PROJECT

OpenAI JavaScript API: Recipe Blog

As a new food blogger looking to create the next big culinary hub for your readers, get ready to transform your coding skills into a flavorful experience!

In this project, you'll build a recipe generator that provides tailored recipe recommendations by providing the model with context about the user's dietary preferences, ingredients on hand, and cuisine favorites. You'll practice prompt engineering to format the model output recipes in the best format for your blog.

Let's get cooking!

Tasks

13/14 complete

Mark the tasks as complete by checking them off

Setting Up the Environment

1.

Start the project by importing the OpenAI class from the openai library.

Use the following syntax:

import ClassName from "libraryname";

2.

Create an instance of the OpenAI class and assign it to a variable named client.

Use the following syntax:

const variable = new ClassName();

Create a User Profile

3.

Create an object named userProfile with no entries in it.

Use the following syntax:

const objectName = {};

4.

Update the previously defined object userProfile by adding a new key-value pair. The key should be dietaryRestrictions and the value should be a string listing the user's dietary restrictions.

The restrictions can be separated by commas or written in any other format that clearly specifies the various restrictions.

Use the following syntax:

objectName.newKey = 'newValue';

5.

Add to the userProfile object by inserting a new key-value pair. Assign the key cuisinePreferences and set its value to a string that indicates the user's favorite cuisines.

Format this list with commas or any other clear way to denote the different preferences.

Use the following syntax:

objectName.newKey = 'newValue';

6.

Finish updating the userProfile object by adding a key called "ingredientsAvailable". The value for this key should be a string listing the ingredients the user has on hand.

The list can be separated by commas or written in a format that clearly outlines the various ingredients available to the user.

Use the following syntax:

objectName.newKey = 'newValue';

Prepare the Prompts

7.

Create an object called systemPrompt which will define the instructions for the AI. This object should instruct the AI to generate HTML code for a recipe blog that takes into account specific dietary restrictions, chosen cuisine type, and a provided list of ingredients.

Ensure that the object includes a key named role with the value "system", and another key named content where the value is the instruction string for the AI.

Example Prompt

"Generate an HTML code for a recipe blog that considers dietary restrictions, cuisine type, and ingredients."

Use the following syntax:

```
const objectName = {
  key1: 'value1',
  key2: 'value2'
};
```

8.

Create a string variable named userContent1 that begins the user prompt. This string should start with a sentence indicating your intention to create a recipe blog post. Then, proceed to include the relevant data from the userProfile object, making sure to specify what each piece of data from userProfile represents.

For example, the string should clearly label the dietary restrictions, cuisine preferences, and available ingredients as provided in the userProfile.

Example Prompt

"I want to create a recipe blog post. Here are my dietary restrictions: INSERT_DIETARY_RESTRICTIONS. My cuisine preferences include: INSERT_CUISINE_PREFERENCES. The ingredients I have available are: INSERT_INGREDIENTS_AVAILABLE."

Use the following syntax:

const stringVariable = `This is a ${variable}`;

9.

Construct a string called userContent2 to continue shaping the user prompt. This string should outline the structure of a blog post, including requests for specific elements such as the title, description, ingredients, and instructions.

Within this string, you have the option to include examples illustrating the preferred format for listing ingredients and instructions, guiding the AI on how to present these sections effectively.

Example Prompt

"Please provide a blog post with a title, description, ingredients, and instructions. Format the ingredients and instructions as follows: Ingredients should be bulleted, and instructions should be numbered."

Use the following syntax:

const stringVariable = `This is a ${variable}`;

10.

Construct a string with the name userContent3 that establishes certain limitations for the recipe creation. This string should specify that:

- the recipes must be made using only the ingredients listed in the user_profile
- the AI's output should be limited to a single blog post
- the recipe instructions should not exceed six steps.

These constraints will guide the AI in generating content that adheres to the given parameters.

Example Prompt

"The recipe must use only the listed ingredients and should result in a single blog post with instructions not exceeding six steps."

Use the following syntax:

const stringVariable = `This is a ${variable}`;

11.

Define an object named userPrompt that will hold the instructions for the AI. This object should include a key role with the value "user".

Additionally, add another key content where the value is a concatenation of the strings userContent1, userContent2, and userContent3.

When combining these strings, ensure you use appropriate punctuation and spacing to distinguish each part clearly. You may use the newline character "\n" to insert a line break between each string, which will help maintain a readable and organized format for the prompt.

Use the following syntax:

```
const objectName = {
  key1: 'value1',
```

```
  key2: `${string1}\n${string2}\n${string3}`
};
```

Make the Chat Completion

12.

Utilize the client variable to initiate a chat completion. Pass the following arguments to the function:

- Assign the model argument a value of either "gpt-3.5-turbo" or "gpt-4-turbo-preview", depending on which model you want to use.
- For the messages argument, provide an array that contains the object systemPrompt as its first item, followed by the object userPrompt as the second item.

Assign the return value of the chat completion to a variable called response.

Use the following syntax:

```
const response = await client.chat.completions.create({
   model:MODEL_STRING,
   messages:[prompt1, prompt2]
})
```

13.

Output the chat completion reply content to the terminal.

Use the following syntax:

```
console.log(response.choices[0].mesasage.content)
```

14.

The result of this process should be HTML code that is prepared for review and can then be published on a website.

Employing generative AI for automating such tasks significantly streamlines the content creation workflow, making it more efficient for producing various types of content.

script.js

```
import OpenAI from "openai";


const client = new OpenAI();


const userProfile = {

}


userProfile.dietaryRestrictions = 'Milk, Cheese, Red Meat, White Bread';


userProfile.cuisinePreferences = "Italian, Spaniard, French, Colombian";
```

```
userProfile.ingredientsAvailable = "Beef Meat, Chicken Meat, Lamb Meat, Pork Chops, Tomatoes, Vegetables, Olive Oil, Tuna, Fish";


const systemPrompt = {
  role: "system",
  content: "Generate an HTML code for a recipe blog that considers dietary restrictions, cuisine type, and ingredients."
}

const userContent1 = `I want to create a recipe blog post. Here are my dietary restrictions: ${userProfile.dietaryRestrictions}. My cuisine preferences include: ${userProfile.cuisinePreferences}. The ingredients I have available are: ${userProfile.ingredientsAvailable}`

const userContent2 = "Please provide a blog post with a title, description, ingredients, and instructions. Format the ingredients and instructions as follows: Ingredients should be bulleted, and instructions should be numbered."

const userContent3 = "The recipe must use only the listed ingredients and should result in a single blog post with instructions not exceeding six steps."

const userPrompt = {
  role: "user",
  content: `${userContent1}\n${userContent2}\n${userContent3}`
}

const response = await client.chat.completions.create({
  model:"gpt-3.5-turbo",
  messages: [systemPrompt, userPrompt]
});

console.log(response.choices[0].message.content)
```

Response

Here's an example of an HTML code for a recipe blog post that considers your dietary restrictions, cuisine preferences, and available ingredients:

```html
<!DOCTYPE html>
<html>
<head>
  <title>Delicious Mediterranean Fish Stew</title>
  <meta charset="UTF-8">
  <meta name="description" content="A healthy and flavorful Mediterranean fish stew recipe suitable for those with dietary restrictions.">
</head>
<body>
  <h1>Delicious Mediterranean Fish Stew</h1>
  <h2>Description:</h2>
  <p>
    Indulge in the rich flavors of Mediterranean cuisine with this hearty fish stew. Perfect for those with dietary restrictions, this recipe excludes milk, cheese, red meat, and white bread. With a combination of seasonal vegetables, tender fish, and aromatic herbs, this stew will transport your taste buds to the sunny coasts of Italy, Spain, France, and Colombia.
  </p>
  <h2>Ingredients:</h2>
  <ul>
    <li>1 lb Fish (Tuna or any white fish of your choice)</li>
    <li>2 tbsp Olive Oil</li>
    <li>1 Onion, chopped</li>
    <li>2 cloves Garlic, minced</li>
    <li>1 Bell Pepper, sliced</li>
    <li>1 Zucchini, sliced</li>
    <li>1 cup Tomatoes, diced</li>
    <li>2 cups Vegetable Broth</li>
    <li>Salt and Pepper to taste</li>
  </ul>
  <h2>Instructions:</h2>
  <ol>
    <li>In a large pot or Dutch oven, heat olive oil over medium heat.</li>
```

Sauté onion and garlic until fragrant and translucent.

Add bell pepper and zucchini, cook for 5 minutes until slightly tender.

Stir in the diced tomatoes and vegetable broth. Season with salt and pepper.

Bring the stew to a simmer and then gently add the fish. Cover and cook for about 10 minutes, or until the fish is cooked through and flakes easily with a fork.

Remove from heat and let the stew rest for a few minutes before serving. Enjoy with a side of crusty bread, if desired.

</body>

</html>

```

Feel free to customize this HTML code based on your preferences and aesthetic preferences for your blog.
```