Exp 9

Admin Module:

a.Add New Product:

i. Cohesion: Exhibits high cohesion as it's directly related to adding new products, aligning with the purpose of the Admin Module.
ii. Coupling: Coupling may involve interactions with the Product Management Module for data entry and validation, but it's justifiable and expected.

b.Manage Employee Accounts:

i. Cohesion: High cohesion as it deals with administrative tasks related to employee accounts and aligns with the Admin Module's purpose.
ii. Coupling: It may interact with the Employee Management Module for account creation and modification, which is reasonable and necessary.

c.Generate Sales Reports:

i. Cohesion: Involves the statistical aspects of sales data, and while it's related to administration, it might be considered a separate concern. Cohesion could be higher in a dedicated Reporting Module.
ii. Coupling: Moderate coupling might exist if it relies on data from the Sales Module and Product Management Module, depending on the extent of data interaction.

d.Manage Supplier Relationships:

i. Cohesion: High cohesion as it relates to managing supplier relationships, which is a core administrative task.
ii. Coupling: It may interact with the Supplier Management Module for supplier data and orders, which is expected and purposeful.

e.Send Notifications:

i. Cohesion: Moderate cohesion, as sending notifications is a general utility that c an be used in different parts of the application.
ii. Coupling: The function is generally standalone and doesn't tightly couple with o ther application modules, making it loosely coupled.

Candidate Module:

a.Browse Products:

i. Cohesion: Exhibits high cohesion as it's directly related to candidates browsing products, which aligns with the purpose of the Candidate Module.
ii. Coupling: It may interact with the Product Display Module for product information, but this coupling is justifiable and expected.

b.Add Products to Cart:

i. Cohesion: High cohesion as adding products to the cart is a core function within the Candidate Module's scope.
ii. Coupling: It interacts with the Shopping Cart Module for cart management and with the Product Management Module for product details, which is expected and not excessively high.

c.Place Orders:

i. Cohesion: High cohesion as it directly relates to the core task of placing orders.
ii. Coupling: It may interact with the Order Processing Module for order submission and the Payment Module for payment processing, but this coupling is purposeful and reasonable.

d.View Order History:

i. Cohesion: High cohesion as viewing order history aligns with a candidate's interactions within the system.
ii. Coupling: Interaction with the Order Management Module for order history is expected and justifiable.

e.Submit Feedback:

i. Cohesion: High cohesion as providing feedback is a candidate-specific task and directly related to their interactions with the system.
ii. Coupling: It may interact with the Feedback Module for feedback submission, which is reasonable and expected.

EXP 10

Risk	**Likelihood**	**Impact**	**Risk Category**	**RMMM**
Product database failure	Medium	High	Technical	**Response:** Establish a disaster recovery plan. **Mitigation:** Implement a data backup and recovery plan. Use a cloud-based database to improve reliability. **Monitoring:** Monitor the database logs for any errors or warnings. Conduct regular database backups.
Customer order loss	Medium	High	Operational	**Response:** Implement a robust order processing system. Use a messaging queue to ensure that orders are not lost. **Mitigation:** Implement a robust order management system. Use a messaging queue to ensure that orders are processed and delivered accurately and on time. **Monitoring:** Monitor the order processing system for any errors or delays. Follow up with customers to ensure that they have received their orders.
Inventory tracking errors	Medium	High	Operational	**Response:** Implement a real-time inventory tracking system. Conduct regular inventory audits. **Mitigation:** Implement a real-time inventory tracking system. Conduct regular inventory audits and reconcile any discrepancies. **Monitoring:** Monitor the inventory tracking system for any errors or inconsistencies. Conduct regular physical inventory audits to verify the accuracy of the system.
Payment processing failure	Medium	High	Financial	**Response:** Use a reputable payment processor. Implement fraud detection measures. **Mitigation:** Use a reputable payment processor. Implement fraud detection measures and monitor for any suspicious transactions. **Monitoring:** Monitor the payment processing system for any errors or delays. Investigate any suspicious transactions.
Customer dissatisfaction	Medium	Medium	Business	**Response:** Provide excellent customer service. Collect feedback from customers and use it to improve the system. **Mitigation:** Provide excellent customer service and collect feedback from customers to identify any areas where the system can be improved. **Monitoring:** Monitor customer satisfaction levels and respond to customer complaints promptly and resolve them to the customer's satisfaction.

Exp 11

Step 1: Create a New Git Repository

```
# Navigate to your project directory
cd path/to/online-grocery-management-system

# Initialize a new Git repository
git init
```

Step 2: Define the Initial Specification
Create an initial specification file (e.g., specification.txt) and add the system specification.

Step 3: Commit the Initial Specification

```
# Stage the specification file
git add specification.txt

# Commit it to the repository
git commit -m "Initial system specification"
```

Step 4: Make Changes and Create Versions

Make changes to the specification file and create different versions.

Create a New Version (e.g., Version 2):

```
# Make changes to the specification as needed
# Save the changes in your text editor

# Stage the modified specification file
git add specification.txt

# Commit the changes with a version tag
git commit -m "Version 2 specification"

# Optionally, you can create a tag to mark this version
git tag -a v2.0 -m "Version 2.0"
```

Create Another Version (e.g., Version 3):

```
# Make changes to the specification as needed
# Save the changes in your text editor
# Stage the modified specification file
git add specification.txt

# Commit the changes with a version tag
git commit -m "Version 3 specification"

# Create a tag for this version
git tag -a v3.0 -m "Version 3.0"
```

Step 5: Switch Between Versions

Switch between versions by checking out a specific commit or tag.

Switch to Version 2:
```
git checkout v2.0
```

Switch to Version 3:
```
git checkout v3.0
```

EXP 12

Sr. No.	Test Case	Expected Output	Actual Output	Pass/Fail
1	Verify that the user can add a product to the cart.	The product should be added to the cart successfully, and the cart total should be updated.	The product is not added to the cart, or the cart total is not updated.	Fail
2	Verify that the user can remove a product from the cart.	The product should be removed from the cart successfully, and the cart total should be updated.	The product is not removed from the cart, or the cart total is not updated.	Fail
3	Verify that the user can proceed to checkout with a non-empty cart.	The user should be able to proceed to checkout without any errors.	The user is unable to proceed to checkout, or an error message is displayed.	Fail
4	Verify that the user can enter a valid shipping address.	The shipping address should be accepted and saved successfully.	The shipping address is not accepted, or an error message is displayed.	Fail
5	Verify that the user can select a valid shipping method.	The shipping method should be selected successfully, and the shipping cost should be calculated.	The shipping method is not selected, or the shipping cost is not calculated correctly.	Fail
6	Verify that the user can enter a valid payment method.	The payment method should be accepted and saved successfully.	The payment method is not accepted, or an error message is displayed.	Fail
7	Verify that the user can successfully place an order.	The order should be placed successfully, and the user should receive a confirmation message.	The order is not placed successfully, or the user does not receive a confirmation message.	Fail