**Project-1: Text Generation**

**Submitted By: Rajesh Kumar Sharma**

**Date: 29-Nov-24**

**Project-1:** **Develop a Text generation application and expose it as a FastAPI**

**Code Link:** <https://github.com/rajeshksharmasls/GenAI/tree/master/GenAI/Test%20Generation>

**Dataset:** *None*

**Process:** Below is the summarization of the code:

This code implements a text generation API using FastAPI, PyTorch, and a pre-trained Qwen model from Hugging Face's Transformers library. Below is a breakdown:

Initialization:

A FastAPI application is set up with a title and version.

A pre-trained language model (Qwen/Qwen2.5-0.5B) and its tokenizer are loaded.

Tokenization Configuration:

Ensures the tokenizer has a valid pad\_token\_id. If missing, it defaults to the eos\_token\_id.

Data Validation with Pydantic:

StoryRequest: Validates incoming requests, requiring a title (string).

StoryResponse: Defines the structure of the API's response with a title and a generated story.

API Endpoint:

A POST endpoint (/story) is created to handle requests for story generation.

The process:

Validates the title field from the request.

Uses the title to craft a prompt: "Write a short story about {title}."

Tokenizes the prompt and generates text using the language model with:

Sampling methods (temperature, top\_k, and top\_p) for more creative outputs.

max\_new\_tokens to limit response length.

Decodes the generated output into a human-readable string.

Returns the title and the generated story as a JSON response.

Error Handling:

Raises a 400 HTTP error if the title field is empty.

**Key Features:**

Interactive Story Generation: Generates short stories based on a user-provided title.

Pre-trained Model Integration: Uses a causal language model (Qwen) for text generation.

FastAPI Benefits: Provides a fast, scalable API with built-in validation and error handling.

Requirements.txt

transformers

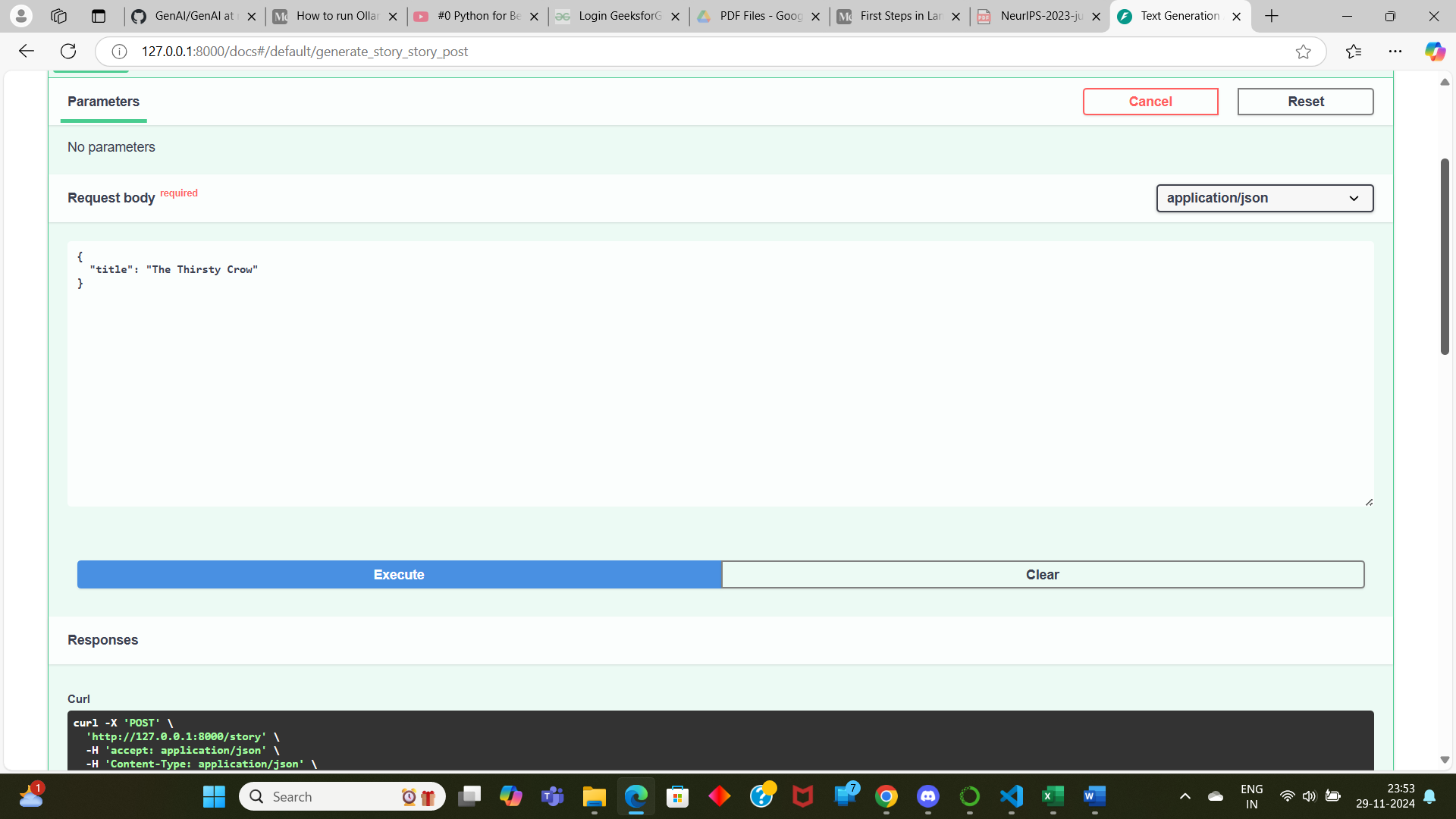
fastapi

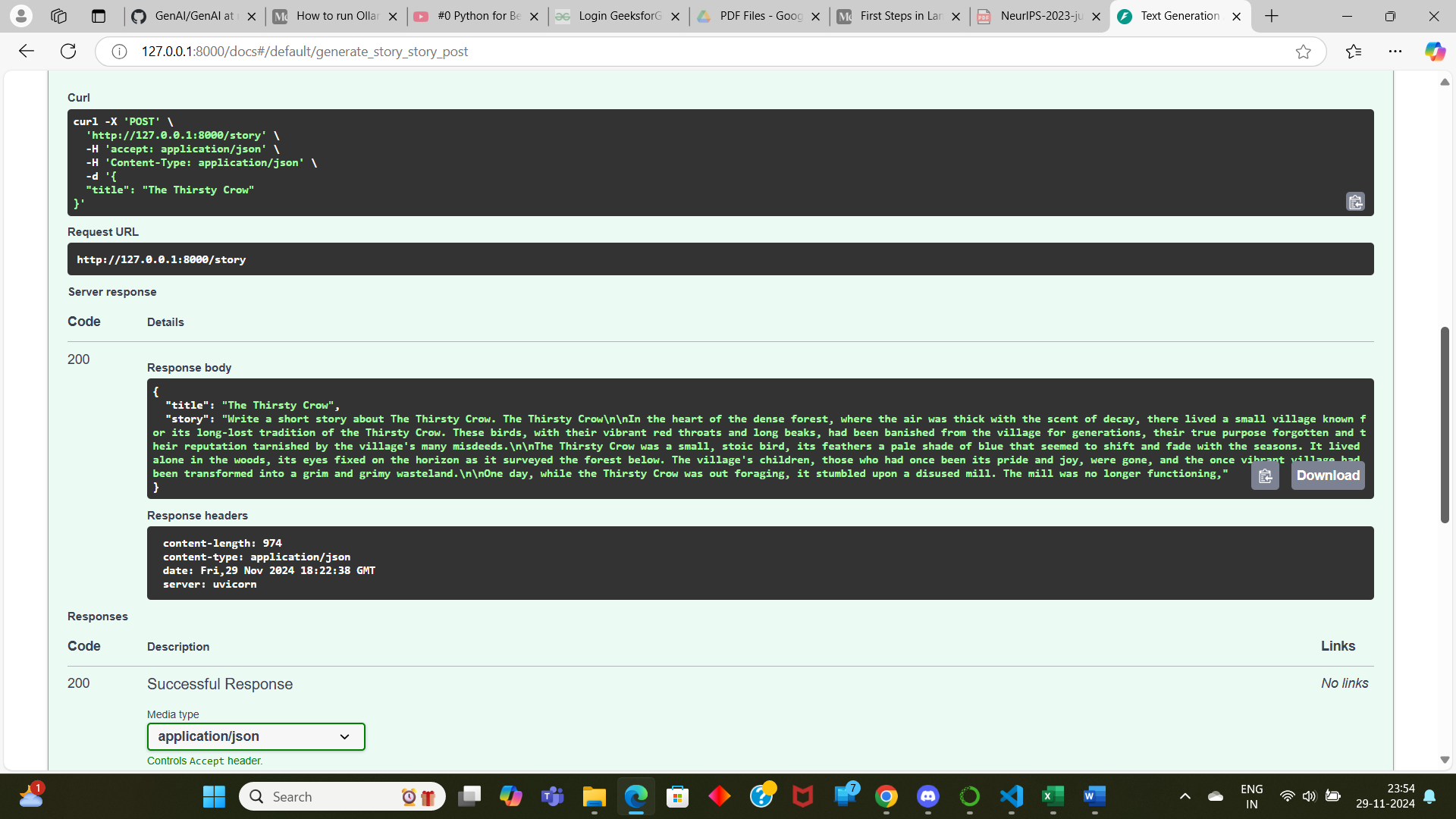
uvicorn

torch

sentencepiece

**Snapshots:**

**

**

**Conclusion:** *Text is getting generated as per the topic provided.*