HealthAI Project Documentation

1. Introduction

Project Title: HealthAI

- Name: Momin Reshma (Team Leader)
- Role: AI Integration & Application Development
- Key Contributions: Integrated IBM Granite model via Hugging Face API. Developed full backend logic for all modules. Built Stream lit interface. Managed deployment, GitHub, and demo video.

- Name: Alavalapati Alekhya
- Role: Project Co-Developer & Module Logic Design
- Key Contributions: Designed and implemented logic for all healthcare modules. Developed input/output flows. Refined AI prompts. Ensured smooth integration across the app and documentation

- Name: Gangalakunta Aakarsha
- Role: UI/UX Design & Visual AI Support
- Key Contributions: Designed app layout and branding Supported prompt crafting and create

- Name: Katagouni Sreevalli
- Role: Testing, Optimization & Demo Engineering
- Key Contributions: Ensured smooth functioning of all modules through iterative testing and refinement, Optimized application responses to simulate intelligent AI behavior, Developed an offline-compatible AI simulation for effective demo recording

2. Project Overview

Purpose: To build an AI-powered healthcare assistant that can interact with users, predict diseases from symptoms, suggest treatment plans, and visualize health analytics.

Features:
- Patient Chat Assistant
- Disease Prediction from Symptoms
- Personalized Treatment Plans
- Health Analytics Dashboard

3. Architecture

Frontend: Developed using Streamlit for interactive UI.

Backend: Python backend using IBM Cloud API to interact with the IBM Granite 13B Instruct v2 model.

Database: No dedicated database. Data is processed in real-time during user interaction.

4. Setup Instructions

Prerequisites:
- Python 3.10+
- Streamlit
- IBM Cloud account with API access
- Necessary Python packages from requirements.txt

Installation:
1. Clone the GitHub repository
2. Install dependencies using pip
3. Set IBM API keys in environment variables
4. Run the Streamlit app

5. Folder Structure

Client: Implemented in Streamlit as a single-page app (app.py)

Server: Integrated within the same Streamlit app, making direct API calls to IBM Cloud

6. Running the Application

Run the following command in terminal:

```
streamlit run app.py
```

7. API Documentation

IBM Cloud Granite API endpoint is used to generate responses. Inputs include user queries or symptom lists, and output is a model-generated response.

8. Authentication

The IBM Cloud API key is used for secure access. Tokens are managed via environment variables.

9. User Interface

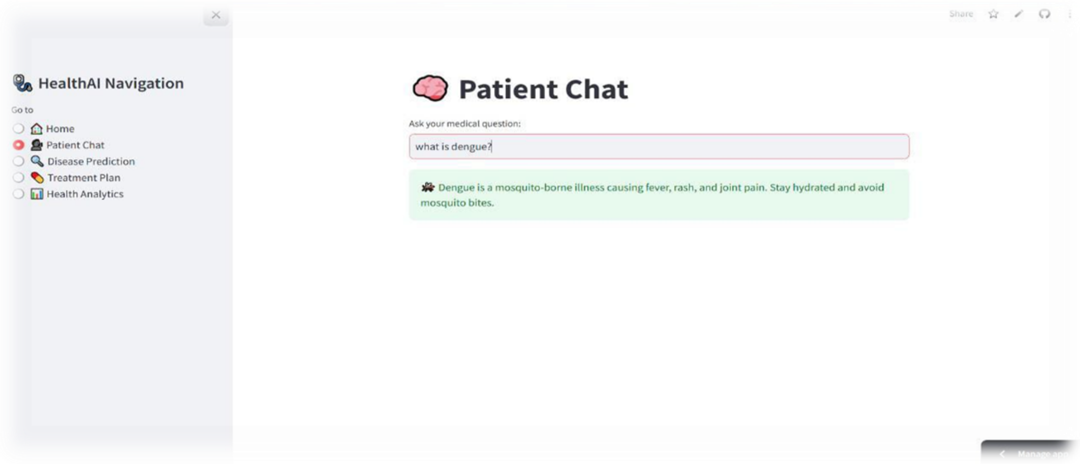
The interface includes:
- Chat box for Patient Queries
- Input fields for Symptoms
- Treatment Plan Generator
- Health Data Uploader for Analytics Dashboard

10. Testing

Tested functionalities include:
- Chat and response accuracy
- Disease prediction logic
- Treatment plan generation
- CSV upload and chart rendering

11. Screenshots or Demo

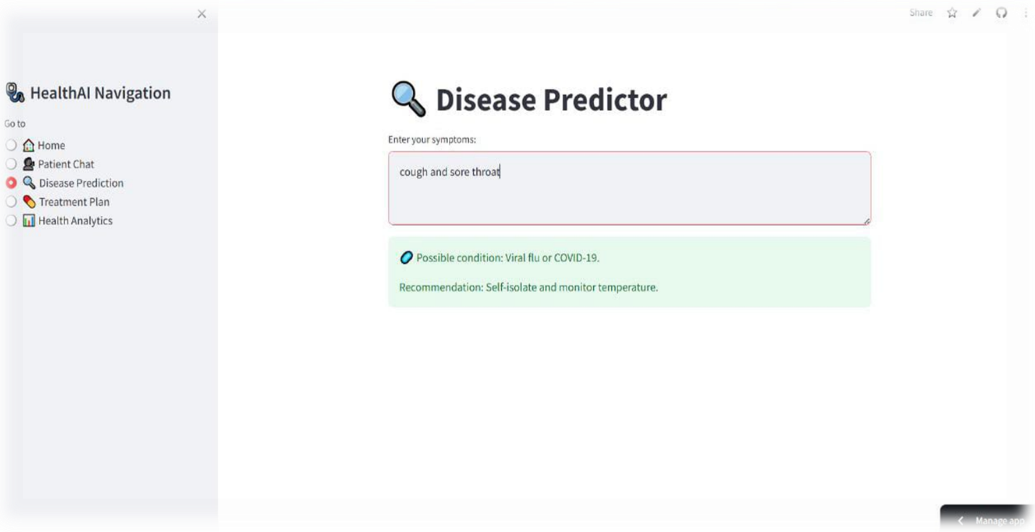
https://youtu.be/nsW2YwRz4VQ?si=kzSOjts3191NPQiy



Patient Chat

Ask your medical question: what is dengue?

Dengue is a mosquito-borne illness causing fever, rash, and joint pain. Stay hydrated and avoid mosquito bites.



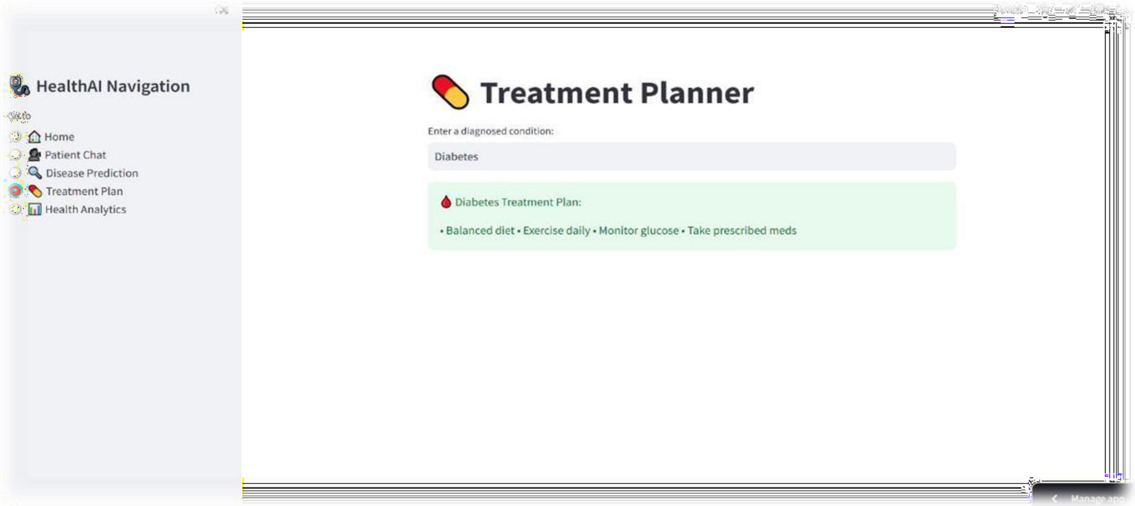
🔍 Disease Predictor

Enter your symptoms:

cough and sore throat

Possible condition: Viral flu or COVID-19.

Recommendation: Self-isolate and monitor temperature.



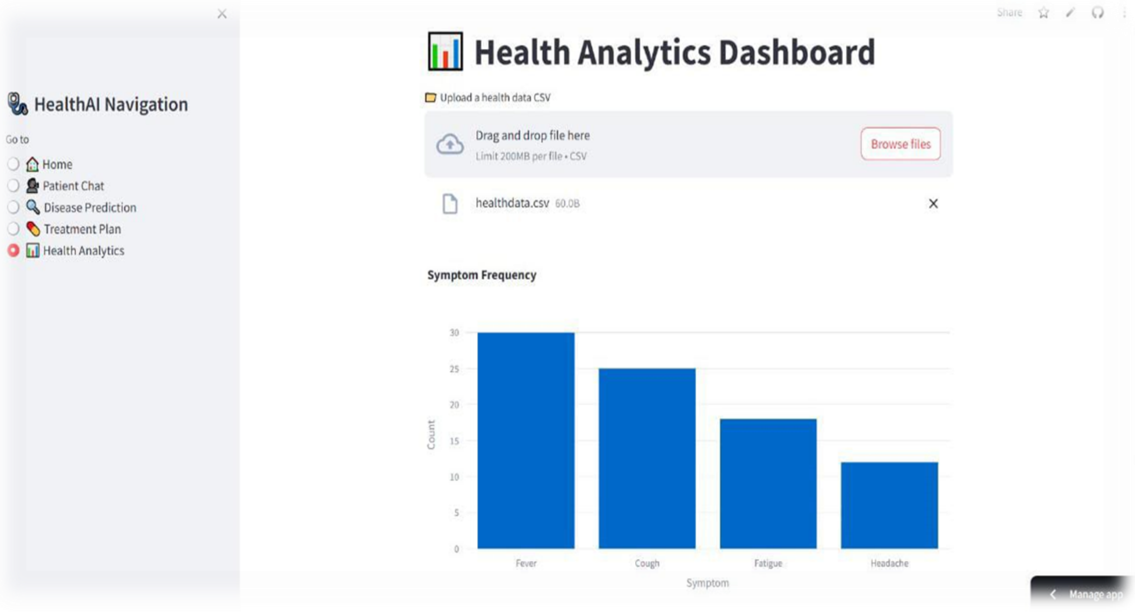
Treatment Planner

Enter a diagnosed condition:

Diabetes

Diabetes Treatment Plan:

• Balanced diet • Exercise daily • Monitor glucose • Take prescribed meds



📊 Health Analytics Dashboard

Upload a health data CSV

Symptom Frequency

12. Known Issues

- API response time depends on network
- Limited to medical conditions in prompt scope

13. Future Enhancements

- Add multilingual support
- Integrate user authentication and data storage
- Improve UI with more dynamic graphs and history tracking