Health AI: Intelligent Healthcare Assistant Using IBM Granite.

Project Documentation.

Introduction:

Healthcare is one of the most essential sectors, yet it faces challenges such as overburdened doctors, limited accessibility in rural areas, and difficulty in managing large volumes of patient data. With the advancement of Artificial Intelligent (AI), it has become possible to provide intelligent healthcare solutions that improves both accessibility and efficiency.

Health AI: Healthcare assistant using IBM Granite is designed to address these challenges but integrating AI – powered language models, machine learning, and secure cloud – based technologies. This system acts as a virtual healthcare assistant, capable of analyzing patient symptoms, providing preliminary health recommendations, forecasting potential risks, and assisting doctors in summarizing medical reports.

The project combines Stream lit (frontend), Fast API (backend), IBM Watsonx Granite (AI/LLM), and Pinecone (vector search) to deliver an interactive, secure, and intelligent healthcare platform. By leveraging AI, Health AI aims to reduce waiting times, improve healthcare decision-making, and offer personalized support to both patients and doctors.

1. **Introduction:**

- Project Title: Health AI: Intelligent Healthcare Assistant Using IBM Granite.
- Team Leader/ID: R.Oviya/NM2025TMID07036
- Team Member: B.Abinaya
- Team Member: C.Darmitha
- Team Member: K.Suganya

2. **Project Overview:**

Health AI is an intelligent healthcare assistant that leverages cutting – edge AI to offer services like disease prediction, personalized treatment plans, patient chat, and health data analytics. The project aims to make health advice accessible and insightful through a user – friendly interface. Its built using Python, with a focus on enhancing personalized health guidance.

Purpose:

The primary purpose of Health AI is to provide quick, reliable, and interactive health – related information. It addresses the need for on – demand health insights, helping users understand their symptoms, explore potential health trends, and receive personalized advice. The system aims to improve healthcare accessibility by assisting users with instant responses, health tips, and reminders, while also reducing dependency on immediate hospital visits for minor issues. Additionally, it provides a secure platform for storing and managing health profiles, medical history, and medication schedules, ensuring a holistic approach to personal healthcare.

Features:

1. **Medical Data Processing**

    - Analyse structured (lab reports, vitals) and unstructured (doctor's notes, medical images) data.
    - Helps in extracting meaningful patterns for diagnosis and treatment.

2. **Clinical Decision Support**

    - Assists doctors by suggesting possible diagnoses and treatment plan
    - Reduce human error and improves evidence – based care.

3. **Predictive Analytics**

    - Uses patient history + lifestyle + genomics to predict disease risks.
    - Helps in early intervention and preventive care.

4. **Medical Imaging & Diagnostics**

    - AI detects patterns in X-rays, MRIs, CT scans faster than humans.
    - Identifies tumours, fractures, or anomalies with high accuracy.

5. **Personalized Medicine**

    - Suggests treatment tailored to patient's genetics, medical history, and lifestyle.
    - Supports precision oncology and rare disease management.

6. **Remote Patient Monitoring**

- Wearable devices + AI track heart rate, glucose, sleep patterns, etc.
- Alerts doctors about abnormalities in real time.

7. **Natural Language Processing (NLP)**
   - Understands clinical notes, discharge summaries, and research papers.
   - Enables chatbots and virtual health assistants.

8. **Drug Discovery & Development**
   - Speeds up identifying drug molecules and testing simulations.
   - Reduces R&D costs and time for new treatments.

9. **Operational Efficiency**
   - AI helps in hospital resource management (beds, staff schedules, equipment).
   - Reduces waiting times and optimizes workflows.

10. **Patient Engagement & Virtual Assistance**
    - AI chatbots for appointment booking, symptom checking, FAQs.
    - Improves patient satisfaction and accessibility.

3. **Architecture:**

- **Frontend (Stream lit)**
  - Provides an interactive dashboard for patients and doctors.
  - Displays real-time health trends, predictions, and chatbot interface.
- **Backend (Fast API)**
  - Acts as the middle layer handling API requests.
  - Manages user data, authentication, and ML model execution.
- **LLM Integration (IBM Watsonx Granite)**
  - Enables natural language interactions.
  - Provides AI-driven health recommendations and answers**.**
- **Vector Search (Pinecone)**

- Stores embeddings of medical knowledge.
- Supports semantic search for quick retrieval of medical documents**.**

- **ML Modules (Forecasting & Anomaly Detection)**
  - **Forecasting:** Predicts health metrics like heart rate, blood pressure, sugar levels.
  - **Anomaly Detection:** Detects abnormal variations in patient data.

4. **Setup Instructions:**

Prerequisites:

- Python 3.9+
- Pip or conda
- IBM Watsonx API key
- Pinecone API key
- Git

Installation Process:

- Clone the repository
- Install dependencies from requirements.txt
- Create a. env file and configure credentials
- Run the backend server using Fast API
- Launch the frontend via Stream lit
- Upload data and interact with the modules

5. **Folder Structure:**

- app/ -- Contains all Fast API backend logic including routers, models, and integrations modules.
- app/api/ -- Subdirectory for modular API routes like chat, feedback, report, and document vectorization.
- ui/ -- Contains frontend components for Stream lit pages, card layouts, and from UIs.
- Smart_dashboard.py – Entry scripts for launching the main Stream lit dashboard.
- Granite_llmy.py – Handles all communication with IBM Watsonx Granite model including summarization and chat.
- document_embedder.py – Converts documents to embeddings and stores in Pinecone.
- Kpi_file_forecaster.py – Forecasts future energy / water trends using regression.

- anomaly_file_checker.py – Flags unusual values in uploaded KPI data.
- report_generator.py – Constructs AI – generated sustainability reports.

6. **Running the Application:**

To start the project:

- ❖ Launch the Fast API server to expose backend endpoints.
- ❖ Run the Stream lit dashboard too access the web interface.
- ❖ Navigates through pages via the sidebar.
- ❖ Upload documents or CSVs, interact with the chat assistant, and view outputs like reports, summaries, and predictions.
- ❖ All interactions are real – time and use backend APIs to dynamically update the frontend

Frontend (Stream lit):

The frontend is built with Stream lit, offering an interactive web UI with multiple pages including dashboards, file uploads, chat interface, feedback forms, and report viewers. Navigation id handled through a sidebar using the stream lit – option – menu library. Each page is modularized for scalability.

Backend (Fast API):

Fast API serves as the backend REST framework that powers API endpoints for document processing, chat interactions, eco tip generation, report creation, and vector embedding. It is optimized for asynchronous performance and easy Swagger integration.

7. **API Documentation:**

- POST / chat / ask –Accepts a user query and responds with an AI – generated message.
- POST / upload – doc – Uploads and embeds documents in Pinecone
- GET / search – doc – Returns semantically similar policies to the input query
- GET / get – eco – tips – Provides Healthcare tips for selected topics like energy, water, or waste
- POST / submit – feedback – Stores patient feedback later review or analytics

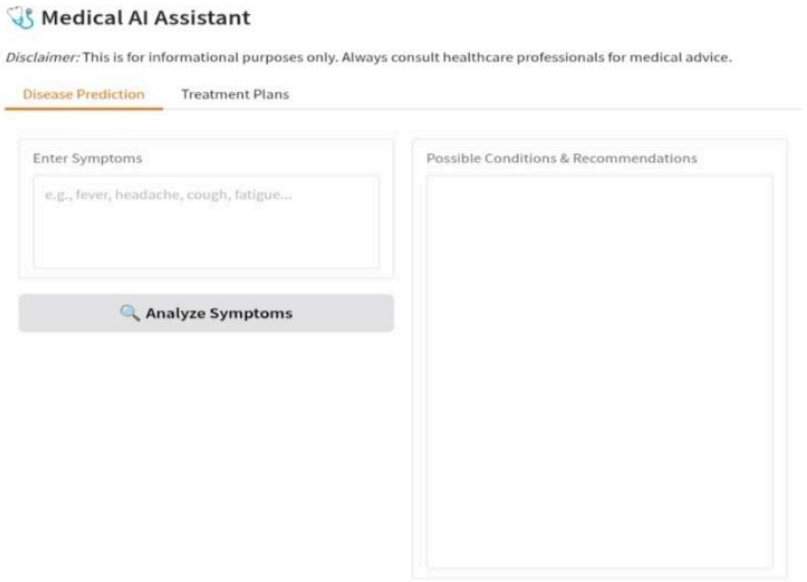
Each endpoint is tested and documented in Swagger UI for quick inspection and trial during development.

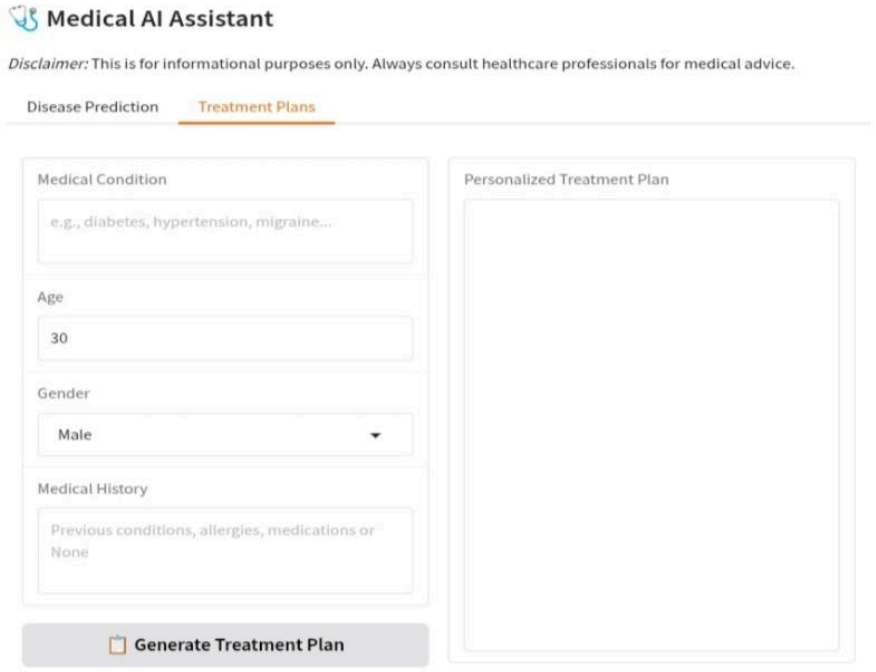
8. **Authentication:**
    - JWT – based authentication system
    - User registration and login functionality
    - Tokens required for accessing secured endpoints
    - Role – based access (Patient or doctor)

9. **User – interface:**
    - Stream lit – based dashboard
    - Patient data input forms
    - AI Chatbot for symptom queries
    - Visualizations of forecasts and anomalies
    - Simple and interactive design

10. **Testing:**
    - **Unit Testing:** Core functionalities and models
    - **Integration Testing:** API and frontend – backend communication
    - **ML Validation:** Accuracy and performance of models
    - **Tools Used**: Pytest, Postman

11. **Screenshot:**


Medical AI Assistant
Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.
Disease Prediction
Treatment Plans
Enter Symptoms
e.g., fever, headache, cough, fatigue...
Analyze Symptoms
Possible Conditions & Recommendations


Medical AI Assistant
Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.
Disease Prediction
Treatment Plans
Medical Condition
e.g., diabetes, hypertension, migraine...
Age
30
Gender
Male
Medical History
Previous conditions, allergies, medications or None
Generate Treatment Plan
Personalized Treatment Plan

12. **Known Issues:**

    - Dependency on IBM Watsonx Granite availability
    - Limited dataset size may affect predictions
    - Performance variation with large medical records
    - Vector database costs for large -scale usage
    - LLM response may sometimes generate generic advice
    - Requires stable internet for LLM & Pinecone integration
    - Forecasting limited to structured datasets.

13. **Future Enhancements:**

    - Add support for wearable IOT devices integration.
    - Multi – language support for patients.
    - Advanced anomaly detection with deep learning models.
    - Integration with hospital database (FHIR/HL7).
    - Mobile applications versions for easier patient access.
    - Provide doctor and hospital recommendation system based on patient symptoms.
    - Expand support for Electronic Health Records (EHR).

14. **Objectives & Scope:**

- **Objectives:**
    - To Provide AI – driven healthcare supports.
    - To make healthcare accessible and affordable.
    - To assist doctors with quick insights.
- **Scope:**
    - Covers patients, doctors, and healthcare institutions.
    - Supports both preventive and diagnostic healthcare.

15. **Problem Statement:**

- Traditional healthcare systems face:
    - Overloaded doctors and long waiting times.
    - Limited access to quality medical guidance in rural areas.
    - Difficulty in maintaining accurate medical records.
- Health AI solves these by offering intelligent, real – time, AI – based healthcare

16. **Use Case Scenarios:**

1. **Patient Use Case –** A patient enters symptoms Health suggests possible causes + next steps.
2. **Doctor Use Case –** A doctor uploads medical reports Health AI generates quick summaries.
3. **Emergency Case –** Wearable device detects abnormal heart rate Health AI sends alerts.

17. **Technology Stack:**
    - **Frontend:** Stream lit
    - **Backend:** Fast API
    - **Database:** MongoDB / PostgreSQL (choose one)
    - **AI / LLM:** IBM Watsonx Granite
    - **Vector Search:** Pinecone
    - **ML Models:** Scikit – learn, TensorFlow, or PyTorch
    - **Authentication:** JWT, OAuth2

18. **Performance Metrics:**
    - ❖ Response time of APIs (should be < 1 sec for queries).
    - ❖ Accuracy of symptom checker (% correctness).
    - ❖ Forecasting error rate (using RMSE / MAE).
    - ❖ Scalability (number of concurrent users supported).

19. **Limitations:**
    - Not a replacement for certified medical diagnosis.
    - Accuracy depends on quality for training data.
    - Requires internet access for AI / LLM queries.
    - Free API limits (IBM Granite, Pinecone).

20. **References:**

- ❖ IBM Watsonx Granite official documentation,
- ❖ Pinecone Vector DB Documentation.
- ❖ Stream lit & Fast API developer docs.
- ❖ Research Papers on AI in healthcare.

Conclusion:

The development of Health AI demonstrates the potential of AI – driven systems in transforming modern healthcare. By integrating IBM Watsonx Granite for intelligent responses, ML modules for forecasting and anomaly detection, and secure APIs for real – time interactions, this project provides a reliable and scalable solutions to many healthcare challenges.

While Health AI is not a substitute for certified medical diagnosis, it serves as a powerful support tool that enhances accessibility, reduces manual workload, and improves patient engagement. With future enhancements such as wearable integration, multilingual supports, and real – time monitoring, Health AI can evolve into a comprehensive digital healthcare companion.

In conclusion, this project highlights how AI – Powered healthcare assistants can bridge the gap between patients and healthcare providers, offering a step forward towards smarter, more accessible, and efficient healthcare systems.