

# Healthcare Assistant using IBM Granite

---

## Overview of the App

- **App Name:** HealthAI Assistant
  - **Backend:** Python (Flask)
  - **LLM Model:** IBM Granite (accessed via IBM watsonx.ai)
  - **Functionality:**
    - Answer basic health-related queries
    - Symptom checker (non-diagnostic)
    - Appointment reminders
    - Medication prompts
- 

## Required Tools

pip install flask ibm-watsonx-ai ibm-cloud-sdk-core

---

## app.py – Sample Code

```
from flask import Flask, request, jsonify

from ibm_watsonx_ai.foundation_models import ModelInference
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

app = Flask(__name__)

# IBM Granite Credentials
API_KEY = 'YOUR_IBM_WATSONX_API_KEY'
PROJECT_ID = 'YOUR_WATSONX_PROJECT_ID'
```

```
MODEL_ID = 'granite-13b-chat-v1' # Change to the correct Granite model
version
```

```
# Authenticate
```

```
authenticator = IAMAuthenticator(API_KEY)
```

```
model = ModelInference(
    model_id=MODEL_ID,
    authenticator=authenticator,
    project_id=PROJECT_ID,
    service_url="https://us-south.ml.cloud.ibm.com"
)
```

```
@app.route('/ask', methods=['POST'])
```

```
def ask():
```

```
    user_input = request.json.get('query')
```

```
    if not user_input:
```

```
        return jsonify({'error': 'Missing query'}), 400
```

```
    prompt = f"You are a healthcare assistant. Answer the query: {user_input}"
```

```
    try:
```

```
        response = model.generate_text(prompt=prompt, max_tokens=200)
```

```
        answer = response['results'][0]['generated_text']
```

```
        return jsonify({'response': answer})
```

```
    except Exception as e:
```

```
        return jsonify({'error': str(e)}), 500
```

```
@app.route('/')
```

```
def index():  
    return "HealthAI Assistant using IBM Granite is running!"  
  
if __name__ == '__main__':  
    app.run(debug=True)
```

---

## Sample Prompts

- "What are the symptoms of dengue?"
  - "Remind me to take my medicine at 9 PM."
  - "Can I take paracetamol for headache?"
- 

## Healthcare Assistant using IBM Granite

### 1. Project Introduction

This project involves developing a Healthcare Assistant using IBM Granite, a powerful foundation model available via IBM watsonx.ai. The assistant is capable of responding to healthcare queries, checking symptoms (non-diagnostically), and assisting users with medication schedules and reminders. It's a lightweight Flask-based web API that connects to IBM Granite to generate intelligent health-related responses.

### 2. IBM Granite Overview

IBM Granite models are part of IBM's family of enterprise-grade foundation models. They are built for performance, responsibility, and transparency. Granite supports natural language processing tasks such as question answering, summarization, and classification. In this application, the Granite model is leveraged through watsonx.ai APIs to serve real-time responses to health-related user queries

### 3. Full app.py Code

```
from flask import Flask, request, jsonify  
from ibm_watsonx_ai.foundation_models  
import ModelInference  
from ibm_cloud_sdk_core.authenticators
```

```

import IAMAuthenticator app = Flask(__name__) API_KEY =
'YOUR_API_KEY' PROJECT_ID = 'YOUR_PROJECT_ID' MODEL_ID =
'granite-13b-chat-v1' authenticator = IAMAuthenticator(API_KEY) model
= ModelInference(model_id=MODEL_ID, authenticator=authenticator,
project_id=PROJECT_ID, service_url="https://us-
south.ml.cloud.ibm.com") @app.route('/ask', methods=['POST']) def
ask(): user_input = request.json.get('query') if not user_input: return
jsonify({'error': 'Missing query'}), 400 prompt = f"You are a healthcare
assistant. Answer: {user_input}" try: response =
model.generate_text(prompt=prompt, max_tokens=200) answer =
response['results'][0]['generated_text'] return jsonify({'response':
answer}) except Exception as e: return jsonify({'error': str(e)}), 500
@app.route('/') def index(): return "HealthAI Assistant using IBM Granite
is running!" if __name__ == '__main__': app.run(debug=True)

```

#### 4. Example Usage POST /ask Headers:

Content-Type: application/json Body: {

```

  "query": "What are the symptoms of dengue?"
}

```

Response:{

```

"response": "Common symptoms of dengue include high fever, headache, pain
behind the eyes, muscle and joint pain, and skin rash."

```

```

}

```

#### 5. Output Screenshots (Sample)

Tested with Postman using /ask endpoint.

Sample Request: { "query": "What should I do if I have a cold?" }

Sample Response: { "response": "If you have a cold, rest, drink fluids,  
and consider over-the-counter medications.  
If symptoms persist, consult a doctor." }



By team member GUNTURU VENKATA NARASIMHA

**Team ID : LTVIP2025TMID34812**