

## Import necessary libraries

```
In [1]: import os  
import openai
```

## Retrieve API key from environment variables

This line sets the OpenAI API key by accessing it from the environment variable 'OPENAI\_API\_KEY'.

```
In [2]: openai.api_key = os.environ['OPENAI_API_KEY']
```

This code initializes an OpenAI client using the API key fetched from the environment variable 'OPENAI\_API\_KEY'.

```
In [3]: from openai import OpenAI  
client = OpenAI(  
    api_key=os.environ['OPENAI_API_KEY'], # this is also the default, it can be omitted  
)
```

## Function to create a prompt for generating a recipe based on a list of ingredients and a category

This function generates a prompt for creating a recipe, incorporating a list of ingredients and requiring a title that begins with 'Recipe Title: '.

```
In [4]: def create_dish_prompt(list_of_ingredients):  
    # Construct a prompt using the provided ingredients list and category  
    prompt = f"Create a detailed recipe using the following ingredients: {', '.join(list_of_ingredients)}.\n"\  
           + f"Additionally, assign a title starting with 'Recipe Title: ' to this recipe."  
    return prompt
```

## Entering the Ingredients

This code generates a recipe prompt using a list of ingredients for a dessert dish.

```
In [5]: recipe = create_dish_prompt(['strawberries', 'vanilla icecream', 'chocolate syrup', 'Whipped cream', 'sprinkles for garnish'])
```

```
In [6]: recipe
```

```
Out[6]: "Create a detailed recipe using the following ingredients: strawberries, vanilla icecream, chocolate syrup, Whipped cream, sprinkles for garnish.\nAdditionally, assign a title starting with 'Recipe Title: ' to this recipe."
```

```
In [7]: #i = ['eggs', 'bacon', 'bread']
```

```
In [8]: #' .join(i)
```

This code initiates a chat completion request using the generated recipe prompt within a chat context using the OpenAI GPT-3.5 Turbo model.

```
In [9]: # Use the prompt in the completion code by initiating a chat completion request
completion = client.chat.completions.create(
    messages=[
        {
            "role": "user",
            "content": recipe,
        }
    ],
    model="gpt-3.5-turbo",
)
```

```
In [10]: # Extract the content from the completion
content = completion.choices[0].message.content
```

```
In [11]: # Split the content into recipe title and detailed recipe
split_content = content.split('\n', 1) # Split into two parts based on the first newline
recipe_title = split_content[0].strip() # Extract the recipe title
detailed_recipe = split_content[1].strip() # Extract the detailed recipe content
```

```
In [12]: # Print the extracted parts
print(recipe_title)
print("\nDetailed Recipe:")
print(detailed_recipe)
```

Recipe Title: Strawberry Ice Cream Delight

Detailed Recipe:

Ingredients:

- 2 cups fresh strawberries, hulled and sliced
- 4 scoops vanilla ice cream
- 1/4 cup chocolate syrup
- 1 cup whipped cream
- Sprinkles for garnish

Instructions:

1. Wash the strawberries thoroughly and remove the stems. Slice them into thin pieces and set aside.
2. Take out four serving bowls or glasses and place one scoop of vanilla ice cream into each. Allow the ice cream to soften slightly, making it easier to combine with the strawberries.
3. Place a handful of sliced strawberries onto each scoop of ice cream. Gently mix the strawberries into the ice cream, ensuring they are evenly distributed.
4. Drizzle the chocolate syrup over the strawberry and ice cream mixture. Use approximately 1 tablespoon of syrup per serving, or adjust to your preference.
5. Next, add a generous dollop of whipped cream on top of each bowl. You can either use store-bought whipped cream or prepare it at home by whipping heavy cream until stiff peaks form.
6. Finally, sprinkle a handful of colorful sprinkles over the whipped cream for a fun and enticing garnish. Feel free to use your favorite sprinkles or choose ones that match the occasion or season.
7. Serve immediately, and enjoy this delightful and refreshing Strawberry Ice Cream treat!

Note: You can get creative with this recipe by adding other toppings like nuts, crushed cookies, or fresh mint leaves. You can also double or halve the ingredients based on the number of servings required.

```
In [13]: #import re
```

```
In [14]: #pip install Pillow
```

```
In [15]: print(recipe_title)
```

Recipe Title: Strawberry Ice Cream Delight

## Generating Image of the Cooked Recipe

This function uses OpenAI's DALL-E model to generate an image based on a provided recipe title, producing an image URL as output.

```
In [16]: def generate_dalle_image(recipe_title):
    response = client.images.generate(
        model="dall-e-3",
        prompt=recipe_title,
        size="1024x1024",
        quality="standard",
        n=1,
    )
    url = response.data[0].url if response.data else None
    return url
```

```
In [17]: url = generate_dalle_image(recipe_title)
```

## Displaying the image within the code

```
In [18]: from IPython.display import display, Image
```

This code generates an image using DALL-E based on a recipe title, displays the image if the URL exists, and prints a message if no image URL is generated.

```
In [19]: # Call the function and store the URL
url = generate_dalle_image(recipe_title,)
# Display the image if URL exists
if url:
    display(Image(url=url))
else:
    print("No image URL generated.")
```





