Source Code:

```
# --- User Class ---
class User:
    def __init__(self):
        self.users = {"Admin": "Admin@123"}
        self.logged_in_user = None

    def login(self):
        print("Login Page")
        username = input("Enter username: ")
        password = input("Enter password: ")

        if username in self.users and self.users[username] == password:
            self.logged_in_user = username
            print(f"Welcome {username}!")
        else:
            print("Invalid credentials. Logged in as public user.")
            self.logged_in_user = "public"

    def is_admin(self):
        return self.logged_in_user.lower() == "admin"

# --- Category Manager ---
class CategoryManager:
    def __init__(self):
        self.categories = {
            "Footwear": [],
            "Clothing": [],
            "Electronics": [],
```

```
            "Accessories": []
        }

    def display_categories(self):
        print("\n📂 Available Categories:")
        for category in self.categories:
            print(f"- {category}")

    def add_category(self):
        new_cat = input("Enter new category name: ").strip().capitalize()
        if new_cat in self.categories:
            print("Category already exists.")
        else:
            self.categories[new_cat] = []
            print(f"Category '{new_cat}' added.")

    def update_category(self):
        old_cat = input("Enter category to rename: ").strip().capitalize()
        if old_cat in self.categories:
            new_cat = input("Enter new category name: ").strip().capitalize()
            self.categories[new_cat] = self.categories.pop(old_cat)
            print(f"Category '{old_cat}' renamed to '{new_cat}'.")
        else:
            print("Category not found.")

# --- Product Manager ---
class ProductManager:
    def __init__(self, categories):
        self.categories = categories
```

```
    def add_product(self):
        category = input("Enter category to add product in: ").strip().capitalize()
        if category not in self.categories:
            print("Category does not exist.")
            return

        name = input("Enter product name: ").strip().capitalize()
        price = float(input("Enter product price: "))
        stock = int(input("Enter stock quantity: "))

        product = {"name": name, "price": price, "stock": stock}
        self.categories[category].append(product)
        print(f"Product '{name}' added to '{category}'.")

    def display_products(self):
        for category, products in self.categories.items():
            print(f"\n📁 {category}:")
            if not products:
                print("  No products.")
            for product in products:
                print(f"  - {product['name']} | ₹{product['price']} | Stock: {product['stock']}")

# --- Cart Class ---
class Cart:
    def __init__(self):
        self.items = []

    def add_to_cart(self, categories):
```

```
    category = input("Enter category: ").strip().capitalize()
    if category not in categories or not categories[category]:
        print("Invalid category or no products.")
        return

    for i, product in enumerate(categories[category]):
        print(f"{i+1}. {product['name']} - ₹{product['price']} - Stock: {product['stock']}")

    choice = int(input("Select product number: ")) - 1
    if 0 <= choice < len(categories[category]):
        product = categories[category][choice]
        quantity = int(input("Enter quantity: "))
        if quantity <= product['stock']:
            self.items.append({
                "name": product['name'],
                "price": product['price'],
                "quantity": quantity
            })
            product['stock'] -= quantity
            print(f"✅ {quantity} x {product['name']} added to cart.")
        else:
            print("❌ Insufficient stock.")
    else:
        print("Invalid selection.")

def remove_from_cart(self):
    if not self.items:
        print("Cart is empty.")
```

```
        return

    for i, item in enumerate(self.items):
        print(f"{i+1}. {item['name']} - Qty: {item['quantity']}")

    choice = int(input("Enter item number to remove: ")) - 1
    if 0 <= choice < len(self.items):
        removed = self.items.pop(choice)
        print(f" ✅ Removed {removed['name']} from cart.")
    else:
        print("Invalid selection.")

def view_cart(self):
    if not self.items:
        print("Cart is empty.")
        return 0

    print("\n🛒 Your Cart:")
    total = 0
    for item in self.items:
        subtotal = item['price'] * item['quantity']
        total += subtotal
        print(f"- {item['name']} x {item['quantity']} = ₹{subtotal}")
    print(f" 🧾 Total: ₹{total}")
    return total

def checkout(self):
    total = self.view_cart()
```

```
        if total == 0:
            return

        print("\n💳 Select Payment Method:")
        print("1. UPI")
        print("2. Debit Card")

        method = input("Enter choice: ")
        if method == "1":
            upi_id = input("Enter UPI ID: ")
            print(f"✅ Payment of ₹{total} successful via UPI ({upi_id})")
        elif method == "2":
            card = input("Enter Debit Card Number: ")
            print(f"✅ Payment of ₹{total} successful via Debit Card")
        else:
            print("❌ Invalid payment method.")

# --- Main Application Flow ---
if __name__ == "__main__":
    user = User()
    user.login()

    category_manager = CategoryManager()
    product_manager = ProductManager(category_manager.categories)

    if user.is_admin():
        # Admin dashboard
        while True:
```

```
        print("\nAdmin Options:")
        print("1. Add Category")
        print("2. Update Category")
        print("3. View Categories")
        print("4. Add Product")
        print("5. View All Products")
        print("6. Exit")

        choice = input("What you want today: ")

        if choice == "1":
            category_manager.add_category()
        elif choice == "2":
            category_manager.update_category()
        elif choice == "3":
            category_manager.display_categories()
        elif choice == "4":
            product_manager.add_product()
        elif choice == "5":
            product_manager.display_products()
        elif choice == "6":
            print("Logging out...")
            break
        else:
            print("Invalid choice.")

else:
    # Public (customer) dashboard
    cart = Cart()
```

```
while True:
    print("\nPublic Options:")
    print("1. View Categories")
    print("2. View Products")
    print("3. Add to Cart")
    print("4. Remove from Cart")
    print("5. View Cart")
    print("6. Checkout")
    print("7. Exit")

    choice = input("What you are looking for: ")

    if choice == "1":
        category_manager.display_categories()
    elif choice == "2":
        product_manager.display_products()
    elif choice == "3":
        cart.add_to_cart(category_manager.categories)
    elif choice == "4":
        cart.remove_from_cart()
    elif choice == "5":
        cart.view_cart()
    elif choice == "6":
        cart.checkout()
    elif choice == "7":
        print("Thanks for visiting!")
        break
    else:
```

```
print("Invalid choice.")
```