Project Documentation

1. Introduction

Project Title: Health AI:Intelligent Healthcare Assistant Using IBM Granite

Team member: Deeparathna P
Team member: Deepa L
Team member: Deepthika H
Team member: Deepika K

2. Project Overview

Purpose:

The purpose of the Medical AI Assistant is to provide users with quick, AI-powered guidance related to health symptoms and medical conditions.

It improves healthcare awareness and accessibility while emphasizing the importance of professional consultation.

Features:

- Disease Prediction: Analyze symptoms and provide possible conditions with recommendations.
- Treatment Plan Generation: Personalized suggestions based on condition, age, gender, and medical history.
- User-Friendly Interface: Built with Gradio for easy interaction.
- AI-Powered Analysis: Uses IBM Granite LLM for natural language processing.

3. Architecture

Frontend (Gradio): Interactive web interface with tabs for disease prediction and treatment planning.

Backend (Python & Hugging Face): Handles model loading, prompts, and responses.

LLM Integration (IBM Granite): Uses ibm-granite/granite-3.2-2b-instruct for medical text generation.

4. Setup Instructions

Prerequisites:

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

Installation:

1. Clone the repository
2. Install dependencies from requirements.txt
3. Run: python healthai.py
4. Open the Gradio link in your browser

5. Folder Structure

project/
■ ■ healthai.py # Main application script
■ requirements.txt # Dependencies
■ ■README.md # Documentation
■
■ ■
■

6. Running the Application

1. Run python healthai.py
2. Open the Gradio link in browser
3. Navigate tabs:
- Disease Prediction: Enter symptoms → Analyze
- Treatment Plan: Enter details → Generate

7. API Documentation

Currently not applicable. Future versions may include REST APIs.

8. Authentication

Current version runs without authentication.
Future deployment may include:
- API key authentication
- Role-based access

9. User Interface

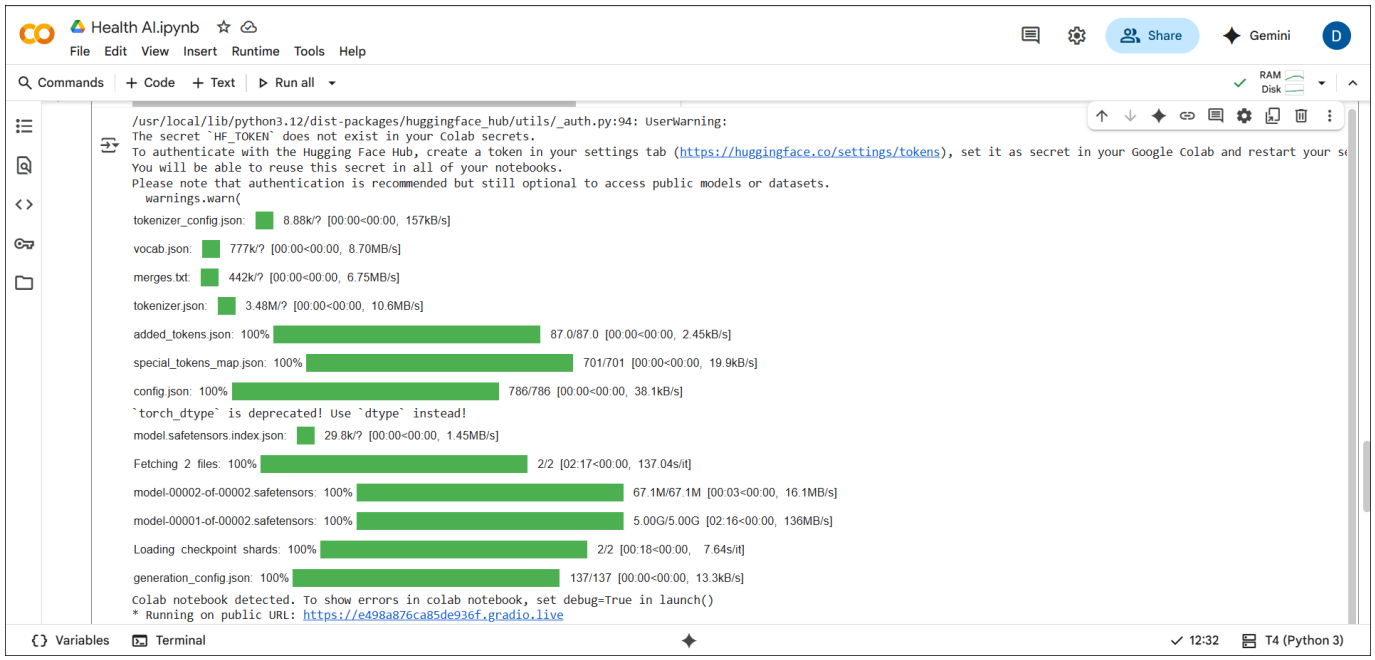
- Clean two-tab layout (Disease Prediction & Treatment Plans)
- Textbox input and detailed text output
- Disclaimer for medical safety

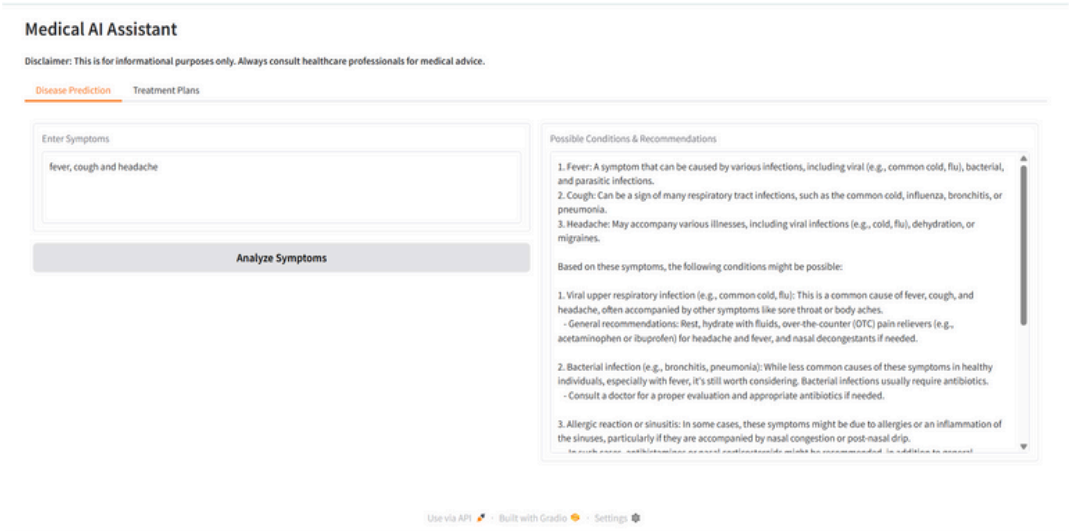
10. Testing

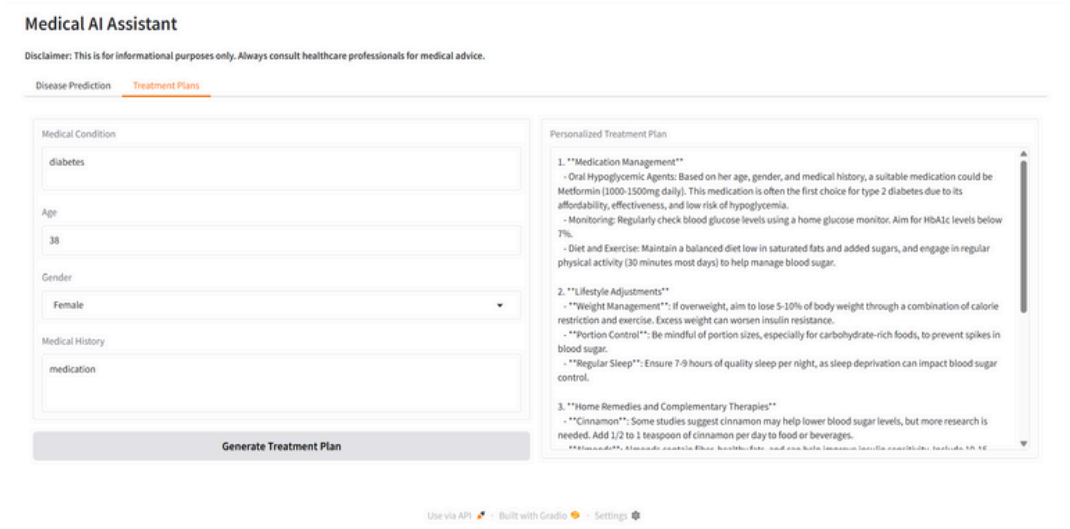
- Unit Testing: Prompt and response functions
- Manual Testing: Symptom entry and treatment plan generation
- Edge Cases: Empty or unusual inputs

11. Screenshots

Below are the screenshots of the Gradio interface of the Medical AI Assistant.


Health AI.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
RAM
Disk
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
tokenizer_config.json: 8.88k/? [00:00<00:00, 157kB/s]
vocab.json: 777k/? [00:00<00:00, 8.70MB/s]
merges.txt: 442k/? [00:00<00:00, 6.75MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 10.6MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 2.45kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 19.9kB/s]
config.json: 100% 786/786 [00:00<00:00, 38.1kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 1.45MB/s]
Fetching 2 files: 100% 2/2 [02:17<00:00, 137.04s/it]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:03<00:00, 16.1MB/s]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [02:16<00:00, 136MB/s]
Loading checkpoint shards: 100% 2/2 [00:18<00:00, 7.64s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 13.3kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://e498a876ca85de936f.gradio.live
Variables
Terminal
12:32
T4 (Python 3)





12. Known Issues

- Responses may vary with inputs
- No guaranteed medical accuracy
- Long symptom entries may be truncated

13. Future Enhancements

- Add API support for mobile apps
- Enable voice input/output
- Improve predictions with medical datasets
- Add data visualization features