HealthAI-Intelligent Healthcare Assistant using IBM Granite

Project Documentation

Project Title: Intelligent Health-Care Assistant using IBM Granite
Team member: Shalini J
Team member: Sharmila G
Team member: Suvvitha A
Team member: Varshene M

1. Introduction

Health-Care AI Assistance is an intelligent software solution that leverages artificial intelligence to provide users with medical guidance and basic health recommendations. The project focuses on making preliminary healthcare support more accessible, giving users the ability to describe symptoms and receive relevant insights. This project is developed as a Python-based application and is simple to install and run, making it ideal for students, researchers, and individuals who wish to explore the application of AI in healthcare.

The system is designed to offer fast responses and can be used as a decision-support tool for initial symptom checking. Although it does not replace professional diagnosis, it encourages users to seek timely medical attention by interpreting their symptoms in a structured way.

2. Project Overview

The main goal of Health-Care AI Assistance is to bridge the gap between symptom onset and medical consultation by providing a first layer of information. Users input their symptoms through the console-based interface, and the application processes this input to generate possible diagnoses or treatment recommendations. The project demonstrates how AI can reduce uncertainty in the early stages of illness and improve awareness of potential health risks.

The scope of this version is limited to local execution and a single-user setup. It does not yet integrate with hospital information systems, wearable devices, or patient record databases. However, it can be extended to include these features in future versions. Its lightweight nature makes it suitable for quick testing, educational use, and demonstration in academic projects.

3. Architecture

The architecture of the project is designed to be minimal and easy to maintain. The application consists of a single Python script, main.py, which handles all functions — input collection, processing, and output display. Users provide their symptoms or concerns through a simple input prompt. The program uses predefined logic or AI models to process this input and returns suggestions or possible conditions. The sequential flow of input → processing → output makes the system straightforward and efficient.

For future scalability, the architecture can be modularized by separating input handling, processing logic, and output presentation into different modules. This would make the system more extensible and allow integration with APIs, medical databases, and graphical user interfaces.

4. Setup Instructions

Setting up Health-Care AI Assistance is a straightforward process. Users must first install Python on their system. The repository can then be cloned from GitHub using the command:

```
git clone https://github.com/shalinij22/Health-Care-AI-Assistance.git
```

Once cloned, users should navigate to the project folder and install any dependencies listed in the repository. After completing the setup, the program can be executed by running:

```
python main.py
```

This launches the application and starts the input prompt, allowing users to begin interacting with the assistant.

5. Folder Structure

The repository follows a simple file organization structure. It contains three main files:
main.py: The core script containing the application logic.
README.md: The documentation file describing how to use the application.
LICENSE: Specifies that the project is open-source and uses the MIT license.

This minimal structure keeps the project lightweight and easy to understand. Future improvements may include adding separate folders for modules, utility functions, test scripts, documentation, and a potential UI layer.

6. Running the Application

To run the Health-Care AI Assistance application, users must execute the main.py file after completing setup. The application will then prompt the user to enter their symptoms or relevant health concerns. Once input is provided, the program processes the data and returns possible conditions or recommended actions. The response is displayed directly in the console, making the interaction quick and efficient. This workflow is designed to be simple so that even first-time users can easily operate the application.

7. API Documentation

The current version of Health-Care AI Assistance does not offer API endpoints, as it is designed for local execution. In the future, RESTful APIs can be implemented using frameworks like Flask or FastAPI. These APIs can include endpoints such as /analyze-symptom for text input analysis and /recommend-treatment for treatment suggestions. This would enable external applications and mobile interfaces to interact with the system programmatically.

8. Authentication & Security

The current implementation does not include authentication mechanisms because it is designed for single-user local use. Since no patient data is stored, security risks are minimal. However, future versions with API access or online deployment must include secure authentication, encrypted data storage, and access control. This is particularly important when handling sensitive medical information to comply with healthcare data protection regulations such as HIPAA.

9. User Interface

Health-Care AI Assistance currently uses a console-based interface, where users type in symptoms or medical concerns and receive textual output. This lightweight interface is easy to use and ensures low resource consumption. However, for improved usability, a graphical interface or web application could be developed using frameworks like Gradio or Streamlit. This would make the application visually appealing and more accessible to non-technical users.

10. Testing

Testing is essential to ensure the reliability of the application. Suggested tests include verifying that the assistant provides meaningful responses to valid symptom inputs, checking how the system behaves with empty or invalid input, and confirming that the program does not crash unexpectedly. Regression tests should be introduced to ensure that future updates do not affect existing functionality. Automated unit tests can also be written for critical functions in main.py to maintain code quality.

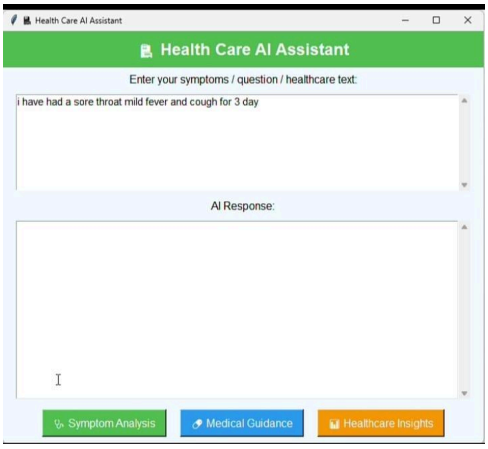
11. Known Issues

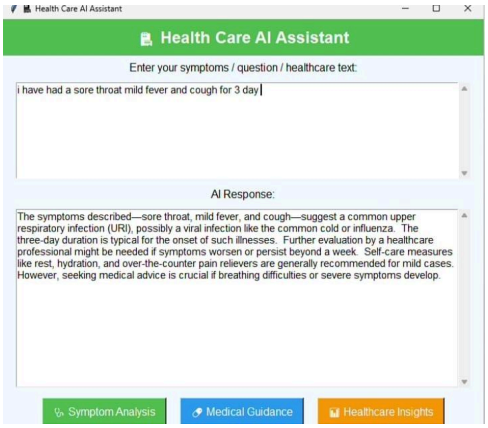
The current version is a prototype and may not always generate medically accurate suggestions. It is not intended for emergency use or professional diagnosis. There is no feature to save or export results, and there is no error handling for missing dependencies or incorrect input formats. Users should treat the outputs as educational and consult a doctor for actual health concerns.

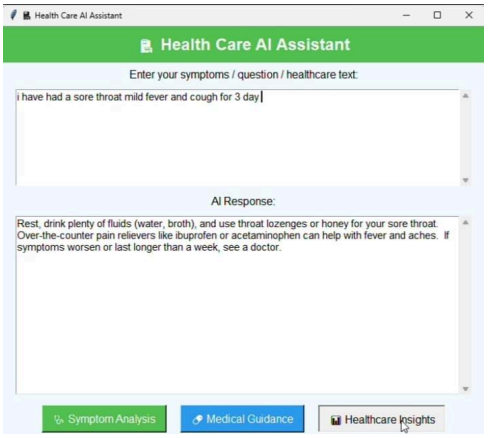
12. Future Enhancements

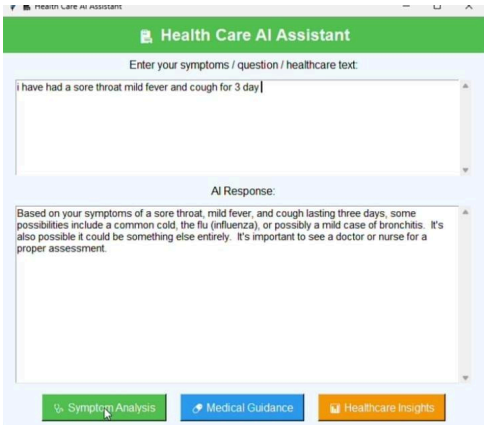
Several improvements are planned for future releases. These include building a web interface for better accessibility, integrating verified medical knowledge bases or AI models to improve the accuracy of results, adding multilingual support, and allowing users to save or export reports. API support and authentication mechanisms will be introduced for multi-user scenarios, making the system suitable for small clinics or educational environments. Additional features like patient history tracking and visualization dashboards can also be considered.

13.Screenshot









14. Conclusion

Health-Care AI Assistance is a promising project that demonstrates how artificial intelligence can be used to support healthcare awareness and education. It provides a simple and accessible way for users to check symptoms and receive helpful suggestions. Though limited in its current form, it lays a strong foundation for future development into a robust healthcare support tool. With enhancements in UI, data security, and medical knowledge integration, the project can grow into a valuable platform for individuals and healthcare providers alike.