**1. Introduction**

**1.1 Project Overview**

AI Storyboarder generates visual storyboards from text prompts. Users input a short story (e.g., "A dragon attacks a village"), and the system outputs three scenes with AI-generated images and captions. It’s built for quick visualization in storytelling, game design, or film pre-production.

**1.2 Purpose**

This document provides a technical guide for setting up, understanding, and extending AI Storyboarder. It covers the tech stack, workflow, installation, and improvement options.

**2. Tech Stack**

**2.1 Tools**

* **OpenAI API**: GPT-4 splits text into scenes; DALL·E 3 generates images.
* **Flask**: Python backend for API handling.
* **JavaScript (Fetch API)**: Frontend logic for API calls and UI updates.
* **HTML/CSS**: Basic interface structure and styling.

**2.2 AI Usage**

* **Text Processing**: GPT-4 divides input into three logical scenes.
* **Image Generation**: DALL·E 3 creates images from scene descriptions.
* **Scene Ordering**: Ensures scenes display sequentially (1-2-3).

**3. How It Works**

**3.1 Workflow**

1. **Input**: User enters a story (e.g., "A pirate finds treasure").
2. **Scene Split**: GPT-4 generates three scenes: "Pirate sails," "Finds chest," "Holds gold."
3. **Image Creation**: DALL·E 3 produces an image per scene.
4. **Output**: Scenes display horizontally with images and captions.
5. **Download**: Users can save images as PNGs.

**3.2 Code Structure**

* **app.py**: Flask backend for API calls and processing.
* **script.js**: Handles frontend Fetch requests and UI rendering.
* **index.html**: Input field, buttons, and output layout.
* **styles.css**: Minimal styling for alignment and readability.

**4. Setup and Installation**

**4.1 Requirements**

* Python 3.8+
* Flask (pip install flask)
* OpenAI API Key
* Browser (Chrome/Firefox/Edge)

**4.2 Steps**

1. Clone repo:

*git clone https://github.com/arthaix/AI-Storyboarder.git*

*cd AI-Storyboarder*

1. Install dependencies:

*pip install flask openai*

1. Set API key:

*export OPENAI\_API\_KEY="youropenaikey"*

1. Run Flask:

*python app.py*

1. Open index.html in a browser or visit http://127.0.0.1:5000/.

**5. Possible Improvements**

**5.1 Features**

* Enable cloud storage (e.g., Google Drive).
* Support more scenes (5+).
* Add multi-language input.

**5.2 Optimizations**

* Reduce API latency with async calls.
* Cache images locally.
* Improve error messages in UI.

**6. Conclusion**

AI Storyboarder uses GPT-4 and DALL·E 3 to turn text into visual storyboards. It’s a practical tool for students and developers to explore AI and web development. This guide provides the essentials to build and enhance it. Next steps: tweak features or deploy it.