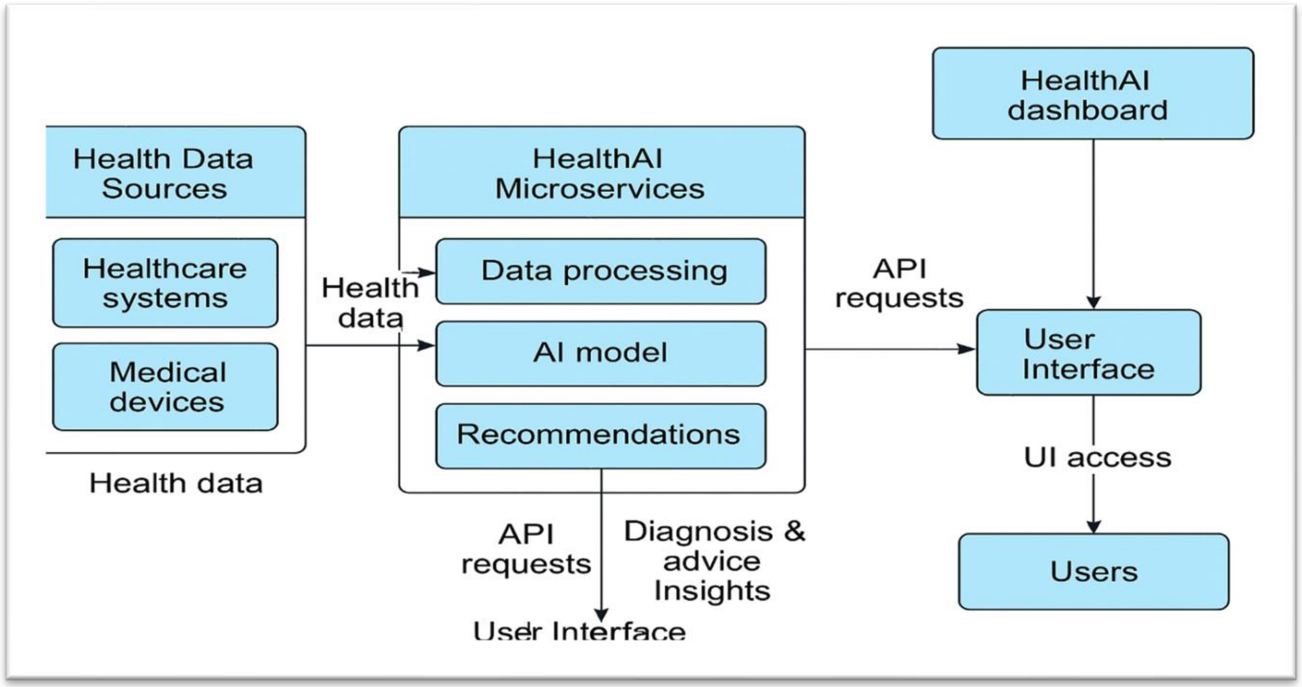
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	19/05/2025 – 30/06/2025
Team ID	LTVIP2025TMID31711
Project Name	HealthAI: Intelligent Healthcare Assistant Using IBM Granite
Maximum Marks	4 Marks

Technical Architecture:



1. Processes (Application Logic / Technology Blocks)

Component	Description	Technology Used
User Interface	Frontend for user interaction (profile, symptoms, chatbot, reports)	Streamlit (Python), HTML
Data Input & Validation	Collects and validates user health data	Python
Symptom Checker	Extracts symptoms and triggers diagnosis logic	Python logic
AI Diagnosis Model	Predicts possible disease	IBM Granite AI via Hugging Face Inference API
Treatment Planner	Suggests remedies/treatment plans	IBM Granite model + rule-based logic
Chronic Management Module	Logs and visualizes glucose, heart rate, etc.	Python, Matplotlib, Pandas
Analytics Dashboard	Visualizes trends and statistics	Streamlit, Plotly, SQLite
Email Notifier (Optional)	Sends treatment advice or reports via email	Gmail SMTP API

2. Infrastructural Demarcation

Layer	Description	Deployment
Frontend/UI	Streamlit app, hosted via cloud	Streamlit Cloud / IBM Cloud
Application Logic Layer	Python backend handling AI calls and processing	Cloud (IBM Cloud Functions or App Engine)
Model Layer	Remote inference of IBM Granite model	Hugging Face Inference API
Database	Stores user profile, logs, history	Local SQLite or IBM Cloudant
File Storage	Store user-generated reports (optional)	Streamlit cache / IBM Object Storage

3. External Interfaces / Third-Party APIs

API Name	Purpose
Hugging Face Inference API	To access IBM Granite AI model
Gmail SMTP	Sending email confirmations or reports
(Optional) IBM Watson STT / TTS	For voice input/output extension
(Optional) Aadhar/Health APIs	For user identity or health records

4. Data Storage Components

Storage	**Usage**	**Technology**
Local Database	User profiles, symptoms, logs	SQLite
Cloud Database (optional)	Scalable storage for large health logs	IBM Cloudant
Temporary Storage	Cached outputs, user-generated reports	Streamlit Cache / Local filesystem

5. Machine Learning Interface

Model	**Purpose**	**Access Method**
IBM Granite (13B Instruct)	To understand symptoms, generate diagnosis, recommend treatment	Accessed via Hugging Face API
(Optional) **Health-specific fine-tuned models**	Future extension for disease classification or prediction	Can be added via custom fine-tuning

Table-1: Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Web app interface	Streamlit, HTML/CSS
2	Application Logic-1	Profile setup, data validation	Python
3	Application Logic-2	Symptom checker	IBM Granite model via Hugging Face API
4	Application Logic-3	Health Assistant Chatbot	IBM Watsonx
5	Database	Temporary data storage	Local JSON/SQLite
6	Cloud Database	Optional for scalable version	IBM Cloudant (future use)
7	File Storage	Upload logs, if any	Local filesystem or IBM Object Storage
8	External API-1	Geolocation or Email Service	IP-API, SMTP Gmail
9	Machine Learning	Disease prediction	IBM Granite-13b-instruct-v2
10	Infrastructure	Deployment	Streamlit Community Cloud or IBM Cloud (optional upgrade)

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	Streamlit, Hugging Face Transformers	Python, Streamlit
2	Security Implementations	Email login, encrypted storage, session timeout	SHA-256 (if storing credentials)
3	Scalable Architecture	Can migrate to 3-tier architecture or microservices if required	IBM Cloud Foundry or Kubernetes
4	Availability	Local version always on, cloud-ready	IBM Cloud Load Balancer (future use)
5	Performance	Lightweight app, fast response with IBM models	CDN (optional), minimal delay AI model