

StoryCrafter - An AI Story Generator

Phase 3: Final Report and Submission

1. Project Title:

StoryCrafter - An AI Story Generator

2. Summary of Work Done

Phase 1 – Proposal and Idea Submission (10 Marks):

In this phase, we identified the problem of generating context-aware text and proposed the idea of building **StoryCrafter**, an AI-based story generation system using **Generative AI**. The key objectives were:

- To understand how large language models generate creative text.
- To use pre-trained models like GPT-2 for storytelling.
- To create an interactive web interface for users to enter prompts and receive AI-generated stories.

A complete proposal was submitted, including the problem definition, objectives, tools to be used, and expected outcomes.

Phase 2 – Execution and Demonstration (15 Marks):

We implemented the project using **Python**, **HuggingFace Transformers**, and **Streamlit**. The following major steps were completed:

- Created a responsive web interface using Streamlit.
- Integrated GPT-2 via HuggingFace Transformers for generating story continuations.
- Allowed user inputs for character and setting and generated full stories based on them.
- Tested the app across different scenarios for coherence and creativity.

3. GitHub Repository Link

You can access the complete codebase, documentation and README here:

 **GitHub Repository** - <https://github.com/CaptainCode2024/StoryCrafter>

4. Testing Phase

4.1 Testing Strategy

The application was tested thoroughly to ensure reliability, creativity, and performance. Key focus areas:

- **Input Handling:** Support for varied sentence prompts and formats.
- **Contextual Coherence:** Ensuring that outputs follow the prompt logically.
- **Edge Case Testing:** Observing behavior for nonsensical or minimal inputs.

4.2 Types of Testing Conducted

1. Unit Testing:

- Functions like prompt building, model generation, and text processing were tested individually.

2. Integration Testing:

- Verified seamless integration between Streamlit interface and the model backend.

3. User Testing:

- Feedback collected from peers who tested the app helped improve clarity and usability.

4. Performance Testing:

- Checked output delays for varying token lengths to ensure responsiveness.

4.3 Results

- **Accuracy & Relevance:**
 - Generated stories were relevant, creative, and engaging for diverse prompts.
 - *Example:* Input: "Luna, a young girl in a forgotten village" → Output: "She discovered a magical stone buried beneath the willow tree that changed her life forever."
- **Performance:**
 - Near-instant generation of stories for inputs under 150 tokens.
- **Robustness:**
 - Even with nonsense inputs, the model handled gracefully without crashing.

5. Future Work

- **Model Fine-tuning:**
 - Train GPT-2 on story-specific datasets (e.g., fairy tales or sci-fi) for domain consistency.
- **Multi-Modal Expansion:**
 - Integrate image-based prompts to enhance storytelling using vision-language models.
- **Collaboration Mode:**
 - Allow real-time co-writing features for multiple users to build stories together.
- **Feedback Mechanism:**
 - Add like/dislike ratings for stories to refine model responses over time.
- **Multi-language Support:**
 - Expand support for non-English stories using multilingual transformer models.

6. Conclusion

StoryCrafter successfully demonstrates how Generative AI can be used to generate rich, imaginative narratives based on user inputs. From proposal to prototype, we explored the potential of LLMs like GPT-2 for creative writing assistance. The project highlights the capabilities of transformer models in enhancing digital storytelling and sets the foundation for advanced features in the future.