Part 1: Test Planning

1. Test Strategy Document

Objectives of Testing:

- Ensure the functionality, usability, and performance of the e-commerce website.
- Identify and mitigate risks related to security and data integrity.
- Validate that the website meets business requirements.

Scope of Testing:

- Cover all critical features, including:
  - User registration
  - Product search
  - Cart management
  - Product Listing Pages
  - Product Display Pages
  - Checkout
  - Order management
- Focus on cross-browser and cross-device compatibility.

Testing Levels:

- Unit testing for individual components.
- Integration testing for component interactions.
- System testing for end-to-end scenarios.
- Acceptance testing to validate against business requirements.

Testing Types:

- Functional testing to validate features.
- Usability testing to ensure a good user experience.
- Performance testing for responsiveness and load handling.

Entry Criteria:

- **Development Completion:**
  - All development activities for the e-commerce website should be completed.
- **Environment Setup:**
  - Test environment is set up and configured, including servers, databases, and necessary third-party services.
- **Code Freeze:**
  - The codebase should be frozen, and no further changes or additions to features are allowed during the testing phase.
- **Test Data Availability:**
  - Sufficient and representative test data is available for different scenarios.
- **Test Resources Ready:**
  - Testing tools, frameworks, and other resources are installed and configured.
- **Documentation Review:**

- Test strategy document and test plan are reviewed and approved by relevant stakeholders.

Exit Criteria:

- **Successful Test Execution:**
  - All test cases have been executed successfully, covering all specified features and scenarios.
- **Defect Resolution:**
  - All critical and high-priority defects identified during testing are resolved, and retested successfully.
- **Approval from Stakeholders:**
  - Stakeholders have reviewed the test results and provide approval for the release.
- **Performance Targets Met:**
  - Performance testing meets predefined targets, ensuring the website's responsiveness and load handling.
- **Usability Criteria Met:**
  - Usability testing confirms a positive user experience, and any usability issues are addressed.
- **Security and Data Integrity Assurance:**
  - Security testing has been conducted, and all identified security vulnerabilities are either resolved or have an agreed-upon risk mitigation plan.
- **Regression Testing Completion:**
  - Regression testing has been conducted to ensure that new changes did not introduce unexpected issues.
- **Test Documentation Review:**
  - Test strategy, test plan, and other documentation are reviewed and finalized.
- **Test Environment Cleanup:**
  - The test environment is restored to its original state, and any test-specific data is removed.
- **Final Test Report:**
  - A comprehensive final test report is generated, summarizing the testing process, results, and any remaining risks or issues.
- **Knowledge Transfer:**
  - Knowledge transfer sessions are conducted to share testing insights and findings with relevant teams.

Test Environment and Tools:

- Environment: Browsers (Chrome, Firefox, Safari), devices (desktop, tablet, mobile), operating systems (Windows, macOS, Linux).
- Tools: Selenium for automation, JMeter for performance testing.

Risk Analysis:

- Identify potential risks and their impact on testing and the project.
- Mitigation strategies for identified risks.

2. Test Plan

Test Deliverables:

- Test strategy document.
- Test cases.
- Automated test scripts.

Test Schedule:

- Define testing milestones and timelines.

Test Resources:

- Specify human resources, testing tools, and hardware.

Test Data and Environment Setup:

- Define test data requirements.
- Provide instructions for setting up the test environment.

Test Execution and Reporting:

- Define how tests will be executed and reported.
- Criteria for passing/failing test cases.

Part 2: Test Case Design

Functional Test Cases:

User Registration:

- Verify successful user registration with valid information.
- Test unsuccessful user registration with invalid information.

Product Search:

- Search for a product and verify results.
- Test search functionality with no results.

Add Items to Cart:

- Add items to the cart and verify the cart content.
- Test adding items with invalid or out-of-stock status.

Checkout Process:

- Complete the checkout process with valid details.
- Test checkout with incomplete or incorrect information.

Order Management:

- Verify order history and details.

- Test order cancellation and modification.

Edge and Boundary Test Cases:

- Test scenarios with extreme values for input fields (e.g., maximum and minimum character lengths, numeric ranges).

Part 3: Test Automation

5. Test Automation Framework

- **Framework Choice:** Selenium for web application testing.
- **Overview:** Selenium is widely used, supports multiple languages (Java, Python, etc.), and has a large community.

6. Automated Test Scripts

Critical Test Scenarios:

Automated test for User Registration:

- Create a script to automate user registration with valid information.
- Create a script to automate user registration with invalid information.

Automated test for Product Search:

- Create a script to automate product search and verify results.
- Create a script to automate search with no results.

Automated test for Checkout Process:

- Create a script to automate the checkout process with valid details.
- Create a script to automate checkout with incomplete or incorrect information.

Positive and Negative Test Cases:

- **Positive scenarios:**
  - Valid user registration.
  - Successful product search.
  - Successful checkout.
- **Negative scenarios:**
  - Invalid user registration.
  - Unsuccessful product search.
  - Failed checkout.

7. Test Data Management

Methods:

- Utilize test data files for different scenarios.
- Create scripts for data generation within the test.