Health AI: Intelligent Healthcare Assistant



# 1. Introduction

• **Project Title:** Health AI: Intelligent Healthcare Assistant

• **Team Members**: Nithyasri P

• **Team Members:** Rohini Priya S

**• Team Members:** Subisha M

• **Team Members:** Vinothini S

# 2. Project Overview

**• Purpose:**

The purpose of Health AI is to provide intelligent, accessible, and secure healthcare assistance powered by IBM Granite LLM models. It enables users to interact via natural conversations for medical guidance, disease prediction, and treatment planning. Health AI also assists doctors with patient data insights and quick decision support.

**• Features:**

- Conversational Patient Chat (Natural language healthcare Q&A)

- Disease Prediction (AI-based risk assessment from symptoms)

- Treatment Plan Suggestions (Evidence-based guidance)

- Health Data Summarization (Simplifies reports & lab results)

- Multimodal Input Support (Text, PDFs, Images like lab reports)

- User-Friendly Gradio UI (Simple healthcare chatbot interface)

- Cloud Deployment (Runs in Google Colab for easy access)

# 3. Architecture

**Frontend (Gradio):** Provides a simple web-based chatbot UI for patients and doctors.

**Backend (Python & Colab):** Runs the core AI pipeline for chat, prediction, and report analysis.

**LLM Integration (IBM Granite via Hugging Face):** Powers natural language understanding & generation.

**Database/Storage:** Optionally integrates with secure cloud storage for patient records.

**Additional Modules:** Symptom checker, treatment recommender, and medical summarizer.

# 4. Setup Instructions

**Prerequisites:**

- Python 3.9 or later

- Gradio installed (`pip install gradio`)

- Hugging Face account and IBM Granite model access

- Google Colab environment with T4 GPU

- Git for version control

**Installation Process:**

1. Clone the repository from GitHub.

2. Install dependencies from requirements.txt.

3. Configure Hugging Face API key in Colab.

4. Run HealthAI.ipynb in Google Colab.

5. Launch Gradio UI and start interacting.

# 5. Folder Structure

app/ – Core backend logic and AI integration

app/api/ – API routes for chat, prediction, summarization

ui/ – Gradio UI components

models/ – Model loading and inference scripts

notebooks/ – Colab notebooks for prototyping

requirements.txt – Dependencies

README.md – Documentation

# 6. Running the Application

➢ Open Google Colab and run HealthAI.ipynb.

➢ Choose T4 GPU for faster inference.

➢ Load the IBM Granite model from Hugging Face.

➢ Start the Gradio interface for patient chat and predictions.

➢ Interact with the chatbot, upload reports, and receive guidance.

# 7. API Documentation

POST /chat – Accepts a user query (symptoms, health questions) and responds with AI guidance.

POST /upload-report – Uploads lab reports or PDFs for summarization.

GET /predict-disease – Returns likely conditions based on symptoms.

GET /treatment-plan – Suggests treatment options and next steps.

# 8. Authentication

• API key-based authentication for Hugging Face and IBM Granite.

• Planned enhancements: JWT tokens, OAuth2 integration, role-based access for patients and doctors.

# 9. User Interface

• Chat interface for patient-doctor-like interaction.

• Dashboard view for prediction results and summaries.

• File upload option for reports (PDFs, images).

• Downloadable treatment summary in PDF format.

# 10. Testing

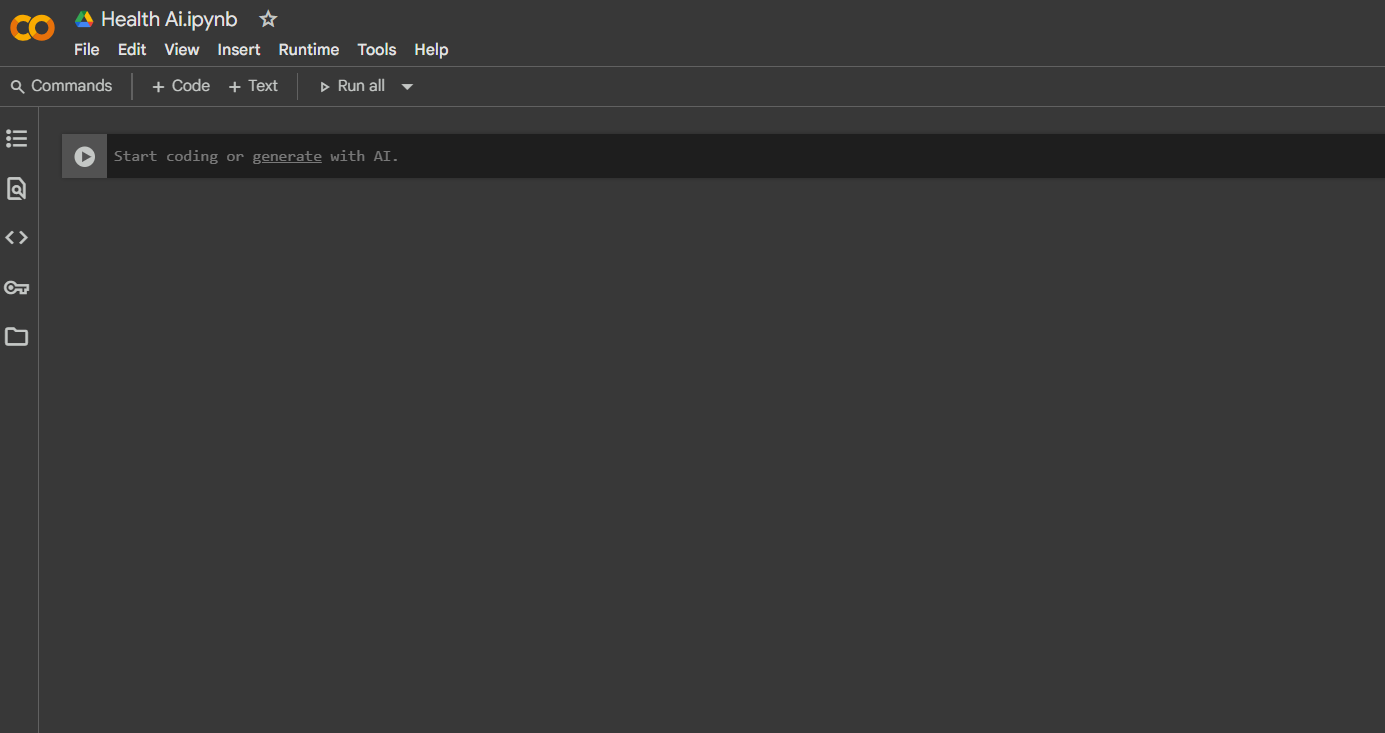
**• Unit Testing:** Core AI modules and symptom checker functions.

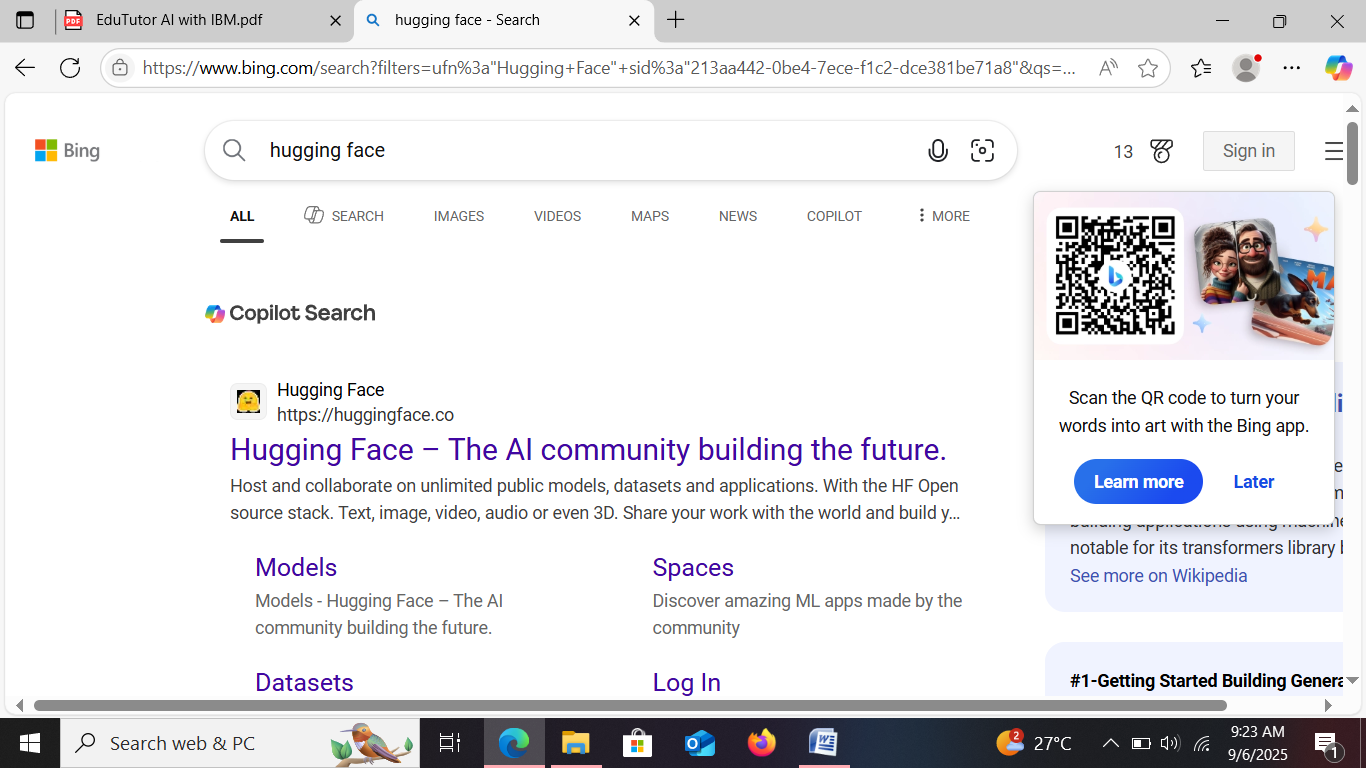
**• API Testing**: Via Postman and Swagger.

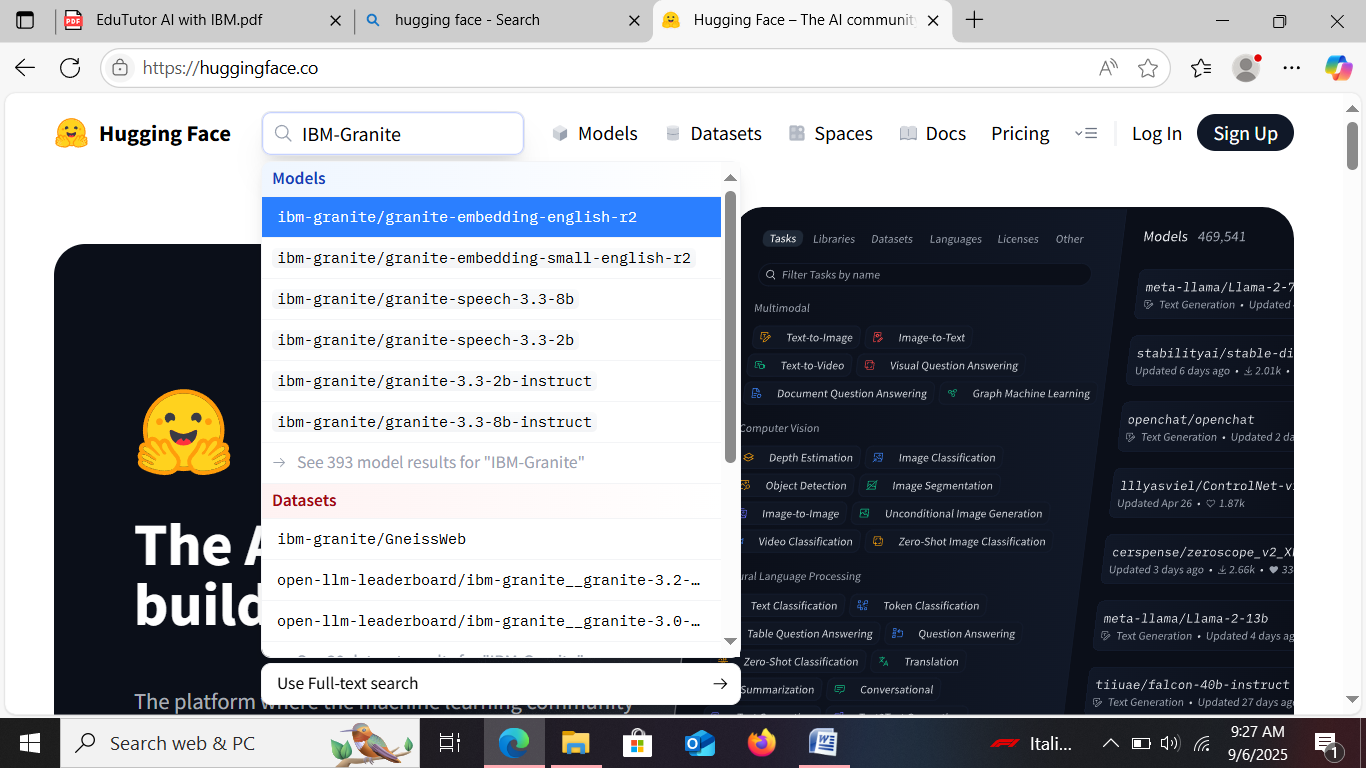
**• Manual Testing:** Chat interactions, prediction accuracy, report uploads.

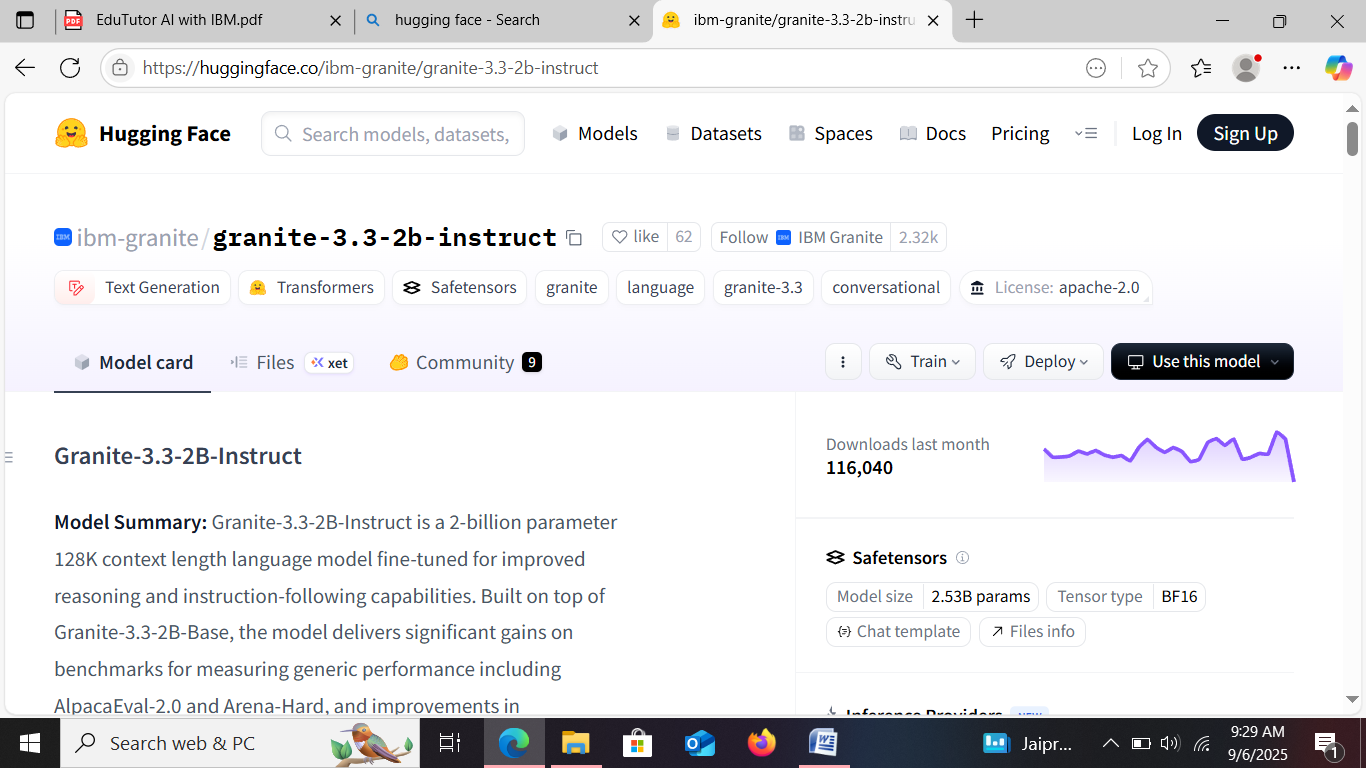
**• Edge Case Handling:** Incomplete symptoms, invalid inputs, empty files.

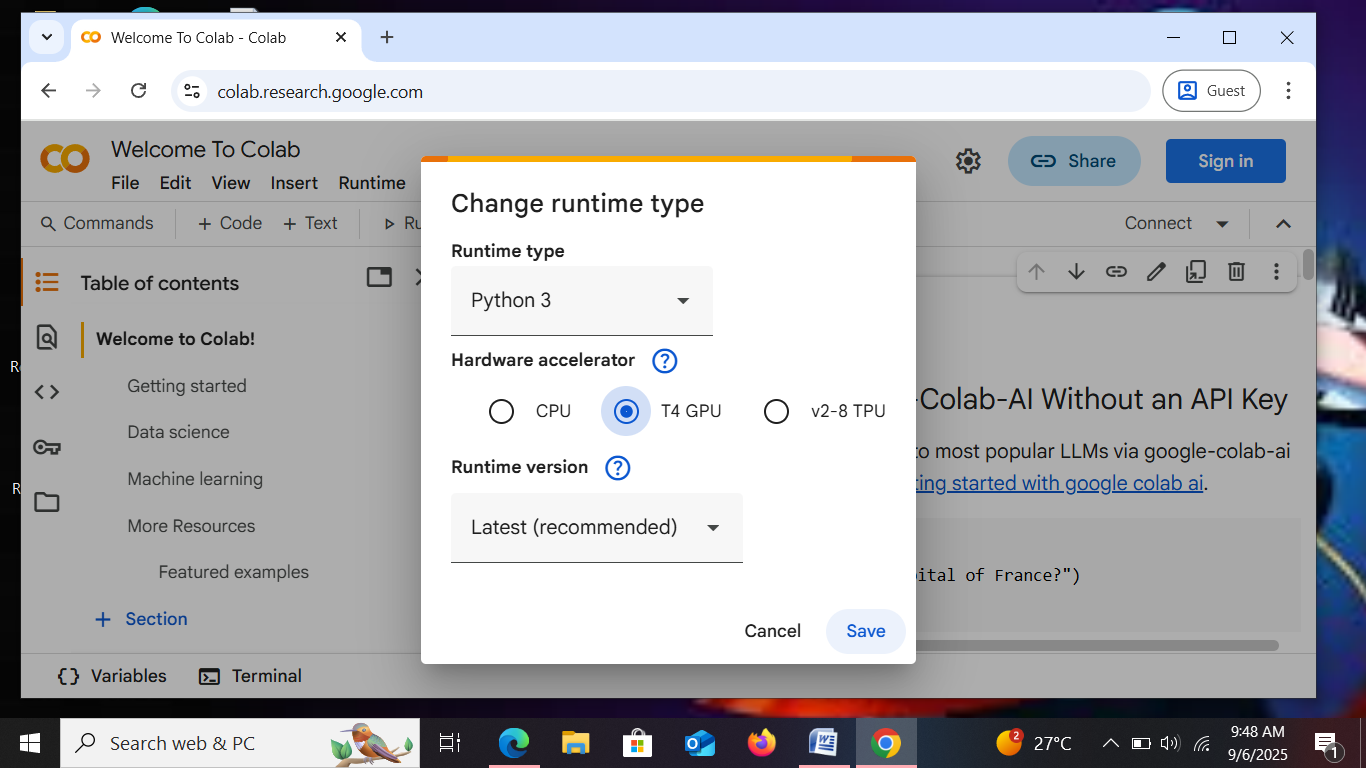
# 11. Screenshots

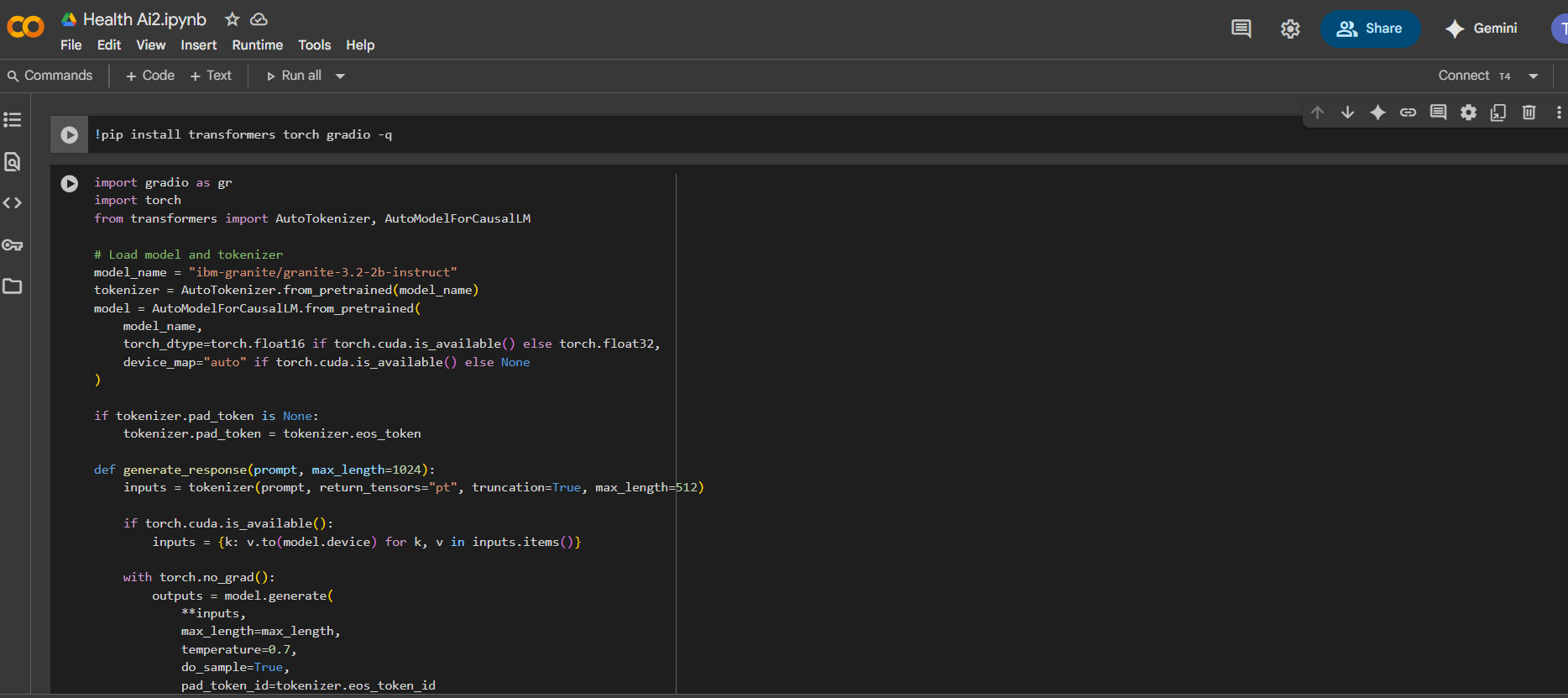


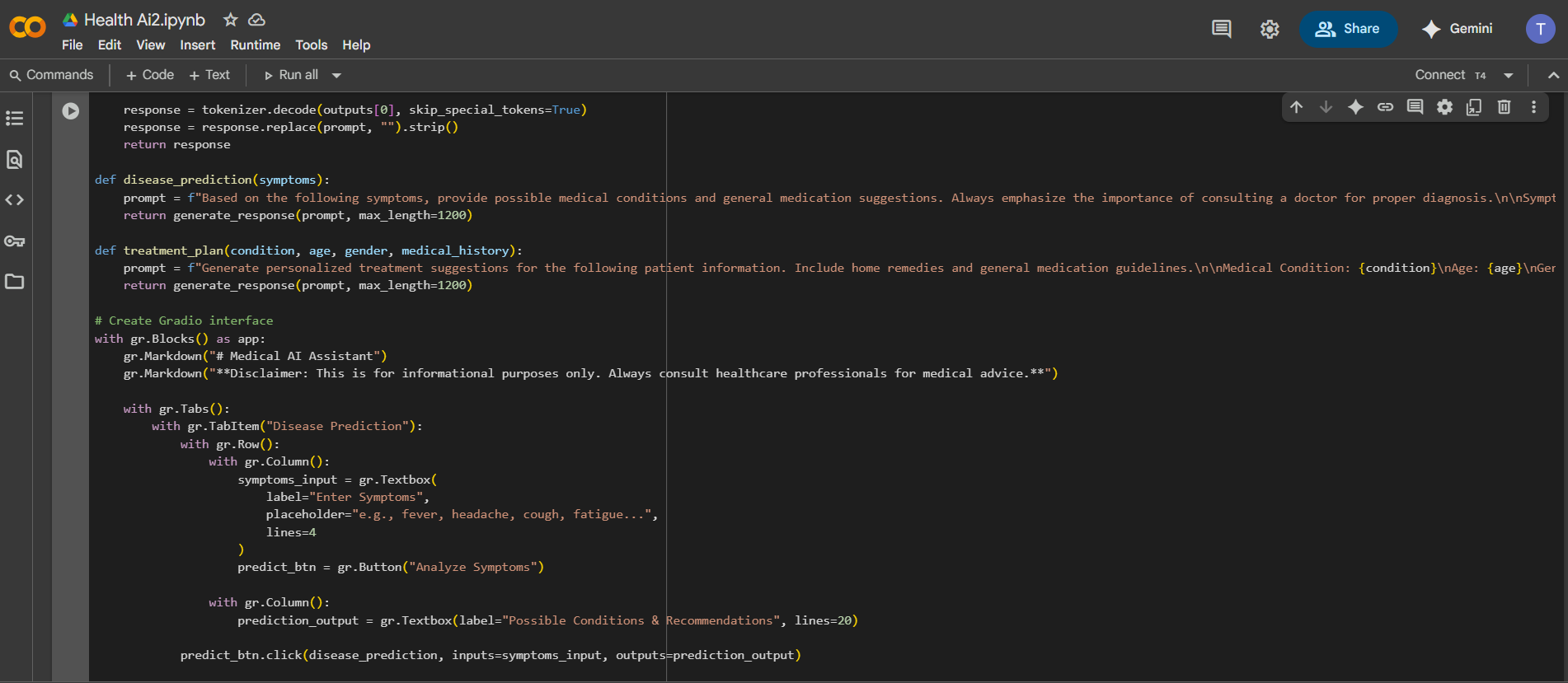


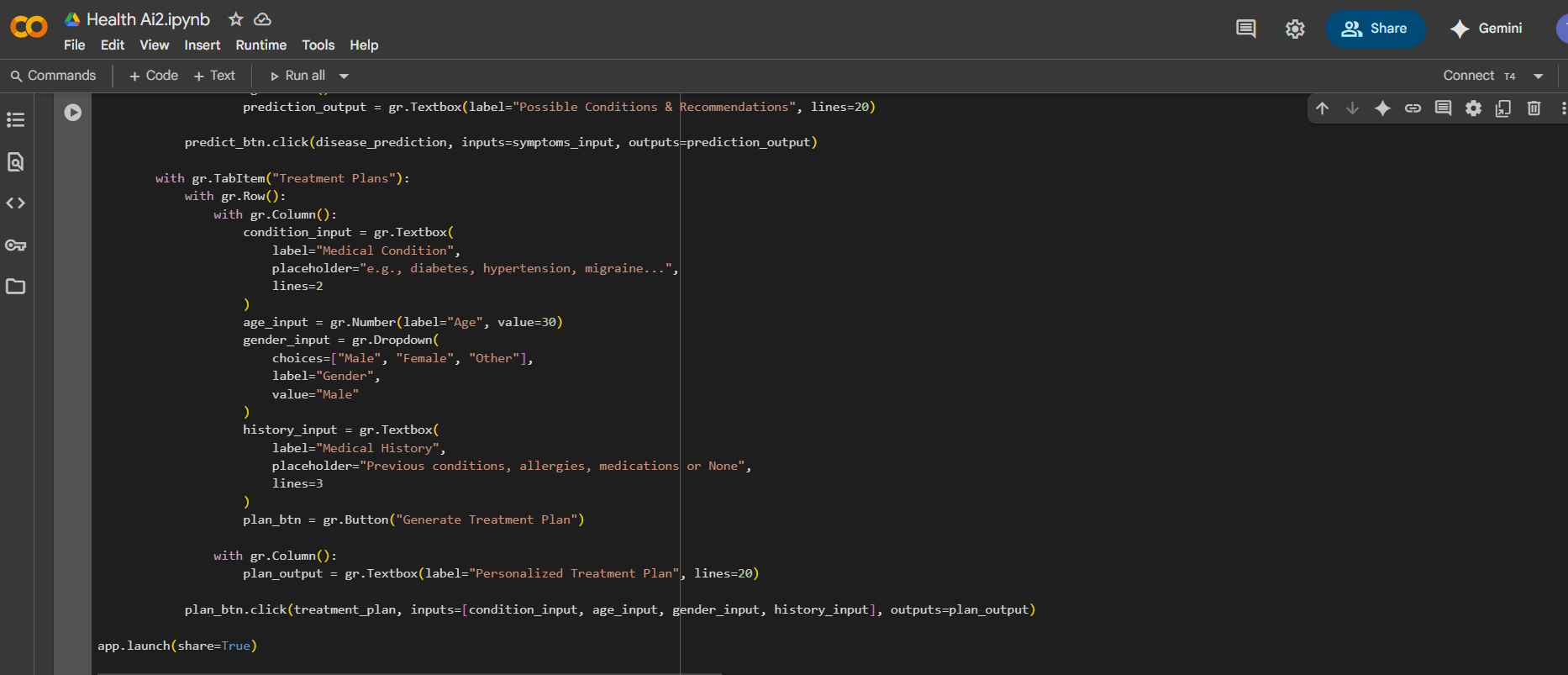




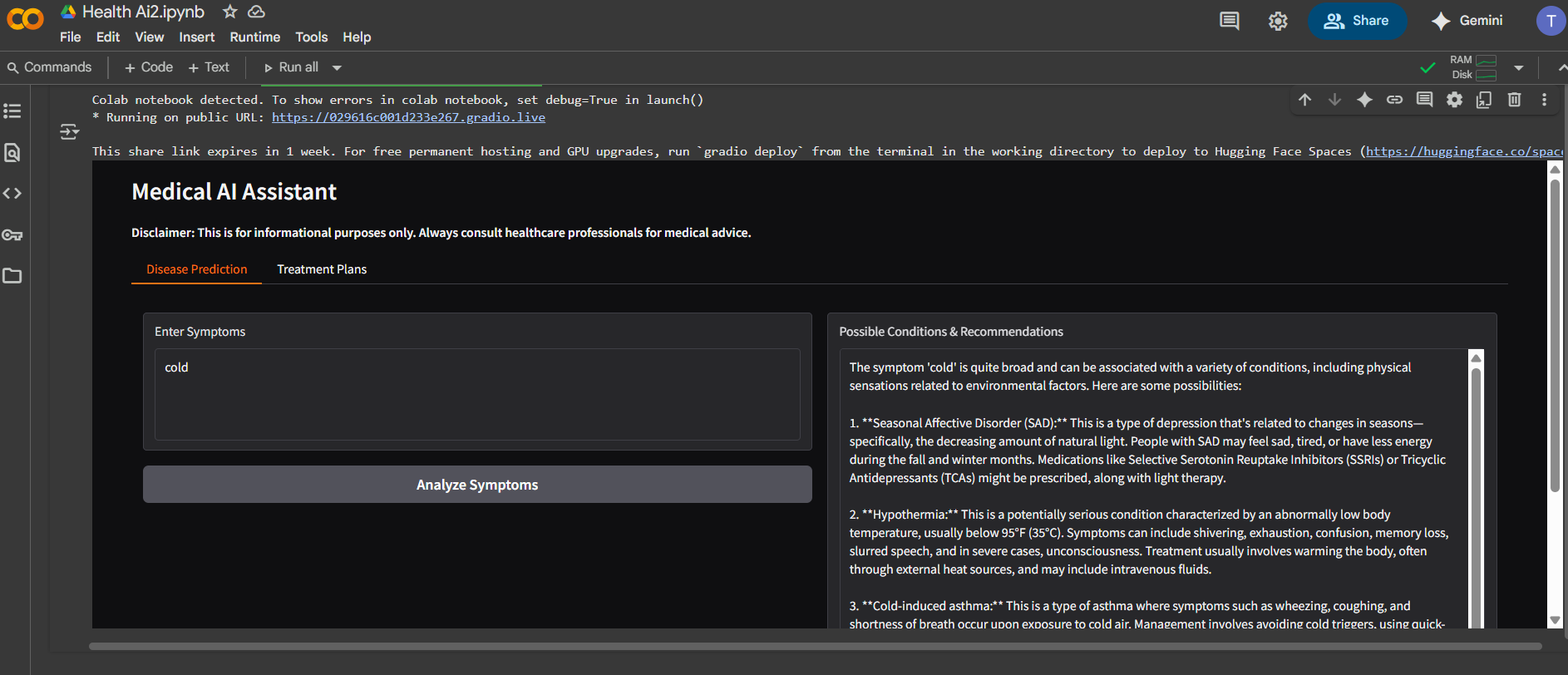


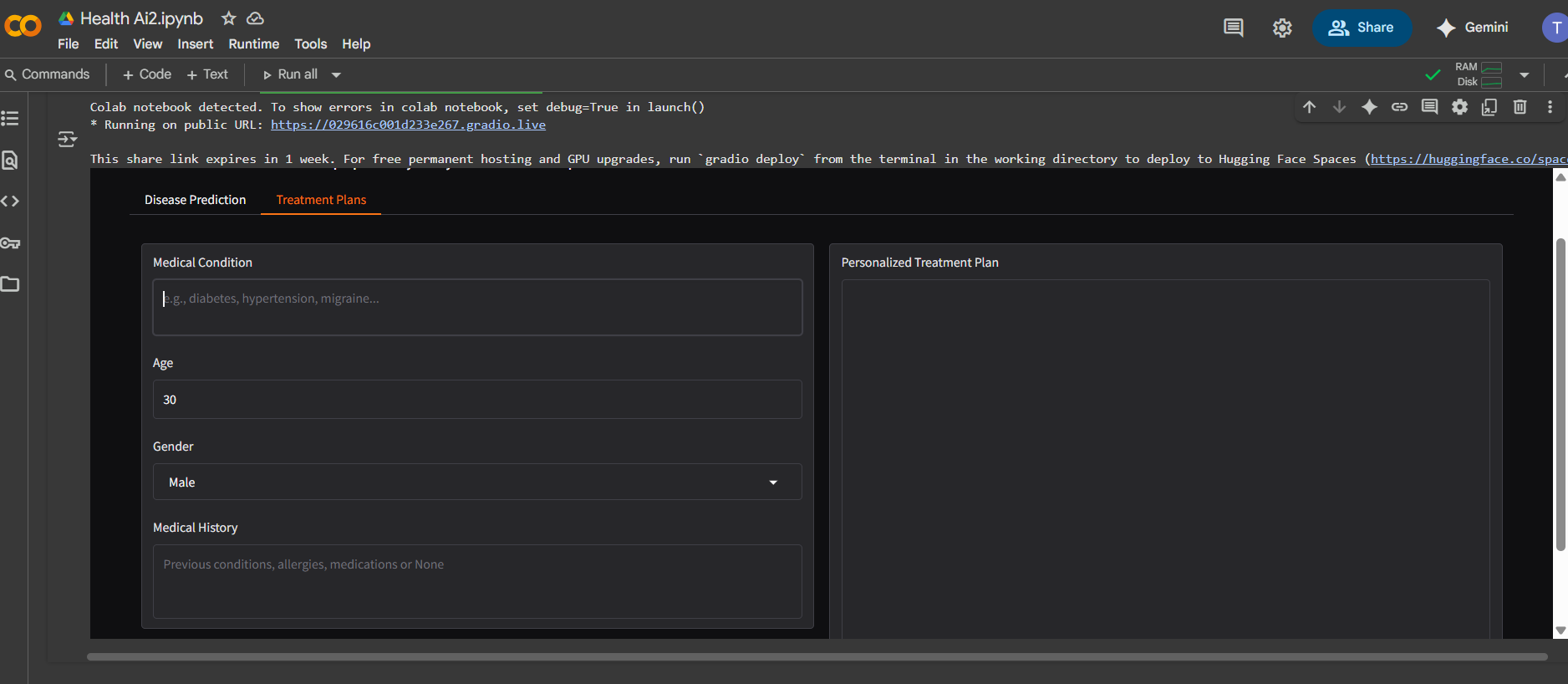






**Output**





# 12. Known Issues

• Limited accuracy in rare diseases due to dataset limitations.

• No direct EHR/EMR integration in current version.

• Dependent on stable internet connection for Hugging Face API.

# 13. Future Enhancements

• Add voice input and speech-to-text for accessibility.

• Expand disease coverage with specialized datasets.

• Add doctor-side dashboard for monitoring multiple patients.

• Enable integration with hospital systems and IoT devices.

• Implement advanced security with encrypted patient storage.