Import necessary libraries

In [32]:
```
import os
import openai
```

Retrieve API key from environment variables

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

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

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

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

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

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

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

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 Delight Sundae

Detailed Recipe:
Ingredients:
- 2 cups of fresh strawberries, washed and hulled
- 2 cups of vanilla ice cream
- 1/4 cup of chocolate syrup
- 1 cup of whipped cream
- Sprinkles (of your choice) for garnish

Instructions:
1. Take half of the strawberries and cut them into small slices. Set aside the remaining strawberries for later use.

2. In serving bowls or mason jars, start by layering a small scoop of vanilla ice cream at the bottom.

3. Add a generous amount of sliced strawberries on top of the ice cream layer, ensuring they are evenly distributed.

4. Drizzle a tablespoon of chocolate syrup over the strawberry layer.

5. Repeat steps 2-4 to create another layer of vanilla ice cream, strawberries, and chocolate syrup.

6. Top off the sundae with another scoop of vanilla ice cream.

7. Drizzle the remaining chocolate syrup over the final layer of ice cream.

8. Cut the reserved strawberries into halves and place them on the edge of the sundae bowls for an appealing visual presentation.

9. Finish off by applying a dollop of whipped cream in the center of each sundae.

10. Sprinkle a generous amount of your preferred sprinkles over the whipped cream for an extra burst of color and texture.

11. Serve immediately and enjoy this heavenly Strawberry Delight Sundae!

Note: You may also consider adding chopped nuts, such as almonds or walnuts, for added crunch and flavor.
```

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

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

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

```
Recipe Title: Strawberry Delight Sundae
```

Generating Image of the Cooked Recipe

In [27]:
```
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 [28]:
```
url = generate_dalle_image(recipe_title)
```

Displaying the image within the code

In [29]:
```
from IPython.display import display, Image
```

In [31]:
```
# 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.")
```


STRAWBERRY DELIGHT
SUNDAE