**Project Title**

**Project Documentation**

# 1.Introduction

* Project title : Health AI
* Team member : Oviya Priya. S
* Team member : Lavanya .P
* Team member : Jeeva.J
* Team member : kalai Selvi .R

# 2.project overview

* **Purpose** :

HealthAI is designed to assist patients, doctors, and healthcare providers by offering AI-powered support for diagnosis, medical record management, and personalized health recommendations. The system leverages AI and data analytics to improve treatment efficiency, provide preventive healthcare tips, and simplify medical decision-making. Ultimately, it bridges healthcare technology and patient engagement for better well-being.

* **Features** :
  + **Conversational Interface**
    - *Key Point*: Natural language interaction
    - *Functionality*: Patients can ask health-related questions and receive AI-generated responses in plain language
  + **Symptom Checker**
    - *Key Point*: Early diagnosis support
    - *Functionality*: Suggests possible conditions based on symptoms entered by the user
  + **Medical Record Management**
  + *Key Point*: Organized patient data
  + *Functionality*: Stores, retrieves, and summarizes patient health records
* **Prescription Reminder**
  + *Key Point*: Patient support
  + *Functionality*: Alerts patients to take medicines on time
* **Health Tips Generator**
  + *Key Point*: Preventive care
  + *Functionality*: Provides daily personalized tips to improve lifestyle and well-being
* **Doctor–Patient Interaction**
  + *Key Point*: Virtual consultation
  + *Functionality*: Enables patients to share symptoms, documents, and get feedback from doctors
* **Anomaly Detection**
  + *Key Point*: Alert system
  + *Functionality*: Identifies abnormal health patterns like unusual heart rate or blood sugar levels
* **Multimodal Input Support**
  + *Key Point*: Flexible data handling
  + *Functionality*: Accepts text, PDFs, or images (like prescriptions, lab reports) for analysis
* **User-Friendly Dashboard**
  + *Key Point*: Accessible interface
  + *Functionality*: Provides an intuitive UI for patients, doctors, and healthcare staff

# 3. Architecture

* **Frontend (Streamlit)** :Web-based interface for patients and doctors, including dashboards, chat, health tips, and medical records viewer
* **Backend (FastAPI)** :REST framework powering APIs for chat, health record management, anomaly detection, and reminders
* **LLM Integration (IBM Watsonx Granite)** : Provides health-related summaries, recommendations, and patient-doctor communication support
* **Database (MongoDB / SQL)** : Stores patient details, health records, prescriptions, and reminders
* **ML Modules (Scikit-learn / TensorFlow)** : o Symptom checker using classification models o Anomaly detection in patient data

o Health forecasting (e.g., sugar levels, BMI trends)

## 4. Setup Instructions

* **Prerequisites** :
  + Python 3.9 or later o pip and virtual environment tools o API keys for Watsonx / cloud integrations o Database setup (MongoDB or SQL)
* **Installation Process** :
  + Clone the repository
  + Install dependencies from requirements.txt o Configure .env file with credentials o Run backend server using FastAPI o Launch frontend via Streamlit
  + Upload patient data and interact with modules

## 5. Folder Structure

 **app/** – Backend logic with routers and services

 **app/api/** – APIs (chat, health records, reminders, anomaly detection)

 **ui/** – Streamlit pages and dashboards

 **health\_dashboard.py** – Launches main Streamlit dashboard

 **symptom\_checker.py** – Suggests possible conditions

 **record\_manager.py** – Stores and summarizes patient data

 **anomaly\_detector.py** – Detects unusual health values

 **reminder\_service.py** – Manages prescription reminders

 **report\_generator.py** – Generates health summary reports

## 6. Running the Application

* Start FastAPI backend server
* Run Streamlit dashboard
* Navigate through sidebar to access modules
* Upload health data or reports
* Chat with AI assistant or view forecasts and reminders
* Download reports or share with doctors

## 7. API Documentation

**POST /chat/ask** – AI health query responses

**POST /upload-record** – Upload patient health record **GET /search-records** – Retrieve patient history

**GET /get-health-tips** – Provides daily health tips

**POST /set-reminder** – Create prescription or appointment reminders

Documented with Swagger UI for testing

## . 8. Authentication

* Current version: Open environment (demo)
* Secure deployment options:

o Token-based authentication (JWT, API keys) o Role-based access (doctor, patient, admin) o Encrypted storage for sensitive data o Planned session & history tracking

## 9. User Interface

The interface is a **clean, user-friendly health platform** designed for accessibility and ease of use across all age groups. Key features include:

* **Sidebar Navigation**

A left-hand sidebar with simple input fields (name, age, gender, medical history, symptoms) for quick patient data entry.

* **Patient Dashboard with Health KPIs**

Displays a card-style summary of key health indicators (conditions, recommendations, therapy plans) with a clear typography hierarchy.

* **Tabbed Layout**

Tabs at the top or side for switching between:

o *Chat / AI Consultation* (for interactive health advice) o *Health Tips* (personalized lifestyle recommendations) o *Records* (medical history, prescriptions, reports)

* **Smart Notifications**

Reminder alerts for upcoming checkups, medications, or exercise sessions.

* **Downloadable PDF Summaries**

One-click option to export patient assessments or therapy plans into neatly formatted PDF files.

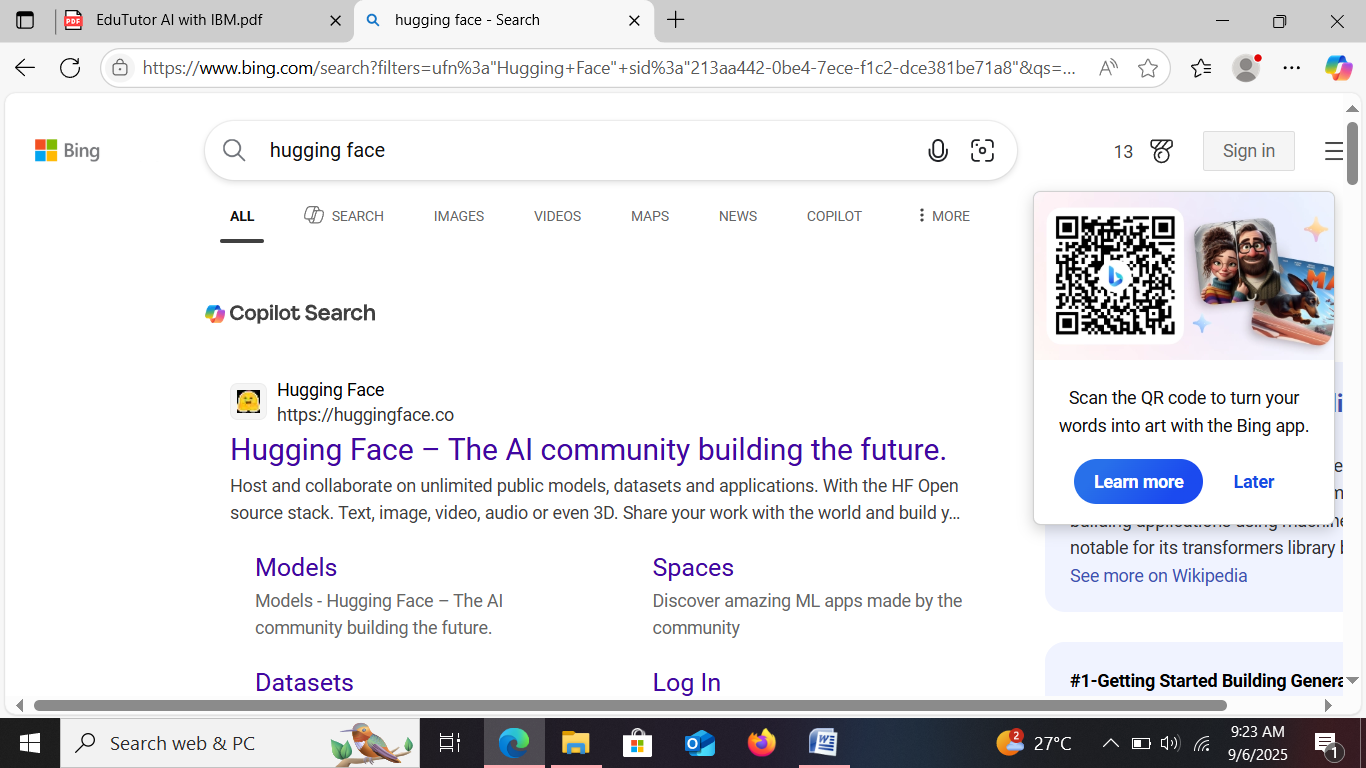
* **Minimalist, Accessible Design** o Large, high-contrast text for readability o Clear icons and spacing for easy navigation o Works seamlessly on desktop and mobile o Inclusive for both tech-savvy and non-technical users

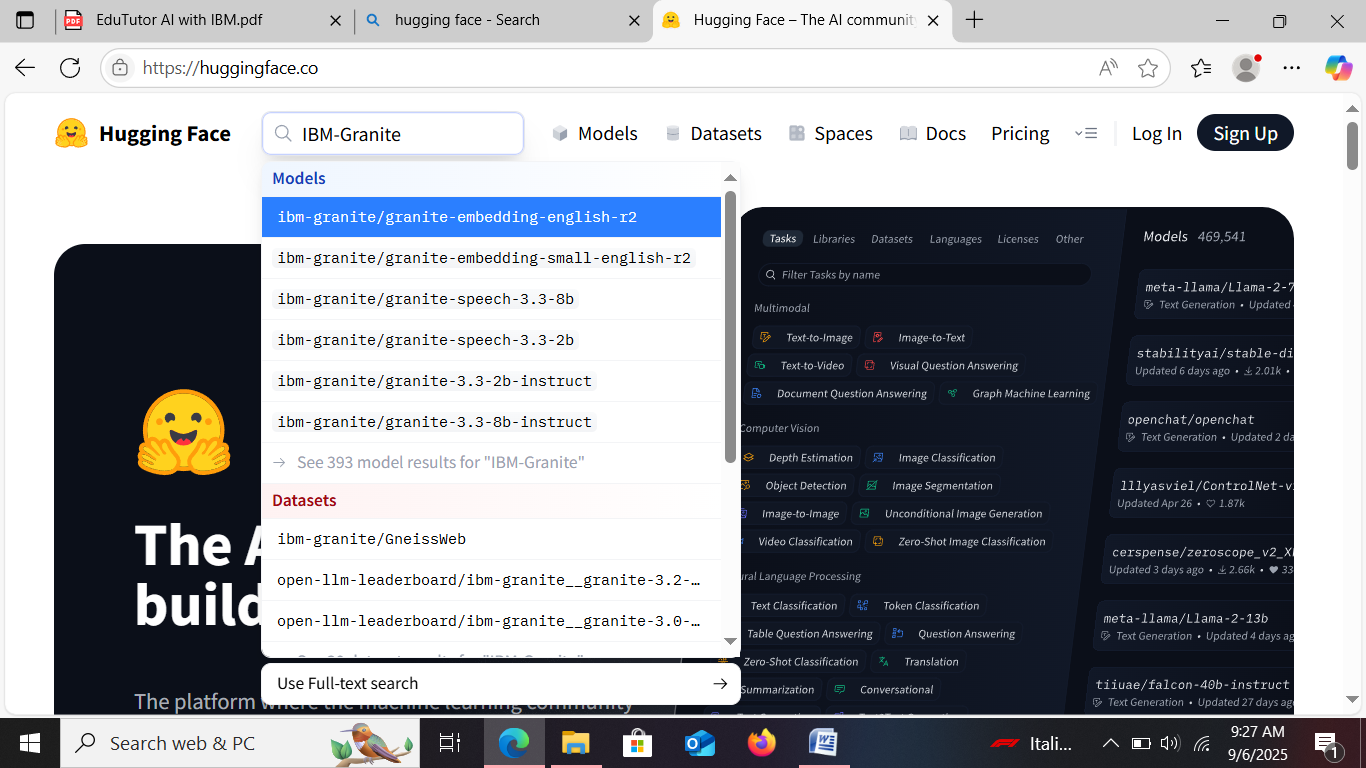
## 10. Testing

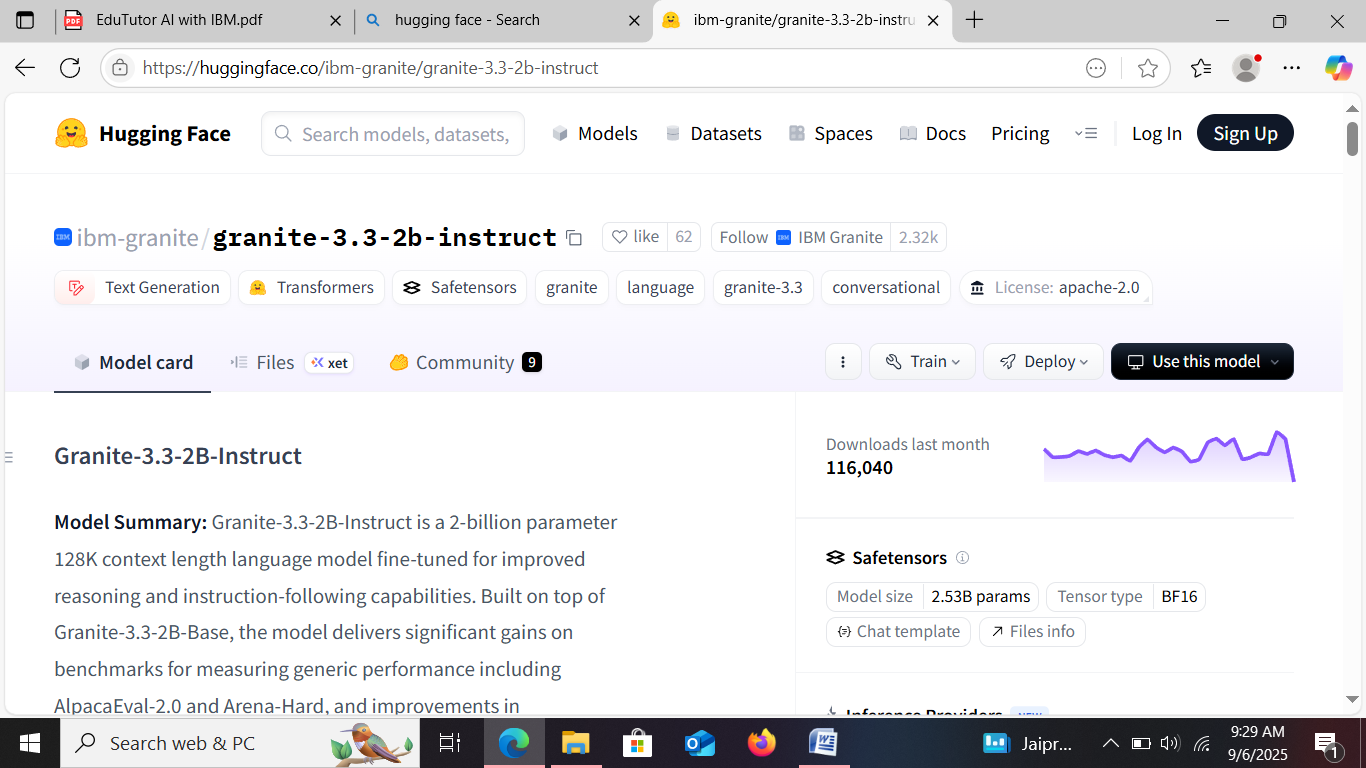
Testing was done in multiple phases:

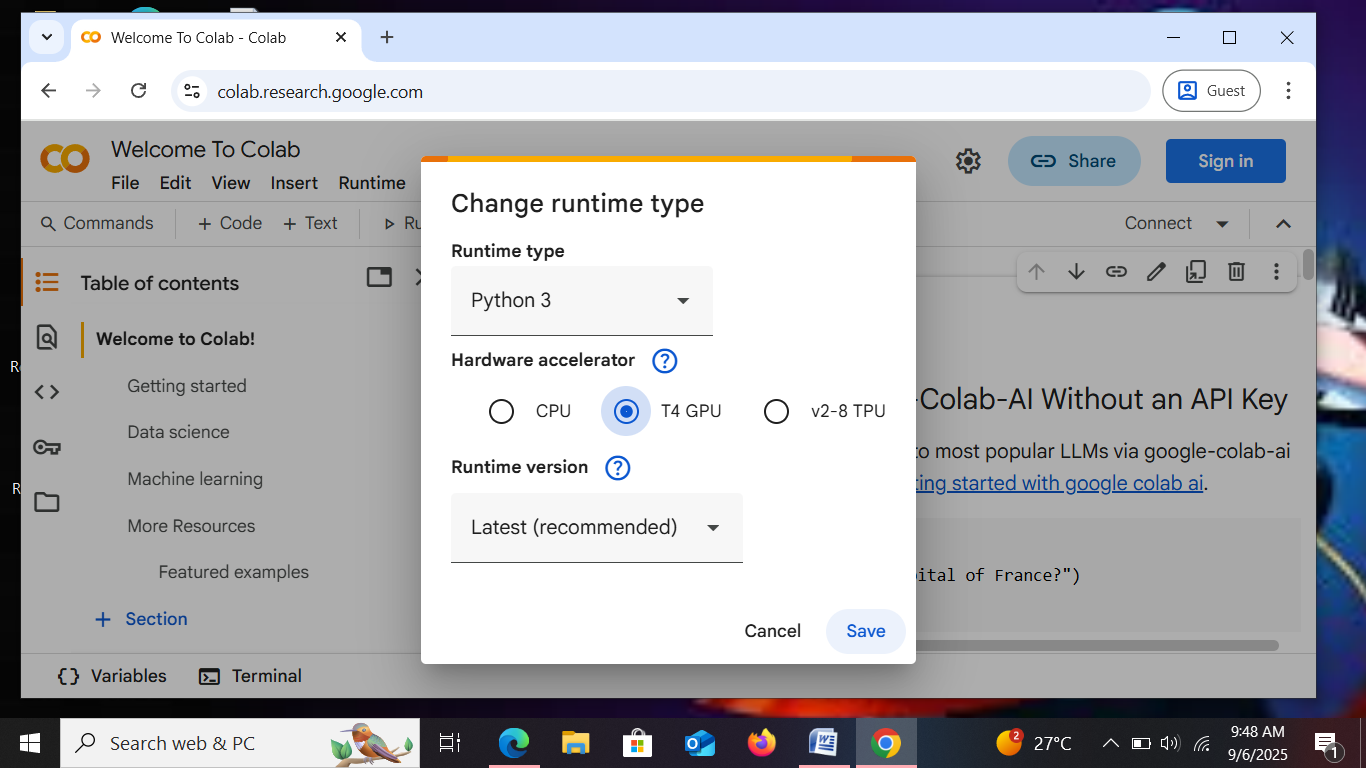
* **Unit Testing**: Symptom checker, anomaly detection, reminders
* **API Testing**: Swagger UI and Postman
* **Manual Testing**: Chat assistant, file uploads, record summaries
* **Edge Cases**: Malformed data, missing health records, invalid inputs

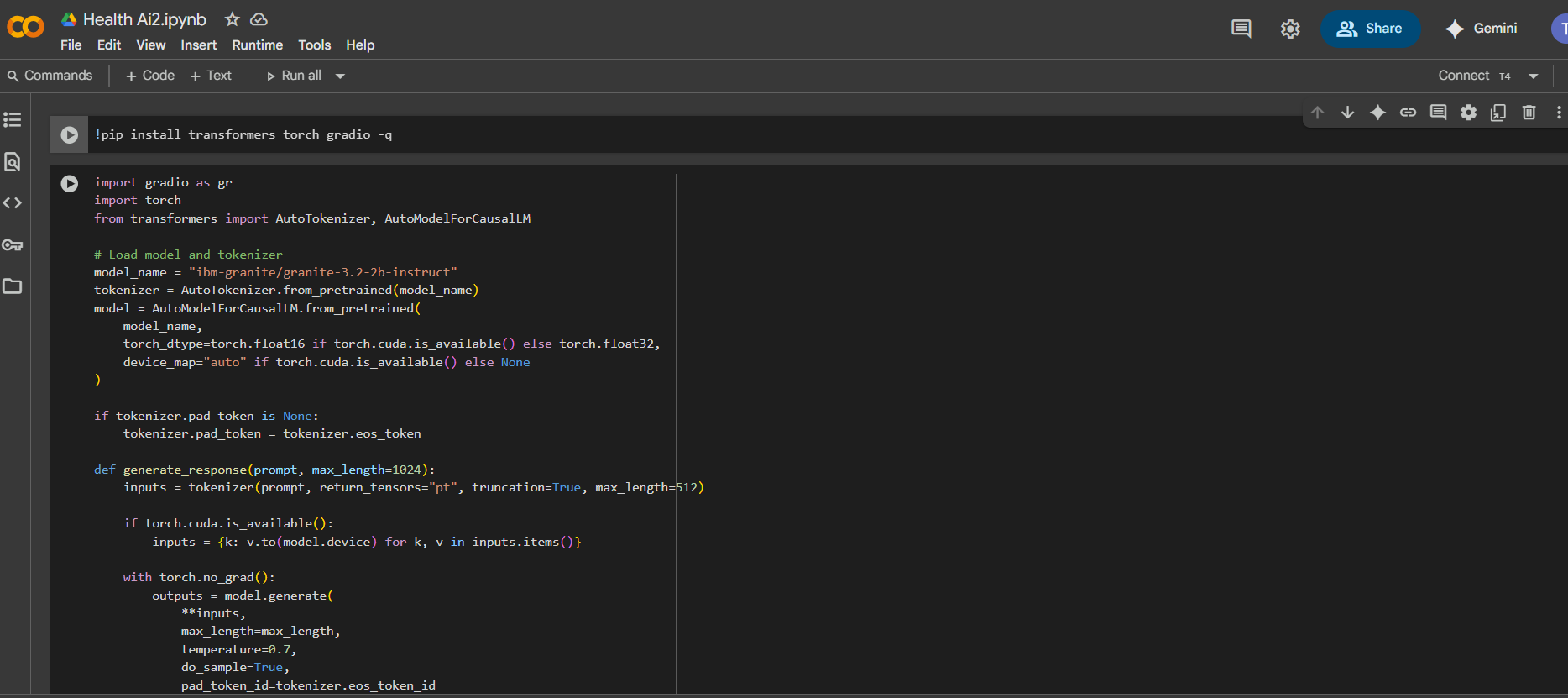
11.screen shots

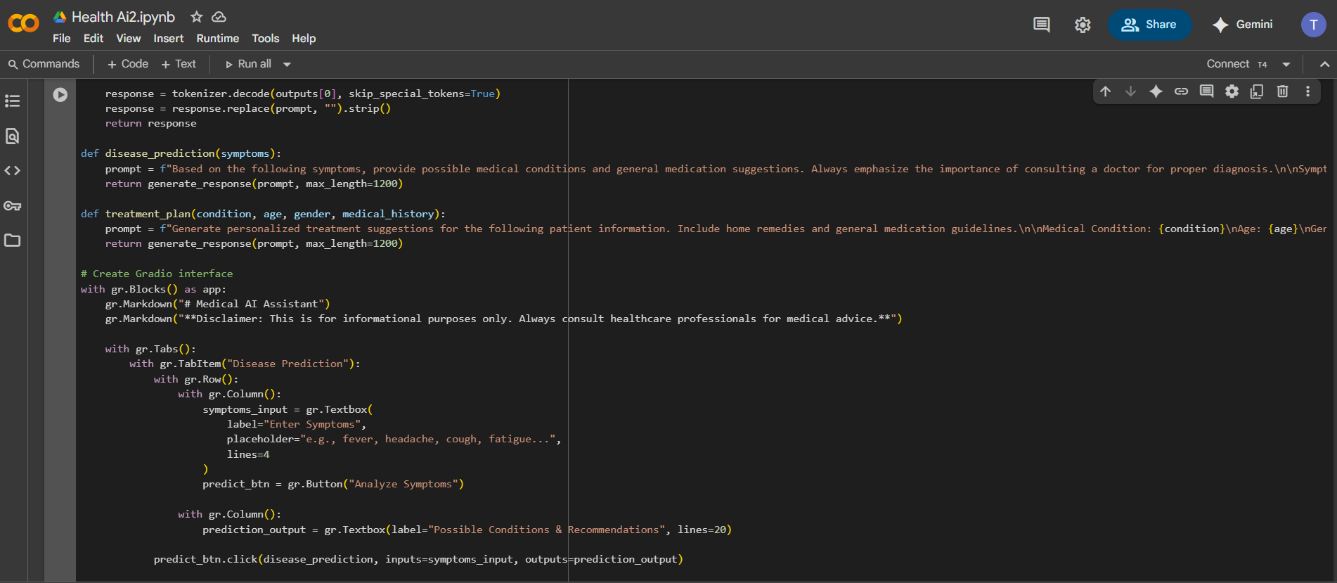


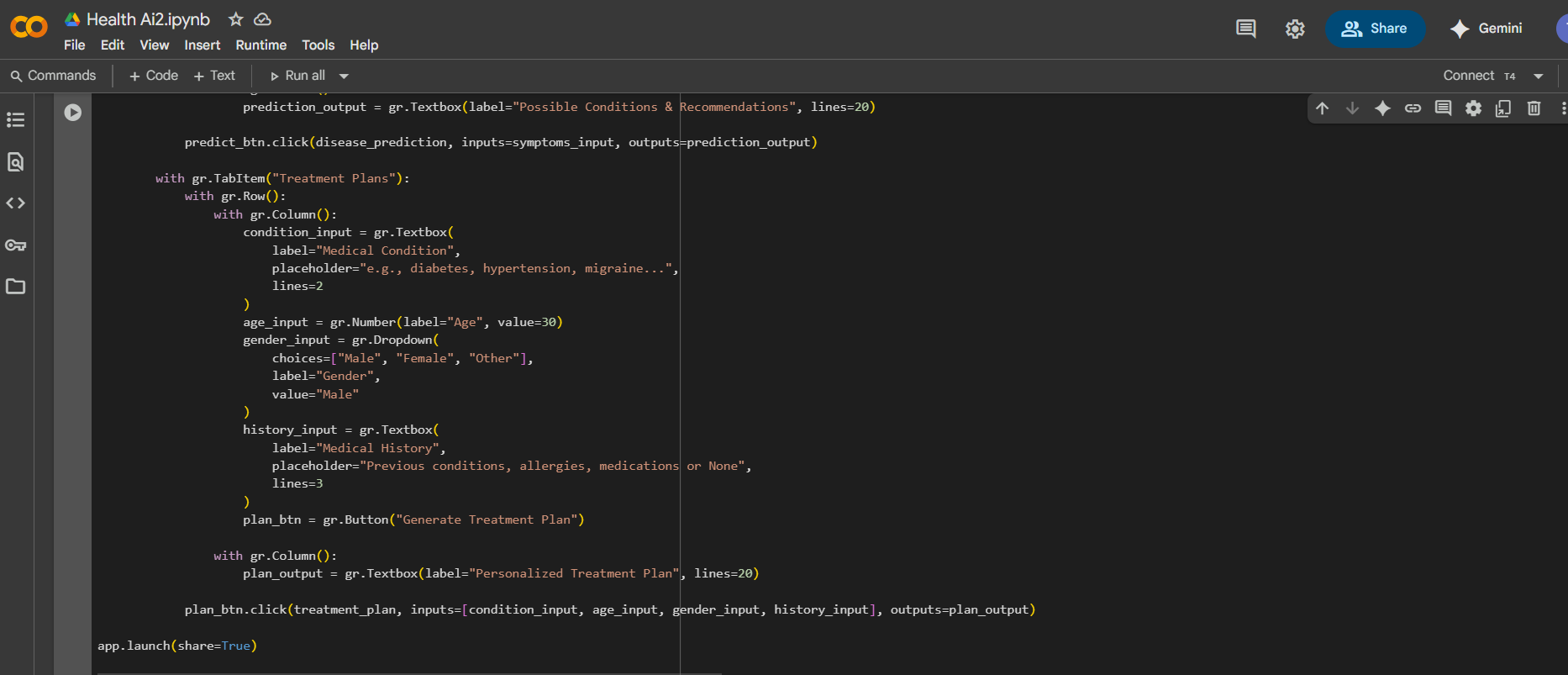


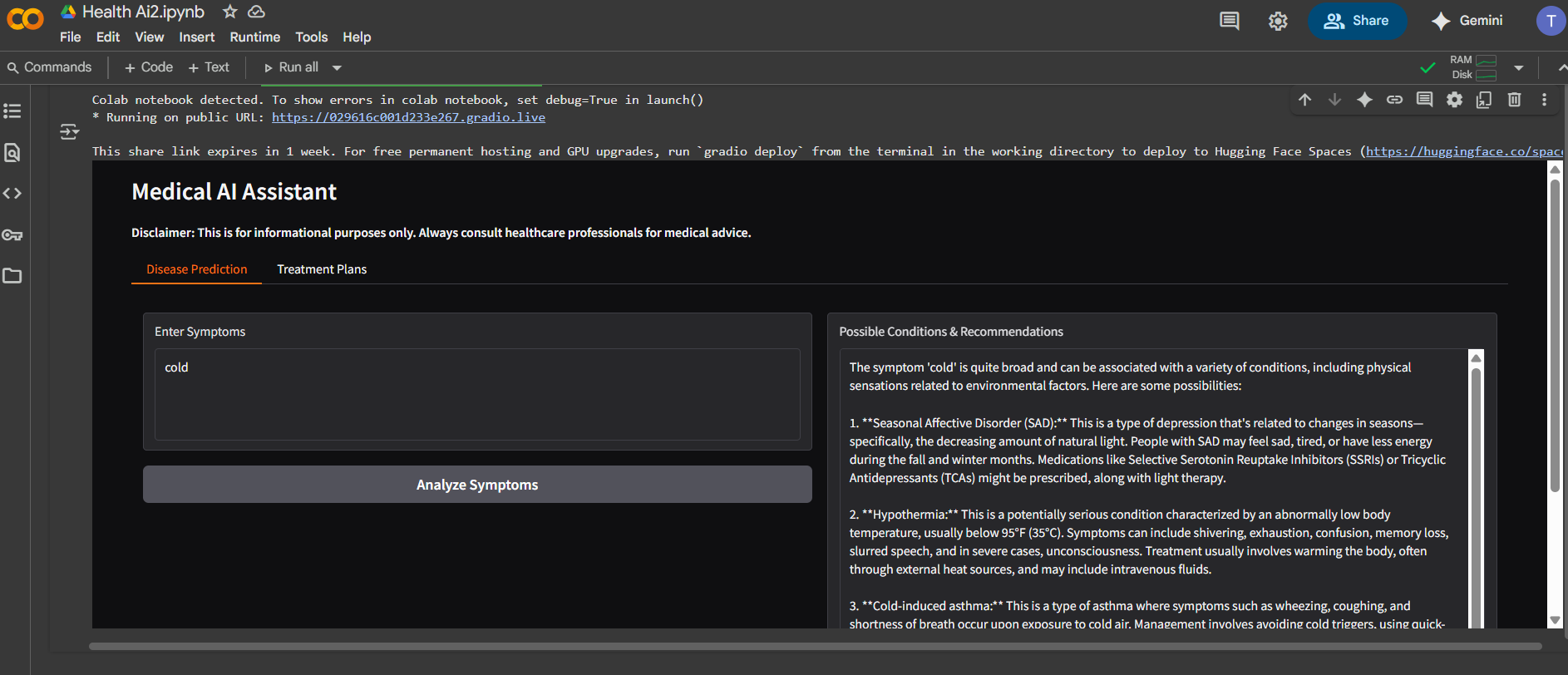


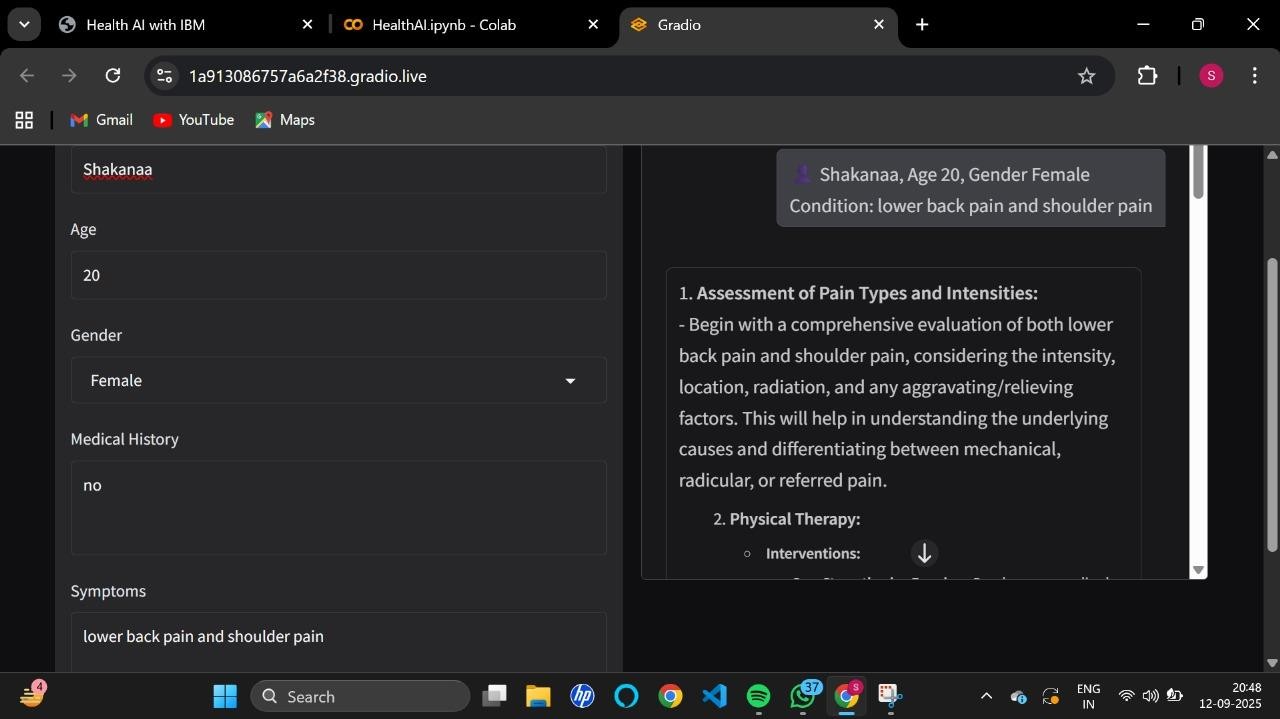












## 12. Known Issues

* Requires reliable internet for AI queries
* Limited offline functionality
* High dependency on cloud APIs

## 13. Future enhancement

* Mobile app version for patients
* Integration with IoT devices (smartwatches, glucose monitors)
* AI-powered disease prediction models
* Multi-language support for regional users
* Blockchain for secure medical record storage