**Overview of the Grocery Store Management System Project**

**Purpose**

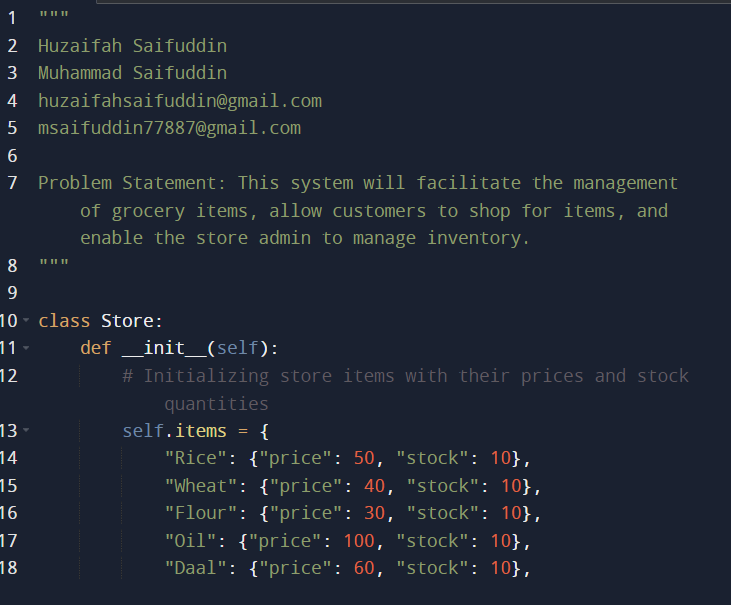
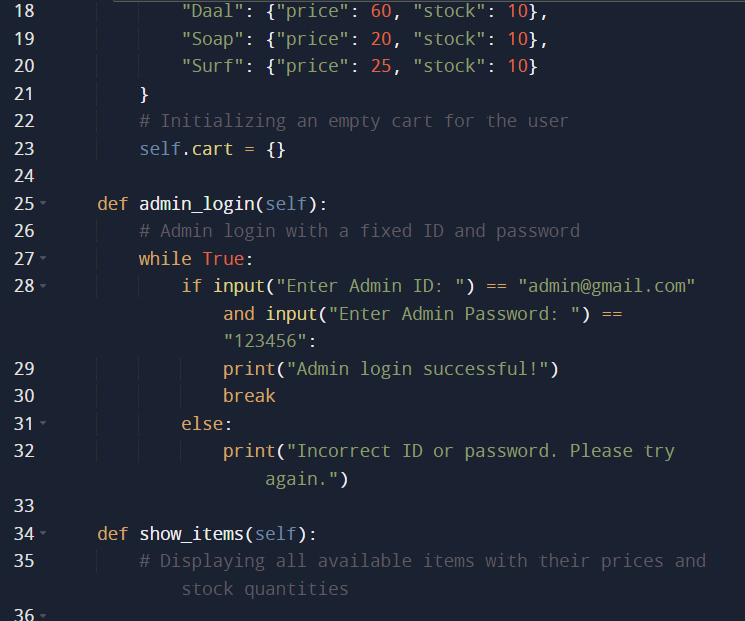
This project aims to create a simple and efficient system for managing a grocery store. It will help the store admin to manage inventory and allow customers to shop for items. The system provides different functionalities for both admin and customers to ensure smooth operations.

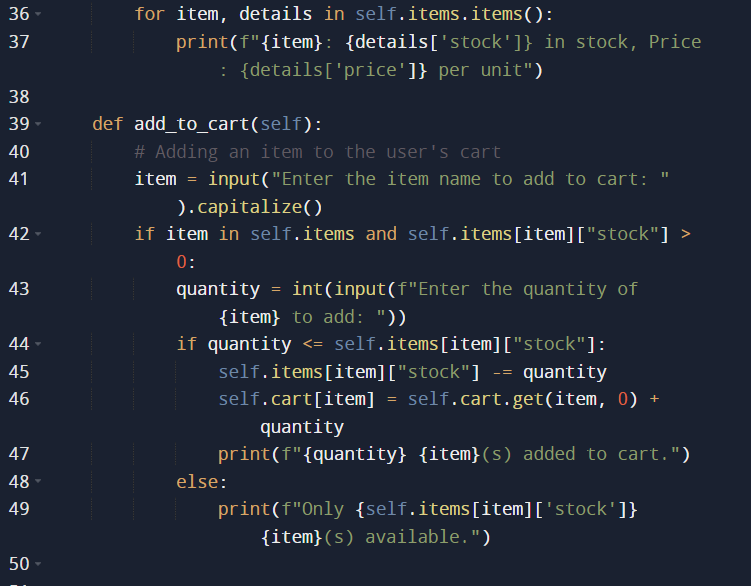
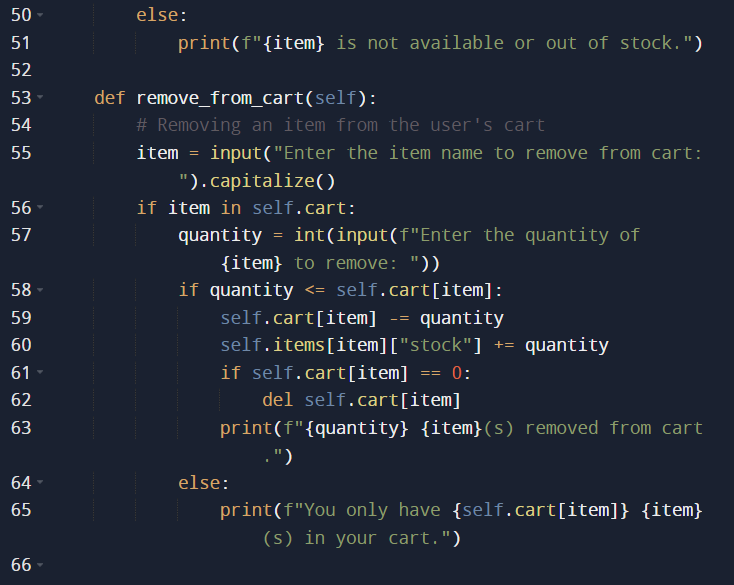
**User Roles**

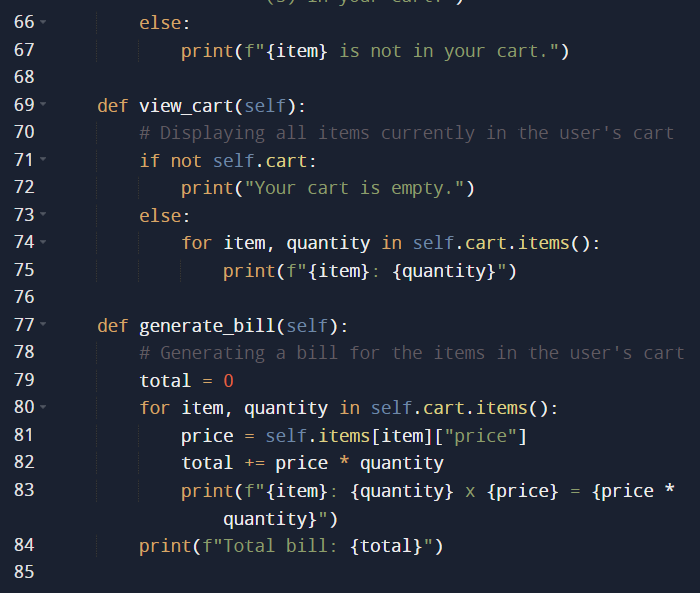
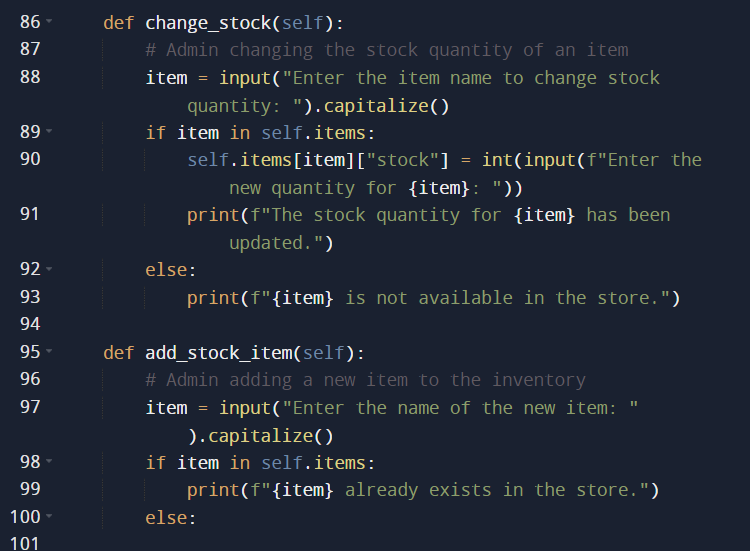
1. **Admin:**
   * Has the ability to log in using a specific ID and password.
   * Can view the current inventory, update stock quantities, add new items, and remove items from the store.
2. **Customer:**
   * Can view the available items in the store.
   * Can add items to their cart, remove items from the cart, view their cart, and proceed to checkout to generate a bill.

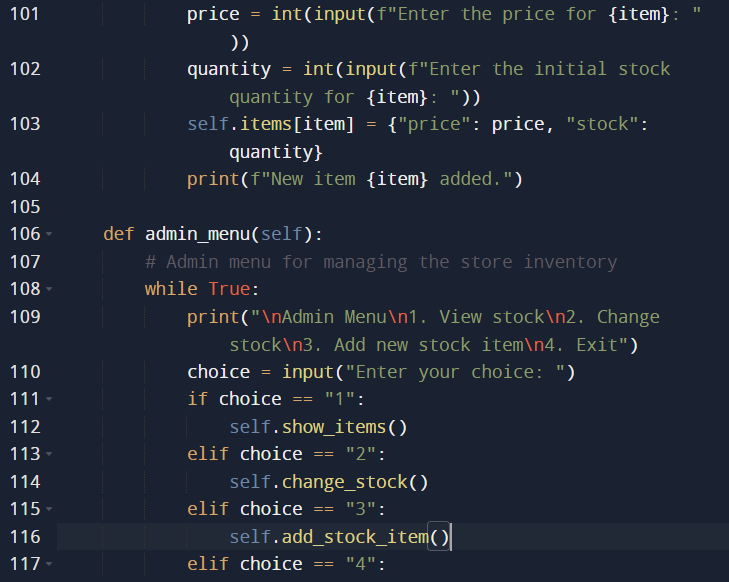
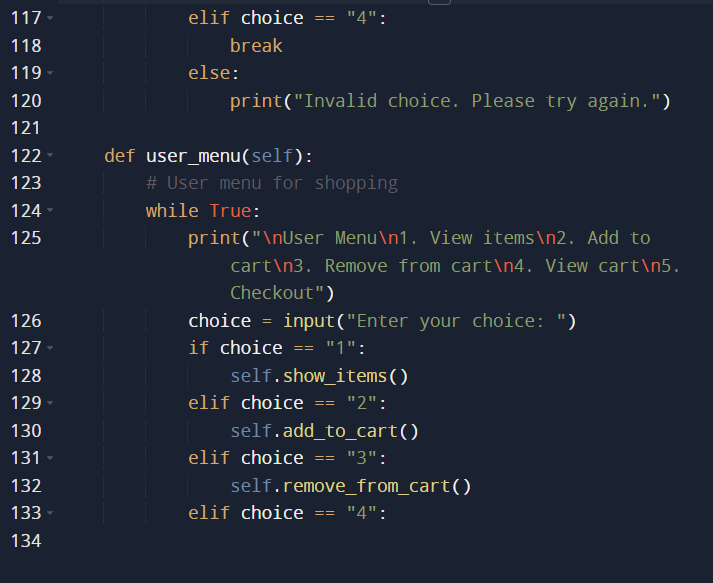
**Key Functionalities**

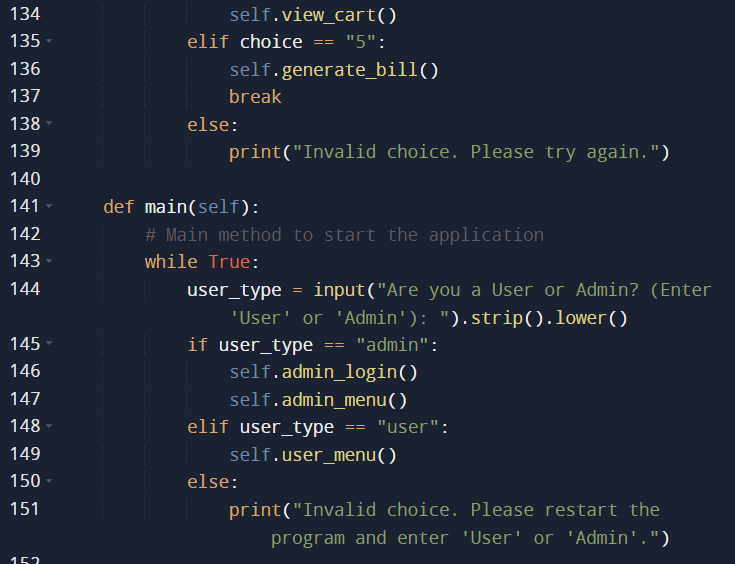
1. **Admin Functions:**
   * **Login:** Admin must log in with a fixed ID ("admin@gmail.com") and password ("123456").
   * **View Stock:** Admin can see all available items along with their prices and stock quantities.
   * **Change Stock:** Admin can modify the stock quantity of existing items.
   * **Add New Item:** Admin can add a new item to the inventory with specified price and quantity.
   * **Remove Item:** Admin can remove an item from the inventory.
2. **Customer Functions:**
   * **View Items:** Customers can see all available items along with their prices and stock quantities.
   * **Add to Cart:** Customers can add a specified quantity of an item to their cart.
   * **Remove from Cart:** Customers can remove a specified quantity of an item from their cart.
   * **View Cart:** Customers can see all items currently in their
   * **Checkout:** Customers can generate a bill for the items in their cart, showing the total amount due.











**explanation of Code Logic and Design**

**Class Definition**

* **Class Store:** The main class that encapsulates all the functionalities of the grocery store. It manages items, the cart, and provides methods for both admin and customer interactions.

**Initialization**

* **\_\_init\_\_ method:**
  + Initializes the store with a dictionary of items, where each item has a price and stock quantity.
  + Initializes an empty dictionary to represent the customer's cart.

**Admin Functions**

* **admin\_login method:**
  + Prompts the admin to enter an ID and password.
  + Continuously asks for credentials until the correct ones are provided.

**show\_items method:**

* Displays all items in the store with their prices and stock quantities.

**change\_stock method:**

* Allows the admin to change the stock quantity of an existing item.

**add\_stock\_item method:**

* Allows the admin to add a new item to the store inventory.

**admin\_menu method:**

* Provides the admin with a menu to view stock, change stock, add new items, or exit the admin menu.

**Customer Functions**

* **add\_to\_cart method:**
  + Allows the customer to add a specified quantity of an item to their cart, if the item is in stock.

**remove\_from\_cart method:**

* Allows the customer to remove a specified quantity of an item from their cart

**view\_cart method:**

* Displays all items currently in the customer's cart

**generate\_bill method:**

* Generates a bill for the items in the customer's cart, showing the total amount due.

**user\_menu method:**

* Provides the customer with a menu to view items, add to cart, remove from cart, view cart, or checkout.

**Main Method**

* **main method:**
  + The entry point of the application that continuously prompts the user to choose between admin and user modes.

**if \_\_name\_\_ == "\_\_main\_\_":**

* Ensures the Store class is instantiated and the main method is called when the script is executed.