

Health AI- Intelligent Healthcare

Assistant

Generative AI with IBM



1.INTRODUCTION:

Health AI – Intelligent Healthcare Assistant

Team Members

- R. Prathiksha
- S. Roobika
- G. Mansi

2.PROJECT OVERVIEW :

Health AI-intelligent healthcare Assistant is an AI-driven system designed to improve healthcare delivery through intelligent support. The solution integrates AI/ML models with LLMs and vector databases, user-friendly healthcare assistance.

- **Conversation Interface**

The conversational interface allows users to interact with the healthcare natural text or voice. It supports health queries, policy guidance and human experts.

- **Policy summarization**

The system automatically extracts and condenses healthcare policies into simple, easy-to-read summaries. It helps patients, citizens and staff quickly understand key update.

- **Eco-Tip Generator**

The eco-tip generator provides simple, actionable health and environment-friendly suggestions. It promotes sustainable practices like waste reduction, energy, saving methods for community health.

- **Citizen feedback loop**

The citizen feedback loop collects patient and public opinion through survey, chat, or forms. It analyses feedback using sentiment and topic detection to improve system enhancement.

- **KPI Forecasting**

KPI forecasting predicts key healthcare performance indicators such as bed occupancy, patients' inflow, staff availability, and resource usage. It helps in planning, decision-making, improving overall Healthcare efficiency.

- **Anomaly Detection**

Anomaly detection identifies unusual pattern in healthcare data, such as sudden diseases spikes or abnormal resources usage. It enables early alerts and quick action to improve patient safety.

- **Multimodal input support**

Multimodal input support allows a healthcare assistant to receive and process text, voice, image and sensor data. This enables more natural, accurate, and efficient interaction for patients and medical staff.

- **Stream lit to Gradio UI**

Stream lit to Gradio UI allows converting python apps into interactive web interface with easy deployment. Gradio provide simple drag-and-drop components for inputs and outputs, user interaction.

3.ARCHITECTURE:

- Front-end Architecture of Health AI-Intelligent Healthcare assistant interaction such as web, mobile or voice app, allowing patients or citizens to accessible way.

- Back-end Architecture of Health AI-Intelligent Healthcare assistant handles business logic, connect LLM for prediction and recommendations provide APLs for secure, real-time communication with the front end

- **LLM Integration:**

The LLM Processes user queries, interpret intent, and generate response while interacting with for knowledge retrieval and personalized recommendations.

- **Vector sector:**

The vector sector in health care AI stores embedding of medical data, policies, records to enable fast semantic search, allowing the AI to Retrieve contextually relevant information guidance.

- **ML Modules:**

The ML Modules provide analytics, anomaly detection and personalized health recommendations by analysing patient data, trends medical knowledge to support.

4.SETUP INTRODUCTION:

- **Prerequisites:**

- Install python, node and code editor
- Access open AI API and vector database
- Basic ML knowledge and libraries
- Front end skills (react, angular or flutter)
- Medical database or pre-trained models

- **Installation process:**

- Install python, node and set up a virtual environment.
- Install backend and AI libraries
- Setup vector database and create index.
- Run back end and front end test queries and AI responses.

5.FOLDER STRUCTURE:

- App: application code
- Data: Storage data files
- Docs: Documentation
- Test: Unit test

- Ven: Virtual environment
- Txt: Text
- Utils: Utility function

6.RUNNING THE APPLICATION:

To start the project:

- Open terminal and clone the project from GitHub.
- Navigate to the project folder
- Health AI-Intelligent healthcare assistant-AI.
- Create and activate a virtual environment.
- Install dependencies using pip install- r requirements. Txt.
- Set up database and vector store (initialize configs).
- Configure API keys (OpenAI/LLM, database, etc.).
- Start backend server using python backend/ API/main.py.
- Run frontend UI using stream lit run frontend/ app. Py Or gradio app. Py.

Frontend (Stream lit):

Enter frontend folder - run stream lit – open browser- use Health AI-Intelligent healthcare assistant UI.

Backend (fast API):

Enter backend folder – run fast API server – open API docs – backend ready for health AI-Intelligent healthcare assistant UI integration.

7.API DOCUMENTATION:

API Documentation provides endpoint for patient interaction, health record management, and AI – drive recommendations, enabling seamless healthcare support through conversational and analytical modules.

8.AUTHENTICATION:

Register user/client – user signs up or application register to receive API credentials.

Generate API key/token – system issues a unique API key or JWT token.

Send authentication request – client sends login request with credentials.

Receive access token – server validates and responds with an access token.

Use token in API calls – includes authorization: bearer <token>in header for every request.

9.USER INTERFACE:

- Provides a simple login for secure access.
- Chat window support text and voice queries.
- Dashboard shows health records and reports.
- AI gives personalized tips and recommendations.

10.TESTING:

- Testing in the health AI – intelligent healthcare assistant ensures the system works reliably and safety for patients and doctors. First, unit testing is performed to verify individual modules like authentication, chat, and health records access.
- User acceptance testing (UAT) then validate that real user finds the system useful, accurate, easy to use. This step-by-step testing process guarantees a robust and trustworthy healthcare assistant.

11.KNOWN ISSUES:

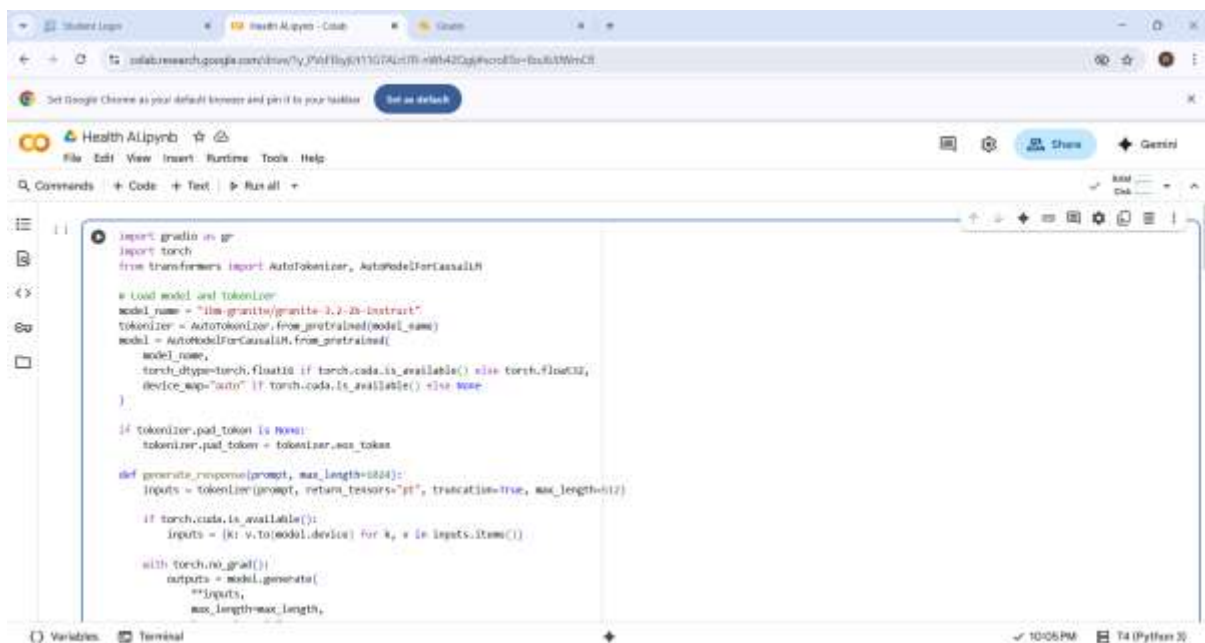
The Health AI-Intelligent healthcare assistant, while highly efficient, currently faces certain limitations. Occasional inaccuracies in symptom analysis may occur due to limited training data for rare conditions. Multimodal input processing, such as interpreting images and text together, may experience delay or misinterpretation under poor-quality inputs. Some users may encounter minor UI inconsistencies across different device these issues, improve reliability and enhance overall user experience. testing and initial deployment, several issues were identified that may affected the system performance or user experience. Some feature may response slowly under heavy load and occasionally UI observed on different device. Certain data inputs may trigger unexpected errors if they do not match the expected format additionally, integration with external APIs can server downtime will be provided in future releases to stability and functionality.

12.FUTURE ENHANCEMENTS:

In future as Health AI-Intelligent healthcare assistant update the system improvement for the health AI assistant will focus on expanding its capabilities and improving patient care. Planned enhancement include incorporating more advanced diagnosis tools, supporting multimodal input such as voice and image, and integrating with additional health data sources for personalized recommendations. The system will also implement predictive analytics for early disease detection, real- time health

monitoring and enhance natural language understanding for more accurate and empathetic interactions. The user interface will be refined for greater accessibility and responsibility while automated error handling and notification system will be implemented to ensure smoother operations. These enhancements aim to provide a more robust, efficient and user- friendly solution.

. 13PROJECT SCREENSHOT:



```
import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# Load model and tokenizer
model_name = "gpt3.5-turbo-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    device_map="auto" if torch.cuda.is_available() else None
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

def generate_response(prompt, max_length=1024):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=128)

    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}

    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
```

```
HealthAllynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text + Run all +
11
    max_length=max_length,
    temperature=0.7,
    do_sample=True,
    pad_token_id=tokenizer.eos_token_id
}

response = tokenizer.decode(outputs[0], skip_special_tokens=True)
response = response.replace(prompt, "").strip()
return response

def disease_prediction(symptoms):
    prompt = f"Based on the following symptoms, provide possible medical condition and general medication suggestions. Always emphasize the importance of consulting a doctor for return generate_response(prompt, max_length=2200)"

def treatment_plan(condition, age, gender, medical_history):
    prompt = f"Generate personalized treatment suggestions for the following patient information. Include how remedies and general medication guidelines, if any medical condition: [return generate_response(prompt, max_length=2200)"

# create gradio interface
with gr.Blocks() as app:
    gr.Markdown("Medical AI Assistant")
    gr.Markdown("Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.")

    with gr.Tab():
        with gr.TabItem("Disease Prediction"):
            with gr.Row():
                with gr.Column():
                    symptoms_input = gr.Textbox(
```

```
HealthAllynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text + Run all +
11
    symptoms_input = gr.Textbox(
        label="Enter symptoms",
        placeholder="e.g., fever, headache, cough, fatigue...",
        lines=4
    )
    predict_btn = gr.Button("Analyze Symptoms")

    with gr.Column():
        prediction_output = gr.Textbox(label="Possible conditions & recommendations", lines=20)

    predict_btn.click(disease_prediction, inputs=symptoms_input, outputs=prediction_output)

    with gr.TabItem("Treatment Plans"):
        with gr.Row():
            with gr.Column():
                condition_input = gr.Textbox(
                    label="Medical condition",
                    placeholder="e.g., diabetes, hypertension, migraine...",
                    lines=2
                )
                age_input = gr.Number(label="Age", value=40)
                gender_input = gr.Dropdown(
                    choices=["Male", "Female", "Other"],
                    label="Gender",
                    value="Male"
                )
                history_input = gr.Textbox(
                    label="Medical history",
```


Student LoginHealth Analyzer - ClinicGrade

Set Google Chrome as your default browser and pin it to your toolbarSet as default

Medical AI Assistant

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

[Browse Predictions](#)[Treatment Plans](#)

Enter Symptoms

Fever

Analyze Symptoms

Possible Conditions & Recommendations

1. Fever is a common symptom that can be caused by various factors, ranging from mild infections to serious diseases.

Possible conditions:

- 1.1. Influenza (Flu): A viral infection that primarily affects the respiratory system. Symptoms often include fever, chills, body aches, fatigue, and headache.
- 1.2. Pneumonia: A bacterial or viral infection that inflames the air sacs in one or both lungs, causing symptoms like fever, cough, chest pain, and difficulty breathing.
- 1.3. COVID-19: A highly contagious viral infection caused by SARS-CoV-2, known for its association with severe respiratory complications, fever, and other systemic symptoms like loss of taste or smell.
- 1.4. Other viral infections (e.g., common cold, rhinovirus): These can also cause fever, typically along with symptoms like nasal congestion, sore throat, and cough.
- 1.5. Bacterial infections: Depending on the location of the infection, bacteria such as Streptococcus, Staphylococcus, or Mycoplasma can potentially cause fever.

* 12 additional conditions have been excluded. For full list, please consult healthcare professionals.

Open in App

Build with Code

Settings

Student LoginHealth Analyzer - ClinicGrade

Set Google Chrome as your default browser and pin it to your toolbarSet as default

Medical AI Assistant

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

[Browse Predictions](#)[Treatment Plans](#)

Enter Symptoms

Fever

Analyze Symptoms

Possible Conditions & Recommendations

1.6. Immune-related or autoimmune disorders: Conditions like lupus, rheumatoid arthritis, or other autoimmune diseases may present with fever, in addition to other systemic symptoms.

General medication suggestions:

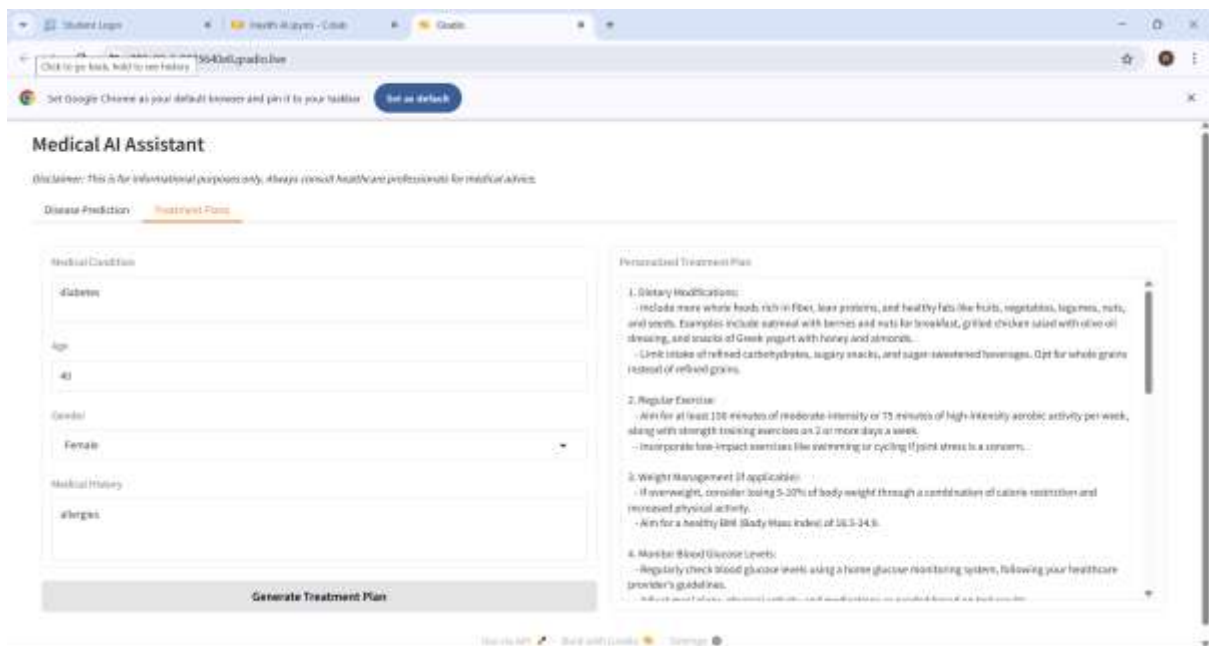
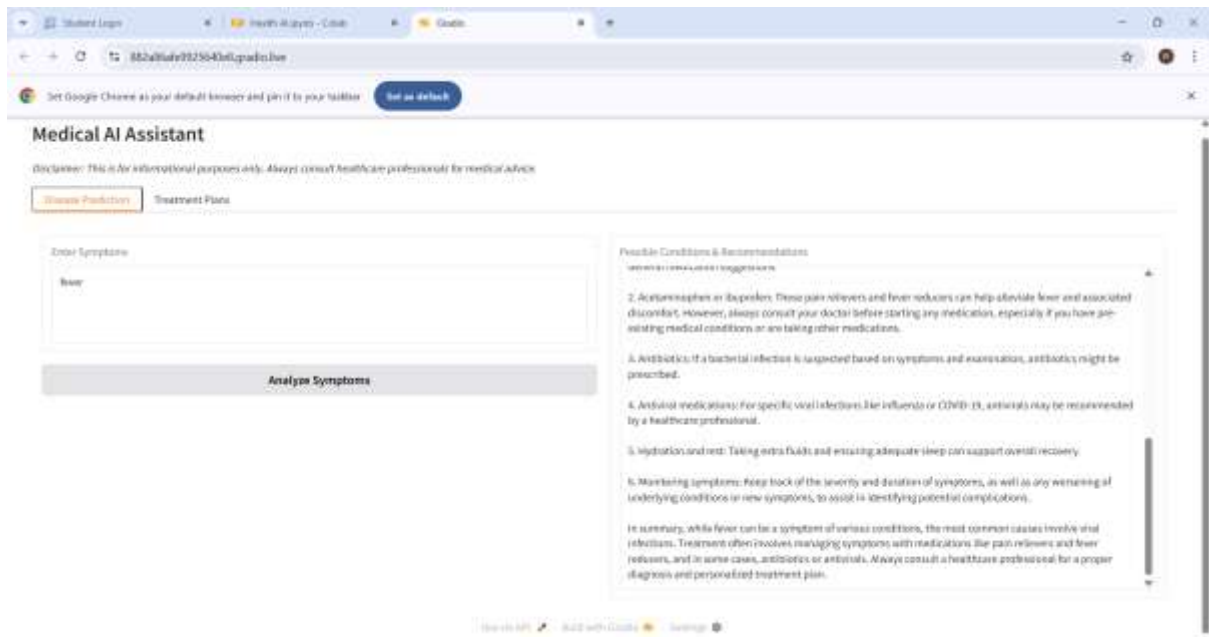
- 2. Acetaminophen or Ibuprofen: These pain relievers and fever reducers can help alleviate fever and associated discomfort. However, always consult your doctor before starting any medication, especially if you have pre-existing medical conditions or are taking other medications.
- 3. Antibiotics: If a bacterial infection is suspected based on symptoms and examination, antibiotics might be prescribed.
- 4. Antiviral medications: For specific viral infections like Influenza or COVID-19, antivirals may be recommended by a healthcare professional.
- 5. Hydration and rest: Taking extra fluids and ensuring adequate sleep can support overall recovery.
- 6. Monitoring symptoms: Keep track of the severity and duration of symptoms, as well as any worsening of underlying conditions or new symptoms, to assist in identifying potential complications.

In summary, while fever can be a symptom of various conditions, the most common causes involve viral

Open in App

Build with Code

Settings



Medical AI Assistant

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

[Disease Prediction](#)
[Treatment Plan](#)

Medical Condition

diabetes

Age

40

Gender

Female

Medical History

allergies

Generate Treatment Plan

Personalized Treatment Plan

- Regularly check blood glucose levels using a home glucose monitoring system, following your healthcare provider's guidelines.
- Adjust meal plans, physical activity, and medications as needed based on test results.

3. Medication Management:

- Oral Anti-Diabetic Medications:
 - Metformin (e.g., Fortamet, Glumetaz): Start with 500mg twice daily, increasing if necessary. It improves insulin sensitivity and reduces hepatic glucose production.
 - Sulfonylureas (e.g., Glipizide, Glyburide): Initiate at 2.5-5mg once daily, but may not be recommended due to potential side effects like hypoglycemia.
 - DPP-4 Inhibitors (e.g., Sitagliptin, Linagliptin): Begin with 100mg daily, usually as a single pill. They increase insulin hormone levels, improving beta-cell function.
 - GLP-1 Receptor Agonists (e.g., Semaglutide, Liraglutide): Start with low doses (e.g., 0.2-0.6mg once daily) and can be gradually increased. They mimic insulin hormones and stimulate insulin secretion while suppressing glucagon.
 - SGLT2 Inhibitors (e.g., Canagliflozin, Empagliflozin): Initiate at 30-100mg once daily, especially for those with kidney issues or obesity. They promote urinary glucose excretion and can lead to weight loss.
- Insulin Therapy (if lifestyle measures aren't sufficient):
 - Begin with basal insulin (e.g., Lantus, Toujeo) at low doses (e.g., 20-30 units/day) and consider adding prandial insulin (e.g., Novolog, Apidra) as needed for mealtime control.
 - Consult a diabetes educator or endocrinologist for individualized insulin dosing and injection techniques.

Medical AI Assistant

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

[Disease Prediction](#)
[Treatment Plan](#)

Medical Condition

diabetes

Age

40

Gender

Female

Medical History

allergies

Generate Treatment Plan

Personalized Treatment Plan

- Consult a healthcare provider for a comprehensive evaluation and to develop a personalized management plan based on your specific needs and preferences.

6. Home Remedies:

- Fenugreek Seeds: May help lower blood glucose levels. Add 1-2 teaspoons of fenugreek powder to meals or drinks.
- Cinnamon: Might improve insulin sensitivity. Add 1/2 to 1 teaspoon of cinnamon to food or beverages daily.
- Bitter Melon: Could potentially aid in glucose metabolism. Consume fresh or cooked bitter melon as a vegetable.
- Turmeric: Possesses anti-inflammatory properties and may support glycemic control. Add curcumin-rich foods (e.g., curry) or curcumin supplements, following healthcare provider recommendations.
- Aloe Vera Juice: Some studies suggest it may normalize blood sugar levels. Drink 1-2 ounces daily, but consult a doctor first.

7. Allergy Management:

- Maintain an up-to-date allergy action plan with your healthcare provider, including emergency medications and allergen avoidance strategies.
- Carry epinephrine auto-injectors (e.g., EpiPen) for severe allergic reactions and ensure easy access in case of an emergency.

8. Regular Follow-ups:

- Schedule appointments with your

14.VIDEO LINK:

<https://drive.google.com/file/d/1tfZugVNZsBl8KKwD-aEexnMUk5M2wpYl/view?usp=sharing>

THANK YOU

