Health AI: Intelligent Healthcare Assistant Using IBM Granite

# 1. Introduction

Project Title: Health AI: Intelligent Healthcare Assistant

Team Members:

• SHREE VAMISHI P K,

• ANUMANTH NARAYAN S,

• GILBERT JOSHUA A,

• THANANJAYAN P,

# 2. Project Overview

Purpose: The purpose of Health AI is to provide a smart, accessible, and user-friendly healthcare assistant. The system uses IBM Granite models from Hugging Face to support patient chat, disease prediction, and treatment plan suggestions. It's designed to be lightweight and fast, deployed on Google Collab with GPU support to ensure quick and secure access for healthcare guidance.

Features:

• Patient Chat: Provides conversational AI support for patient queries.

• Disease Prediction: Offers AI-driven predictions based on user-provided symptoms.

• Treatment Plans: Suggests personalized remedies and care plans.

• Lightweight & Fast: Optimized for Google Collab with GPU acceleration.

# 3. Architecture

• Frontend (Gradio): A simple, web-based user interface for all user interactions.

• Backend (Python + Transformers): This is the core of the application, handling predictions, chat functionality, and treatment generation using the IBM Granite models.

• Deployment (Google Collab): The application runs on a cloud-based environment with GPU acceleration.

# 4. Setup Instructions

Prerequisites:

• Python 3.9 or later

• Gradio Framework

• Transformers & Torch libraries

• IBM Granite models (via Hugging Face)

• Google Collab T4 GPU access

Installation Process:

1. Open Google Collab and create a new notebook.

2. Change the runtime type to T4 GPU.

3. Install the necessary dependencies by running the following command: !pip install transformers torch gradio -q

4. Import the IBM Granite model from Hugging Face.

5. Run the provided project code.

# 5. Project Workflow

• Activity 1: Exploring the Naan Mudhalvan Smart Interz Portal.

• Activity 2: Choosing an IBM Granite model from Hugging Face.

• Activity 3: Running the Application in Google Colab.

# 6. API Documentation

• POST /predict-symptoms: Accepts user symptoms and returns possible medical conditions.

• POST /generate-treatment: Generates a treatment plan based on a condition, age, and medical history.

• GET /chat-response: Provides conversational support for patient queries.

# 7. User Interface

• Disease Prediction Tab: Allows users to input symptoms and receive a list of possible conditions with recommendations.

• Treatment Plans Tab: Accepts patient details such as age, gender, and medical history to generate personalized treatment guidance.

• Minimal & User-friendly: The interface, built with Gradio, is designed for simplicity and ease of use.

# 8. Testing

• Unit Testing: Verified that the symptom prediction and treatment generation functions work correctly.

• API Testing: Checked API responses using various test inputs to ensure consistency and reliability.

• Manual Testing: Validated predictions by entering real-world symptom sets (e.g., fever, cough, cold).

• Edge Cases: Tested the system's ability to handle empty inputs, unknown symptoms, and invalid data.

# 9. Known Issues

• Limited to text-based interaction (no voice input).

• Predictions are dependent on the available training data; therefore, rare diseases may not be covered.

• Requires a stable internet connection due to its Google Collab dependency.

# 10. Future Enhancements

• Integration with real-time health monitoring devices.

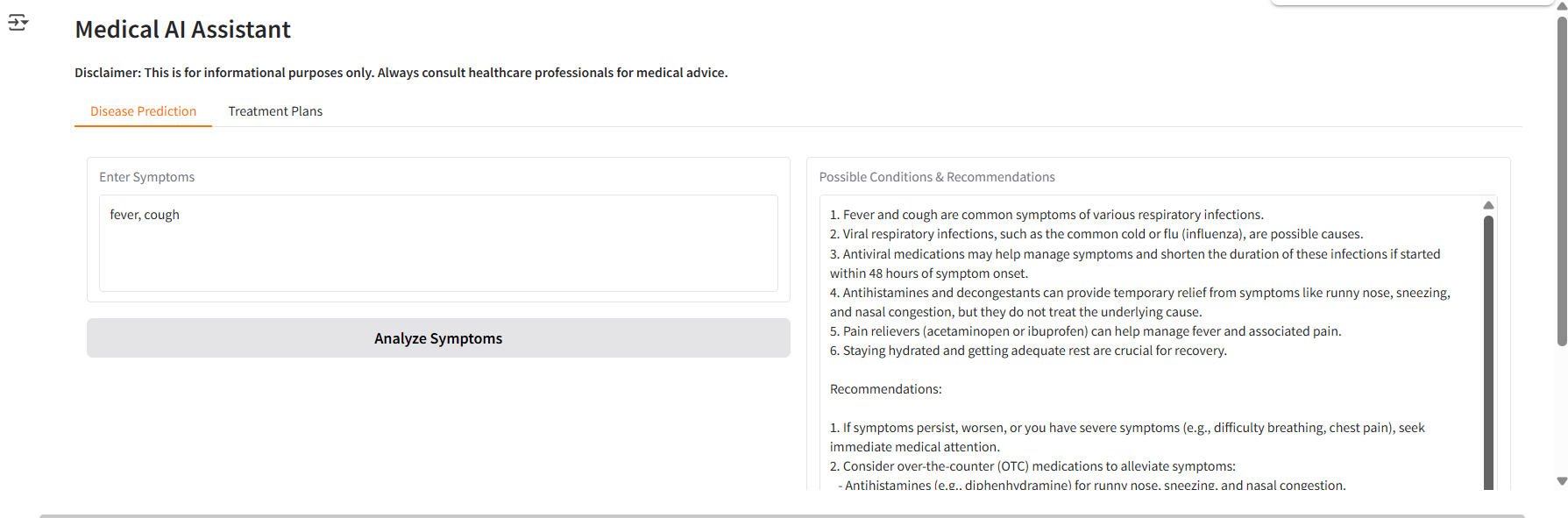
• Expansion of the disease prediction database.

• Addition of multilingual support for global accessibility.

• Deployment on scalable cloud platforms beyond Collab for broader availability.

# 11. Output Screenshots

📌 Figure 1: Disease Prediction Output



📌 Figure 2: Treatment Plan Output

