ComicCrafter AI is a generative AI based comic generator running locally on edge devices that generates a comic style story based on the input prompts given by the user.

Pre-requisites

1. Knowledge of Large Language Models (LLMs): Understanding of how LLMs work, including prompt engineering and deploying with hardware acceleration on local computing devices.
2. Familiarity with Image Generation Tools: Experience with tools like Stable Diffusion, or similar platforms for generating images which can be deployed on the edge via opensource projects like comfy UI or Intel AI Playground.
3. Programming Skills: Proficiency in Python or other relevant programming languages.
4. Edge Computing: Basic knowledge of deploying AI models on edge devices.
5. App Development: Experience in developing applications, preferably with a focus on integrating multiple AI services.

Problem Description

The objective of this project is to develop an application that generates a comic book-style short story based on a user-provided prompt. The story will be divided into four parts: introduction, storyline, climax, and moral of the story. The project will be executed in four phases:

1. Phase 1: LLM Story Generation using Prompting
    - Develop a module that uses LLMs to generate a coherent story based on the user's prompt.
    - The story should be divided into four distinct parts: introduction, storyline, climax, and moral.
2. Phase 2: Image Generation
    - Create a module that generates images corresponding to each part of the story using AI-based image generation tools.
    - Ensure that the images align with the narrative.
3. Phase 3: Merging Story Prompts and Images
    - Develop a system to merge the generated text and images into a cohesive comic book format.
    - Ensure that the text and images are appropriately aligned and formatted.
4. Phase 4: Integration into an App

- Integrate the story generation, image generation, and merging modules into a single web application.
- Ensure the app is user-friendly and can run efficiently on edge devices.

Outcomes Expected

1. Functional Application: A fully functional app that can generate comic book-style stories based on user prompts.
2. User Engagement: An engaging user interface that allows users to input prompts and view the generated comic book.
3. Edge Deployment: Successful deployment of the application on edge devices, ensuring efficient performance.

Challenges Involved

1. Story Coherence: Ensuring that the LLM generates a coherent and engaging story based on the user's prompt.
2. Image Relevance: Generating images that accurately represent the story and are in a comic book style.
3. Integration: Seamlessly integrating the text and images into a cohesive comic book format.
4. Edge Deployment: Optimizing the application to run efficiently on edge devices with limited resources.

Tools & Resources to be Used

1. LLMs: Use latest open weight models e.g Llama, Deepseek, Mistral etc, evaluate the best model for this requirement.
2. Image Generation Tools: Use open weights Stable Diffusion, or similar platforms for generating images which can be deployed on the edge via opensource projects like comfy UI or Intel AI Playground.
3. Programming Languages: Python for developing the modules.
4. Edge Devices: Intel-based edge devices for deploying the application.
5. App Development Frameworks: Use python based web app framework like streamlit for integration the all parts in to a easy to use web UI.

Submission Format

A fully open-source GitHub repo consisting of the following:

1. Project Report: A detailed report covering the problem statement, methodology, challenges faced, and solutions implemented.
2. Source Code: Well-documented source code for all modules and the integrated application.
3. Demo Video: A video demonstrating the functionality of the application.
4. User Manual: A manual explaining how to use the application, including installation and usage instructions.