**Medical AI Assistant – Project Documentation**

# Introduction

**Project title** **:** HealthAI-Intelligent Healthcare Assistant Using IBM Granite

* + Team member :D.KIRUTHIKA
  + Team member :S.PRIYANKA
  + Team member :R.MOWLICA
  + Team member :B.CHARMI

**1. Project Overview**

The **Medical AI Assistant** is an AI-powered healthcare support tool built using **Gradio**, **Hugging Face Transformers**, and **IBM Granite LLM**. It is designed to provide **informational assistance** to patients by analyzing symptoms, generating possible treatment suggestions, and offering interactive chat support. Additionally, the system provides a **health analytics dashboard** to visualize key health metrics. HealthAI is an AI-powered healthcare assistant designed to provide **informational medical support** through a user-friendly **Gradio interface**. Leveraging IBM’s Granite large language model (LLM), the system enables users to interact with healthcare features such as **Disease Prediction, Treatment Plan Generation, Patient Chat, and Health Analytics Visualization**.

This project demonstrates how AI and natural language processing (NLP) can improve healthcare accessibility by offering patients **personalized insights, condition assessments, and treatment suggestions** while maintaining the disclaimer that professional medical consultation is essential.

⚠️ **Disclaimer**: This system is intended for **educational and informational purposes only**. It does not replace professional medical advice, diagnosis, or treatment.

**2. Purpose**

The purpose of the project is to:

* Help patients **understand symptoms** by providing AI-generated medical insights.
* Suggest **general treatment plans** based on user inputs (age, gender, history).
* Provide an **interactive chatbot** to answer medical-related queries.
* Display **health metrics analytics** in an easy-to-understand dashboard.

**Objectives**

* To build an **AI-powered healthcare support system** using IBM Granite models.
* To provide a **symptom-based disease prediction tool** with likelihood and recommendations.
* To generate **personalized treatment plans** considering patient details.
* To enable **interactive patient chat** for general health questions.
* To offer **visual health analytics dashboards** for monitoring health metrics.
* To design a **scalable, modular, and user-friendly Gradio interface**.
* To ensure **responsible AI usage** with clear disclaimers and ethical handling of medical information.

**Scope**

* **User Accessibility**: Patients, students, and healthcare learners can use the system.
* **Disease Prediction**: Supports **multiple common health conditions** based on user-reported symptoms.
* **Treatment Recommendations**: Provides **generalized treatment guidelines** (lifestyle, medication categories, follow-up steps).
* **Data Visualization**: Allows **visual tracking of health metrics** such as heart rate, blood pressure, glucose, and cholesterol.
* **Conversational AI**: Provides **interactive health consultations** through the chatbot.
* **Extensible Design**: Can be expanded to include **more advanced AI models** and **integration with wearable devices (IoT)**.

**Key Features**

* **Disease Prediction**
  + Input symptoms and get possible medical conditions.
  + Provides general recommendations and emphasizes consulting doctors.
* **Treatment Plan Generation**
  + Creates personalized treatment suggestions based on age, gender, and medical history.
* **Patient Chat**
  + Conversational chatbot for healthcare-related queries.
  + Maintains chat history for context.
* **Health Analytics Dashboard**
  + Displays mock data for blood pressure, heart rate, blood sugar, and cholesterol.
  + Provides bar chart visualization for quick insights.

**Benefits**

* Improves **accessibility to healthcare information** for all users.
* Offers **fast and interactive health assistance** without waiting times.
* Reduces dependency on static online searches by providing **contextual AI answers**.
* Assists **patients, students, and medical learners** with simplified insights.
* Enables **data-driven healthcare awareness** through analytics dashboards.
* Provides **cost-effective and scalable solution** that can be deployed on cloud platforms.

**Technology Stack**

* **Programming Language**: Python
* **Frameworks/Libraries**: Gradio, Transformers (Hugging Face), Torch, Pandas, Matplotlib
* **AI Model**: IBM Granite instruct model (health-focused prompting)
* **Deployment**: Gradio interface with local/streamlit cloud deployment
* **Data Handling**: Pandas DataFrame for structured patient health metrics

**4. Resource Forecasting (Functionality)**

Instead of city resources, here it **forecasts patient health patterns**:

* Future enhancement can include **time-series forecasting** of health data (e.g., sugar levels).
* Detecting anomalies in metrics for **early alerts** (e.g., unusually high BP).

**5. Architecture**

**Frontend (Gradio):**

* Provides a **user-friendly, tabbed interface**.
* Four key modules: **Disease Prediction, Treatment Plan, Patient Chat, Health Dashboard**.
* Real-time interaction through text inputs, buttons, and chatbot.
* Ensures accessibility for non-technical users.

**Backend (Transformers + PyTorch):**

* Built on **Hugging Face Transformers** framework.
* Utilizes **IBM Granite LLM** for natural language understanding and medical text generation.
* **PyTorch** handles model execution with GPU/CPU support for performance.
* Implements **prompt-based generation** for consistent and context-aware outputs.

**Visualization (Matplotlib + Pandas):**

* Handles **mock patient health data** (BP, sugar, cholesterol, heart rate).
* **Pandas** for structured data management.
* **Matplotlib** for visualizing health metrics as charts and summaries.
* Supports future integration of **real patient data** and anomaly detection.

**6. Setup Instructions**

**Prerequisites**

* Python 3.9+
* pip package manager
* Torch and Transformers libraries
* Gradio for UI
* Matplotlib and Pandas for analytics

**Installation Process**

1. Clone the repository.
2. Install dependencies:
3. pip install torch transformers gradio pandas matplotlib
4. Run the application:
5. python app.py
6. Open the Gradio link in the browser to access the assistant.

**7. Folder Structure**

medical\_ai\_assistant/

│

├── app.py # Main script with Gradio UI

├── model\_setup/ # Model loading and tokenization

├── utils/ # Helper functions (response generation, analytics)

└── assets/ # Charts, reports, or stored chat logs

**8. Running the Application**

* Launch the script with python app.py.
* Access the Gradio UI in your browser.
* Use the tabs for disease prediction, treatment plan, chat, and dashboard.

**9. API Documentation**

(Current version runs as UI, but API can be added in the future.)

Possible API endpoints:

* POST /predict-disease – Accepts symptoms, returns conditions.
* POST /treatment-plan – Returns AI-generated plan.
* POST /chat – Conversational query handling.
* GET /health-dashboard – Provides patient health metrics.

**10. Authentication**

* Present version: Open demo (no authentication).
* Future: Token-based authentication or role-based access for doctors/patients.

**11. User Interface**

* Built with **Gradio Blocks**.
* **Tabbed navigation** for features.
* Chatbot for real-time interaction.
* Data visualization with bar charts.

**12. Testing**

* **Unit Testing**: Prompt generation and chatbot logic.
* **Manual Testing**: Inputting symptoms and validating AI output.
* **Edge Cases**: Empty inputs, irrelevant medical queries, invalid numeric values.

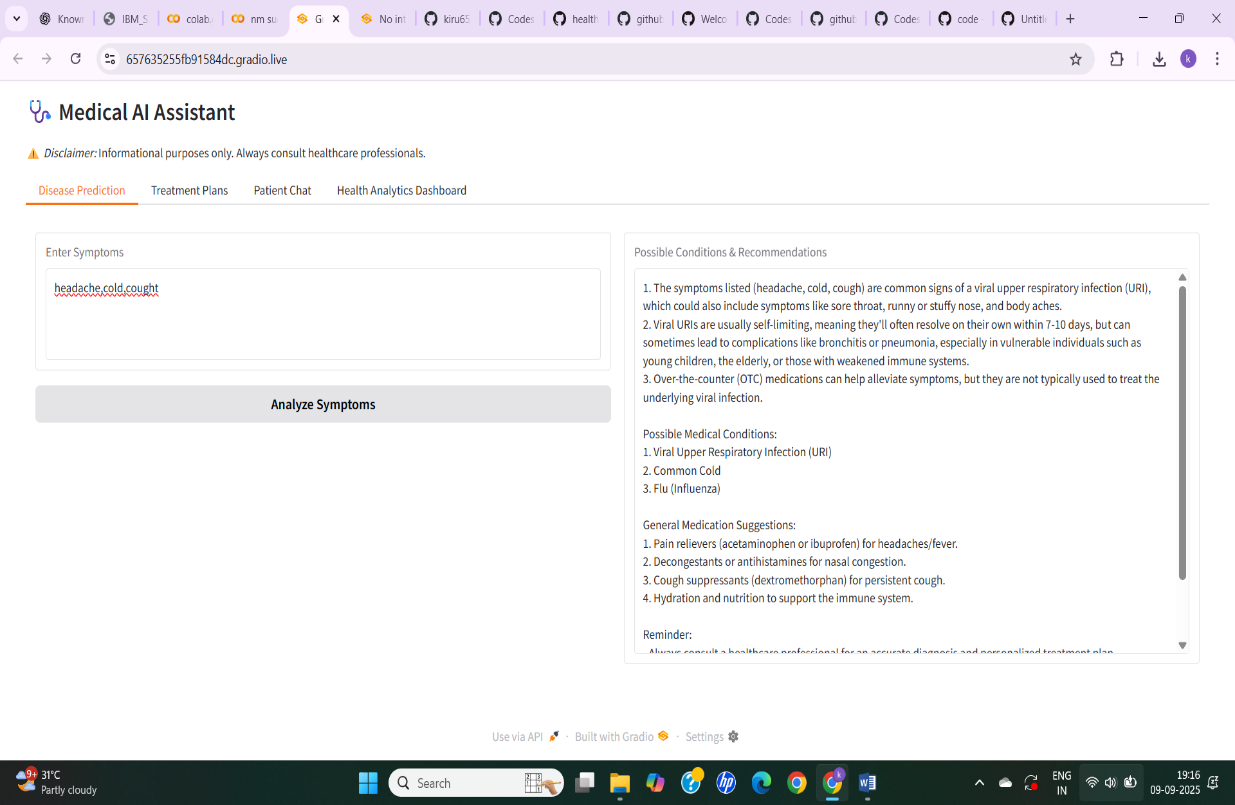
**13. Known Issues**

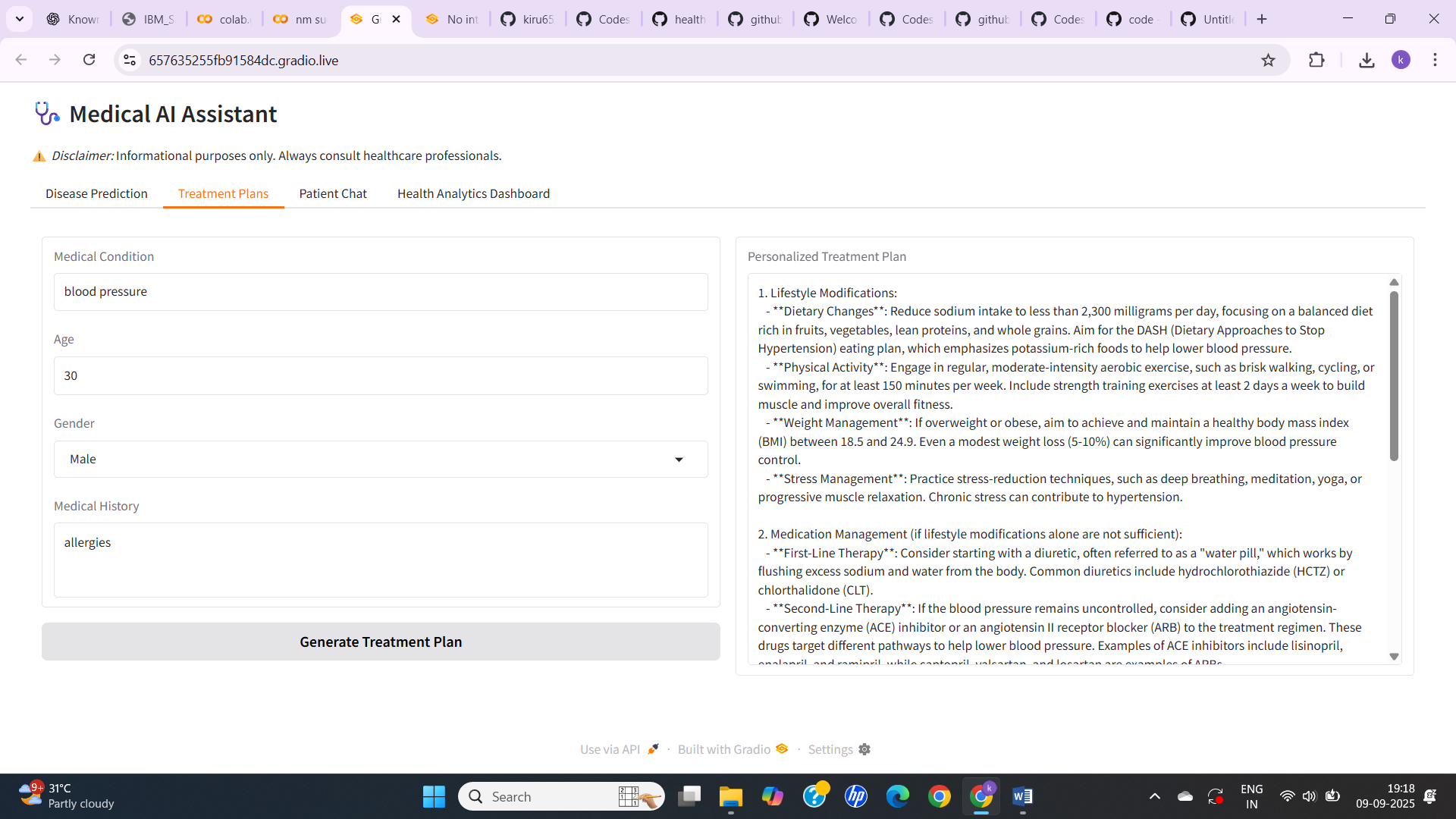
* AI outputs may sometimes be **too generic** or not medically accurate.
* No real patient dataset (currently using mock values).
* Model responses depend heavily on prompt engineering.

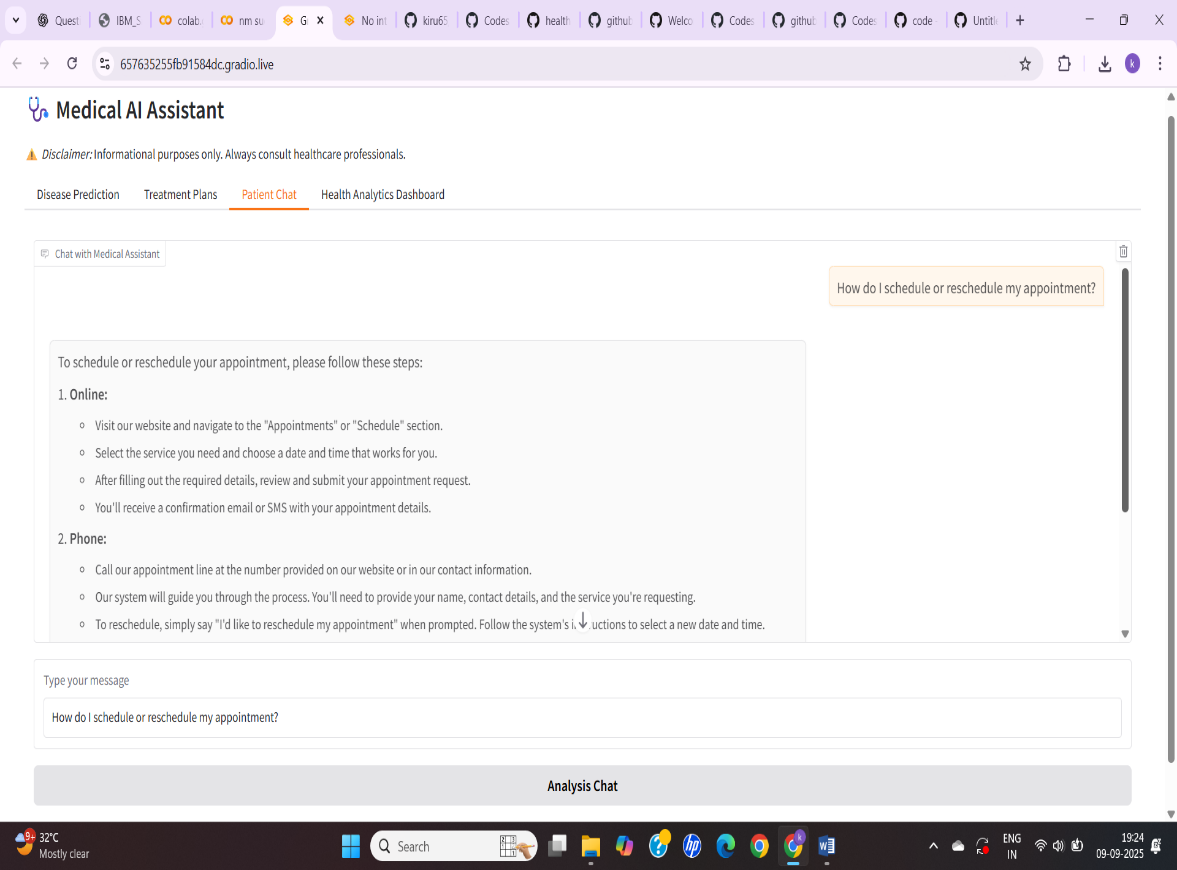
**14. Future Enhancements**

* Integration with **real patient data** (via secure APIs & IoT devices like wearables).
* Add **forecasting and anomaly detection** for health metrics.
* Secure login with **doctor-patient roles**.
* Option to generate **downloadable medical reports (PDF/CSV)**.
* Expand to include **multi-language support**.
* Advanced **multi-language support** for global accessibility.
* Implementation of **voice-based chat assistant**.
* Integration with **Electronic Health Records (EHR)** systems.
* **Automated emergency alerts** for abnormal health readings.
* Expansion of disease prediction with **deep learning medical datasets**.
* Deployment on **mobile platforms (Android/iOS)** for accessibility.

**15.Screenshots**







Patient Health Data

| Patient Health Data | |
| --- | --- |
| **Metric** | **Value** |
| Blood Pressure | 120 |

| Patient Health Data | |
| --- | --- |
| **Metric** | **Value** |
| Blood Pressure | 120 |
| Heart Rate | 75 |
| Blood Sugar | 95 |
| Cholesterol | 180 |

