Test Strategy for the Sauce Labs Demo website

1. Introduction

This document outlines the testing strategy for the online marketplace platform. The primary goal of this strategy is to ensure that the platform functions as expected, is user-friendly, secure, and performs optimally under different conditions. The strategy will cover functional, non-functional, security, and performance testing for various components of the marketplace.

2. Scope of Testing

- **In-Scope**:
  - User login, and authentication
  - Product listing and browsing
  - Shopping cart functionality
  - Payment gateway integration
  - Order processing and tracking
  - User profile management
  - Customer reviews and ratings
  - Admin panel for product management and reporting
- **Out-of-Scope**:
  - External payment gateway functionality (covered by third-party testing)
  - Mobile app testing (if applicable, separate testing strategy required)

3. Testing Types

- **Functional Testing**:
  - Verify that all features (e.g., product sorting, cart addition, checkout process) work as expected.
  - Ensure correct error handling and messaging.
  - Test the integration of third-party services like payment gateways.
- **Usability Testing**:
  - Evaluate the user interface for ease of use, navigation, and clarity of instructions.
  - Perform exploratory testing based on common user scenarios.
- **Compatibility Testing**:
  - Test on different browsers (Chrome, Firefox, Safari, Edge).
  - Test responsiveness on various devices (desktop, tablet, mobile).
- **Performance Testing**:
  - Load testing to simulate multiple users (e.g., peak traffic).
  - Stress testing to evaluate how the system behaves under extreme conditions.
- **Security Testing**:
  - Verify secure authentication methods (e.g., OAuth, two-factor authentication, inputs validation).

    - Test for vulnerabilities like SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).
    - Ensure sensitive data like passwords and payment details are encrypted.
- **Regression Testing**:
    - Ensure that new features do not impact existing functionality.

4. Test Environments

- **Live Environment**: Environment in the public domain that does not contain real Data.
- **Staging Environment**: A replica of the production environment used for integration testing, performance testing, and final validation.
- **Test Data**: Use both real (scrubbed) and synthetic test data to simulate various scenarios (e.g., different product categories, order quantities, customer profiles).

5. Test Tools

- **Test Automation**:
    - Playwright (TS binding) + Cucumber (BDD) for automating functional tests (browser-based testing).
    - JUnit/TestNG for writing and managing test cases.
    - Postman for API testing.
- **Performance Testing**:
    - Apache JMeter for load and stress testing.
- **CI tool**:
    - GitHub Actions.
- **Bug Tracking**:
    - Jira to report and track bugs.
    - X-ray Jira extension to maintain test cases and test scenarios.

6. Test Execution and Schedule

- **Test Phases**:
    - **Unit Testing**: Developers perform unit testing during development.
    - **Integration Testing**: After the development of modules, integration testing ensures that they interact correctly.
    - **System Testing**: The entire platform is tested to ensure that all components work together as intended.
    - **User Acceptance Testing (UAT)**: End-users validate the functionality and usability of the platform.
    - **Regression Testing**: Conducted after bug fixes or new features to ensure no existing functionality is broken.
- **Schedule**:
    - Sprint-based testing (2-week sprints).
    - Each sprint will include functional, regression, and performance testing.
    - A final round of testing will be conducted after all features are implemented and prior to the release.
- **Automation and Manual Execution:**

- On the project, while both Automation and Manual tests are provided, the goal is to achieve as much test automation as possible. Main focus of the testers team is to make sure the quality of existing product and upcoming releases is on acceptable level and focus on automating existing and new test cases and scenarios.

7. Risk Mitigation

- **Unknown Defects**: Comprehensive testing across multiple phases and close collaboration between developers and testers will minimize unknown defects.
- **Time Constraints**: To mitigate time pressure, automated tests for critical workflows (login, checkout) will be prioritized, and performance testing will be executed early.
- **Third-Party Dependencies**: Test the integration points with third-party services in isolation to ensure no disruptions during the final integration.

8. Exit Criteria

- All critical and major defects identified during testing are resolved.
- Test coverage is at 90% or higher for the critical features.
- Automation test coverage is at 90% of existing test cases.
- The application meets performance and security requirements under expected loads.
- The user interface is user-friendly and compatible across supported browsers and devices.

9. Most Important and Least important Features

- **Most Important**:
  - User Profile (Login feature).
  - Shopping feature (products correct representation and adding to the cart).
  - Checkout and payment feature.
- **Least important:**
  - Sorting functionality.
  - About shop page.
  - Social media connections.

10. Conclusion

This testing strategy aims to ensure the success of the online marketplace by thoroughly validating its functionality, usability, security, and performance. By following this strategy, we aim to deliver a reliable and seamless platform for users and administrators.