Code Documentation

Overview:

This is a shopping store management system implemented in Python using object-oriented programming (OOP) principles. The system has four main components:

1. Product management (Base and Derived classes)
2. User management (Customers)
3. Admin functionality (Owner)
4. Order and payment processing

Key Components are explained below:

1. Product Classes (Products\_Classes.py)

The product hierarchy forms the core of the system:

**Base Class (Product)**

* Contains common attributes for all products (ID, name, description, price, quantity, category, gender).
* Provides a string representation of product details.

**Derived Classes** (Clothing, Electronics, Footwear, Beauty)

* Each inherits from Product and adds specialized attributes:
  + Clothing: Size and material
  + Electronics: Brand and warranty period
  + Footwear: Size and shoe type
  + Beauty: Brand and skin type
* Each overrides the string representation to include its specific attributes in product details.

2. Product Management (Products\_Functions.py)

Contains functions to display products in different categories:

* display\_products() - Shows all products
* display\_clothing\_products() - Shows only clothing items
* Similar functions exist for electronics, footwear, and beauty products
* clear\_screen() - Utility function to clear the terminal and also print Welcome message

3. User System (User.py)

**User Class**

* Represents store customers with personal details (name, email, address, etc.)
* Each user has a ShoppingCart
* Provides string representation of user details

**User Authentication Functions**

* user\_sl\_menu() - Shows signup/login options
* sign\_up\_user() - Handles new user registration with validation
* user\_log\_in() - Authenticates existing users

4. Shopping Cart (Shopping\_Cart.py)

**ShoppingCart Class**

* Tracks items, quantities, and total amounts for each user
* cart\_items is a list which contains product\_id, quantity, unit\_price for each one item in shopping cart
* Key functions:
  + add\_to\_cart() - Adds products with quantity validation
  + update\_cart() - Modifies item quantities
  + remove\_from\_cart() - Removes items
  + checkout() - Finalizes orders and processes payments

5. Admin System (Admin.py)

**Admin Class**

* Manages store operations with functions to:
  + Add/edit/delete products
  + View and verify payments
* Each operation maintains consistency across all product lists

6. Order Processing (Order\_Payment.py)

**Order Class**

* Tracks order details (ID, user, amount, shipping address, status)
* Handles payment creation

**Payment Class**

* Manages payment details (method, amount, verification status)
* Supports different payment methods (credit card, PayPal, etc.)

7. Main Program Flow (main.py)

* Initializes sample data (products, users, orders)
* Provides the main menu interface to navigate between admin and user functions
* Coordinates all system components

Data Structures

The system uses several global lists to manage data:

* Products\_List - All products
* Category-specific lists (Clothing, Electronics, etc.)
* User\_List - Registered customers
* Orders\_List - All orders
* Payments\_List - Payment records