Project Introduction: Advanced Text Generation Application

Objective

The goal of this project is to create an advanced text generation application using the Hugging Face transformers library and the GPT-2 model. This application will allow users to input a prompt, customize text generation parameters, generate multiple text outputs, and save these outputs to a file.

Prerequisites

Before you start, ensure you have the following:

1. **Python Version**: Python 3.7 or later installed on your machine.
2. **Package Installation**: Necessary libraries required for this project (refer below).
3. **Basic Python Knowledge**: Familiarity with Python programming concepts.
4. **Operating System**: Compatibility with Windows, macOS, or Linux (recommended).
5. **Development Environment**: A text editor or IDE such as VSCode, PyCharm, or Jupyter Notebook.

Step-by-Step Instructions

Step 1: Install Required Packages

1. **Open your terminal or command prompt**.
2. **Install Required Libraries**: Run the following command to install the necessary packages:

```
pip install transformers torch
```

Step 2: Create the Python Script

1. **Create a new Python file**:
    - Name it 'text_generator.py' or any name you prefer.
2. **Open the file in your text editor or IDE**.

Step 3: Implement the Code

Copy and paste the following code into your text_generator.py file. This code includes the necessary logic for text generation.

```
import time
from transformers import pipeline

# Load the text generation pipeline using GPT-2
generator = pipeline('text-generation', model='gpt2', device=-1)  # Use
CPU

def generate_text(prompt, max_length=100, temperature=0.7,
num_return_sequences=1):
    results = generator(prompt, max_length=max_length,
num_return_sequences=num_return_sequences, temperature=temperature)
    return [result['generated_text'] for result in results]

def save_to_file(text, filename='generated_text.txt'):  # Change the
filename to save the results to a different file
    with open(filename, 'w', encoding='utf-8') as file:
        file.write(text)
    print(f"Generated text saved to {filename}")

def main():
    print("Welcome to the Advanced Text Generator!")

    prompt = input("Enter your prompt text: ")
    max_length = int(input("Enter the maximum length of generated text
(recommended 30-100): "))
    temperature = float(input("Enter temperature (recommended range 0.5-
1.0, higher means more random): "))
    num_return_sequences = int(input("Enter the number of text paragraphs
to generate: "))

    # Start timing
    start_time = time.time()

    # Generate text
    generated_texts = generate_text(prompt, max_length=max_length,
temperature=temperature, num_return_sequences=num_return_sequences)

    # Stop timing
    end_time = time.time()
    print(f"\nText generation took: {end_time - start_time:.2f}
seconds\n")

    for i, generated_text in enumerate(generated_texts, 1):
        print(f"Generated Text Paragraph {i}:")
        print(generated_text)
        print("-" * 40)
```

```
    save_choice = input("Would you like to save the generated text to a
file? (yes/no): ")
    if save_choice.lower() == 'yes':
        all_texts = "\n\n".join(generated_texts)
        save_to_file(all_texts)

if __name__ == "__main__":
    main()
```

Step 4: Run the Application

1. **Open your terminal/command prompt**.
2. **Navigate to the directory** where your text_generator.py file is located.
3. **Run the Python script**:

```
python text_generator.py
```

4. **Follow the prompts** in the terminal to input your text prompt, set text generation parameters, and choose whether to save the generated text.

Step 5: Test and Experiment

1. **Try different prompts**: Explore how the model responds to various input prompts.
2. **Adjust the parameters**: Experiment with different values for maximum length, temperature, and number of outputs. Observe how it influences the generated text.

Step 6: Make Observation and Conclusion

1. Make your own observations and conclusions based on the results and save it to a Word Document.
2. Upload your conclusion to the quiz.