

StoryCrafter - An AI Story Generator

Phase 2: Project Execution and Demonstration

1. Project Title:

StoryCrafter - An AI Story Generator

2. Objective Recap:

The objective of this project is to build a story generation system using Generative AI models. It takes user input such as character and setting, and produces a coherent and creative short story using the GPT-2 language model. This demonstrates the power of NLP in assisting creative writing and narrative construction.

3. Technologies Used:

Name	What it is	Use in Project
Python	High-level programming language	Core development of backend and frontend logic
HuggingFace Transformers	NLP library with pre-trained language models	Used to load and run language models like GPT-2
GPT-2 (via Transformers)	Generative Pre-trained Transformer 2 language model	Used for generating creative and coherent stories
PyTorch	Deep learning framework	Backend engine for model computation and inference
Streamlit	Python library for building web apps	Used to build the user-friendly web interface
VS Code	Source code editor	Used for writing, managing, and testing project code
GitHub	Online code hosting and version control platform	Used for version control, collaboration, and sharing the code

4. Full Code Implementation:

Step 1: Install Required Libraries

```
pip install transformers streamlit torch
```

Step 2: Import Required Libraries

```
import streamlit as st
from transformers import pipeline, set_seed
import torch
```

Step 3: Load the Pretrained GPT-2 Model

```
device = 0 if torch.cuda.is_available() else -1
generator = pipeline("text-generation", model="gpt2", device=device)
set_seed(42)
```

Step 4: Build Streamlit Interface

```
st.set_page_config(page_title="StoryCrafter", page_icon="📝")
st.title("📝 StoryCrafter")
st.markdown("Craft short stories by entering a character and setting below.")

character = st.text_input("Character Name")
setting = st.text_input("Setting or Scenario")
length = st.slider("Story Length (tokens)", min_value=50, max_value=200, value=100)

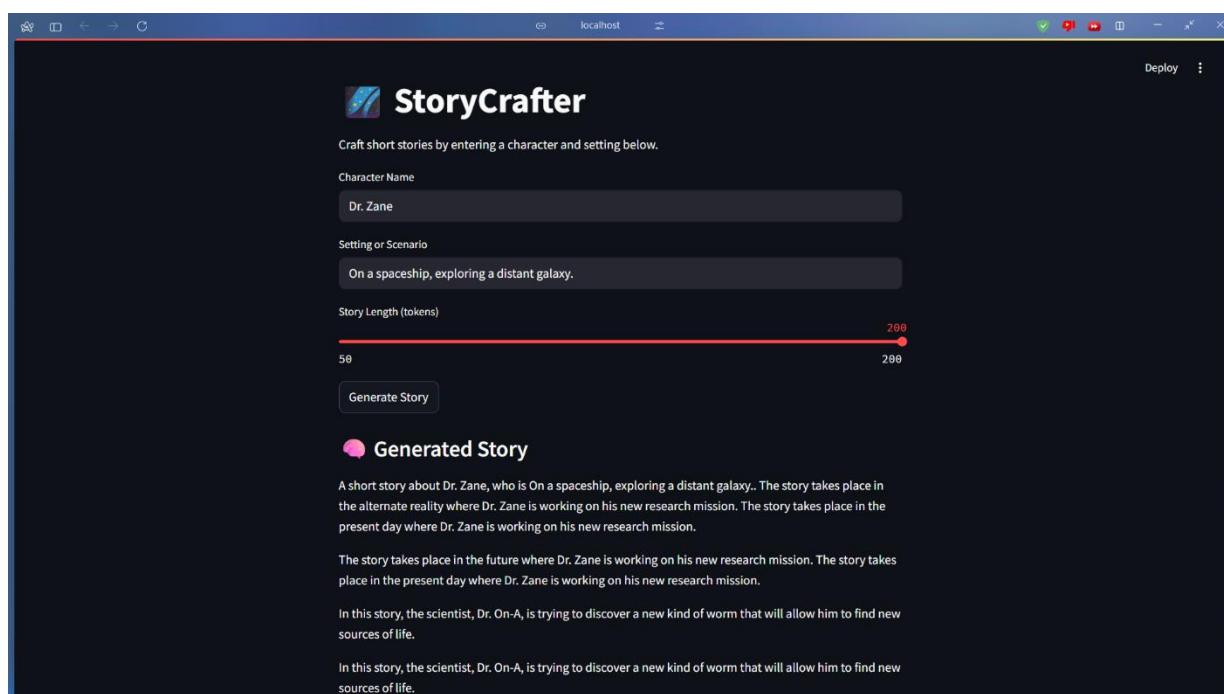
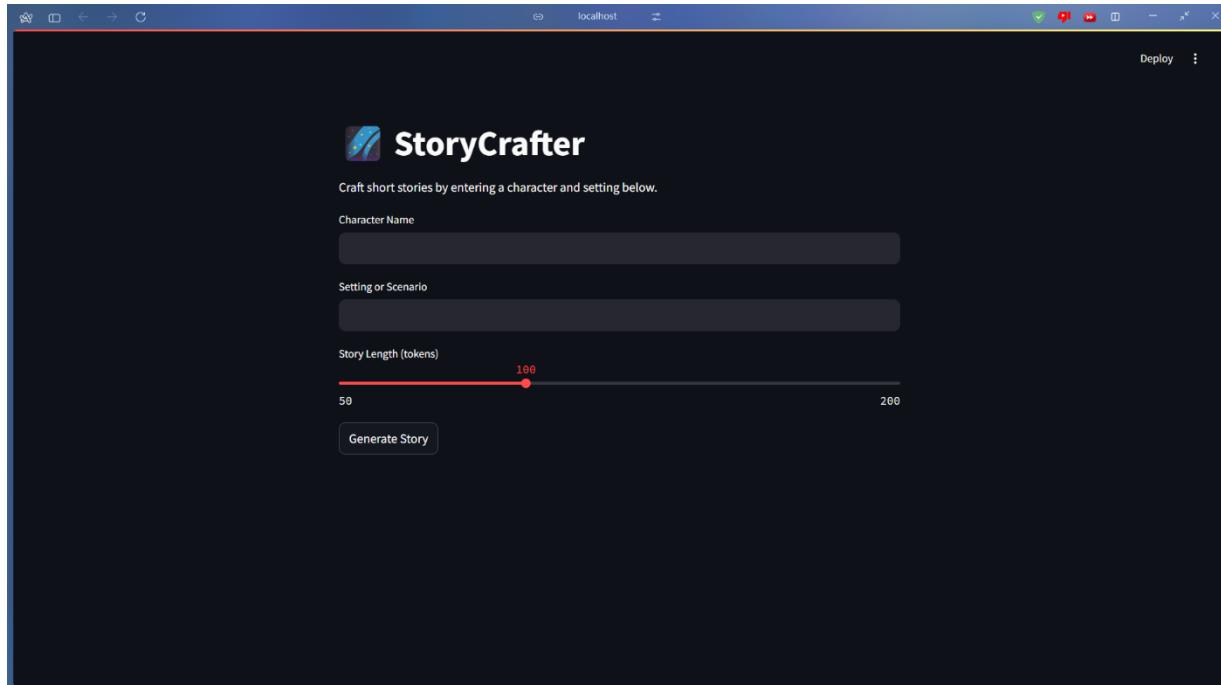
if st.button("Generate Story"):
    if not character.strip() or not setting.strip():
        st.warning("Please enter both a character and a setting.")
    else:
        prompt = f"A short story about {character}, who is {setting}."
        st.markdown("### 📝 Generated Story")
        result = generator(
            prompt,
            max_length=length,
            num_return_sequences=1,
            do_sample=True,
            temperature=0.8,
            top_p=0.95
        )
```

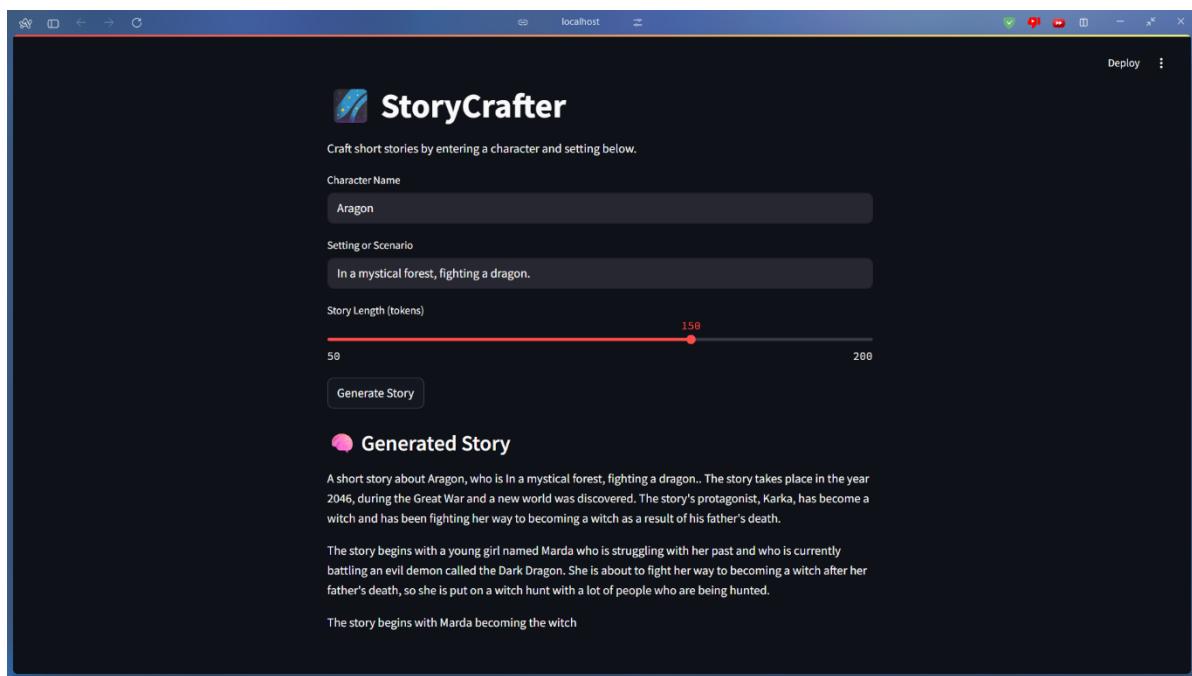
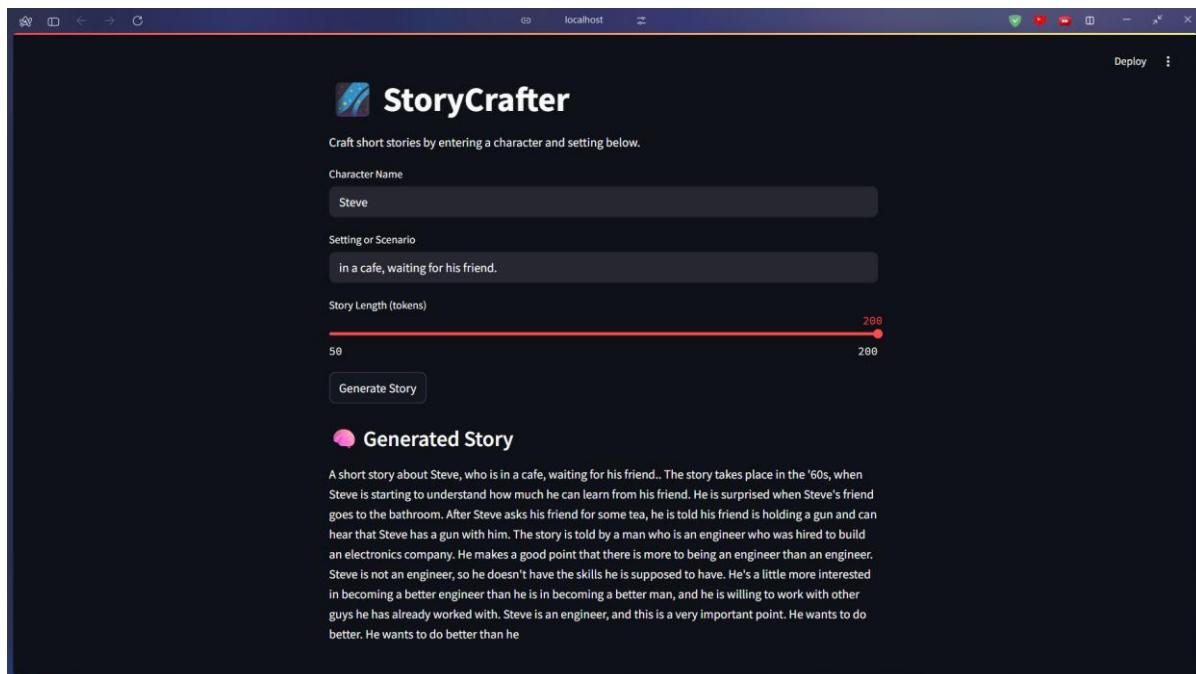
```
story = result[0]["generated_text"]
st.write(story)
```

Step 5: Run the Streamlit App

```
streamlit run app_streamlit.py
```

5. Output Screenshots:





6. Conclusion:

This project demonstrates the application of Generative AI through GPT-2 for real-time story generation based on user input. It showcases how powerful language models can be leveraged for creative writing tools, narrative support, and interactive AI-based content generation. The seamless web interface allows users to experiment and enjoy AI-generated storytelling in an intuitive way.

7. References:

- HuggingFace Transformers Documentation
 - <https://huggingface.co/docs/transformers/index>
- OpenAI GPT-2 Documentation
 - <https://openai.com/research/language-unsupervised>
- Streamlit Documentation
 - <https://docs.streamlit.io/>
- NLP and Text Generation Projects on GitHub
 - <https://github.com/topics/text-generation>