**STRAIV TEST DOCUMENT**

# **Introduction**

The website [Automation-Exercise](https://www.automationexercise.com/) serves as a platform for users to practice and enhance their testing skills. The website simulates real-world e-commerce operations, including user registration, login, product searches, cart management, and payment workflows, offering an environment for developing and refining testing techniques.

The main objective is to ensure that the platform performs optimally under various conditions, providing a smooth user experience. This involves testing functionality, usability, performance, and compatibility across devices and browsers.

Since this platform is widely used for learning and educational purposes, it's vital to ensure its reliability, responsiveness, and security to guarantee a positive experience for users. Identifying potential risks and defects early on will help in maintaining the stability and accessibility of the platform across different devices and environments. This document outlines the approach, scope, risks, and priorities associated with achieving these objectives.

This report focuses on evaluating the key functionalities of the website, identifying potential issues, and providing insights on how to improve performance and usability through structured testing approaches.

# **Testing Approach**

The test approach defines the strategies and methods that will be used to assess the functionality, performance, usability, and security of the Automation Exercise website. It ensures a structured and comprehensive evaluation across all key areas, applying a combination of manual and automated testing techniques to guarantee thorough coverage.

## **Testing Types**

### **Functional Testing:**

* + - **Objective:** Validate that all key user interactions (e.g., User registration, login, product management, checkout, etc.) function as intended.
    - **Approach:** Perform both positive and negative testing to verify correct behavior in expected conditions and the system's handling of invalid inputs or actions.
    - **Tools:** Manual testing for basic flows, Cypress for automating functional tests.

### **Integration Testing:**

* + - **Objective:** Verify the interactions between various system components, such as the connections between login and registration or between the product page and the shopping cart.
    - **Approach:** Ensure smooth integration across all modules, with a particular focus on validating the data flow between different sections of the website.
    - **Tools:** Cypress will be used to automate user workflows and interactions across multiple components.

### **Regression Testing**:

* + - **Objective:** Confirm that recent updates, bug fixes, or new features do not adversely affect the existing functionality of the website.
    - **Approach:** Automate key test scenarios using Cypress to quickly identify any unintended side effects caused by changes to the system.
    - **Tools:** Cypress automation for detecting potential regressions.

### **User Interface (UI) and User Experience (UX) Testing:**

* + - **Objective:** Ensure the website provides an intuitive, user-friendly experience with a clean design and consistent performance across various browsers and devices.
    - **Approach:** Conduct manual testing to assess usability and responsiveness across a range of devices, ensuring the user interface is visually appealing and easy to navigate. Confirm that all interactive elements, such as buttons, menus, and links, function correctly.
    - **Tools:** Use browser developer tools to inspect responsiveness and layout, along with manual testing on desktop, tablet, and mobile devices.

### **Compatibility Testing**:

* + - **Objective:** Verify that the website is fully functional and visually consistent across multiple browsers (Chrome, Firefox)
    - **Approach:** Perform cross-browser testing to ensure that all features work the same across different platforms. Include mobile devices in testing to ensure the design is responsive.
    - **Tools:** manual testing using different browsers

## **Testing Phases**

* + **Requirement Analysis:** Identify the essential functional and non-functional requirements of the website, focusing on key user interaction flows, such as registration, login, product management, and checkout.
  + **Test Planning and Test Case Design:** Develop detailed test cases for each identified workflow, incorporating both positive and negative scenarios. Prioritize testing for critical paths, such as user registration and checkout processes.
  + **Test Environment Setup:** Ensure that the testing environment is configured with proper configurations for browsers, devices, and network conditions. Ensure that required test data is available for user interactions.
  + **Test Execution:** Perform both manual and automated testing in alignment with the established test cases. Execute tests across various browsers and devices to ensure compatibility across platforms.
  + **Defect Reporting and Fixing:** Log any bugs or issues discovered during testing and prioritize them for resolution. Report defects in a bug-tracking system and track their status.
  + **Regression Testing: Re-execute** test cases after bug fixes to verify that no existing functionality is affected by recent changes.
  + **Test Closure: Once** all critical defects are resolved and the website meets the quality standards, conclude the testing phase by delivering a final sign-off report summarizing the results.

## **Automation Testing Approach**

* + **Tools:** Cypress (for UI automation)
  + **Scope:** Automate the core functional workflows: user registration, login,
  + **Execution:** Schedule automated tests to run periodically

## **Exit** **and** **Entry** **Criteria**

#### **Entry Criteria**:

* + - Test environment is set up and functional.
    - Test cases are ready and approved.
    - Test data is available.

#### **Exit Criteria**:

* + - All critical and high-priority test cases have been executed.
    - All identified defects have been resolved or deferred for appropriate reasons.
    - Key functional workflows are validated as stable across multiple browsers and devices.

## **Defect** **Management**

* + **Defect Reporting:** Log defects with details like steps to reproduce, expected results, actual results, screenshots, and severity level.
  + **Defect Lifecycle:** Assign defects to the relevant developers, retest after fixes, and close the defect once resolved.

## **Metrics for Evaluation**

* + **Test Coverage:** Percentage of test cases executed vs. planned.
  + **Defect Density:** Number of defects found per module or feature.
  + **Test Pass/Fail Ratio:** Ratio of passed to failed test cases.
  + **Automation Coverage:** Percentage of manual test cases automated.

# **Bug Report**

Please refer to the Bug Reports document added in the folder.

# **Test Plan**

Please refer to the Test Plan document added in the folder.

# **Test Cases**

Please refer to the Test Cases document

# **Automation Scripts**

Please refer to the below GitHub link to access the documents and the source code. All relevant documentation and information regarding the source code should be available in the repository.

GitHub: [AutomationScripts GitHub](https://github.com/MeghaVernekar23/straiv_assesment_task.git)