**Name – Vaibhav Jindal**

**Registration No. – 22BCE10067**

# GEN AI PROJECT PHASE 3 SUBMISSION DOCUMENT

## Phase 3: Final Report and Submission

### 1. Project Title:

AI Poetry Generator

### 2. Summary of Work Done

### Phase 1 – Proposal and Idea Submission:

In this phase, we identified the problem statement and proposed the idea of developing an AI Poetry Generator using Generative AI models. The primary objective was to create an application that can craft expressive, structured, and thematically rich poems based on user-defined themes, emotions, styles, and forms. The objectives defined during this phase included:

- Understanding how generative models can be applied to creative text generation, specifically poetry.
- Leveraging pre-trained models to generate coherent and stylistically appropriate poetic structures.
- Building an interactive interface to allow users to specify parameters like theme, emotion, and style for personalized poem generation.

We submitted a detailed proposal outlining the problem definition, objectives, tools required, and expected outcomes.

### Phase 2 – Execution and Demonstration:

In the second phase, we implemented the proposed solution using Python, the HuggingFace Transformers library, and Streamlit. The following tasks were accomplished:

- Developed a web-based interface using Streamlit for user interaction.
- Loaded a pre-trained GPT-based model capable of creative text generation.
- Designed an interface to allow users to input themes, emotions, and styles to generate poems.
- Set up the application to process user inputs, pass them to the model, and display the generated poetry in real-time.
- Tested the model for various poetic forms and themes to ensure stylistic coherence and emotional impact.

Sample outputs, the complete code, and UI demonstrations were documented and submitted.

### 3. GitHub Repository Link

You can access the complete codebase, README instructions, and related resources at the following GitHub link:

**🔗 GitHub Repository – AI Poetry Generator**

https://github.com/Vaibhavjindal23/IBM-Project

## 4. Testing Phase

### 4.1 Testing Strategy

The system was tested across multiple scenarios to validate its robustness and artistic coherence. Both **manual testing** and **automated testing** methods were applied to ensure quality and reliability. The testing strategy covered:

- **Input Handling:** Verified the system's ability to accept different types of input (e.g., short phrases, complex themes, and emotional tones).
- **Poetic Coherence:** Ensured that the generated poetry is stylistically appropriate and maintains thematic consistency.
- **Edge Case Testing:** Tested the model with rare themes, abstract emotions, and unconventional styles to observe how creatively the model responds.

### 4.2 Types of Testing Conducted

1. **Unit Testing**
    - Each function and module (e.g., poem generation logic, UI components, and API requests) was tested individually for correctness and reliability.
2. **Integration Testing**
    - The integration of the GPT-based model with the Streamlit interface was tested to ensure smooth user interaction and real-time poetry display.
3. **User Testing**
    - A group of test users interacted with the system to evaluate its ease of use, creative quality, and interface design. Feedback was collected to improve user experience.
4. **Performance Testing**
    - The application was tested with various input complexities to observe processing time and generation speed, ensuring smooth operation during text generation.

### 4.3 Results

- **Artistic Quality:** The system generated stylistically appropriate and coherent poems across different themes and styles.
- **Response Time:** The application performed efficiently, generating complete poems within seconds of user input.
- **Edge Cases:** For abstract or surreal prompts, the system still maintained a poetic structure, demonstrating its creative flexibility.

## 5. Future Work

Although the AI Poetry Generator is fully functional, there are several enhancements planned for the future:

1. **Model Fine-tuning:**
    - Fine-tuning the pre-trained model on a corpus of classical and contemporary poetry to improve stylistic accuracy and thematic richness.
2. **Expansion to Multi-Modal Poetry:**

- Integrating image prompts to generate poems that reflect visual themes or moods, extending its application to art galleries or creative workshops.

3. **Collaborative Poetry Creation:**
   - Adding a collaborative interface where multiple users can co-create and refine poetry in real-time.
4. **User Feedback Loop:**
   - Implementing a feedback mechanism for users to rate and refine generated poems, which can be used to fine-tune future outputs.
5. **Multi-Language Support:**
   - Extending the model's capabilities to generate poems in multiple languages to broaden its usability.

## 6. Conclusion

The AI Poetry Generator project successfully demonstrates how Generative AI can be used to create expressive and structured poetry. Throughout the three phases, the project evolved from conceptualization to full implementation, showcasing the application of transformer-based models in creative text generation. This project highlights the potential for AI in literature and artistic expression, opening up new possibilities in automated poetry and creative writing.