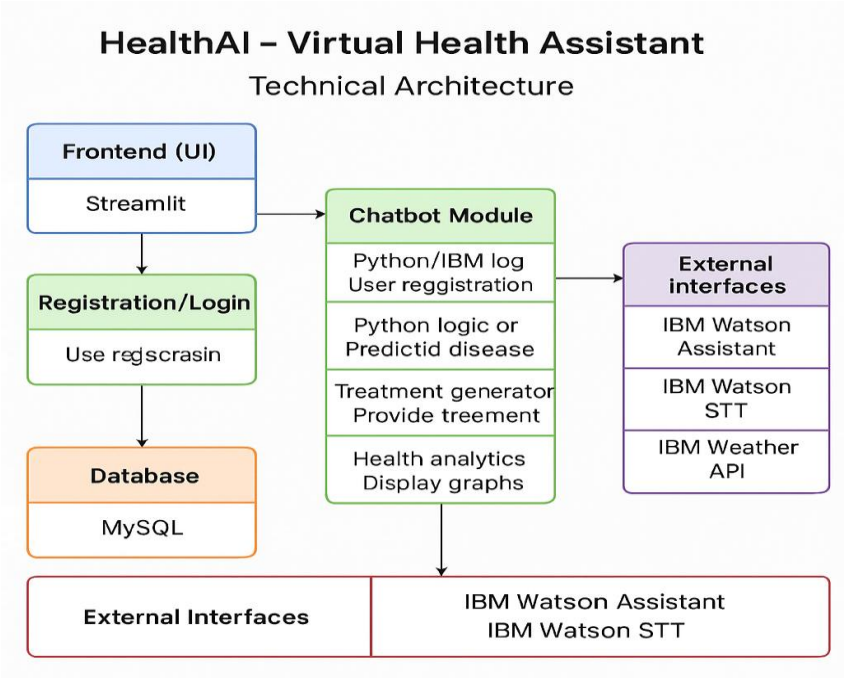
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	05 June 2025
Team ID	LTVIP2025TMID33353
Project Name	**HealthAI: Intelligent Healthcare Assistant Using IBM Granite**
Maximum Marks	4 Marks

Technical Architecture:

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

Example: Virtual health consultation and disease prediction during pandemics or in remote areas using offline mode – HealthAI



Guidelines:

- Provide **basic AI-driven health assistance** without relying on live internet access.
- Ensure that **user data and responses are stored locally** and sync when online.
- Improve **health access during pandemics or emergencies** when mobility and connectivity are limited.

Table-1: Components & Technologies:

S.No	Component	Technology
1	User Interface	HTML, CSS, Flask (Python-based Web UI)
2	Application Logic-1	Python for backend processing, symptom analysis
3	Application Logic-2	(Optional) IBM Watson STT service for voice-to-text (future scope)
4	Application Logic-3	IBM Watson Assistant for chatbot response (future scope)
5	Database	MySQL for user registration and symptom data
6	Cloud Database	IBM DB2 (optional for scaling)
7	File Storage	Local Filesystem for temporary session data
8	External API-1	IBM Weather API (optional: future health suggestions by weather)
9	External API-2	Aadhaar API (optional for authentication in national deployment)
10	Machine Learning Model	Virtual Health Assit. chat using ML (Simulated / IBM Granite 13B API)
11	Infrastructure (Server / Cloud)	Local (Streamlit) / IBM Cloud (future deployment via Cloud Foundry)

Table-2: Application Characteristics

S.No	Characteristics	Technology
1	Open-Source Frameworks	Flask(Python), Scikit-learn
2	Security Implementations	Password hashing, MySQL user authentication, OWASP guidelines(Optional)
3	Scalable Architecture	Three-tier: UI + Logic + DB; future microservices deployment possible
4	Availability	Can be hosted on IBM Cloud with autoscaling and high uptime guarantees
5	Performance	Optimized Python backend, low-latency response, suitable for 100+ requests/sec with lightweight architecture