# Health AI

# Project Documentation

## 1.Introduction

- Project title : Health AI
- Team member : Sayyeda Taskeen Fatema
- Team member : Wafa Zainab T
- Team member : Fizza Banu K
- Team member : Ayesha Fathima S

## 2.project overview

- Purpose :
- The purpose of HealthAI is to empower individuals with accessible, AI-driven healthcare assistance. By harnessing IBM Watson Machine Learning and Granite-13b-instruct-v2, the system provides intelligent insights into medical queries, disease predictions, treatment plans, and health analytics. HealthAI ensures that patients can make informed health decisions with confidence through a user-friendly platform.
- Features:

**Patient chat**

*Key Point:* Conversational healthcare assistant

*Functionality:* Allows users to ask health-rela6ted questions in natural language and receive accurate AI-driven responses.

**Disease Prediction**

*Key Point:* Symptom-based condition evaluation

*Functionality:* Analyses user-reported symptoms and health data to predict possible conditions with likelihood assessments.

**Treatment plans**

*Key Point:* Personalized medical guidance

*Functionality:* Generates evidence-based recommendations including medication guidance, lifestyle modifications and follow-up steps.

**Health Analytics**

*Key Point:* Visual health monitoring
*Functionality:* Tracks and visualizes health metrics (e.g., blood pressure, glucose, heart rate) and provides AI-powered improvement insights.

**Secure API Key Management**
*Key Point:* Data protection
*Functionality:* Ensures safe integration with IBM Watsonx Granite and responsible handling of sensitive health data.

**Streamlit-Based UI**
*Key Point:* Simple and accessible dashboard
*Functionality:* Provides intuitive navigation for patient chat, predictions , analytics, and personalized recommendations.

## 3. Architecture

**Frontend (Stream lit):**

Built with Stream lit, offering an interactive healthcare dashboard with sections for patient chat, disease prediction, treatmeant planning, and health analytics.

**Backend (Fast API):**

Handles API endpoints for chat interactions, disease prediction, treatment generation, and analytics visualization. Optimized for secure and scalable healthcare data handling.

**LLM Integration (IBM Granite-13b-instruct-v2):**

Power natural language understanding for medical queries, symptom analysis, and treatment recommendations.

**Health Data Management:**

Patient inputs and health records are securely processed, analyzed, and visualized.

**ML Modules (Prediction & Insights):**

Machine learning models analyze user symptoms and health metrics to generate predictions, trends, and alerts.

## 4. Setup Instructions

**Prerequisites:**

- Python 3.9 or later
- pip and virtual environment tools
- API keys for IBM Watsonx Granite
- Internet access to access cloud services

**Installation Process:**

- Clone the repository
- Install dependencies from requirements.txt
- Configure a .env file with API credentials
- Run the backend server using Fast API
- Launch the frontend via Stream lit
- Interact with chat, predictions, and analytics

## 5. Folder Structure

app/ – Contains all Fast API backend logic including APIs for chats, predictions and analytics.

app/api/ – Modular endpoints for disease prediction, treatment plans, health insights.

ui/ – Streamlit frontend for dashboards and patient interaction.

health_chat.py – Handles AI-driven patient converstaions.

disease_predictor.py – Symptom analysis and condition prediction

treatment_generator.py – Personalized treatment plan creation

analytics_dashboard.py – Visualization of patient health metrics

granite_llm.py – Handles communication with IBM Granite LLM.

## 6. Running the Application

To start the project:

1. Launch the FastAPI backend server

2. Start the Streamlit dashboard

3. Navigate via sidebar (Chat, Prediction, Treatment, Analytics)

4. Enter symptoms, conditions, or health data

5. View predictions, treatment suggestions, and insights in real-time

**Frontend (Stream lit):**

Start the Streamlit dashboard to access the web interface.

Navigate through pages via the sidebar (Chat, Prediction, Treatment, Analytics).

User Flow:

Enter symptoms in the Disease Prediction page → view possible conditions.

Use the Treatment Plan page for personalized recommendations.

Explore the Health Analytics dashboard to visualize health metrics.

Chat with the AI assistant for general health-related queries.

**Backend (Fast API):**

Launch the FastAPI server to expose API endpoints for chat, disease prediction, treatment generation, and health analytics.

## 7. API Documentation

Backend APIs available include:

POST /chat/ask – Patient chat queries

POST /predict-symptoms – Symptom-based disease prediction

POST /generate-treatment – Personalized treatment plan generator

POST /upload-health-data – Upload patient metrics for analytics

GET /get-insights – AI-generated health trend insights

## 8. Authentication

Authentication in HealthAI is designed to ensure secure access to sensitive healthcare data and protect patient privacy. Since the platform handles medical queries, patient health metrics, and treatment recommendations, robust authentication is critical.

However, secure deployments can integrate:

- Token-based authentication (JWT or API keys)
- OAuth2 with IBM Cloud credentials
- Role-based access (patient, doctor, admin)
- Future:session history tracking & multi-user profilesOAuth2 with IBM Cloud credentials

## 9. User Interface

The interface is minimalist and functional, focusing on accessibility for nontechnical users. It includes:

Sidebar with navigation Sidebar navigation for features

Chat interface for Q&A

Disease prediction and treatment dashboards

Health analytics with visual charts

Downloadable PDF reports for medical summaries

The design prioritizes clarity, speed, and user guidance with help texts and intuitive flows.

## 10. Testing

Testing was done in multiple phases:

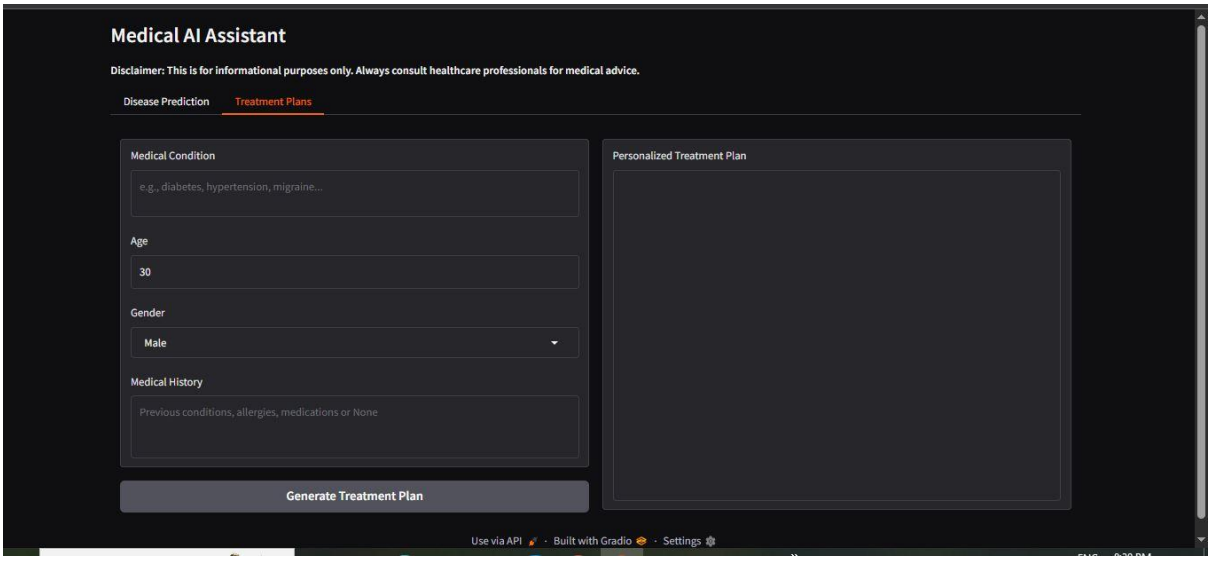
Unit Testing: Symptom analysis, chat response validation

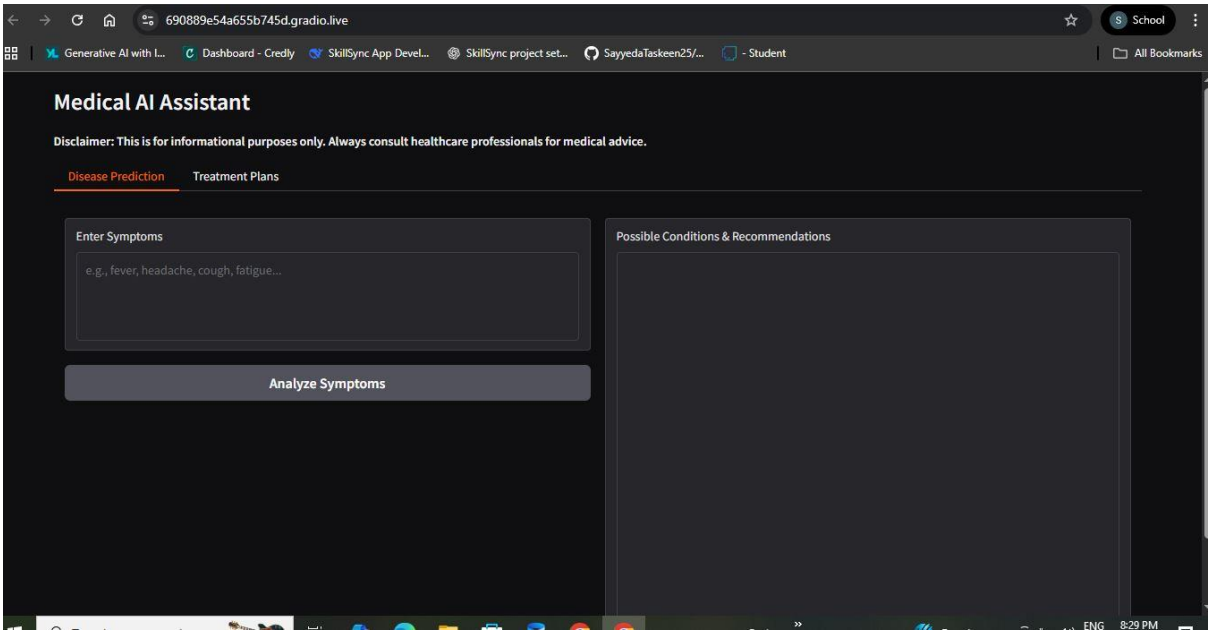
API Testing: Swagger UI, Postman

Manual Testing: Patient flow (chat → prediction → treatment)

Edge Cases: Unclear symptoms, large data files, invalid API keys

## 11.screen shots






## 12. Known Issues

## 13. Future enhancement