Test Plan: AutomationExercise Website Testing

1. Introduction

Project Overview

Automation engineers tend to practice and improve their abilities in automated and API testing situations by using AutomationExercise, a training website.

Purpose and Objectives

This test plan's primary goals are to:

• Use Playwright automation to validate functionality throughout the AutomationExercise website's many elements.
• Write automated test scripts and run them using predefined user stories and acceptance standards.
• Create a test plan to guarantee that every aspect of website functionality is covered.
• Boost output with effective testing procedures and automation.
• Document findings from tests, along with analysis and suggestions for upgrades or repairs.

2. Project Setup

Tools and Technologies Used

- Automation Framework: Playwright
- Programming Language: TypeScript
- Runtime Environment: Node.js
- IDE: Visual Studio Code (VS Code)
- Version Control: Git

3. Test Strategy

Testing Approach

• Automation Focus: To accomplish reliable and repeatable testing, use Playwright for browser automation.
• Types of Testing: Regression, integrated, functional, and usability testing are among them.
• Testing Levels: Discuss user acceptance testing (UAT), system testing, integration testing, and unit testing.
• Entry and Exit Criteria: Specify what has to happen before and after each testing step.

Test Scenarios and Coverage

• Functional Testing: Check that key features such as user registration, login, product browsing, and checkout procedures work as intended.
• Usability testing: Verify the accessibility and intuitiveness of user interface components.
• Integration Testing: Examine how various modules and external APIs interact with one another.
• Performance testing: Evaluate the load management and responsiveness of websites.
• Security testing: Examine procedures for user authentication and data processing for weaknesses.
• Compatibility testing: Examine various operating systems, devices, and browsers.

4. Test Cases

https://mayumishehara517.atlassian.net/projects/SCRUM?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.kanoah.test-manager__main-project-page#!/v2/testCases?projectId=10000

5. Test Environment

Hardware and Software Requirements

- Operating System: Compatible with Windows, macOS, and Linux.
- Browser Compatibility: Chrome, Firefox, Safari, Edge.
- Tools and Dependencies: Playwright, Vscode, Nodejs

6. Test Execution

Test Execution Plan

• Execution Schedule: Set deadlines for testing according with project milestones.
• Execution Methodology: Use Playwright to run automated test scripts, and manually check test cases as necessary.
• Reporting: Use an identical structure to record test findings, including defects and observations.

7. Risk Management

Risk Identification and Mitigation

• Recognize possible dangers such compatibility problems, data security flaws, or environmental problems.
• Reduce risks by working with development teams, preparing for emergencies, and implementing proactive testing techniques.

8. Deliverables

- **Test Reports**: Generate detailed reports summarizing test results, including metrics on test coverage, pass/fail rates, and defect density.
- **Documentation**: Update documentation with any changes, issues encountered, and resolutions implemented.

9. Approval and Sign-Off

• Identify those stakeholders who will be reviewing and giving their approval the test plan and any related deliverables.
• Obtain approval to move on with the project's later phases and the execution of the tests.