

# Online Shopping App

```

"""
A simple shopping application to demonstrate the use of classes and objects in Python.
"""

class Cart:
    """cart class to handle shopping cart operations.
    """

    def __init__(self) -> None:
        """Initialize the cart.
        """
        self.cart = []

    def display_cart(self, session_id):
        """Display the items in the cart for a specific session.

        Args:
            session_id (str): session ID of the user
        """
        if not self.cart:
            print("Your cart is empty.")
        else:
            print("\nYour Cart:")
            for item in self.cart:
                if item["session"] == session_id:
                    print(f"ID: {item['id']}, Name: {item['name']}, Quantity: {
                        item['quantity']}, Price: ${item['price']}")

    def add_to_cart(self, session_id, product_id, quantity):
        """Add a product to the cart for a specific session.

        Args:
            session_id (str): The session ID of the user.
            product_id (int): The ID of the product to add.
            quantity (int): The quantity of the product to add.
        """
        for product in catalog:
            if product["id"] == product_id:
                self.cart.append({"session": session_id,
                                   "id": product_id,
                                   "name": product["name"],
                                   "quantity": quantity,
                                   "price": product["price"] * quantity})
                print(f"Added {quantity} of {product['name']} to your cart.")

```

```

        return

    print("Product not found.")

def delete_from_cart(self, session_id, product_id):
    """Remove a product from the cart for a specific session.

    Args:
        session_id (str): The session ID of the user.
        product_id (int): The ID of the product to remove.
    """
    for item in self.cart.copy():
        if item["session"] == session_id and item["id"] == product_id:
            self.cart.remove(item)
            print(f"Removed {item['name']} from your cart.")
    return

    print("Product not found in cart.")

catalog = [
    {"id": 1, "name": "Boots", "category": "Footwear", "price": 50},
    {"id": 2, "name": "Coat", "category": "Clothing", "price": 80},
    {"id": 3, "name": "Jacket", "category": "Clothing", "price": 60},
    {"id": 4, "name": "Cap", "category": "Accessories", "price": 20}
]

cart = Cart()

class Authenticator:
    """Authenticator class to handle user and admin login.

    """

    def __init__(self) -> None:
        """Initialize the user and admin databases.

        """

        self.user_db = {"user1": "password1", "user2": "password2"}
        self.admin_db = {"admin": "adminpass"}

    def user_login(self):
        """user_login method to handle user login.

        """

```

```

        username = input("Enter your username: ")
        password = input("Enter your password: ")

        if username in self.user_db and self.user_db[username] == password:
            print(f"Login successful as user: {username}")
            user_session(username)
        else:
            print("Login failed")

def admin_login(self):
    """admin_login method to handle admin login.
    """

    username = input("Enter admin username: ")
    password = input("Enter admin password: ")

    if username in self.admin_db and self.admin_db[username] == password:
        print(f"Login successful as admin: {username}")
        admin_session(username)
    else:
        print("Login failed")

def generate_session_id(username):
    """Hash the user name to generate a session ID.

    Args:
        username (String): username of the user

    Returns:
        int: session ID
    """
    return (hash(username) * -1) // 100000000

def display_catalog():
    """Display the product catalog.
    """
    print("\nProduct Catalog:")
    for product in catalog:
        print(f"ID: {product['id']}, Name: {product['name']}, Category: {
            product['category']}, Price: ${product['price']}")

```

```

def checkout():
    """Checkout the items in the cart.
    """
    print("\nSelect a payment method:")
    print("1. Net Banking")
    print("2. PayPal")
    choice = input("Enter your choice: ")

    if choice == "1":
        print("You will be redirected to the Net Banking portal.")
    elif choice == "2":
        print("You will be redirected to the PayPal portal.")
    else:
        print("Invalid choice.")
        return

    print("Your order is successfully placed.")


def user_session(username):
    """Start a user session.

    Args:
        username (String): username of the user
    """
    session_id = generate_session_id(username)
    print(f"User session ID: {session_id}")
    while True:
        print("\nOptions:")
        print("1. View catalog")
        print("2. View cart")
        print("3. Add to cart")
        print("4. Remove from cart")
        print("5. Checkout")
        print("6. Logout")

        choice = input("Enter your choice: ")

        if choice == "1":
            display_catalog()
        elif choice == "2":
            cart.display_cart(session_id)
        elif choice == "3":
            product_id = int(input("Enter the product ID to add to cart: "))

```

```

        quantity = int(input("Enter the quantity: "))
        cart.add_to_cart(session_id, product_id, quantity)
    elif choice == "4":
        product_id = int(
            input("Enter the product ID to remove from cart: "))
        cart.delete_from_cart(session_id, product_id)
    elif choice == "5":
        checkout()
    elif choice == "6":
        print("Logged out.")
        break
    else:
        print("Invalid choice.")

def admin_session(username):
    """Start an admin session.

    Args:
        username (String): username of the admin
    """
    session_id = generate_session_id(username)
    print(f"Admin session ID: {session_id}")
    while True:
        print("\nAdmin Options:")
        print("1. View catalog")
        print("2. Add product")
        print("3. Update product")
        print("4. Remove product")
        print("5. Logout")
        choice = input("Enter your choice: ")

        if choice == "1":
            display_catalog()
        elif choice == "2":
            add_product()
        elif choice == "3":
            update_product()
        elif choice == "4":
            remove_product()
        elif choice == "5":
            print("Logged out.")
            break
    else:

```

```
        print("Invalid choice.")

def add_product():
    """Add a new product to the catalog.
    """
    product_id = int(input("Enter product ID: "))
    name = input("Enter product name: ")
    category = input("Enter product category: ")
    price = float(input("Enter product price: "))
    catalog.append({"id": product_id, "name": name,
                    "category": category, "price": price})
    print(f"Product {name} added successfully.")

def update_product():
    """Update an existing product in the catalog
    """
    product_id = int(input("Enter product ID to update: "))
    for product in catalog:
        if product["id"] == product_id:
            product["name"] = input("Enter new name: ")
            product["category"] = input("Enter new category: ")
            product["price"] = float(input("Enter new price: "))
            print(f"Product {product_id} updated successfully.")
            return
    print("Product not found.")

def remove_product():
    """Remove a product from the catalog.
    """
    product_id = int(input("Enter product ID to remove: "))
    for product in catalog.copy():
        if product["id"] == product_id:
            catalog.remove(product)
            print(f"Product {product_id} removed successfully.")
            return
    print("Product not found.")

def main():
    """Main function to run the shopping application.
    """
```

```
print("Welcome to the Demo Marketplace")
authenticator = Authenticator()
while True:
    print("\nLogin as:")
    print("1. User")
    print("2. Admin")
    print("3. Exit")
    user_type = input("Enter your choice: ")

    if user_type == "1":
        authenticator.user_login()
    elif user_type == "2":
        authenticator.admin_login()
    elif user_type == "3":
        print("Exiting the application.")
        break
    else:
        print("Invalid choice.")

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