📚 Project Documentation: AI Comic Factory

🧩 Problem Statement

Creating illustrated comics typically requires collaboration between writers and artists, which is time-consuming and expensive. For many creators, especially individual storytellers or educators, this poses a significant barrier.

Goal:
Develop an AI-powered web application that automates the comic creation process—from generating a story outline to creating consistent visuals and compiling everything into a downloadable comic PDF.

🚀 Project Overview

AI Comic Factory is a Streamlit-based web app that:

- Takes a user-defined story idea
- Automatically generates a story with a protagonist and multiple scenes
- Creates consistent character and scene illustrations using Stable Diffusion
- Exports the result as a printable PDF

🛠️ Tech Stack

Component	**Technology / Library**
Frontend	Streamlit
Image Generation	🤗 HuggingFace diffusers (Stable Diffusion v1.5)
Backend Logic	Python
PDF Export	reportlab, PIL
Optimization	CUDA, Torch with memory efficiency options

📂 Project Structure.

```
├──── app.py              # Main Streamlit application
├──── image_generator.py     # Handles AI-based character/scene generation
├──── story_generator.py    # Generates story outline (not shown above)
├──── pdf_export.py        # Converts story and images to comic-style PDF
├──── requirements.txt      # Python dependencies
```

📋 How It Works

1. User Input

Users enter a story premise and customize settings via the sidebar:

- Low Memory Mode
- Image Strength
- Generation Steps
- Scenes per Batch

2. Story Generation

A story is generated with:

- Title, Genre
- Main Character (name, appearance, personality, special traits)
- 5 Scene Descriptions

Fallback story generation is available if the model fails.

3. Character Illustration

Stable Diffusion is used to create a high-quality character image based on the main character's description.

4. Scene Generation

For each scene, the app:

- Uses the character image as a base
- Applies img2img technique with memory-optimized Stable Diffusion
- Generates scene visuals consistent with the character

5. PDF Creation

All scenes and story details are compiled into a PDF using pdf_export.create_comic_pdf.

🧠 AI & Memory Optimization

Techniques Used:

- fp16 model loading for lower VRAM usage
- CUDA settings: max_split_size_mb, xformers (if available)
- Explicit garbage collection (gc.collect())
- Manual cache clearing: torch.cuda.empty_cache()

These ensure the app can run on systems with limited GPU memory.

🧪 UI Features

Main Sections:

- Sidebar Inputs: User preferences and "Generate Comic" button
- Loading States: Step-by-step progress with logging
- Results Display:
  - Character Sheet
  - Scene-by-scene comic with descriptions
- Export Options:
  - PDF download
  - Individual PNG downloads (character + scenes)

📦 Session State

The app uses st.session_state to:

- Cache the ImageGenerator instance
- Store generation results (story, character, images, PDF)
- Prevent redundant model loading and preserve generated content across reruns

🔒 Error Handling

- Safe fallback for story generation if an error occurs
- Try/Except blocks around model initialization and generation
- Logging for debugging and transparency
- Graceful cleanup and recovery if GPU memory runs out

✅ Usage Example

Input:
A robot detective solves mysteries in a futuristic city

Output:

- 👦 Character: Sleek metallic detective with glowing eyes

- 🖼️ Scenes: Alleyway mystery, lab investigation, rooftop chase, etc.
- 📄 PDF: Downloadable, print-ready comic

📥 Installation Instructions

Clone the repository:

- **git clone <repo-url>**
- **cd ai-comic-factory**

Create and activate a virtual environment:

- **python -m venv venv**
- **source venv/bin/activate**

Install dependencies:

- **pip install -r requirements.txt**

Run the app:

- **streamlit run app.py**

📌 Requirements

- Python ≥ 3.8
- CUDA-enabled GPU (recommended for performance)
- diffusers, transformers, torch, PIL, reportlab, streamlit

📈 Future Enhancements

- Custom character editing and multiple characters
- Scene reordering and editing
- Multi-page PDF with cover and credits
- Story continuation & chapter support
- Audio narration or speech bubble integration