



Task Title: Product Discount Management

Objective: Create a logic module to manage product discounts for an e-commerce application, where discounts are applied based on specific rules. The task will test the candidate's understanding of business logic, coding skills, and ability to handle different conditions.

Requirements:

1. Problem Statement:

- You are building a discount management system for an e-commerce application.
- The system needs to manage different types of discounts for products.

2. Discount Rules:

- Flat Discount: A fixed amount is subtracted from the product price (e.g., 20 USD off).
- Percentage Discount: A percentage is deducted from the product price (e.g., 10% off).
- Seasonal Discount: If a product is on sale during a specific season, a seasonal discount (e.g., 25% off) is applied.

The system should handle the following:

- The discount can either be a flat amount or a percentage.
- Discounts cannot exceed the original product price (i.e., the discounted price should never be negative).
- The discount should only apply if the product is in stock (product quantity must be greater than zero).
- The discount logic should check if the product is within a valid season (e.g., the current date is within the discount season).

3. API Requirements:

1. POST /product/discount

- Request Body:

```
``json
{
  "productId": "123",
  "discountType": "percentage", // "flat" or "percentage"
  "discountValue": 10,         // Discount value (e.g., flat amount or percentage)
  "seasonalDiscountActive": true,
  "productPrice": 100,         // Product price
  "quantity": 5                // Product quantity in stock
}
```

- This API will calculate and return the final price after applying the discount and return an error if:
 - The product is out of stock.
 - The discount exceeds the product price.
 - Invalid discount type.

2. GET /product/{productId}

- Fetches the product details, including the discounted price if applicable.



4. Business Logic:

1. If the discount type is percentage, the discount value should be applied as a percentage (e.g., 10% off means ``productPrice - (productPrice * 0.10)``).
2. If the discount type is flat, subtract the flat discount value from the price (``productPrice - discountValue``).
3. If the product is out of stock, return a suitable error.
4. If the seasonal discount is active, apply the seasonal discount (e.g., 25%) in addition to the flat or percentage discount.
5. Ensure the final price is never negative. If the calculated discount exceeds the product price, set the price to zero.

Deliverables:

1. Spring Boot Application with:

- ``POST /product/discount`` and ``GET /product/{productId}`` endpoints.
- Use PostgreSQL Database schema to store product data.
- Logic to calculate and apply discounts.

2. README file with:

- How to run the application.
- API test examples using Postman or cURL.

3. PDF file of your output of all the cases.

Time to Complete: 3 hours

Please submit the source code either via GitHub or as a zip file attached to the email.