Health AI: Intelligent Healthcare Assistant

Generative AI with IBM

1. Introduction

Project title: Health AI: Intelligent Healthcare Assistant

Team Member: Maheswar K Dileesh

Team Member: Pursothaman P

Team Member: Dinesh B

Team Member: Kathikeyan P

1. Project Overview

Purpose:

Health AI harnesses IBM Watson Machine Learning and Generative AI to provide intelligent healthcare assistance, offering users accurate medical insights. The platform includes Patient Chat for general queries, Disease Prediction for symptom evaluation, Treatment Plans for personalized medical guidance, and Health Analytics for monitoring patient health trends. By leveraging IBM’s Granite-13b-instruct-v2 model, it empowers individuals to make informed health decisions with confidence while ensuring responsible data handling.

Features:

Patient Chat:

* Key Point: Conversational health guidance
* Functionality: Provides AI-driven responses to general health-related questions.

Disease Prediction:

* Key Point: Symptom-based diagnosis
* Functionality: Analyses reported symptoms and validates potential health conditions with probability estimates.

Treatment Plan Generator:

* Key Point: Personalized recommendations
* Functionality: Generates evidence-based treatment suggestions including lifestyle changes and medical guidelines.

Health Analytics:

* Key Point: Data visualization
* Functionality: Allows users to view health trends such as vitals, reports, and AI-driven insights.

Secure Data Handling:

* Key Point: Responsible data management
* Functionality: Ensures user health data privacy, API key protection, and ethical AI usage.

Scenarios:

* Scenario 1: A user inputs symptoms (e.g., headache, fatigue, fever). Health AI returns possible conditions with recommendations.
* Scenario 2: A patient seeks treatment advice for an existing diagnosis. Health AI suggests an evidence-based personalized plan.
* Scenario 3: A user wants to track their long-term health trends. Health AI visualizes metrics and provides preventive suggestions.

1. Architecture

Frontend (Streamlit):

Provides an intuitive dashboard with chat, symptom checker, treatment generator, and analytics visualization. Supports real-time feedback and report downloads.

Backend (Fast API):

Handles API requests including chat, treatment generation, disease predictions, and analytics processing. Optimized for high performance and Swagger documentation.

LLM Integration (IBM Watsonx Granite):

Utilizes IBM Granite-13b-instruct-v2 for NLP tasks like health question answering, predictions, and medical plan generation.

Vector Search (Optional with Pinecone):

Medical knowledge bases and documents can be embedded using Sentence Transformers for quick semantic search.

ML Modules:

* Disease prediction using trained models on symptom datasets.
* Health analytics visualization with time-series tools (pandas, matplotlib).

1. **Setup Instructions**
2. **Prerequisites:**

* **Python 3.9+**
* **pip and venv**
* **API keys for IBM Watson and Pinecone (if vector search enabled)**
* **Internet connection for cloud integration**

1. **Installation Process:**

* **Clone the repository**
* **Install requirements.txt dependencies**
* **Configure credentials in .env**
* **Run backend with FastAPI (uvicorn app.main:app --reload)**
* **Launch frontend with Streamlit (streamlit run health\_dashboard.py)**

**5.** **. Folder Structure**

* **app/ – Backend FastAPI logic**
* **app/api/ – Routes: chat, prediction, analytics, plans**
* **ui/ – Streamlit frontend pages & layouts**
* **health\_dashboard.py – Launch script for dashboard**
* **granite\_llm.py – Interface to IBM Granite for Q&A, predictions, plans**
* **disease\_predictor.py – ML-based condition checker**
* **treatment\_recommender.py – Personalized medical recommendation generator**
* **health\_analytics.py – Visualizations and metrics reporting**
* **report\_generator.py – AI-generated treatment/health reports**

**6. Running the Application**

* **Start FastAPI backend**
* **Run Streamlit frontend**
* **Use sidebar for navigation (Chat, Disease Prediction, Treatment Plans, Analytics)**
* **Upload data or input symptoms to interact with the AI assistant**
* **Download reports for treatments and insights**

**API Documentation**

* **POST /chat/ask – Accepts user queries about health and responds with AI-generated insights**
* **POST /predict-disease – Processes symptoms and returns potential conditions with confidence values**
* **POST /generate-plan – Creates tailored treatment plans for conditions**
* **GET /analytics – Returns visualizations and insights from user health data**
* **POST /feedback – Stores patient feedback**

**Swagger UI is provided for testing endpoints during development.**

**8. Authentication**

* **Token-based (JWT or API keys)**
* **Optional OAuth2 with IBM credentials**
* **Role-based access (patient, doctor, admin) planned for secure deployments**

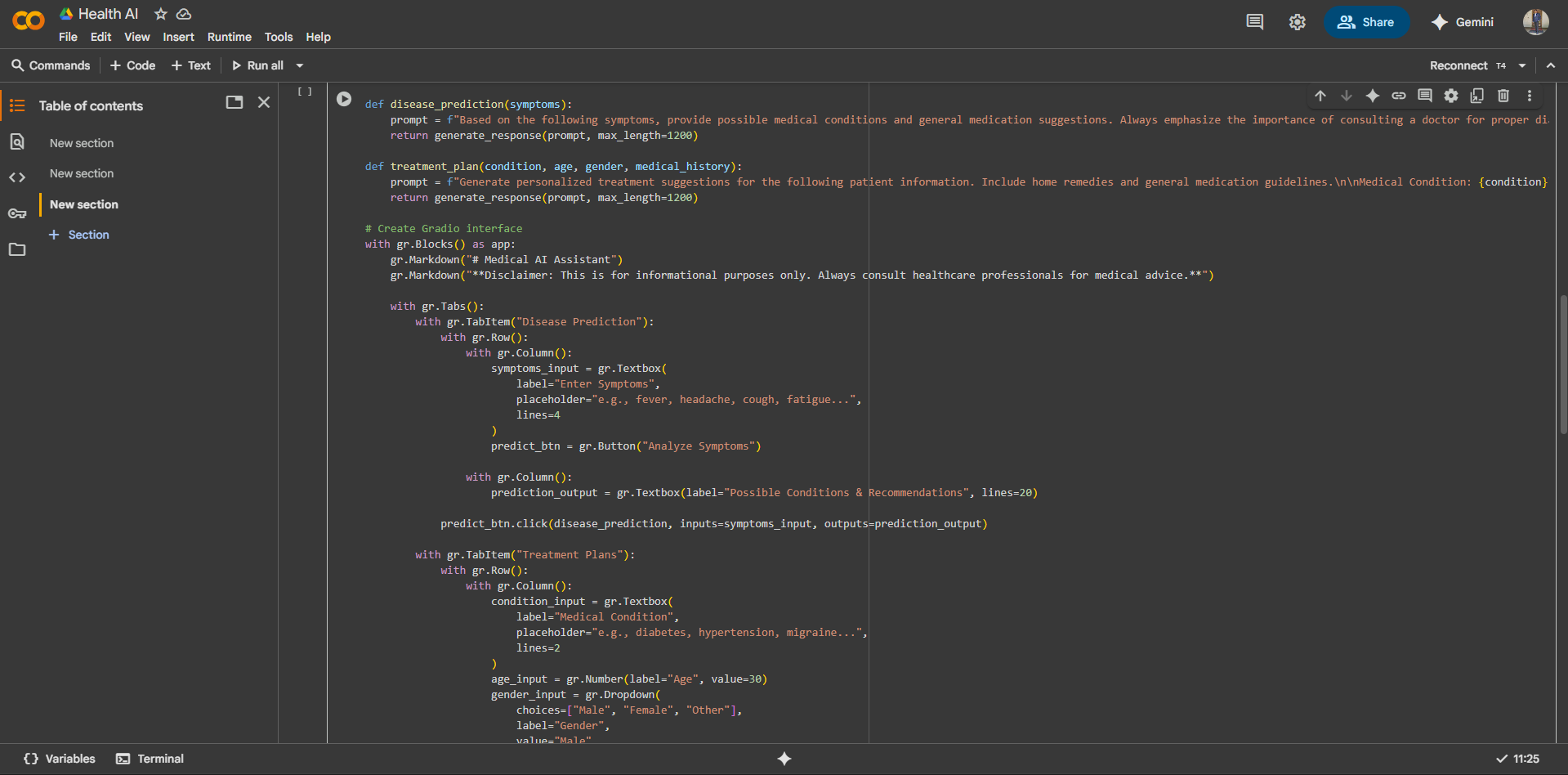
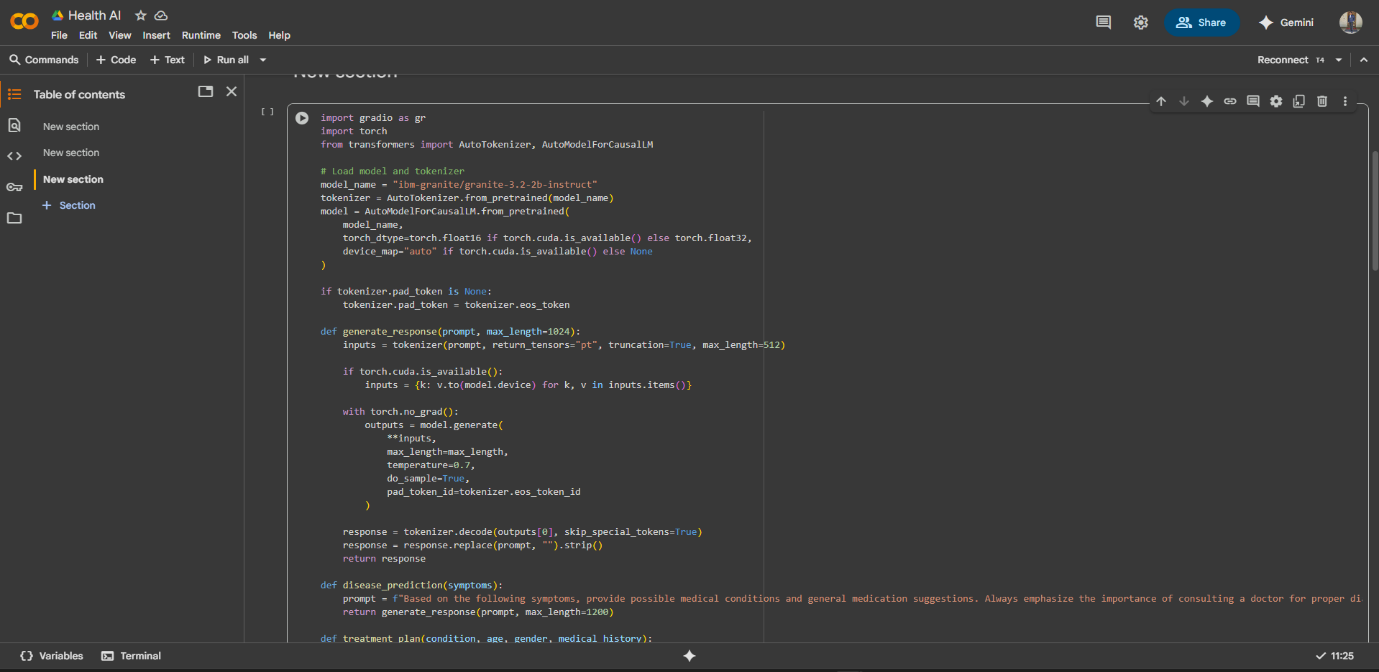
**9. User Interface**

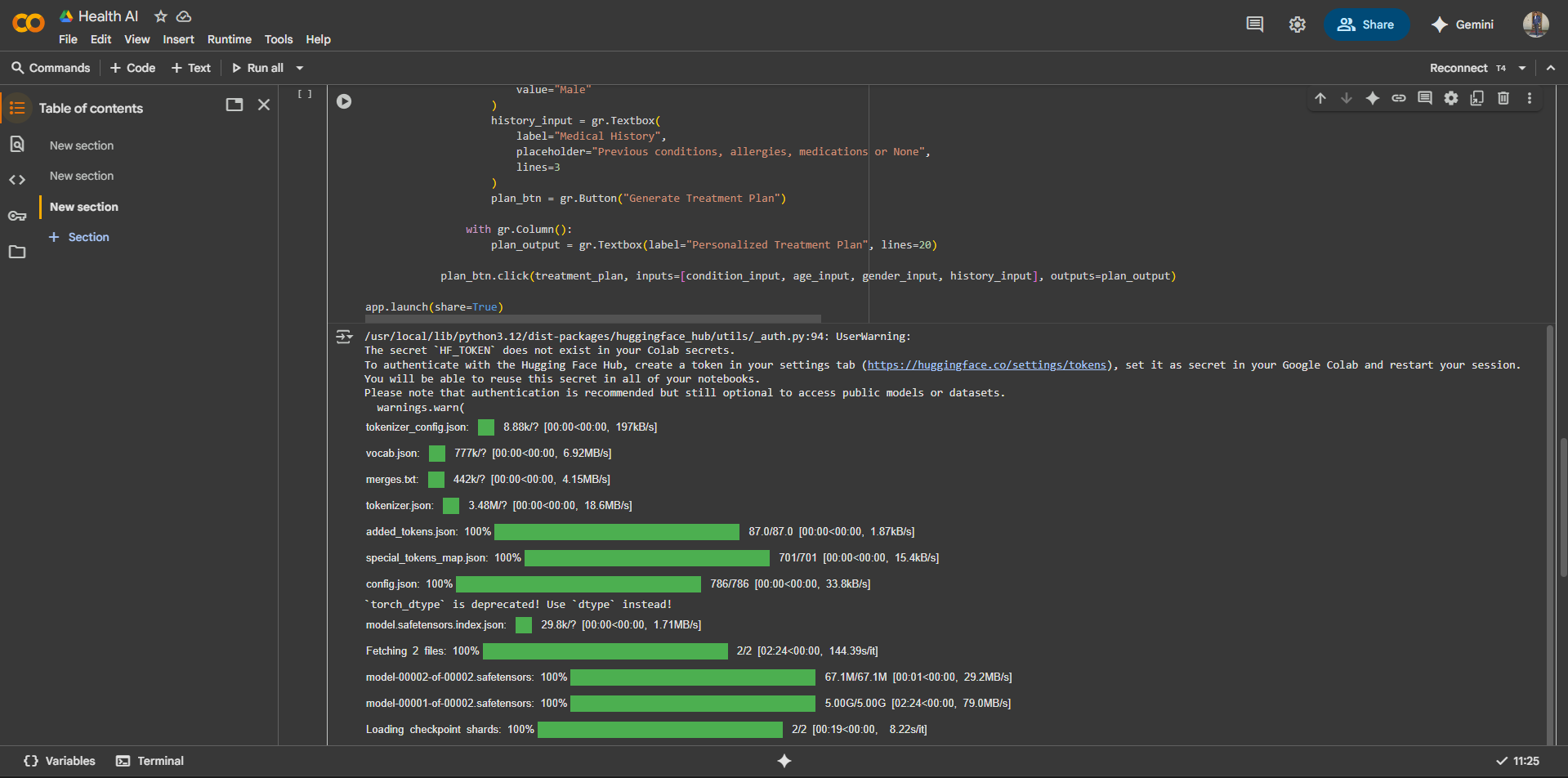
* **Sidebar navigation**
* **Patient Chat, Symptom Checker, Treatment Plans, and Health Trends modules**
* **Downloadable reports in PDF**
* **Graphs, cards, and summaries for easy interpretation**

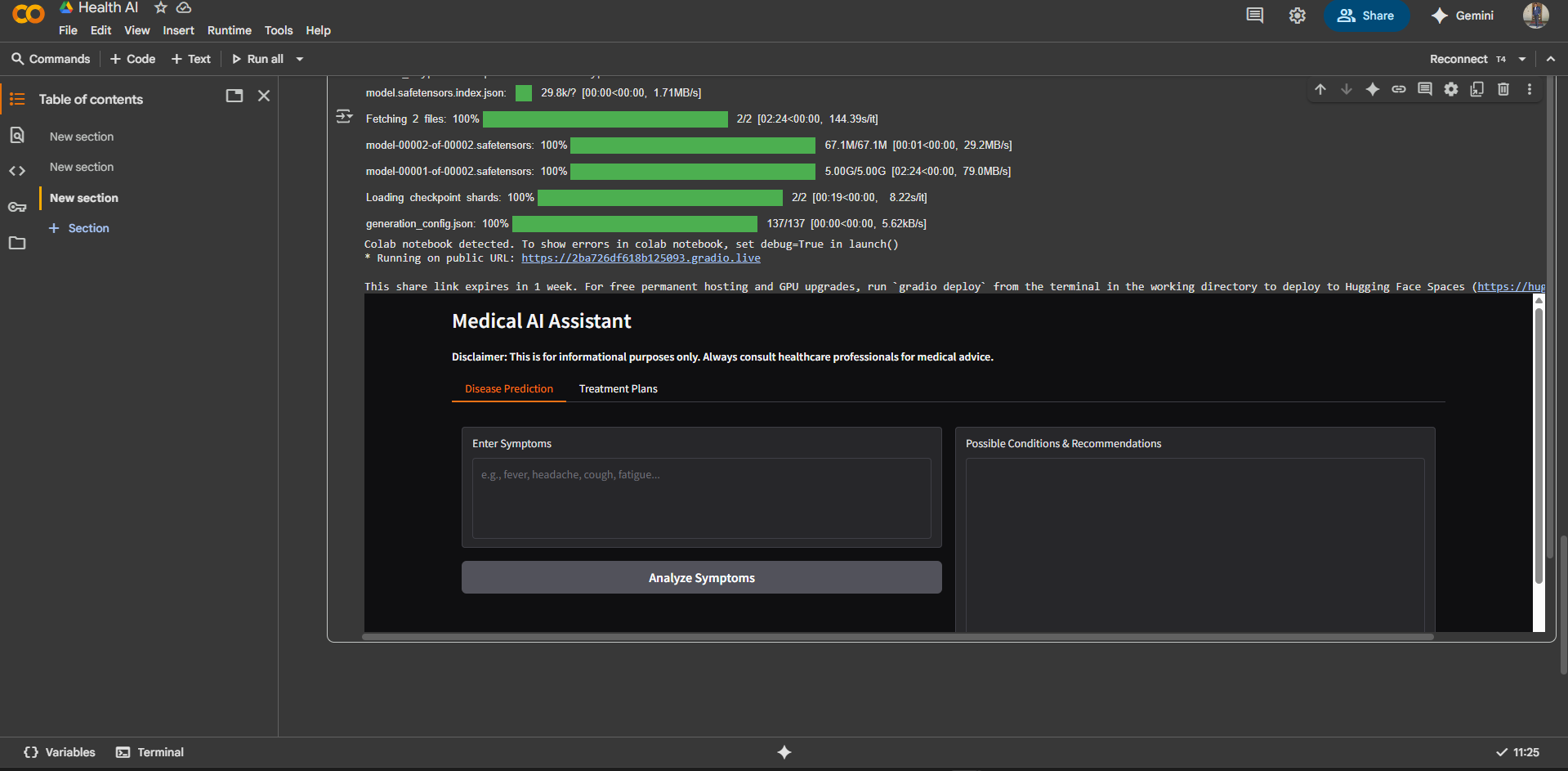
**10. Testing**

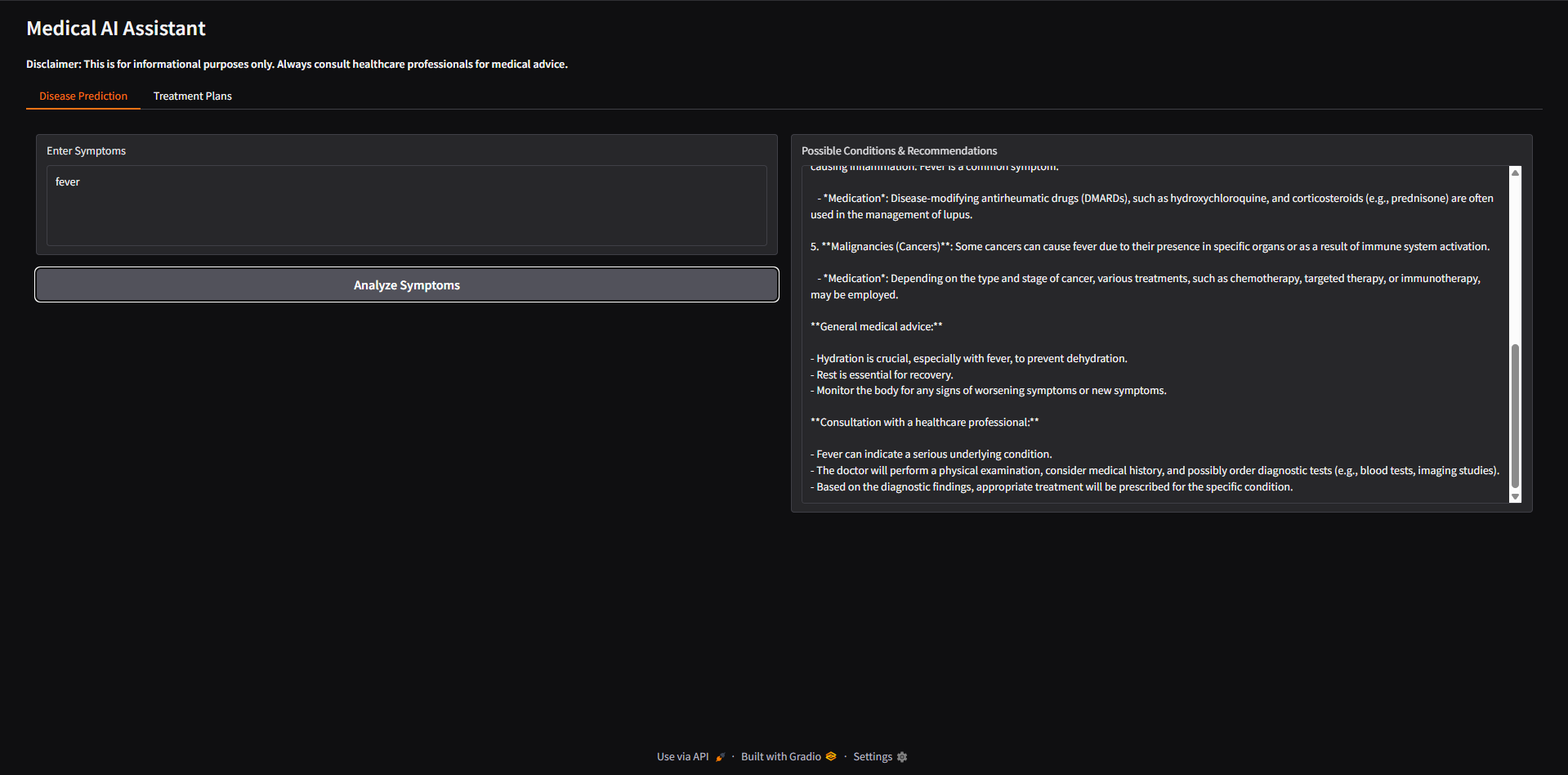
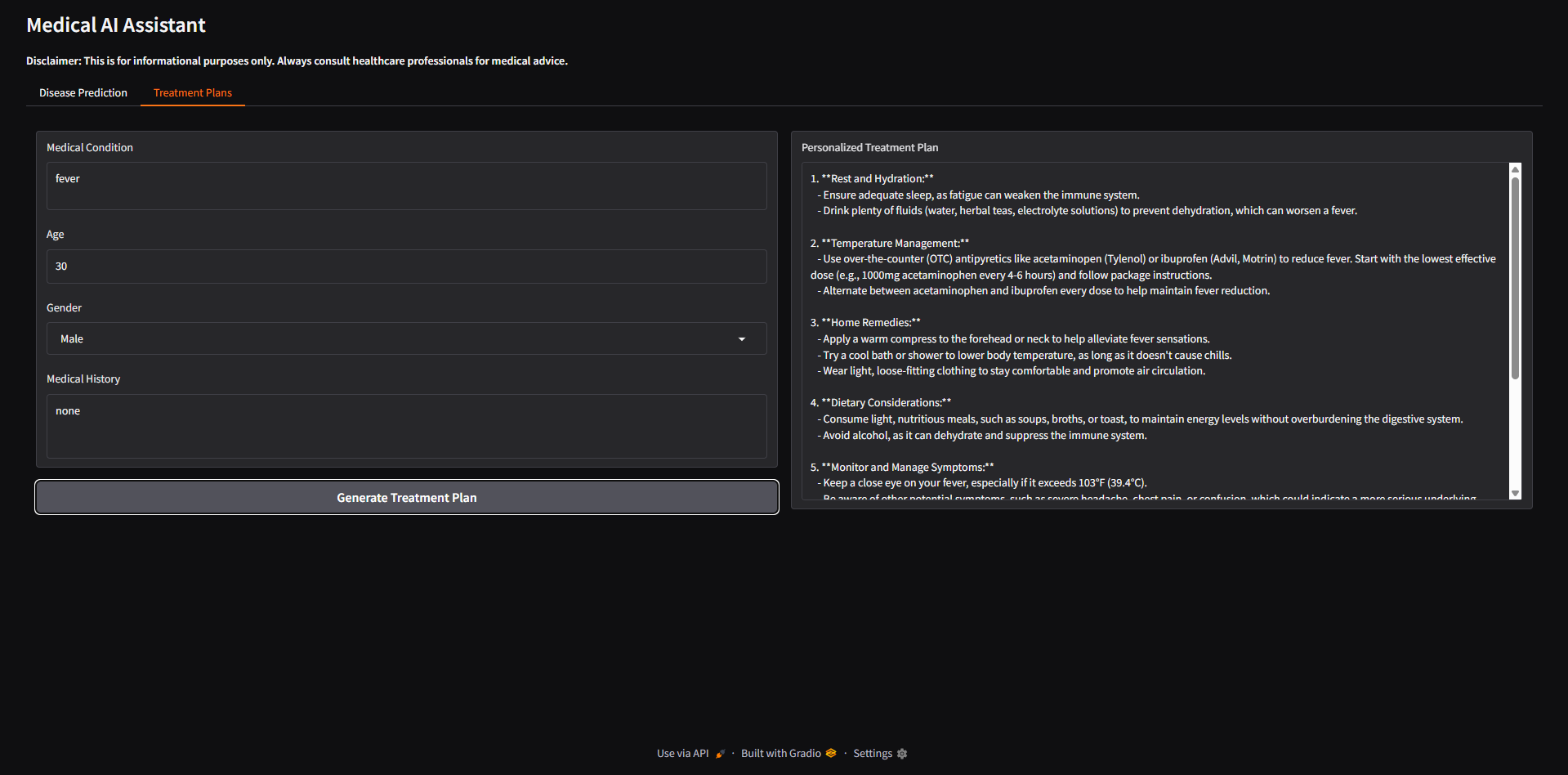
* **Unit Testing: For prediction functions and LLM prompts**
* **API Testing: Swagger UI, Postman, and pytest scripts**
* **Manual Testing: Symptom inputs, personalized plan outputs, analytics validation**
* **Edge Cases: Invalid inputs, large datasets, security failures**

**11. Screenshots**

****

****

****

****

**12. Known Issues**

* Symptom predictions require broader dataset for higher accuracy
* Treatment recommendations are advisory—not a replacement for professional medical care
* Dependence on stable internet for Granite queries

**13. Future Enhancements**

* Integration with wearable devices (IoT health trackers)
* Voice-based conversational health assistant
* Integration with EHR (Electronic Health Records) for personalized care
* Expanded medical dataset for improved accuracy