Practice Assignment 2

Problem 1. (50%) Dependency Injection Pattern

Scenario: you will create a simplified e-commerce system with user authentication, product management, and order processing. Before making any purchase, the user must be authorized first, only then they can place an order (List of products), finally you system calculate final payment. Various services such as OrderService, PaymentService, and UserService will be used to handle requests.

***Entities**: User and Product are simple entities representing a user and a product, respectively.*

1. *User*: ▪ *Username* ▪ *Password*	2. *Product*: ▪ *Id* ▪ *Name* ▪ *Price*

***Service**:*

1. *User Service (UserService): interface defines a method for user authentication, and UserServiceImpl provides a simple implementation. The authentication logic is straightforward—comparing the provided username and password with registered data.*
    - *Interface: UserService*
    - *Implementation: UserServiceImpl*
    - *Method: authenticateUser*
2. *Order Service (OrderService): Processes the order for a given user and products. For simplicity placeOrder method accepted user's name and their order (List of products) then print out user and name of all their products.*
    - *Interface: OrderService*
    - *Implementation: OrderServiceImpl*
    - *Method: placeOrder*
3. *Payment Service (PaymentService):*
    - *Interface: PaymentService*

- *Implementation: PaymentServiceImpl*
- *Method: processPayment.*

Implement Dependency Injection for the application using your own approach.

Problem 2. (50%) Filter Design Pattern

Scenario: Imagine you're managing a large company with hundreds of employees. You need a system to display a ***filtered*** *list of employees based on various criteria, including:*

- *Department: Filter employees belonging to specific departments.*
- *Role: Filter employees holding specific job roles.*
- *Experience: Filter employees with a minimum or maximum experience level.*
- *Salary: Filter employees based on their salary.*
- *Skills: Filter employees possessing specific skill sets.*

Users can use as many filter as possible, in addition, users should be able to ***sort*** *the filtered list by different criteria, such as name, department, or salary.*

Read the scenario carefully then provide your own implementation of Filter Pattern.

~The end~

Your project/ solution should be submitted to Moodle before the deadline.