```
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"  # Automatically places model on GPU/CPU
)

# Function to get response
def ask_question(prompt):
    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    outputs = model.generate(
        **inputs,
        max_new_tokens=200,
        temperature=0.7,
        top_p=0.9,
```

```
        do_sample=True
    )
    return tokenizer.decode(outputs[0], skip_special_tokens=True)

# Gradio UI
demo = gr.Interface(
    fn=ask_question,
    inputs=gr.Textbox(lines=3, placeholder="Ask Edu Tutor AI..."),
    outputs="text",
    title="Edu Tutor AI",
    description="Personalized Learning with Generative AI + LMS Integration"
)

demo.launch()
```