**Test Plan for E-Commerce System**

**1. Introduction**

**1.1 Purpose**

The purpose of this test plan is to define the objectives, scope, strategy, resources, and schedule for testing the e-commerce platform. The goal is to ensure that the system meets functional and non-functional requirements and adheres to business needs.

**1.2 Scope**

The testing will cover:

* **Functional Testing**: Authentication, product browsing, shopping cart, payments, order management, and admin functionalities.
* **API Testing**: Verification of API endpoints for authentication, product data, checkout, and payments.
* **Performance Testing**: Load, stress, and scalability testing.
* **Security Testing**: SQL injection.
* **Usability Testing**: Ensuring a smooth user experience across multiple devices.

**1.3 Testing Objectives**

* Validate that the system meets business requirements.
* Identify and resolve defects early in the development cycle.
* Ensure that the platform supports multiple concurrent users efficiently.
* Verify that the e-commerce system works on various browsers and devices.
* Ensure secure transactions and data protection.

**2. Test Strategy**

**2.1 Testing Types**

1. **Unit Testing** – Performed by developers to test individual components.
2. **Integration Testing** – Ensuring modules work together (e.g., API and UI integration).
3. **System Testing** – End-to-end testing of the entire e-commerce flow.
4. **User Acceptance Testing (UAT)** – Validating business scenarios with stakeholders.
5. **Performance Testing** – Load, stress, and scalability testing.
6. **Usability Testing** – Checking UI/UX responsiveness and accessibility.

**2.2 Tools and Technologies**

* **UI Automation**: Selenium WebDriver, TestNG
* **API Testing**: Postman, Rest Assured
* **Performance Testing**: JMeter
* **Test Management**: Jira

**3. Test Deliverables**

* **Test Plan** – Defines strategy, objectives, and execution details.
* **Test Schedule** – Timeline for different testing phases.
* **List of Risks** – Identified risks and mitigation strategies.
* **Risk Register** – Documentation of risks, their impact, and mitigation plans.
* **Entry and Exit Criteria** – Conditions to start and complete testing.
* **Test Cases** – Detailed scenarios covering all functionalities.
* **Test Execution Report** – Logs test results, defects, and fixes.
* **Bug Reports** – Documentation of defects found.
* **Final Test Summary** – Provides insights into test coverage and results.

**4. Test Environment**

* **Operating Systems**: Windows
* **Browsers**: Chrome, Firefox, Edge
* **Devices**: Desktop, Mobile, Tablet
* **Test Environments**: Staging, UAT

**5. Test Schedule**

| **Phase** | **Activity** | **Timeline** |
| --- | --- | --- |
| Week 1 | Test Case Development | 5 days |
| Week 2 | Functional Testing | 7 days |
| Week 3 | Performance & Security Testing | 5 days |
| Week 4 | UAT | 7 days |
| Week 5 | Test Closure & Reporting | 3 days |

**6. Defect Management**

* Defects will be logged in **Jira**.
* Priority Levels:
  + **Critical** – Blocks major functionality.
  + **High** – Major defect with no workaround.
  + **Medium** – Defect with minor impact.
  + **Low** – Cosmetic or minor issue.

**7. Entry and Exit Criteria**

**Entry Criteria**

* Test environment is set up and stable.
* Test cases are reviewed and approved.
* Required test data is available.
* Development team has completed initial testing.

**Exit Criteria**

* All critical and high-priority defects are resolved.
* Functional and non-functional test cases have been executed successfully.
* UAT sign-off from business stakeholders.
* System performance meets defined benchmarks.

**8. Risks and Mitigation**

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Tight deadlines | Prioritize test cases and automate tests. |
| Last-minute changes | Implement continuous testing. |
| Security threats | Conduct regular security audits. |

**9. Risk Register**

| **Risk ID** | **Description** | **Impact** | **Probability** | **Mitigation Strategy** |
| --- | --- | --- | --- | --- |
| R001 | Tight deadlines may impact test coverage | High | Medium | Prioritize critical test cases and automate where possible |
| R002 | Unexpected last-minute feature changes | Medium | High | Implement continuous testing and have a flexible testing schedule |
| R003 | Security vulnerabilities may be exploited | High | Medium | Conduct regular security audits |
| R004 | Performance issues under high load | High | Medium | Perform load and stress testing early |

**10. Conclusion**

This test plan ensures a structured and systematic approach to testing the e-commerce platform. The combination of manual and automated testing will help achieve a robust, high-performing, and secure e-commerce system.