Test Plan for the Online Shopping Website Development: -

**Project:** Development of an Online Shopping Website

**Description:** This test plan outlines the approach for ensuring the quality and functionality of the online shopping website throughout the development process. It aims to identify and address issues early on, minimizing impact on timelines and budget.

**Testing Scope:** This plan covers all stages of development, including requirements gathering, user interface design, coding, integration, and deployment. It focuses on core functionalities like browsing, search, product details, shopping cart, checkout, account management.

**1.Testing Objectives:**

* Verify system functionalities meet user requirements and design specifications.
* Identify and resolve bugs and defects early in the development lifecycle.
* Measure system performance and optimize loading times.
* Ensure system security and prevent vulnerabilities.
* Enhance user experience through usability testing.
* Identify and mitigate integration issues between different modules.

processing, order tracking, and administrative features.

* Checks the compatibility with the different browsers and devices.

**2. Testing Scope**

* **Features:** User registration, product browsing, search, cart management, checkout, order tracking, account management, customer support.
* **Devices:** Desktop, laptops, tablets, smartphones.
* **Browsers:** Latest versions of Chrome, Firefox, Safari, Edge.
* **Operating Systems:** Windows, macOS, Android, iOS.

**3.Test Schedule**

START: [Insert Start Date]

End: [ Insert End Date]

**4.Testing Phases:**

1. **Requirements Review:** Analyse the user requirements documents for completeness, clarity, and potential errors.
2. **Unit Testing:** Test individual modules and components in isolation.
3. **Integration Testing:** Test interaction between different modules and systems.
4. **Functional Testing:** Verify end-to-end functionalities against requirements and user stories.
5. **Non-Functional Testing:** Assess performance, usability, security, accessibility, and compatibility aspects.
6. **Regression Testing:** Re-test existing functionalities after bug fixes or new features are implemented.
7. **Deployment Testing:** Ensure smooth transition from development to production environment.

**5.Testing Techniques:**

* **Manual Testing:** Exploratory and scripted testing on different functionalities.
* **Automated Testing:** Utilize frameworks and tools for repetitive tasks and regression testing.
* **Performance Testing:** Load testing, stress testing, and scalability testing using dedicated tools.
* **Usability Testing:**Observe real users interacting with the website and collect feedback.
* **Security Testing:** Penetration testing, vulnerability scanning, and code review for security flaws.

**6.Test Cases:**

Develop test cases for each functionality, covering positive and negative scenarios, edge cases, and user roles. Examples include:

* **Registration Test Case:** Verify successful registration with valid and invalid data.
* **Product Search Test Case:** Test searching by name, category, brand, price, filters, and combinations.
* **Search Test Case:** Test product search by name, category, brand, price range, etc.
* **Cart Test Case:** Add, remove, update items, apply discounts, checkout with different payment methods.
* **Checkout Test Case:** Test guest and registered user checkout, address validation, order confirmation.
* **Account Management Test Case:** Register, login, update account information, manage addresses, order history.
* **Performance Test Case:** Measure website loading times under various user loads and network conditions.
* **Usability Test Case:** Observe users navigating the website, identify confusing elements, and suggest improvements.

**6.1 Risk:**

Identify potential risks associated with the test case.

* + **Functionality**: Is there a high chance of unexpected behavior or failure?
  + **Performance**: May the test be resource-intensive, impacting other tests?
  + **Data:** Could specific data values or types lead to inaccurate results?
  + **Environment**: Could external factors like network issues affect the test?

Define the severity and likelihood of each risk. High-risk scenarios may require additional test cases or mitigation strategies.

**6.2 Dependencies:**

* Clearly identify any dependencies for the test case to execute successfully.
  + **Pre-requisite conditions:** Specific states or configurations needed before the test.
  + **Data dependencies:** Required data elements or test fixtures to be available.
  + **External systems:** Reliance on other applications or services functioning properly.
  + Documenting dependencies ensures the test setup is correct and avoids confusion and wasted effort.

**6.3. Approach:**

* + **Manual or automated testing:** Which method is employed for this particular case?
  + **Steps and actions:** Clearly define the sequence of actions to be performed.
  + **Expected outcomes:** Specify the desired results and success criteria for the test.
  + A defined approach ensures consistent execution and improves reproducibility of the test results.

**7.Test Data**

**7.1. Product Search:**

**Positive Cases:**

* + Search for existing products by different names, brands, categories, keywords.
  + Use combinations of search terms with special characters or numbers.
  + Test searching with synonyms or misspelled words.

**Negative Cases:**

* + Search for non-existent products or irrelevant terms.
  + Leave the search bar empty or enter invalid characters.
  + Simulate network errors or slow response times.

**7.2. Product Details:**

**Positive Cases:**

* + Verify product images, descriptions, and specifications are accurate and complete.
  + Test different user actions like "Add to Cart," "Compare Products," and "Add to Wishlist."
  + Confirm availability and price information under various scenarios.

**Negative Cases:**

* Test missing or incorrect product information.
* Try adding unavailable products to the cart.
* Simulate invalid price formats or unexpected values.

**7.3.Shopping Cart and Checkout:**

**Positive Cases:**

* + Add, remove, and update items in the shopping cart with different quantities.
  + Apply discount codes and promotional offers.
  + Test various payment methods (credit card, debit card, PayPal, etc.).
  + Validate address validation and order confirmation process.

**Negative Cases:**

* + Try adding invalid quantities or exceeding stock limits.
  + Use expired or invalid discount codes.
  + Simulate payment gateway errors or fraudulent transactions.
  + Test invalid or incomplete address information during checkout.

**8. Resources:**

* Testers with relevant expertise in web testing, performance testing, security testing, and usability testing.
* Testing tools and frameworks for different purposes.
* Access to development and test environments, user accounts, and test data.

**Testing Tools and Frameworks:** Tools and frameworks for different purposes, including:

* + **Web Testing:** Selenium, Cypress, BrowserStack, TestComplete
  + **Performance Testing:** JMeter, LoadRunner, Gatling
  + **Security Testing:** Burp Suite, OWASP ZAP, Nessus
  + **Usability Testing:** UserTesting, Lookback, Crazy Egg

**Testing Environments:**

* + **Development Environment:** Replica of the development environment for initial testing.
  + **Staging Environment:** Replica of the production environment for final testing before deployment.
  + **Production Environment:** Live environment for the actual online shopping website.

**User Accounts:** Different user accounts with varying roles and permissions for testing different functionalities.

**Test Data:** Pre-populated data sets representing various scenarios and edge cases for testing.

**9.Success Criteria**

* All critical functionalities are tested and functioning as expected.
* No major bugs or security vulnerabilities are identified after deployment.
* Performance meets acceptable standards and can handle expected user traffic.
* User feedback is incorporated to improve interface usability and navigation.

**10.Reporting and Defect Tracking**

Test results will be documented in detail, including screenshots and logs. A bug tracking system will be used to record, prioritize, and track defects until resolution.