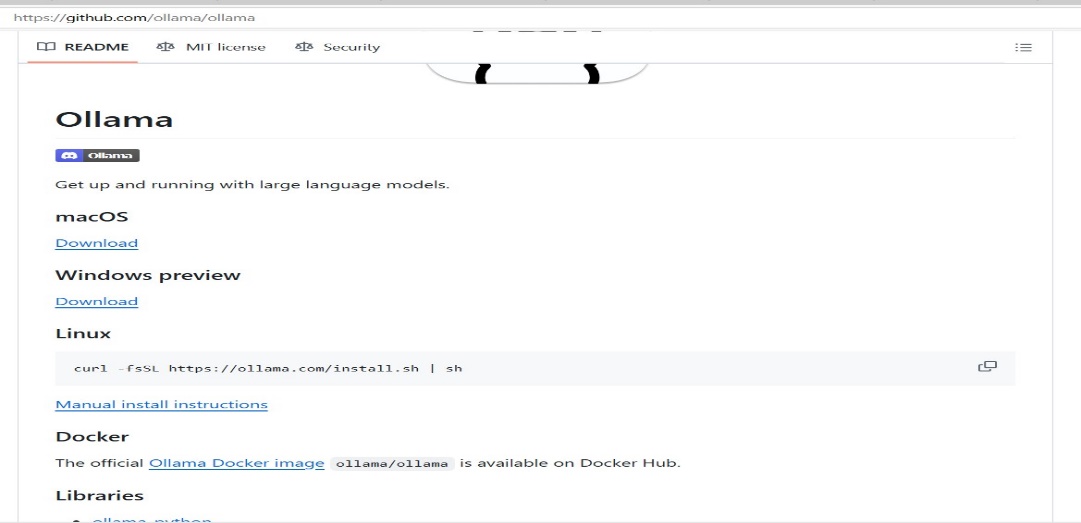
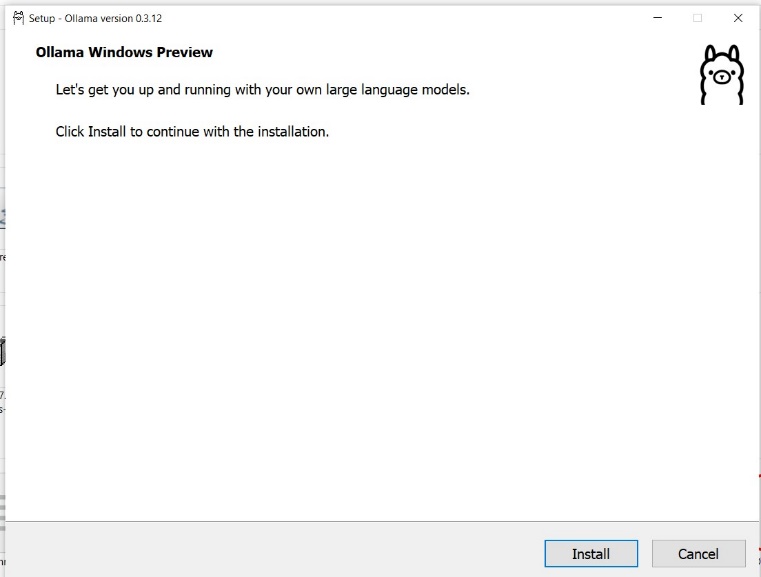
**Part – 1**

**1.1 Local LLM Setup and Interaction:**

**Process:**

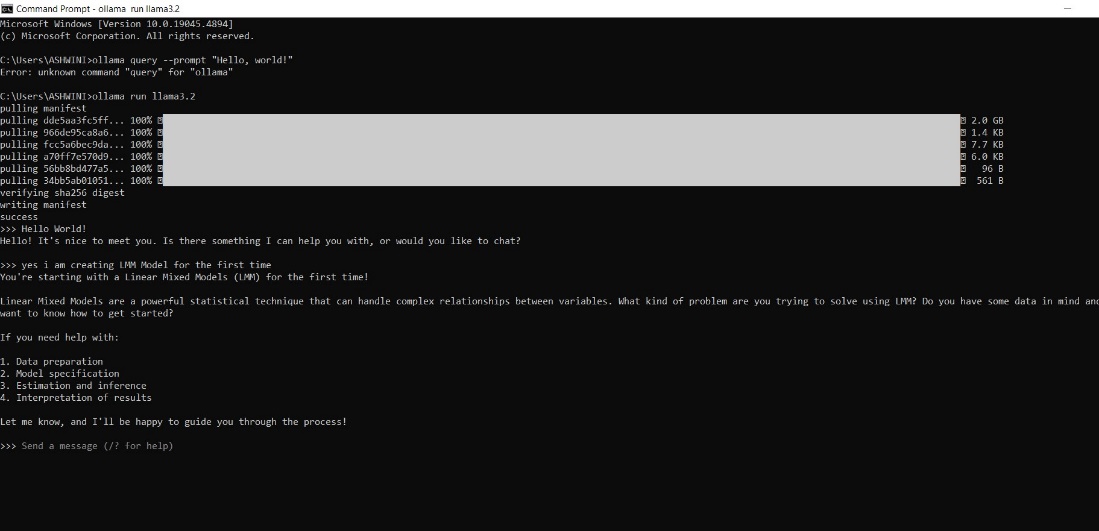


Download the file for Windows:



-Installed the Ollama.

-ollama run llama3.2



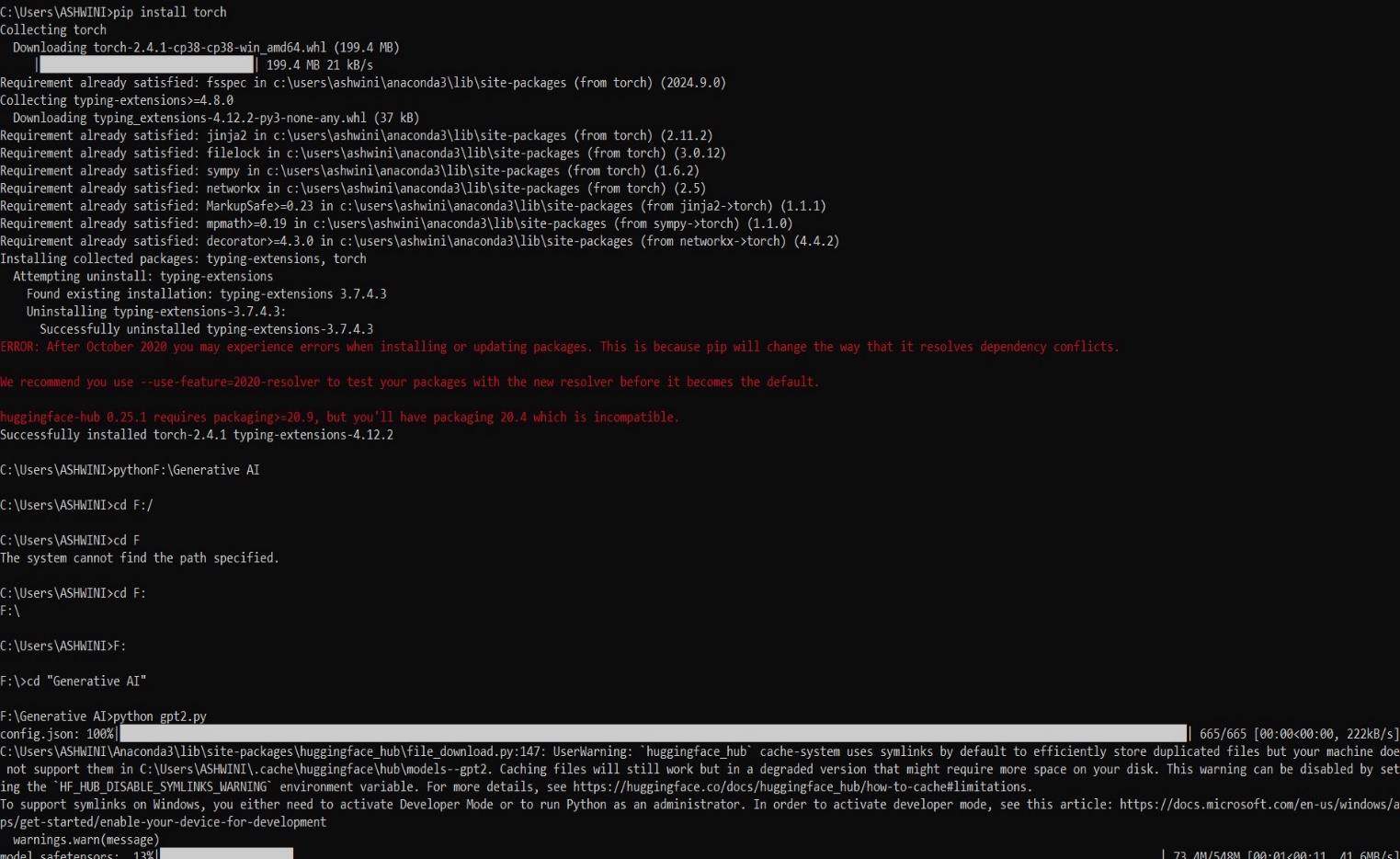
**1.2 Choose and Set Up a Second LLM**

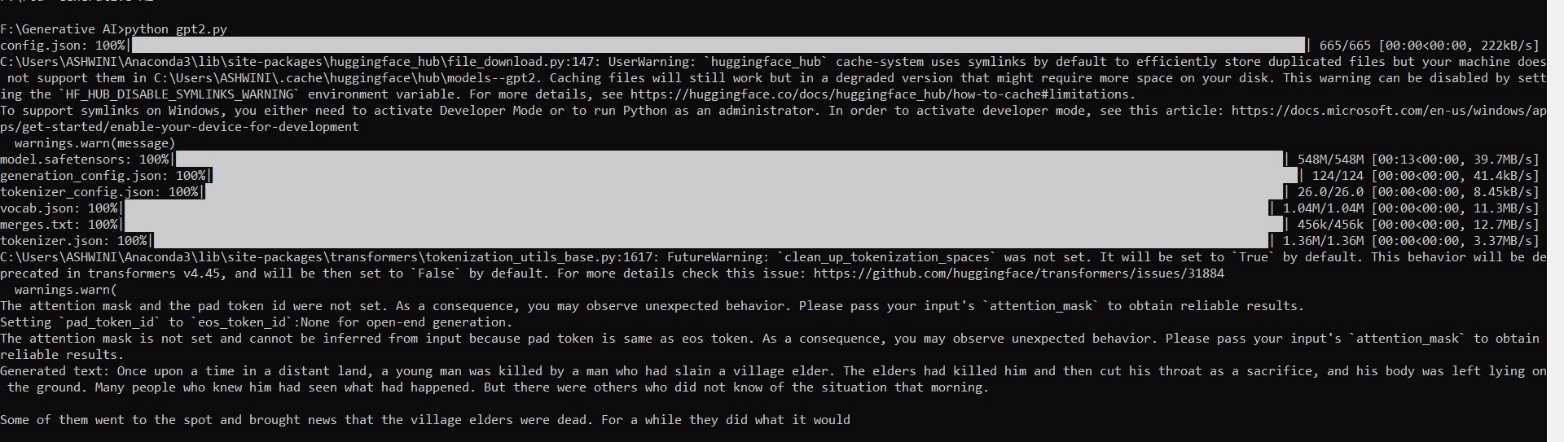
gpt2 Model

pip install transformers

pip install torch

Run the Python File 🡪 python gpt2.py





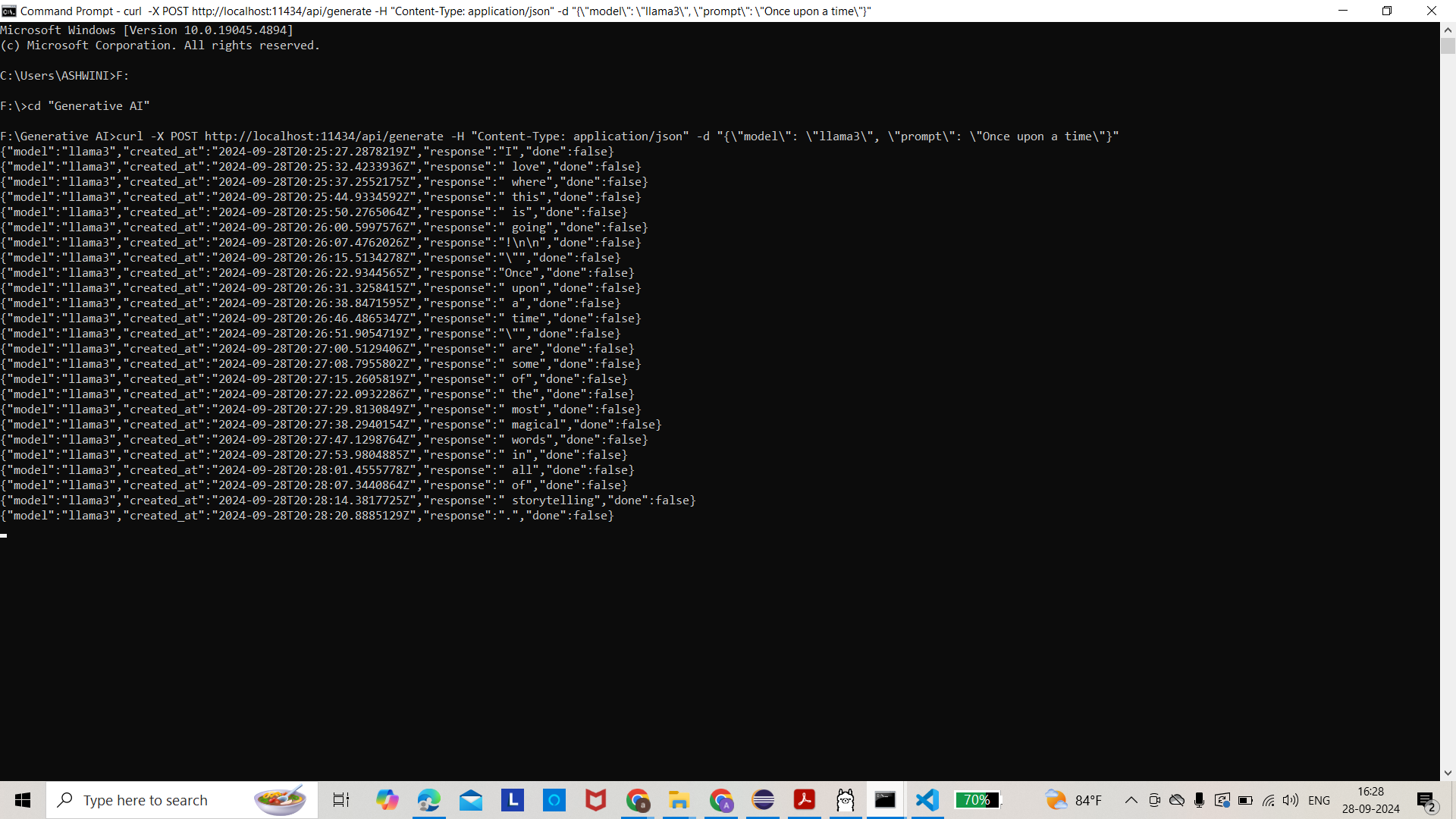
**1.3 Command-Line Interaction Using Curl**

**a. Send a simple query b. Send a complex, multi-turn conversation c. Experiment with different**

**parameters (e.g., temperature, max\_tokens) d. Test the model's performance on specific tasks**

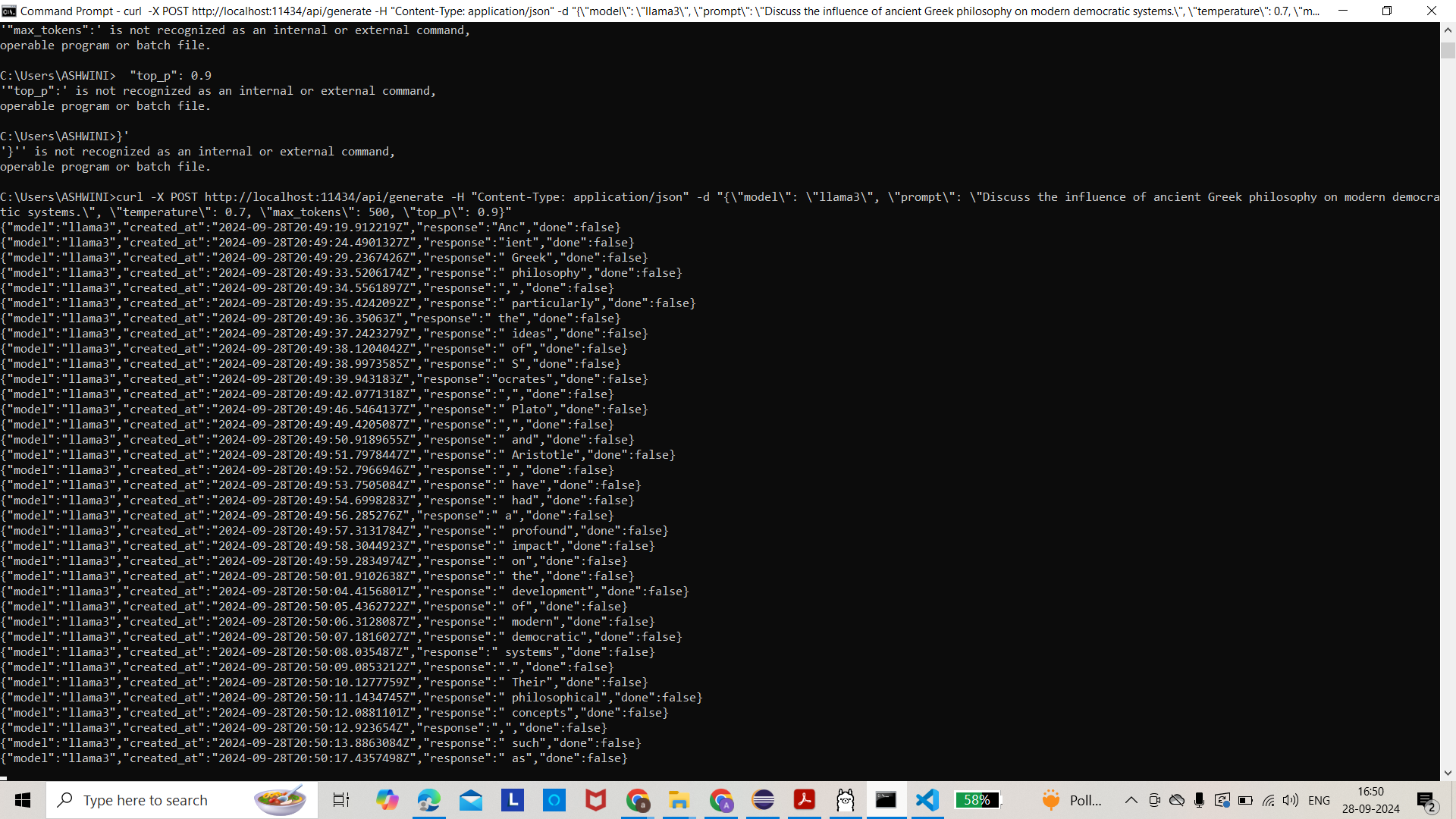
**Simple Query – llama3.2**

curl -X POST http://localhost:11434/api/generate -H "Content-Type: application/json" -d "{\"model\": \"llama3\", \"prompt\": \"One upon a time\"}"



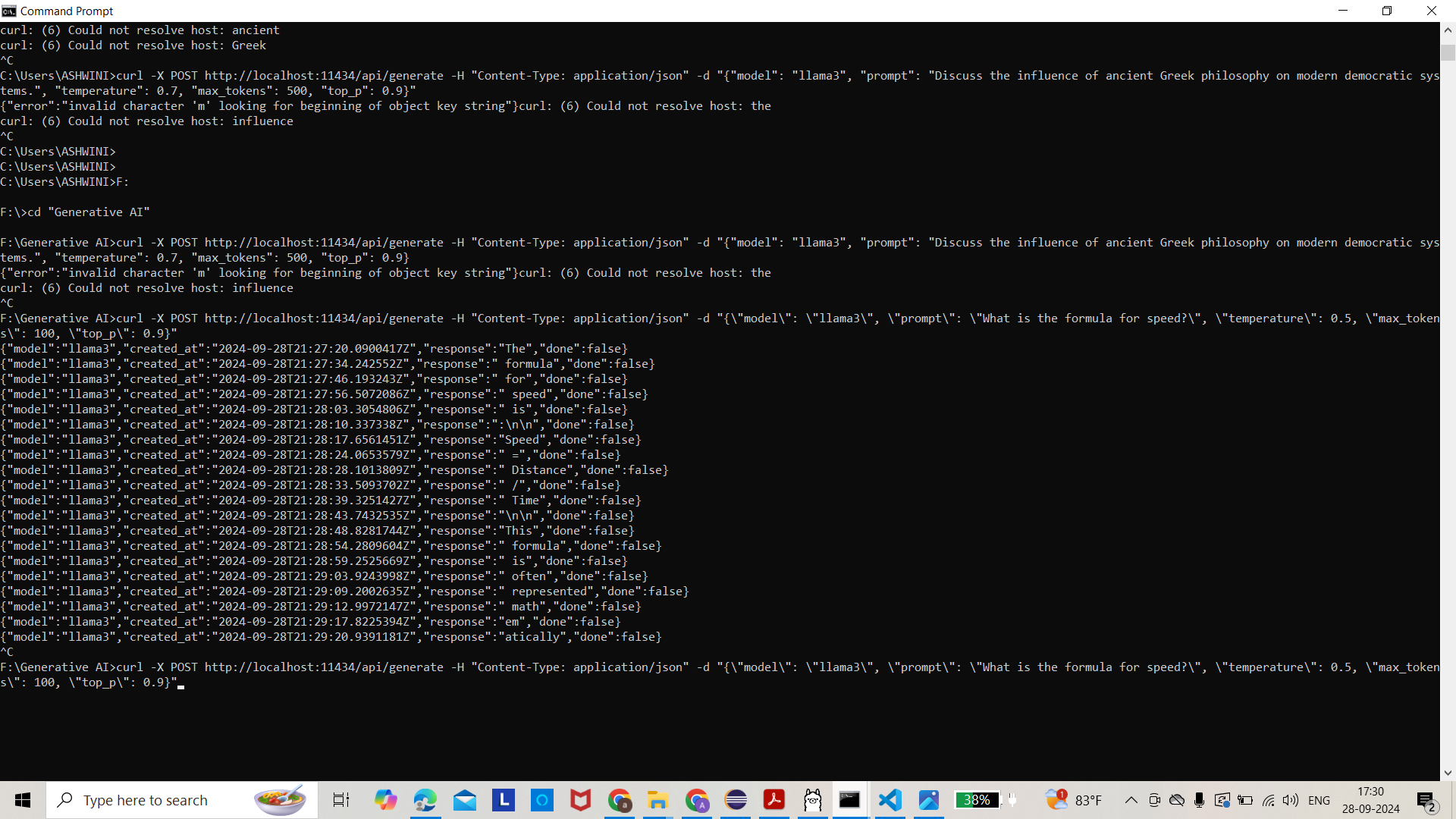
Change the parameters – llama3.2

curl -X POST [http://localhost:11434/api/generate](https://www.google.com/url?q=http%3A%2F%2Flocalhost%3A11434%2Fapi%2Fgenerate) -H "Content-Type: application/json" -d "{"model": "llama3", "prompt": "Discuss the influence of ancient Greek philosophy on modern democratic systems.", "temperature": 0.7, "max\_tokens": 500, "top\_p": 0.9}"



Complex Query:-

curl -X POST [http://localhost:11434/api/generate](https://www.google.com/url?q=http%3A%2F%2Flocalhost%3A11434%2Fapi%2Fgenerate) -H "Content-Type: application/json" -d "{"model": "llama3", "prompt": "What is the formula for speed?", "temperature": 0.5, "max\_tokens": 100, "top\_p": 0.9}"



Use curl commands to interact with GPT2

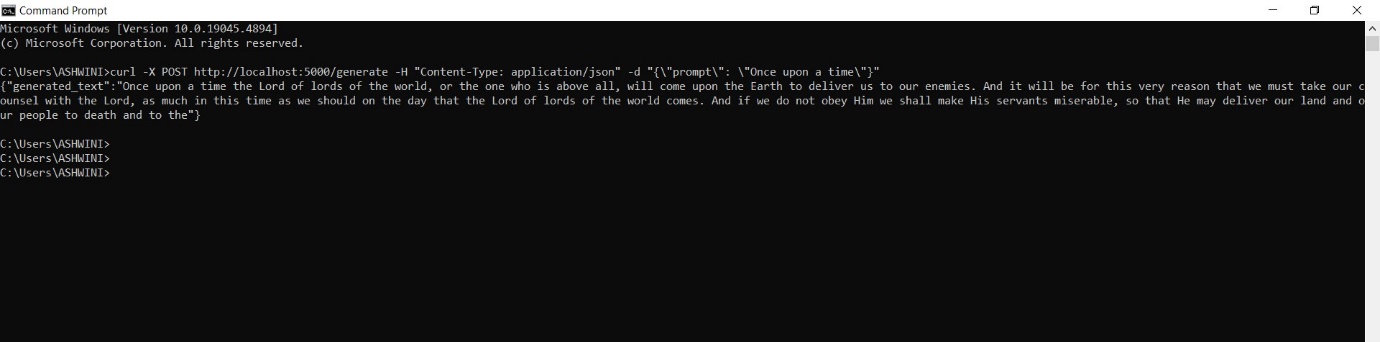
-pip install transformers gradio flask torch

* Launch the **Gradio interface** for web-based interaction with GPT-2.
* Start a **Flask server** to allow interaction with GPT-2 via curl commands.

-Created the python file 🡪 python gpt2\_gradio\_flask.py

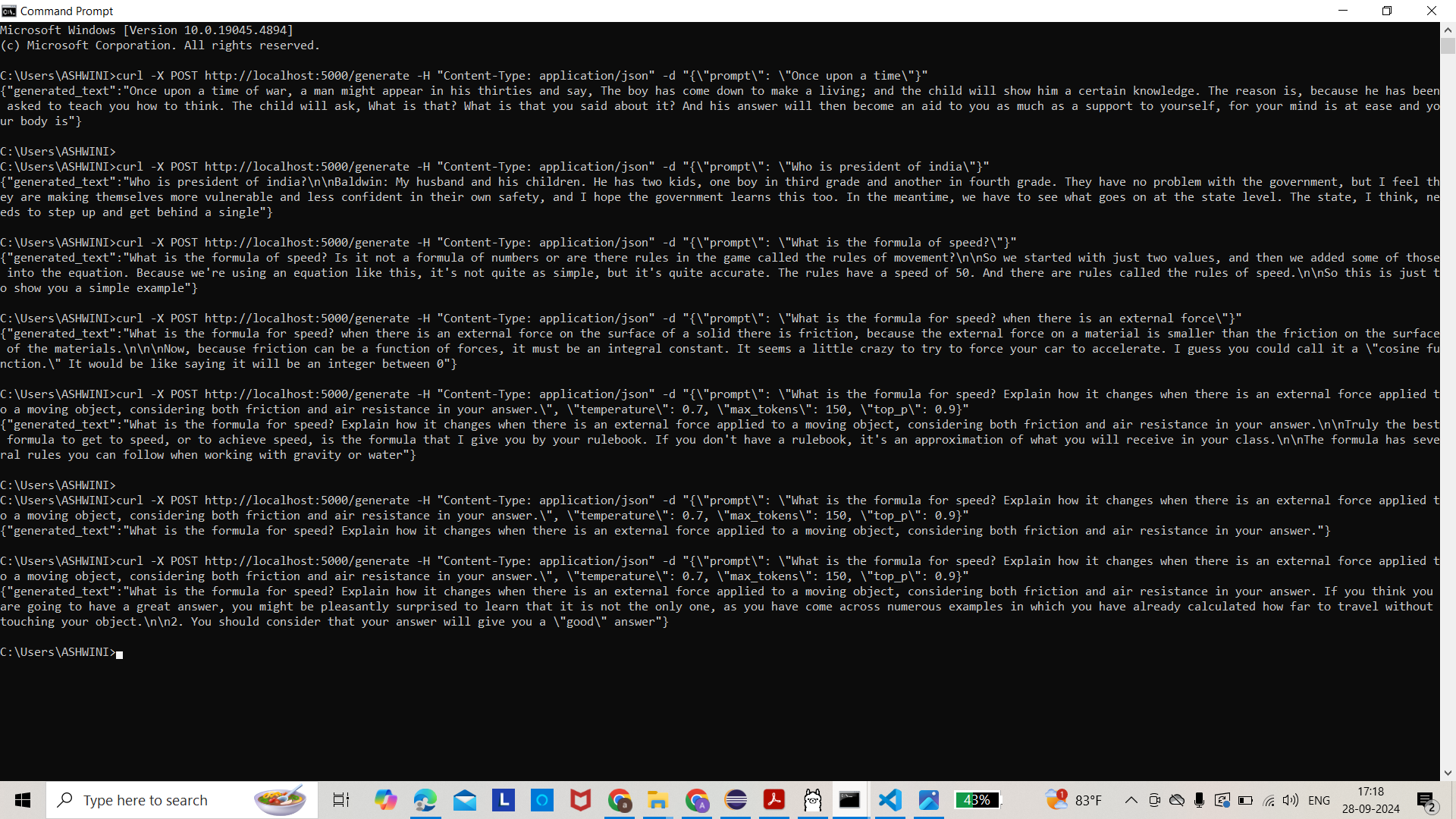
-Simple Query:

curl -X POST http://localhost:5000/generate -H "Content-Type: application/json" -d "{\"prompt\": \"Once upon a time\"}"



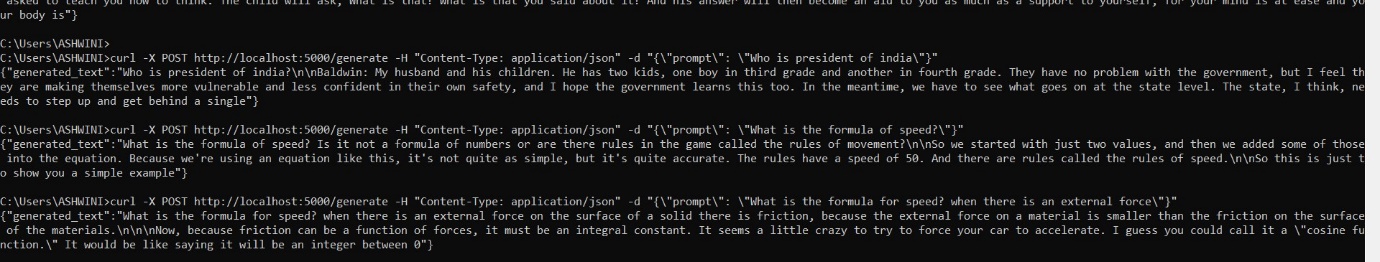
Change parameters :

curl -X POST [http://localhost:5000/generate](https://www.google.com/url?q=http%3A%2F%2Flocalhost%3A5000%2Fgenerate) -H "Content-Type: application/json" -d "{"prompt": "What is the formula for speed? Explain how it changes when there is an external force applied to a moving object, considering both friction and air resistance in your answer.", "temperature": 0.7, "max\_tokens": 150, "top\_p": 0.9}"



Complex Query:

curl -X POST http://localhost:5000/generate -H "Content-Type: application/json" -d "{\"prompt\": \"What is the formula for speed? when there is an external force\"}"



**Observations on response quality, speed, and limitations**

Simple Query Observations:

Llama3.2:

* Query: "Once upon a time"
* Response Quality: The response tends to produce coherent stories, often generating a continuation based on the provided phrase. Llama3.2, being a more advanced and larger model, generates longer and more contextually relevant outputs than GPT-2. The storytelling is usually consistent, with logical continuation of the input text.
* Speed: Slightly slower than GPT-2, likely due to its larger size and complexity, but still fast for generating short stories or narrative continuations.
* Limitations: While Llama3.2 tends to be consistent, it can sometimes generate overly creative or fantastical elements that may not align with the user's intended context.

GPT-2:

* Query: "Once upon a time"
* Response Quality: GPT-2 is capable of generating simple continuations, but the quality might degrade faster than Llama3.2, with less coherence in longer stories. The response might also be less creative compared to Llama3.2, but for simpler stories, it can perform adequately.
* Speed: Faster than Llama3.2 due to its smaller size, with shorter response times for small queries.
* Limitations: GPT-2 often fails to maintain context in longer generations and may produce repetitive or shallow content, especially when compared to larger models like Llama3.2.

2. Changing Parameters Observations:

Llama3.2:

* Query: "Discuss the influence of ancient Greek philosophy on modern democratic systems."  
  Parameters: temperature=0.7, max\_tokens=500, top\_p=0.9
* Response Quality: The model is able to generate detailed and in-depth responses with good historical context. The output quality with temperature set to 0.7 ensures the response is creative without being too random. It typically covers key figures like Socrates, Plato, and Aristotle, and links their ideas (like democracy, justice, and citizenship) to modern systems of governance.
* Speed: Slower due to the higher token limit and complexity of the query. The model takes longer to generate more thoughtful responses but remains reasonable for short-to-medium queries.
* Limitations: While Llama3.2 performs well, it can occasionally generate overly verbose responses or delve too deep into philosophical aspects, making the answer tangential to the main query.

GPT-2:

* Query: "What is the formula for speed? Explain how it changes when there is an external force applied to a moving object, considering both friction and air resistance."  
  Parameters: temperature=0.7, max\_tokens=150, top\_p=0.9
* Response Quality: GPT-2 can produce a reasonably good explanation for simpler physics-based questions and provide a surface-level explanation about external forces like friction and air resistance. However, it may lack the depth and detailed examples that Llama3.2 provides.
* Speed: Faster than Llama3.2, especially for smaller queries, due to the smaller size of the model.
* Limitations: The response is often more straightforward, with less depth and creativity. It may not fully explain complex concepts like air resistance and friction in detail, and the examples provided might be less relevant or incomplete.

3. Complex Query Observations:

Llama3.2:

* Query: "What is the formula for speed? when there is an external force"
* Response Quality: Llama3.2 provides a detailed explanation that begins with the formula for speed (speed = distance/time) and expands on how external forces (like friction, air resistance, or applied force) alter speed over time. It often brings in complex physics concepts such as Newton's Second Law and discusses how force influences acceleration and speed.
* Speed: Slightly slower due to the complexity of the query and the need to generate a more technical explanation. However, it remains responsive within reasonable limits for this task.
* Limitations: Llama3.2 may sometimes provide more information than necessary, including tangential concepts like acceleration and mass, which might not be directly related to the query.

GPT-2:

* Query: "What is the formula for speed? when there is an external force"
* Response Quality: GPT-2 gives a simple explanation of the formula for speed (speed = distance/time) but struggles with deeper physics explanations. It might mention friction or external forces but lacks detailed analysis and tends to stop at superficial explanations without providing comprehensive insight into the topic.
* Speed: Very fast due to the model’s smaller size, but the response is much less detailed compared to Llama3.2.
* Limitations: GPT-2’s performance on more complex queries is limited. It may generate short and incomplete answers or fail to fully address complex aspects like how speed is affected by external forces (friction, air resistance, etc.).

4. Overall Observations on Performance:

* Llama3.2:
  + Strengths: High-quality, in-depth, and creative responses, especially for complex queries. Good for generating long-form answers with historical or technical content. The model is better suited for more involved conversations that require reasoning.
  + Weaknesses: Slower
* GPT-2:
  + Strengths: Faster and simpler responses, good for basic queries and short explanations. Works well when you need a concise answer quickly.
  + Weaknesses: Limited depth and coherence for longer or more complex queries. Tends to produce more repetitive or generic content when dealing with higher-level topics.

Conclusion:

* Llama3.2 is better for handling complex, multi-part questions, delivering higher-quality responses but at the cost of speed.
* GPT-2 is faster and handles simple queries well but struggles with complex topics, making it less suitable for technical or detailed explanations.

**Analysis of the Interaction Method Using curl for Llama3 and GPT-2**

**1. Ease of Use with curl**

**Llama3:**

* **Interaction**: Llama3 is accessed via the **Ollama** API using the curl command. The interaction is simple when the JSON format is properly structured (correct usage of quotes and escape characters in Windows).
* **Response Handling**: The API for Llama3 handles complex queries better due to its more extensive training on recent data and advanced architecture. Once the request is properly structured, Llama3 can respond to both simple and complex prompts efficiently.

**GPT-2:**

* **Interaction**: Using curl with a **Flask** API is straightforward and allows flexibility to handle multiple parameters such as temperature, max\_tokens, etc. This setup is particularly beneficial for users looking to integrate GPT-2 into web services.
* **Response Handling**: Since GPT-2 is smaller and less capable than Llama3, it tends to respond faster but with less coherence, especially for more complex queries.

**Observations:**

* **JSON Formatting**: one of the most common pitfalls when using curl with these APIs is incorrect JSON formatting, especially in Windows. Proper use of escaped quotes is essential to avoid errors like invalid character 'm' looking for beginning of object key string.

**2. Response Quality**

**Llama3:**

* **Strengths**: Llama3 produces more coherent, context-aware responses for both simple and complex queries. For example, queries like "Discuss the influence of ancient Greek philosophy on modern democratic systems" are handled with more in-depth insights and greater contextual knowledge.
* **Complex Queries**: Llama3 excels at handling multi-turn conversations and can maintain context across different questions. This is valuable when users want a sophisticated conversation or explanation of complex subjects.
* **Weaknesses**: overly detailed responses that are tangential to the user's request, particularly when generating long-form content.

**GPT-2:**

* **Strengths**: GPT-2 performs well with simple, short queries, such as "What is the formula for speed?" It provides concise answers but may lack the depth needed for complex technical questions.
* **Complex Queries**: GPT-2 struggles with maintaining coherence for multi-turn conversations or complex queries. For example, it may generate non-sequiturs or irrelevant information when asked to explain advanced concepts like the impact of external forces on speed.
* **Weaknesses**: For complex queries, the model tends to become repetitive or fails to provide technically accurate answers. This was evident when it returned responses about friction in terms that seemed incomplete or off-topic.

**3. Speed and Performance**

**Llama3:**

* **Speed**: Llama3 is slower in generating responses due to its larger size and more complex architecture. This is especially true for long, detailed responses where the model needs more time to process and generate.
* **Performance**: While slower, Llama3’s output is more accurate, consistent, and sophisticated, especially in handling queries that require deeper reasoning or technical knowledge.

**GPT-2:**

* **Speed**: GPT-2 responds faster due to its smaller size, making it suitable for quick responses in real-time applications. However, its responses may lack depth when compared to Llama3.
* **Performance**: For quick, simple answers, GPT-2 performs efficiently. However, its inability to maintain context or deliver high-quality responses in complex queries is a limiting factor.

**4. Parameter Control (temperature, max\_tokens, etc.)**

**Llama3:**

* **Temperature**: Adjusting the temperature for Llama3 yields good control over the creativity of the responses. Lower temperatures produce factual, concise responses, while higher values introduce more creativity. For example, in your queries, lowering the temperature helped focus the response on technical accuracy rather than creative storytelling.
* **Max Tokens**: Llama3 handles larger max\_tokens well, generating more complete answers when asked for detailed explanations. This is helpful in cases like discussing the influence of philosophical ideas or technical breakdowns like speed and force.

**GPT-2:**

* **Temperature**: GPT-2's responses become less coherent with higher temperatures, leading to more random outputs. For technical queries, it is recommended to keep the temperature low to avoid irrelevant responses.
* **Max Tokens**: Increasing the max\_tokens value can help GPT-2 generate more extensive responses, but the quality diminishes quickly, especially in complex topics where the model struggles to maintain coherence.

**5. Limitations**

**Llama3:**

* **Limitations**: Llama3 may sometimes over-explain or provide unnecessary detail, especially in technical or philosophical discussions. Also, the model's processing time can be a drawback for real-time applications requiring fast responses.

**GPT-2:**

* **Limitations**: GPT-2’s limitations are more apparent when asked complex or multi-turn questions. It tends to generate incomplete or irrelevant responses. Its smaller training size also limits its understanding of highly technical concepts.

**Conclusion:**

* **Llama3** is the better choice for handling complex queries, multi-turn conversations, and deeper reasoning, despite being slower than GPT-2.
* **GPT-2** works well for simple, straightforward queries but is less reliable for complex or technical questions.

By adjusting parameters like temperature and max\_tokens, both models can be tuned to suit specific tasks, though Llama3 offers more versatility and quality, especially for knowledge-heavy tasks.