Project Title: HealthAI – Intelligent Healthcare Assistant

📄 Description

HealthAI is an intelligent web-based healthcare assistant built using **Streamlit**, **IBM Granite LLM via Hugging Face**, and **Python**. It allows users to interact with AI for health-related queries, get disease predictions based on symptoms, generate treatment plans, and analyze personal health data through uploaded CSV files.

The app leverages natural language processing (NLP) and machine learning to simulate an intelligent assistant that can offer insights and support — although it is not a replacement for professional medical advice.

🎯 Usefulness & Purpose

HealthAI is designed to:

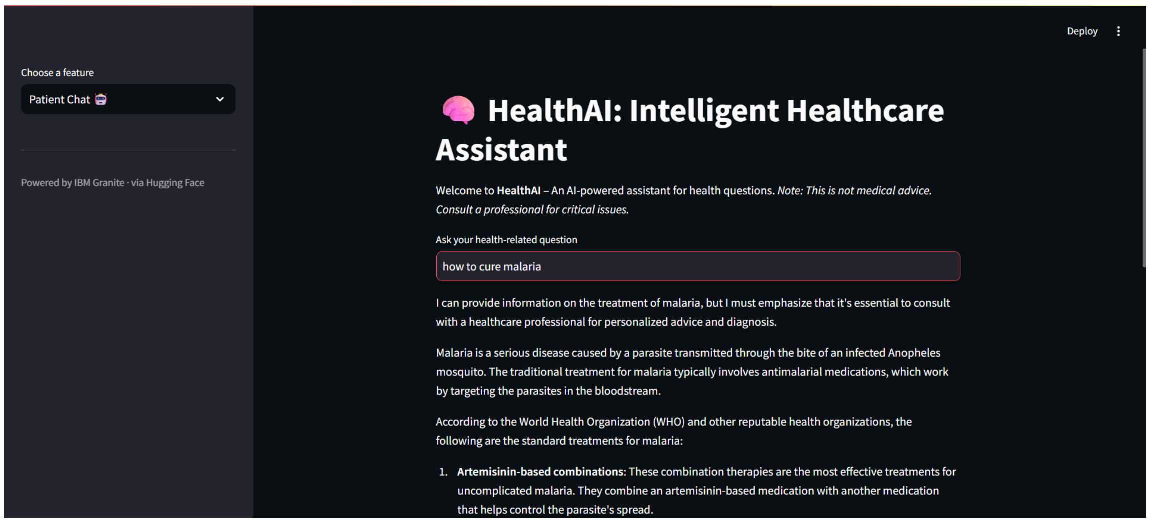
- Empower individuals with basic medical information.
- Aid in understanding symptoms and possible conditions.
- Support doctors by generating treatment outlines.
- Analyze trends in health metrics like glucose, heart rate, and blood pressure.

> 🔐 Note: While AI-driven, HealthAI is for **informational purposes only** and should **not** be considered a substitute for professional medical diagnosis or treatment.

⚙️ Features with Examples

1. 👨💎 Patient Chat (Health Q&A)

- **Function**: Users can ask any health-related question and get an AI-generated answer.
- **Example**: *"What are the early signs of diabetes?"*



🧠 HealthAI: Intelligent Healthcare Assistant

Welcome to **HealthAI** – An AI-powered assistant for health questions. *Note: This is not medical advice. Consult a professional for critical issues.*

Ask your health-related question

how to cure malaria

I can provide information on the treatment of malaria, but I must emphasize that it's essential to consult with a healthcare professional for personalized advice and diagnosis.

Malaria is a serious disease caused by a parasite transmitted through the bite of an infected Anopheles mosquito. The traditional treatment for malaria typically involves antimalarial medications, which work by targeting the parasites in the bloodstream.

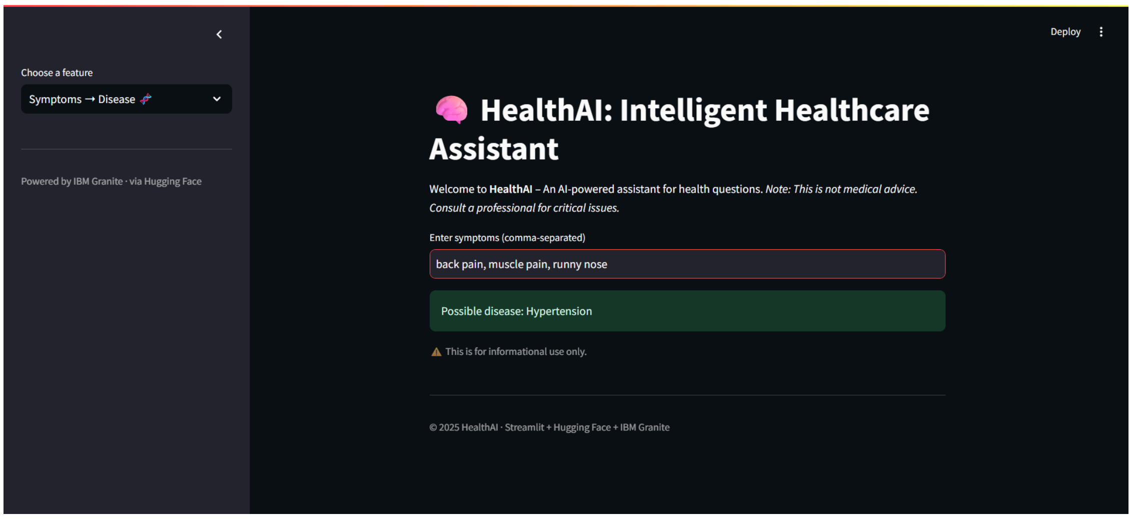
According to the World Health Organization (WHO) and other reputable health organizations, the following are the standard treatments for malaria:

1. **Artemisinin-based combinations:** These combination therapies are the most effective treatments for uncomplicated malaria. They combine an artemisinin-based medication with another medication that helps control the parasite's spread.

---

2. 🔍 Disease Prediction

- **Function**: Based on input symptoms, age, and gender, the AI suggests the most probable conditions.
- **Example**: *Symptoms: fatigue, dry mouth, frequent urination. → Possible Conditions: Diabetes Type 2, Urinary Tract Infection, etc.*



🧠 HealthAI: Intelligent Healthcare Assistant

Welcome to **HealthAI** – An AI-powered assistant for health questions. *Note: This is not medical advice. Consult a professional for critical issues.*

Enter symptoms (comma-separated)

back pain, muscle pain, runny nose

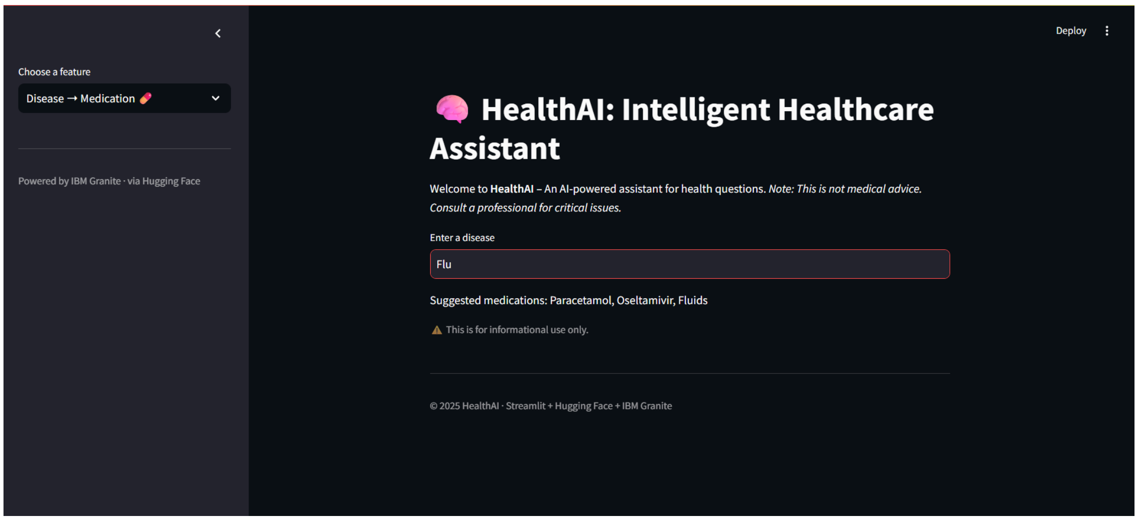
Possible disease: Hypertension

⚠ This is for informational use only.

© 2025 HealthAI · Streamlit + Hugging Face + IBM Granite

3. 💊 Treatment Plans

- **Function**: Users can input a diagnosed condition to get a detailed treatment plan.
- **Example**: *Condition: Hypertension → Medication: ACE inhibitors, Lifestyle: Low-sodium diet, Exercise, Tests: Blood pressure monitoring.*



🧠 HealthAI: Intelligent Healthcare Assistant

Welcome to **HealthAI** – An AI-powered assistant for health questions. *Note: This is not medical advice. Consult a professional for critical issues.*

Enter a disease: Flu

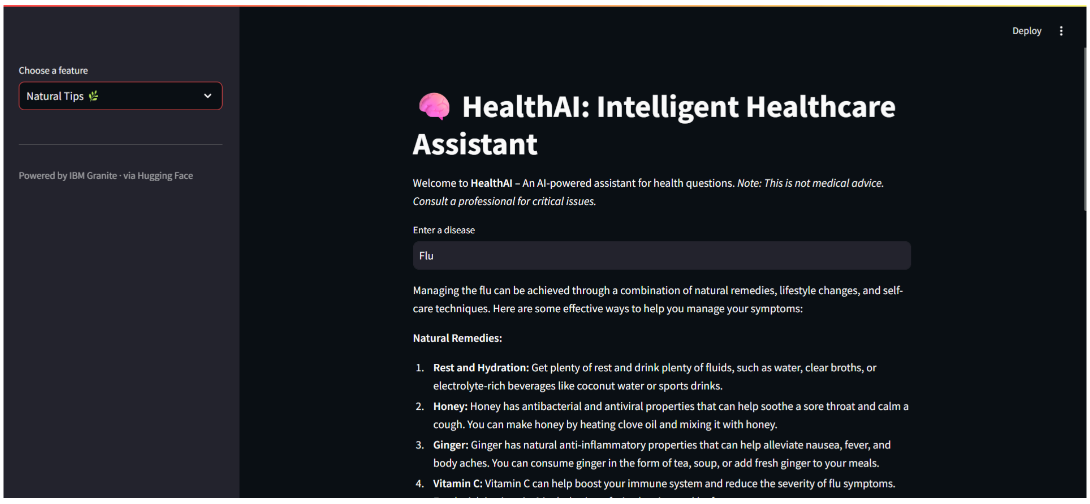
Suggested medications: Paracetamol, Oseltamivir, Fluids

⚠ This is for informational use only.

© 2025 HealthAI · Streamlit + Hugging Face + IBM Granite

4. 📊Natural Remedies

- Suggests natural or home-based treatments for common conditions.
- **Example:**
  - Condition: Indigestion
  - → Remedies: Ginger tea, avoiding spicy food, yoga-based digestion exercises.



🧠 HealthAI: Intelligent Healthcare Assistant

Welcome to **HealthAI** – An AI-powered assistant for health questions. *Note: This is not medical advice. Consult a professional for critical issues.*

Enter a disease: Flu

Managing the flu can be achieved through a combination of natural remedies, lifestyle changes, and self-care techniques. Here are some effective ways to help you manage your symptoms:

Natural Remedies:

1. **Rest and Hydration:** Get plenty of rest and drink plenty of fluids, such as water, clear broths, or electrolyte-rich beverages like coconut water or sports drinks.
2. **Honey:** Honey has antibacterial and antiviral properties that can help soothe a sore throat and calm a cough. You can make honey by heating clove oil and mixing it with honey.
3. **Ginger:** Ginger has natural anti-inflammatory properties that can help alleviate nausea, fever, and body aches. You can consume ginger in the form of tea, soup, or add fresh ginger to your meals.
4. **Vitamin C:** Vitamin C can help boost your immune system and reduce the severity of flu symptoms.

🧾 Code with Explanation

python
CopyEdit
```
# Required Libraries
import streamlit as st
import requests
import os
import pandas as pd
from dotenv import load_dotenv
```

- **Purpose**: Import libraries. `dotenv` loads API key securely from `.env` file.

---

python
CopyEdit
```
# Load API Key
load_dotenv()
HF_API_KEY = os.getenv("HF_API_KEY")
```

- **Purpose**: Securely load Hugging Face API key from environment.

---

python
CopyEdit
```
API_URL =
"https://api-inference.huggingface.co/models/ibm-granite/granite-3.3
-2b-instruct"
HEADERS = {"Authorization": f"Bearer {HF_API_KEY}"}
```

- **Purpose**: Set endpoint for IBM Granite model hosted on Hugging Face.

---

python
CopyEdit
```
# Query Model Function
def query_model(prompt, max_tokens=300, temperature=0.7):
    payload = {
        "inputs": prompt,
        "parameters": {
            "max_new_tokens": max_tokens,
            "temperature": temperature,
            "do_sample": True
        }
    }
    response = requests.post(API_URL, headers=HEADERS, json=payload)
    if response.status_code == 200:
        result = response.json()
        return result[0]["generated_text"]
    else:
        return f"⚠️ API Error: {response.status_code} -
{response.text}"
```

- **Purpose**: Sends prompt to IBM Granite LLM and returns response.

🖥️ Streamlit Layout + Feature Logic

python
CopyEdit
```
# Streamlit Setup
st.set_page_config(page_title="HealthAI", layout="wide")
st.title("🧠 HealthAI: Intelligent Healthcare Assistant")
```

- Sets up page title and header.

python
CopyEdit
```
menu = st.sidebar.selectbox("Choose a Module", [
    "👨💎 Patient Chat",
    "🔍 Disease Prediction",
    "💊 Treatment Plans",
    "📊 Health Analytics"
])
```

- Allows navigation between 4 features/modules.

📌 **Patient Chat**

python

CopyEdit

```
if menu == "👨💎 Patient Chat":
    st.header("💬 Ask a Health Question")
    question = st.text_input("What would you like to ask?")
    if st.button("Get Answer") and question.strip():
        prompt = f"User: {question}\nHealthAI:"
        result = query_model(prompt)
        response = result.split("HealthAI:")[-1].strip()
        st.success(response)
```

📌 **Disease Prediction**

python

CopyEdit

```
elif menu == "🔍 Disease Prediction":
    symptoms = st.text_area("Enter your symptoms (comma separated)")
    age = st.slider("Age", 0, 100, 30)
    gender = st.selectbox("Gender", ["Male", "Female", "Other",
"Prefer not to say"])
    if st.button("Predict Disease") and symptoms:
        prompt = (
            f"A patient reports the following symptoms: {symptoms}.
"
            f"Age: {age}, Gender: {gender}. "
            f"Based on this, what are the top 3 likely conditions
with brief explanations?"
        )
        result = query_model(prompt, max_tokens=400,
temperature=0.8)
        response = result.split("conditions")[-1].strip()
        st.success("🔍 Predicted Conditions:")
        st.write(response)
```

📌 **Treatment Plan**

python
CopyEdit

```
elif menu == "💊 Treatment Plans":
    condition = st.text_input("Enter a diagnosed condition")
    if st.button("Generate Plan") and condition:
        prompt = (
            f"Create a detailed treatment plan for {condition}. "
            f"Include medication, lifestyle changes, and recommended
tests."
        )
        result = query_model(prompt, max_tokens=400)
        response = result.split("plan")[-1].strip()
        st.success("🧾 Suggested Treatment:")
        st.write(response)
```

📌 **Health Analytics**

python
CopyEdit

```
elif menu == "📊 Health Analytics":
    uploaded_file = st.file_uploader("Upload your health data CSV
(date, heart_rate, bp, glucose)", type="csv")
    if uploaded_file:
        df = pd.read_csv(uploaded_file, parse_dates=["date"])
        df.set_index("date", inplace=True)
        metric = st.selectbox("Select a metric to visualize",
df.columns)
        st.line_chart(df[metric])
        st.write(df[metric].describe())
        prompt = f"Analyze this time series health data:
{df[metric].tolist()[:20]}. What patterns or risks do you notice?"
        result = query_model(prompt, max_tokens=200,
temperature=0.6)
        insight = result.split("data:")[-1].strip()
        st.success("🧠 AI Insight:")
        st.write(insight)
```

🚀 Future Enhancements

- **Multi-language Support**: Extend to Hindi, Spanish, etc.
- **Voice Input & Output**: Enable voice commands and text-to-speech responses.
- **EMR Integration**: Link with Electronic Medical Records for deeper analytics.
- **Offline Mode**: Use lightweight on-device models for basic Q&A.
- **User Authentication**: Allow personal health dashboards and history tracking.

🧾 Conclusion

HealthAI demonstrates how modern LLMs can be integrated with intuitive interfaces to create intelligent, accessible healthcare tools. While not a replacement for medical professionals, it empowers users with insights and supports basic decision-making. With future upgrades, it has potential for greater personalization, predictive accuracy, and utility in clinical workflows.