```
!pip install transformers
!pip install streamlit
!pip install torch
!
```

```
Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (24.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 26.5 MB/s eta 0:00:00
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (883 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 44.7 MB/s eta 0:00:00
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl (664.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.8 MB/s eta 0:00:00
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl (211.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 5.8 MB/s eta 0:00:00
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl (56.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 12.1 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl (127.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 7.2 MB/s eta 0:00:00
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl (207.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 6.0 MB/s eta 0:00:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 57.2 MB/s eta 0:00:00
Installing collected packages: nvidia-nvjitlink-cu12, nvidia-curand-cu12, nvidia-cufft-c
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidi
```

```
import os
import torch
import streamlit as st
from transformers import AutoModelForCausalLM, AutoTokenizer

try:
    import transformers
except ImportError:
    os.system("pip install transformers")
    import transformers

# Select a model (Modify this based on resources)
AVAILABLE_MODELS = {
    "gpt2": "gpt2",
    "gpt-neo": "EleutherAI/gpt-neo-1.3B",
    "opt": "facebook/opt-1.3b"
}

MODEL_NAME = AVAILABLE_MODELS["gpt2"]  # Change this key to load a different model

# Load tokenizer and model
print(f"Loading model: {MODEL_NAME}...")
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
model = AutoModelForCausalLM.from_pretrained(MODEL_NAME)
```

```
Loading model: gpt2...
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
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 (https://
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public model
  warnings.warn(
tokenizer_config.json: 100%                 26.0/26.0 [00:00<00:00, 1.76kB/s]
config.json: 100%                           665/665 [00:00<00:00, 30.0kB/s]
vocab.json: 100%                            1.04M/1.04M [00:00<00:00, 7.30MB/s]
merges.txt: 100%                            456k/456k [00:00<00:00, 914kB/s]
tokenizer.json: 100%                        1.36M/1.36M [00:00<00:00, 10.0MB/s]
model.safetensors: 100%                     548M/548M [00:03<00:00, 143MB/s]
generation_config.json: 100%                124/124 [00:00<00:00, 4.98kB/s]
```

```
def generate_text(prompt, max_length=50, temperature=0.7, top_p=0.9):
    inputs = tokenizer(prompt, return_tensors="pt")
    outputs = model.generate(
        input_ids=inputs["input_ids"],  # Explicitly set input_ids
        attention_mask=inputs["attention_mask"],  # Add attention mask for better output
        max_length=max_length,
        temperature=temperature,
        top_p=top_p,
        do_sample=True  # Enables sampling for more varied responses
    )
    return tokenizer.decode(outputs[0], skip_special_tokens=True)
```

```
# Chatbot Interface
def chatbot():
    """
    Implements a simple chatbot loop that interacts with the user.
    - Type 'exit' to end the conversation.
    """
    print("Chatbot is ready! Type 'exit' to stop.")
    while True:
        user_input = input("You: ")
        if user_input.strip() == "":
            print("Chatbot: Please enter a valid input.")
            continue
        if user_input.lower() == "exit":
            print("Chatbot: Goodbye!")
            break

        try:
            response = generate_text(user_input, temperature=0.8, max_length=100)
        except Exception as e:
            response = f"Error processing input: {e}"

        print(f"Chatbot: {response}")

if __name__ == "__main__":
    print("Setup complete. Model and tokenizer are ready.")
    print(f"Loaded model: {MODEL_NAME}")
    chatbot()
```

```
Setup complete. Model and tokenizer are ready.
Loaded model: gpt2
Chatbot is ready! Type 'exit' to stop.
You: hello
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Chatbot: hello for the most part.

But, of course, there are some things that we can't control. For example, we have to do

We have to understand
You: exit
Chatbot: Goodbye!
```

```
# Testing and Iteration
def test_chatbot():
    """
    Runs predefined test cases to evaluate chatbot performance.
    """
    test_cases = [
        "Hello!",
        "Tell me a joke.",
        "What is AI?",
        "Can you write a poem?",
        "Who is the president of the United States?",
        "Give me a random fact!",
        "exit"
    ]

    print("\nRunning chatbot test cases...")
    for test in test_cases:
        print(f"You: {test}")
        response = generate_text(test, temperature=0.8, max_length=100)
        print(f"Chatbot: {response}\n")
    print("Testing complete.")
```

```
if __name__ == "__main__":
    print("Setup complete. Model and tokenizer are ready.")
    print(f"Loaded model: {MODEL_NAME}")

    # Run chatbot
    chatbot()

    # Run test cases after chatbot session
    test_chatbot()
```

```
Setup complete. Model and tokenizer are ready.
Loaded model: gpt2
Chatbot is ready! Type 'exit' to stop.
You: who is the president of nigeria
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Chatbot: who is the president of nigeria!

The question is whether I'm doing this with his permission or if he's doing it to himsel

I'm not going to be talking about his own political views or his personal views. I'm goi

I'm not going to be talking about his actions. I'm going to be talking about his choices

I'm going to be talking about his thoughts, his choices, his thoughts.
You: exit
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Chatbot: Goodbye!

Running chatbot test cases...
You: Hello!
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Chatbot: Hello! (Sigh) I'm sure you'll be able to do a good job in my house. I'm sure yo

You: Tell me a joke.
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Chatbot: Tell me a joke. You can't get me to do it. Why do you think you are doing this?

"Because I have no idea what it means to be a woman. I have no idea what it means to be

"You have no idea what it means to be a woman. It is the only way. It is not a privilege

You: What is AI?
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Chatbot: What is AI?

AI is the ability to predict and control what you do with your brain. This is a way of t

There are two kinds of computers: computers that are smart, and computers that are not.

In AI, we're talking about the kind of computers that are smart because they don't know



You: Can you write a poem?
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Chatbot: Can you write a poem?

Yes! I can write a poem if I want, but I don't want to write it as a story. I want to wr



You: Who is the president of the United States?
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Chatbot: Who is the president of the United States? Is it because he is a member of the
```

```
That's not how it works. You know, he's not a member of the Democratic Party, you know,
```

```
import streamlit as st

st.title("Chatbot")
user_input = st.text_input("You: ")
if st.button("Send"):
    response = generate_text(user_input)
    st.write(f"Chatbot: {response}")
```

```
2025-03-21 13:59:13.752 WARNING streamlit.runtime.scriptrunner_utils.script_run_context:
2025-03-21 13:59:14.448
  Warning: to view this Streamlit app on a browser, run it with the following
  command:

    streamlit run /usr/local/lib/python3.11/dist-packages/colab_kernel_launcher.py [ARGU
2025-03-21 13:59:14.452 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.455 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.457 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.460 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.462 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.464 Session state does not function when running a script without `s
2025-03-21 13:59:14.467 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.469 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.472 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.474 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.476 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.478 Thread 'MainThread': missing ScriptRunContext! This warning can
2025-03-21 13:59:14.479 Thread 'MainThread': missing ScriptRunContext! This warning can
```

```
!pip install -q streamlit
```

```
!npm install localtunnel
```

```
added 22 packages in 4s

3 packages are looking for funding
  run `npm fund` for details
```

```
!streamlit run app.py &>/content/logs.txt &
```

```
!npx localtunnel --port 8501
```

```
your url is: https://slick-terms-kick.loca.lt
```