**HealthAI: Intelligent Healthcare Assistant Using IBM Granite Documentation**

**Introduction:**

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



Team member: Lakshaya M

Team member: Faahin Fathima M  
Team member: Nandhini S

Team member: Pavithra B(4/12/2005)

**Project overview**

**Purpose :**

The purpose of HealthAI is to assist patients, healthcare professionals, and general users by providing intelligent, AI-powered healthcare support. By leveraging LLMs (IBM Watsonx Granite) and real- time data, HealthAI offers:

 Symptom-based disease predictions

 Personalized treatment and lifestyle recommendations  Medical history–aware suggestions

 Easy access to reliable healthcare information

This assistant acts as a decision-support partner for medical professionals and a guidance tool for patients, ensuring better health awareness, preventive care, and timely medical attention.

**Features:**

**Symptom Analysis (Disease Prediction)**

*Key Point: Early health risk detection*

*Functionality: Users enter symptoms, and the system predicts possible conditions with recommendations.*

**Personalized Treatment Plans**

*Key Point: Tailored healthcare guidance*

*Functionality: Provides lifestyle, medication, and diet suggestions based on user profile (age, gender, history).*

**Medical History Integration** *Key Point: Context-aware AI Functionality: Takes into account allergies, past conditions, or chronic diseases.*

**Medication & Lifestyle Advice**

*Key Point: Preventive healthcare support*

*Functionality: Takes into account allergies, past conditions, or chronic diseases.*

**Report Summarization**

*Key Point: Simplified medical understanding*

*Functionality: Converts long medical documents or reports into easy-to-read summaries.*

**Anomaly Detection (Vitals/Reports)**

*Key Point: Early warnings*

*Functionality: Flags abnormal patterns in patient vitals or lab reports.*

**Multimodal Input Support**

*Key Point: Flexible healthcare data handling*

*Functionality: Accepts text, PDFs, and CSVs (lab reports, prescriptions, datasets).*

**Gradio UI (Frontend)**

*Key Point: User-friendly interface*

*Functionality: Provides a clean, accessible dashboard for patients and doctors.*

**Architecture**

**Frontend (Gradio)**

 Built using Gradio for interactive UI.  Pages include:

 Symptom analysis

 Treatment recommendations

 File upload (medical history/reports)  Chat assistant for health queries

**Backend (FastAPI)**

 FastAPI powers all API endpoints:  Symptom analysis

 Report summarization

 Treatment plan generation  Chatbot responses

**Vector Database (Pinecone)**

 Stores medical knowledge base embeddings.

 Supports semantic search of medical literature and reports.

**ML Modules (Forecasting & Anomaly Detection)**

 Forecast patient vitals trends.

 Detect anomalies in medical records or lab results using Scikit- learn and pandas.

**Setup Instructions**

**Prerequisites:**

 Python 3.9+

 pip + virtual environment tools

 API keys for IBM Watsonx and Pinecone  Internet access

**Installation Process:**

Clone the repository

Install dependencies via requirements.txt 3.Configure .env with API keys (IBM, Pinecone)

Start the FastAPI backend

Launch the Gradio UI

Upload symptoms/data and interact with the assistant

**Folder Structure**

* app/ – Backend logic (APIs, models, integrations)
* app/api/ – Symptom analysis, treatment, summarization routes ui/ – Gradio UI components
* granite\_llm.py – Handles IBM Watsonx Granite integration medical\_embedder.py – Converts medical docs into embeddings (Pinecone) anomaly\_checker.py – Detects unusual patterns in health data report\_generator.py – AI-generated medical summaries and recommendations
* Running the Application
* Start the FastAPI backend server
* Run the Gradio UI
* Navigate across modules (symptoms, treatment, reports)
* Upload medical files for summarization/predictions
* Interact with the chatbot for personalized health guidance
* API Documentation
* POST /symptoms/analyze → Returns possible conditions POST /treatment/plan → Generates treatment plan POST
* /upload-doc → Upload and embed reports GET /search- docs → Retrieve similar medical documents GET /health- tips → AI-powered wellness tips POST /feedback → Collects patient feedback

**Authentication**

For secure deployment:

Token-based authentication (JWT/API keys)  Role-based access (doctor, patient, admin)



 Future: Session tracking and patient history

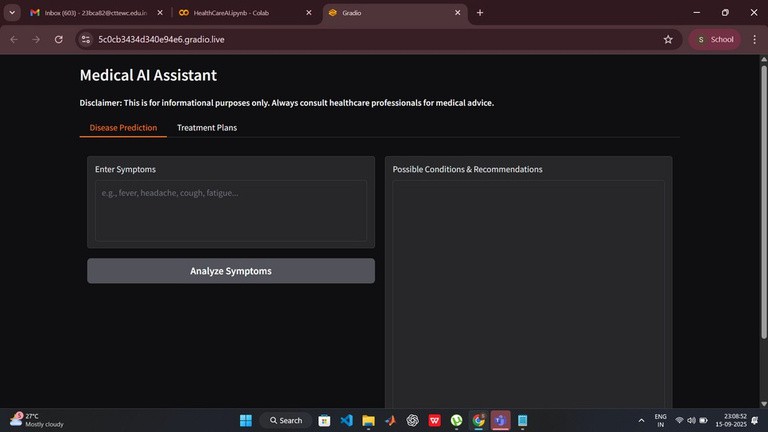
**User Interface**

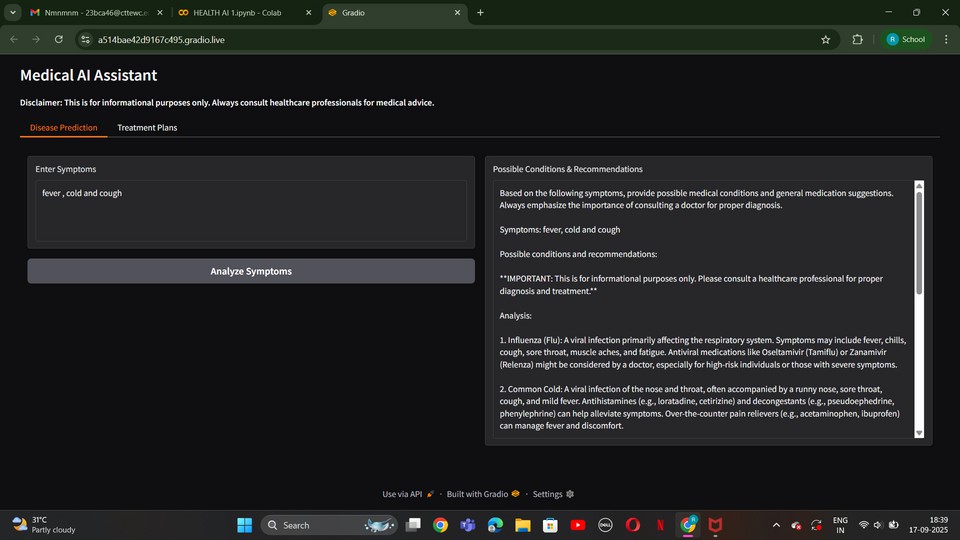
* Tabs: Disease Prediction | Treatment Plans | Medical Report Summarization
* Input: Symptoms, medical history, reports
* Output: Possible conditions, lifestyle tips, and AI-generated reports Extra: PDF download of treatment plan

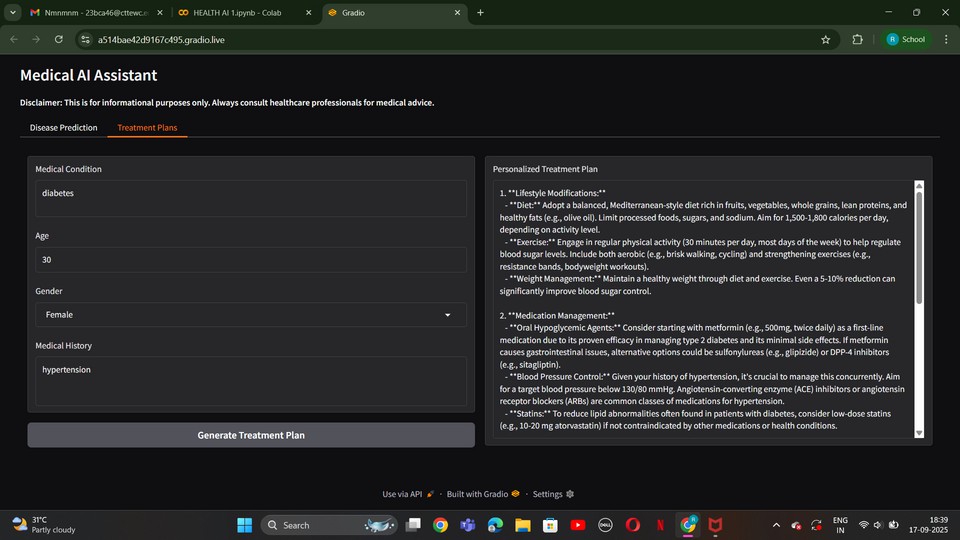
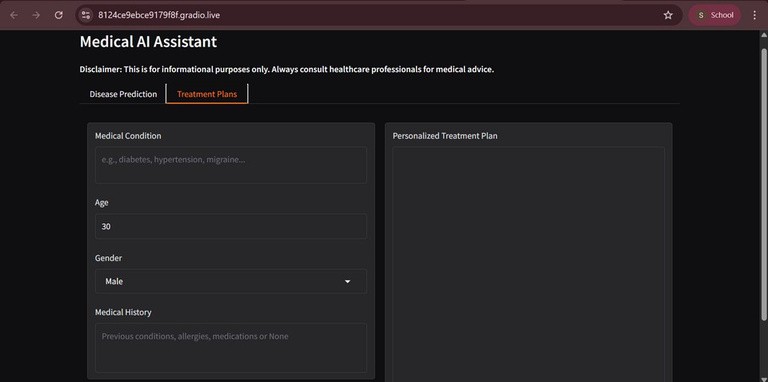
**Testing**

* Unit Testing: For prediction and summarization modules API Testing: Swagger/Postman
* Manual Testing: User flows in Gradio
* Edge Case Testing: Rare symptoms, empty inputs, invalid files

**Screenshots**

****



****

**Known Issues**

* Limited accuracy for rare diseases
* Dependent on quality of medical dataset embeddings Requires medical professional validation

**Future enhancement**

* Real-time IoT integration (patient monitoring)  Voice-based interaction for accessibility
* Multi-language support
* Integration with hospital EHR systems

