

HealthAI: Intelligent Healthcare Assistant Using IBM Granite

Project Documentation

1. Introduction

Project Title:

HealthAI: Intelligent Healthcare Assistant Using IBM Granite

Team Members:

- Member 1: Deepak R
- Member 2: Gopinath S
- Member 3: Nagappan K
- Member 4: Dhanush

Introduction:

Healthcare systems worldwide face challenges such as **delayed diagnosis, lack of access to doctors in rural areas, and overwhelming patient loads** in hospitals. With the growth of **Artificial Intelligence (AI) in Natural Language Processing (NLP)**, there is a rising opportunity to build **AI-powered healthcare assistants** that provide **informational support to patients**.

The **Medical AI Assistant** is an intelligent system that enables users to:

1. Enter **symptoms** and receive possible medical conditions with general recommendations.
2. Provide **personal information (age, gender, medical history, condition)** to receive a **personalized treatment plan** with general guidance.

⚠ Disclaimer: This tool is for **informational and educational purposes only**. It cannot replace professional medical advice, diagnosis, or treatment.

2. Project Overview

Purpose:

- Provide **instant informational support** to patients about possible health conditions.
- Encourage users to **seek medical consultation** when symptoms may indicate serious conditions.
- Promote **awareness of home remedies and lifestyle adjustments** for minor illnesses.

Objectives:

1. Build an **AI-based medical assistant** accessible through a web interface.

2. Enhance **early awareness** of possible health conditions.
3. Support **educational use** in medical training environments.
4. Ensure **ethical AI use** by emphasizing doctor consultation.

Features:

1. **Disease Prediction**
 - Input: Symptoms (fever, cough, headache, fatigue, etc.)
 - Output: Possible conditions + general advice.
 - Example: “fever, cough, fatigue” → possible flu, COVID-19, common cold.
2. **Personalized Treatment Plans**
 - Input: Medical condition, age, gender, history.
 - Output: Generalized treatment suggestions + home remedies.
 - Example: Diabetes (Age: 45, Male, history of hypertension) → Suggest diet changes, glucose monitoring, exercise.
3. **Gradio Web Interface**
 - Intuitive tabs for **disease prediction** and **treatment generation**.
 - Works in **Google Colab** or local Python environment.

3. Architecture

Frontend (Gradio)

- Interactive UI built using **Gradio Blocks**.
- Tabs for **Disease Prediction** and **Treatment Plans**.
- Real-time input and output updates.

Backend (IBM Granite + PyTorch)

- Model: **ibm-granite/granite-3.2-2b-instruct**
- Runs with Hugging Face Transformers and PyTorch.
- GPU acceleration for performance (in Colab).

Core Modules

- `generate_response()` – Handles LLM inference.
- `disease_prediction()` – Maps symptoms → conditions.
- `treatment_plan()` – Generates treatment suggestions.

Data Flow:

1. User inputs →
2. Preprocessing (tokenization) →
3. LLM Inference →
4. Output displayed in Gradio.

4. Setup Instructions

Prerequisites:

- Python 3.9+
- Google Colab or local Python environment
- Install required packages:
- pip install gradio torch transformers

Installation Steps (Colab):

1. Open Google Colab
2. Paste the code into a new cell
3. Run cell → dependencies install, model loads
4. Gradio generates a public link
5. Open the link in browser

5. Folder Structure

```
medical-ai-assistant/
|—— app.ipynb          # Main project notebook
|—— requirements.txt    # Dependencies
|—— README.md          # Documentation
```

6. Running the Application

1. Run the notebook in Colab.
2. Wait for model loading (~500MB).
3. Click the **Gradio link**.
4. Use:
 - **Disease Prediction Tab** → Input symptoms → AI suggests conditions.

- **Treatment Plans Tab** → Enter patient info → AI generates plan.
-

7. API Documentation (Internal Functions)

- **generate_response(prompt, max_length)**
 - Input: String prompt
 - Output: AI-generated text
- **disease_prediction(symptoms)**
 - Input: Symptom list
 - Output: Possible conditions and advice
- **treatment_plan(condition, age, gender, medical_history)**
 - Input: Patient details
 - Output: Personalized treatment suggestions

8. Authentication

- Current version: Open demo.
- Planned enhancements:
 - **API key support**
 - **Role-based access** (doctor, patient, researcher)
 - **Secure storage** for medical history

9. User Interface

- **Tabs:** Disease Prediction, Treatment Plans.
- **Inputs:**
 - Text (symptoms, conditions, history).
 - Age (numeric).
 - Gender (dropdown).
- **Outputs:**
 - Possible conditions.
 - Treatment plan.

10. Testing

Unit Testing

- Check accuracy of function calls (disease_prediction, treatment_plan).

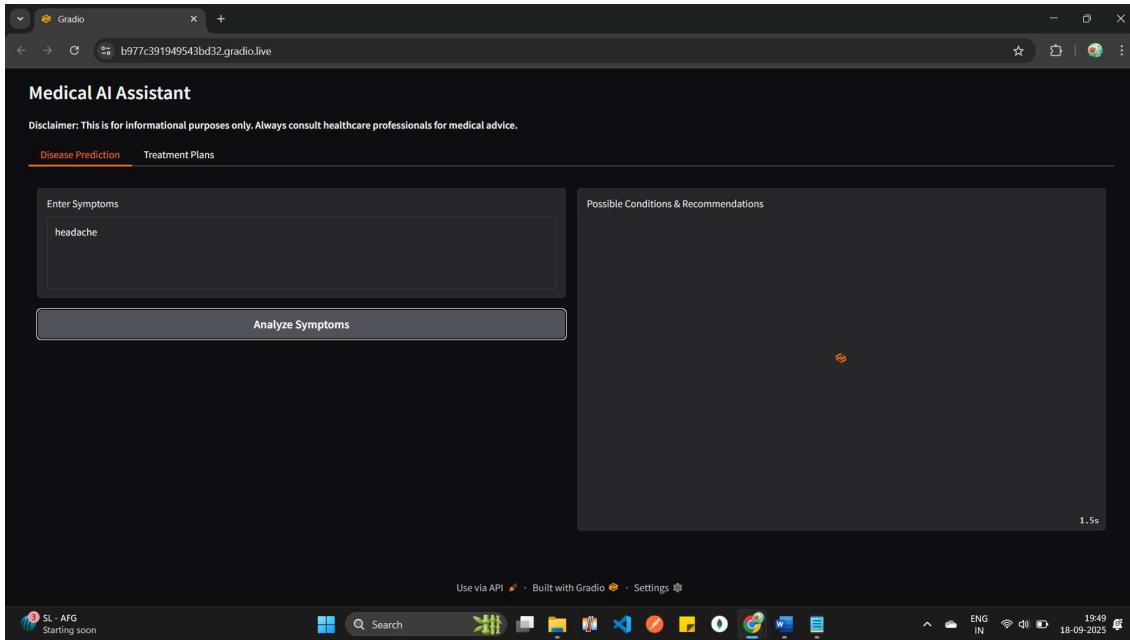
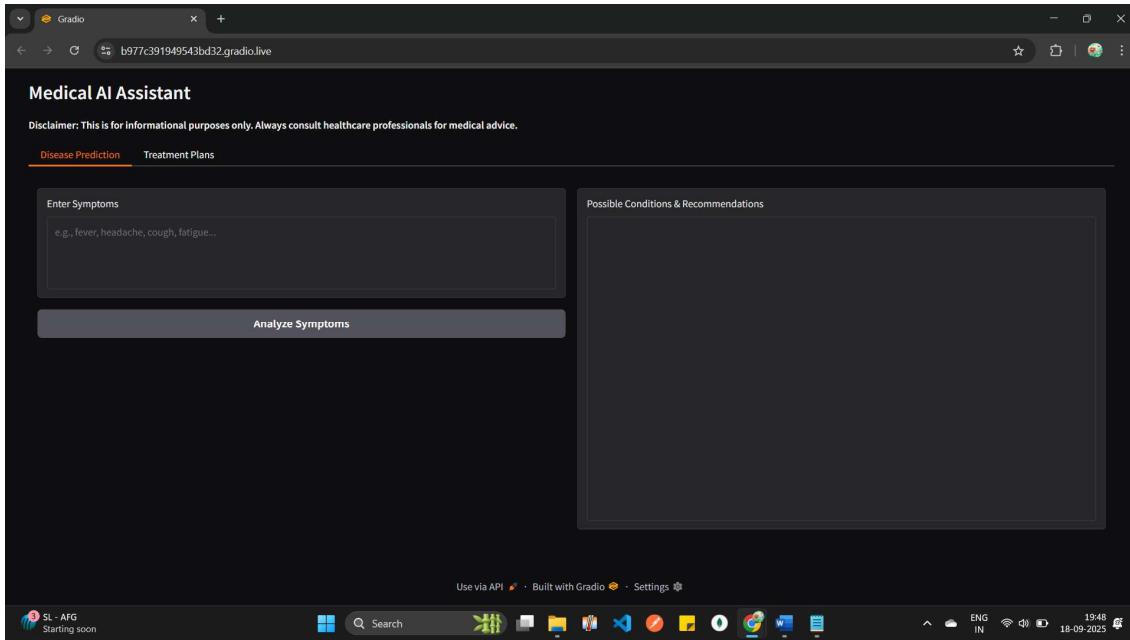
Manual Testing

- Input different symptoms and verify responses.
- Ensure disclaimers are included.

Edge Cases

- Empty inputs.
- Rare diseases.
- Unclear medical history.

11. Screenshots



The screenshot shows the 'Medical AI Assistant' application running in a browser window. The title bar reads 'Gradio' and the URL is 'b977c391949543bd32.gradio.live'. The main header says 'Medical AI Assistant' with a disclaimer: 'Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.' Below the header, there are two tabs: 'Disease Prediction' (which is active) and 'Treatment Plans'. A large input field labeled 'Enter Symptoms' contains the text 'headache'. Below it is a button labeled 'Analyze Symptoms'. To the right, a panel titled 'Possible Conditions & Recommendations' lists various types of headaches and their treatments. At the bottom of this panel is a note about medication side effects. The status bar at the bottom of the screen shows 'SL - AFG Starting soon' and a system tray with icons for search, file explorer, and other applications.

The screenshot shows the 'Medical AI Assistant' application running in a browser window. The title bar reads 'Gradio' and the URL is 'b977c391949543bd32.gradio.live'. The main header says 'Medical AI Assistant' with a disclaimer: 'Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.' Below the header, there are two tabs: 'Disease Prediction' (which is active) and 'Treatment Plans' (which is currently inactive). On the left, there are several input fields: 'Medical Condition' (with placeholder text 'e.g., diabetes, hypertension, migraine...'), 'Age' (set to '30'), 'Gender' (set to 'Male'), and 'Medical History' (with placeholder text 'Previous conditions, allergies, medications or None'). Below these is a button labeled 'Generate Treatment Plan'. To the right, there is a large, empty box labeled 'Personalized Treatment Plan'. The status bar at the bottom of the screen shows 'Finance headline India's new jobs...' and a system tray with icons for search, file explorer, and other applications.

Medical AI Assistant

Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.

Disease Prediction Treatment Plans

Medical Condition
diabetes

Age
25

Gender
Female

Medical History
allergies

Generate Treatment Plan

Personalized Treatment Plan

1.1s

Use via API 🔍 · Built with Gradio 🎨 · Settings ⚙

Finance headline India's new jobs... Windows Search Bar Taskbar icons ENG IN 19:50 18-09-2025

This screenshot shows the 'Treatment Plans' tab selected in the Medical AI Assistant interface. The user has inputted 'diabetes' as the medical condition, '25' as the age, 'Female' as the gender, and 'allergies' as the medical history. A 'Generate Treatment Plan' button is visible. On the right, a 'Personalized Treatment Plan' section is displayed, which includes a summary of the input and a detailed list of recommendations under 'Lifestyle Modifications' and 'Medication Management'. The interface is dark-themed with orange highlights for active tabs and buttons.

Medical AI Assistant

Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.

Disease Prediction Treatment Plans

Medical Condition
diabetes

Age
25

Gender
Female

Medical History
allergies

Generate Treatment Plan

Personalized Treatment Plan

1. Lifestyle Modifications

- **Diet:** Adopt a balanced, low-glycemic index diet rich in whole grains, lean proteins, fruits, vegetables, and healthy fats. Limit processed foods, sugars, and saturated fats. Consider meal planning and tracking to maintain consistent blood sugar levels.
"Home Remedy:" Incorporate chia seeds, nuts, and avocados, which are high in fiber and can help regulate blood sugar.
- **Exercise:** Engage in regular physical activity, aiming for at least 150 minutes of moderate-intensity aerobic exercise or 75 minutes of vigorous activity each week, along with strength training exercises at least 2 days a week. Exercise helps improve insulin sensitivity and overall glucose metabolism.
"Home Remedy:" Try yoga or pilates, which combine flexibility, strength, and mindfulness; they can be gentle on the joints and still provide benefits for blood sugar control.
- **Weight Management:** Maintain a healthy weight, if overweight or obese. Every 1.2 kg (2.6 lbs) of excess weight can lower insulin sensitivity by up to 11%.
"Home Remedy:" Focus on portion control and prioritize nutrient-dense foods to promote sustainable weight loss.

2. Medication Management

Use via API 🔍 · Built with Gradio 🎨 · Settings ⚙

Very humid Now Windows Search Bar Taskbar icons ENG IN 19:51 18-09-2025

This screenshot shows the 'Treatment Plans' tab selected in the Medical AI Assistant interface. The user has inputted 'diabetes' as the medical condition, '25' as the age, 'Female' as the gender, and 'allergies' as the medical history. A 'Generate Treatment Plan' button is visible. On the right, a 'Personalized Treatment Plan' section is displayed, which includes a summary of the input and a detailed list of recommendations under 'Lifestyle Modifications' and 'Medication Management'. The interface is dark-themed with orange highlights for active tabs and buttons.

12. Known Issues

- Model may give **generic advice**.
- No integration with verified medical databases.
- Cannot handle **real-time emergencies**.
- First run is slow due to model download.

13. Future Enhancements

- Integration with **WHO/CDC health databases**.
- Add **drug interaction warnings**.
- Support **voice input/output**.
- Provide **diet and exercise recommendations**.
- Enable **multi-language support**.

14. Ethical & Safety Considerations

- **Medical Disclaimer:**
Informational only, not a substitute for a doctor's consultation.
- **Data Privacy:**
Current version doesn't store patient data. Future versions can add **GDPR/HIPAA compliance**.
- **Bias Risks:**
AI-generated results may not cover all populations equally. Continuous refinement required.

15. Use Cases

1. **Patients:** Quick insights into symptoms before visiting a doctor.
2. **Doctors:** Support tool for preliminary analysis.
3. **Medical Students:** Educational use in diagnosis and treatment planning exercises.
4. **Rural Communities:** Preliminary awareness in areas with limited healthcare access.

16. Societal Impact

- Promotes **health awareness** among users.
- Reduces unnecessary hospital visits for minor illnesses.

- Encourages **early diagnosis** and medical consultation.
- Provides **educational support** in medical training.

17. Limitations

- Cannot provide emergency medical support.
- Dependent on internet connectivity.
- Accuracy varies based on input clarity.
- Not suitable for life-threatening conditions.

18. References

- IBM Granite LLM Documentation (Hugging Face).
- PyTorch Documentation.
- Gradio UI Framework.
- WHO Guidelines for Symptom-based Diagnosis.