

HEALTH AI

- ❖ PROJECT TITLE: HEALTH AI
- ❖ TEAM MEMBER: THITIKSHA.D
- ❖ TEAM MEMBER: TAMILARASI.R
- ❖ TEAM MEMBER: JOSPHIN SYNTHIA.J
- ❖ TEAM MEMBER: VIJAYALAKSHIMI.R
- ❖ TEAM MEMBER: POOVARASI.D

INTRODUCTION:

The project is called **Medical AI Assistant**. It is a simple but powerful software tool that uses **Artificial Intelligence (AI)** to help people understand their health better.

- A user enters their **symptoms** (for example: fever, cough, headache).
- The system uses an AI model to suggest **possible medical conditions**.
- It can also create a **basic treatment plan** based on details such as age, gender, and medical history.

This project is not designed to replace doctors. Instead, it provides **general knowledge** and encourages users to consult a healthcare professional for proper medical advice. Every response comes with a disclaimer for safety.

THE SYSTEM IS BUILT USING:

- **IBM Granite Large Language Model (LLM)** for generating answers.
- **Gradio** for building a simple and interactive user interface.
- **Transformers** and **PyTorch** for model loading and execution.

This project shows how modern AI tools can be combined to create useful and educational applications in healthcare.

2. PROJECT OVERVIEW PURPOSE :

The main purpose of the Medical AI Assistant is to make **basic health information accessible** in simple language. Many people search online for their health problems, but they often find too much confusing or unreliable information. This tool provides **short, clear, and direct responses** that are easy to understand.

FEATURES

- **Disease Prediction:** Enter symptoms and receive possible conditions.
- **Treatment Plan:** Provide condition, age, gender, and history to get a general treatment plan.
- **Conversational AI:** Uses IBM Granite to respond in natural, human-like language.
- **Easy-to-Use Interface:** Gradio Tabs make navigation simple for anyone.
- **Safety Measures:** Built-in disclaimers to remind users to consult real doctors.

Significance

Healthcare is one of the most sensitive areas for AI. This project shows how AI can **support** but not **replace** doctors.

IT IS USEFUL FOR:

- Students learning about AI and healthcare.
- Patients who want general guidance.
- Developers as a demo of real-world AI applications.

3.ARCHITECTURE:

THE SYSTEM HAS THREE MAIN COMPONENTS:

1. Frontend (Gradio UI)

- The user interacts with the system through a web page.
- Two main tabs: **Disease Prediction** and **Treatment Plan**.
- Designed for non-technical users with simple inputs and outputs.

2. Backend (Python Functions)

- Core functions handle the logic of the app.
- `disease_prediction()` prepares prompts from symptoms.
- `treatment_plan()` builds prompts from user details.
- `generate_response()` connects to the AI model and generates text.

3. Model (IBM Granite via Hugging Face)

- **Granite 3.2-2B Instruct** is the AI model used.
- Hugging Face **Transformers** library loads the model.
- **PyTorch** runs the model with GPU acceleration if available.

DATA FLOW

1. User enters symptoms or patient details in the interface.
2. Input is sent to the backend function.
3. The function creates a **prompt** for the AI model.
4. The AI model processes the input and generates a response.
5. The response is decoded and shown to the user.

4. SETUP INSTRUCTION REQUIERMENTS:

- Python 3.9 or newer
- PyTorch
- Hugging Face Transformers
- Gradio

Steps to Run

1. Install Python and required libraries.
2. Download or clone the project files.
3. Install dependencies using:

4. `pip install -r requirements.txt`
5. Start the app with:
6. `python app.py`
7. Gradio will open the application in your browser.

If a GPU is available, the system will use it for faster performance.

5. FOLDER STRUCTURE:

The suggested folder structure is:

- **app.py** → Main code file with all functions and Gradio interface.
- **requirements.txt** → List of required Python libraries.
- **/models** → (Optional) for storing model weights locally.
- **/tests** → Contains test scripts.
- **/docs** → Documentation files.

This structure makes the project easy to manage and expand.

6. RUNNING THE APPLICATION:

Steps to use the application:

1. Run the Python script: `python app.py`.
2. A Gradio link will appear in the terminal.
3. Open the link in your browser.
4. **Disease Prediction Tab**: Enter symptoms and click “Analyze”.
5. **Treatment Plan Tab**: Enter condition, age, gender, and medical history.
6. The system will generate the results instantly.

7. FUNCTION DOCUMENTATION:

Generate_response (prompt, max_length)

- Main function that talks to the AI model.
- Converts user text into tokens, runs the model, and returns the response.

Disease_Prediction(symptoms)

- Input: a list of symptoms.
- Builds a prompt like:
“Based on symptoms: fever, cough, headache...”
- Output: possible medical conditions and general advice.
-

Treatment_Plan (condition, age, gender, medical_history)

- Input: condition name and patient details.
- Creates a prompt for the AI model.
- Output: a treatment plan with home remedies and general suggestions.
- Reminder: always consult a doctor.

8.USER INTERFACE :

The user interface is built with **Gradio Blocks**.

- **Disease Prediction Tab:** Input box for symptoms, a button to analyze, and an output box.
- **Treatment Plan Tab:** Input boxes for condition, age, gender, and medical history, plus a button to generate the plan.
- **Disclaimer:** Displayed clearly at the top to ensure safe use.

The design is simple and clear so that anyone can use it without training.

9.TESTING:

Testing was done in different ways:

- **Unit Testing:** Each function was tested separately (for example, checking if `generate_response()` gives output).

- **Integration Testing:** Verified that data flows correctly from the UI to the backend functions and then to the model.
- **Manual Testing:** Entered different symptoms and medical conditions to check results.
- **Edge Case Testing:** Tried empty inputs, very long inputs, and invalid entries.

This process ensured that the app works smoothly and gives meaningful results.

10.KNOWN ISSUES AND LIMITATIONS:

- The tool cannot replace a real doctor.
- The model may sometimes give inaccurate or general advice.
- Requires internet access to download the model.
- Works only in **English** for now.

11.FUTURE ENCHANTMENTS:

The project can be improved in many ways:

- Add **multilingual support** for wider use.
- Connect to trusted **medical databases** for more accurate responses.
- Create a **mobile-friendly version**.
- Add **user login** and history tracking.
- Improve accuracy by training the AI on specialized healthcare data.

12. SCREENSHOTS :

This section can include screenshots of the app if available, such as:

- Disease Prediction Tab.
- Treatment Plan Tab.

DISEASE

Medical AI Assistant

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

Disease Prediction

Treatment Plans

Enter Symptoms

cough

Analyze Symptoms

Possible Conditions & Recommendations

The primary symptom you've mentioned is a cough. A cough can be a manifestation of various conditions, both acute and chronic. Categorizing possible conditions based on the duration and severity of the cough:

- Acute cough (lasting less than 3 weeks):**
 - Upper Respiratory Infection (URI):** Commonly known as a cold, this viral infection can cause a persistent cough. While there's no specific antibiotic for viral URIs, supportive care such as rest, hydration, and over-the-counter (OTC) medications can help manage symptoms. Consider using decongestants, expectorants, or anti-tussives for cough relief. Examples of OTC medications include:
 - Pseudoephedrine or Phenylephrine (decongestants)
 - Guaifenesin (expectorant)
 - Dextromethorphan (anti-tussive)
- Persistent or chronic cough (lasting more than 8 weeks):**
 - Chronic Obstructive Pulmonary Disease (COPD):** This includes conditions like emphysema and chronic bronchitis, which are often associated with long-term smoking. While there's no cure for COPD, various medications can help manage symptoms and slow disease progression. Consult a doctor for appropriate treatment options, which may include:
 - Inhaled corticosteroids (e.g., Budesonide, fluticasone)

Use via API - Built with Gradio - Settings

TREATMENT

Medical AI Assistant

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

Disease Prediction

Treatment Plans

Medical Condition

pcod

Age

18

Gender

Female

Medical History

none

Generate Treatment Plan

Personalized Treatment Plan

- Lifestyle Modifications:**
 - Maintain a balanced diet:** Encourage a calorie-restricted diet (500-800 kcal/day) with a focus on whole, unprocessed foods, lean proteins, and healthy fats. Include non-starchy vegetables, fruits, whole grains, and legumes. Limit processed foods, sugars, and saturated fats.
 - Regular exercise:** Prescribe 150 minutes of moderate-intensity aerobic activity (such as brisk walking, cycling, or swimming) and 75 minutes of moderate-to-vigorous strength training per week. Aim for at least two strength training sessions.
 - Stress management:** Promote techniques like yoga, meditation, deep breathing exercises, or mindfulness to help reduce stress and improve mental well-being.
- Home Remedies and Natural Supplements:**
 - Curcumin (Turmeric):** Add curcumin-rich turmeric to meals for its anti-inflammatory properties. A typical dosage is 500-1000 mg daily, divided into 2-3 doses.
 - Ashwagandha:** This adaptogenic herb can help manage stress and anxiety levels. A daily dosage of 500-1000 mg, divided into 2-3 doses, is recommended.
 - Mint (Mentha):** Incorporate peppermint or spearmint in beverages or tea for digestive benefits. A decoction of fresh mint leaves can be consumed daily.
 - Aloe vera gel:** Apply topically to soothe skin irritations or rashes associated with Polycystic Ovary Syndrome (PCOS).

Use via API - Built with Gradio - Settings