**Project-2: Text Translation**

**Submitted By: Rajesh Kumar Sharma**

**Date: 29-Nov-24**

**Project-2:** **Develop a Text translation application and expose it as a FastAPI**

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

**Dataset:** *None*

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

This code implements a **translation API** using **FastAPI** and Hugging Face's **T5-small** model for text-to-text generation. Below is a concise breakdown of its purpose and functionality:

**Key Components:**

1. **FastAPI Setup**:
   * A FastAPI application is initialized to handle HTTP requests.
2. **Model and Tokenizer**:
   * The T5-small model and its tokenizer from Hugging Face are loaded.
   * The model specializes in tasks like translation, summarization, and other text generation tasks.
3. **Input Validation**:
   * A TranslationRequest Pydantic model validates incoming requests, ensuring the following fields:
     + text: The input text to translate.
     + source\_lang: The source language of the text.
     + target\_lang: The desired language to translate the text into.
4. **Translation Logic**:
   * The translate\_text function:
     + Constructs a prompt like "translate {source\_lang} to {target\_lang}: {text}" for T5.
     + Tokenizes the prompt and generates the translation using the model.
     + Decodes the generated tokens into human-readable text.
5. **API Endpoint**:
   * A **POST** endpoint (/translate/) is created to handle translation requests.
   * The process:
     + Validates the request using the TranslationRequest schema.
     + Calls the translate\_text function to generate the translation.
     + Returns the original text and its translation as a JSON response.
   * **Error Handling**:
     + Catches and returns any exceptions with a 500 status code.
6. **Configuration Options**:
   * **TF\_ENABLE\_ONEDNN\_OPTS**: Disables oneDNN optimizations for consistent floating-point results.
   * **TF\_CPP\_MIN\_LOG\_LEVEL**: Reduces TensorFlow log verbosity.

**Key Features:**

* **Dynamic Translation**: Can translate text between any supported language pairs using the prompt-based T5 model.
* **Robust API**: FastAPI ensures efficient request handling and error management.
* **Customizable Output**: Allows users to specify both source and target languages dynamically.

This API can serve as the backbone of a lightweight, flexible translation service!

Requirements.txt

transformers

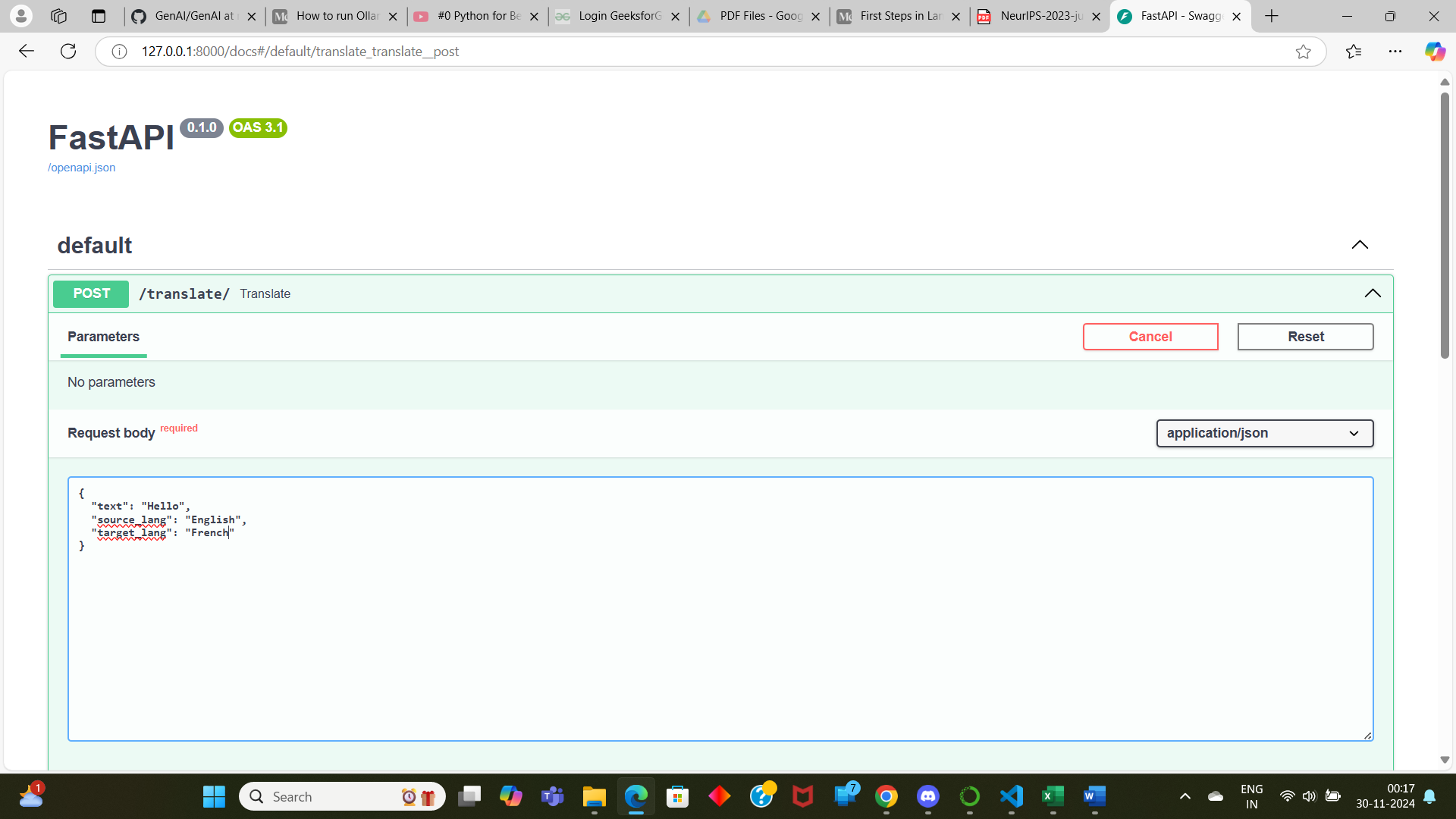
fastapi

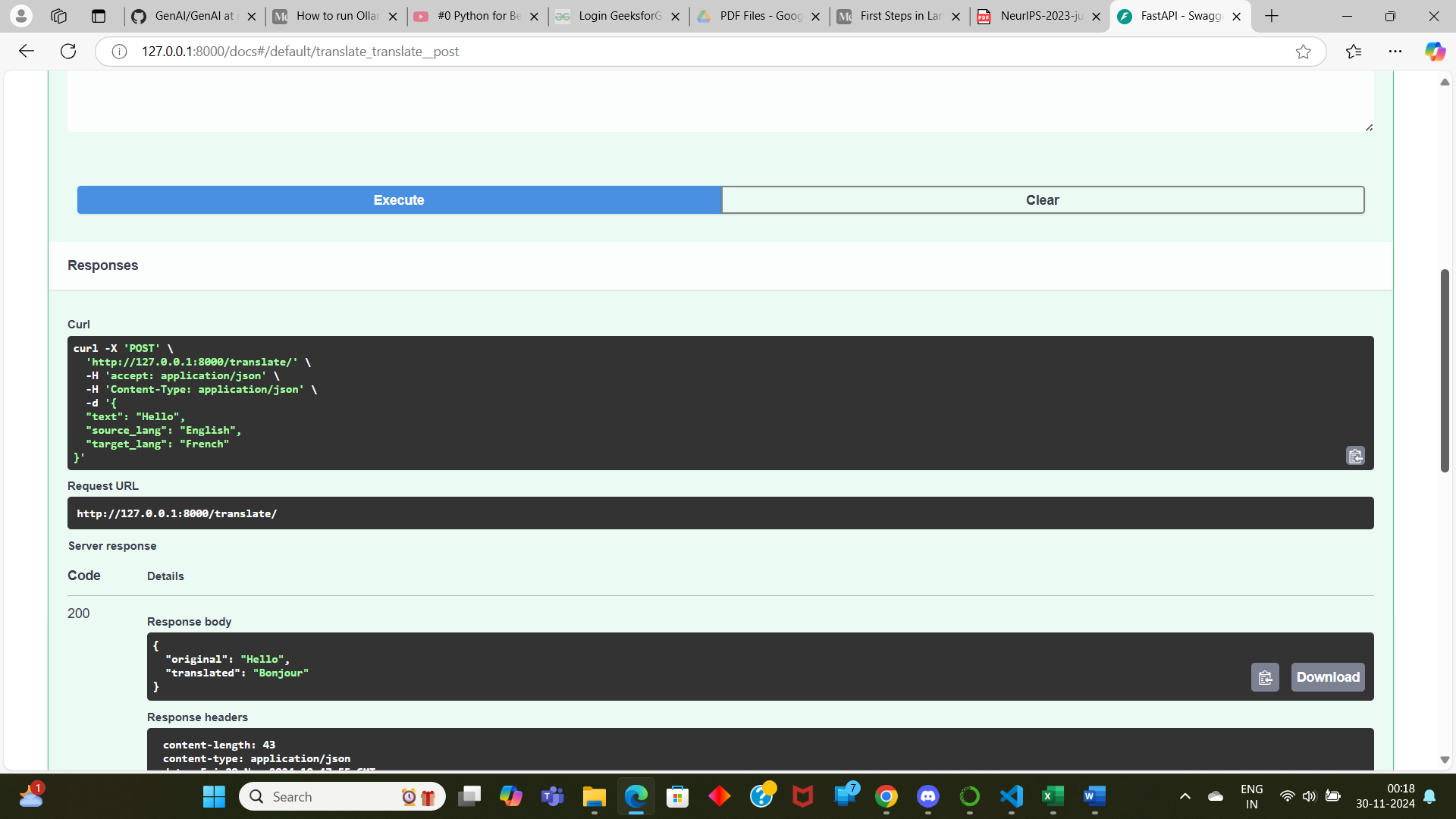
uvicorn

torch

sentencepiece

**Snapshots:**

**

**

**Conclusion:** *Text is getting translated as per the text provided.*