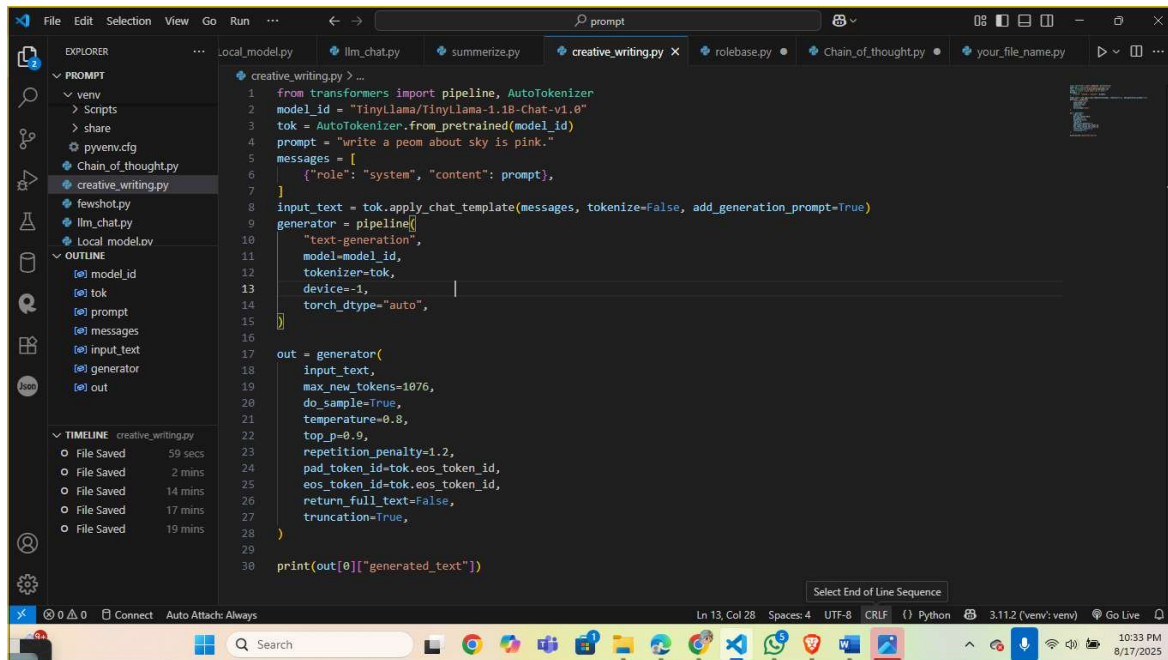


Build Python scripts for text generation and chat, Error handling.

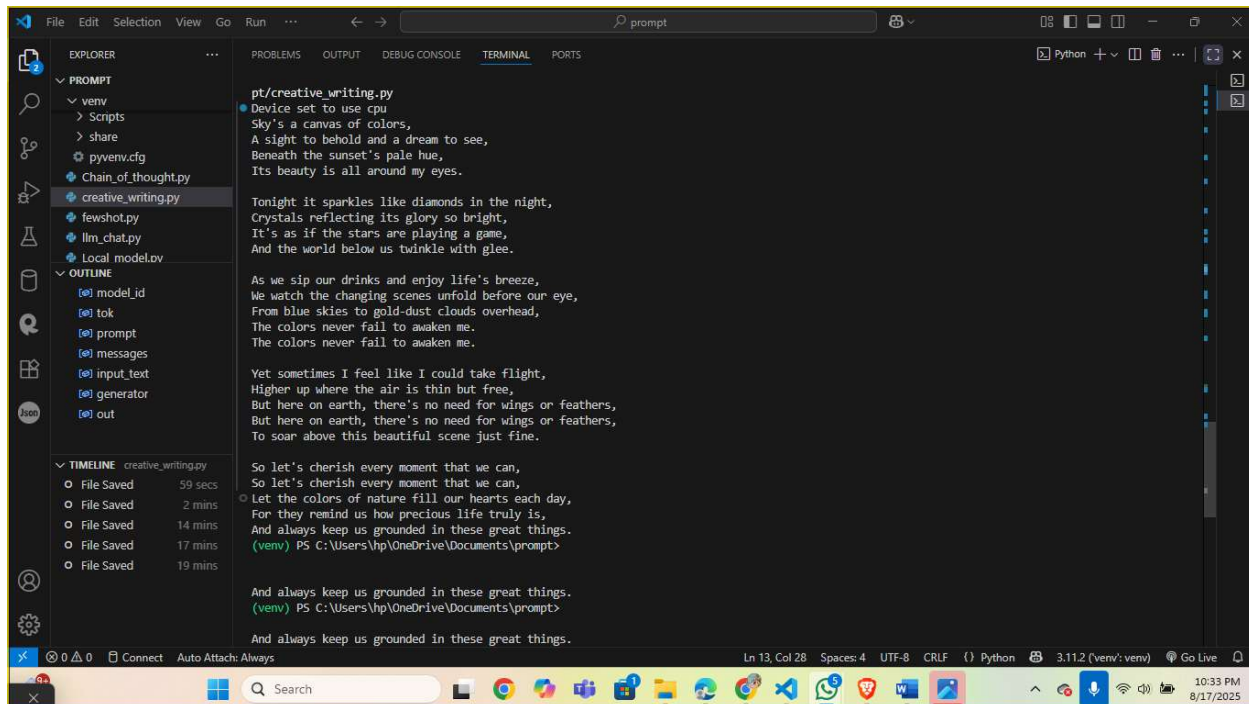
1. Text Generation

Code snippet:



```
1 from transformers import pipeline, AutoTokenizer
2 model_id = "TinyLlama/TinyLlama-1.1B-Chat-v1.0"
3 tok = AutoTokenizer.from_pretrained(model_id)
4 prompt = "write a poem about sky is pink."
5 messages = [
6     {"role": "system", "content": prompt},
7 ]
8 input_text = tok.apply_chat_template(messages, tokenize=False, add_generation_prompt=True)
9 generator = pipeline(
10     "text-generation",
11     model=model_id,
12     tokenizer=tok,
13     device=-1,
14     torch_dtype="auto",
15 )
16
17 out = generator(
18     input_text,
19     max_new_tokens=1076,
20     do_sample=True,
21     temperature=0.8,
22     top_p=0.9,
23     repetition_penalty=1.2,
24     pad_token_id=tok.eos_token_id,
25     eos_token_id=tok.eos_token_id,
26     return_full_text=False,
27     truncation=True,
28 )
29
30 print(out[0]["generated_text"])
```

Output:



```
pt/creative_writing.py
Device set to use cpu
Sky's a canvas of colors,
A sight to behold and a dream to see,
Beneath the sunset's pale hue,
Its beauty is all around my eyes.

Tonight it sparkles like diamonds in the night,
Crystals reflecting its glory so bright,
It's as if the stars are playing a game,
And the world below us twinkle with glee.

As we sip our drinks and enjoy life's breeze,
We watch the changing scenes unfold before our eye,
From blue skies to gold-dust clouds overhead,
The colors never fail to awaken me.
The colors never fail to awaken me.

Yet sometimes I feel like I could take flight,
Higher up where the air is thin but free,
But here on earth, there's no need for wings or feathers,
But here on earth, there's no need for wings or feathers,
To soar above this beautiful scene just fine.

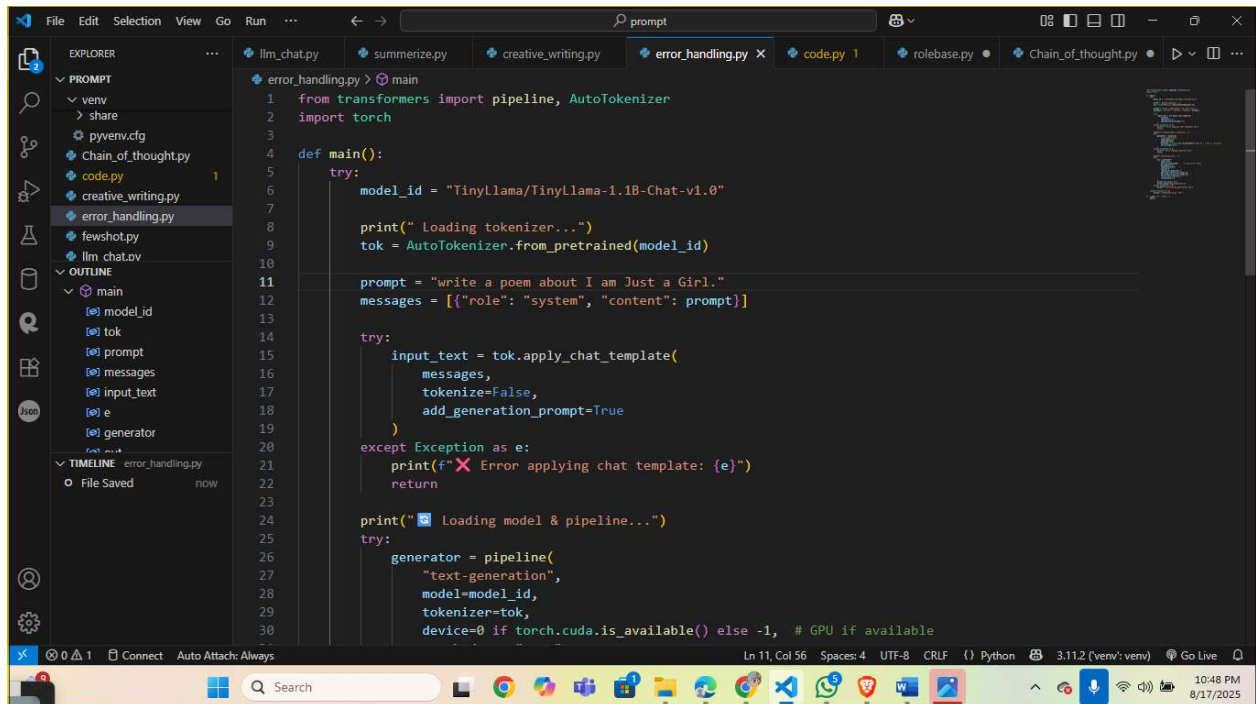
So let's cherish every moment that we can,
So let's cherish every moment that we can,
Let the colors of nature fill our hearts each day,
For they remind us how precious life truly is,
And always keep us grounded in these great things.
(venv) PS C:\Users\hvp\OneDrive\Documents\prompt>

And always keep us grounded in these great things.
(venv) PS C:\Users\hvp\OneDrive\Documents\prompt>

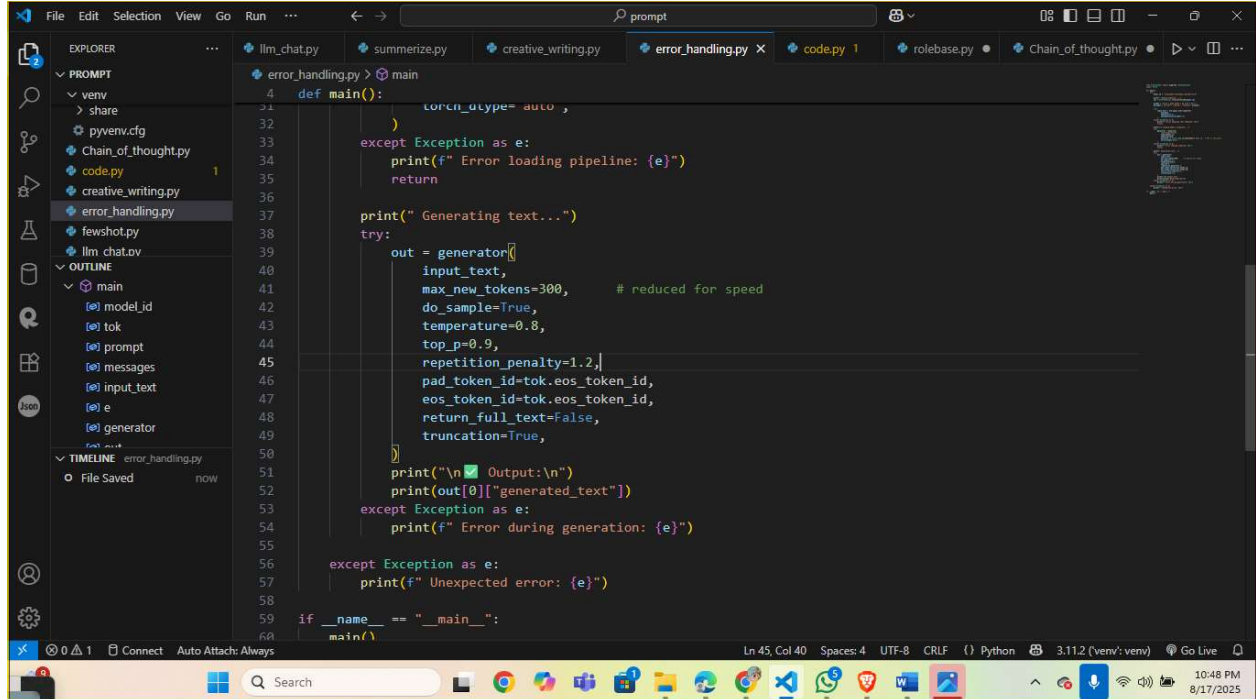
And always keep us grounded in these great things.
```


2. Error handling

Code snippet

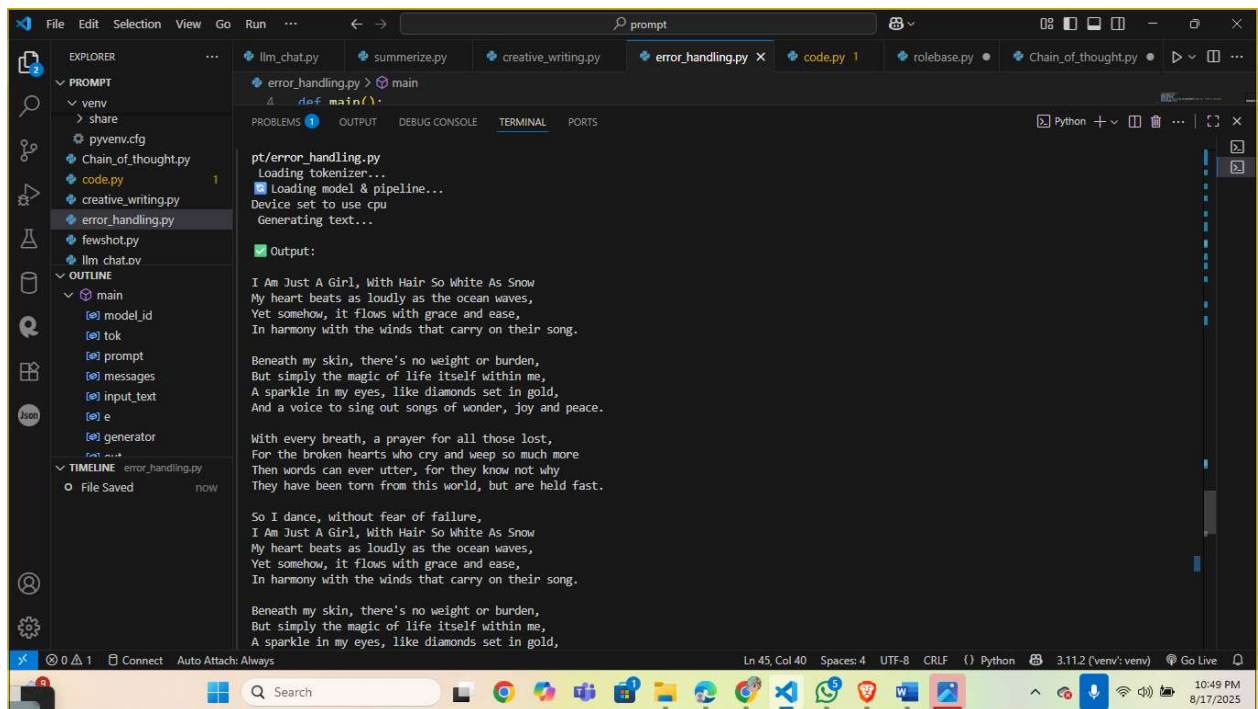


```
1 from transformers import pipeline, AutoTokenizer
2 import torch
3
4 def main():
5     try:
6         model_id = "TinyLlama/TinyLlama-1.1B-Chat-v1.0"
7
8         print(" Loading tokenizer...")
9         tok = AutoTokenizer.from_pretrained(model_id)
10
11         prompt = "write a poem about I am Just a Girl."
12         messages = [{"role": "system", "content": prompt}]
13
14         try:
15             input_text = tok.apply_chat_template(
16                 messages,
17                 tokenize=False,
18                 add_generation_prompt=True
19             )
20
21         except Exception as e:
22             print(f"❌ Error applying chat template: {e}")
23             return
24
25         print("📄 Loading model & pipeline...")
26         try:
27             generator = pipeline(
28                 "text-generation",
29                 model=model_id,
30                 tokenizer=tok,
31                 device=-1 if torch.cuda.is_available() else -1, # GPU if available
```



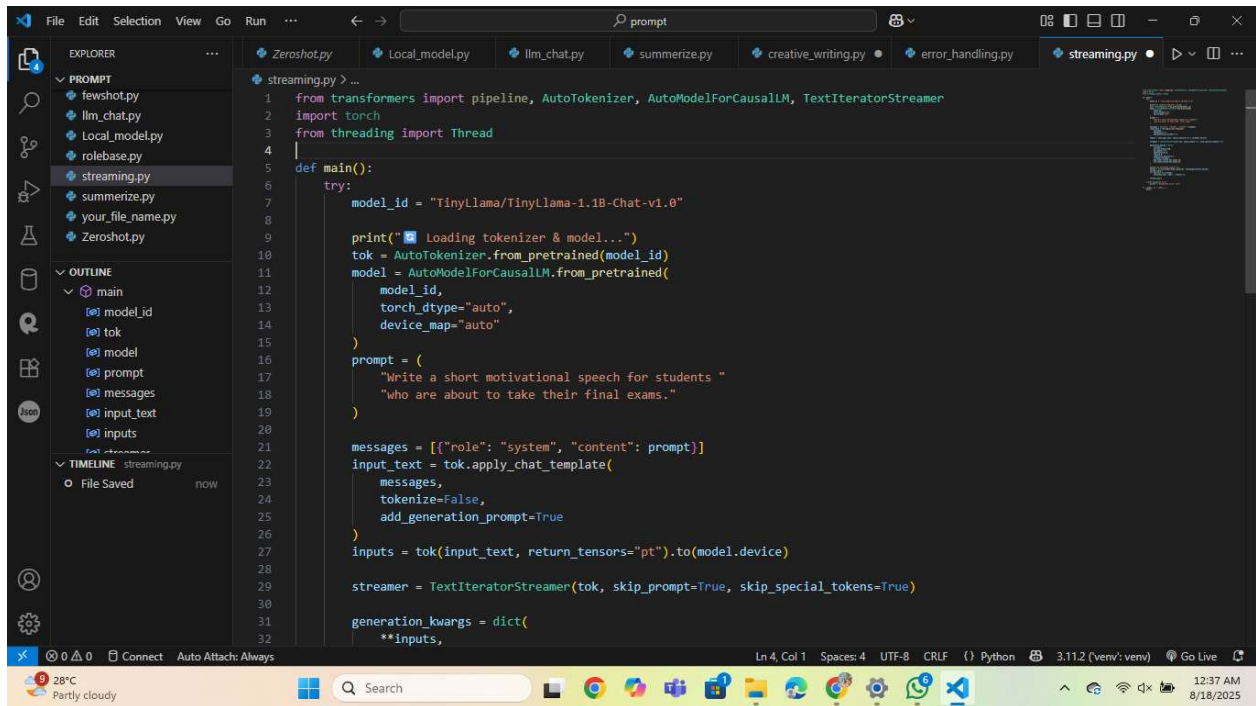
```
32         torch_dtype=torch.float16,
33     )
34 except Exception as e:
35     print(f" Error loading pipeline: {e}")
36     return
37
38 print(" Generating text...")
39 try:
40     out = generator(
41         input_text,
42         max_new_tokens=300, # reduced for speed
43         do_sample=True,
44         temperature=0.8,
45         top_p=0.9,
46         repetition_penalty=1.2,
47         pad_token_id=tok.eos_token_id,
48         eos_token_id=tok.eos_token_id,
49         return_full_text=False,
50         truncation=True,
51     )
52     print("\n✅ Output:\n")
53     print(out[0]["generated_text"])
54 except Exception as e:
55     print(f" Error during generation: {e}")
56
57 except Exception as e:
58     print(f" Unexpected error: {e}")
59
60 if __name__ == "__main__":
61     main()
```

Output:

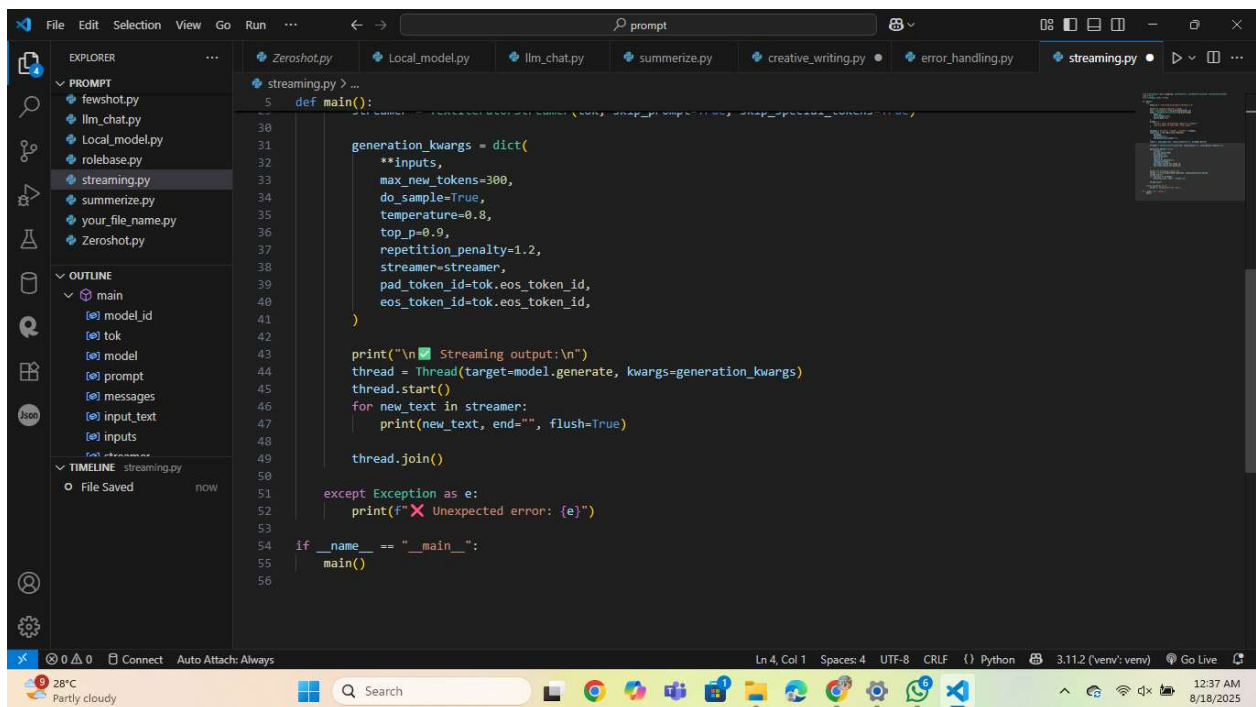


3. Streaming response:

Code

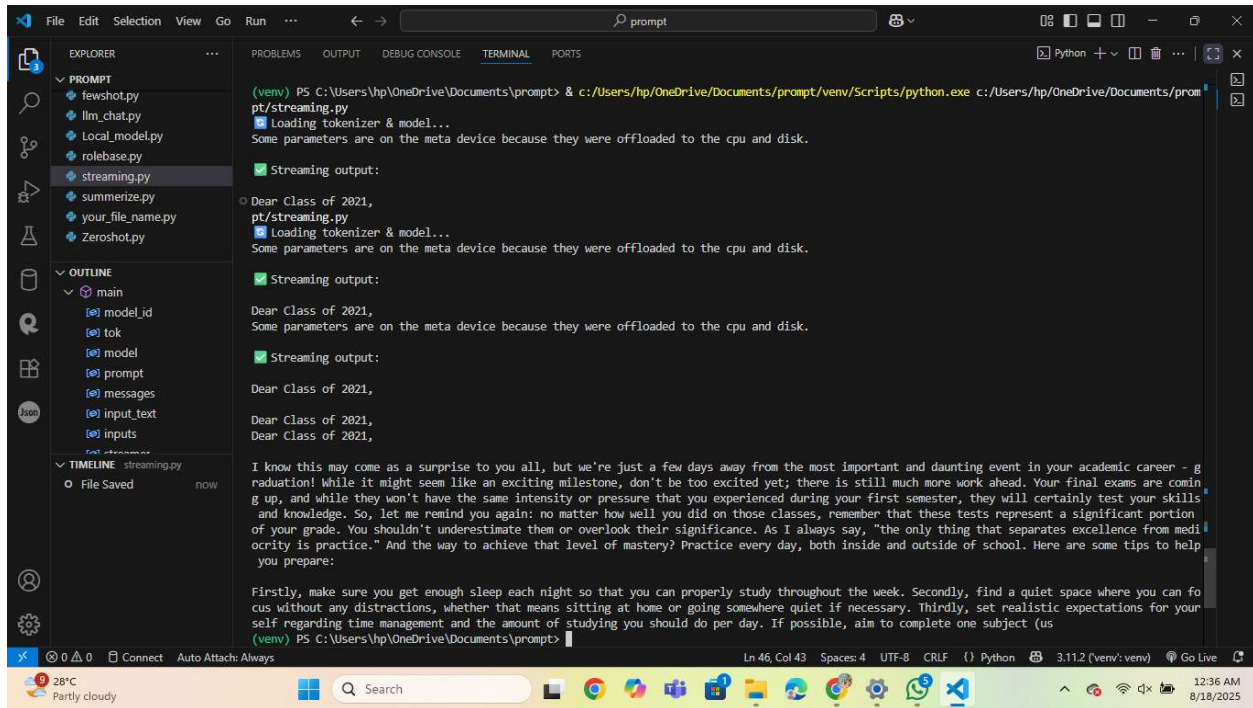


```
1 from transformers import pipeline, AutoTokenizer, AutoModelForCausalLM, TextIteratorStreamer
2 import torch
3 from threading import Thread
4
5 def main():
6     try:
7         model_id = "TinyLlama/TinyLlama-1.1B-Chat-v1.0"
8
9         print(" Loading tokenizer & model...")
10        tok = AutoTokenizer.from_pretrained(model_id)
11        model = AutoModelForCausalLM.from_pretrained(
12            model_id,
13            torch_dtype="auto",
14            device_map="auto"
15        )
16        prompt = (
17            "Write a short motivational speech for students "
18            "who are about to take their final exams."
19        )
20
21        messages = [{"role": "system", "content": prompt}]
22        input_text = tok.apply_chat_template(
23            messages,
24            tokenize=False,
25            add_generation_prompt=True
26        )
27        inputs = tok(input_text, return_tensors="pt").to(model.device)
28
29        streamer = TextIteratorStreamer(tok, skip_prompt=True, skip_special_tokens=True)
30
31        generation_kwargs = dict(
32            **inputs,
```



```
33        generation_kwargs = dict(
34            **inputs,
35            max_new_tokens=300,
36            do_sample=True,
37            temperature=0.8,
38            top_p=0.9,
39            repetition_penalty=1.2,
40            streamer=streamer,
41            pad_token_id=tok.eos_token_id,
42            eos_token_id=tok.eos_token_id,
43        )
44
45        print("\n Streaming output:\n")
46        thread = Thread(target=model.generate, kwargs=generation_kwargs)
47        thread.start()
48        for new_text in streamer:
49            print(new_text, end="", flush=True)
50
51        thread.join()
52
53        except Exception as e:
54            print(f" Unexpected error: {e}")
55
56        if __name__ == "__main__":
57            main()
```


Output:



```
(venv) PS C:\Users\hp\OneDrive\Documents\prompt> & c:\Users\hp\OneDrive\Documents\prompt\venv\Scripts\python.exe c:\Users\hp\OneDrive\Documents\prompt\streaming.py
Loading tokenizer & model...
Some parameters are on the meta device because they were offloaded to the cpu and disk.
[X] Streaming output:
Dear Class of 2021,
pt/streaming.py
Loading tokenizer & model...
Some parameters are on the meta device because they were offloaded to the cpu and disk.
[X] Streaming output:
Dear Class of 2021,
Some parameters are on the meta device because they were offloaded to the cpu and disk.
[X] Streaming output:
Dear Class of 2021,
Dear Class of 2021,
Dear Class of 2021,
Dear Class of 2021,
I know this may come as a surprise to you all, but we're just a few days away from the most important and daunting event in your academic career - graduation! While it might seem like an exciting milestone, don't be too excited yet; there is still much more work ahead. Your final exams are coming up, and while they won't have the same intensity or pressure that you experienced during your first semester, they will certainly test your skills and knowledge. So, let me remind you again: no matter how well you did on those classes, remember that these tests represent a significant portion of your grade. You shouldn't underestimate them or overlook their significance. As I always say, "the only thing that separates excellence from mediocrity is practice." And the way to achieve that level of mastery? Practice every day, both inside and outside of school. Here are some tips to help you prepare:
Firstly, make sure you get enough sleep each night so that you can properly study throughout the week. Secondly, find a quiet space where you can focus without any distractions, whether that means sitting at home or going somewhere quiet if necessary. Thirdly, set realistic expectations for yourself regarding time management and the amount of studying you should do per day. If possible, aim to complete one subject (us
(venv) PS C:\Users\hp\OneDrive\Documents\prompt>
```