Inteligent Healthcare Assistant Using Ibm Granite

**Project Report**

Submitted by: G.venkat reddy,Guvvala Dhanush kumar,koka yogesh,vasavi.T

Technology Stack: Streamlit+ IBM granite API key

**Project Description:**

**HealthAI** is an intelligent healthcare assistant that combines **IBM Watson Machine Learning** and **Generative AI** to deliver accurate, personalized medical support. **treatment suggestions**, and track their health through **visual analytics**. Users can chat with a virtual assistant for health questions, get symptom-based **disease predictions**, receive **customized**

Powered by **IBM’s Granite-13b-instruct-v2** model and built with **Streamlit**, HealthAI offers a secure, easy-to-use experience. It helps users make informed health decisions with confidence and convenience.

Before deepdive into project , experience our

HEALTH AI from this URL :

https://healthaiassistant.streamlit.app/

**Scenario:**

Scenario 1 – Chat with AI Doctor: Users can enter medical queries and get generated responses from ibm granite.

Scenario 2 – Disease Predictor: Users enter symptoms (e.g., "fever and cough"), and the system predicts likely conditions like "Flu".

Scenario 3 – Treatment Plan Generator: The user provides a disease name (e.g., "Malaria"), and the system generates a simple treatment outline.

**Technical Architecture:**



**Project workflow:**

**Activity 1: Model Selection and Architecture**

Activity 1.1: Research and select the appropriate AI model from IBM Watson for medical assistance : **IBM’s Granite-13b-instruct-v2**

Activity 1.2: Define the architecture of the application, detailing interactions between the frontend, backend, and AI integration.

Activity 1.3: Set up the development environment, installing necessary libraries and dependencies for Streamlit and IBM Watson ML.

**Activity 2: Core Functionalities Development**

Activity 2.1: Develop the core functionalities: Patient Chat, Disease Prediction, Treatment Plan Generation, and Health Analytics.

Activity 2.2: Implement patient data utilities to manage and visualize health metrics.

**Activity 3: App.py Development**

Activity 3.1: Written the main application logic in app.py, establishing functions for each feature and integrating AI responses.

Activity 3.2: Create prompting strategies for the IBM Granite model to generate high-quality medical content.

**Activity 4: Frontend Development**

Activity 4.1: Design and develop the user interface using Streamlit components, ensuring a responsive and intuitive layout.

Activity 4.2: Create dynamic visualizations with Plotly to display health metrics and trends.

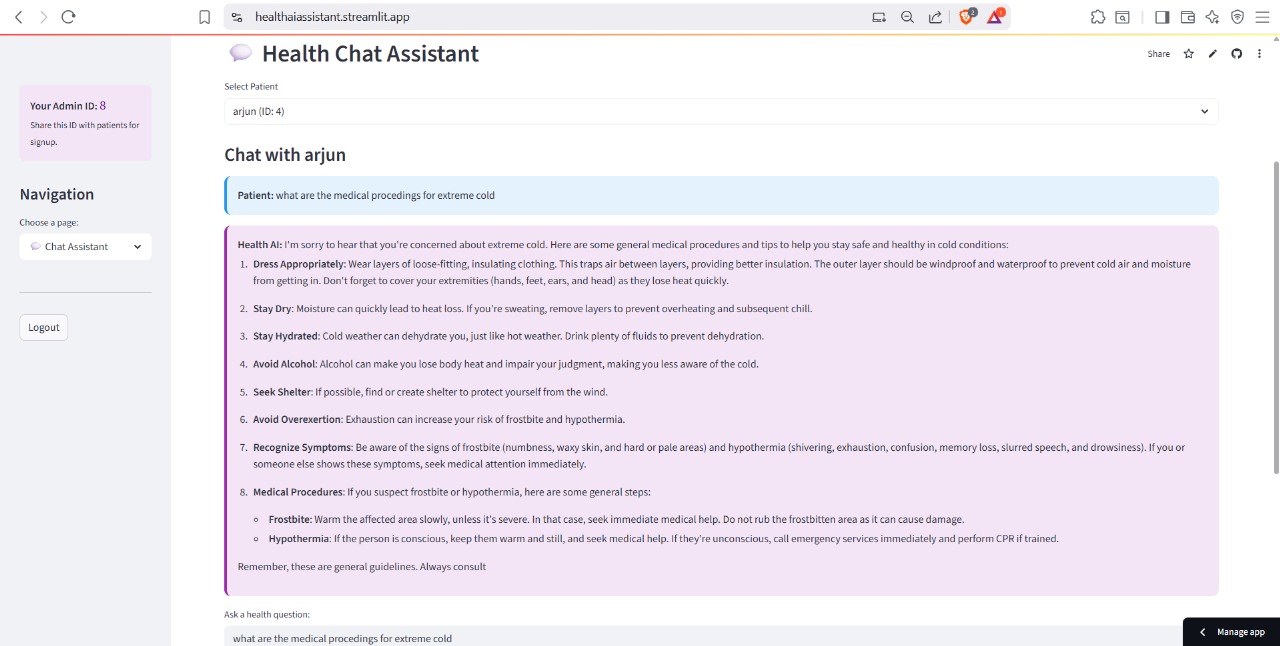
**Activity 5: Deployment**

Activity 5.1: Prepare the application for deployment by configuring environment variables for API credentials.

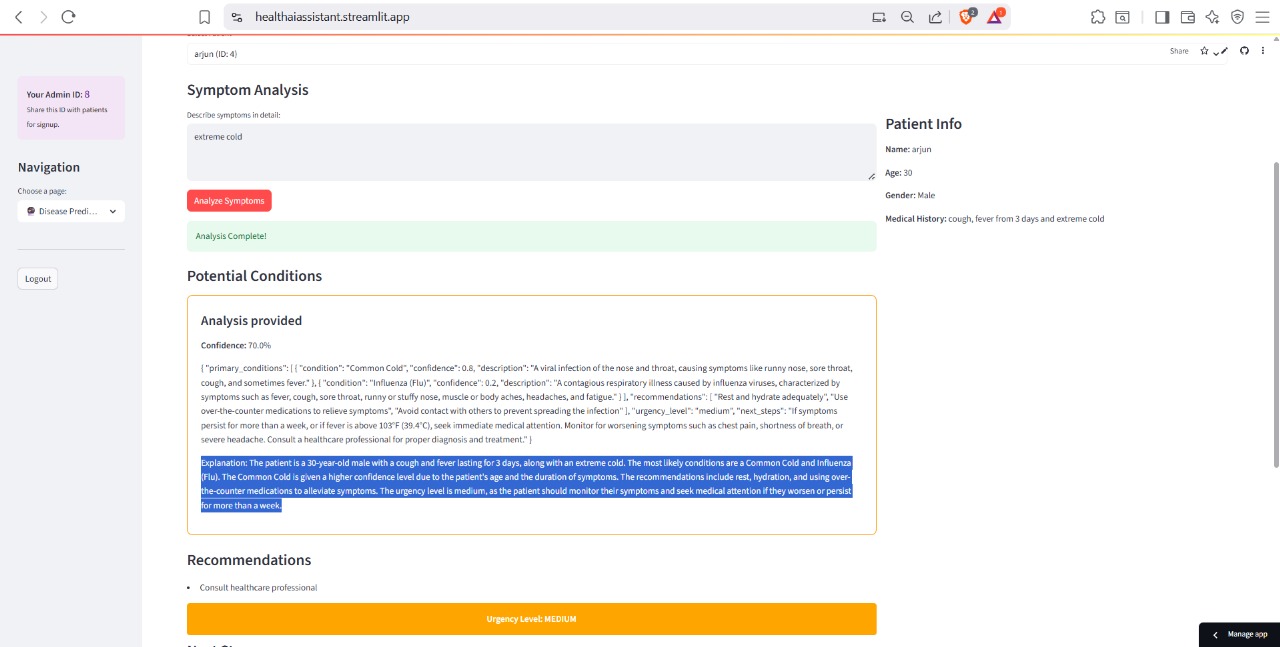
Activity 5.2: Deploy the Sample Screenshot – Disease Predictor:on a suitable hosting platform to make it accessible to users.

**Sample Screenshot – Disease Predictor:**

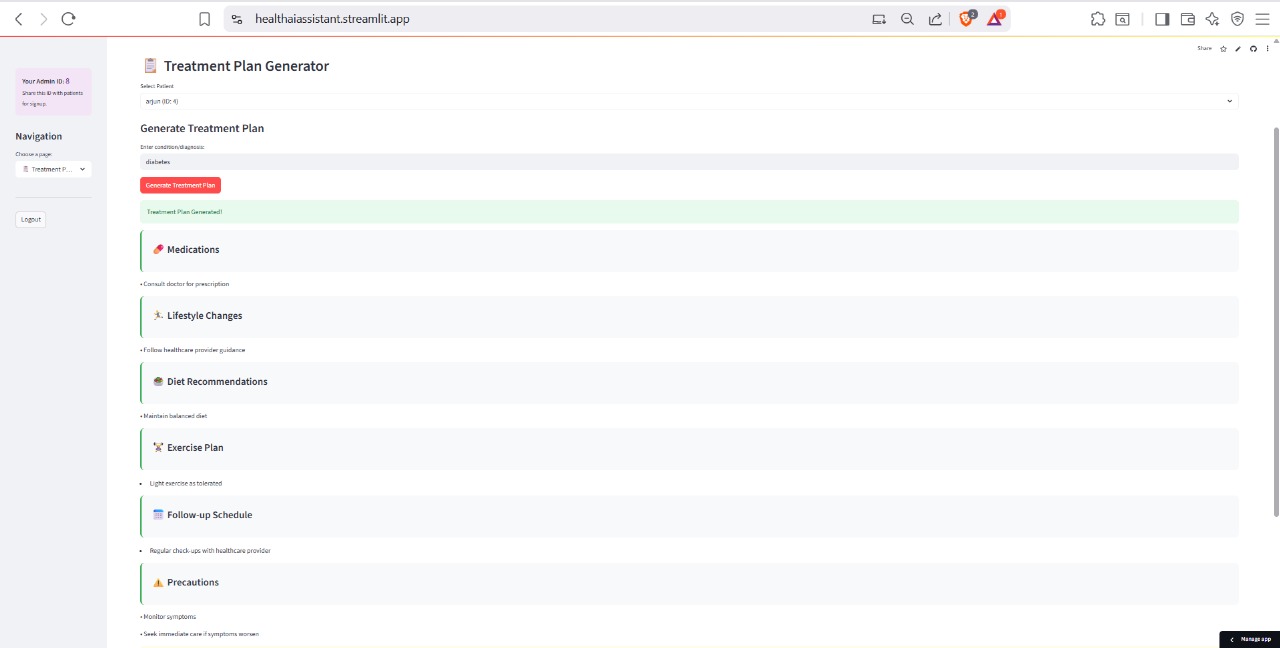
CHAT WITH AI DOCTOR



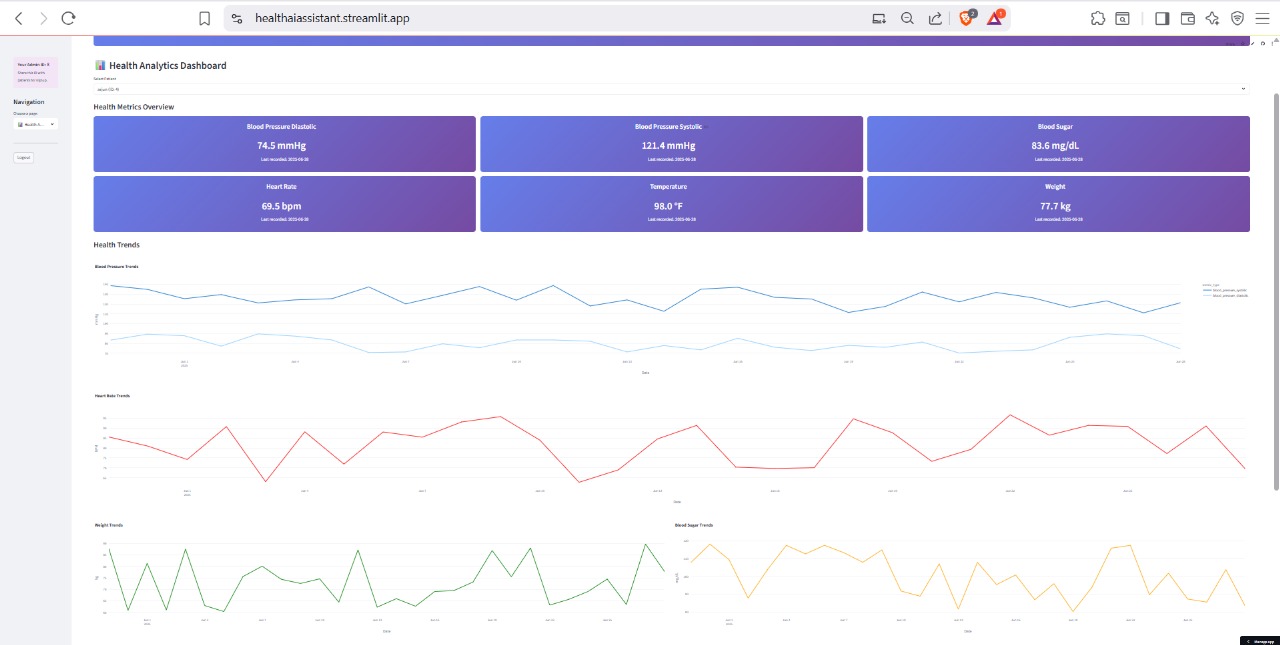
Diseaseprediction



Treatment generator



Health analysis chart



**Milestone Summary**:

Milestone 1: Setup and Architecture - Model selection, architecture design, and dependency installation.

Milestone 2: Core Features Implementation - Coded chatbot, prediction logic, and treatment generator.

Milestone 3: UI and Interaction - Created and connected a streamlit ui components

Milestone 4: Testing and Demonstration - Final testing, screenshots, and documentation creation.

**Detailed Feature Breakdown:**

1. Chat with AI Doctor: This feature allows users to type medical-related questions or concerns. Using Hugging Face’s flan-t5-base model, the app interprets user input and returns a concise, context-aware answer. The goal is not to replace a professional diagnosis but to provide basic information and guide users to formal care when needed.

2. Disease Predictor: Users provide common symptoms (e.g., “fever and cough”), and the system performs rule based matching to predict common conditions such as flu, COVID-19, or migraine. The logic can be extended using scikit-learn classifiers trained on real symptom-disease datasets for greater accuracy.

3. Treatment Plan Generator: This feature uses the same generative model to output simple, general-purpose treatment guidance for common diseases. The generated response typically includes home remedies, dietary suggestions, and general steps. Users are explicitly informed that it is not a replacement for a prescription or diagnosis.

**Technologies Used:**

- streamlit: For building the web-based UI with tab navigation and real-time interaction.

-IBM GRANITE MODEL API KEY: For accessing **IBM’s Granite-13b-instruct-v2** (or equivalent) and using the to generate AI responses.

- Python: Core backend logic for connecting user input to model outputs.

**limitations**:

- All responses are AI-generated and may lack medical accuracy.

- The model relies on internet connectivity and Hugging Face API limits.

**Future Enhancements**:

- Integrate machine learning classifiers trained on a real dataset like SymCat or WHO.

- Add health analytics dashboard using Plotly for time series visualizations.

- Implement session-based user history for persistent interaction.

**Conclusion**:

HealthAI demonstrates how modern AI tools like IBM GRANITE AND streamlit can be integrated to build intelligent, accessible healthcare assistants. By offering basic prediction, chat, and treatment features in a simple UI, the project showcases a practical use case of generative AI in health tech. Running fully in VS CODE locally, HealthAI serves as a minimal, no server solution for intelligent health guidance.