

```

ort gradio as gr
ort torch
n transformers import AutoTokenizer, AutoModelForCausalLM

oad model and tokenizer
el_name = "ibm-granite/granite-3.2-2b-instruct"
enizer = AutoTokenizer.from_pretrained(model_name)
el = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.f
    device_map="auto" if torch.cuda.is_available() else None

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

generate_response(prompt, max_length=512):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, ma

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

concept_explanation(concept):
    prompt = f"Explain the concept of {concept} in detail with examples
    return generate_response(prompt, max_length=800)

quiz_generator(concept):
    prompt = f"Generate 5 quiz questions about {concept} with different
    return generate_response(prompt, max_length=1000)

reate Gradio interface
h gr.Blocks() as app:
    gr.Markdown("# Educational AI Assistant")

    with gr.Tabs():
        with gr.TabItem("Concept Explanation"):

```

```

concept_input = gr.Textbox(label="Enter a concept", placeholder="Concept")
explain_btn = gr.Button("Explain")
explanation_output = gr.Textbox(label="Explanation", lines=10)

explain_btn.click(concept_explanation, inputs=concept_input, outputs=explanation_output)

with gr.TabItem("Quiz Generator"):
    quiz_input = gr.Textbox(label="Enter a topic", placeholder="Topic")
    quiz_btn = gr.Button("Generate Quiz")
    quiz_output = gr.Textbox(label="Quiz Questions", lines=15)

    quiz_btn.click(quiz_generator, inputs=quiz_input, outputs=quiz_output)

app.launch(share=True)

```

Loading checkpoint shards: 100%

2/2 [00:11<00:00, 11.79s/it]

Colab notebook detected. To show errors in colab notebook, set debug=True in `gradio.launch(debug=True)`.  
 \* Running on public URL: <https://13664142e826d00d2a.gradio.live>

This share link expires in 1 week. For free permanent hosting and GPU upgrade, see [Gradio Cloud](#).

## Explain

### Explanation

Explain Gen AI, also known as Explainable AI (XAI), is a subfield of artificial intelligence that focuses on developing AI models and systems whose actions can be understood by human experts. The primary goal of Explain Gen AI is to create AI that is transparent, interpretable, and understandable, thereby fostering trust and facilitating better decision-making processes.

The importance of Explain Gen AI lies in addressing the "black box" problem prevalent in many contemporary AI systems. These "black boxes" refer to complex models with intricate architectures, such as deep neural networks, which are powerful but lack interpretability. As a result, human users often struggle to comprehend how these models derive predictions or decisions, leading to concerns about accountability, fairness, and reliability.

To illustrate the concept, consider a credit-scoring AI model:

1. Traditional AI model: A deep neural network (DNN) with approximately 10 layers, utilized for predicting the creditworthiness of loan applicants based on various features like income, credit history, and employment duration.

