**1. Introduction**

* **Objective**: Explain the purpose of the project – to create a storytelling AI system capable of generating coherent and creative stories.
* **Scope**: Describe the system workflow, from dataset generation to story generation using embeddings.
* **Overview**: Briefly mention the tools and technologies used (e.g., Ollama, OpenAI API, JSON datasets).

1:Introduction.

**Objective**: The goal is to leverage generative AI models to create a system capable of taking user prompts and transforming them into rich, narrative-driven outputs. By combining advanced AI techniques, the system ensures that the generated stories are contextually relevant and tailored to user inputs while maintaining creativity and flow. This project emphasizes creating a storytelling platform that enhances the user experience by seamlessly integrating multiple AI models for high-quality story generation

**SCOPE**:The project workflow involves a multi-stage process, beginning with dataset generation and passing the dataset to a storytelling AI that dynamically creates new stories based on user prompts. The scope of the project includes:

1. **Dataset Generation**: Using locally installed Ollama models to produce an initial dataset of diverse stories. Multiple AI agents with varied configurations and prompts are employed to ensure the stories reflect a range of creative styles and themes. Different tests were done by using diffent prompts to get the most out of the model. A more detailed explanation will be done in the system architecture on the experiments to create the dataset
2. **Dataset Structuring**: The generated stories are organized into a structured format (JSON), making them accessible and reusable for downstream tasks which are for finetuning the OpenAI model and for testing phases the image generation model.
3. **Story Generation**: Utilizing embeddings derived from the dataset and integrating OpenAI’s generative models, the system creates new stories by referencing prior narratives. This step ensures that the generated content is both coherent and contextually linked to the user's input and making sure the model is tailored to create story’s and not used for other functions. Different testes where done to ensure that the AI system only creates story’s and nothing else
4. **Comparative Analysis**: Comparing outputs from different AI agents to select the best stories in terms of creativity, coherence, and relevance, to the story title created by one of the agents.

**2. Design Choices for Generative Models**

* **Ollama for Dataset Generation**

Dataset Creation: As for the creation of the dataset which will be used for finetuning the model ollama3.2 3billion parameters was chosen. Using a locally installed model has the benefits of privacy but it gave a huge understanding how to use Ollama on PyCharm and how different prompts can significantly influence the model. The stories could have easily been created using ChatGPT with a simple prompt but this method helped have a better understanding in prompt design and how to have a better understanding on how prompts effect outcomes.

Also a localy installed model meant that I could create the sotry dataset in places where I had limited connectivity which was the case since I work in a particular field and access to the internet is limited and finally a vast handon experience was learned.

For the dataset creation the idea was used to utilize AI agents but for our case we would have multiple ai agents which all has a significant role in creating the title and story. In total 12 AI agents are being used to create the dataset. The number is high because for this assignment an experiment was done two experiments,

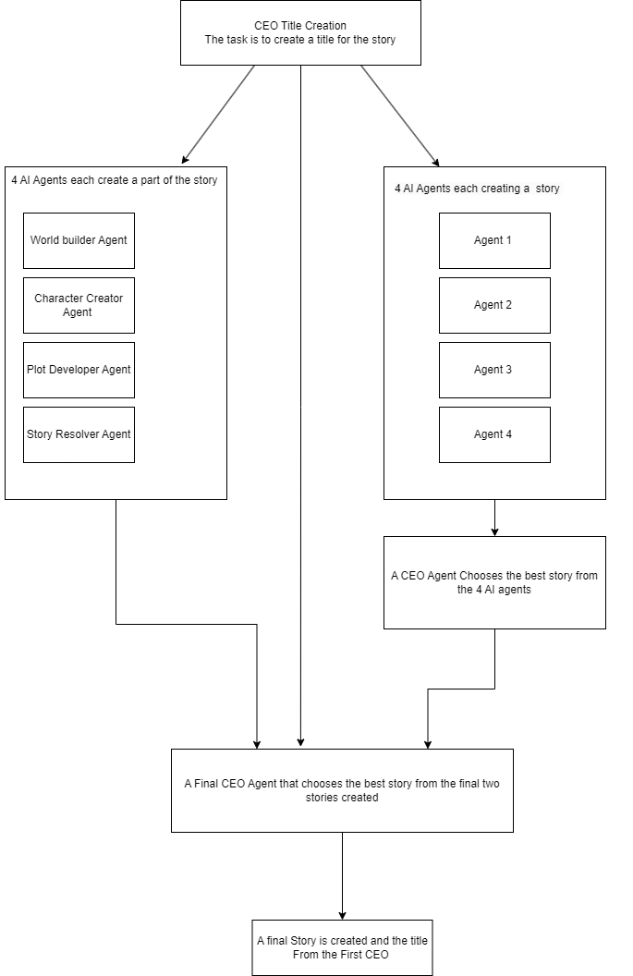
1. Split the story generation between for AI agents, and each ai agent had a specific task, such as worldbuilder, character creator, plot developer and story resolver. Each AI agent had to build a specific part of the story.
2. The other experiment was that different four AI agents would be utilized each with the task to create a story based on the title. Finally, another AI agent was delegated to choose the best story from
3. The final experiment was to have a finally AI agent acting as the CEO to choose the best story between the ai agent where the task was split up or the AI agents doing all the work on their own.

**OpenAI API for Story Generation**

Once the story json file has been created the next task to to utilize the file and using embeddings create a new story that the user has given inside the prompt. The Idea for this model is to use OPENAI gpt 3.5 turbo to do this task and using the locally run webui the user can insert the prompt and see the generated reply form the model. The OPENAI model excel’s at producing human-like narratives with minimal effort, making them ideal for storytelling tasks. The temperature for the model could also be changed making finetuning more easily. The main reason why the openai model was selected for it’s ease to allow incremental inputs which our case is the custom dataset and finally is the request time. Since hardware is a limited issue utilizing gpt 3.5 would drastically increase the generated story response time.

**3: System Architecture**

**Overall System Workflow**



**Data Generation**: As discussed in the in chapter 2 there are in total 12 AI agents that have each carefully tasked into contributing to create a final story and title.

The first AI Agent is called the Creative CEO and the task of this agent is to create a compelling story title from a randomly selected theme choice. The list of theme choices is a list of 36 different themes varying from space exploration to psychological thrillers and parallel universe. The CEO title creator has a very specific prompt and set of guidelines when creating the title. The first guideline is that a set of words can not be used when creating the title and this is done because in testing phase it was seen that the model uses a lot of similar words when creating the story titles such as Echoes of, The Last, Beyond the. It was seen that most titles started this way hence in the prompt it was guided not to use titles starting with these phrases.

The second prompt was that the titles need to be limited to not more than 7 words, ideally between 3-7 words max. The CEO was instructed to create strong emotion and vivid story titles and finally since a large dataset was being created the model needed to know all the story titles that were created as not to have any duplicated story titles, therefore each time a title is created it is stored in an array that the model has access to each time it tries to generate a title. This has been implemented after seeing that at around 50 story titles there where roughly each time 5 story titles that are the same.

**The next four AI agents** work collaboratively to create a complete and cohesive story based on a title generated by the CEO Story Title Generator. This approach divides the storytelling process into four specialized tasks, allowing each AI agent to focus on a specific aspect of narrative development, ensuring the final product is both creative and coherent.

1. World Builder: The first AI agent, nicknamed *World Builder*, is tasked with creating the setting and atmosphere for the story. This agent focuses on crafting a vivid and immersive environment that aligns with the title. By dedicating attention to world-building, the agent lays the foundation for the story’s tone and mood in a concise, 25-word description.
2. Character Creator: The second AI agent, *Character Creator*, is responsible for developing the characters who drive the story forward. This agent ensures that the characters are unique, relatable, and gender-neutral to avoid bias and enhance inclusivity. In exactly 25 words, this agent introduces the characters, emphasizing their personalities and conflicts, contributing to the emotional depth of the narrative.
3. Plot Developer: The third AI agent, *Plot Developer*, handles the main action of the story. This includes defining the central conflict and key events that propel the narrative. By focusing on the storyline's core dynamics, the agent creates tension and intrigue in 25 words, ensuring that the narrative stays engaging and purposeful.
4. Story Resolver: The fourth and final AI agent, *Story Resolver*, resolves the conflicts and concludes the story. This agent’s role is to tie up loose ends, delivering a meaningful resolution that aligns with the preceding narrative and the title’s theme. Its concise 25-word conclusion ensures the story feels complete and satisfying.

Together, these four AI agents form a collaborative framework that efficiently distributes the workload, ensuring that every aspect of the story is thoughtfully developed. Each agent’s specialized focus enhances the narrative's overall quality and creativity, creating a seamless and engaging storytelling experience.

**The next set of four AI agents** where designed to create whole stories independently, each with unique prompts yet with the same overarching goal: crafting vivid, imaginative narratives based on the given title and theme. Each agent operates autonomously to deliver a complete story, focusing on engaging world-building, gender-neutral characters, and a cohesive narrative arc. These agents were the most challenging to train due to their dual responsibility of producing a self-contained story while adhering strictly to the goal of generating only the story itself, with no additional explanations or justifications.

To achieve this, the prompts were meticulously refined to eliminate ambiguity, ensuring the agents stayed focused on storytelling without veering into analysis or reasoning. Each prompt was tailored to guide the agents in creating immersive settings, rich descriptions, and emotionally resonant character actions. The iterative refinement process was key to resolving early issues, such as overly verbose outputs or narratives lacking depth, by enforcing constraints like word count and a single-paragraph structure.

A critical feature of these agents is their adaptability to diverse themes. Regardless of whether the theme is "space exploration," "ancient myths," or "urban legends," the agents consistently generate stories that align with the intended mood and style of the theme. This adaptability ensures a wide range of creative outputs that remain true to the prompts’ directives.

Additionally, the prompts were designed to encourage variety across the agents. By focusing on different aspects of storytelling—such as emotional depth, narrative flow, or descriptive imagery—the agents avoid redundancy, producing unique interpretations of the same title and theme. This ensures that while the agents share a common goal, each story feels distinct, showcasing the creative breadth of the AI.

Finally, the process of training these agents highlighted the importance of balance: maintaining creative freedom while ensuring structure and coherence. The result is a team of storytelling agents capable of delivering rich, compelling narratives tailored to specific themes and titles, offering a fresh perspective with every story they create.

To explain the role of the CEO in choosing the best story from the four independently created by the AI agents, you can write something like this:

The **CEO Story Selector** represents an expert decision-making AI tasked with evaluating and selecting the best story among four independently crafted narratives by AI agents. This AI embodies the critical perspective of a publishing CEO, combining expertise in storytelling with a focus on market viability and creative excellence. Its purpose is to ensure that the chosen story aligns seamlessly with the provided title, showcasing the highest quality and narrative coherence.

The next AI agent is another CEO choosing agent that selects the best story from the 4 individual AI agents. This selection process is designed to simulate the real-world decision-making of a publishing house CEO. The CEO evaluates the four stories based solely on their alignment with the given title, ensuring that the chosen narrative is the most compelling, imaginative, and thematically appropriate. By focusing exclusively on the story quality and its connection to the title, the CEO AI avoids bias and guarantees that the most impactful story is selected.

The **decision criteria** include:

1. **Relevance to the Title**: How well the story reflects the essence and mood of the title.
2. **Creativity and Originality**: The uniqueness of the story’s concept, world, characters, and narrative style.
3. **Narrative Quality**: The coherence, structure, and emotional resonance of the story.
4. **Thematic Consistency**: How well the story adheres to the theme implied by the title.
5. **Engagement**: The ability of the story to captivate and hold the reader’s attention.

The prompt for the CEO AI is specifically structured to ensure impartiality and focus. By requesting the AI to consider only the title and the stories themselves, extraneous factors are excluded, enabling an objective and fair evaluation.

This final layer of decision-making adds a level of quality control to the collaborative efforts of the storytelling agents, ensuring that the output meets the highest standards of creativity and narrative integrity. Through this process, the CEO AI demonstrates how technology can emulate professional judgment in creative industries.

**The Final CEO Decision** represents the ultimate stage in the story selection process, where the CEO AI compares the collaboratively created story (the team effort) against the best story chosen from the four independently crafted narratives. This ensures that the most compelling and impactful story is selected for publication, combining the strengths of collaborative creativity with the individuality of independent storytelling.

In this phase, the CEO AI functions as a discerning evaluator with expertise in identifying stories that align closely with the provided title and theme while resonating deeply with the target audience. The final decision process involves assessing two stories on key parameters such as:

1. **Alignment with the Title and Theme**: How well each story reflects the essence of the given title and the nuances of the selected theme.
2. **Narrative Strength**: The coherence, structure, and emotional resonance of the story.
3. **Creative Merit**: The originality and imaginative quality of the world-building, characters, and plot.
4. **Market Potential**: Which story is likely to captivate and engage readers, ensuring commercial success.

The CEO AI evaluates these two stories without bias or external influence, considering only the intrinsic qualities of each narrative. By following a structured prompt, the CEO AI ensures impartiality and focus, delivering a single, definitive output: the best story, ready for publication.

This final decision-making layer underscores the collaborative synergy and creative potential of AI storytelling while ensuring that the chosen story meets professional standards of excellence and audience appeal. It highlights the ability of AI to not only generate content but also emulate expert-level judgment in the creative process.

Once the final story has been selected by the final ceo the title and the best story are saved in a dataset/jsonfile so the sotry can be used by gpt3.5 turbo as training data from which the story is created based on the prompt of the user.

* 1. Detail how Ollama and OpenAI were integrated into the pipeline.

Once the dataset which in our case is the json file is created the next task is to use the data set as training data for gpt 3.5 turbo to create its story’s. The model works by using embedings which are numerical representation of data that capture the semantic meaning of words, sentences, or even larger text structures. In the context of GPT-3.5 Turbo, embeddings are the foundation for understanding and processing the training data to create coherent and contextually appropriate stories.

The JSON dataset serves as the **training corpus**. It contains structured examples of story elements, such as settings, characters, plot points, and conclusions, organized in a way that GPT-3.5 Turbo can learn patterns and relationships. When generating a story, GPT-3.5 Turbo draws on the embeddings to predict the most likely sequence of words or sentences, ensuring that the output aligns with the provided input while maintaining creativity and coherence.

Code explanation to integrate GPT 3.5 turbo and the json/traing file and final webpage for the user to enter a prompt and display the image

The code implements a web application using Streamlit that allows users to generate short stories based on custom prompts with the help of OpenAI's GPT-3.5 Turbo model. The application starts by initializing the page with a title and description using the setup\_page() function. Users are provided a text input field to enter their story prompt. When the "Generate Story" button is clicked, the application executes the handle\_story\_generation() function, which processes the user's input and sends it to the generate\_story() function. This function communicates with OpenAI's API by defining a system-level instruction for the model to act as a "creative storyteller" and a user-level instruction to craft a story based on the provided prompt. The model generates a response, which is then processed to extract the story title and content.

The generated story is stored in a JSON file, enabling persistent storage and retrieval of past stories through the load\_stories() and save\_stories() functions. These functions handle file operations, ensuring that stories are either loaded into memory or saved back to the file for future use. The app displays the generated story, including the title and content, on the Streamlit interface, accompanied by a visual image for added aesthetic appeal. If the user doesn't provide a valid prompt, the app displays an error message prompting the user to enter appropriate input.

The application maintains a clean and user-friendly interface while leveraging the power of GPT-3.5 Turbo to generate creative content. By splitting tasks into modular functions and integrating OpenAI's language model, the app ensures a seamless workflow for generating, storing, and presenting stories. The functionality to persist stories across sessions using the JSON file further enhances the application's usability, while the integration of a print statement allows developers to debug or view generated stories directly in the console. The code achieves a balance between simplicity for users and advanced functionality for generating creative outputs.