Project Title

# Medical AI Assistant - Project Documentation

1. **Introduction**
   * Project Title: Medical AI Assistan
   * Team Leader : Bharath.V
   * Team Member :Ashwin .V.P
   * Team Member :Kesavan .C
   * Team Member :Dheeranbala .M

# Project Overview

* + Purpose:

The purpose of the Medical AI Assistant is to provide an AI-powered web interface where users can enter symptoms and receive possible conditions and recommendations. It is intended purely for informational use and does not replace professional medical consultation.

* + Features:
* Symptom Input Box: Users can type symptoms like fever, cough, headache.
* Disease Prediction: AI model analyzes symptoms and suggests possible conditions.
* Treatment Plan Tab: Provides basic treatment recommendations.
* User-Friendly Interface: Built with Gradio, ensuring accessibility.
* Disclaimer: Clear note that it is not a substitute for professional advice.

# Architecture

Frontend (Gradio): Provides simple and interactive UI with tabs (Disease Prediction & Treatment Plan).

Backend (FastAPI/Transformers): Processes input symptoms and generates predictions. Model (Hugging Face LLMs): Utilizes pretrained models for mapping symptoms to possible conditions.

# Setup Instructions

* Install Python 3.9 or later.
* Clone the repository.
* Install dependencies from requirements.txt.
* Run app.py to launch the Gradio app.
* Open the localhost or public Gradio link to access the assistant.

# Folder Structure

app.py – Main script for launching the application. requirements.txt – Python dependencies.

models/ – Stores AI/ML models for predictions.

utils/ – Utility scripts for preprocessing and predictions.

# Running the Application

* Run app.py to start the Medical AI Assistant. ➢ Enter symptoms in the text box (e.g., fever, cough). ➢ Click 'Analyze Symptoms' to view possible conditions. ➢ Navigate to the 'Treatment Plan' tab for recommendations.

# API Documentation

POST /analyze-symptoms – Accepts symptoms text and returns possible conditions. GET /treatment-plan – Returns general treatment suggestions.

# Authentication

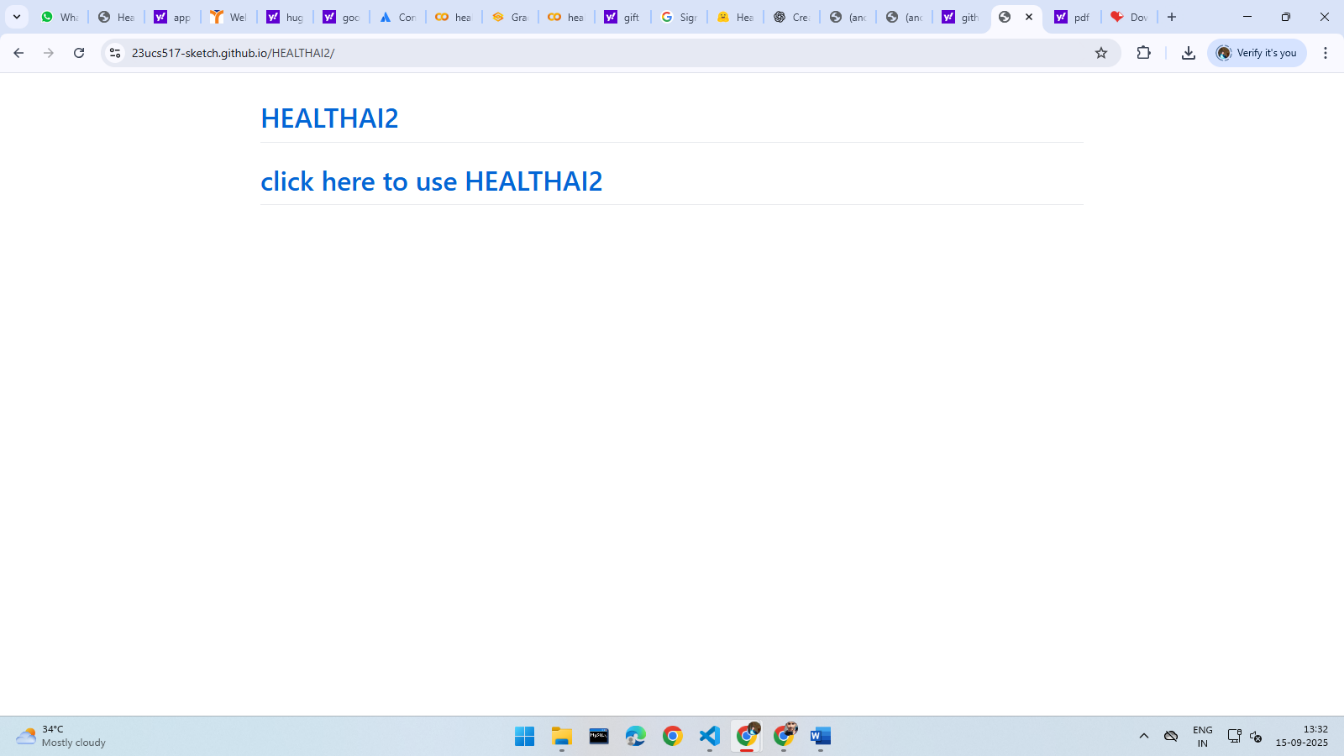
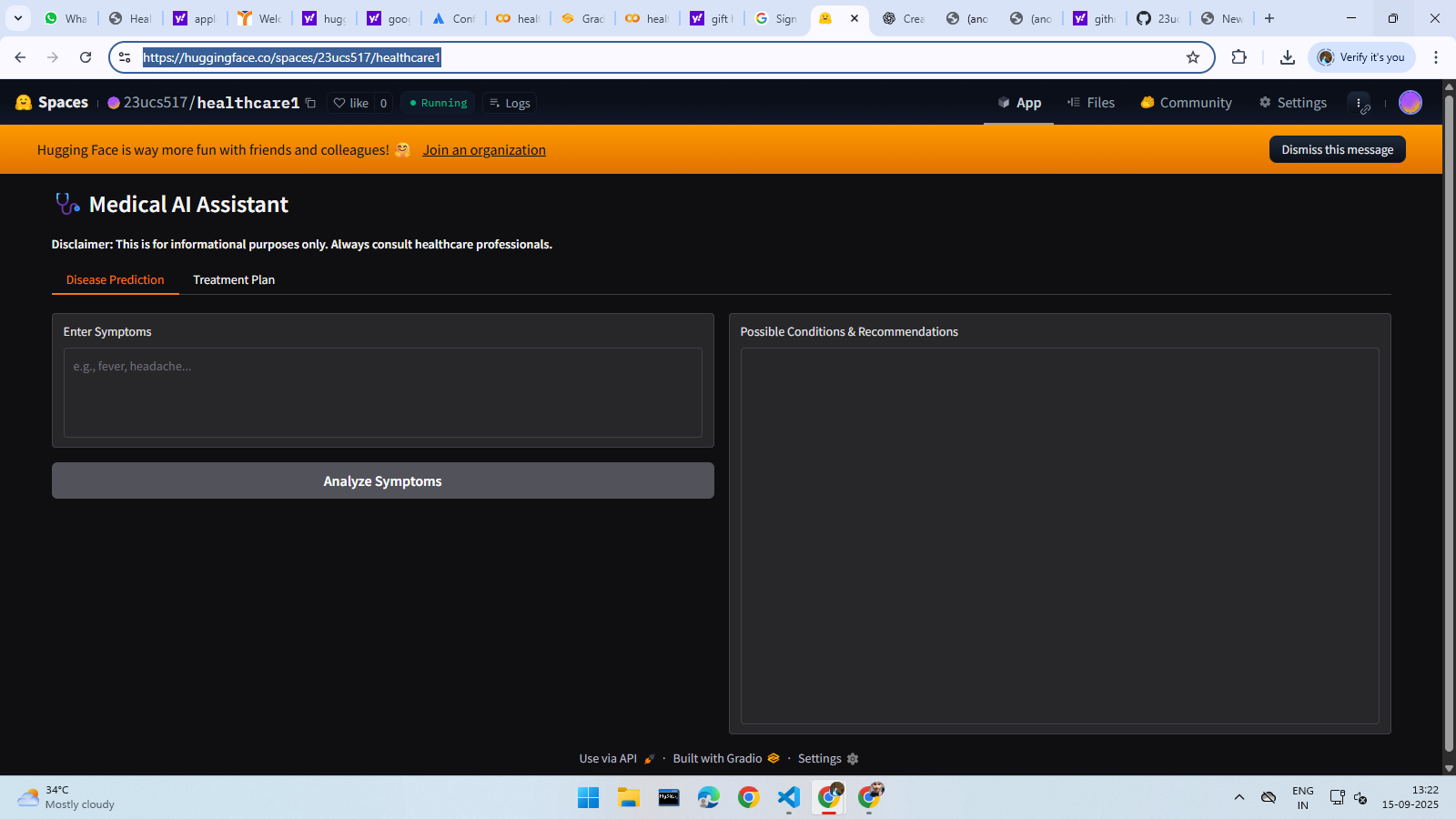
This demo does not include authentication. In production, secure methods such as JWT tokens, API keys, or OAuth2 should be implemented.

# User Interface

The UI consists of two main tabs: - Disease Prediction: Symptom input and analysis. - Treatment Plan: Displays general treatment recommendations. The design is lightweight, intuitive, and responsive for easy use.

# Testing

Testing was carried out with various inputs to validate predictions and ensure stability. Unit tests were created for symptom parsing and backend APIs. Edge cases (empty input, invalid text) were also tested for robustness.



# Future Enhancements

* Integration with real medical knowledge bases.
* Support for multiple languages.
* Advanced treatment recommendations with dosage guidance.
* User login and health history tracking.
* Mobile application for wider accessibility.