
Shopping System Practice Questions

1. Add Products to Inventory

- Create a program to dynamically add products to an inventory. Each product should be a `Map` with fields like `id`, `name`, `price`, `quantity`, and `category`. Display the updated inventory after adding products.

2. Search Product by Name

- Write a program to search for a product by its name. If the product exists, display its details; otherwise, show *"Product not found."*

3. Filter Products by Category

- Write a program to filter all products by a specific `category`, e.g., "electronics" or "clothing", and display their details.

4. Calculate Total Stock Value

- Write a program to calculate the **total stock value** in the inventory. For each product, multiply its `price` by its `quantity` and sum up all values.

5. Sort Products by Price

- Write a program to sort all products in the inventory by their `price` in ascending or descending order.

6. Update Product Price

- Write a program to update the `price` of a specific product by its `id`.

7. Add to Cart

- Create a cart as a `List<Map>` and write a program to:
 - Add products to the cart by `id` from the inventory.
 - If the product is already in the cart, increase its `quantity`.
-

8. Remove from Cart

- Write a program to remove a product from the cart by its `id`.
-

9. Calculate Cart Total

- Write a program to calculate the **total price** of all items in the cart, considering their `quantity` and `price`.
-

10. Apply Discount

- Write a program to apply a discount to the cart. For example:
 - If the cart total is > 5000 , apply a 10% discount.
 - Otherwise, apply a 5% discount.
-

11. Out-of-Stock Check

- Write a program to check if any product in the inventory has a `quantity` of 0. Display a list of out-of-stock products.
-

12. Generate Bill

- Write a program to generate a bill for the customer. The bill should include:
 - List of all items in the cart (name, quantity, price per unit, total price per item).
 - Cart total (before and after discount).
-

13. Add Product Reviews

- Add a `reviews` key to each product. Write a program to:
 - Add customer reviews for products.
 - Display all reviews for a specific product.
-

14. Find Best-Selling Products

- Write a program to find the top 3 products with the highest `quantity sold`.
-

15. Low Stock Alert

- Write a program to display a **low stock alert** for products whose `quantity` is less than 5.
-

16. Compare Prices Between Categories

- Write a program to compare the average price of products in two categories, e.g., "electronics" and "clothing".
-

17. Group Products by Category

- Write a program to group all products by their `category` and display the grouped lists.
-

18. Restock Inventory

- Write a program to restock a product by its `id`. Add a specific `quantity` to the product's stock.
-

19. Wishlist Feature

- Create a `wishlist` as a separate list. Write a program to:
 - Add products to the wishlist by `id`.
 - Move products from the wishlist to the cart.

20. Top Categories by Sales

- Write a program to find the top 3 categories based on the total sales (price × quantity sold).

Example Dataset for Practice

You can use this example dataset for solving the above questions:

```
List<Map<String, dynamic>> inventory = [
  {'id': 1, 'name': 'Laptop', 'price': 80000, 'quantity': 5, 'category':
'electronics', 'reviews': []},
  {'id': 2, 'name': 'T-Shirt', 'price': 1200, 'quantity': 15, 'category':
'clothing', 'reviews': []},
  {'id': 3, 'name': 'Phone', 'price': 60000, 'quantity': 3, 'category':
'electronics', 'reviews': []},
  {'id': 4, 'name': 'Shoes', 'price': 3000, 'quantity': 7, 'category':
'clothing', 'reviews': []},
  {'id': 5, 'name': 'Headphones', 'price': 4000, 'quantity': 10, 'category':
'electronics', 'reviews': []},
];
```

Cart Example for Practice

```
List<Map<String, dynamic>> cart = [
  {'id': 1, 'name': 'Laptop', 'price': 80000, 'quantity': 1},
  {'id': 4, 'name': 'Shoes', 'price': 3000, 'quantity': 2},
];
```

Additional Challenges

1. **Add Taxes:** Calculate and add taxes (e.g., 8%) to the cart total.
2. **Customer Information:** Store customer details (name, address, phone number) and include it in the bill.
3. **Bulk Discounts:** Apply a discount to a specific product if the customer buys more than a set quantity.
4. **Search with Multiple Filters:** Search products by name, category, or price range.

Agar kisi bhi question ka implementation ya logic mein help chahiye, batao! 😊