College Name: VIT Bhopal University
Student Name: Ishan Peshkar

Sample -GEN AI PROJECT PHASE 1 SUBMISSION DOCUMENT

Phase 1: Proposal & Idea Submission

1. Project Title:

AI-Powered Poem Generator

2. Domain:

Generative AI | Natural Language Processing (NLP) | Creative Text Generation

3. Problem Statement:

Creative writing, particularly poetry, can be a challenging and time-consuming endeavor. Many individuals seek inspiration or assistance in crafting poems for personal expression, educational purposes, or content creation. There is a need for accessible tools that can quickly generate poetic text based on user-defined themes, democratizing the creative process and offering a starting point for poetic exploration.

4. Proposed Solution:

This project will implement an AI-Powered Poem Generator using pre-trained Generative AI models. The system will:

- Take a user-provided topic or theme as input.

- Utilize a large language model (LLM) from the Hugging Face transformers library (e.g., GPT-2) to generate a short, creative poem.

- Provide parameters for users to influence the generated output, such as desired poem length or "creativity" level (temperature).

- Offer a simple, interactive interface for users to input topics and view the generated poems.

College Name: VIT Bhopal University
Student Name: Ishan Peshkar

5. Objectives:

- To develop a functional prototype of an AI Poem Generator that produces coherent and thematically relevant poetry.

- To leverage pre-trained language models (e.g., GPT-2) via the Hugging Face transformers library for poem generation.

- To create an intuitive user interface (initially CLI, potentially extended to a web interface using Streamlit) for easy interaction.

- To experiment with different prompt engineering techniques and generation parameters (e.g., max_length, temperature, num_beams) to optimize poem quality.

6. Expected Outcome:

- A working Python application that can successfully generate short poems based on user-specified topics.

- A command-line interface (CLI) for basic interaction, with the potential for a simple web-based demonstration (e.g., using Streamlit).

- An understanding of the capabilities and limitations of pre-trained models for creative text generation tasks like poetry.

- A documented codebase and a report summarizing the project's development and findings.

7. Tools & Technologies to be Used:
- **Programming Language:** Python (Primary)

- **Key Libraries:**

  - Hugging Face transformers (for accessing and using pre-trained models)

  - PyTorch (as a backend for transformers)

  - Streamlit (for the optional web interface)

- **Pre-trained Models:**

  - GPT-2 (initial focus due to its balance of performance and resource requirements)

- o Potentially gpt2-medium or other similar models for experimentation.

- **Development Environment:**

    - o Local Python environment with venv for dependency management.

    - o Text editor or IDE (e.g., VS Code).

    - o Jupyter Notebook (for initial experimentation with prompts and models).

- **Version Control:** Git & GitHub (for code management and portfolio showcasing).

8. References:

- Hugging Face Transformers
  Documentation: https://huggingface.co/docs/transformers

- OpenAI GPT-2 Blog/Paper: https://openai.com/research/better-language-models

- Streamlit Documentation: https://docs.streamlit.io/

- Various examples and tutorials on creative text generation using LLMs.