Grocery assistant application

Title:

Grocery assistant application using python.

Introduction:

The aim of this project is to develop a user-friendly application that serves as a virtual grocery shopping assistant. The application will help users streamline their grocery shopping experience, making it more efficient and organized. It will address the daily life problem of time-consuming and disorganized grocery shopping trips.

Implementation:

The application will be developed using the Python programming language. It will utilize various libraries and frameworks, such as barcode scanning libraries and map APIs for store location services. The user interface will be designed to be intuitive and user-friendly, with a focus on simplicity and ease of use.

Algorithm:

➢ The code you provided defines three classes: Product, ShoppingList, and Store. It also creates instances of these classes and demonstrates their usage.

➢ The Product class represents a product with a name and a price. The ShoppingList class represents a list of items, where you can add and remove items, display the items in the list, and calculate the total price of the items. The Store class represents a store with a name and a list of products. You can add products to the store and display the products it offers.

➢ In the sample data section, several instances of the Product class are created, representing different grocery items. An instance of the Store class

is created, representing a grocery store. Products are added to the store using the add_product method, and then the products are displayed using the display_products method.

➢ After that, a shopping list object, my_shopping_list, is created using the ShoppingList class. The user is prompted to enter product names to add to the shopping list. The program checks if the entered product name matches any product in the store and adds it to the shopping list if found. If the product is not found, an error message is displayed.

➢ Finally, the shopping list is displayed using the display_items method, and the total price of the items in the shopping list is calculated and displayed using the calculate_total_price method.

➢ Please note that there are some typos in the code. The initialization methods (__init__) of the classes are defined with a single underscore instead of two underscores before and after "init". Therefore, the initialization methods are not being called. To fix this, you should change _init_ to __init__ in all three classes.

Program:

In [1]:
```
class Product:
    def __init__(self, name, price):
        self.name = name
        self.price = price

class ShoppingList:
    def __init__(self):
        self.items = []

    def add_item(self, item):
        self.items.append(item)

    def remove_item(self, item):
        self.items.remove(item)

    def display_items(self):
        if self.items:
            print("Shopping List:")
            for item in self.items:
                print(f"- {item.name}")
        else:
            print("Shopping List is empty.")

    def calculate_total_price(self):
        total_price = sum(item.price for item in self.items)
        print(f"Total Price: ${total_price:.2f}")

class Store:
    def __init__(self, name):
        self.name = name
        self.products = []

    def add_product(self, product):
        self.products.append(product)

    def display_products(self):
        print(f"{self.name} Products:")
        for product in self.products:
            print(f"- {product.name} (${product.price})")
```

```
apple = Product("Apple", 0.5)
banana = Product("Banana", 0.3)
milk = Product("Milk", 1.99)
juice = Product("juice",2.55)
bread = Product("bread",1.44)

grocery_store = Store("My Grocery Store")
grocery_store.add_product(apple)
grocery_store.add_product(banana)
grocery_store.add_product(milk)
grocery_store.add_product(juice)
grocery_store.add_product(bread)

grocery_store.display_products()

my_shopping_list = ShoppingList()

while True:
    item_name = input("Enter product name (or 'done' to finish): ")
    if item_name == "done":
        break
    product = None
    for store_product in grocery_store.products:
        if store_product.name.lower() == item_name.lower():
            product = store_product
            break

    if product:
        my_shopping_list.add_item(product)
    else:
        print(f"Product '{item_name}' not found in the store.")

my_shopping_list.display_items()
my_shopping_list.calculate_total_price()
```

Output:

```
My Grocery Store Products:
- Apple ($0.5)
- Banana ($0.3)
- Milk ($1.99)
- juice ($2.55)
- bread ($1.44)
Enter product name (or 'done' to finish): apple
Enter product name (or 'done' to finish): milk
Enter product name (or 'done' to finish): bread
Enter product name (or 'done' to finish): done
Shopping List:
- Apple
- Milk
- bread
Total Price: $3.93
```

Expected Outcome:

The final outcome of this project will be a fully functional grocery shopping assistant application that simplifies and enhances the grocery shopping experience for users. The application will provide a convenient and organized way to manage shopping lists, compare prices, locate stores, and receive notifications, ultimately saving users time and effort