Session 5 — Text Playground

Space Builder: Text Playground

What You'll Build

A text generation Space with three sliders that control how the AI writes. Temperature controls creativity, Top-p controls word diversity, and Max Length controls how much it writes.

Step 1 — Create a New Space

1. Go to huggingface.co/new-space (log in if needed)
2. In the "Space name" field, type: text-playground
3. Under "Select the Space SDK", choose Gradio
4. Under "Select the Space hardware", choose Free — CPU basic
5. Leave everything else as default
6. Click Create Space

Step 2 — Create the requirements.txt File

1. On your Space page, click the Files tab (near the top)
2. Click Add file → Create a new file
3. In the filename field at the top, type: requirements.txt
4. In the big text area below, copy and paste this exactly:

Your Space page will open. It's empty right now — we're about to add the code.

5. Click Commit new file to main (the blue button at the bottom)

Step 3 — Create the app.py File

1. Click Add file → Create a new file (again)
2. In the filename field, type: app.py
3. In the big text area, copy and paste ALL of the code below:

```
transformers
torch
gradio
```

4. Click Commit new file to main

Step 4 — Wait for It to Build

1. Click the App tab (at the top of your Space page)

2. You’ll see "Building" — this takes 2–5 minutes

3. When it’s done, your Space will appear!

Step 5 — Try It Out!

1. Set temperature to 0.1 and generate — then set it to 2.0 with the same prompt. See the difference?

2. Try the same prompt 3 times with the same settings — you get different outputs each time!

3. Set Max Length to 20 vs 200 — how does length affect quality?

Troubleshooting

```
from transformers import pipeline
import gradio as gr

# Load a small text generation model (fast on free CPU)
print("Loading distilgpt2...")
generator = pipeline("text-generation", model="distilgpt2")
print("Model loaded!")

def generate_text(prompt, temperature, top_p, max_length):
    if not prompt or not prompt.strip():
        return "Type a prompt above first!"

    result = generator(
        prompt,
        temperature=max(temperature, 0.01),  # avoid division by zero
        top_p=top_p,
        max_length=int(max_length),
        do_sample=True,
        num_return_sequences=1,
    )
    return result[0]["generated_text"]

demo = gr.Interface(
    fn=generate_text,
    inputs=[
        gr.Textbox(
            lines=3,
            placeholder="Start your text here...",
            label="Prompt",
        ),
        gr.Slider(
            minimum=0.1,
            maximum=2.0,
            value=0.7,
            step=0.1,
            label="Temperature (creativity)",
        ),
        gr.Slider(
            minimum=0.1,
            maximum=1.0,
            value=0.9,
            step=0.05,
            label="Top-p (diversity)",
        ),
        gr.Slider(
            minimum=20,
            maximum=200,
            value=100,
            step=10,
            label="Max Length (words-ish)",
        ),
    ],
    outputs=gr.Textbox(label="Generated Text", lines=10),
    title="Text Playground",
    description="Type a prompt and use the sliders to control how the AI writes. Temperature co
  ntrols creativity (low = predictable, high = wild). Top-p controls word diversity. Max length
   controls how much it writes.",
    examples=[
        ["Once upon a time in a school where robots", 0.7, 0.9, 100],
        ["The secret ingredient in the recipe was", 1.2, 0.9, 80],
        ["Dear Principal, I am writing to request", 0.3, 0.9, 100],
        ["Breaking news: scientists discover that cats", 0.9, 0.95, 120],
        ["The haunted house at the end of the street", 1.5, 0.8, 150],
    ],
)

demo.launch()
```

```
Make sure you copy the ENTIRE code block — from the very first line to the very last. Missing e
  ven one line can cause errors.
```

```
If you see a red error: Click the Logs tab to read the error message. The most common fix is to
   double-check that requirements.txt has the right contents and that app.py was copied complet
  ely.
```

Problem	Fix
"Runtime error"	Check the Logs tab. Usually means a typo in app.py. Re-copy the code carefully.
Space stuck on "Building"	Wait up to 5 minutes. Free CPU Spaces can be slow. If it’s been more than 10 minutes, try deleting the Space and starting over.
"ModuleNotFoundError"	Your requirements.txt is missing a library. Make sure it matches exactly what’s shown above.
Space loads but nothing happens	Make sure the last line of app.py is demo.launch() with no extra spaces before it.