HealthAI: Intelligent Healthcare Assistant Using IBM Granite

**Category: Cloud Application Development**

**Skills Required:**Python,IBM Cloud,Scikit-Learn

**Project Description:**

Project Description:

HealthAI harnesses IBM Watson Machine Learning and Generative AI to provide intelligent healthcare assistance, offering users accurate medical insights. The platform includes a Patient Chat for answering health-related questions, Disease Prediction that evaluates user-reported symptoms to deliver potential condition details, Treatment Plans that provide personalized medical recommendations, and Health Analytics to visualize and monitor patient health metrics.

Utilizing IBM's Granite-13b-instruct-v2 model, HealthAI processes user inputs to deliver personalized and data-driven medical guidance, improving accessibility to healthcare information. Built with Streamlit and powered by IBM Watson, the platform ensures a seamless and user-friendly experience. With secure API key management and responsible data handling, HealthAI empowers users to make informed health decisions with confidence.

Scenarios:

Scenario 1:A user inputs their symptoms into the Disease Prediction system, describing issues like persistent headache, fatigue, and mild fever. The system analyzes the symptoms along with the patient's profile and health data to provide potential condition predictions, including likelihood assessments and recommended next steps.

Scenario 2:A user needs personalized treatment recommendations for a diagnosed condition. By entering their condition in the Treatment Plans generator, the AI processes the information along with patient data to create a comprehensive, evidence-based treatment plan that includes medications, lifestyle modifications, and follow-up testing.

Scenario 3:A user wants insights about their health trends. Using the Health Analytics dashboard, they can visualize their vital signs over time (heart rate, blood pressure, blood glucose, etc.) and receive AI-generated insights about potential health concerns and improvement recommendations.

Scenario 4:A user has a health-related question. Through the Patient Chat interface, they can ask any medical query and receive a clear, empathetic response that includes relevant medical facts, acknowledges limitations, and suggests when to seek professional medical advice.

**Technical Architecture:**



**Milestone 1: Model Selection and Architecture**

In this milestone, we focus on selecting the appropriate AI model from IBM Watson for our medical assistance needs. This involves researching the capabilities and performance of various models, ensuring that the chosen model aligns well with our application's objectives of creating a Patient Chat system, Disease Prediction, Treatment Plan Generation, and Health Analytics.

**Research and select the appropriate AI model**

1. Understand the Project Requirements: Review the specific needs of the healthcare application.
2. Explore IBM Watson ML Documentation: Examine the various models available, including their functionalities and limitations.
3. Select the Optimal Model: Choose IBM's Granite 13B Instruct v2 model for its strong performance with healthcare-related content.

Activity 1.2: Define the architecture of the application

1. Draft an Architectural Diagram: Create a visual representation of the application architecture.
2. Detail Frontend Functionality: Outline how users will interact with the application through a Streamlit interface.
3. Outline Backend Responsibilities: Specify how the backend will process user input and communicate with the IBM Granite model.
4. Describe AI Integration Points: Define how the application will make API calls to IBM Watson ML.

Activity 1.3: Set up the development environment

1. Install Python and Pip: Ensure Python is installed along with pip for managing dependencies.
2. Create a Virtual Environment: Set up a virtual environment to isolate project dependencies.
3. Install Required Libraries:  
   pip install streamlit pandas numpy plotly ibm-watson-machine-learning python-dotenv
4. Set Up Application Structure: Create the initial directory structure for the HealthAI application.

