Health AI : Intelligence Healthcare Assistant

1. Introduction

• Project title : Health AI : Intelligence 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: Intelligence Healthcare Assistant is to provide users with AI-driven healthcare support through a simple, interactive web application. The system uses IBM Granite large language models integrated with a Gradio-based interface to predict possible health conditions, suggest treatment plans, and maintain a history of user interactions. It also provides a secure login/signup system and includes medical disclaimers to ensure the information is treated as guidance, not a substitute for professional advice.

• Features:

Login & Signup System
Key Point: Secure authentication
Functionality: Allows users to create accounts and log in before accessing healthcare features.

Disease Prediction
Key Point: Symptom-based AI analysis
Functionality: Suggests possible medical conditions based on user-provided symptoms.

Treatment Plan Generator
Key Point: Personalized suggestions
Functionality: Creates a treatment plan using condition, age, gender, and medical history.

History Tracking
Key Point: Review past interactions
Functionality: Maintains a timestamped record of queries and responses for user reference.

Clear History Function
Key Point: Data management
Functionality: Users can clear stored interaction history.

Disclaimer Integration
Key Point: Safe use of information
Functionality: Displays disclaimers ensuring users understand it is for informational purposes only.

About Section
Key Point: Transparency
Functionality: Provides information on the project, technology stack, and limitations.

3. Architecture

Frontend (Gradio): Interactive UI with multiple tabs for login, signup, home, disease prediction, treatment plan, history, and about.

Core Logic (Python Functions): Manages authentication, symptom analysis, treatment plan generation, and history tracking.

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): Runs seamlessly on Google Colab with GPU support for faster model inference.

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 GPU (e.g., T4)

o Run: !pip install transformers torch gradio -q
o Load IBM Granite model using Hugging Face
o Copy and paste the source code into Colab cells
o Execute cells sequentially to launch the Gradio app
o Access the generated 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/ – Model integration with IBM Granite
history.py – Handles user query and response history
app.py – Main application launcher

6. Running the Application

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

7. API Documentation

POST /login – Authenticates user credentials
POST /signup – Registers a new user
POST /predict – Analyzes symptoms and predicts conditions
POST /treatment – Generates a treatment plan
GET /history – Retrieves past interactions
POST /clear-history – Clears stored history

8. Authentication

The application uses a simple username-password authentication system. Users must sign up before logging in. Incorrect credentials are rejected, and new users can register securely. Future upgrades may include encrypted storage, token-based authentication, and role-based access control.

9. User Interface

The Gradio-based interface includes:
• Login & Signup screens

• Home tab with instructions
• Disease Prediction tab with input box for symptoms and AI-generated output
• Treatment Plan tab with input fields for condition, age, gender, and history
• History tab to review past queries and clear records
• About tab with project details and disclaimers

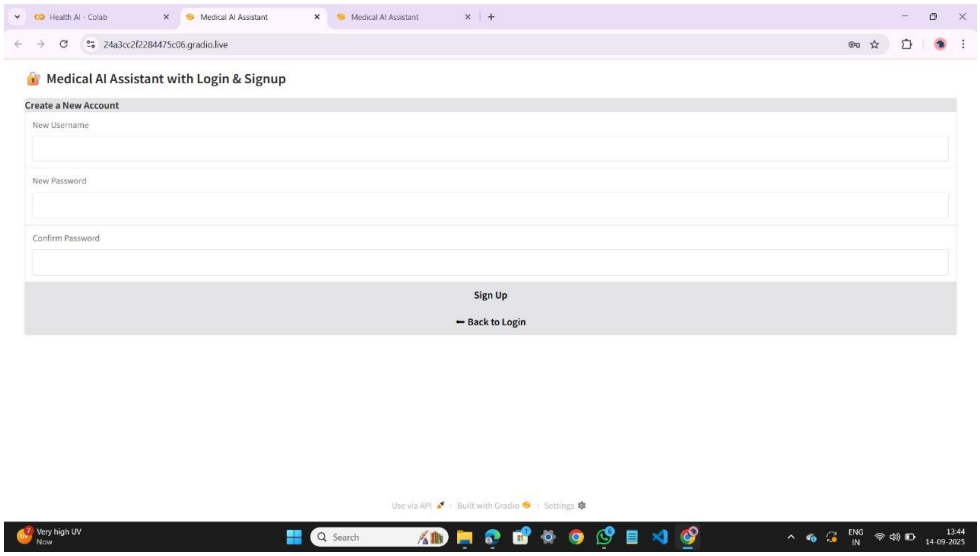
10. Testing

Testing was carried out in multiple phases:
• Unit Testing – Validated individual functions for login, signup, prediction, and treatment
• Functional Testing – Verified interaction between Gradio UI and backend logic
• Manual Testing – Tested full workflow in Colab (login → predict → treatment → history)
• Edge Case Testing – Empty inputs, invalid credentials, mismatched passwords

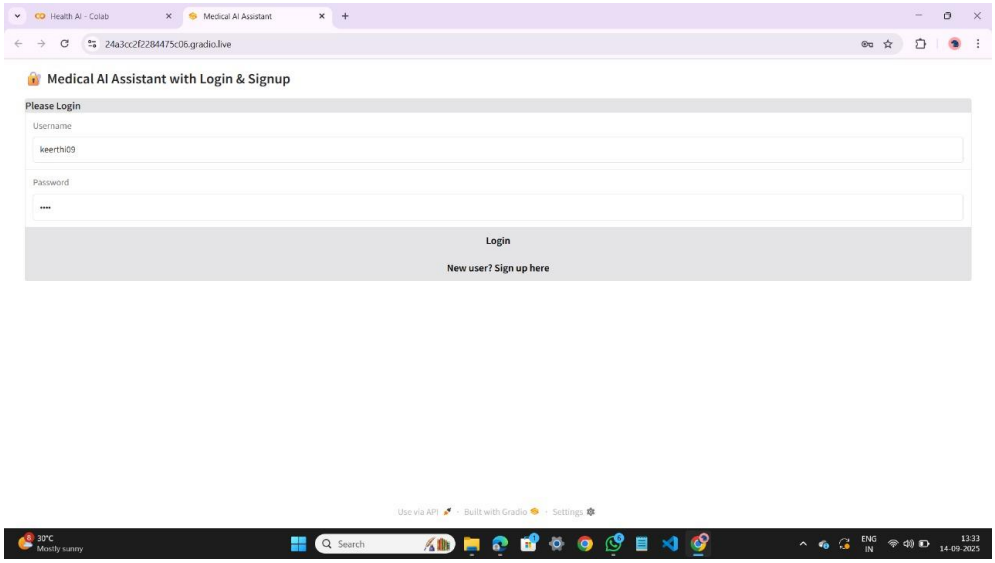
11. Screenshots

The following screenshots demonstrate the user interface:

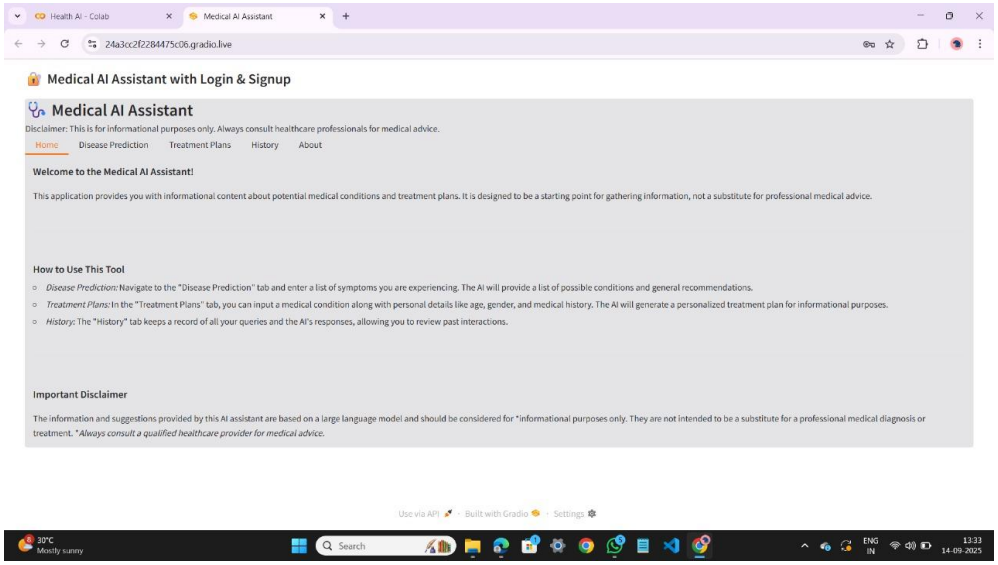
• Signup Page –



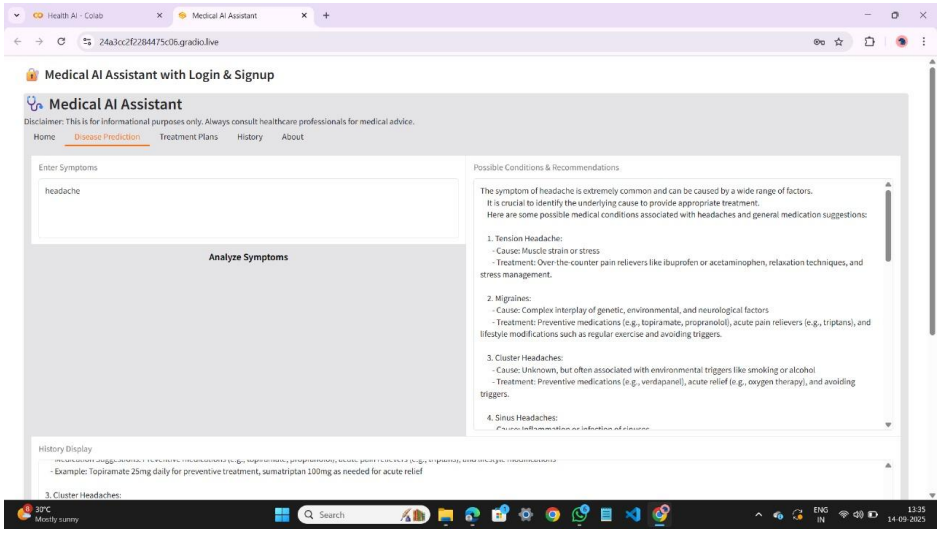
• Login Page–



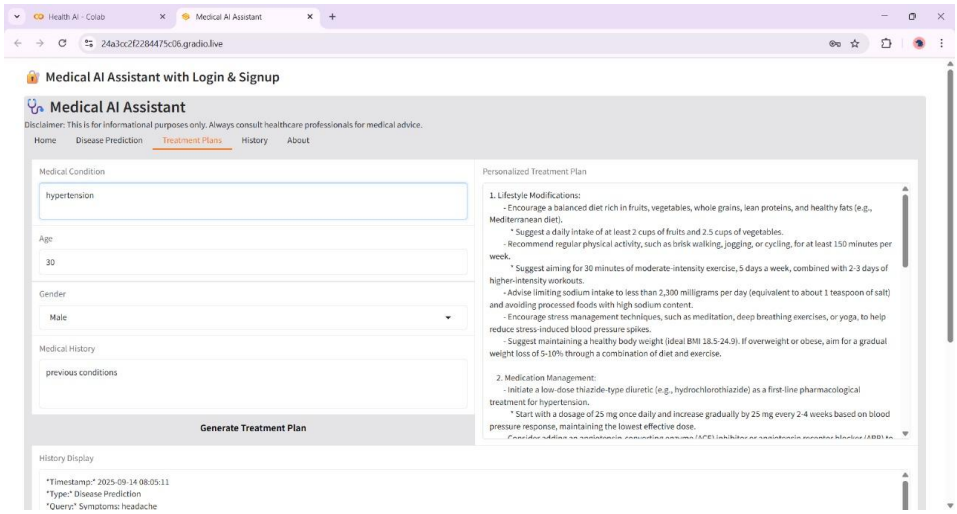
• Home Tab –



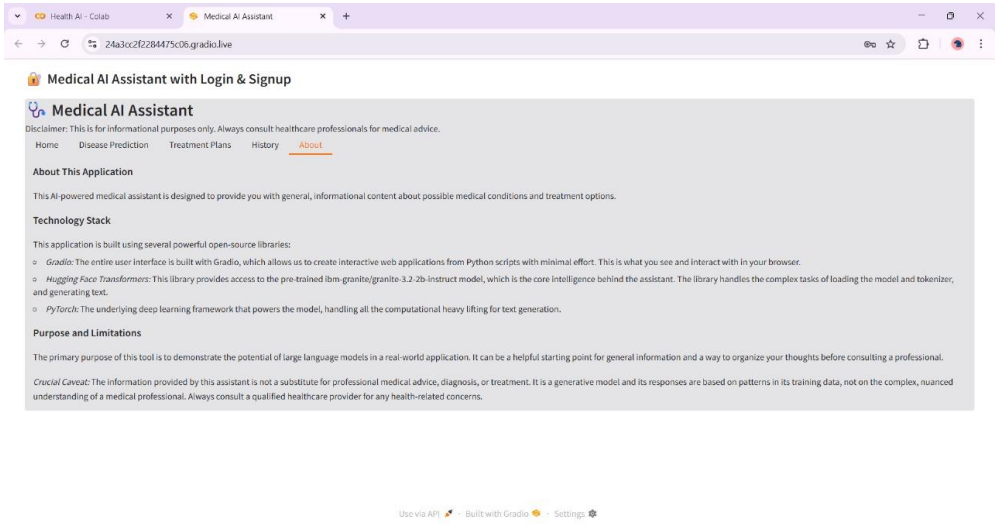
• Disease Prediction Tab –



• Treatment Plan Tab –



• About Tab –



12. Known Issues

• Predictions are AI-generated and not a replacement for medical advice
• Requires internet connectivity for Hugging Face model access
• No persistent storage of user accounts or history (session-based only)
• Basic authentication without encryption

13. Future Enhancements

• Database integration for persistent user data
• Enhanced authentication (e.g., JWT, OAuth2)
• Mobile app version for easier access
• Multi-language support for wider usability
• Integration with hospital/healthcare APIs for real-world applications