Demo Marketplace Python Project Documentation

The Demo Marketplace is a Python project that simulates a simple marketplace where users can browse products, add them to their cart, and place orders. The project provides functionality for both regular users and administrators. Users can create accounts, log in, view the product catalog, add products to their cart, remove products from the cart, view their cart, and place orders. Administrators have additional privileges such as add/edit product or category

Functionality

The Demo Marketplace project provides the following functionality:

1. Display a Welcome Message

The `display_welcome_message()` function displays a welcome message when the program starts.

2. Create Admin

The `create_admin(username)` function allows you to create an administrator account. Administrators have additional privileges compared to regular users.

3. Create User

The `create_user(username)` function allows you to create a regular user account.

4. Create Sample Products

The `create_sample_products()` function creates some sample products and adds them to the product catalog. Each product has a unique product ID, name, category, and price.

5. Login

The `login(username)` function allows users and administrators to log in to their accounts. After logging in, the current user is stored in the `current_user` variable, which is used for performing user-specific actions

6. Logout

The `logout()` function logs out the current user by setting the `current_user` variable to `None`.

7. View Catalog

The `view_catalog()` function displays the product catalog. It lists the product ID, name, category, and price of each product.

8. Add to Cart

The `add_to_cart(product_id, quantity)` function allows the current user to add a product to their cart. The function checks if the user is logged in and if the product ID is valid before adding the product to the cart.

9. Remove from Cart

The `remove_from_cart(product_id)` function allows the current user to remove a product from their cart. The function checks if the user is logged in and removes the product from the cart if it exists.

10. View Cart

The `view_cart()` function displays the contents of the current user's cart. It lists the products in the cart along with their quantities and calculates the total price of the cart.

11. Checkout

The `checkout(payment_method)` function allows the current user to place an order and complete the checkout process. The function checks if the user is logged in, if the cart is empty, and if the payment method is valid. If the payment method is valid, it simulates a payment process by generating a random transaction ID.

☐ **User Authentication:**

Test **user** login with an invalid username. Test **admin** login with invalid username

Data Validation:

Validate that the application properly handles invalid **data** inputs and **provides appropriate error messages**.

Product Catalog:

Verify that the **product** catalog is displayed correctly. Check if users can see product details such as name, category, and price. Verify that the admin can see the **product catalog**.

User Cart:

Add items to the user's **cart**. Remove items from the user's cart. **Verify** that the cart contents are displayed correctly. Ensure that users cannot access admin-specific cart functions.

Admin Product Management:

Add a new **category** to the catalog as an admin. Ensure that users cannot access admin-specific product management functions.

Session Management:

Test user session management **(login, logout)**. Test admin session management (login, logout). Check if users and admins are redirected to the login page when not authenticated. Verify that error messages are displayed when appropriate.

```
import random

products = []
users = []
admins = []
categories = []
current_user = None
category_name = []
product_id = []


def display_welcome_message():
    print("Welcome to the Demo Marketplace")

def create_admin(username):
    admins.append(username)

def create_user(username):
    users.append({"username": username, "cart": []})

def create_sample_categories():
    categories.append({"category_name": "Footwear"})
    categories.append({"category_name": "Clothing"})
    categories.append({"category_name": "Clothing"})
    categories.append({"category_name": "Electronics"})


def create_sample_products():
    products.append({"product_id": 1, "name": "HIGH HEEL", "category": "Footwear", "price": 59.0})
    products.append({"product_id": 2, "name": "COATS", "category": "Clothing", "price": 99.0})
    products.append({"product_id": 3, "name": "SHIRTS", "category": "Clothing", "price": 89.0})
    products.append({"product_id": 4, "name": "LAPTOPS", "category": "Electronics", "price": 999.0})

def login(username):
    global current_user
    if username in [user["username"] for user in users]:
        current_user = next(user for user in users if user["username"] == username)
        return f"User {username} logged in."
    elif username in admins:
        current_user = username
        return f"Admin {username} logged in."
    else:
        return "Invalid username."

def logout():
    global current_user
    current_user = None

def view_catalog():
    for product in products:
        print(f"Product ID: {product['product_id']}, Name: {product['name']}, Category: {product['category']}, Price: {product['price']}")

def add_to_cart(product_id, quantity):
    global current_user
    if not current_user:
        return "Please log in first."

    product = next((p for p in products if p["product_id"] == product_id), None)
    if product:
        current_user["cart"].append({"product": product, "quantity": quantity})
        return f"Added {quantity} {product['name']} to the cart."
    else:
        return "Invalid product ID."

def remove_from_cart(product_id):
    global current_user
    if not current_user:
        return "Please log in first."

    current_user["cart"] = [item for item in current_user["cart"] if item["product"]["product_id"] != product_id]
    return "Item removed from the cart."

def view_cart():
    global current_user
    if not current_user:
        return "Please log in first."

    cart = current_user["cart"]
    if not cart:
        return "Your cart is empty."

    total_price = sum(item["product"]["price"] * item["quantity"] for item in cart)
    print("Your cart:")
    for item in cart:
        print(f"Product: {item['product']['name']}, Quantity: {item['quantity']}")
    print(f"Total Price: Rs. {total_price}")

def checkout(payment_method):
    global current_user
    if not current_user:
        return "Please log in first."

    cart = current_user["cart"]
    if not cart:
        return "Your cart is empty."

    total_price = sum(item["product"]["price"] * item["quantity"] for item in cart)

    if payment_method == "UPI" or payment_method == "Debit Card":
        transaction_id = random.randint(1000, 9999)
        return f"Your order is successfully placed. Total amount: Rs. {total_price}. Transaction ID: {transaction_id}"
    else:
        return "Invalid payment method."


def admin_add_or_edit_category():
    global current_user
    if current_user not in admins:
        return "Only admins can add/edit categories."

    category_name = input("Enter category name (ADD or EDIT to add/edit a new category): ")
    if category_name == "ADD":
        category = {}
        category["category_name"] = len(categories) + 1
        category["name"] = input("Enter new category name: ")
        categories.append(category)
        return f"Category '{category['name']}' added."
    elif category_name == "EDIT":
        category_name = input("Enter category name to edit: ")
        category = next((c for c in categories if c["category_name"] == category_name), None)
        if category:
            new_name = input("Enter new category name: ")
            category["name"] = new_name
            return f"Category name {category_name} updated to '{new_name}'."
        else:
            return f"Category name '{category_name}' not found."
    else:
        return "Invalid choice"



def admin_add_or_edit_product():
    global current_user
    if current_user not in admins:
        return "Only admins can add/edit products."

    product_id = input("Enter product name (ADD or EDIT to add/edit a new product): ")
    if product_id == "ADD":
        product = {}
        product["product_id"] = len(products) + 1
        new_product_id = int(input("Enter new product ID: "))
        new_product_name = str(input("Enter new product name: "))
        new_product_category = str(input("Enter new product category: "))
        new_product_price = float(input("Enter new product price: "))
        product = {
        "product_id": new_product_id,
        "name": new_product_name,
        "category": new_product_category,
        "price": new_product_price}
        products.append(product)
        return f"Product '{new_product_name}' added."
    elif product_id == "EDIT":
        product_id = int(input("Enter product ID to edit: "))
        product = next((p for p in products if p["product_id"] == product_id), None)
        if product:
            new_product_id = int(input("Enter new product ID: "))
            product["ID"] = new_product_id
            return f"Product ID {product_id} updated to '{new_product_id}'."
        else:
            return f"Product ID '{product_id}' not found."
    else:
        return "Invalid choice"


if __name__ == "__main__":
    display_welcome_message()
    create_admin("admin")
    create_user("user1")
    create_user("user2")
    create_sample_products()
    create_sample_categories()

    while True:
        print("\nMain Menu:")
        print("1. Login")
        print("2. Logout")
        print("3. View Catalog")
        print("4. Add to Cart")
        print("5. Remove from Cart")
        print("6. View Cart")
        print("7. Checkout")
        print("8. Admin: Add/Edit Category")
        print("9. Admin: Add/Edit Product")
        print("10. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            username = input("Enter username: ")
            print(login(username))
        elif choice == "2":
            logout()
            print("Logged out successfully.")
        elif choice == "3":
            view_catalog()
        elif choice == "4":
            product_id = int(input("Enter product ID to add to cart: "))
            quantity = int(input("Enter quantity: "))
            print(add_to_cart(product_id, quantity))
        elif choice == "5":
            product_id = int(input("Enter product ID to remove from cart: "))
            print(remove_from_cart(product_id))
        elif choice == "6":
            view_cart()
        elif choice == "7":
            payment_method = input("Enter payment method (UPI/Debit Card): ")
            print(checkout(payment_method))
        elif choice == "8":
            print(admin_add_or_edit_category())
        elif choice == "9":
            print(admin_add_or_edit_product())
        elif choice == "10":
            print("Exiting the application.")
            break
        else:
            print("Invalid choice. Please try again.")
```