JARVIS AI Assistant (Flan■T5)
Inline Fill■in■the■Blanks Workshop Code

Complete the missing parts directly inside the code below. All blanks are marked with _____.

```
# STEP 1: Install packages (uncomment if needed)
# !pip install transformers gradio torch accelerate _____ -q


# STEP 2: Import libraries
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, _____
import _____ as gr
import _____
from gtts import _____
import _____


# STEP 3: Load a SMARTER conversational AI model
print("Loading your AI assistant... (this might take a minute)")

model_name = "google/_________________"
tokenizer = AutoTokenizer._________________(model_name)
model = AutoModelForSeq2SeqLM._________________(model_name)

print("Assistant loaded! Ready to chat.")


# STEP 4: Create the chat function with better prompting
def chat_with_assistant(user_input, history):

    context = "You are ___________, an intelligent and helpful AI assistant. Be conversational and friendly

    if history:
        for human, assistant in history[-___:]:
            context += f"_________: {human}\n_________: {assistant}\n"

    context += f"_________: {user_input}\n_________:"

    inputs = tokenizer(context, return_tensors="pt", truncation=True, max_length=_____)
    outputs = model.generate(
        **inputs,
        max_length=_____,
        num_beams=_____,
        temperature=_____,
        do_sample=True,
        top_p=_____
    )

    response = tokenizer._________________(outputs[0], skip_special_tokens=True)
    return response


# STEP 5: Create Gradio interface with voice
def gradio_chat_with_voice(message, history):
    response = chat_with_assistant(message, history)

    try:
        tts = gTTS(text=response, lang="__", slow=False)
        audio_file = "_____________"
        tts.save(audio_file)
        return response, audio_file
```

```
    except Exception as e:
        print(f"TTS Error: {e}")
        return response, None


# STEP 6: Create beautiful Gradio interface
with gr._____(theme=gr.themes._____()) as demo:
    gr.Markdown("""
    # ■ Your Personal JARVIS Assistant
    ### Built in Google Colab with AI superpowers!
    Ask me anything - I'll respond with text and voice.
    """)

    with gr.Row():
        with gr.Column(scale=2):
            chatbot = gr.Chatbot(height=500, bubble_full_width=False)

            with gr.Row():
                msg = gr.Textbox(
                    label="Your message",
                    placeholder="Ask me anything...",
                    scale=4
                )
                submit = gr.Button("Send ■", scale=1, variant="primary")

            clear = gr.Button("Clear Chat ■■")

        with gr.Column(scale=1):
            audio_output = gr.Audio(label="■ Voice Response", autoplay=True)

            gr.Markdown("### Quick Examples:")
            example_btns = [
                gr.Button("■ Introduce yourself"),
                gr.Button("■ Tell me a joke"),
                gr.Button("■ Explain AI simply"),
                gr.Button("■ Give me a fun fact"),
            ]


    def respond(message, chat_history):
        if not message.strip():
            return "", chat_history, None

        bot_response, audio_file = gradio_chat_with_voice(message, chat_history)
        chat_history.append((message, bot_response))
        return "", chat_history, audio_file


    msg._____(respond, [msg, chatbot], [msg, chatbot, audio_output])
    submit._____(respond, [msg, chatbot], [msg, chatbot, audio_output])
    clear.click(lambda: [], None, chatbot, queue=False)


print("■ Launching your AI assistant...")
demo.launch(share=_____, debug=_____)
```