

MAS Final Project - Annabell

Parimala Ravali Mukkavilli

Project Overview

The goal of my project was to explore the power of AI in bringing storytelling to life, especially within the **horror genre**, which has always fascinated me. My story, titled “**Annabell**,” centers around **Anna**, a black cat with a mysterious silver bell around her neck, living with her owner **Ravali**. What begins as a seemingly ordinary tale of companionship slowly transforms into a chilling supernatural experience.

I chose this concept because I have a deep interest in **horror narratives**, especially ones that use ordinary settings to build suspense and fear. The combination of an innocent cat and a silent bell introduced a subtle but eerie element that I felt could be amplified using AI tools. My aim was to tell a short, unsettling story that leaves viewers with a sense of unease, demonstrating how AI can be used to generate powerful visual and auditory storytelling with minimal manual intervention.

Through this project, I wanted to not only create something engaging and atmospheric but also explore how generative AI can empower individuals to become digital creators—without needing a full film crew or expensive equipment.

Creative Process

My creative process for *Annabell* began with integrating **CrewAI**, where I designed a multi-agent system powered by GPT-4 to handle both storytelling and video production coordination. I structured the agents with clear goals and backgrounds to create a collaborative, AI-powered narrative development workflow.

Story Planning & Script Writing

To bring *Annabell* to life, I used a custom **story_writer agent** within CrewAI whose role was to act as a **Horror Story Writer Specializing in Short Films**. The agent was instructed to write a 2-minute horror script that began with a peaceful scene between a cat named Anna and her owner Ravali. The story slowly built suspense through the mysterious ringing of Anna’s bell—an event that only happens when she is possessed.

Initially, the script was generated as a full paragraph. However, I realized that to produce a compelling video with good pacing, I needed discrete scenes. I iterated my prompt multiple times, eventually updating it to output **6–10 second scenes** with cinematic detail. I also made

sure to emphasize **Anna's possession being revealed through the bell**, to enhance the eerie atmosphere.

Video Generation Workflow

I experimented with various AI video generation tools including **Veo 2, Veo 3, Kling 2.0, InVideo, and Monica**, but ultimately chose **LTX Studio** for its ability to interpret scripts effectively and generate cinematic results.

- First, **LTX Studio** converted my script into reference images that matched the mood and setting of each scene.
- Then, it transformed these stills into animated video segments, maintaining the horror tone I envisioned.
- I organized these segments into a cohesive timeline within LTX Studio.

Audio Integration

For audio, I used **MyEdit.online**, an AI-powered tool to generate ambient and suspenseful soundtracks. I crafted prompts tailored to each scene's emotion—from soft eerie tones to sudden jarring sounds.

These audio tracks were imported into **LTX Studio**, where I synchronized them with corresponding video scenes. I also finalized the **background music selection and transitions** within LTX Studio's timeline editor, ensuring that the pacing and emotion were consistent throughout the video.

AI Tools and LLMs Used

For the *Annabell* project, I used a variety of advanced AI tools and platforms that enabled me to go from concept to a complete horror short film. Each tool played a distinct role in the creative pipeline—from scriptwriting to video and audio generation.

Language Models

- **ChatGPT-4 (via CrewAI integration)**
 - Used for generating the storyline, structuring scenes, and iterating prompts.
 - Created the horror narrative tone, emotional pacing, and descriptive dialogue.

CrewAI Agents

- **story_writer agent**
 - Designed to simulate a professional horror scriptwriter.
 - Delivered a structured script with eerie buildup and emotional progression.
- **heygen_video_agent**
 - Guided the process of visualizing and formatting the video using tools like HeyGen and LTX.

Video Generation Tools

- **LTX Studio**
 - Main platform for transforming the script into visual scenes.
 - Generated initial image concepts and then animated them into video clips.
 - Served as the timeline editor for organizing scenes and syncing audio.
- **Other tools tried (but not used in final video):**
 - **Veo 2 / Veo 3** – Early testing for cinematic rendering.
 - **Kling 2.0** – Explored for realistic animation. Unable to generate using the free version available
 - **InVideo & Monica** – Tried for combining visuals and effects but lacked the cinematic control I needed.

Audio Generation Tools

- **MyEdit.online**
 - Used to generate background music and scene-specific sound effects.
 - Prompt-driven AI audio that matched the emotional tones of each scene (e.g., soft, sinister, thunderous).

- Exported audio tracks were stitched and timed precisely in LTX Studio.

Prompt Design and Evolution

Prompt design was one of the most critical aspects of the *Annabell* project. I learned that clear, structured, and emotionally-aware prompts were necessary to guide AI toward producing meaningful and engaging outputs—especially in a genre like horror, where pacing and atmosphere are essential.

Initial Prompt (Script Writing)

The first version of the prompt was designed to generate a short horror story with a beginning, middle, and end in paragraph form. I used the following high-level task with the **story_writer agent** in CrewAI:

yaml

CopyEdit

```
write_horror_script:
```

```
  description: >
```

```
    Write a 2-minute horror short film script titled 'Anna Bell'.
```

```
    The story is between Anna, a cat, and her owner.
```

```
    Anna wears a bell.
```

```
    As the story progresses, the bell begins to ring, revealing something sinister. Build suspense and end with a disturbing twist.
```

```
  expected_output: >
```

```
    A full-length short horror film script ready for video generation.
```

Result: The story came back as a single paragraph, with some emotional detail but no clear scene division.

Prompt Iteration

To make the script production-ready for video generation, I updated the prompt to enforce structured **scene-by-scene output**. I asked for:

- Scene durations between 6–10 seconds
- Visual descriptions
- Dialogues where needed
- Strong emotional cues and lighting cues (e.g., “eerie glow”, “ominous silence”)

This helped AI produce something more **cinematically coherent** and suited for tools like LTX Studio, which benefit from structured scenes.

Enhancing Symbolism

I also revised the story focus to emphasize **Anna’s bell only ringing during possession**. This change added a layer of psychological horror and symbolism. The bell became a key storytelling device, and I used prompts that highlighted moments when it jingled—subtly building tension before the final disturbing twist.

Audio Prompting

In **MyEdit.online**, I provided short, vivid prompts like:

- *“Eerie ambient drone with soft whispers”*
- *“Thunder crack with high-pitched ringing, fading into silence”*
- *“Low heartbeat sound building up with cat growl overlay”*

These cues helped generate audio that synchronized emotionally with the scenes and enriched the immersive experience.

Challenges and Lessons Learned

I approached this project in two different ways.

Make the entire project (Script, Video and Image) with CrewAI locally.

- a. The approach is define Agent for generating script with scenes from prompt
- b. Pass this scenes as prompt to a video generation agent
- c. Then pass the same audio prompts to audio generation agent
- d. Combine all the 6 second audio, video into a full video using ffmpeg.

Sadly, I got stuck on the second step of my grand plan. I tried to run a video generation model in docker on my mac and expose an endpoint that would take a prompt to generate video. I tried to integrate this api as a CustomTool in the crewAI project. Here is below of everything I tried and lessons learned.

The models I tried are T2V, **AnimateDiff** (with ComfyUI) and SVD (Stochastic Video Diffusion). I couldn't run any of these models successfully on my mac or pc. Then I tried to run these models on an EC2 instance but got blocked by their policy with resource allocation on EC2.

#	Approach	What I Tried	Challenges Faced	Lessons Learned
1	ModelScope T2V (Local Setup)	Installed and tested Alibaba's ModelScope for text-to-video using Python scripts.	Required GPU for practical performance. CPU inference was extremely slow (~10+ mins per 4s clip).	Open-source models demand GPU acceleration for usable performance.
2	Dockerized Text-to-Video Pipeline	Created a Dockerfile with Python, ModelScope, and ffmpeg. Generated and stitched short clips into a 2-minute video.	Docker ran failed on M2 Mac & Windows PCs with NVIDIA GPU.	Docker doesn't support GPU passthrough on M1/M2 Mac, and Windows Docker Desktop lacks native CUDA integration without WSL.

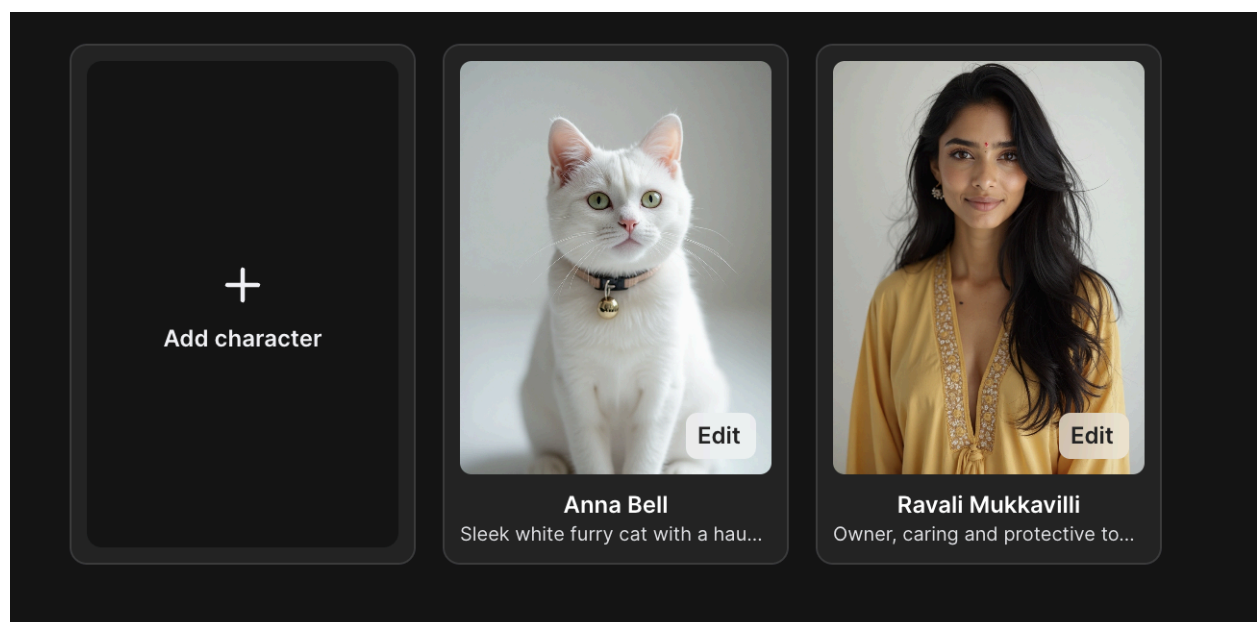
3	Video Stitching via FFmpeg	Generated 20 short clips using different prompts, then combined using <code>ffmpeg</code> .	Some clips had visual/style inconsistency. Manual reordering needed.	Best practical way to simulate longer videos from open-source models that support only short output.
4	AWS EC2 GPU Deployment	Tried launching <code>g4dn.xlarge</code> instance with NVIDIA T4 GPU to run the Docker container.	Instance launch failed due to vCPU quota = 0 for GPU instances.	AWS requires explicit quota increase for new users to run GPU workloads. Takes time.

Use existing models on internet

Retaining Context across scenes

The biggest challenge in this approach is to maintain proper context across scenes. For example, in one scene, the cat is brown. In subsequent scenes, the cat is white with a brown bell. The same with the character. Each generation generated unique characters that do not match with the previous scene. I solved this by using a feature in LTX Studio to pre-define characters, their appearance and facial expressions.

I defined these characters below to inform AI of a outline of the cast in the video.



Tool Compatibility & Limitations

One of the initial challenges I faced was choosing the right AI video generation tool. While platforms like **Veo 2**, **Veo 3**, **Kling 2.0**, and **InVideo** offered high-quality visuals, they lacked either the timeline flexibility or emotional depth I needed for a horror short film. Many tools could not interpret long scripts or lacked synchronization capabilities for audio layering. After several rounds of testing and discarded renders, I finally found **LTX Studio**, which offered a good balance of **script-based control** and **visual quality**.

Prompt Structuring

Another hurdle was getting AI to break down the story into **well-paced scenes**. Initially, the output was just a single paragraph, which didn't suit video production. I had to learn how to structure prompts in a way that guided the model to return **scene-by-scene breakdowns**, with the right emotional cues and visual details. This involved multiple iterations and fine-tuning, especially around the bell's symbolism.

Audio Timing

While **MyEdit.online** generated impressive atmospheric sounds, syncing them with the video was not automatic. I had to manually **time and layer the sound effects and background music** in LTX Studio to match the tone of each scene. This was a time-consuming process but necessary to maintain the emotional rhythm of the story.

Final Output and Links

Video URL:

https://www.linkedin.com/posts/parimala-ravali-mukkavilli_ai-horrorstory-generativeai-activity-7339515053060079616-kfdJ?utm_source=share&utm_medium=member_desktop&rcm=ACoAADlvFVsB3ydcdu1JgTPqX3XMefzTGjq2ry4

Source Code:

<https://github.com/ravali0423/annabell>

Ltx Studio

<https://app.ltx.studio/projects/b2843583-e96a-4130-bbae-b0d033a62eaf>