* **REPO PROJECT RT**

**HealthAI: Intelligent Healthcare Assistant Using IBM Granite Companion (React)**

YEAR : 2024 – 2025

COLLEGE NAME : K.C.S KASI NADAR COLLEGE OF ARTS & SCIENCE

CODE : UNM203

DEPARTMENT : COMPUTER SCIENCE

PROGRAM : B.C.A

SEMESTER V

PROJECT SUBMITTED TO: UNIVERSITY OF MADRAS / NAAN MUDALVAN

Course Name : GEN AI WITH IBM

TEAM LEADER: Chenchu Gayathri C

MEMBERS:

1. CHENCHU GAYATHRI C

2. MANJU SHREE G

3. MADHUMITHA S

4. VEDHAPRIYA B

**HealthAI: Intelligent Healthcare Assistant Using IBM Granite Companion (React)**

**Introduction**

Health is a dynamic state of complete physical, mental, and social well-being, not merely the absence of disease or infirmity, as defined by the World Health Organization. It is a multidimensional concept influenced by biological, psychological, environmental, and sociocultural factors, encompassing one's ability to live a fulfilling and productive life by realizing their potential and coping with life's challenges.

The WHO Definition of Health

A Holistic State: Health is viewed as a positive concept involving the integration of three key dimensions.

More Than Absence of Disease: It's not just about not being sick, but about thriving in all aspects of life.

The Three Dimensions of Health

Physical Health: Involves the proper functioning of the body's anatomical structure and physiological systems, allowing all parts to work harmoniously.

Mental Health: Encompasses the ability to learn, think clearly, handle daily life events, work productively, and manage stress.

Social Health: Relates to one's ability to interact positively with others, form relationships, and function effectively within their community and society.

**Sample Coding for Health AI:**

Import gradio as gr

Import torch

From transformers import AutoTokenizer, AutoModelForCausalLM

# Load model and tokenizer

Model\_name = “ibm-granite/granite-3.2-2b-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=512)

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,

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 conditions and general medication suggestions. Always emphasize the importance of consulting a doctor for proper diagnosis.\n\nSymptoms: {symptoms}\n\nPossible conditions and recommendations:\n\n\*\*IMPORTANT: This is for informational purposes only. Please consult a healthcare professional for proper diagnosis and treatment.\*\*\n\nAnalysis:”

Return generate\_response(prompt, max\_length=1200)

Def treatment\_plan(condition, age, gender, medical\_history):

Prompt = f”Generate personalized treatment suggestions for the following patient information. Include home remedies and general medication guidelines.\n\nMedical Condition: {condition}\nAge: {age}\nGender: {gender}\nMedical History: {medical\_history}\n\nPersonalized treatment plan including home remedies and medication guidelines:\n\n\*\*IMPORTANT: This is for informational purposes only. Please consult a healthcare professional for proper treatment.\*\*\n\nTreatment Plan:”

Return generate\_response(prompt, max\_length=1200)

# 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.Tabs():

With gr.TabItem(“Disease Prediction”):

With gr.Row():

With gr.Column():

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=30)

Gender\_input = gr.Dropdown(

Choices=[“Male”, “Female”, “Other”],

Label=”Gender”,

Value=”Male”

)

History\_input = gr.Textbox(

Label=”Medical History”,

Placeholder=”Previous conditions, allergies, medications or None”,

Lines=3

)

Plan\_btn = gr.Button(“Generate Treatment Plan”)

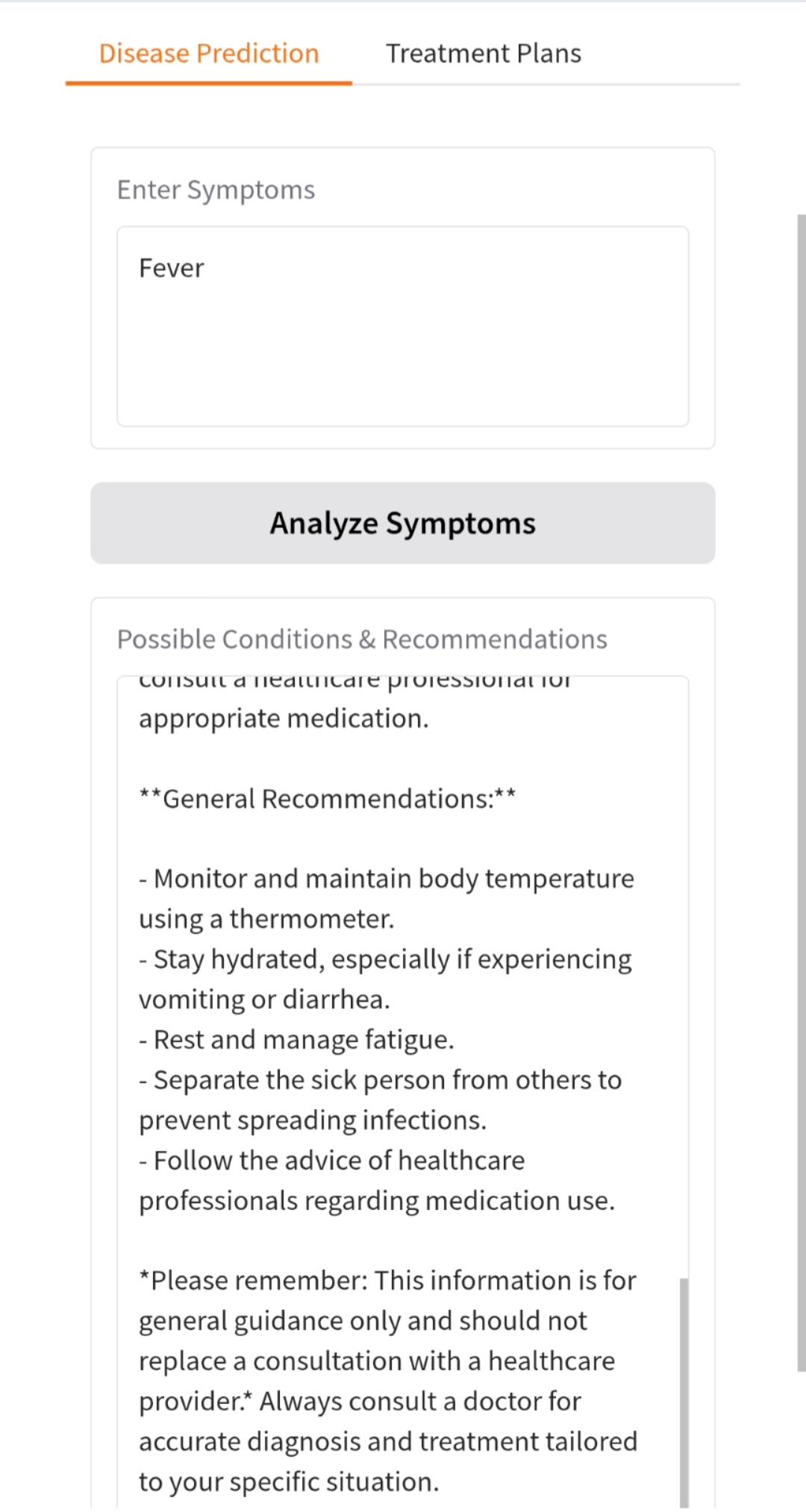
With gr.Column():

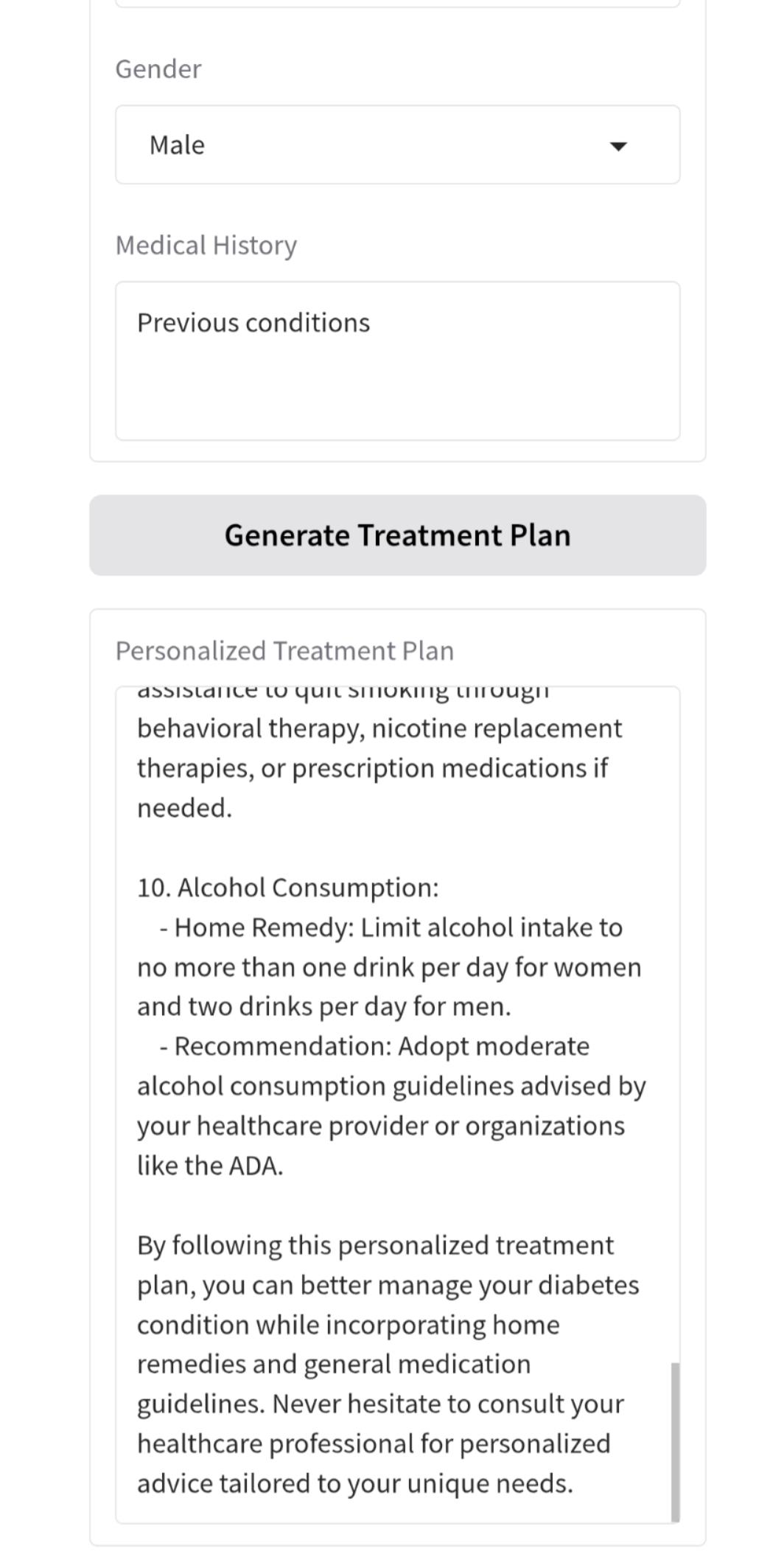
Plan\_output = gr.Textbox(label=”Personalized Treatment Plan”, lines=20)

Plan\_btn.click(treatment\_plan, inputs=[condition\_input, age\_input, gender\_input, history\_input], outputs=plan\_output)

App.launch(share=True)

**Project Execution**

****

****