A2Z Shopping App

```
class ECommerceApp: # Creating class
    def __init__(self): #Initialization
        # Sample data
        # Categories Dictionary/DB
        self.categories_db = {
            1: "Footwear",
            2: "Clothing",
            3: "Electronics",
            4: "Kitchenware"
        }

        # Catalog Dictionary/DB
        self.catalog_db = {
            1001: {"name": "Shoes", "category_id": 1, "price": 1500},
            1002: {"name": "Sleeper", "category_id": 1, "price": 500},
            1003: {"name": "Crocs", "category_id": 1, "price": 800},
            1004: {"name": "Sandal", "category_id": 1, "price": 1000},
            1005: {"name": "Loafer", "category_id": 1, "price": 3000},
            1006: {"name": "Shirt", "category_id": 2, "price": 1500},
            1007: {"name": "TShirt", "category_id": 2, "price": 1000},
            1008: {"name": "Pant", "category_id": 2, "price": 1700},
            1009: {"name": "Saree", "category_id": 2, "price": 4000},
            1010: {"name": "TV", "category_id": 3, "price": 40000},
            1011: {"name": "Fridge", "category_id": 3, "price":
35000},
            1012: {"name": "Fan", "category_id": 3, "price": 5000},
            1013: {"name": "Tubelight", "category_id": 3, "price":
300},
            1014: {"name": "AC", "category_id": 3, "price": 60000},
            1015: {"name": "Camera", "category_id": 3, "price":
70000},
            1016: {"name": "Pan", "category_id": 4, "price": 1500},
            1017: {"name": "Blender", "category_id": 4, "price":
6000},
            1018: {"name": "Oven", "category_id": 4, "price": 10000},
            1019: {"name": "Stove", "category_id": 4, "price": 7000},
            1020: {"name": "Cooker", "category_id": 4, "price": 5000}
        }

        # Users Dictionary/DB
        self.users_db = {
            "user1@a2z.com": "password1",
            "user2@a2z.com": "password2",
            "user3@a2z.com": "password3",
            "user4@a2z.com": "password4"
        }
```

```
        #Admins Dictionary/DB
        self.admins_db = {
            "admin1@a2z.com": "adminpassword1",
            "admin2@a2z.com": "adminpassword2"
        }

        #Creating empty session and cart
        self.sessions_db = {}
        self.carts_db = {}

    # Display Welcome Message
    def display_welcome_message(self):
        print("Welcome to the A2Z Marketplace! :)")

    # Validate Admin or User login, create the session and assign role
user/admin
    def login(self, email, password):
        if email in self.users_db and self.users_db[email] ==
password:
            session_id = f"user_{email}"
            self.sessions_db[session_id] = {"role": "user"}
            return session_id
        elif email in self.admins_db and self.admins_db[email] ==
password:
            session_id = f"admin_{email}"
            self.sessions_db[session_id] = {"role": "admin"}
            return session_id
        return None

    # Get all Categories
    def get_categories(self):
        return self.categories_db

    # Get all Products in the Catalog
    def get_products_by_category(self, category_id):
        return {prod_id: prod_details for prod_id, prod_details in
self.catalog_db.items() if prod_details["category_id"] == category_id}

    # Add Product and quantity to the cart
    def add_to_cart(self, session_id, product_id, quantity):
        if session_id not in self.carts_db:
            self.carts_db[session_id] = {}
        if product_id in self.carts_db[session_id]:
            self.carts_db[session_id][product_id] += quantity
        else:
            self.carts_db[session_id][product_id] = quantity

    # View Product and quantity to the cart
    def view_cart(self, session_id):
```

```
        cart = self.carts_db.get(session_id, {})
        cart_items = []
        total_items = 0
        total_price = 0
        for prod_id, quantity in cart.items():
            if prod_id in self.catalog_db:
                prod_details = self.catalog_db[prod_id]
                total_items += quantity
                total_price += prod_details["price"] * quantity
                cart_items.append({
                    "product_id": prod_id,
                    "name": prod_details["name"],
                    "quantity": quantity,
                    "price": prod_details["price"],
                    "total_price": prod_details["price"] * quantity
                })
        return cart_items, total_items, total_price

    # Delete Product and quantity to the cart
    def delete_from_cart(self, session_id, product_id):
        if session_id in self.carts_db and product_id in
self.carts_db[session_id]:
            del self.carts_db[session_id][product_id]

    # Clear the cart after order placed
    def clear_cart(self, session_id):
        if session_id in self.carts_db:
            del self.carts_db[session_id]

    # Checkout and place the order
    def checkout(self, session_id, payment_method):
        self.clear_cart(session_id)
        return "Your order is successfully placed."

    # View Categories
    def view_categories(self):
        return self.categories_db

    # View all Products in the Catalog with a specific category
    def view_catalog(self):
        catalog_list = []
        for prod_id, prod_details in self.catalog_db.items():
            cat_name =
self.categories_db.get(prod_details["category_id"], "Unknown")
            catalog_list.append({
                "ID": prod_id,
                "Name": prod_details["name"],
                "Category": cat_name,
                "Price": prod_details["price"]
            })
```

```
        return catalog_list

    # Add Category
    def add_category(self, session_id, category_id, category_name):
        if session_id not in self.sessions_db or
self.sessions_db[session_id]["role"] != "admin":
            return "Access denied."
        if category_id in self.categories_db:
            return "Category ID already exists."
        self.categories_db[category_id] = category_name
        return None

    # Delete Category
    def delete_category(self, session_id, category_id):
        if session_id not in self.sessions_db or
self.sessions_db[session_id]["role"] != "admin":
            return "Access denied."
        if category_id not in self.categories_db:
            return "Category ID not found."
        del self.categories_db[category_id]
        # Remove all products associated with this category
        self.catalog_db = {prod_id: details for prod_id, details in
self.catalog_db.items() if details["category_id"] != category_id}
        return None

    # Add Product
    def add_product(self, session_id, product_name, category_id,
price):
        if session_id not in self.sessions_db or
self.sessions_db[session_id]["role"] != "admin":
            return "Access denied."
        next_id = self.get_next_product_id()
        self.catalog_db[next_id] = {"name": product_name,
"category_id": category_id, "price": price}
        return next_id

    # Delete Product
    def delete_product(self, session_id, product_id):
        if session_id not in self.sessions_db or
self.sessions_db[session_id]["role"] != "admin":
            return "Access denied."
        if product_id not in self.catalog_db:
            return "Product ID not found."
        del self.catalog_db[product_id]
        return None

    # Generate the Product ID
    def get_next_product_id(self):
        if not self.catalog_db:
            return 1001
```

```
        return max(self.catalog_db.keys()) + 1

def main(): # Defining the main function to call the programme
    app = ECommerceApp()
    app.display_welcome_message()

    # User login Inputs
    email = input("Enter your email: ")
    password = input("Enter your password: ")
    session_id = app.login(email, password)

    if not session_id:
        print("Invalid credentials.")
        return

    # Assigning role user/admin
    role = app.sessions_db[session_id]["role"]

    # User interactive session flow
    if role == "user":
        while True:
            print("\nCategories:") # Print all categories to choose
from for shopping
            categories = app.get_categories()
            for cat_id, cat_name in categories.items():
                print(f"{cat_id}: {cat_name}")
            category_id = int(input("Select category ID to view
products or 0 to exit: "))

            if category_id == 0:
                break

            products = app.get_products_by_category(category_id)
            if not products:
                print("No products found in this category.")
                continue

            print("\nProducts:")  # Print all the Products from the
selected categories and request
            for prod_id, prod_details in products.items():
                print(f"ID: {prod_id}, Name: {prod_details['name']},
Price: {prod_details['price']}")

            product_id = int(input("Enter product ID to add to cart or
0 to shop more: "))
            if product_id == 0:
                continue

            quantity = int(input("Enter quantity: ")) # Ask for the
quantity
```

```
            app.add_to_cart(session_id, product_id, quantity)
            print("Product added to cart.")

            # Ask if more shopping is to be done or go to the cart
            while True:
                action = input("Enter 'shop' to continue shopping,
'cart' to view cart, or 'delete' to remove an item from cart:
").strip().lower()
                if action == 'cart':
                    cart_items, total_items, total_price =
app.view_cart(session_id)
                    print("\nYour Cart:")
                    for item in cart_items:
                        print(f"Product: {item['name']}, Quantity:
{item['quantity']}, Price: {item['price']}, Total:
{item['total_price']}")
                    print(f"\nTotal Items: {total_items}, Total
Payable Amount: {total_price}")

                    if cart_items:
                        print("\nPayment Methods:")
                        print("1. UPI")
                        print("2. Debit Card")
                        print("3. Credit Card")
                        print("4. Net Banking")
                        payment_option = int(input("Select payment
method (1-4): "))
                        payment_methods = {1: "UPI", 2: "Debit Card",
3: "Credit Card", 4: "Net Banking"}
                        payment_method =
payment_methods.get(payment_option)
                        if not payment_method:
                            print("Invalid payment method selected.")
                        else:
                            print(app.checkout(session_id,
payment_method))
                        break
                    else:
                        break

                elif action == 'shop':
                    break

                elif action == 'delete':
                    cart_items, _, _ = app.view_cart(session_id)
                    if not cart_items:
                        print("Your cart is empty.")
                        break
                    print("\nYour Cart:")
                    for item in cart_items:
```

```
                        print(f"Product ID: {item['product_id']},
Name: {item['name']}, Quantity: {item['quantity']}, Price:
{item['price']}, Total: {item['total_price']}")
                    product_id = int(input("Enter product ID to delete
from cart: "))
                    app.delete_from_cart(session_id, product_id)
                    print("Product removed from cart.")
                else:
                    print("Invalid action. Please try again.")

    # Session for admin
    elif role == "admin":
        # Print all the actions for the Admins and choose as input
        while True:
            print("\nAdmin Actions:")
            print("1. View Categories")
            print("2. View Catalog")
            print("3. Add Category")
            print("4. Delete Category")
            print("5. Add Product")
            print("6. Delete Product")
            print("7. Exit")

            choice = int(input("Enter your choice: "))
            if choice == 1:   # Print all Categories if choice is 1
                categories = app.view_categories()
                print("\nCategories:")
                for cat_id, cat_name in categories.items():
                    print(f"ID: {cat_id}, Name: {cat_name}")
            elif choice == 2:  # Print all catalogs if choice is 2
                catalog = app.view_catalog()
                print("\nCatalog:")
                for item in catalog:
                    print(f"ID: {item['ID']}, Name: {item['Name']},
Category: {item['Category']}, Price: {item['Price']}")
            elif choice == 3:  # Allow admin to add Category if the
choice is 3
                while True:
                    category_id = int(input("Enter new category ID:
"))
                    category_name = input("Enter new category name: ")
                    result = app.add_category(session_id, category_id,
category_name)
                    if result:
                        print(result)
                    else:
                        print("Category added successfully.")
                    if input("Enter 'back' to return to previous
actions or 'exit' to go back to the main menu: ").strip().lower() ==
```

```
'exit':
                        break
            elif choice == 4:  # Allow admin to delete Category if the
choice is 4
                while True:
                    category_id = int(input("Enter category ID to
delete: "))
                    result = app.delete_category(session_id,
category_id)
                    if result:
                        print(result)
                    else:
                        print("Category deleted successfully.")
                    if input("Enter 'back' to return to previous
actions or 'exit' to go back to the main menu: ").strip().lower() ==
'exit':
                        break
            elif choice == 5:  # Allow admin to add Product if the
choice is 5
                while True:
                    product_name = input("Enter product name: ")
                    category_id = int(input("Enter category ID for the
product: "))
                    price = float(input("Enter product price: "))
                    product_id = app.add_product(session_id,
product_name, category_id, price)
                    if isinstance(product_id, int):
                        print(f"Product added successfully with ID:
{product_id}")
                    else:
                        print(product_id)
                    if input("Enter 'back' to return to previous
actions or 'exit' to go back to the main menu: ").strip().lower() ==
'exit':
                        break
            elif choice == 6:  # Allow admin to delete Product if the
choice is 6
                while True:
                    product_id = int(input("Enter product ID to
delete: "))
                    result = app.delete_product(session_id,
product_id)
                    if result:
                        print(result)
                    else:
                        print("Product deleted successfully.")
                    if input("Enter 'back' to return to previous
actions or 'exit' to go back to the main menu: ").strip().lower() ==
'exit':
```

```
                        break
            elif choice == 7:  # Exit from Menu if choice is 7
                break
            else:
                print("Invalid choice.")

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

```
Welcome to the A2Z Marketplace! :)
```