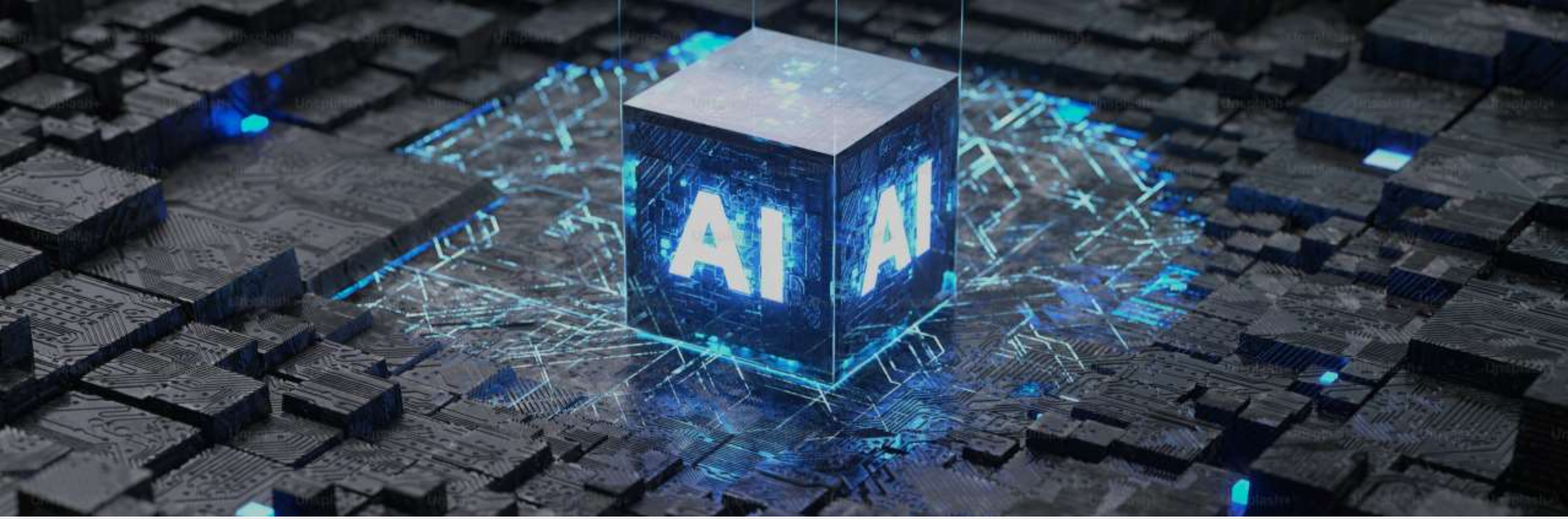


Developing AI Applications Using Prompts: Simplifying Artificial Intelligence

Tutorial 9

José Alfredo Arriaga Rosillo

La Salle Bajio

Who am I?

Jar76774@lasallebajio.edu.mx

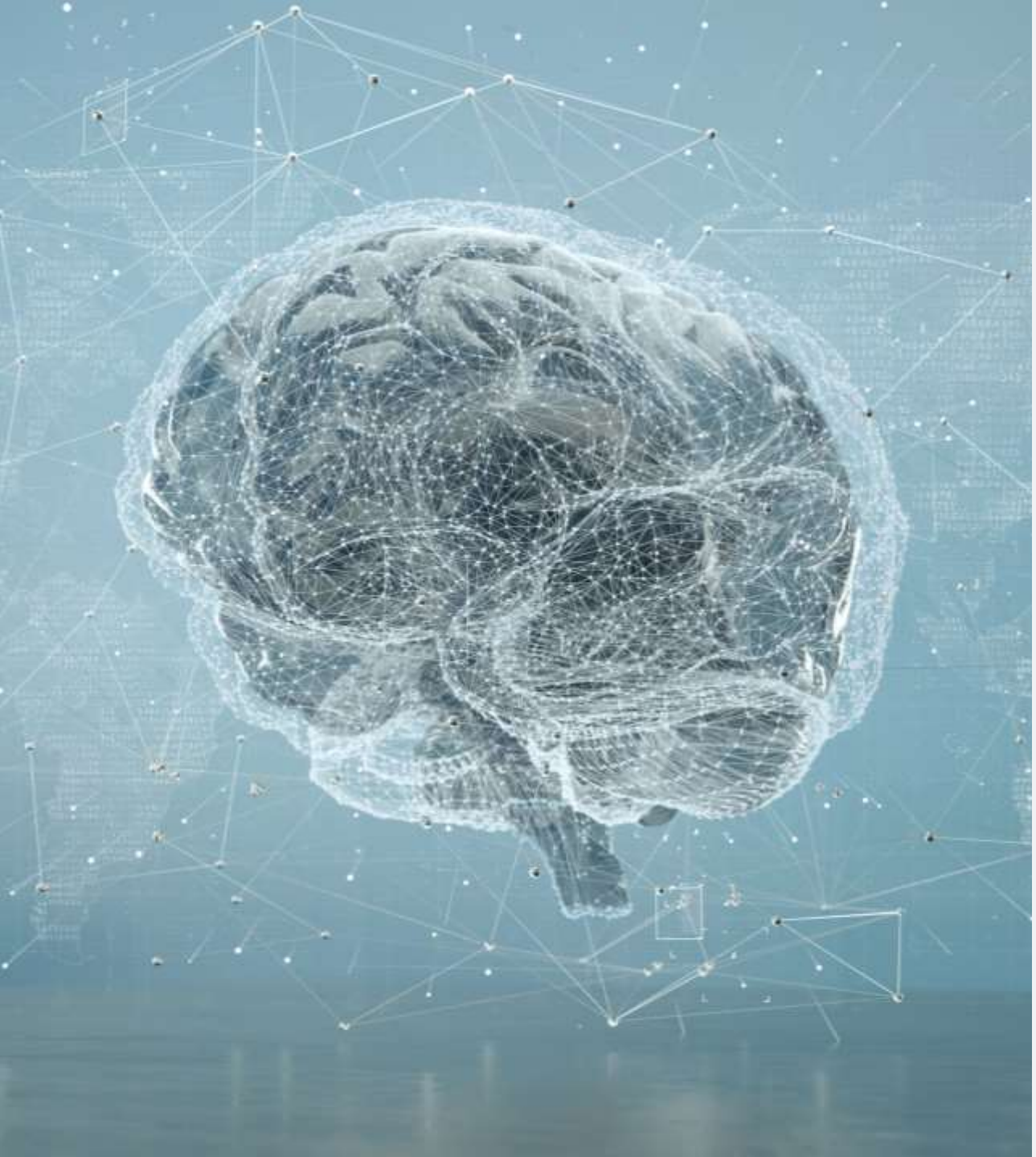
@JAlfredAR

Developer	• Skydome (Meta Quest 2, 3 and 3s) • Echo Insight (IOS, IPadOS and MacOS) • Meeting Checker (Windows)
Student Ambassador	• Intel • Microsoft • Postman
Certifications	• Mongo DB “Associate Developer” • Postman Api Fundamentals Student Expert
Company Founder	• Currently in the process of founding a company called ”ARXBITE" focused on AR/MR/VR, software development, videogames development and physical gadgets

What will we see?

Using prompts, we can get so much information and receive it in any format we want.

We can use this to build our own apps using LLM´s API



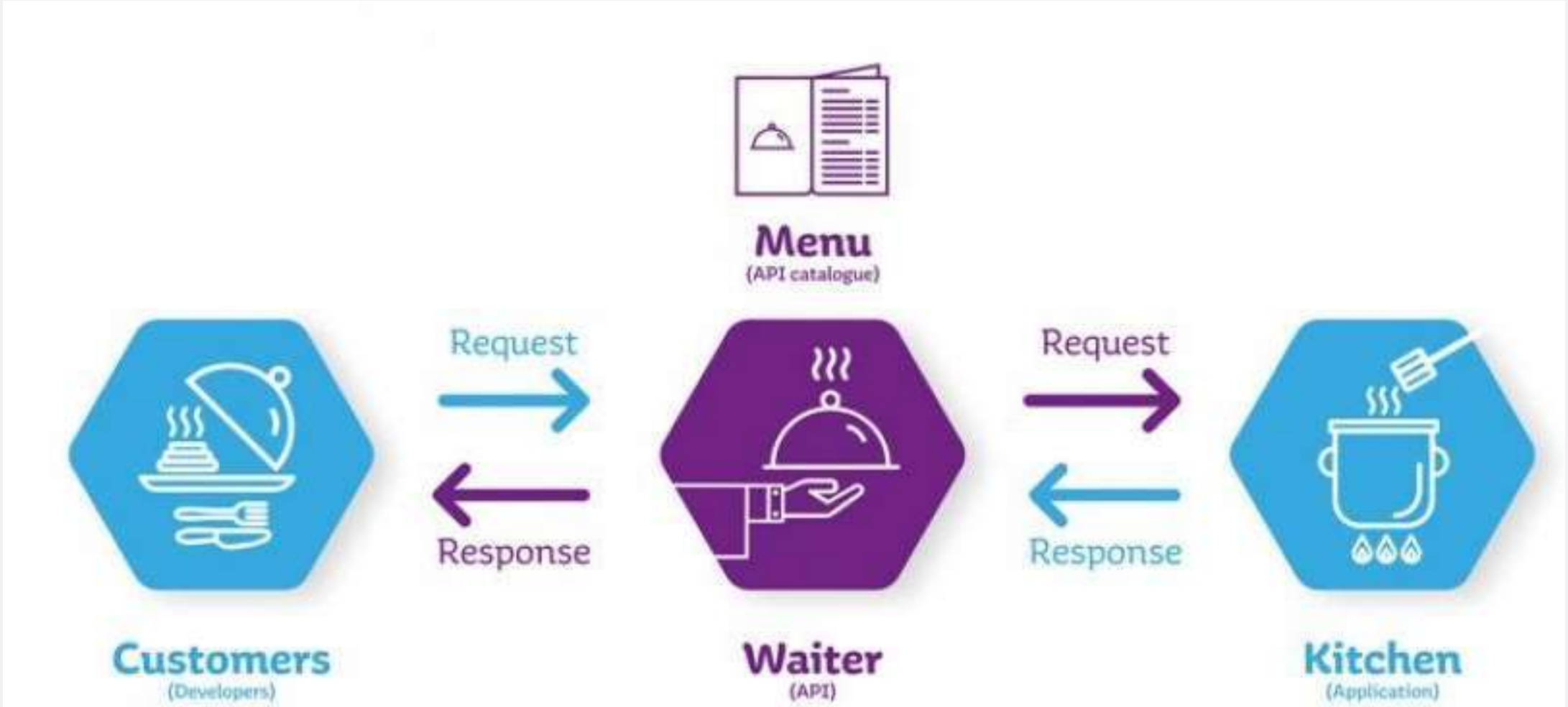
What is an LLM´s

- Large Language Models, are a type of artificial intelligence (AI) model designed to understand and generate human-like text based on vast amounts of data.

What is a prompt?

- A prompt is an instruction, question, or piece of information provided to elicit a response or guide action.

What is an API?



Chat GPT

- ChatGPT is an advanced conversational AI developed by OpenAI.
- It got released on November 30 of 2022
- It can do:
  - Text generation
  - Image Processing
  - Files processing
  - Real time interactions through WebSocket

Chat GPT API

- GPT-3.5 Turbo
- GPT-4
- GPT-4 Turbo
- GPT-4o
- GPT-4o Mini

Let´s build a movies recommendation App

Create a OpenAI API Account

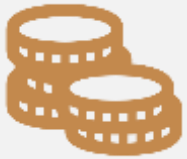
Open https://***platform.openai.com***/

Click Sign Up

Settings > Billing

What are Chat GPT credits?

Chat GPT Credits is the Pre-paid balance you have on your Chat GPT Account



You pay based on the input and output tokens you use

What are Chat GPT tokens?

- Represent the amount of text (words, characters, punctuation) sent to the model for processing or generated by the AI model.
- There are two types of Tokes
  - Input tokens (You can control them)
    - These refer to the number of tokens in the text **you send** to the model, so you can easily use the number of tokens you want to use
  - Output tokens (You can´t easily control them)
    - This refer to the number of tokens generated by the AI model in response to your input, so even you can tell it through the prompt with which information to answer with it may also complement it with natural language due it is an LLM so that would also cost tokens.

How does Chat GPT Tokens work and what´s the pricing?

- Tokenization: Text is split into smaller units called tokens. A token can be as small as a single character or as large as a full word, depending on the complexity of the word and its representation in the model's vocabulary. For example, simple words like "hello" may count as one token, while more complex words or phrases may count as multiple tokens.
- Cost
  - We will use GPT-4o and it costs
    - $2.50 / 1M input tokens
    - $10.00 / 1M output tokens

Dataset Download

- https://***bit.ly/MicaiTutorial9DataBase***

What´s a Json

- Its like a dictionary it have **Arrays** that are like the dictionary section for example: **Fruits**, then you have the **objects** that are like the container of information that is stored for each object for example **on a traditional dictionary the word and the definition**, and finally you have the **keys** that are how you access the **values** of the data of the object for example **name and description**

Next Steps

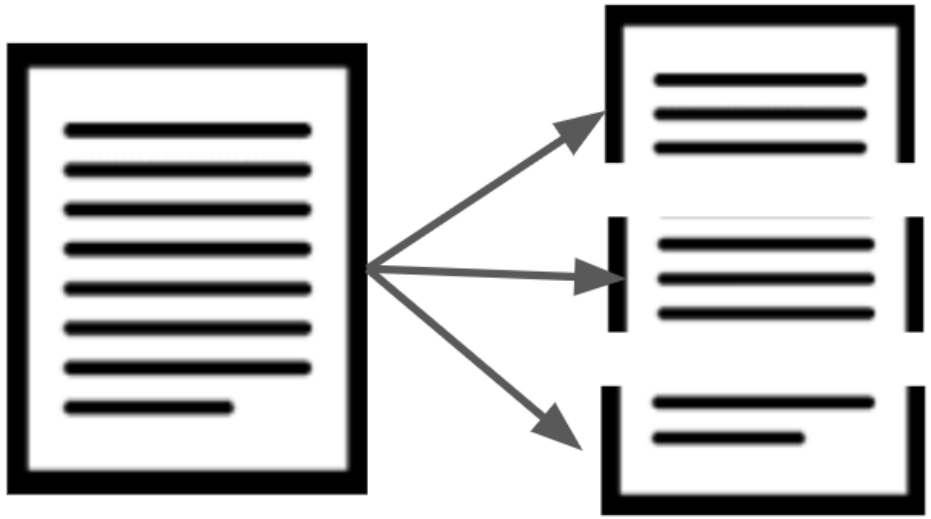
CREATE PROJECT

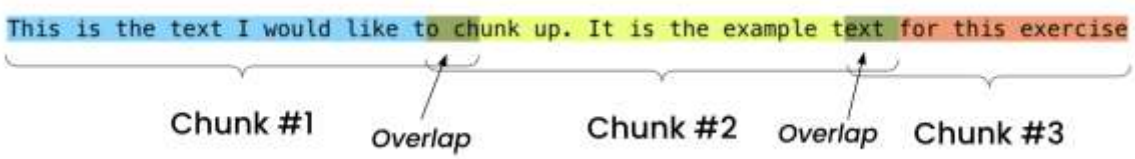
DASHBOARD

VECTOR STORE (UPLOAD DATASET)

Chunk Size and Chunk Overlap

- Chunk Size
  - Chunk size refers to the number of tokens or characters that each text fragment (or chunk) will contain when split for indexing or storage in a vector database
  - Use less chunk size for less context for example separate a book on chapter chunks
- Chunk Overlap
  - Chunk overlap refers to how much adjacent chunks "overlap" in content. This ensures that no important information is lost between chunks
  - Use more chunk size if it have long words and if the idea is not well separated





Tokenizer

- https://**platform.openai.com/tokenizer**
- About 200 tokens per Movie/Series
- Overlap about 20 tokens

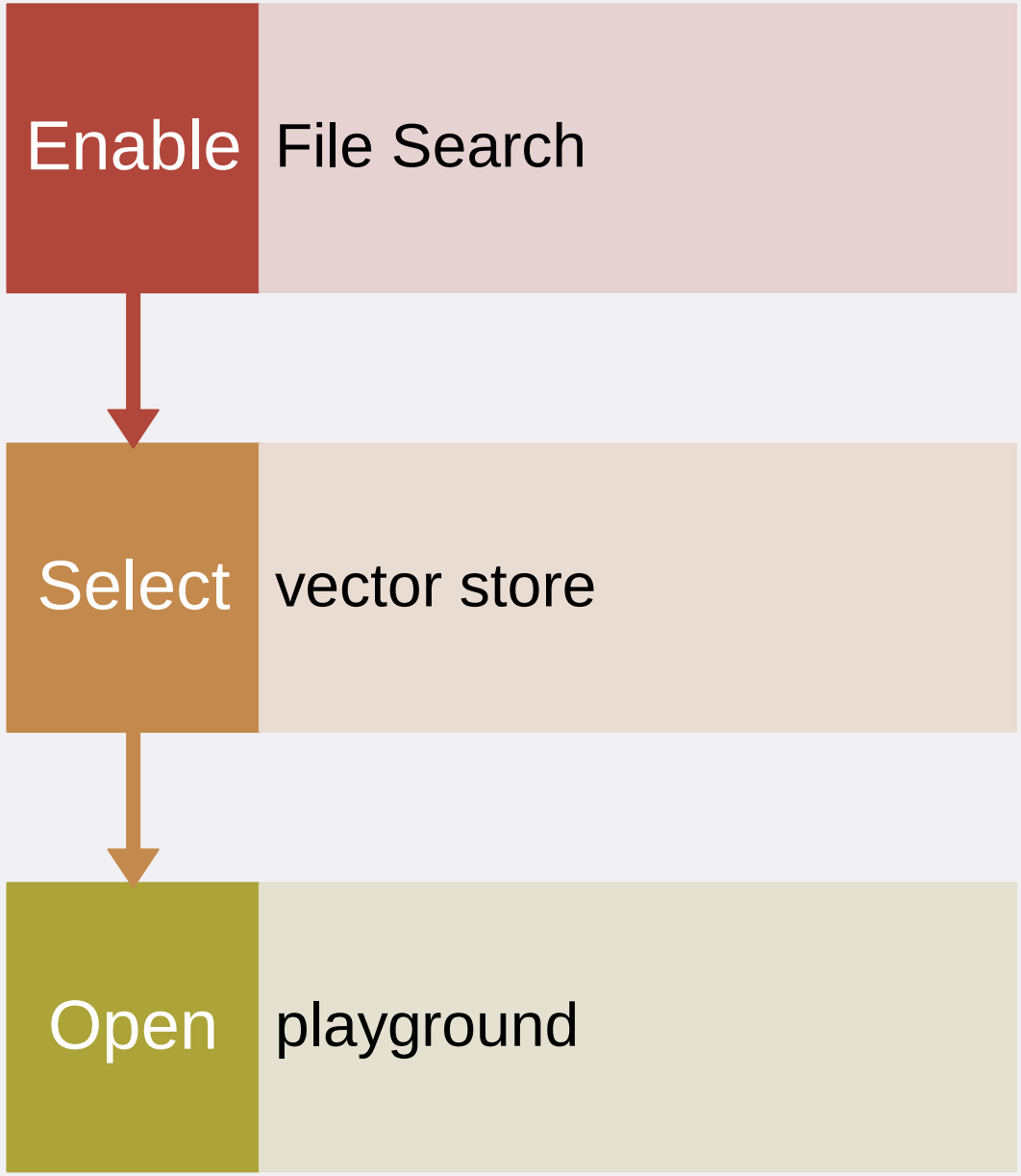
ChatGPT Assistants

- Is a customizable chatbot or conversational agent powered by OpenAI's models, you can give them custom orders to receive the desired output

System Instructions

- https://**bit.ly/MicaiTutorial9Main**

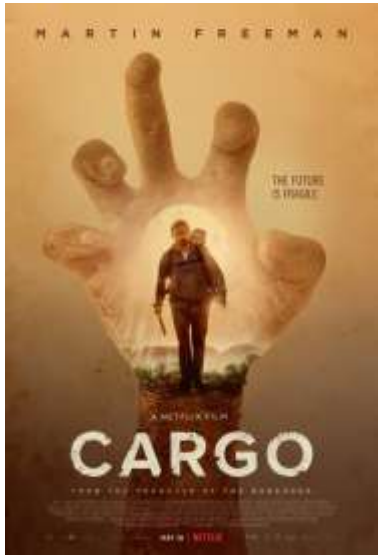
Next Steps




SQUID GAME


DE NIRO
PACINO
PESCI
THE IRISHMAN
NETFLIX


BLACK MIRROR
NETFLIX

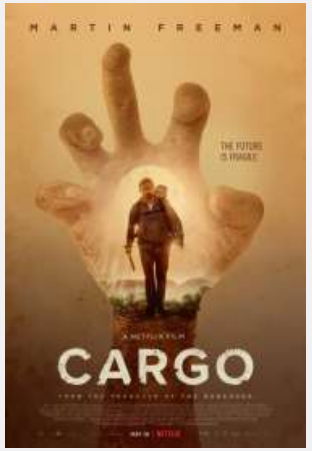

MARTIN FREEMAN
THE FUTURE IS FRAGILE
CARGO


THE LEGO MOVIE
THE MITCHELLS VS THE MACHINES
APRIL 30 | NETFLIX


STRANGER THINGS

Title: The Mitchells vs. the Machines
Genres: Comedy, Family, Sci-Fi, Animation
Production Countries: US
Age Certification: PG
Release Year: 2021
Type: Movie

Title: Black Mirror
Genres: Drama, Sci-Fi, Thriller, European
Production Countries: GB
Age Certification: TV-MA
Release Year: 2011
Type: Show



Title: Cargo
Genres: Fantasy, Sci-Fi, Drama
Production Countries: IN
Age Certification: PG-13
Release Year: 2019
Type: Movie

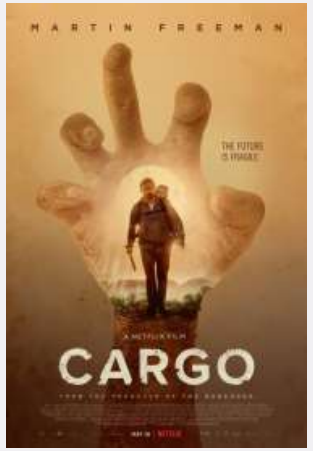
Title: Stranger Things
Genres: Sci-Fi, Thriller, Drama, Fantasy, Horror
Production Countries: US
Age Certification: TV-14
Release Year: 2016
Type: Show

Title: The Irishman
Genres: Crime, Drama, History, Thriller
Production Countries: US
Age Certification: R
Release Year: 2019
Type: Movie

Title: Squid Game
Genres: Action, Thriller, Drama
Production Countries: KR
Age Certification: TV-MA
Release Year: 2021
Type: Show

Title: The Mitchells vs. the Machines
Genres: Comedy, Family, Sci-Fi, Animation
Production Countries: US
Age Certification: PG
Release Year: 2021
Type: Movie

Title: Black Mirror
Genres: Drama, Sci-Fi, Thriller, European
Production Countries: GB
Age Certification: TV-MA
Release Year: 2011
Type: Show



Title: Cargo
Genres: Fantasy, Sci-Fi, Drama
Production Countries: IN
Age Certification: PG-13
Release Year: 2019
Type: Movie

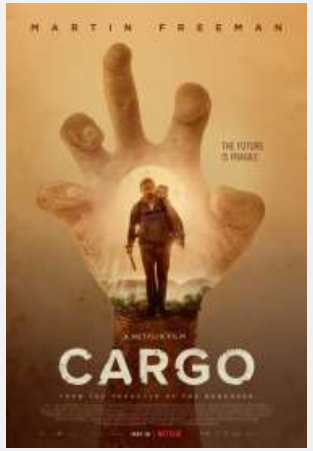
Title: Stranger Things
Genres: Sci-Fi, Thriller, Drama, Fantasy, Horror
Production Countries: US
Age Certification: TV-14
Release Year: 2016
Type: Show

Title: The Irishman
Genres: Crime, Drama, History, Thriller
Production Countries: US
Age Certification: R
Release Year: 2019
Type: Movie

Title: Squid Game
Genres: Action, Thriller, Drama
Production Countries: KR
Age Certification: TV-MA
Release Year: 2021
Type: Show

Title: The Mitchells vs. the Machines
Genres: Comedy, Family, Sci-Fi, Animation
Production Countries: US
Age Certification: PG
Release Year: 2021
Type: Movie

Title: Black Mirror
Genres: Drama, Sci-Fi, Thriller, European
Production Countries: GB
Age Certification: TV-MA
Release Year: 2011
Type: Show

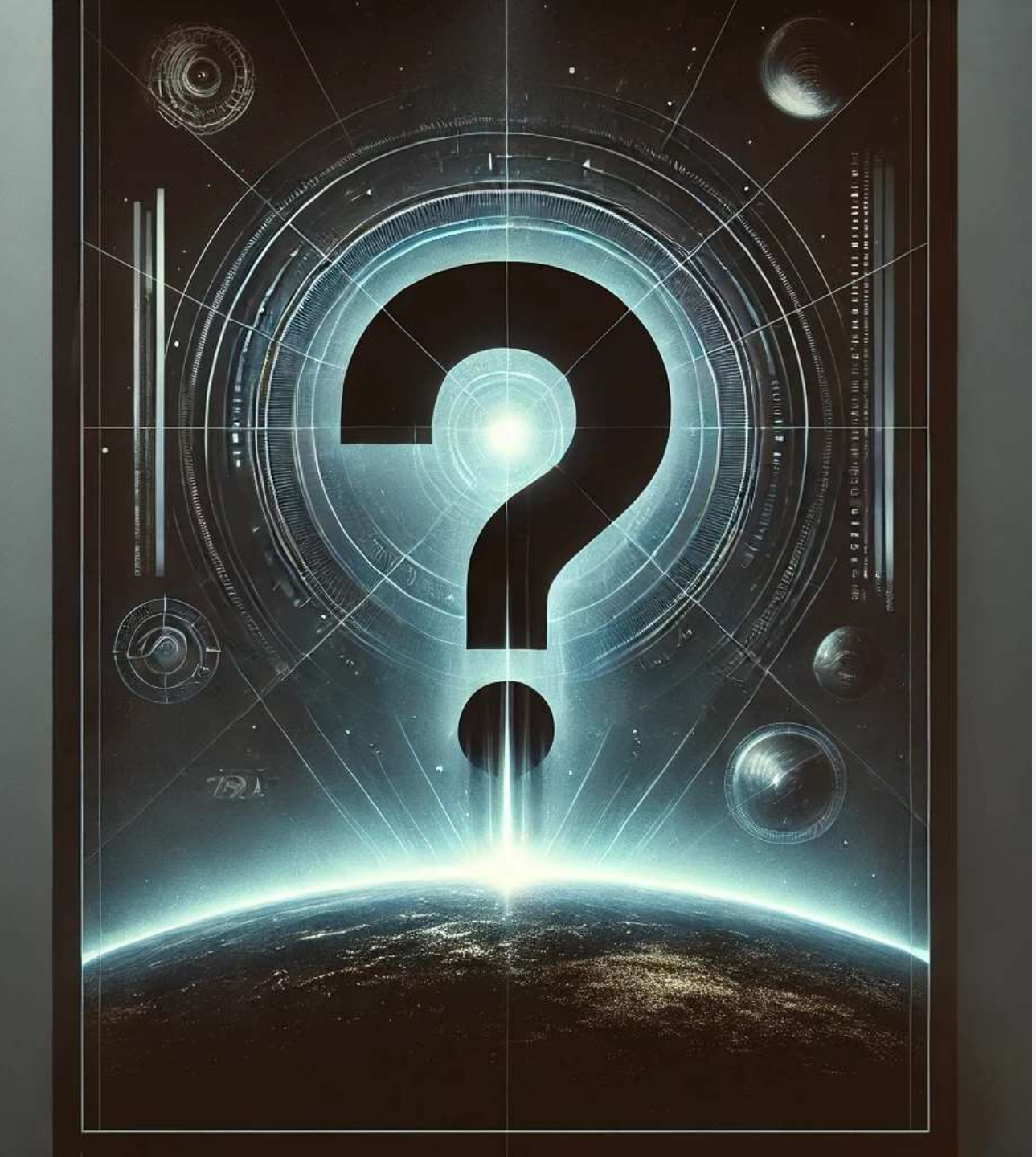


Title: Cargo
Genres: Fantasy, Sci-Fi, Drama
Production Countries: IN
Age Certification: PG-13
Release Year: 2019
Type: Movie

Title: Stranger Things
Genres: Sci-Fi, Thriller, Drama, Fantasy, Horror
Production Countries: US
Age Certification: TV-14
Release Year: 2016
Type: Show

Title: The Irishman
Genres: Crime, Drama, History, Thriller
Production Countries: US
Age Certification: R
Release Year: 2019
Type: Movie

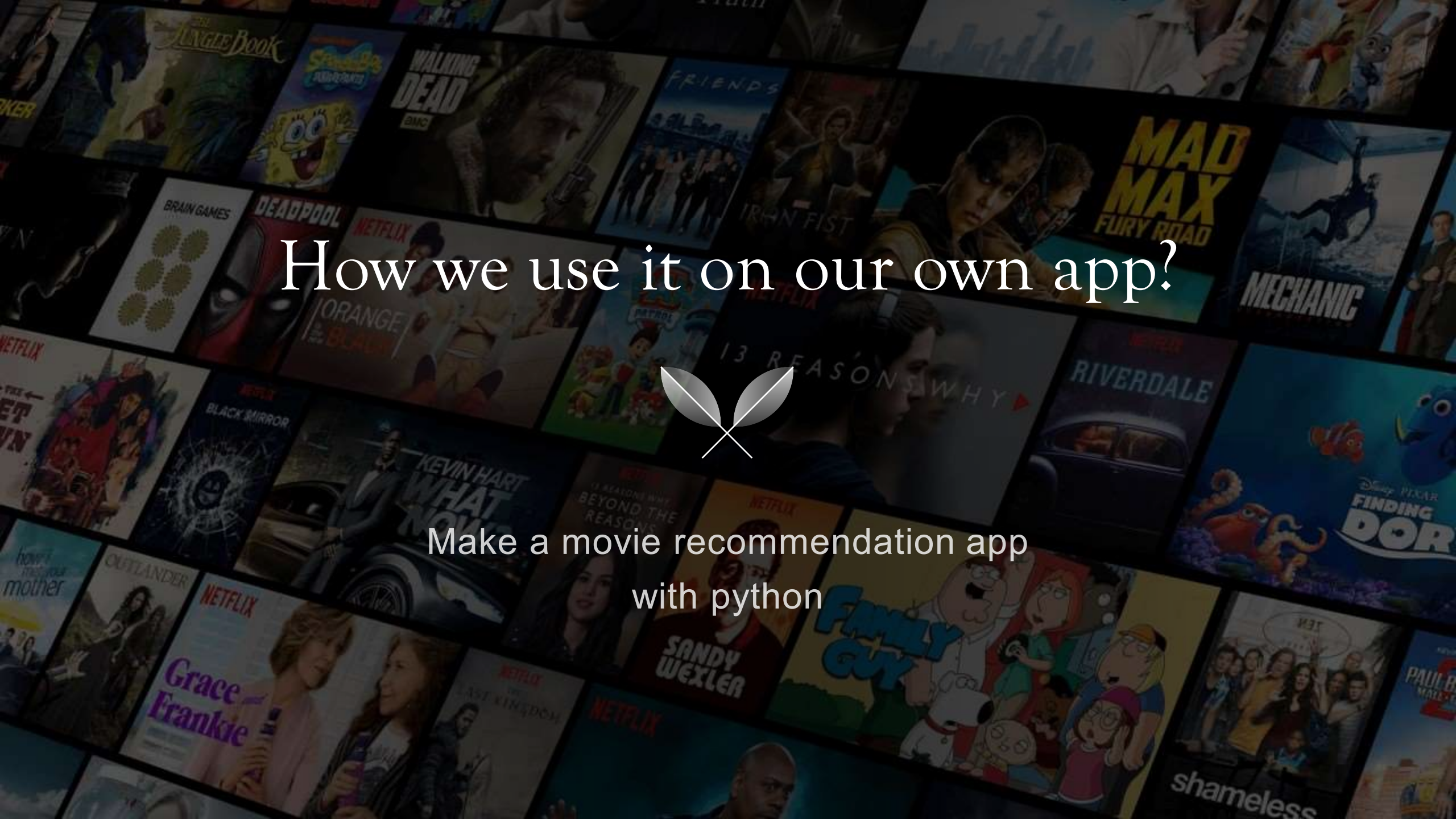
Title: Squid Game
Genres: Action, Thriller, Drama
Production Countries: KR
Age Certification: TV-MA
Release Year: 2021
Type: Show



User Likes Most

- **Genres**: Sci-Fi, Thriller, Drama, Fantasy
- **Production Countries**: US
- **Age Certification**: TV-MA
- **Release Year**: 2019 and 2021

How we use it on our own app?



Make a movie recommendation app
with python

Install python

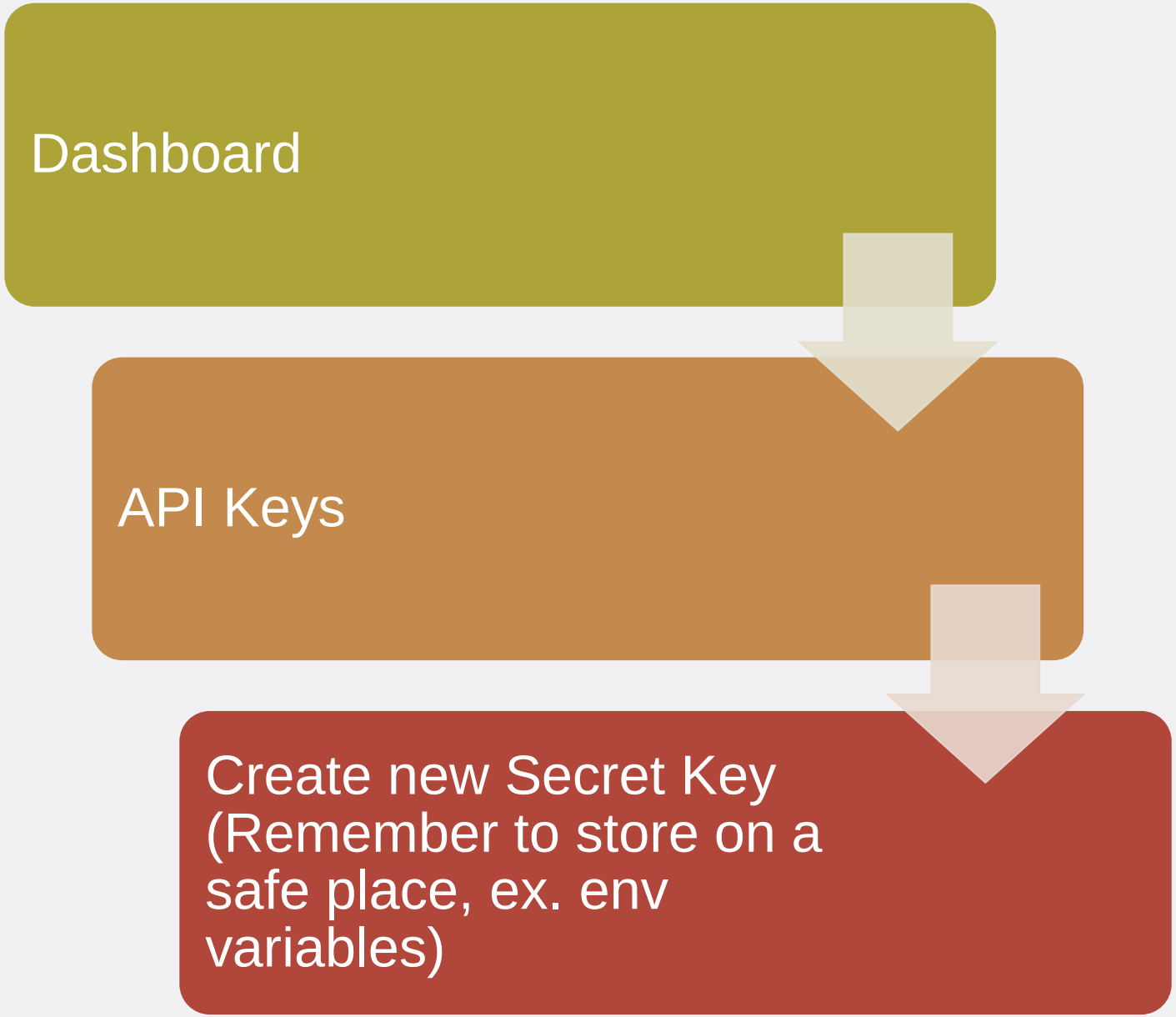
https://**www.python.org/downloads/**

python3 -m pip install OpenAI

```python
import json
from openai import OpenAI

def add_movie_menu():
    movies = []
    while True:
        movie = input("Enter the name of the movie: ")
        movies.append(movie)
        another = input("Do you want to add another movie? (yes/no): ").lower()
        if another == 'yes':
            print("")
        elif another == 'no':
            break
        else:
            print("Invalid Option | Defaults to add movie")
    return movies
```

Get your OpenAI Api Key



```python
def recommend_movies(moviesArr: list):
    api_key = "Open AI Token"
    client = OpenAI(api_key=api_key)
    assistantId = "Open AI Assistant"

    movies_json = json.dumps({"films": moviesArr})

    try:
        thread = client.beta.threads.create(
            messages=[
                {
                    "role": "user",
                    "content": movies_json,
                    "attachments": []
                }
            ]
        )
    except Exception as e:
        print(f"Error creating thread: {e}")
        return

    try:
        run = client.beta.threads.runs.create(
            thread_id=thread.id, assistant_id=assistantId
        )

        while run.status != "completed":
            run = client.beta.threads.runs.retrieve(
                thread_id=thread.id, run_id=run.id
            )
        messages = client.beta.threads.messages.list(thread_id=thread.id)
        #print(messages)
        parsed_content = messages.data[0].content[0].text.value
        parsed_content_json = json.loads(parsed_content)
        return parsed_content_json

    except Exception as e:
        print(f"Error during run: {e}")
```

You should have

- Your Open AI Token
- Open AI Assistant ID

```python
# Run the menu and Get Recommendations
added_movies = add_movie_menu() # Get the data to give to the AI
recommended_movies = recommend_movies(added_movies) # Get AI Response

# Print the data to user
userPrint(recommended_movies) # Show the data to the user
```

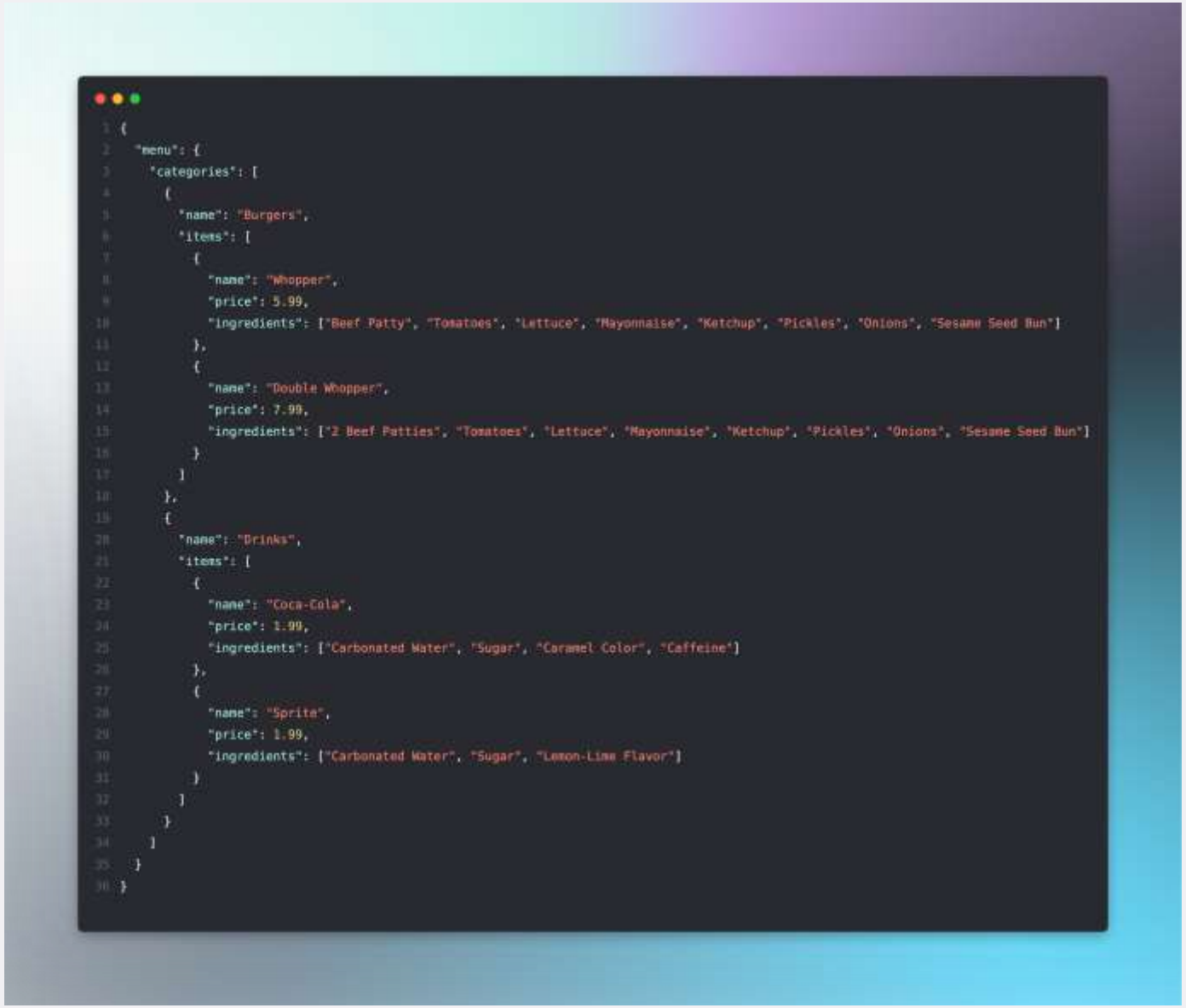
What can we do with the Json and why to use it as AI response?

USE IT ON AN APP

ADD IT TO AN API

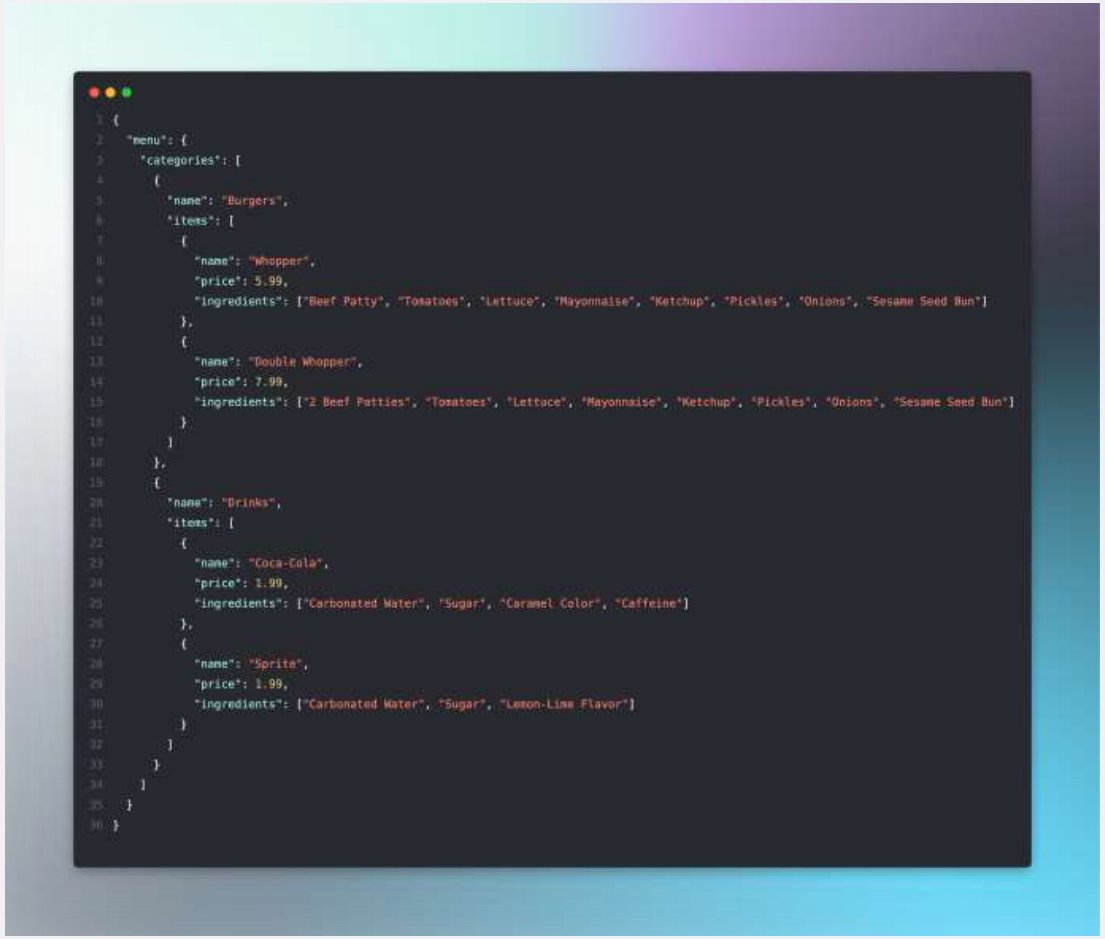
USE IT FOR TEXT TO SPEECH

Example of text to speech on restaurant



```json
{
  "menu": {
    "categories": [
      {
        "name": "Burgers",
        "items": [
          {
            "name": "Whopper",
            "price": 5.99,
            "ingredients": ["Beef Patty", "Tomatoes", "Lettuce", "Mayonnaise", "Ketchup", "Pickles", "Onions", "Sesame Seed Bun"]
          },
          {
            "name": "Double Whopper",
            "price": 7.99,
            "ingredients": ["2 Beef Patties", "Tomatoes", "Lettuce", "Mayonnaise", "Ketchup", "Pickles", "Onions", "Sesame Seed Bun"]
          }
        ]
      },
      {
        "name": "Drinks",
        "items": [
          {
            "name": "Coca-Cola",
            "price": 1.99,
            "ingredients": ["Carbonated Water", "Sugar", "Caramel Color", "Caffeine"]
          },
          {
            "name": "Sprite",
            "price": 1.99,
            "ingredients": ["Carbonated Water", "Sugar", "Lemon-Lime Flavor"]
          }
        ]
      }
    ]
  }
}
```

Example of text to speech on restaurant



```
{
  "menu": {
    "categories": [
      {
        "name": "Burgers",
        "items": [
          {
            "name": "Whopper",
            "price": 5.99,
            "ingredients": ["Beef Patty", "Tomatoes", "Lettuce", "Mayonnaise", "Ketchup", "Pickles", "Onions", "Sesame Seed Bun"]
          },
          {
            "name": "Double Whopper",
            "price": 7.99,
            "ingredients": ["2 Beef Patties", "Tomatoes", "Lettuce", "Mayonnaise", "Ketchup", "Pickles", "Onions", "Sesame Seed Bun"]
          }
        ]
      },
      {
        "name": "Drinks",
        "items": [
          {
            "name": "Coca-Cola",
            "price": 1.99,
            "ingredients": ["Carbonated Water", "Sugar", "Caramel Color", "Caffeine"]
          },
          {
            "name": "Sprite",
            "price": 1.99,
            "ingredients": ["Carbonated Water", "Sugar", "Lemon-Lime Flavor"]
          }
        ]
      }
    ]
  }
}
```

- System instructions
  - Get the user order and return it on json format with an array of the order with the Name, ingredients he choose, (if he doesn’t mention any ingredient include all ingredients that the chosen item have) for each item in the order, and after the array add a field of “VoiceMessage” confirming the order and asking the user to go to the next window.

Other Use Cases

- **Food Ordering App**: Personalized Recommendations based on the day, time and other orders, by not giving the old suggestions based on the restaurant or food, you can base it even on ingredients the user have ordered food with.
- **Financial Apps**: Personalized Money-Saving Recommendations based on what the user purchases and when to do a purchase to save money.
- **Investment Risk and Complex Statistics**: Personalized investing suggestions based on the company history recent news and its impact on the stock market suggest to the user when to invest on which company.
- **Travel Apps**: Personalized Itinerary Planning based on what the user do on other travels considering the type of activities and find similar ones.
- **Personalized Weather Insights**: Based on what the user is worried on based on the day, ex. If its Monday on morning the user may be worried for an storm, but if its Sunday he may not be worried about that

Time for questions

Finished Code

Thank you for your assistance

Jar76774@lasallebajio.edu.mx

@JAlfredAR