Project Title: Create a shopping application using Python

Project Objective:

Create a shopping application or e-commerce application in Python.

Note - Scroll down to see the technical documentation and output screenshot.

Problem Statement:

You have to develop a shopping application or e-commerce application which has login and public login features on the Python platform. The applications that have been developed should also include categories, such as 3–4 for footwear, clothing, electronics, etc. It should be possible to add and update categories in the application. Additionally, it must contain a feature that allows you to add or remove items from your cart. Finally, the program needs to support a variety of payment options, including UPI and debit cards. This should be only backend implementation, and UX/UI and database connectivity are not required.

Instructions to Perform:

- •A welcome message should initially be displayed in the e-commerce application, such as "Welcome to the Demo Marketplace".
- •User login and admin login should be created once the code for the welcome message has been written. For the creation of demo login and admin login, demo databases for those should be created for the user and admin verification, and session id creation.
- •After databases are created, create admin and user logins. It is necessary to construct a sample product catalog with three to four product categories, such as Boots, Coats, Jackets, and Caps. The product id, name, category id, and price should all be present for each item in the dummy database of the catalog. Both users and administrators can view the catalog.
- •User login rights include the ability to view cart contents, add items to carts, and remove items from carts. The user should be able to add items or delete items in the cart using session id, product id, and quantity.
- •Next, the program should provide demo payment checkout options, like Net banking, PayPal, UPI, etc. After selecting the payment option, a checkout message like, "Your order is successfully placed" or "You will be shortly redirected to the portal for Unified Payment Interface to make a payment of Rs. 2000", etc., should be displayed.
- •Additionally, the admin can only log in using his credentials, and if the admin attempts to log in using another set of credentials, an error notice must appear.

•Admin should not be able to interfere with any of the functions that the user can perform, as discussed above. An error should appear if the admin tries to carry out those tasks.

•Furthermore, using the previously selected attributes, the admin should be able to add new products to the catalog. Additionally, the program needs to make it possible for an existing product to be modified using an admin session id.

•The admin should then have the ability to remove any existing products from the already-generated catalog.

•Lastly, understanding the dynamic demands of the market, the admin should be able to add a new category of product and delete the prevailing category of product from the catalog.

•Users should not be able to take advantage of any of the admin's rights, as described above.

***************Shopping App technical document ***************

This document outlines the structure and functionality of a shopping application developed in Python. The application allows users and administrators to perform various tasks related to shopping, product management, including user registration, and order processing.

Features –

1.**Start**

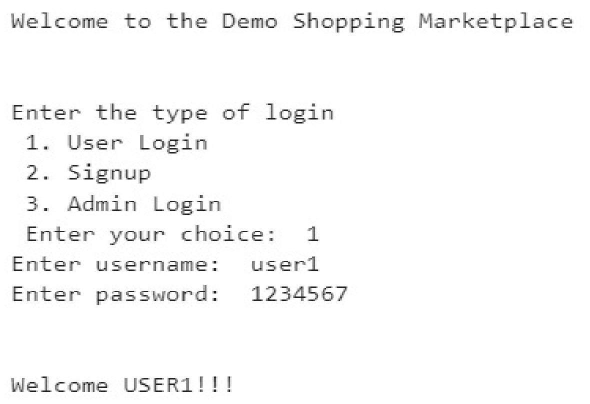
Function: start()

Description: Entry point of the application.

Options:

- Sign Up (signUp())
- User Login (userLogin())
- Admin Login (adminLogin())

*******JUPYTER OUTPUT*******



```
Welcome to the Demo Shopping Marketplace


Enter the type of login
 1. User Login
 2. Signup
 3. Admin Login
 Enter your choice:  1
Enter username:  user1
Enter password:  1234567


Welcome USER1!!!
```

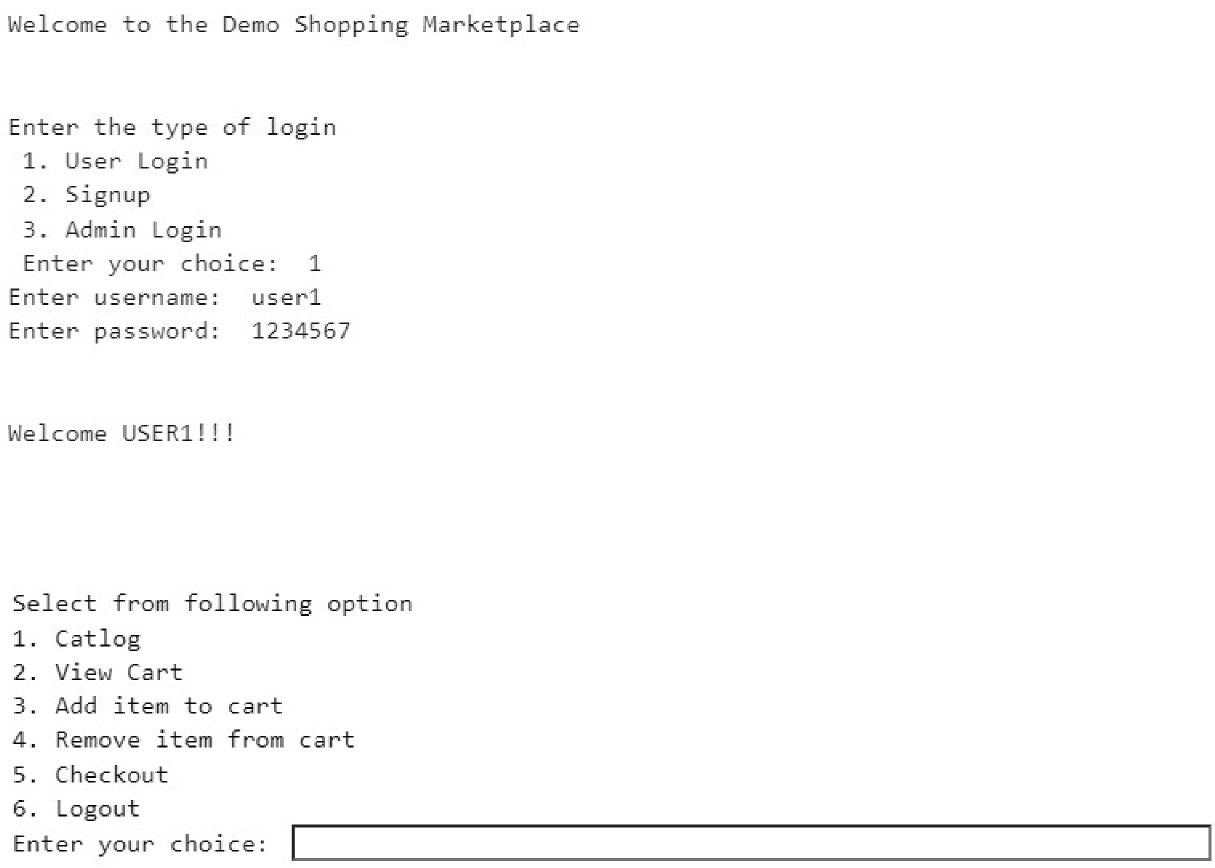
2. **Signup**

- **Function**: signUp()
- **Description**: Allows new users to create an account.
- **Inputs**: Username, password.
- **Storage**: User credentials are stored in a dictionary.

3. **User Login**

- **Function**: userLogin()
- **Description**: Authenticates user credentials.
- **Inputs**: Username and password.
- **Flow**:
    - Validates credentials against the users dictionary.
    - On success, redirects to user menu (userMenu()).

*******JUPYTER OUTPUT*******



```
Welcome to the Demo Shopping Marketplace


Enter the type of login
 1. User Login
 2. Signup
 3. Admin Login
 Enter your choice:  1
Enter username:  user1
Enter password:  1234567


Welcome USER1!!!




 Select from following option
 1. Catlog
 2. View Cart
 3. Add item to cart
 4. Remove item from cart
 5. Checkout
 6. Logout
 Enter your choice:
```

4. **User Menu**

- **Function**: userMenu()
- **Description**: Displays options for logged-in users.
- **Options**:
  1. View Catalog (viewCatalog())
  2. View Cart (viewCart())
  3. Add Item to Cart (addToCart())
  4. Remove Item from Cart (removeFromCart())
  5. Checkout (checkout())
  6. Logout (logout())

  5. Admin Login

5. **Admin Login**

- **Function**: adminLogin()
- **Description**: Authenticates admin credentials.
- **Inputs**: Admin username and password.
- **Flow**:
  - Validates credentials against the users dictionary.
  - On success, redirects to admin menu (adminMenu()).

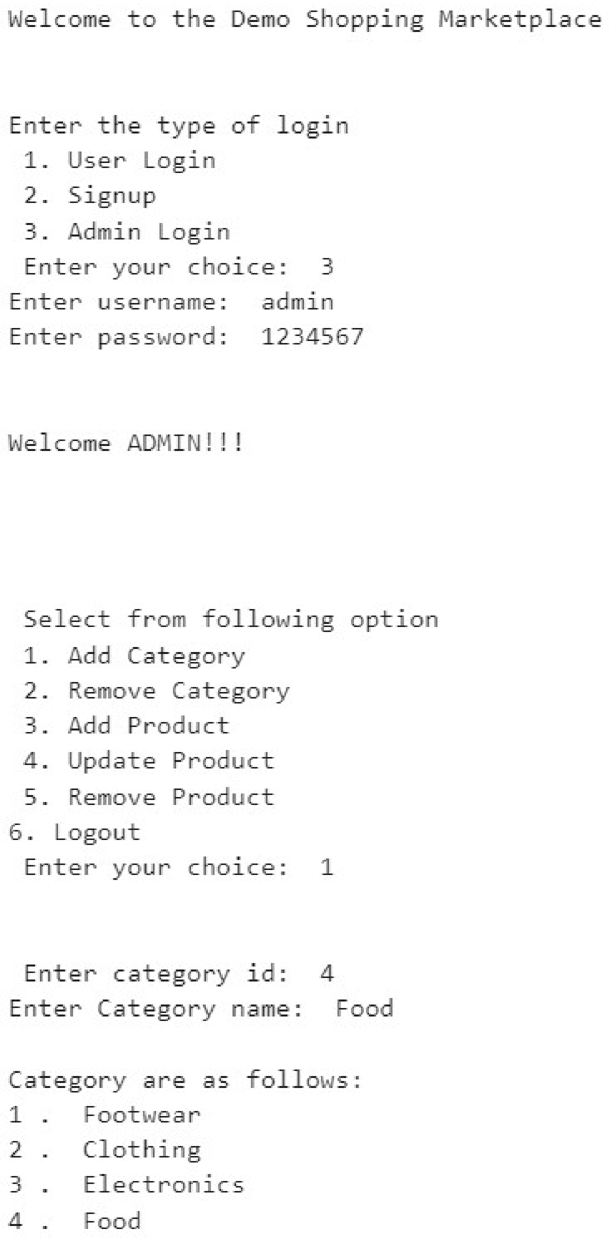
6. **Admin Menu**

- **Function**: adminMenu()
- **Description**: Displays options for logged-in administrators.
- **Options**:
  1. Add Category (addCategory())
  2. Remove Category (removeCategory())
  3. Add Product (addProduct())
  4. Remove Product (removeProduct())
  5. Update Product (updateProduct())
  6. Logout (logout())
  7. Add Category

7. **Add Category**

- **Function**: addCategory()
- **Description**: Allows admin to add a new product category.
- **Inputs**: Category ID, category name.

*******JUPYTER OUTPUT*******



```
Welcome to the Demo Shopping Marketplace


Enter the type of login
 1. User Login
 2. Signup
 3. Admin Login
 Enter your choice:  3
Enter username:  admin
Enter password:  1234567


Welcome ADMIN!!!




 Select from following option
 1. Add Category
 2. Remove Category
 3. Add Product
 4. Update Product
 5. Remove Product
6. Logout
 Enter your choice:  1


 Enter category id:  4
Enter Category name:  Food

Category are as follows:
1 .  Footwear
2 .  Clothing
3 .  Electronics
4 .  Food
```

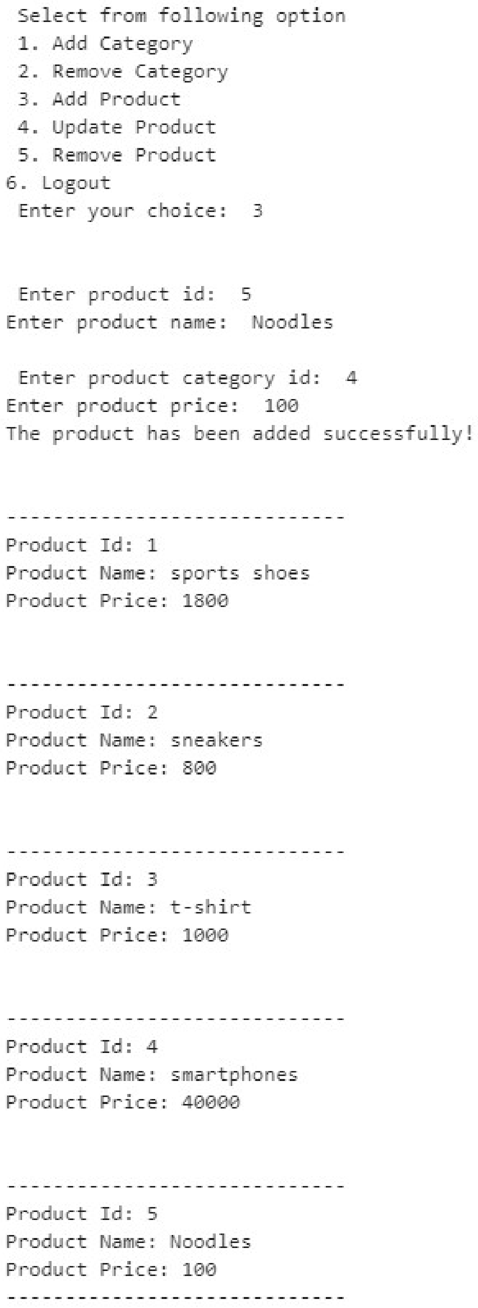
8. **View Category**
   - **Function**: viewCat()
   - **Description**: Displays all categories in the application.

9. **Remove Category**
   - **Function**: removeCategory()
   - **Description**: Allows admin to remove a category and all associated products

10. **Add Product**

- **Function**: addProduct()
- **Description**: Allows admin to add a new product to a category.
- **Inputs**: Product ID, product name, product price, category ID

*******JUPYTER OUTPUT*******



```
 Select from following option
 1. Add Category
 2. Remove Category
 3. Add Product
 4. Update Product
 5. Remove Product
6. Logout
 Enter your choice:  3


 Enter product id:  5
Enter product name:  Noodles

 Enter product category id:  4
Enter product price:  100
The product has been added successfully!


-----------------------------
Product Id: 1
Product Name: sports shoes
Product Price: 1800


-----------------------------
Product Id: 2
Product Name: sneakers
Product Price: 800


-----------------------------
Product Id: 3
Product Name: t-shirt
Product Price: 1000


-----------------------------
Product Id: 4
Product Name: smartphones
Product Price: 40000


-----------------------------
Product Id: 5
Product Name: Noodles
Product Price: 100
-----------------------------
```

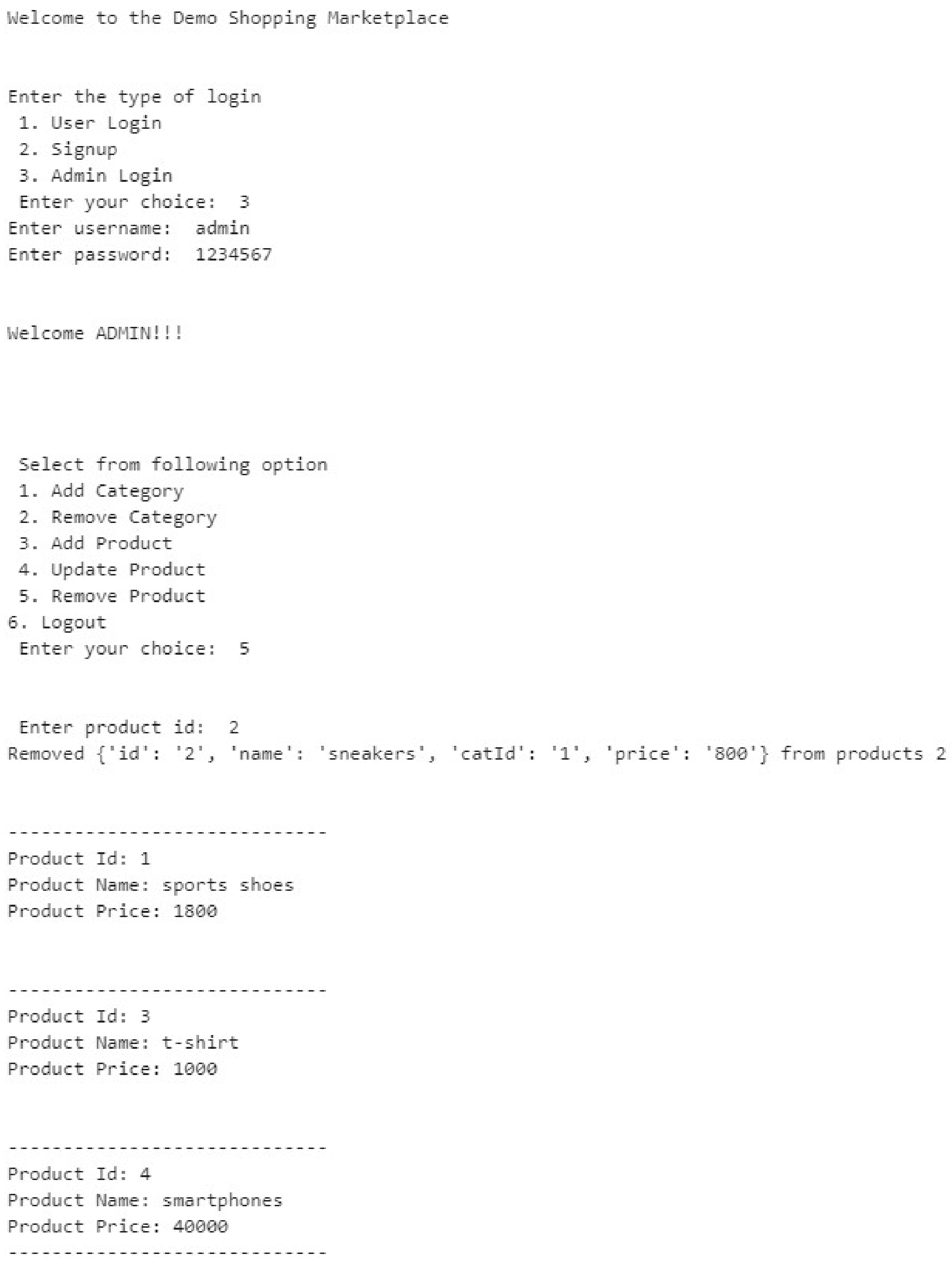
11. **View Product**

- **Function**: viewProduct()
- **Description**: Displays all products within a specific category.

12. **Remove Product**

- **Function**: removeProduct()
- **Description**: Allows admin to remove a specific product from the application.

*******JUPYTER OUTPUT*******



```
Welcome to the Demo Shopping Marketplace


Enter the type of login
 1. User Login
 2. Signup
 3. Admin Login
 Enter your choice:  3
Enter username:  admin
Enter password:  1234567


Welcome ADMIN!!!




 Select from following option
 1. Add Category
 2. Remove Category
 3. Add Product
 4. Update Product
 5. Remove Product
6. Logout
 Enter your choice:  5


 Enter product id:  2
Removed {'id': '2', 'name': 'sneakers', 'catId': '1', 'price': '800'} from products 2


-----------------------------
Product Id: 1
Product Name: sports shoes
Product Price: 1800


-----------------------------
Product Id: 3
Product Name: t-shirt
Product Price: 1000


-----------------------------
Product Id: 4
Product Name: smartphones
Product Price: 40000
-----------------------------
```

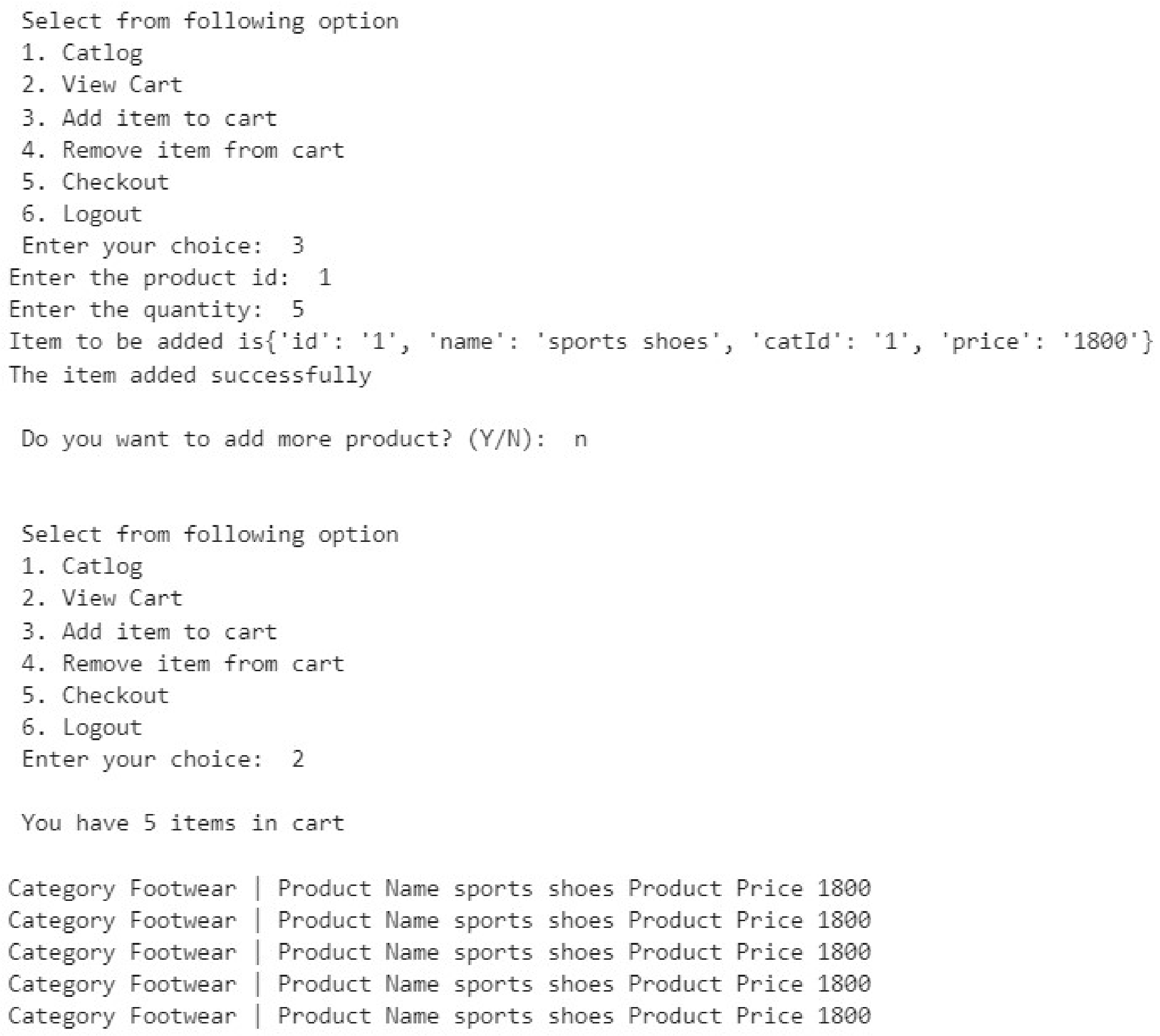
13. **Update Product**

- **Function**: updateProduct()
- **Description**: Allows admin to update product details.

14. **Add Item to Cart**

- **Function**: addToCart()
- **Description**: Allows users to add items to their shopping cart.
- **Inputs**: Category, product ID, quantity, price.

*******JUPYTER OUTPUT*******

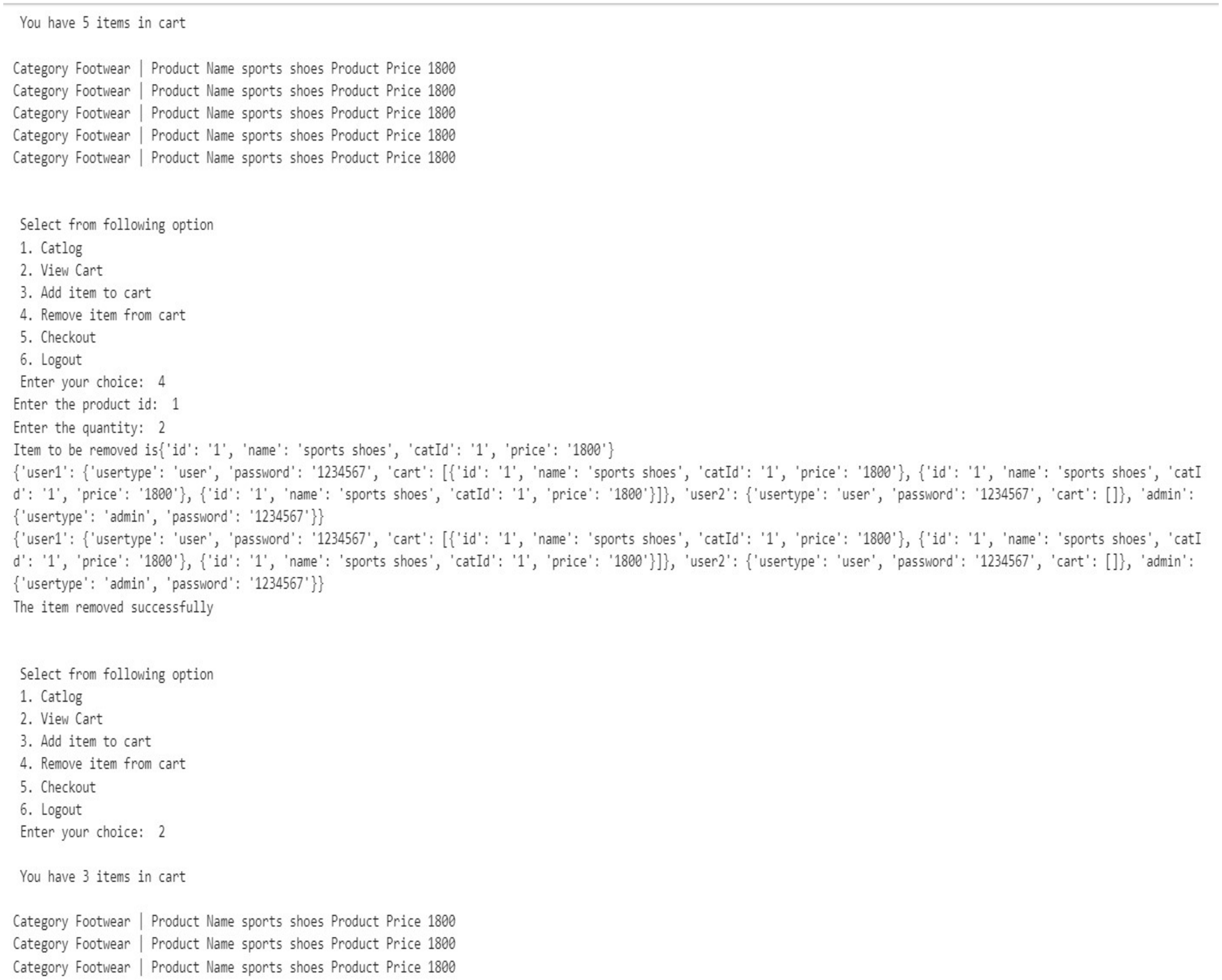


```
 Select from following option
 1. Catlog
 2. View Cart
 3. Add item to cart
 4. Remove item from cart
 5. Checkout
 6. Logout
 Enter your choice:  3
Enter the product id:  1
Enter the quantity:  5
Item to be added is{'id': '1', 'name': 'sports shoes', 'catId': '1', 'price': '1800'}
The item added successfully

 Do you want to add more product? (Y/N):  n


 Select from following option
 1. Catlog
 2. View Cart
 3. Add item to cart
 4. Remove item from cart
 5. Checkout
 6. Logout
 Enter your choice:  2

 You have 5 items in cart

Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
```

15. **Remove Item from Cart**

- **Function**: removeFromCart()
- **Description**: Allows users to remove items from their cart.
- **Inputs**: Product ID, quantity.

*******JUPYTER OUTPUT*******

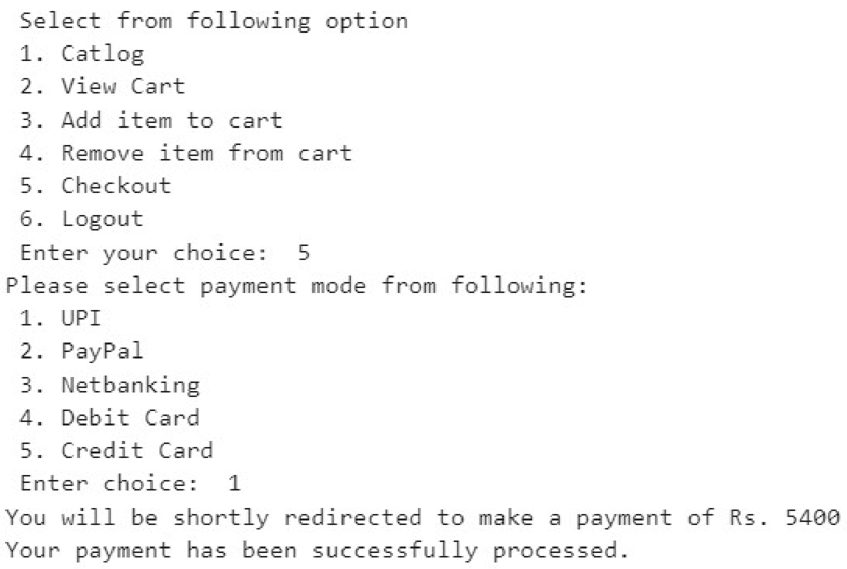


```
 You have 5 items in cart

Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800


 Select from following option
 1. Catlog
 2. View Cart
 3. Add item to cart
 4. Remove item from cart
 5. Checkout
 6. Logout
 Enter your choice:  4
Enter the product id:  1
Enter the quantity:  2
Item to be removed is{'id': '1', 'name': 'sports shoes', 'catId': '1', 'price': '1800'}
{'user1': {'usertype': 'user', 'password': '1234567', 'cart': [{'id': '1', 'name': 'sports shoes', 'catId': '1', 'price': '1800'}, {'id': '1', 'name': 'sports shoes', 'catI
d': '1', 'price': '1800'}, {'id': '1', 'name': 'sports shoes', 'catId': '1', 'price': '1800'}]}, 'user2': {'usertype': 'user', 'password': '1234567', 'cart': []}, 'admin':
{'usertype': 'admin', 'password': '1234567'}}
{'user1': {'usertype': 'user', 'password': '1234567', 'cart': [{'id': '1', 'name': 'sports shoes', 'catId': '1', 'price': '1800'}, {'id': '1', 'name': 'sports shoes', 'catI
d': '1', 'price': '1800'}, {'id': '1', 'name': 'sports shoes', 'catId': '1', 'price': '1800'}]}, 'user2': {'usertype': 'user', 'password': '1234567', 'cart': []}, 'admin':
{'usertype': 'admin', 'password': '1234567'}}
The item removed successfully


 Select from following option
 1. Catlog
 2. View Cart
 3. Add item to cart
 4. Remove item from cart
 5. Checkout
 6. Logout
 Enter your choice:  2

 You have 3 items in cart

Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
Category Footwear | Product Name sports shoes Product Price 1800
```

16. **Checkout**

- **Function**: checkout()
- **Description**: Facilitates the checkout process for users.
- **Inputs**: Payment mode.
- **Outputs**: Total amount to be paid.

*******JUPYTER OUTPUT*******

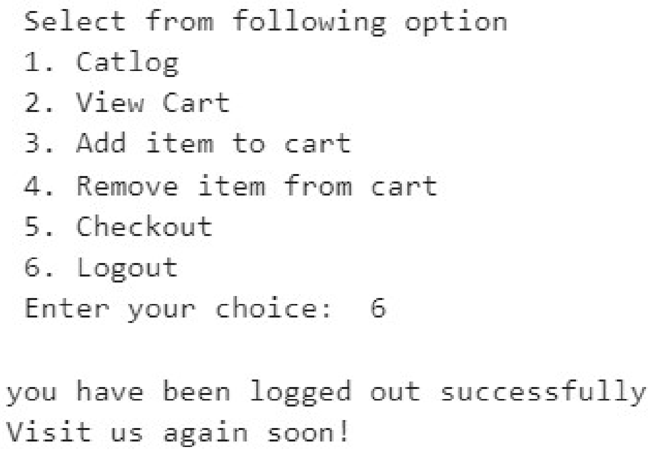


```
 Select from following option
 1. Catlog
 2. View Cart
 3. Add item to cart
 4. Remove item from cart
 5. Checkout
 6. Logout
 Enter your choice:  5
Please select payment mode from following:
 1. UPI
 2. PayPal
 3. Netbanking
 4. Debit Card
 5. Credit Card
 Enter choice:  1
You will be shortly redirected to make a payment of Rs. 5400
Your payment has been successfully processed.
```

17. **Logout**

- **Function**: logout()
- **Description**: Logs the user out of the application.

*******JUPYTER OUTPUT*******



```
 Select from following option
 1. Catlog
 2. View Cart
 3. Add item to cart
 4. Remove item from cart
 5. Checkout
 6. Logout
 Enter your choice:  6

you have been logged out successfully
Visit us again soon!
```