

ElectroPro Ltd is a company dealing with electronic goods such as mobile phones, laptops, desktops and printers etc. They deliver the products ordered by customers manually from time to time. It would be better if an application is available for maintaining the customer and product details.

A Web Application has to be developed according to the given specifications.

1. Design a class CustomerBean having following private instance variables

| | |
|--------------|--------|
| customerID | int |
| customerName | String |
| mobileNumber | String |
| address | String |

The CustomerBean class should be stored in a package com.eshopz.model.
Create the setter and getter methods for all the private instance variables.
Create a parameterized constructor for initializing the variables.

The parameterized constructor should perform the following validations and then initialize the variables accordingly.

- The Customer ID should have only 6 digits.
- The mobile number should be 10 digits.
- Customer Name and Address should not be empty.

2. Create a User Defined Exception named CustomerDetailsInvalidException which is raised when any of the above validation fails.

The CustomerDetailsInvalidException should be stored in a package com.eshopz.exception.

The exception should display relevant error message according to which validation has failed such as

- "Customer ID cannot be less or more than 6 digits",
- "Mobile Number should be 10 digits",
- "Customer Name cannot be empty",
- "Customer Address cannot be empty"

3. Create a ProductBean class containing the following private variables.

| | |
|-------------|--------|
| productID | int |
| productName | String |

The ProductBean class should be stored in a package com.eshopz.model.

Generate the getters and setters for the private variables.

4. Create a MainServlet class and it should be stored in a package com.eshopz.web
The MainServlet method should contain the following

| | |
|--------------|-------------------------------|
| customerObj | Object of class CustomerBean |
| productsList | ArrayList of type ProductBean |

public void addProduct(Product) to add a product to productsList.

public ArrayList<ProductBean> getProducts() to return an ArrayList of Products for a particular customer

5. Create a CustomerBean object and set the Customer Details in the MainServlet.

6. Create 2 Products and add them to the productsList collection using the MainServlet.

7. Display the Customer and Product Details using a jsp page named CustomerOrders.jsp. The CustomerOrders.jsp page should be placed inside the views directory.

The details about suppliers of various products for an organization should be stored in the database with the help of a web application built using servlets for the front end.

8. Create the following table in oracle server.

```
create table SUPPLIERS
(
    SUP_ID integer NOT NULL,
    SUP_NAME varchar(40) NOT NULL,
    STREET varchar(40) NOT NULL,
    CITY varchar(20) NOT NULL,
    STATE char(2) NOT NULL,
    ZIP char(5), PRIMARY KEY (SUP_ID)
)
```

9. Create a WebApplication named SupplierWeb.

10. Create a class named SupplierDAO with the following methods

getConnection() – to obtain connection to the database

storeSupplier() – stores the supplier details to the database.

getSupplierDetail() – accepts the supplier id as a parameter and gets the supplier detail.

11. Create a servlet StoreSupplier which accepts the supplier detail and stores them in the database.
12. The StoreSupplier servlet should use the SupplierDAO to store the details.
13. Create a servlet QuerySupplier which accepts the supplier id and then uses the SupplierDAO to search the database. If the information is found it should display the details, otherwise it should display “Supplier not found!”.

A MVC application has to be created using servlets and jsp which displays information about the various products sold by an e-commerce portal such as Mobiles and Laptops. The project team has decided to use Maven as the build tool. The application has to be created with the following details

14. Create a controller named ProductController.
15. The controller has two methods getMobiles and getLaptops.
16. Create a JSP named as Mobiles.jsp
17. Create a JSP named as Laptops.jsp
18. The Request Mapping for url “/mobiles” should be mapped to the Mobiles.jsp page.
19. Create a MobileBean which stored mobile details. Add the following attributes to the bean in the ProductController.
 - a. Attribute Name : mobile
 - b. Attribute Value : SamsungGalaxyA7
20. The Request Mapping for url “/laptops” should be mapped to Laptops.jsp.
21. Create a LaptopBean which stored laptop details. Add the following attributes to the bean in the ProductController.
 - a. Attribute Name : laptop
 - b. Attribute Value : SonyVaio
22. Design a Products.jsp which contains links to Mobile.jsp and Laptops.jsp

=====