

Dream Text Forge – AI based Story Generator

1. Project Overview

DreamText Forge is an interactive web app built with **Streamlit** that lets you turn your ideas into AI-generated text magic. You can choose from curated story prompts or create your own, then watch the AI weave them into creative narratives.

2. Features

- **Prompt Suggestions** — Start with one of the built-in story starters.
- **Custom Prompt Input** — Write your own ideas from scratch.
- **Adjustable Output Length** — Fine-tune the maximum length of generated text.
- **Real-Time Generation** — Get AI-generated text instantly.
- **Lightweight UI** — Clean and easy-to-use Streamlit interface.
- **Cached Model Loading** — Faster performance thanks to caching.

3. How It Works

i. Model Loading

- a. When you first start the app, it loads the GPT-2 model from Hugging Face's transformers library using the pipeline("text-generation") API.
- b. This is wrapped in `@st.cache_resource` so the model stays in memory between runs, making subsequent generations much faster.

ii. User Prompt Selection

- a. You can either:
 - i. Pick from one of the built-in prompt suggestions (story starters), or
 - ii. Write your own from scratch.
- b. The selected or written prompt is editable before submission.

iii. Parameter Configuration

- a. You can control how long the generated text will be using the Max Length slider.

- b. In the code, other parameters like temperature (creativity) and repetition_penalty (avoid repeated phrases) are preset but can be changed in the source.

iv. Text Generation

- a. When you click Generate, the app sends your prompt to the GPT-2 model.
- b. GPT-2 predicts the next words in the sequence until it reaches the max_length or the end-of-sequence token.

v. Result Display

- a. The generated text is displayed immediately below in a clean, scrollable format.
- b. If no prompt is entered, a warning is shown.

vi. Caching for Speed

- a. Because load_model() is cached, the heavy GPT-2 model is only loaded once, greatly reducing response time after the first generation.

4. Code Explanation

```

1  import streamlit as st
2  from transformers import pipeline
3
4  # Cache the model loading to improve performance
5  @st.cache_resource
6  def load_model():
7      return pipeline("text-generation", model="gpt2")
8
9  generator = load_model()
10
11 # --- UI Config ---
12 st.set_page_config(page_title="DreamText Forge")
13 st.title("✨ DreamText Forge")
14 st.write("Shape your ideas into text magic with ✨")
15
16 # Prompt suggestions
17 prompt_suggestions = [
18     "Once upon a time, in a city where dreams wa",
19     "The scientist's latest invention would chan",
20     "On the edge of the galaxy, a lone explorer",
21     "The forest was silent, except for the whisp",
22     "In the year 2150, humans had finally master",
23 ]
24
25 # Dropdown for selecting a prompt
26 selected_prompt = st.selectbox(
27     "Choose a prompt suggestion:",
28     ["(Write my own)"] + prompt_suggestions
29 )
30
31 # Text area for custom or editable prompt
32
33 if selected_prompt == "(Write my own)":
34     prompt = st.text_area("Enter your prompt:", "")
35 else:
36     prompt = st.text_area("Edit or extend the chosen prompt:", selected_prompt)
37
38 # Max length setting
39 max_length = st.slider("Max length of output", 1, 100, 50)
40
41 # Generate button
42 if st.button("Generate"):
43     if prompt.strip() != "":
44         with st.spinner("Weaving your story..."):
45             result = generator(
46                 prompt,
47                 max_length=max_length,
48                 num_return_sequences=1,
49                 temperature=0.7,
50                 repetition_penalty=1.2,
51                 eos_token_id=50256 # End-of-seq
52             )
53         st.subheader("✨ Generated Output:")
54         st.write(result[0]['generated_text'])
55     else:
56         st.warning("Please enter or select a prompt")
57
58

```

Fig1: Input Snapshot

5. Output Snapshots

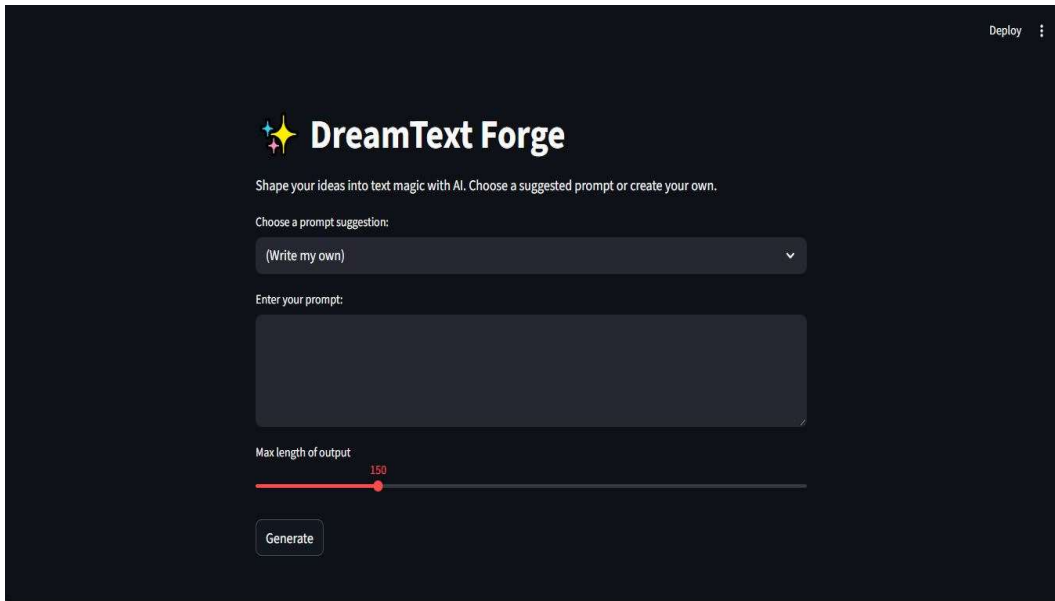


Fig2: Output Snapshot 1

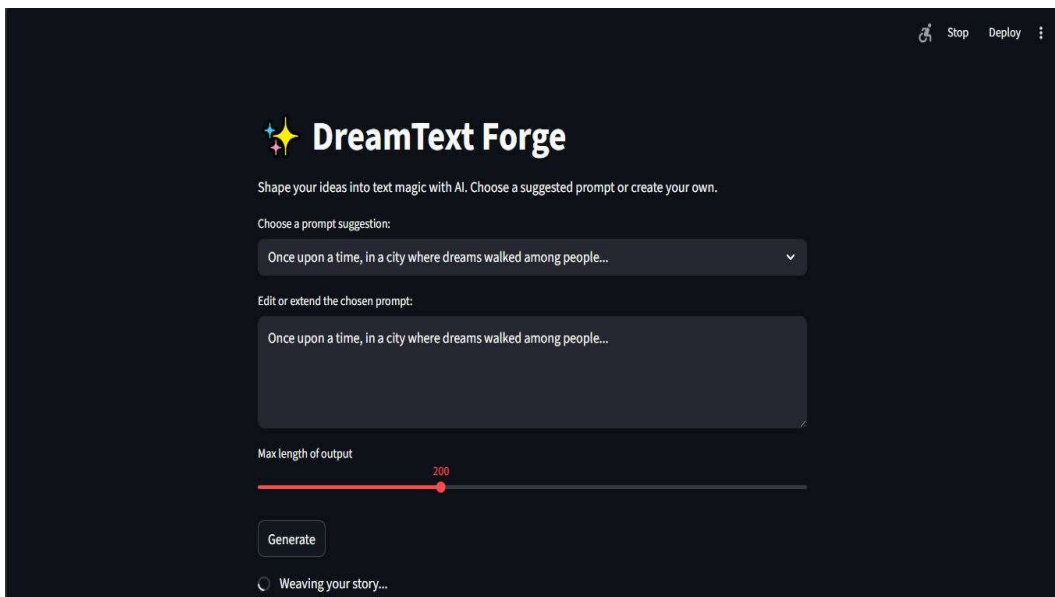


Fig3: Output Snapshot 2

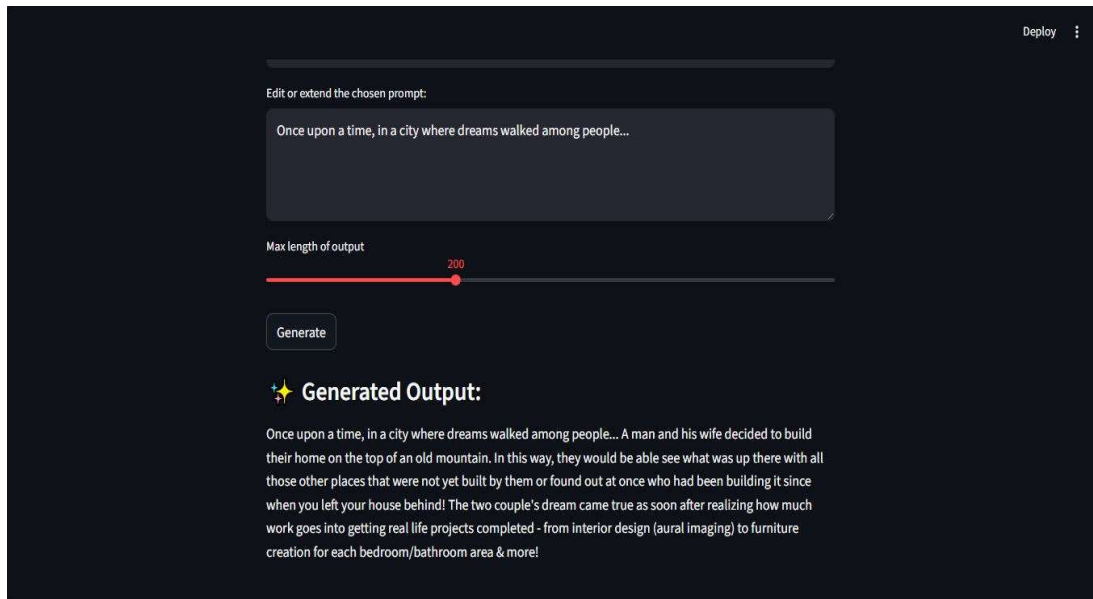


Fig4: Output Snapshot 3