HealthAI - Intelligent Healthcare Assistant Using IBM Granite

Project Documentation

1. **Introduction**

   Project title: HealthAI - Intelligent Healthcare Assistant Using IBM Granite

- **Team member:** Manoj Kumar
- **Team member:** Manikandan
- **Team member:** Kabil Raj
- **Team member:** Jeswan Infant

---

2. Project Overview

Purpose:
The purpose of a Health Care AI Assistant is to empower patients, healthcare providers, and students by offering quick, reliable, and AI-driven medical assistance. By leveraging AI and knowledge-based rules, the assistant helps users analyze symptoms, calculate BMI, estimate calorie requirements, and provide general health tips. For healthcare professionals, it serves as a decision-support partner—offering simplified insights and structured outputs for patient engagement. Ultimately, this assistant bridges technology and healthcare to foster accessible, efficient, and user-friendly medical guidance.

Features:

- **Conversational Interface**
  - *Key Point:* Natural language interaction
  - *Functionality:* Allows users to ask questions, describe symptoms, or request health calculations in plain language.
- **Symptom Checker**
  - *Key Point:* Quick health insights
  - *Functionality:* Matches symptoms with possible conditions and provides basic advice.

- **BMI Calculator**
  - *Key Point:* Body Mass Index calculation
  - *Functionality:* Calculates BMI and categorizes underweight, normal, overweight, or obese.
- **Calorie Calculator**
  - *Key Point:* Daily calorie estimation
  - *Functionality:* Provides calorie requirements based on weight, height, age, gender, and activity level.
- **Chatbot Mode**
  - *Key Point:* Interactive Q&A
  - *Functionality:* Provides simple responses for greetings, health tips, and general queries.
- **User-Friendly Interface**
  - *Key Point:* Accessibility
  - *Functionality:* Provides a simple, colorful, interactive chat-like window for smooth user experience.

---

3. Architecture

Frontend (Tkinter):
The frontend is built with Python Tkinter, offering a chat-like desktop interface. It includes a scrollable chat box, input box, and send button. Messages are color-coded for user and bot, with Enter key support for quick input.

Backend (Python Logic):
Python functions handle health logic, including symptom matching, BMI calculations, and calorie estimation.

AI Integration (Rule-based + Extendable):
Currently rule-based, but can be extended with ML models or APIs (e.g., Gemini, OpenAI).

---

4. Setup Instructions

Prerequisites:

- Python 3.9 or later
- pip and virtual environment tools

Installation Process:

- Download or clone the project folder
- Install dependencies (if any)
- Run the program:
- python gui.py
- Start interacting with the chatbot

---

5. Folder Structure

```
HealthCareAI_Project/
│
├── gui.py              # Tkinter frontend
├── ai_client.py          # AI logic (symptom checker, BMI, calories, chatbot)
├── utils.py            # Helper functions
├── Requirements.txt      # Dependencies (if needed)
```

---

6. Running the Application

To start the project:

1. Run the GUI:
2. python gui.py

3. Type prompts like:
    - hello
    - I have fever and cough
    - bmi 70 175
    - calories 70 175 25 male moderate
4. View the responses in the interactive chat window.

---

7. API Documentation (Future Scope)

If extended with API backend, possible endpoints:

- POST /chat/ask – Accepts a user query and responds with an AI-generated message.
- POST /symptom-check – Matches symptoms to conditions.
- POST /bmi – Calculates BMI.
- POST /calories – Calculates daily calories.

---

8. Authentication (Future Scope)

For secure deployments, planned enhancements include:

- Token-based authentication (JWT or API keys)
- Role-based access (user, doctor, admin)
- User session management

---

9. User Interface

- **Sidebar / Chat Interface** for interaction
- **Color-coded messages** for user and bot
- **Real-time form handling** for BMI and calorie calculation

- **Simple design** prioritizing clarity and accessibility

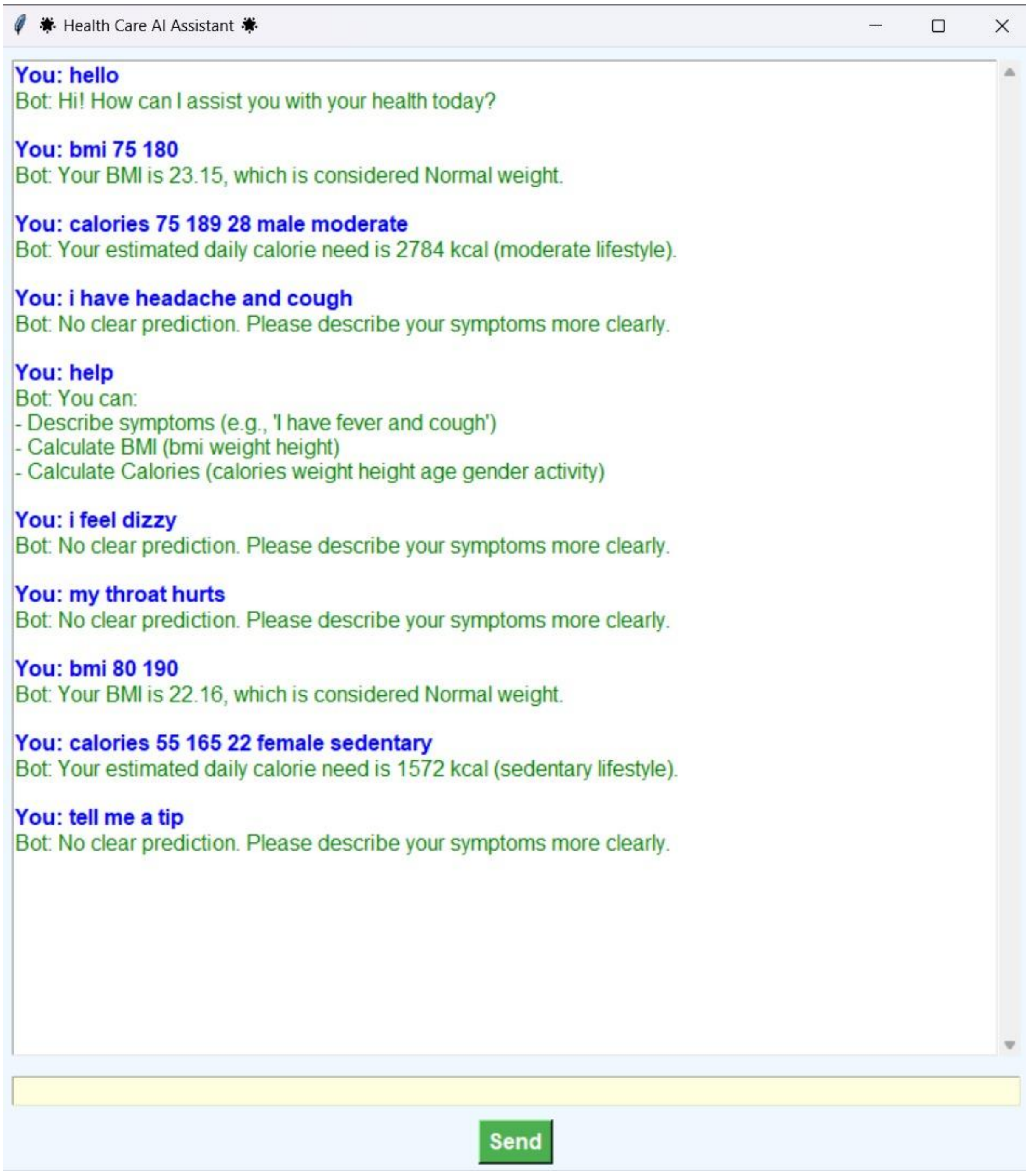
---

10. Testing

- **Unit Testing:** For BMI and calorie functions
- **Manual Testing:** For symptom inputs and chatbot queries
- **Edge Case Handling:** Wrong inputs (e.g., “bmi abc 123”)

---

11. Screenshots



```
You: hello
Bot: Hi! How can I assist you with your health today?

You: bmi 75 180
Bot: Your BMI is 23.15, which is considered Normal weight.

You: calories 75 189 28 male moderate
Bot: Your estimated daily calorie need is 2784 kcal (moderate lifestyle).

You: i have headache and cough
Bot: No clear prediction. Please describe your symptoms more clearly.

You: help
Bot: You can:
- Describe symptoms (e.g., 'I have fever and cough')
- Calculate BMI (bmi weight height)
- Calculate Calories (calories weight height age gender activity)

You: i feel dizzy
Bot: No clear prediction. Please describe your symptoms more clearly.

You: my throat hurts
Bot: No clear prediction. Please describe your symptoms more clearly.

You: bmi 80 190
Bot: Your BMI is 22.16, which is considered Normal weight.

You: calories 55 165 22 female sedentary
Bot: Your estimated daily calorie need is 1572 kcal (sedentary lifestyle).

You: tell me a tip
Bot: No clear prediction. Please describe your symptoms more clearly.
```

---

12. Known Issues

- Limited to rule-based AI (not advanced diagnosis)
- Needs proper medical dataset for stronger predictions
- No backend API yet (currently standalone GUI)

13. Future Enhancements

- Integration with real AI APIs (Gemini / OpenAI)
- Secure authentication for user data
- Advanced ML for disease prediction
- Cloud-hosted version (Flask/Django + React)
- Mobile application interface