

# Health AI with IBM

---

Intelligent Healthcare Assistant using IBM Granite & Generative AI

Prepared by: [ LINGESH G ]

[ MOHAMED PAYAZ ]

[ MOHAMED JAVID ]

[ MOHAMED MANSOOR ]

Institution: [Your Institution Name]

Date: [Insert Date]

# ***Table of Contents***

- 1. Introduction*
- 2. Project Overview*
- 3. Objectives*
- 4. Literature Review*
- 5. Architecture*
- 6. Technology Stack*
- 7. Project Workflow*
- 8. Setup Instructions*
- 9. Folder Structure*
- 10. Running the Application*
- 11. API Documentation*
- 12. Authentication & Security*
- 13. User Interface*
- 14. Testing & Validation*
- 15. Screenshots (Placeholders)*
- 16. Known Issues*
- 17. Future Enhancements*
- 18. References*
- 19. Conclusion*

# 1. Introduction

Healthcare is one of the most critical domains where Artificial Intelligence (AI) can create transformative impact. 'Health AI with IBM' is a project focused on integrating IBM Granite models with healthcare data to provide patients, clinicians, and administrators with intelligent assistance. The system is designed to address challenges such as interpreting medical policies, predicting disease risk, providing personalized health recommendations, and enabling efficient hospital management.

## 2. Project Overview

Purpose:

The purpose of this project is to bridge the gap between raw medical information and actionable healthcare insights. By leveraging IBM Watsonx Granite large language models, the system aims to deliver:

- Simplified summaries of medical guidelines.
- AI-driven forecasting of disease outbreaks.
- Personalized health suggestions for patients.
- Decision support tools for clinicians and policymakers.

Key Features:

- Conversational Interface – Engages with patients and doctors through natural language queries.
- Clinical Policy Summarization – Extracts key points from lengthy medical documents.
- Disease Risk Forecasting – Estimates likelihood of outbreaks and readmission rates.
- Personalized Health Tips – Provides customized wellness advice.
- Patient Feedback Loop – Captures insights from patients for continuous improvement.
- KPI Forecasting – Assists administrators in managing hospital resources.
- Anomaly Detection – Identifies irregular medical patterns for early intervention.
- Multimodal Input Support – Processes text, CSV, and PDF files.
- Streamlit/Gradio UI – Provides interactive dashboards for end-users.

### 3. Objectives

- Enhance patient engagement through AI-driven communication.
- Assist healthcare staff in policy interpretation.
- Provide predictive insights for resource planning.
- Detect anomalies in medical data early to reduce risks.
- Improve accessibility of healthcare information for rural and underserved populations.

### 4. Literature Review

Numerous studies highlight the role of AI in healthcare. Traditional systems often focus on diagnostic imaging, electronic health records, or administrative optimization. However, the unique contribution of Health AI with IBM lies in combining conversational AI with predictive analytics and summarization of medical policies. Unlike single-purpose tools, this assistant provides a holistic platform integrating multiple healthcare workflows.

### 5. Architecture

The architecture integrates frontend, backend, AI models, and vector search into a modular system. The frontend is designed for accessibility, while the backend ensures scalability and API-driven communication.

### 6. Technology Stack

- Programming Language: Python 3.9+
- Backend: FastAPI
- Frontend: Streamlit/Gradio
- AI Models: IBM Watsonx Granite LLMs
- Vector Database: Pinecone
- Libraries: Transformers, Scikit-learn, Pandas, Matplotlib
- Deployment: Google Colab & GitHub

## 7. Project Workflow

1. Activity 1: Access Naan Mudhalvan Smart Interz portal and enroll in the project.
2. Activity 2: Select IBM Granite model from Hugging Face (e.g., granite-3.2-2b-instruct).
3. Activity 3: Run application in Google Colab using T4 GPU with pre-installed libraries.
4. Activity 4: Upload project files to GitHub for version control and collaboration.

## 8. Setup Instructions

Prerequisites:

- Python 3.9 or later
- FastAPI, Streamlit/Gradio
- IBM Watsonx API keys
- Pinecone API key
- GitHub account
- Google Colab with GPU access

Steps:

1. Clone repository.
2. Install dependencies.
3. Configure .env file.
4. Start backend and frontend.
5. Upload medical documents and interact with assistant.

## 9. Folder Structure

- app/ – FastAPI backend code.
- app/api/ – API routes for chat, summarization, and reporting.
- ui/ – Streamlit frontend pages.
- granite\_llm.py – Granite model integration.
- document\_embedder.py – Converts documents into embeddings.
- kpi\_health\_forecaster.py – Forecasting KPIs.
- anomaly\_health\_checker.py – Detects abnormal health data.
- report\_generator.py – Generates healthcare reports.

## 10. Running the Application

1. Start FastAPI server to enable APIs.
2. Launch Streamlit dashboard.
3. Navigate via sidebar to access chat, summaries, and reports.
4. Upload files for analysis.
5. Review predictions and generated reports in real-time.

## 11. API Documentation

- POST /chat/ask – Submits query and returns AI response.
- POST /upload-doc – Uploads medical files for processing.
- GET /search-docs – Finds relevant documents via semantic search.
- GET /get-health-tips – Provides personalized advice.
- POST /submit-feedback – Captures user feedback.

## 12. Authentication & Security

Security is ensured via token-based authentication (JWT, API keys) and OAuth2 integration with IBM Cloud. Future improvements will include multi-factor authentication and encrypted session storage.

## 13. User Interface

The UI is designed for simplicity and accessibility. Features include:

- Sidebar navigation
- Dashboard KPIs
- Chat interface
- Upload forms
- Report downloads

The design follows a minimalist approach, ensuring healthcare staff and patients can use it with minimal training.

## 14. Testing & Validation

The application has undergone multiple testing phases:

- Unit Testing: Validation of data parsing, prompt functions, and ML models.
- API Testing: Using Postman and Swagger UI.
- Manual Testing: For file uploads and UI interactions.
- Edge Cases: Invalid file formats, large datasets, missing credentials.

## 15. Screenshots

[Placeholder – Dashboard Screenshot]

[Placeholder – Chat Interface Screenshot]

[Placeholder – Policy Summarization Screenshot]

## 16. Known Issues

- Dependency on stable internet connectivity.
- Limited support for image-based medical data.
- Requires anonymization of sensitive patient data before upload.

## 17. Future Enhancements

- Integration with Electronic Health Records (EHR).
- Advanced medical image analysis.
- Voice-enabled AI assistant.
- Real-time patient alerts for critical anomalies.

## 18. References

1. IBM Granite Models: <https://huggingface.co/ibm-granite>
2. Gradio Framework: <https://www.gradio.app/guides/>
3. Python Documentation: <https://docs.python.org/3/>
4. Git Documentation: <https://git-scm.com/docs/git>
5. Google Colab: <https://colab.research.google.com/>
6. Naan Mudhalvan Portal: <https://naanmudhalvan.smartinternz.com/>

## **19. Conclusion**

Health AI with IBM demonstrates how generative AI and predictive analytics can revolutionize healthcare. By integrating conversational interfaces, policy summarization, forecasting, and patient engagement tools, the system addresses multiple pain points in healthcare delivery. With further development and integration, this project has the potential to scale into a comprehensive healthcare AI platform.