# DreamWeaver AI – Interactive Story Generator

## 1. Introduction

The project DreamWeaver AI is an AI-powered storytelling application built using Streamlit. It allows users to:
- Choose a genre (Fantasy, Horror, Mystery, Sci-Fi, Romance, Adventure).
- Define characters (Protagonist and optional Antagonist).
- Generate interactive or full stories using GPT-Neo 125M (from Hugging Face).
- Download their generated stories.

The goal is to merge user creativity with AI story generation in an immersive, game-like interface.

## 2. System Architecture

The project follows a multi-page Streamlit application structure:

1. app.py → Home page with animated title, background, and a "Play" button.
2. Menu.py → Genre selection page (cards for each genre).
3. Character.py → Character creation (protagonist and antagonist).
4. Story.py → Story generation using GPT-Neo in two modes:
   - Interactive Mode
   - Full Story Mode

Folder Setup:

```
project/
│── app.py
│── Menu.py
│── Character.py
│── Story.py
│── img/ (backgrounds & genre images)
```

## 3. Detailed Code Explanation

### a) app.py – Home Page

- Hides Streamlit sidebar using CSS.
- set_background() loads and encodes background image in Base64.

- Adds glowing animated header 'DreamWeaver AI'.
- Play button redirects to Menu.py.

### b) Menu.py – Genre Selection

- Loads multiple genre images.
- get_base64_image() encodes images for inline CSS.
- Displays genre cards with title + description.
- Next button → Character.py.

### c) Character.py – Character Setup

- Custom background with transparency effects.
- Inputs for protagonist (required) and antagonist (optional).
- Back button → Home, Continue button → Story.py.
- Validates protagonist input, stores characters in session_state.

### d) Story.py – Story Generation

- Loads GPT-Neo 125M model + tokenizer.
- Handles session_state for genre, characters, story.
- Interactive Mode → step-by-step AI-assisted storytelling.
- Full Story Mode → AI generates full story.
- Download buttons for story output.

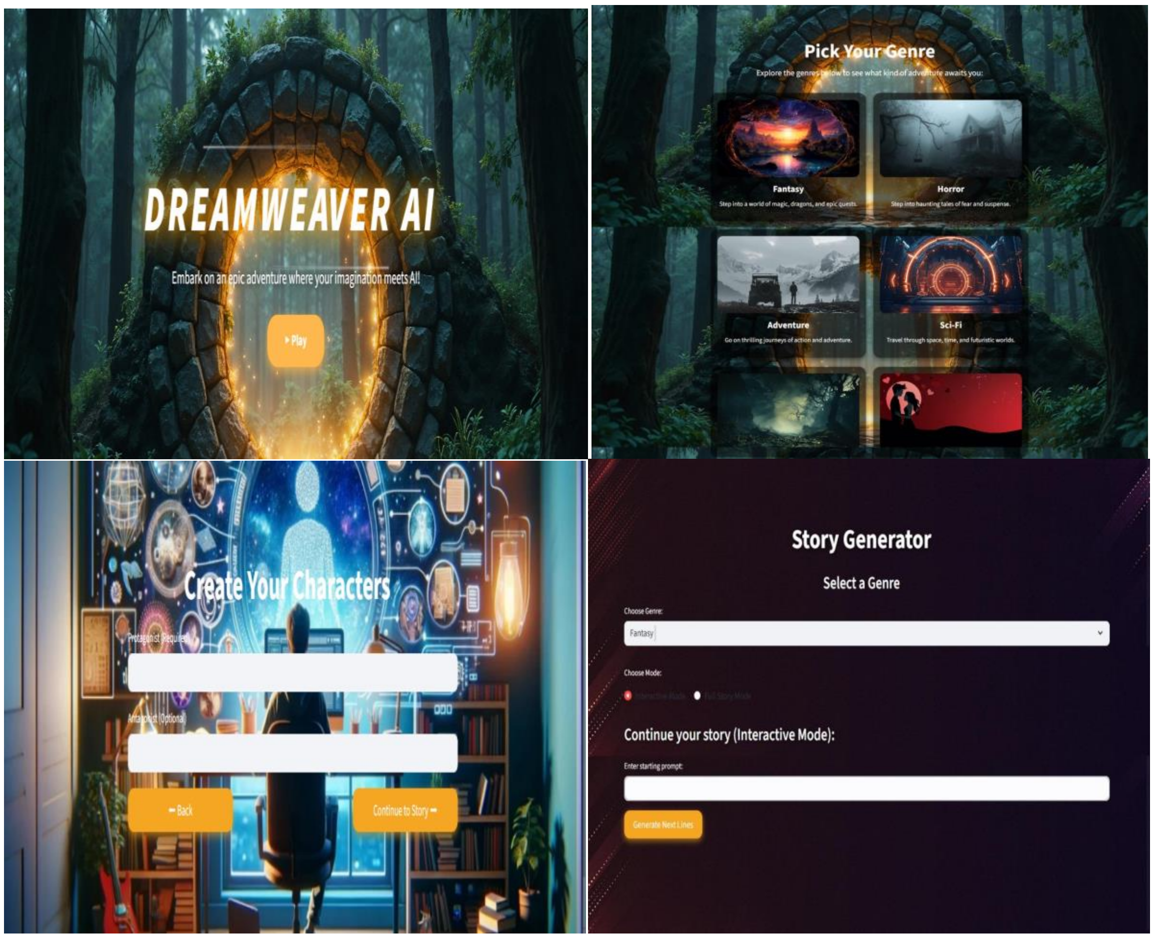
## 4. Technologies Used

- Frontend/UI: Streamlit, HTML/CSS
- Backend/AI: Hugging Face Transformers (GPTNeoForCausalLM, GPT2Tokenizer)
- Programming Language: Python
- Data Handling: st.session_state
- Deployment Ready: Streamlit Cloud

## 5. Unique Features

✅Multi-page game-like navigation
✅Animated glowing header and styled backgrounds
✅Genre-based story personalization
✅Interactive storytelling mode
✅Downloadable stories

## 6. Result


DREAMWEAVER AI
Embark on an epic adventure where your imagination meets AI!
▸ Play
Pick Your Genre
Explore the genres below to see what kind of adventure awaits you:
Fantasy
Step into a world of magic, dragons, and epic quests.
Horror
Step into haunting tales of fear and suspense.
Adventure
Go on thrilling journeys of action and adventure.
Sci-Fi
Travel through space, time, and futuristic worlds.
Create Your Characters
Protagonist (Required)
Antagonist (Optional)
⬅ Back
Continue to Story ➡
Story Generator
Select a Genre
Choose Genre:
Fantasy
Choose Mode:
Continue your story (Interactive Mode):
Enter starting prompt:
Generate Next Lines

## 7. Conclusion

The project showcases a creative use of AI + UI/UX design to build an engaging storytelling platform. It demonstrates:
- Strong Streamlit UI customization.
- Integration of Transformer models for text generation.
- A seamless flow from genre selection → character creation → AI-generated stories.

## 8. Future Enhancements

- Multi-language story generation (support for Hindi, Tamil, etc.).
- Voice integration: Convert stories into audiobooks using TTS (Text-to-Speech).
- Story continuation from saved progress with user accounts.
- Integration with image generation models (e.g., Stable Diffusion) for story illustrations.

- Mobile-friendly interface or Android/iOS app version.
- Adding collaborative storytelling where multiple users can co-create a story in real time.
- Enhanced personalization by tracking user preferences for genres, characters, and themes.

## 9. References

- Streamlit Documentation: https://docs.streamlit.io/
- Hugging Face Transformers: https://huggingface.co/transformers/
- EleutherAI GPT-Neo Model: https://huggingface.co/EleutherAI/gpt-neo-125M
- Python Official Documentation: https://docs.python.org/3/
- CSS Styling for Streamlit: Community discussions on https://discuss.streamlit.io/