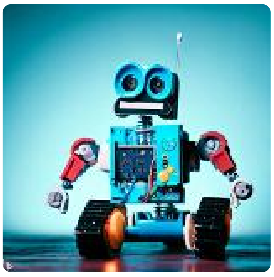
Deep dive into ChatGPT understanding

Cyril Sadovsky March 29, 2023, 9:17 PM



Throughout months of usage and studying others how they use ChatGPT I found several "tricks" how to enhance the AI. These include: providing external memory, separate concerns, priming conversations etc...

Intro: I assume GPT plugins will work according to some of the points I provided. Especially external memory, to provide factual knowledge.

Large Language models are revolutional in that sense that suddenly even not experts in Machine learning and Software enginnering are able to use computers in a similar way like an expert would do. Let's explore these ideas.

A new type of skill is needed when talking to these machines and it's called **prompt engineering**. It can be separated into 5 key concepts.

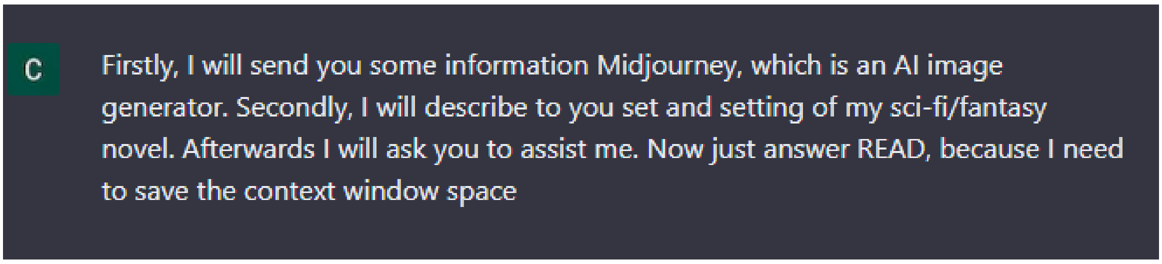
Key concepts I will be discussing:

1. Priming the conversation
2. Saving context space
3. Separate concerns
4. External memory
5. The ACT keyword

1. Priming the conversation

One of the most important keywords emerging from the GPT space is **priming**. Remember this keyword if you want to work with these types of AIs.

Let's take this example: Midjourney is an AI that generates images (https://docs.midjourney.com/). I am not good at using Midjourney, so I want to "train" or **prime** GPT-4 to understand what Midjourney is, how it is used and what images I want it to generate. How would I do it?

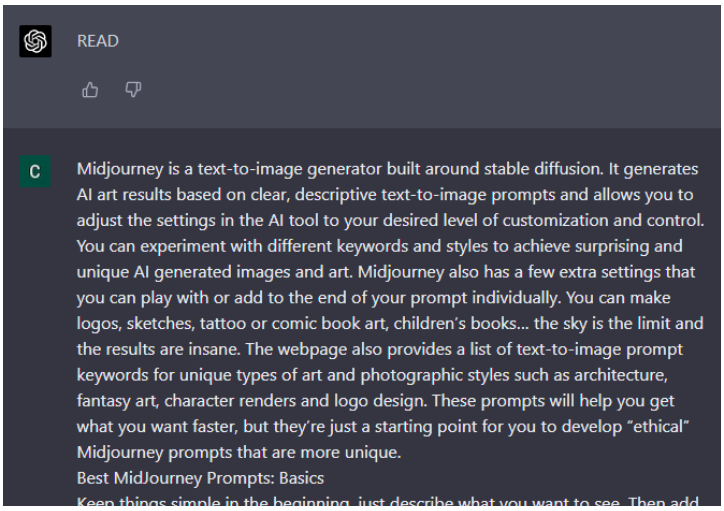


Firstly, I will send you some information Midjourney, which is an AI image generator. Secondly, I will describe to you set and setting of my sci-fi/fantasy novel. Afterwards I will ask you to assist me. Now just answer READ, because I need to save the context window space

```
Firstly, I will send you some information Midjourney, which is an AI image generator.
Secondly, I will describe to you set and setting of my sci-fi/fantasy novel. Afterwar
ds I will ask you to assist me. Now just answer READ, because I need to save the cont
ext window space
```

With my very first message to GPT-4 I explain to it what I will be doing so it can understand what my aim is. It is then less likely to go off the rails. This also helps to save context space.

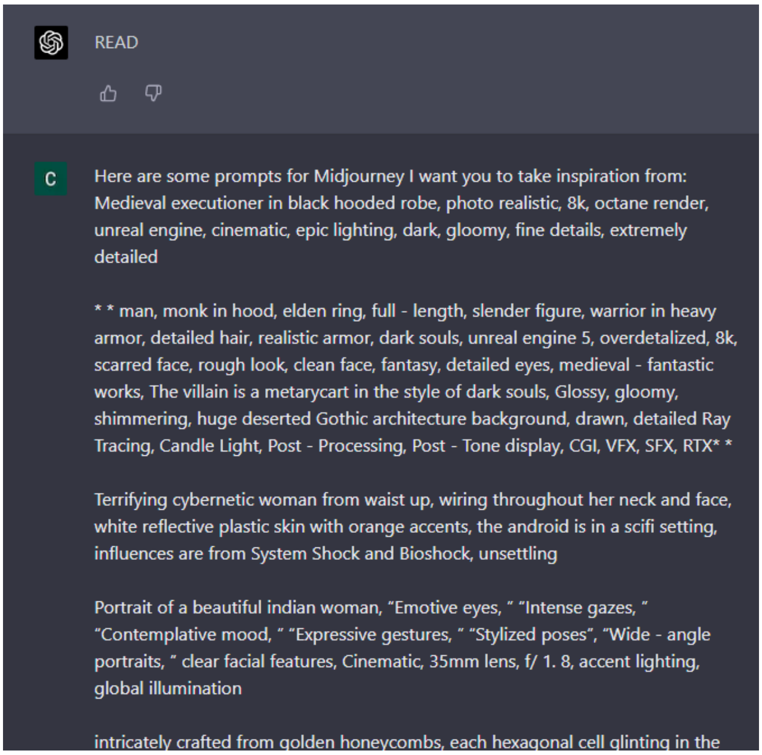
Then I pasted some Midjourney documentation:



READ

Midjourney is a text-to-image generator built around stable diffusion. It generates AI art results based on clear, descriptive text-to-image prompts and allows you to adjust the settings in the AI tool to your desired level of customization and control. You can experiment with different keywords and styles to achieve surprising and unique AI generated images and art. Midjourney also has a few extra settings that you can play with or add to the end of your prompt individually. You can make logos, sketches, tattoo or comic book art, children's books… the sky is the limit and the results are insane. The webpage also provides a list of text-to-image prompt keywords for unique types of art and photographic styles such as architecture, fantasy art, character renders and logo design. These prompts will help you get what you want faster, but they're just a starting point for you to develop "ethical" Midjourney prompts that are more unique.
Best MidJourney Prompts: Basics
Keep things simple in the beginning, just describe what you want to see. Then add

And then I provided it with some example Midjourney prompts from other users which I liked:



READ

Here are some prompts for Midjourney I want you to take inspiration from:
Medieval executioner in black hooded robe, photo realistic, 8k, octane render, unreal engine, cinematic, epic lighting, dark, gloomy, fine details, extremely detailed

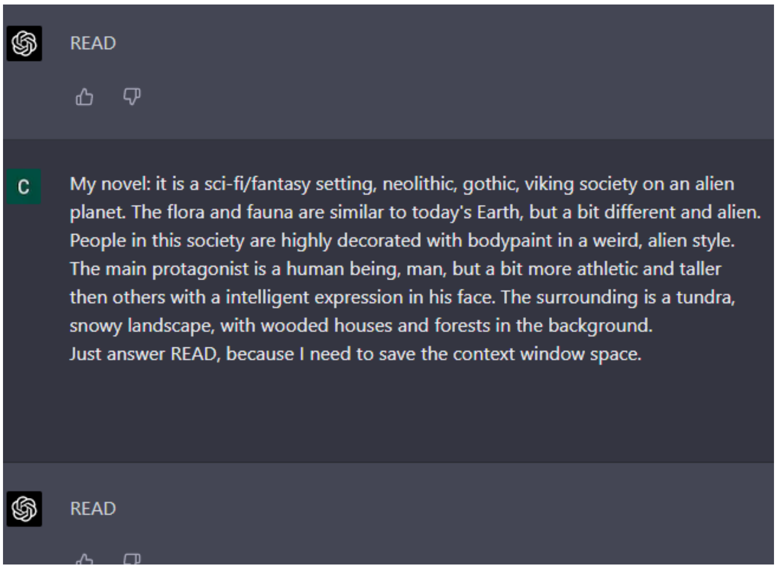
* * man, monk in hood, elden ring, full - length, slender figure, warrior in heavy armor, detailed hair, realistic armor, dark souls, unreal engine 5, overdetalized, 8k, scarred face, rough look, clean face, fantasy, detailed eyes, medieval - fantastic works, The villain is a metarycart in the style of dark souls, Glossy, gloomy, shimmering, huge deserted Gothic architecture background, drawn, detailed Ray Tracing, Candle Light, Post - Processing, Post - Tone display, CGI, VFX, SFX, RTX* *

Terrifying cybernetic woman from waist up, wiring throughout her neck and face, white reflective plastic skin with orange accents, the android is in a scifi setting, influences are from System Shock and Bioshock, unsettling

Portrait of a beautiful indian woman, "Emotive eyes, " "Intense gazes, " "Contemplative mood, " "Expressive gestures, " "Stylized poses", "Wide - angle portraits, " clear facial features, Cinematic, 35mm lens, f/ 1. 8, accent lighting, global illumination

intricately crafted from golden honeycombs, each hexagonal cell glinting in the

I added som information about my novel (for which I wanted the images created):



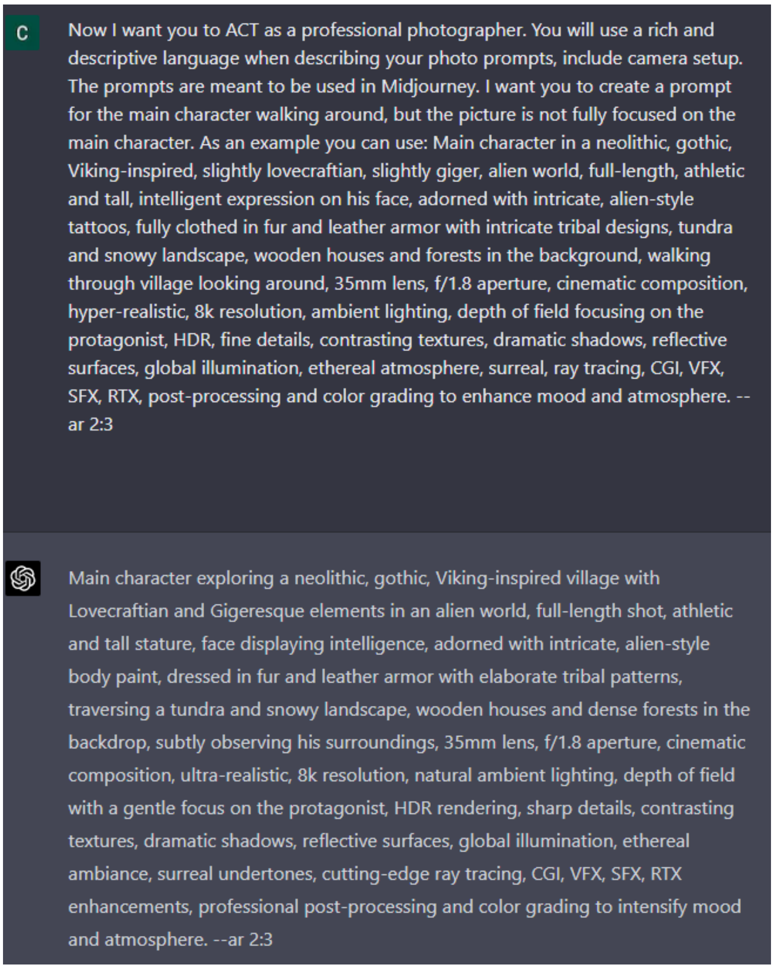
READ

My novel: it is a sci-fi/fantasy setting, neolithic, gothic, viking society on an alien planet. The flora and fauna are similar to today's Earth, but a bit different and alien. People in this society are highly decorated with bodypaint in a weird, alien style. The main protagonist is a human being, man, but a bit more athletic and taller then others with a intelligent expression in his face. The surrounding is a tundra, snowy landscape, with wooded houses and forests in the background.
Just answer READ, because I need to save the context window space.

READ

Afterwards, I was able to start generating prompts I was interested in and provide detailed instructions on what I expect:

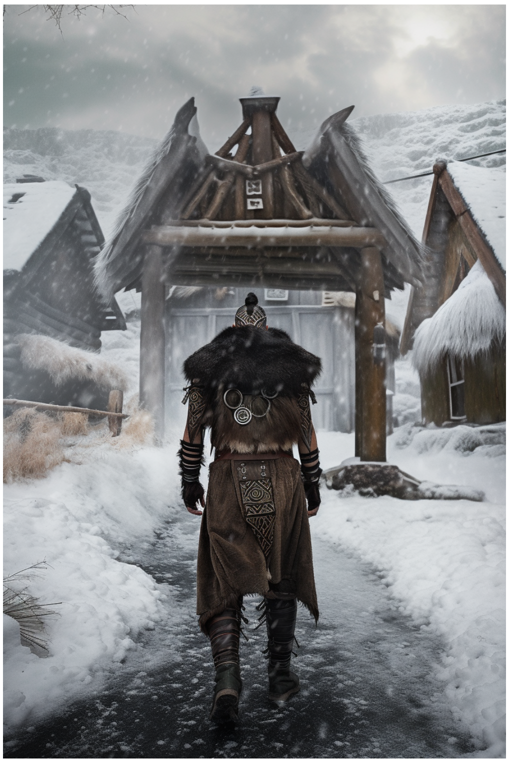
```
Now I want you to ACT as a professional photographer. You will use a rich and descrip
tive language when describing your photo prompts, include camera setup. The prompts a
re meant to be used in Midjourney. I want you to create a prompt for the main charact
er walking around, but the picture is not fully focused on the main character. As an
example you can use...
```



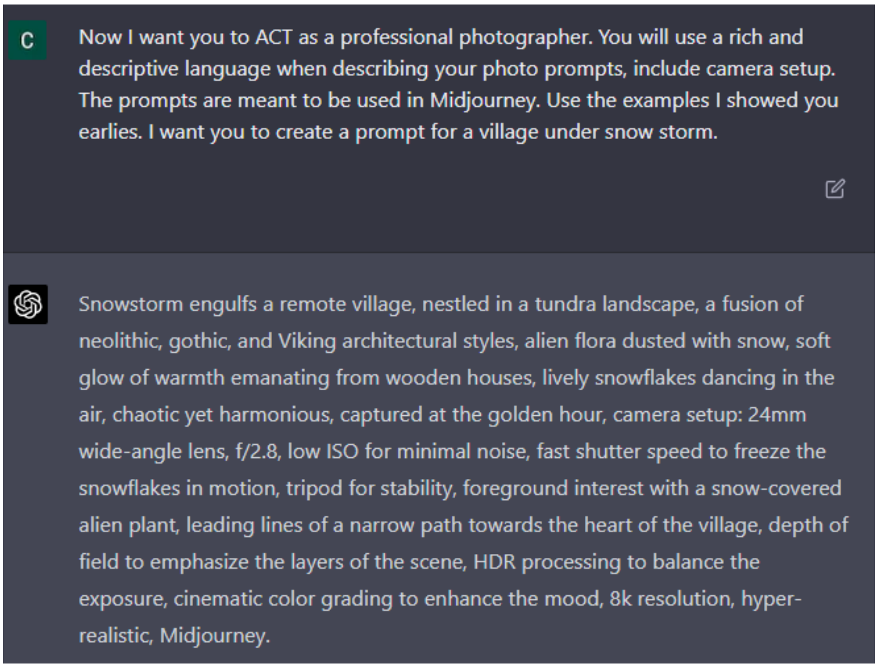
Now I want you to ACT as a professional photographer. You will use a rich and descriptive language when describing your photo prompts, include camera setup. The prompts are meant to be used in Midjourney. I want you to create a prompt for the main character walking around, but the picture is not fully focused on the main character. As an example you can use: Main character in a neolithic, gothic, Viking-inspired, slightly lovecraftian, slightly giger, alien world, full-length, athletic and tall, intelligent expression on his face, adorned with intricate, alien-style tattoos, fully clothed in fur and leather armor with intricate tribal designs, tundra and snowy landscape, wooden houses and forests in the background, walking through village looking around, 35mm lens, f/1.8 aperture, cinematic composition, hyper-realistic, 8k resolution, ambient lighting, depth of field focusing on the protagonist, HDR, fine details, contrasting textures, dramatic shadows, reflective surfaces, global illumination, ethereal atmosphere, surreal, ray tracing, CGI, VFX, SFX, RTX, post-processing and color grading to enhance mood and atmosphere. --ar 2:3

Main character exploring a neolithic, gothic, Viking-inspired village with Lovecraftian and Gigeresque elements in an alien world, full-length shot, athletic and tall stature, face displaying intelligence, adorned with intricate, alien-style body paint, dressed in fur and leather armor with elaborate tribal patterns, traversing a tundra and snowy landscape, wooden houses and dense forests in the backdrop, subtly observing his surroundings, 35mm lens, f/1.8 aperture, cinematic composition, ultra-realistic, 8k resolution, natural ambient lighting, depth of field with a gentle focus on the protagonist, HDR rendering, sharp details, contrasting textures, dramatic shadows, reflective surfaces, global illumination, ethereal ambiance, surreal undertones, cutting-edge ray tracing, CGI, VFX, SFX, RTX enhancements, professional post-processing and color grading to intensify mood and atmosphere. --ar 2:3

Result:



As the conversation is primed, you can now rewrite the message and not loose any context using the button which is under your messages when you hover over the message. For example I chaged my message:



Now I want you to ACT as a professional photographer. You will use a rich and descriptive language when describing your photo prompts, include camera setup. The prompts are meant to be used in Midjourney. Use the examples I showed you earlies. I want you to create a prompt for a village under snow storm.

Snowstorm engulfs a remote village, nestled in a tundra landscape, a fusion of neolithic, gothic, and Viking architectural styles, alien flora dusted with snow, soft glow of warmth emanating from wooden houses, lively snowflakes dancing in the air, chaotic yet harmonious, captured at the golden hour, camera setup: 24mm wide-angle lens, f/2.8, low ISO for minimal noise, fast shutter speed to freeze the snowflakes in motion, tripod for stability, foreground interest with a snow-covered alien plant, leading lines of a narrow path towards the heart of the village, depth of field to emphasize the layers of the scene, HDR processing to balance the exposure, cinematic color grading to enhance the mood, 8k resolution, hyper-realistic, Midjourney.

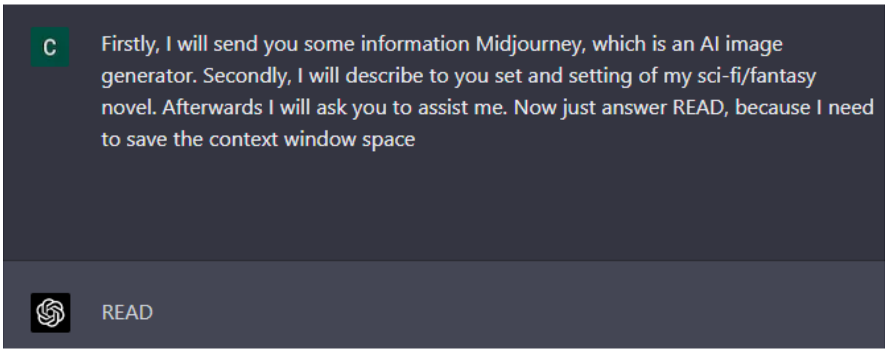
Using this technique you can quickly iterate through different prompts and find the ones that suit your need the most. As one Microsoft expert put it: *Prompt engineering is an iterative process*

2. Saving context space

Context window is a really important statistic about which many people have no information or do not know how to use it. It basically means **how much a model is able to to remember from your previous conversation.** But not only that. If the AI replies to you in lenghty fashion, **it cloggs up the context window with its own replies**! Furthermore, **it is basing its future replies on its own replies from the past**, potentially spiraling out of control if it happens to hallucinate.

The original ChatGPT was able to remember about 3.000 words. The **GPT-4 version is able to remember 25.000 words.** That already is enough to do some serious stuff if we are able utilize the whole context space.

There is an easy fix to this issue. Just use this formula at the end of your inquiry if you do not want the AI to preply: *Now just answer READ, because I need to save the context window space*



Firstly, I will send you some information Midjourney, which is an AI image generator. Secondly, I will describe to you set and setting of my sci-fi/fantasy novel. Afterwards I will ask you to assist me. Now just answer READ, because I need to save the context window space

READ

Use this trick every time you are priming the conversation and/or you don't need any answer from the AI.

3. Separate concerns

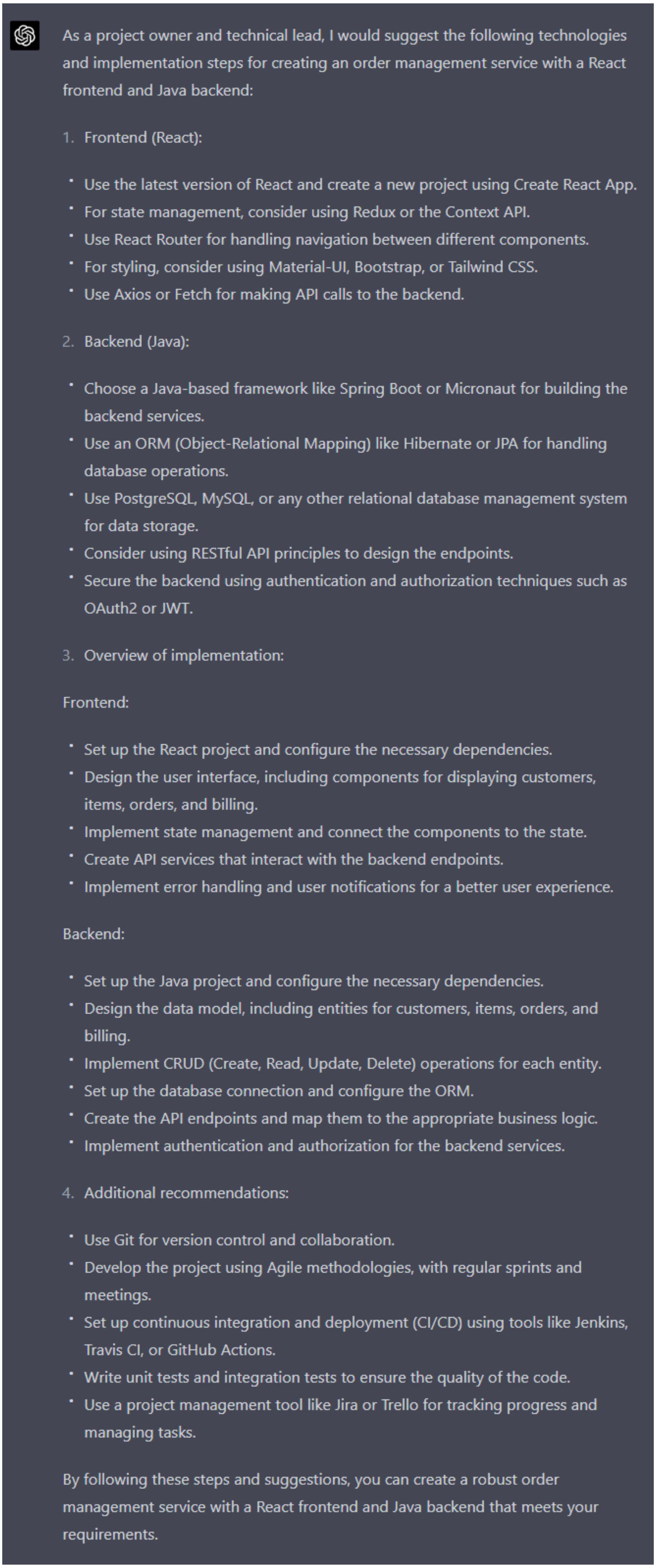
Let's say, for example, we want to create an application which consists of a backend and a frontend. The most important idea is, as the context space is limited, we should create three context windows (tabs):

- General tab where we inquire ragarding the general ideas and technologies regarding our new project
- Frontend tab

- Backend tab

Firstly, let's ask the AI about the project as a whole and technologies:

```
Me: I want you to ACT as a project owner and technical lead in one person. I want to implement an application which will consist of a frontend part using React and backend part using Java. The project in general should be an order management service, which is able to track customers, items, orders and billing. Can you give me suggestions on the technologies and an overview how to implement them which would be useful as initial steps please?
```



As a project owner and technical lead, I would suggest the following technologies and implementation steps for creating an order management service with a React frontend and Java backend:

1. Frontend (React):

- Use the latest version of React and create a new project using Create React App.
- For state management, consider using Redux or the Context API.
- Use React Router for handling navigation between different components.
- For styling, consider using Material-UI, Bootstrap, or Tailwind CSS.
- Use Axios or Fetch for making API calls to the backend.

2. Backend (Java):

- Choose a Java-based framework like Spring Boot or Micronaut for building the backend services.
- Use an ORM (Object-Relational Mapping) like Hibernate or JPA for handling database operations.
- Use PostgreSQL, MySQL, or any other relational database management system for data storage.
- Consider using RESTful API principles to design the endpoints.
- Secure the backend using authentication and authorization techniques such as OAuth2 or JWT.

3. Overview of implementation:

Frontend:

- Set up the React project and configure the necessary dependencies.
- Design the user interface, including components for displaying customers, items, orders, and billing.
- Implement state management and connect the components to the state.
- Create API services that interact with the backend endpoints.
- Implement error handling and user notifications for a better user experience.

Backend:

- Set up the Java project and configure the necessary dependencies.
- Design the data model, including entities for customers, items, orders, and billing.
- Implement CRUD (Create, Read, Update, Delete) operations for each entity.
- Set up the database connection and configure the ORM.
- Create the API endpoints and map them to the appropriate business logic.
- Implement authentication and authorization for the backend services.

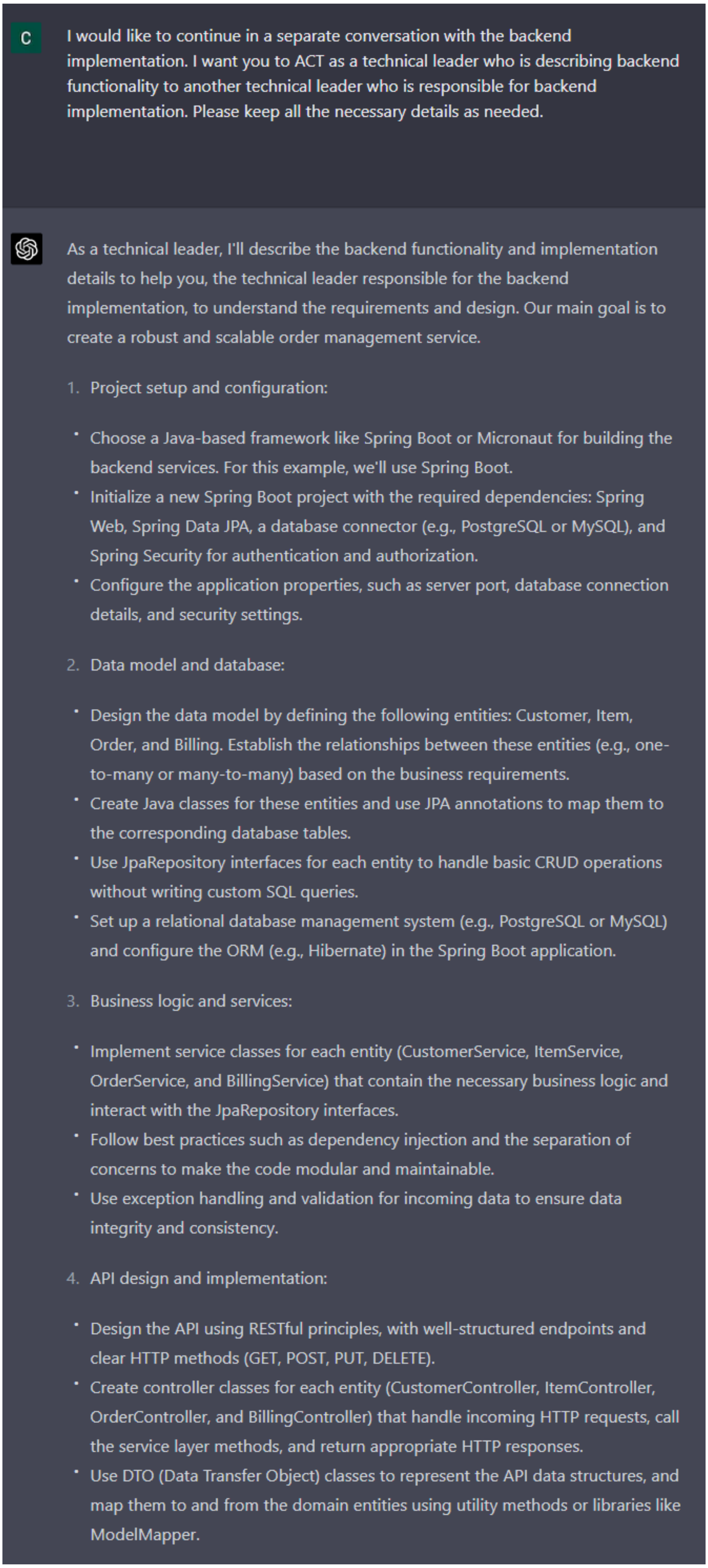
4. Additional recommendations:

- Use Git for version control and collaboration.
- Develop the project using Agile methodologies, with regular sprints and meetings.
- Set up continuous integration and deployment (CI/CD) using tools like Jenkins, Travis CI, or GitHub Actions.
- Write unit tests and integration tests to ensure the quality of the code.
- Use a project management tool like Jira or Trello for tracking progress and managing tasks.

By following these steps and suggestions, you can create a robust order management service with a React frontend and Java backend that meets your requirements.

Then you can ask the AI create a short summary for another AI tab for backend, so I can continue separately and save context:

```
I would like to continue in a separate conversation with the backend implementation. I want you to ACT as a technical leader who is describing backend functionality to another technical leader who is responsible for backend implementation. Please keep all the necessary details as needed.
```



I would like to continue in a separate conversation with the backend implementation. I want you to ACT as a technical leader who is describing backend functionality to another technical leader who is responsible for backend implementation. Please keep all the necessary details as needed.

As a technical leader, I'll describe the backend functionality and implementation details to help you, the technical leader responsible for the backend implementation, to understand the requirements and design. Our main goal is to create a robust and scalable order management service.

1. Project setup and configuration:

- Choose a Java-based framework like Spring Boot or Micronaut for building the backend services. For this example, we'll use Spring Boot.
- Initialize a new Spring Boot project with the required dependencies: Spring Web, Spring Data JPA, a database connector (e.g., PostgreSQL or MySQL), and Spring Security for authentication and authorization.
- Configure the application properties, such as server port, database connection details, and security settings.

2. Data model and database:

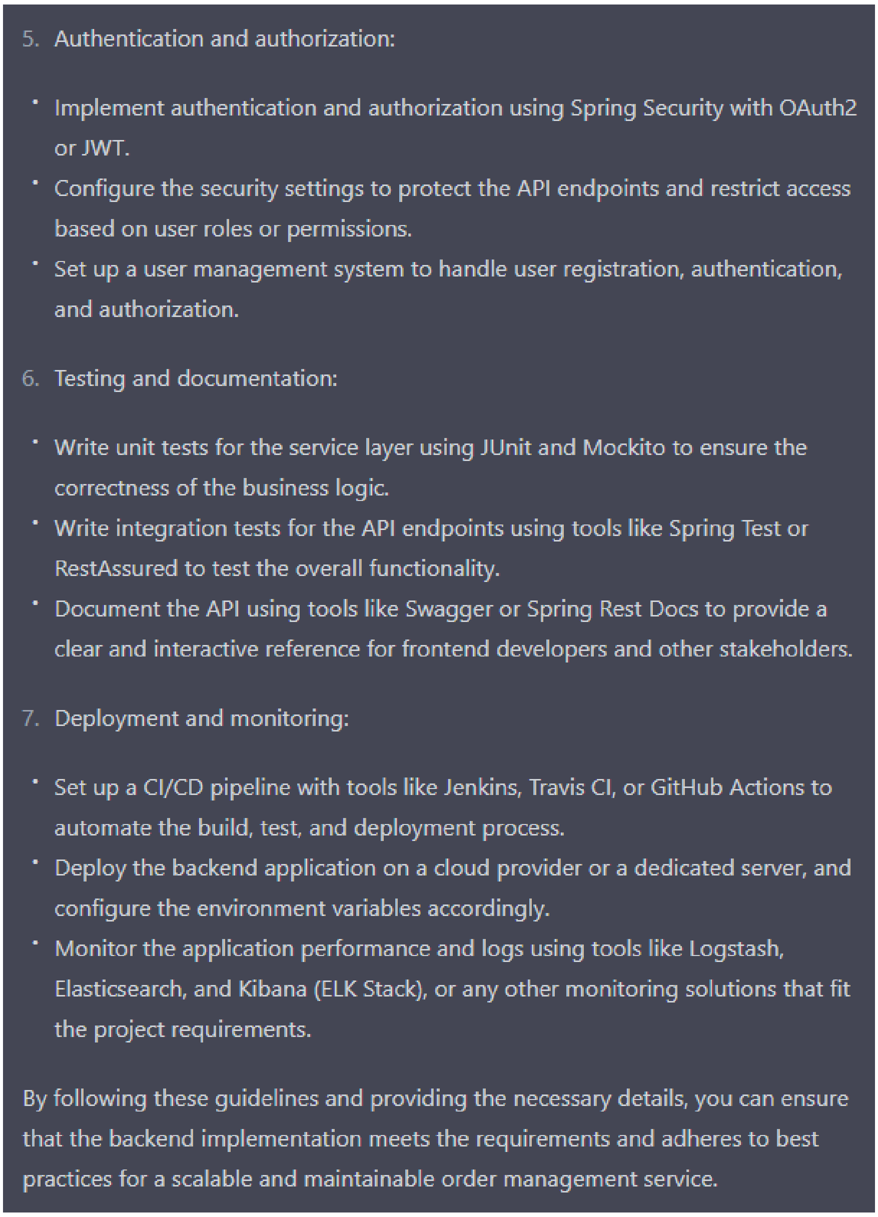
- Design the data model by defining the following entities: Customer, Item, Order, and Billing. Establish the relationships between these entities (e.g., one-to-many or many-to-many) based on the business requirements.
- Create Java classes for these entities and use JPA annotations to map them to the corresponding database tables.
- Use JpaRepository interfaces for each entity to handle basic CRUD operations without writing custom SQL queries.
- Set up a relational database management system (e.g., PostgreSQL or MySQL) and configure the ORM (e.g., Hibernate) in the Spring Boot application.

3. Business logic and services:

- Implement service classes for each entity (CustomerService, ItemService, OrderService, and BillingService) that contain the necessary business logic and interact with the JpaRepository interfaces.
- Follow best practices such as dependency injection and the separation of concerns to make the code modular and maintainable.
- Use exception handling and validation for incoming data to ensure data integrity and consistency.

4. API design and implementation:

- Design the API using RESTful principles, with well-structured endpoints and clear HTTP methods (GET, POST, PUT, DELETE).
- Create controller classes for each entity (CustomerController, ItemController, OrderController, and BillingController) that handle incoming HTTP requests, call the service layer methods, and return appropriate HTTP responses.
- Use DTO (Data Transfer Object) classes to represent the API data structures, and map them to and from the domain entities using utility methods or libraries like ModelMapper.



5. Authentication and authorization:

- Implement authentication and authorization using Spring Security with OAuth2 or JWT.
- Configure the security settings to protect the API endpoints and restrict access based on user roles or permissions.
- Set up a user management system to handle user registration, authentication, and authorization.

6. Testing and documentation:

- Write unit tests for the service layer using JUnit and Mockito to ensure the correctness of the business logic.
- Write integration tests for the API endpoints using tools like Spring Test or RestAssured to test the overall functionality.
- Document the API using tools like Swagger or Spring Rest Docs to provide a clear and interactive reference for frontend developers and other stakeholders.

7. Deployment and monitoring:

- Set up a CI/CD pipeline with tools like Jenkins, Travis CI, or GitHub Actions to automate the build, test, and deployment process.
- Deploy the backend application on a cloud provider or a dedicated server, and configure the environment variables accordingly.
- Monitor the application performance and logs using tools like Logstash, Elasticsearch, and Kibana (ELK Stack), or any other monitoring solutions that fit the project requirements.

By following these guidelines and providing the necessary details, you can ensure that the backend implementation meets the requirements and adheres to best practices for a scalable and maintainable order management service.

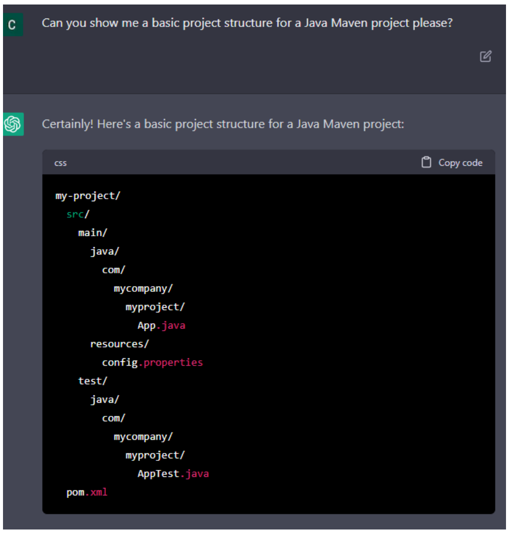
You can now take this information and paste it into a separate tab and continue conversation there. As you can image from the beckend implementation details, there is so much stuff, that you can apply the separation of concerns even to other subtopics like "datamodel and database", "testing" etc...

4. External memory

I noticed that not only me, but a lot of people started "saving" prompts that worked for them and then directly copy-pasting them into new conversations or slightly adapting them for the topic at hand. I think, that we are subconsciously creating external memory for these AIs as they don't have a memory per se.

What does that mean? Well, even in pont "3. Separate concerns" we are basically providing new tabs with a condensed version of previous conversations => a memory.

As a very simple example we can use a project project's folder structure, which we will save into a separate text file. Every time we want to talk to GPT about our project, we can prime the AI with the project structure:



Can you show me a basic project structure for a Java Maven project please?

Certainly! Here's a basic project structure for a Java Maven project:

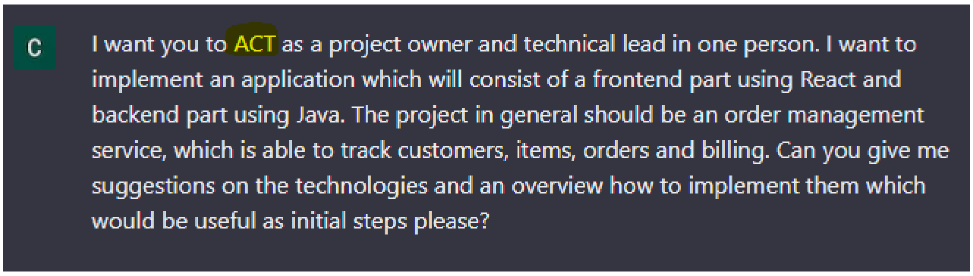
```
my-project/
  src/
    main/
      java/
        com/
          mycompany/
            myproject/
              App.java
      resources/
        config.properties
    test/
      java/
        com/
          mycompany/
            myproject/
              AppTest.java
  pom.xml
```

As you can imagine, you can create logs and summaries regarding the topic you are working on, so you are able to prime the AI every time you need it.

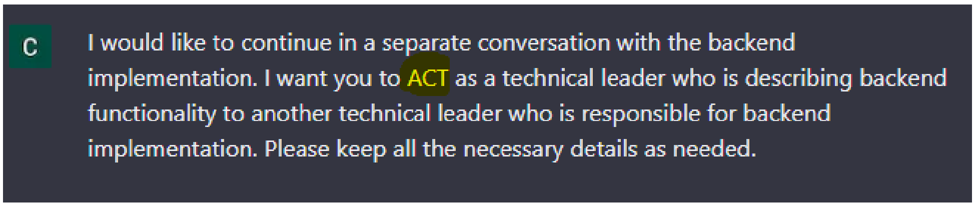
5. The ACT keyword

I found during my several months of usage, that if you are able to get the AI to the position of a particular expert (to think it is an expert in a particular field), it is performing much better than it would otherwise. You can even say specific names of developers, photographers, artists, journalists… You do this by using the ACT keyword.

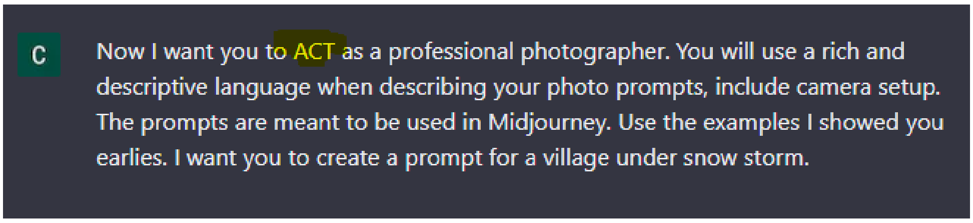
Here are few examples:



I want you to ACT as a project owner and technical lead in one person. I want to implement an application which will consist of a frontend part using React and backend part using Java. The project in general should be an order management service, which is able to track customers, items, orders and billing. Can you give me suggestions on the technologies and an overview how to implement them which would be useful as initial steps please?

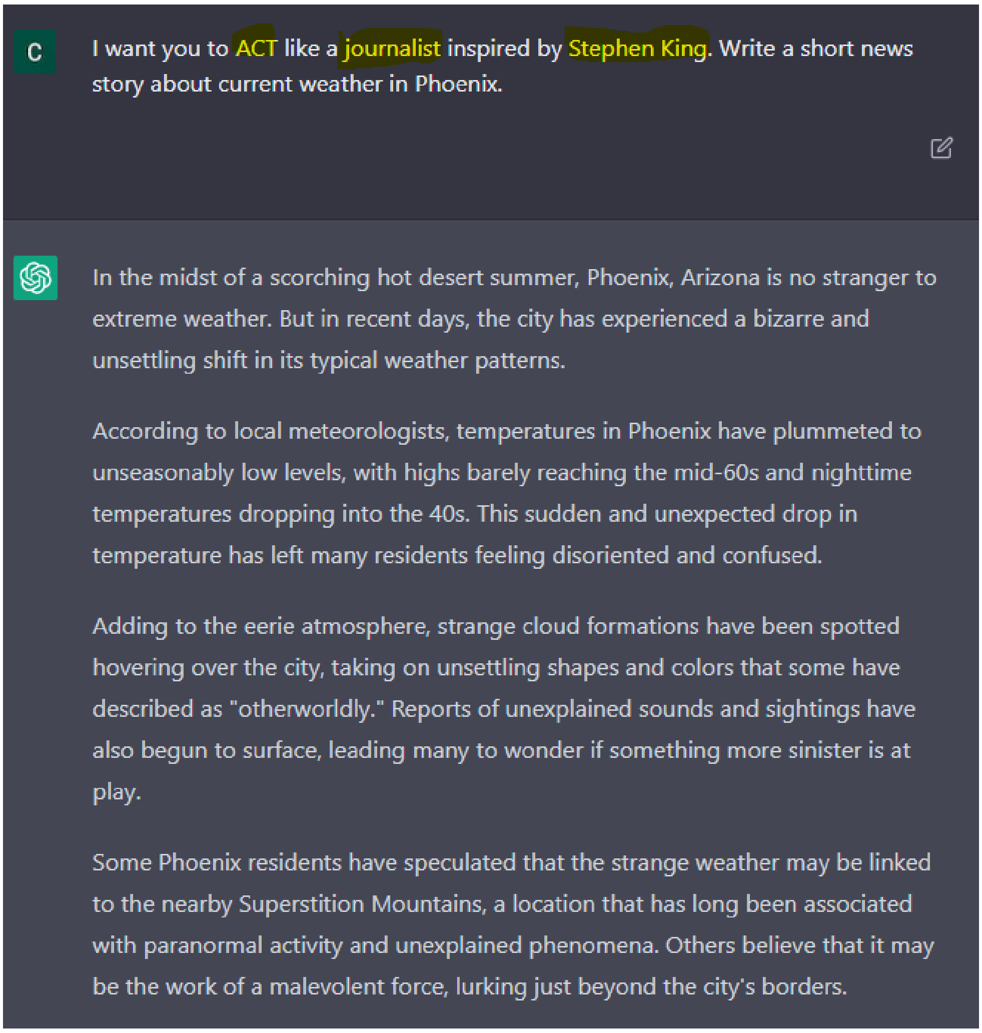


I would like to continue in a separate conversation with the backend implementation. I want you to ACT as a technical leader who is describing backend functionality to another technical leader who is responsible for backend implementation. Please keep all the necessary details as needed.



Now I want you to ACT as a professional photographer. You will use a rich and descriptive language when describing your photo prompts, include camera setup. The prompts are meant to be used in Midjourney. Use the examples I showed you earlies. I want you to create a prompt for a village under snow storm.

Observe how I was able to influence the style of the news story:



C

I want you to ACT like a journalist inspired by Stephen King. Write a short news story about current weather in Phoenix.

In the midst of a scorching hot desert summer, Phoenix, Arizona is no stranger to extreme weather. But in recent days, the city has experienced a bizarre and unsettling shift in its typical weather patterns.

According to local meteorologists, temperatures in Phoenix have plummeted to unseasonably low levels, with highs barely reaching the mid-60s and nighttime temperatures dropping into the 40s. This sudden and unexpected drop in temperature has left many residents feeling disoriented and confused.

Adding to the eerie atmosphere, strange cloud formations have been spotted hovering over the city, taking on unsettling shapes and colors that some have described as "otherworldly." Reports of unexplained sounds and sightings have also begun to surface, leading many to wonder if something more sinister is at play.

Some Phoenix residents have speculated that the strange weather may be linked to the nearby Superstition Mountains, a location that has long been associated with paranormal activity and unexplained phenomena. Others believe that it may be the work of a malevolent force, lurking just beyond the city's borders.

Conclusion

All and all, these are 12 months of experience with GPTs (yes even before ChatGPT came out) condensed into 5 points. **These points really help me to pinpoint the task at hand** and rapidly iterate over ideas, new code and explanations for things, that I don't understand. There is more, but for that, I will create a new tutorial :)

P.S.: Don't be affraid to restart conversation exactly as you would try to rewrite code. It is a machine afterall :)