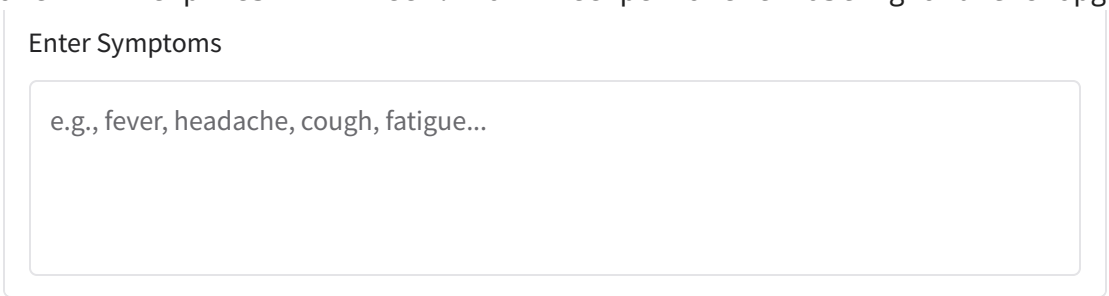
```
import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# 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 gen
    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 inform
    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 heal

    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 & Recommendat

            predict_btn.click(disease_prediction, inputs=symptoms_input, outputs=prediction

        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

app.launch(share=True)
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserW
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access publ
  warnings.warn(
tokenizer_config.json:        8.88k/?  [00:00<00:00,  599kB/s]
vocab.json:          777k/?  [00:00<00:00,  9.96MB/s]
merges.txt:          442k/?  [00:00<00:00,  15.9MB/s]
tokenizer.json:         3.48M/?  [00:00<00:00,  57.7MB/s]
added_tokens.json:  100%                                  87.0/87.0  [00:00<00:00,  7.02kB/s]
special_tokens_map.json:  100%                            701/701  [00:00<00:00,  80.7kB/s]
config.json:  100%                                        786/786  [00:00<00:00,  53.1kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json:        29.8k/?  [00:00<00:00,  2.23MB/s]
Fetching  2  files:  100%                                 2/2  [01:23<00:00,  83.02s/it]
model-00001-of-                                           5.00G/5.00G  [01:22<00:00,  97.1MB/s]
00002.safetensors:  100%
model-00002-of-                                           67.1M/67.1M  [00:01<00:00,  63.6MB/s]
00002.safetensors:  100%
Loading  checkpoint  shards:  100%                        2/2  [00:19<00:00,    8.11s/it]
generation_config.json:  100%                             137/137  [00:00<00:00,  5.06kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in lau
* Running on public URL: https://e0ab5584c002d7088c.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades,
```



Analyze Symptoms