

Health AI : Healthcare Assistant

Project Documentation

1. Introduction

- Project title : Health AI : Healthcare Assistant
- Team member : Keerthana M.S
- Team member : Kavya V.
- Team member : Kiruthiga Devi R.
- Team member : Kirthika V.

2. Project Overview

- Purpose :

The purpose of Health AI: Healthcare Assistant is to provide patients and users with accessible, AI-driven healthcare support. Using IBM Granite models and a Gradio-based user interface, the assistant can analyze symptoms, suggest possible conditions, generate treatment plans, and maintain a history of interactions. While it provides valuable guidance, it emphasizes that professional medical advice is always necessary.

- Features:

Login & Signup System Key Point: Secure authentication Functionality: Allows users to create accounts and log in before using the assistant.

Disease Prediction Key Point: AI-powered symptom analysis Functionality: Predicts possible medical conditions from symptoms provided by the user.

Treatment Plan Generator Key Point: Personalized healthcare suggestions Functionality: Generates a treatment plan based on condition, age, gender, and medical history.

History Tracking Key Point: Review past interactions Functionality: Saves timestamped queries and AI responses for user reference.

Clear History Function Key Point: Data management Functionality: Allows users to clear saved query history.

Disclaimer Integration Key Point: Safe usage Functionality: Provides medical disclaimers to ensure users understand the assistant is for informational purposes only.

About Section Key Point: Transparency Functionality: Explains project purpose, technology stack, and limitations.

3. Architecture

Frontend (Gradio): Provides an interactive UI with tabs for login, disease prediction, treatment plan, history, and about section.

Core Logic (Python Functions): Manages user inputs, processes AI queries, handles authentication, and stores history.

LLM Integration (IBM Granite): Uses the Granite 3.2 2B Instruct model from Hugging Face for natural language understanding and text generation.

Deployment (Google Colab): Application is deployed in Google Colab with GPU support for fast response times.

4. Setup Instructions

Prerequisites:

o Python 3.9 or later
o Hugging Face account for IBM Granite model access
o Gradio framework
o PyTorch installed
o Google Colab access with GPU
o Git for version control

Installation Process:

o Open Google Colab and create a new notebook
o Change runtime type to T4 GPU
o Run: `!pip install transformers torch gradio`
o Load IBM Granite model using Hugging Face
o Paste the source code into Colab cells and run sequentially
o Access the Gradio link to interact with the assistant

5. Folder Structure

app/ – Core logic (authentication, disease prediction, treatment plan)
ui/ – Gradio UI components (login, signup, chatbot, tabs)
models/ – AI model integration with IBM Granite
history.py – Manages user query and response
historyapp.py – Main application launcher

6. Running the Application

➤ Open Google Colab and upload project code
➤ Install dependencies using pip
➤ Run the notebook cells in order
➤ Gradio will launch a shareable web link
➤ Login or Signup to access features
➤ Use Disease Prediction, Treatment Plan, and History modules
➤ Clear history or log out when finished

7. API Documentation

POST /login – Authenticates user credentials
POST /signup – Registers a new user
POST /predict – Accepts symptoms and predicts conditions
POST /treatment – Generates treatment plan
GET /history – Retrieves past queries and responses
POST /clear-history – Clears saved history

8. Authentication

Authentication is handled using a simple username-password system. The application allows new user signup and secure login. Future enhancements may include token-based authentication, OAuth2 integration, and role-based access control for admins and patients.

9. User Interface

The interface is built with Gradio and includes:

- Login & Signup screens
- Tabs for Home, Disease Prediction, Treatment Plan, History, and About
- Input fields for symptoms, condition details, and medical history
- Outputs for AI-generated predictions, treatment suggestions, and query history
- Buttons to clear history and navigate between sections

10. Testing

Testing was performed in multiple phases:

- Unit Testing – Functions for login, signup, and history management
- API Testing – Prediction and treatment endpoints
- Manual Testing – End-to-end flow in Google Colab with Gradio UI
- Edge Cases – Empty input, invalid login, mismatched passwords

11. Screenshots

(Screenshots of Gradio UI to be added after deployment)

12. Known Issues

- Predictions are AI-generated and should not replace professional advice
- Limited offline functionality
- Requires stable internet connection
- Basic authentication (username-password only)

13. Future Enhancement

- Integration with hospital databases for real patient data
- Advanced authentication and role-based access
- Mobile application support
- Multi-language support
- Improved disease prediction accuracy with fine-tuned models