**1. INTRODUCTION:**

**Project Title:**

Intelligent Healthcare Assistant Using IBM Granite

The Intelligent Healthcare Assistant using IBM Granite LLM is an AI-powered digital healthcare companion designed to support patients, doctors,and medical administrators with real-time health recommendations, medical data analysis, and intelligent communication. Built on IBM’s Granite LLM, the assistant offers personalized healthcare insights and streamlines hospital workflows using generative AI capabilities.

**Team Member:**G Sailaja

**2. PROJECT OVERVIEW:**

The Intelligent Healthcare Assistant leverages IBM’s enterprise-grade large language model to offer natural conversation, data-backed health suggestions, and intelligent decision-making in the medical field. It is designed for hospitals, clinics, telehealth platforms, and individual users to simplify complex healthcare tasks and enhance accessibility.

**Objectives:**

* Utilize IBM Granite LLM for accurate and context-aware medical conversations.
* Provide real-time, personalized health recommendations.
* Enable quick retrieval of medical guidelines and patient data.
* Enhance doctor-patient communication with AI-powered interactions.

**Key Features:**

**1. Medical Query Handling:**

Users can ask about:

* Symptoms and health conditions
* Medication guidelines
* Doctor availability
* Lab report interpretation
* Backed by IBM Granite LLM trained in medical language understanding.

**2. Personalized Health Advisor:**

* Recommends preventive care tips based on user profile.
* Sends medication and appointment reminders.
* Suggests diet or lifestyle changes using patient history.

**3. Smart Alerts & Reports:**

* Sends alerts for missed medications or abnormal vitals (if connected to IoT).
* Generates periodic health summaries for patients and doctors.

**4. IoT & EHR Integration:**

* Works with smart devices like fitness bands and health monitors.
* Retrieves and analyzes Electronic Health Records (EHR) for decision support.

**5. Multilingual Support:**

* Communicates in multiple languages to assist diverse users.
* Uses IBM Watson Language Translator + Granite LLM for regional accessibility.

**Architecture:**



**Flow Chart:**

**Program:**

from ibm\_watson\_machine\_learning.foundation\_models import Model

from ibm\_watson\_machine\_learning.metanames import GenTextParams

from ibm\_watson\_machine\_learning import APIClient

api\_key = "YOUR\_IBM\_CLOUD\_API\_KEY"

project\_id = "YOUR\_PROJECT\_ID"

region = "us-south"

wml\_credentials = {

"apikey": api\_key,

"url": f"https://{region}.ml.cloud.ibm.com"

}

client = APIClient(wml\_credentials)

client.set.default\_project(project\_id)

model\_id = "ibm/granite-13b-chat-v1"

model = Model(model\_id=model\_id, credentials=wml\_credentials, project\_id=project\_id)

health\_context = """

Patient: John Doe

Symptoms: Headache, Nausea, Blurry vision

Recent Vitals: BP 140/90, Heart rate 98 bpm

Medical History: Hypertension

Current Medications: Amlodipine

Objective: Provide preliminary diagnosis and advice

"""

prompt = f"""

You are a healthcare AI assistant. Based on the following patient information, provide potential causes, preliminary advice, and next steps:

{health\_context}

Output in structured format.

"""

params = GenTextParams(decoding\_method="greedy", max\_new\_tokens=250, temperature=0.5)

response = model.generate\_text(prompt=prompt, params=params)

print("🩺 Intelligent Healthcare Assistant Output:")

print(response)

**Output:**

**Milestone 1: Requirement Specification**

* Project Name
* Objective
* Functional Requirements
* Non-Functional Requirements
* LLM Model & Platform (IBM Granite)
* Basic Implementation Flow
* Example Prompts
* Sample Python Code

**Milestone 2: Environment Setup**

* Create IBM Cloud project
* Generate API Key
* Install dependencies (Python SDK)
* Authenticate via APIClient
* Deploy Granite LLM

**Milestone 3: AI Model Integration**

* Uses IBM Watsonx Foundation Models
* Processes medical queries and health advice
* Interfaces with health records and devices

**Milestone 4: Backend API**

1. Language Model: IBM watsonx.ai (ibm/granite-13b-chat-v1)
2. Backend Framework: Python (Flask / FastAPI)
3. API Use Cases:

* "What are possible causes of chest pain in elderly patients?"
* "Suggest a balanced diet for diabetic patients."
* "Retrieve last 3 lab test reports of patient X."

**Milestone 5: Frontend UI (Streamlit)**

**Backend:**

* IBM Watsonx.ai with Granite for NLP
* REST API for communication

**Frontend:**

* Streamlit-based web app
* Clean chat interface for patient queries
* Health cards, charts, and alert banners

**Project Structure:**

* Configure IBM Watsonx
* Set up .env for credentials
* Create ibm\_api.py for Granite model wrapper
* Design app.py for Streamlit interface
* Optional EHR integration