Health AI - Project Documentation

1. Introduction

• Project Title: Health AI

• Team Leader:Nithish kumar .M

• Team member: Karthikeyan.S

• Team member:Madhan.S

• Team member:Elumalai.E

Health AI is designed as a Medical AI Assistant to provide disease predictions and treatment recommendations based on user-entered symptoms. The system serves as an informational support tool to assist patients and healthcare professionals by leveraging AI to suggest possible conditions and care plans.

2. Project Overview

• Purpose:

The purpose of Health AI is to support citizens with quick health insights. By entering symptoms, users can get possible conditions and recommended treatment plans. This system is for educational and informational purposes and does not replace professional medical advice.

• Features:

- Symptom-based disease prediction
- Suggested treatment plans
- Conversational interface
- Gradio-based UI
- Informational medical assistant

3. Architecture

• Frontend: Gradio-based interface

• Backend: Python logic for analyzing symptoms and retrieving predictions

• AI Integration: Hugging Face models for disease prediction

• Modules: Disease Prediction, Treatment Plans

• Hosting: Hugging Face Spaces

4. Setup Instructions

Prerequisites:

- Python 3.9 or later
- pip and virtual environment tools
- Internet access

Installation Process:

- Clone the repository
- Install dependencies from requirements.txt
- Run the Gradio app (app.py)
- Open the Hugging Face Space URL

5. Folder Structure

- app.py : Main application entry point
- requirements.txt : Dependencies
- /ui : Frontend Gradio interface components
- /api : Backend modules for analysis
- /assets : Screenshots and static files

6. Running the Application

- Launch the backend by running: python app.py
- Open Hugging Face Spaces hosted link
- Enter symptoms in Disease Prediction tab
- View predicted conditions and recommendations
- Open Treatment Plans tab for detailed guidance

7. User Interface

The Health AI interface includes:

- Disease Prediction tab (symptom input + condition results)
- Treatment Plans tab
- Real-time analysis
- Clean design for accessibility

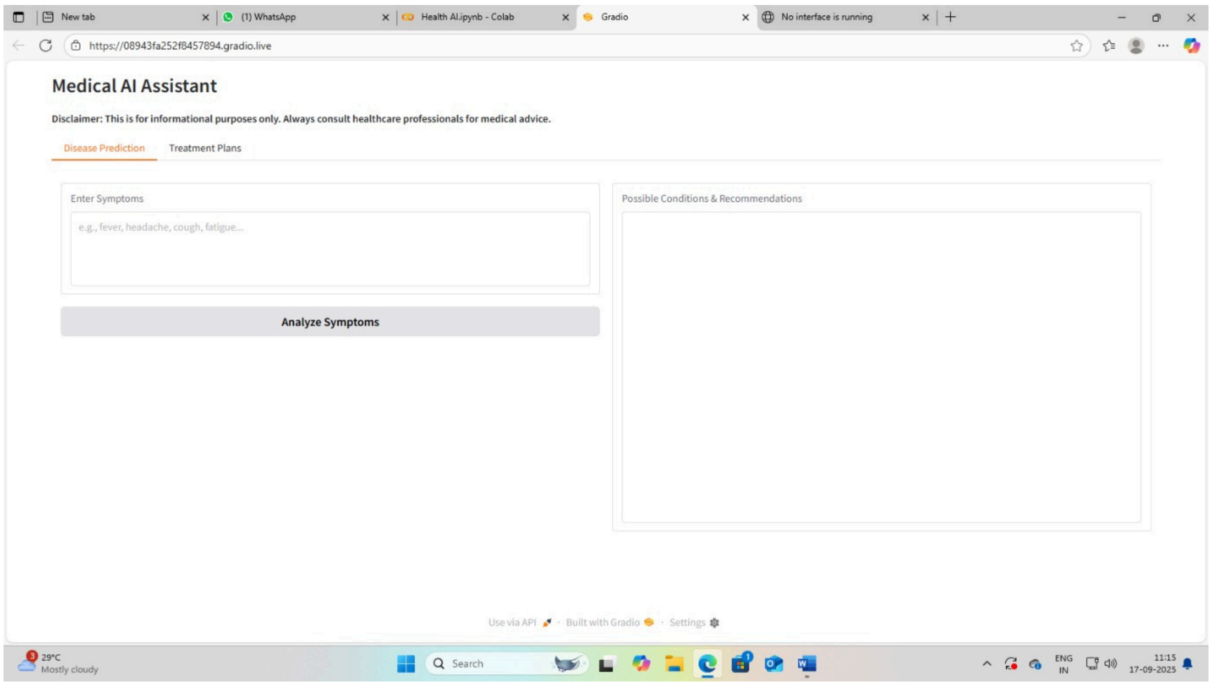
8. Testing

Testing performed includes:

- Unit Testing: Symptom analysis functions
- Manual Testing: Interface workflows
- API Testing: Model predictions

- Edge Case Handling: Invalid or incomplete symptom inputs

9. Screenshots



Medical AI Assistant

Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.

10.

https://drive.google.com/file/d/1hG5eeBxXsdHW_ett8qIVzKkPI3_05uyg/view?usp=drive_link

Demo link

11. Future Enhancements

- Integration with real-world medical datasets
- Advanced diagnostic recommendations
- Multi-language support for rural outreach
- Secure patient data handling
- Integration with wearable devices for real-time monitoring