**Problem Statement : Inventory Management System**

**Duration: 4 hrs**

# Instructions

1. Please make sure that your code does not have any compilation errors while submitting your case study solution.
2. Implement the code using best design standards.
3. Do not use System out statements for logging. Use appropriate logging methods for logging statements/variable/return values.
4. Use Maven to build the project and create WAR file. Include required dependencies in POM.xml
5. Implement the code using REST API.
6. Write web service which takes input and return required details from database.
7. Associate need to upload /share War file.

**Software Requirements**

|  |  |
| --- | --- |
| Tools/APIs | JDK 1.8, SpringtoolsSuite  Spring Boot and Spring MVC  Hibernate 4.x  MySQL 8  REST WebServices  Maven 3.x |
| Server | Tomcat 7/Embedded Tomcat container |

SpringBoot with application to perform CRUD operation with Rest Controller and Write testapplication isRestTemplate

**Problem Statement: Inventory Management System**

Create Product table with following columns:

1. Id
2. Description
3. qty
4. Price
5. Manufacturing Date
6. Use Before Months 3
7. Expiry Date Md+userbeforemonths -java8 data time API

try to use Lambda, Optional,streams.

Inventory contains following Items

Here is the functionality for the initial version of your prototype:

## Functional Specifications:

1. **Item Initialization**

**Post – add number of products**

1. **Calculate Expiry Date**

Every item has a manufacturing date and for many months that item can be consumed. In this functionality, you have to calculate the **Expiry Date** of the **item.**

1. **Remove Expired Items**

In this functionality you have to remove the expired items from the inventory. For this, compare the expiryDate calculated in the previous functionality with the **today’s date**.

You can find today’s date from the system using the java.util.Date API. Make sure your system date gives correct date.

1. **Sort Items**

Sort the item list updated in the previous method in descending order of **expiryDate.**

1. **Apply Discounts**

|  |
| --- |
|  |
|  |

In this functionality you have to reduce the price of the items by applying 20% discounts on theexisting price which will expire in 6 months. For this compare if the **expiryDate** of the item is within 6 months from now.

1. **Search Item**

1.REST API

2.Mysql, JPa Repository

3.Swagger API

4.Validation

5.Exception Handling

6.Actuator.

In case, none of the items has the itemType as the description, the method should throw an exception**“Item NotFound Exception”**.

->Use Chrome with Postman to test all Http methods (or)

->Use SOAPUI to test all service methods