Creating a Shopping App Using Python

Phase-End Project Problem Statement

simplilearn

Get Certified. Get Ahead.

Phase-End Project: Creating a Shopping App Using Python

Problem Scenario: 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.

Guidelines:

- 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.

Project Code with Output

Code:

```
 def welcome_message():
     print("Welcome to the Demo Marketplace"


 users_db = {
    'user1': {'password': 'pass1', 'cart': []},
    'user2': {'password': 'pass2', 'cart': []}
}


admins_db = {
    'admin': {'password': 'adminpass'}
}


categories  = {
    1: 'Footwear',
    2: 'Clothing',
    3: 'Electronics'
}
```

```
products = [
    {'id': 101, 'name': 'Boots', 'category_id': 1, 'price': 1000},
    {'id': 102, 'name': 'Jackets', 'category_id': 2, 'price': 2000},
    {'id': 103, 'name': 'Coats', 'category_id': 3, 'price': 3000},
    {'id': 104, 'name': 'Caps', 'category_id': 4, 'price': 500}
]
session_data = {}
def login(username, password, is_admin=False):
    if is_admin:
        if username in admins_db and admins_db[username]['password']  == password:
            session_data['admin'] = username
            print("Admin logged in successfully")
        else:
            print("Invalid admin credentials")
    else:
        if username in users_db and users_db[username]['password']  == password:
            session_data['user'] = username
            print(f"User {username} logged in successfully")
        else:
            return print("Invalid user credentials")
```

```
def view_catalog():
    for product in products:
        category_name = categories.get(product['category_id'],  "Unknown Category")
        print(f"ID: {product['id']}, Name: {product['name']}, Category: {category_name}, Price: {product['price']}")


def view_category():
    if not categories:
        print("No categories available.")
    else:
        print("Available Categories:")
        for category_id, category_name in categories.items():
            print(f"ID: {category_id}, Name: {category_name}")
```

```
def add_to_cart(product_id, quantity):
    if 'user' not in session_data:
        print("Login as user to add items to cart")
        return

    user_cart = users_db[session_data['user']]['cart']
    user_cart.append({'product_id': product_id, 'quantity': quantity})
    print(f"Added product {product_id} with quantity {quantity} to cart")

def view_cart():
    if 'user' not in session_data:
        print("Login as user to view cart")
        return

    user_cart = users_db[session_data['user']]['cart']
    if not user_cart:
        print("Cart is empty")
    else:
        for item in user_cart:
            # Provide a default value (None or a custom message) if product is not found
            product = next((p for p in products if p['id'] == item['product_id']), None)
            if product:
                print(f"Product: {product['name']}, Quantity: {item['quantity']}")
            else:
                print(f"Product with ID {item['product_id']} not found.")

def remove_from_cart(product_id):
    if 'user' not in session_data:
        print("Login as user to remove items from cart")
        return

    user_cart = users_db[session_data['user']]['cart']
    users_db[session_data['user']]['cart']  = [item for item in user_cart if item['product_id'] != product_id]
    print(f"Removed product {product_id} from cart")
```

```
def checkout(payment_method):
    if 'user' not in session_data:
        print("Login as user to checkout")
        return

    user_cart = users_db[session_data['user']]['cart']
    if not user_cart:
        print("Cart is empty")
        return

    total_amount = sum(next(p['price'] for p in products if p['id'] == item['product_id']) * item['quantity'] for item in user_cart)
    print(f"Your total amount is {total_amount}")
    print(f"You will be redirected to complete your payment using {payment_method}.")
    # Clear cart after checkout
    users_db[session_data['user']]['cart'] = []
```

```
def add_product(name, category_id, price):
    if 'admin' not in session_data:
        print("Login as admin to add products")
        return

    new_id = max(product['id'] for product in products) + 1
    products.append({'id': new_id, 'name': name, 'category_id': category_id, 'price': price})
    print(f"Product {name} added successfully")

def update_product(product_id, name=None, category_id=None, price=None):
    if 'admin' not in session_data:
        print("Login as admin to update products")
        return

    product = next(p for p in products if p['id'] == product_id)
    if name:
        product['name'] = name
    if category_id:
        product['category_id'] = category_id
    if price:
        product['price'] = price
    print(f"Product {product_id} updated successfully")
```

```
def remove_product(product_id):
    if 'admin' not in session_data:
        print("Login as admin to remove products")
        return

    global products
    products = [p for p in products if p['id'] != product_id]
    print(f"Product {product_id} removed successfully")


def add_category(category_name):
    if 'admin' not in session_data:
        print("Login as admin to add categories")
        return

    # Find the next available category ID
    new_category_id = max(categories.keys()) + 1 if categories else 1
    categories[new_category_id] = category_name
    print(f"Category '{category_name}' added successfully with ID {new_category_id}")
```

```
def main():
    welcome_message()
    while True:
        print("\n1. User Login\n2. Admin Login\n3. Exit")
        choice = input("Choose option: ")

        if choice == '1':
            username = input("Username:  ")
            password = input("Password: ")
            login(username, password, is_admin=False)

            while 'user' in session_data:
                print("\n1. View Catalog\n2. Add to Cart\n3. View Cart\n4. Remove from Cart\n5. Checkout\n6. Logout")
                user_choice = input("Choose option: ")

                if user_choice == '1':
                    view_catalog()
                elif user_choice == '2':
                    product_id = int(input("Enter Product ID: "))
                    quantity = int(input("Enter Quantity: "))
                    add_to_cart(product_id, quantity)
                elif user_choice == '3':
                    view_cart()

                elif user_choice == '4':

                    product_id = int(input("Enter Product ID to remove: "))

                    remove_from_cart(product_id)

                elif user_choice == '5':

                    payment_method = input("Enter Payment Method
(UPI/NetBanking/PayPal): ")

                    checkout(payment_method)

                elif user_choice == '6':

                    session_data.pop('user')

                    print("Logged out successfully")


        elif choice == '2':

            username = input("Admin Username: ")

            password = input("Password: ")

            login(username, password, is_admin=True)


            while 'admin' in session_data:

                print("\n1. View Catalog\n2. View Category\n3. Add Product\n4. Update
Product\n5. Remove Product\n6. Add Category\n7. Logout")

                admin_choice = input("Choose option: ")


                if admin_choice == '1':

                    view_catalog()
                elif admin_choice == '2':
                    view_category()
                elif admin_choice == '3':
                    name = input("Enter Product Name: ")
                    category_id = int(input("Enter Category ID: "))
                    price = float(input("Enter Price: "))
                    add_product(name, category_id, price)
                elif admin_choice == '4':
                    product_id = int(input("Enter Product ID to update: "))
                    name = input("Enter new Product Name (leave blank to skip): ")
                    category_id = input("Enter new Category ID (leave blank to skip): ")
                    price = input("Enter new Price (leave blank to skip): ")
                    update_product(product_id, name or None, int(category_id) if category_id else None, float(price) if price else None)
                elif admin_choice == '5':
                    product_id = int(input("Enter Product ID to remove: "))
                    remove_product(product_id)
                elif admin_choice == '6':
                    category_name = input("Enter new category name: ")
                    if category_name != "":
                           add_category(category_name)
                    else:
                        return print('Please enter category name')
                elif admin_choice == '7':
                    session_data.pop('admin')
                    print("Logged out successfully")

        elif choice == '3':
            break

if __name__ == '__main__':
    main()
```

Admin Output:

```
 Welcome to the Demo Marketplace
1. User Login
2. Admin Login
3. Exit
Choose option:  2
Admin Username:  admin
Password:  adminpass
Admin logged in successfully

1. View Catalog
2. View Category
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
```

```
7. Logout
Choose option:  1
ID: 101, Name: Boots, Category: Footwear, Price: 1000
ID: 102, Name: Jackets, Category: Clothing, Price: 2000
ID: 103, Name: Coats, Category: Electronics, Price: 3000
ID: 104, Name: Caps, Category: Unknown Category, Price: 500

1. View Catalog
2. View Category
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
7. Logout
Choose option:  3
Enter Product Name:  Shoes
Enter Category ID:  1
Enter Price:  5000
Product Shoes added successfully

1. View Catalog
2. View Category
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
7. Logout
Choose option:  1
```

```
ID: 101, Name: Boots, Category: Footwear, Price: 1000
ID: 102, Name: Jackets, Category: Clothing, Price: 2000
ID: 103, Name: Coats, Category: Electronics, Price: 3000
ID: 104, Name: Caps, Category: Unknown Category, Price: 500
ID: 105, Name: Shoes, Category: Footwear, Price: 5000.0

1. View Catalog
2. View Category
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
7. Logout
Choose option:  2
Available Categories:
ID: 1, Name: Footwear
ID: 2, Name: Clothing
ID: 3, Name: Electronics

1. View Catalog
2. View Category
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
7. Logout
Choose option:  4
Enter Product ID to update:  104
Enter new Product Name (leave blank to skip):

Enter new Category ID (leave blank to skip):  2

Enter new Price (leave blank to skip):

Product 104 updated successfully



1. View Catalog

2. View Category

3. Add Product

4. Update Product

5. Remove Product

6. Add Category

7. Logout

Choose option:  1

ID: 101, Name: Boots, Category: Footwear, Price: 1000

ID: 102, Name: Jackets, Category: Clothing, Price: 2000

ID: 103, Name: Coats, Category: Electronics, Price: 3000

ID: 104, Name: Caps, Category: Clothing, Price: 500

ID: 105, Name: Shoes, Category: Footwear, Price: 5000.0



1. View Catalog

2. View Category

3. Add Product

4. Update Product

5. Remove Product

6. Add Category

7. Logout

Choose option:  5

Enter Product ID to remove:  105
```

```
Product 105 removed successfully

1. View Catalog
2. View Category
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
7. Logout
Choose option:  1
ID: 101, Name: Boots, Category: Footwear, Price: 1000
ID: 102, Name: Jackets, Category: Clothing, Price: 2000
ID: 103, Name: Coats, Category: Electronics, Price: 3000
ID: 104, Name: Caps, Category: Clothing, Price: 500

1. View Catalog
2. View Category
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
7. Logout
Choose option:  6
Enter new category name Toys
Category 'Toys' added successfully with ID 4

1. View Catalog
2. View Category
```

```
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
7. Logout
Choose option:  3
Enter Product Name:  Toy car
Enter Category ID:  4
Enter Price:  200
Product Toy car added successfully


1. View Catalog
2. View Category
3. Add Product
4. Update Product
5. Remove Product
6. Add Category
7. Logout
Choose option:  1
ID: 101, Name: Boots, Category: Footwear, Price: 1000
ID: 102, Name: Jackets, Category: Clothing, Price: 2000
ID: 103, Name: Coats, Category: Electronics, Price: 3000
ID: 104, Name: Caps, Category: Clothing, Price: 500
ID: 105, Name: Toy car, Category: Toys, Price: 200.0


1. View Catalog
2. View Category
3. Add Product
```

```
4. Update Product

5. Remove Product

6. Add Category

7. Logout

Choose option:  2

Available Categories:

ID: 1, Name: Footwear

ID: 2, Name: Clothing

ID: 3, Name: Electronics

ID: 4, Name: Toys


1. View Catalog

2. View Category

3. Add Product

4. Update Product

5. Remove Product

6. Add Category

7. Logout

Choose option:  6

Enter new category name

Please enter category name


1. View Catalog

2. View Category

3. Add Product

4. Update Product

5. Remove Product

6. Add Category
```

```
7. Logout

Choose option:  7

Logged out successfully



1. User Login

2. Admin Login

3. Exit

Choose option:  4



1. User Login

2. Admin Login

3. Exit

Choose option:
```

User Output:

```
Welcome to the Demo Marketplace


1. User Login
2. Admin Login
3. Exit
Choose option:  1
Username:  user1
Password:  pass1
User user1 logged in successfully


1. View Catalog
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Logout
Choose option:  1
ID: 101, Name: Boots, Category: Footwear, Price: 1000
ID: 102, Name: Jackets, Category: Clothing, Price: 2000
ID: 103, Name: Coats, Category: Electronics, Price: 3000
ID: 104, Name: Caps, Category: Unknown Category, Price: 500


1. View Catalog
2. Add to Cart
3. View Cart
4. Remove from Cart
```

```
5. Checkout
6. Logout
Choose option:  2
Enter Product ID:  101
Enter Quantity:  2
Added product 101 with quantity 2 to cart


1. View Catalog
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Logout
Choose option:  2
Enter Product ID:  102
Enter Quantity:  4
Added product 102 with quantity 4 to cart


1. View Catalog
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Logout
Choose option:  3
Product: Boots, Quantity: 2
Product: Jackets, Quantity: 4
```

```
1. View Catalog
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Logout
Choose option:  4
Enter Product ID to remove:  101
Removed product 101 from cart

1. View Catalog
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Logout
Choose option:  3
Product: Jackets, Quantity: 4

1. View Catalog
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Logout
Choose option:  5
Enter Payment Method (UPI/NetBanking/PayPal):  UPI
Your total amount is 8000
```

```
You will be redirected to complete your payment using UPI.


1. View Catalog

2. Add to Cart

3. View Cart

4. Remove from Cart

5. Checkout

6. Logout

Choose option:  6

Logged out successfully


1. User Login

2. Admin Login

3. Exit

Choose option:
```