Comparison Error Checklist

Error Checklist for E-commerce Website

- **User Authentication**: Failure to properly authenticate users during registration and login processes.
- **Product Availability Check**: Not verifying if the desired product is in stock before allowing purchase.
- **Shipping Errors**: Inaccurate handling of shipping addresses or failure to provide shipping options.
- **Payment Processing**: Issues with processing payments securely and accurately.
- **Order Tracking**: Lack of functionality to track orders and provide status updates to customers.
- **Inventory Management**: Failure to update inventory levels in real-time, leading to out-of-stock items being available for purchase.
- **Customer Support**: Inadequate support system for addressing customer inquiries, complaints, and returns.

Version 1:

Basic E-commerce website system:

Code: Product Availability

```
public class ECommerceSystem {


  public boolean isProductAvailable(String productId) {

    return Math.random() < 0.5

  }


 public static void main(String[] args) {

    ECommerceSystem ecommerceSystem = new ECommerceSystem();

    String productId = "12345";       boolean isAvailable = ecommerceSystem.isProductAvailable(productId);

    if (isAvailable) {

      System.out.println("Product " + productId + " is available.");

    } else {

      System.out.println("Product " + productId + " is out of stock.")
```

```
}}}
```

Test Cases for Version 1: Basic E-commerce website System

Test	Test Case Description	Input Data	Expected	Verdict
1	**Product availability**	100 products. 50 in stock Product id: '12345	Product available	Passed
2	**Shipping errors**	Invalid shipping address provided Address: "123 main st, city"	INVALID ADRESS	Failed
3	**Payment processing**	Purchase total: $100, Payment method: credit card	Payment successful	passed
4	**Order tracking**	Order id: "abcde12345	Shipped	Passed
5	**Inventory management**	50 products, 5 sold Product id: 67890	Out of stocked	Failed

Improved version of the code:

```java
import java.util.HashMap;

import java.util.Map;


public class ECommerceSystem {
  private Map<String, Integer> productsInventory;


  public ECommerceSystem() {
    productsInventory = new HashMap<>();
    productsInventory.put("Product1", 50);
    productsInventory.put("Product2", 20);
    productsInventory.put("Product3", 0);
  }
```

```java
public boolean isProductAvailable(String productId) {

    if (productsInventory.containsKey(productId)) {

        int availableQuantity = productsInventory.get(productId);

        return availableQuantity > 0;

    }

    return false;

}


public static void main(String[] args) {

    ECommerceSystem ecommerceSystem = new ECommerceSystem();


    String productId = "Product1";

    if (ecommerceSystem.isProductAvailable(productId)) {

        System.out.println(productId + " is available.");

    } else {

        System.out.println(productId + " is not available.");

    }  }}
```

Test	Test Case Description	Input Data	Expected	Verdict
1	**Product availability**	100 products. 50 in stock Product id: '12345	Product available	Passed
2	**Shipping errors**	Invalid shipping address provided Address: "123 main st, city"	INVALID ADRESS	Passed
3	**Payment processing**	Purchase total: $100, Payment method: credit card	Payment successful	passed
4	**Order tracking**	Order id: "abcde12345	Shipped	Passed
5	**Inventory management**	50 products, 5 sold Product id: 67890	Product not available	Passed