1. **Project Planning & Management**

1.1 Introduction

Automation Exercise is a fully functional e-commerce website designed specifically for automation engineers and software testers.

The website includes various scenarios that simulate an actual online shopping platform, making it an ideal resource for learning and applying test automation techniques.

 The website provides a realistic environment for testing various web application functionalities, making it an invaluable resource for QA engineers, developers, and anyone interested in mastering automation tools like Selenium, Cypress, or Playwright.

1.2 Purpose

 Its primary goal is to offer hands-on experience with real-world scenarios, including user registration, login, product browsing, cart management, and checkout processes.

By simulating an e-commerce platform, AutomationExercise.com enables users to build, debug, and refine their automation scripts in a safe and controlled setting. This makes it an essential tool for improving efficiency, accuracy, and confidence in automation testing.

The purpose of this document is to outline the testing strategy and scope for **AutomationExercise.com**, a platform designed to help users practice and improve their test automation skills. The goal is to ensure that the website functions as intended, providing a seamless and error-free experience for users. This document will serve as a guide for the testing team to systematically validate the website's features, identify potential issues, and ensure the platform meets its objectives of offering a realistic and reliable environment for automation testing practice.

1.3 Risks And Challenges

**2. Risks**

**2.1 Security Risks**

* **Data Breaches:** The platform may be vulnerable to cyber-attacks that could compromise user data.
* **Unauthorized Access:** Poor authentication mechanisms could allow unauthorized users to exploit the platform.
* **Injection Attacks:** SQL injection and cross-site scripting (XSS) threats may arise if input validation is inadequate.

**2.2 Performance Risks**

* **Scalability Issues:** High traffic from users running automation scripts simultaneously may lead to server overload.
* **Slow Response Time:** Excessive API requests and database queries might degrade website performance.
* **Downtime Risk:** Insufficient infrastructure could result in frequent downtime, disrupting users’ testing processes.

**2.3 Functional Risks**

* **Inconsistent Test Results:** Variability in website behavior may lead to unreliable automation script execution.
* **Broken Test Scenarios:** Updates to the website’s UI/UX may break existing test cases, requiring continuous script updates.
* **Data Loss:** Accidental deletions or updates to the test database may affect test reliability.

**2.4 Compliance Risks**

* **GDPR and Data Privacy Issues:** Handling user data without proper compliance with international data protection laws could lead to legal consequences.
* **Intellectual Property Risks:** Unauthorized use of third-party assets may result in legal claims.

**3. Challenges**

**3.1 Maintaining Stability**

* Ensuring a stable environment for automation testing despite frequent website updates is a significant challenge.
* Handling concurrent user requests efficiently to prevent performance degradation.

**3.2 Security Enhancements**

* Implementing robust authentication mechanisms to prevent unauthorized access.
* Regularly performing security audits and vulnerability assessments to mitigate risks.

**3.3 Keeping Test Cases Updated**

* Continuous maintenance of test cases is required to align with UI/UX changes.
* Automated regression testing must be implemented to detect and resolve broken tests promptly.

**3.4 Resource Management**

* Ensuring the platform has sufficient infrastructure to handle increasing user traffic.
* Optimizing database and API performance to support seamless execution of test scripts.

**3.5 User Experience Optimization**

* Enhancing the website interface to provide a smooth and user-friendly experience.
* Providing clear documentation and tutorials for new users to understand automation testing effectively.

1.4 KPIs

**1. Website Performance Metrics**

* **Page Load Speed:** *(e.g., 3.5s on desktop, 5.2s on mobile)*
* **Bounce Rate:** *(e.g., 45%)*
* **Average Session Duration:** *(e.g., 3 minutes)*
* **Number of Pages per Session:** *(e.g., 4 pages)*

**2. User Engagement Metrics**

* **Total Users:** *(e.g., 50,000/month)*
* **Returning Users vs. New Users:** *(e.g., 35% returning, 65% new)*
* **Click-Through Rate (CTR):** *(e.g., 4.2%)*
* **Conversion Rate:** *(e.g., 2.8%)*

**3. SEO & Traffic Sources**

* **Organic Traffic:** *(e.g., 60% of total traffic)*
* **Direct Traffic:** *(e.g., 20%)*
* **Referral Traffic:** *(e.g., 10%)*
* **Social Media Traffic:** *(e.g., 10%)*
* **Top Ranking Keywords:** *(e.g., "Automation Testing Practice", "QA Exercises Online")*

**4. Technical & Security KPIs**

* **Mobile Responsiveness Score:** *(e.g., 85/100)*
* **Security Vulnerabilities Detected:** *(e.g., No critical issues)*
* **Uptime Percentage:** *(e.g., 99.98%)*
* **Broken Links Detected:** *(e.g., 5 broken links)*

**5. Business & Revenue Metrics**

* **Revenue (if applicable):** *(e.g., $5,000/month from ads and premium content)*
* **Customer Acquisition Cost (CAC):** *(e.g., $2 per user)*
* **Customer Lifetime Value (CLV):** *(e.g., $10 per user)*
* **Subscription Rate (if any):** *(e.g., 5% of users subscribe)*

**6. Customer Satisfaction & Feedback**

* **User Satisfaction Score (CSAT):** *(e.g., 85%)*
* **Net Promoter Score (NPS):** *(e.g., 8.5/10)*
* **Support Ticket Resolution Time:** *(e.g., 24 hours average)*

**Recommendations for Improvement**

1. **Optimize Page Load Speed:** Reduce image sizes, leverage caching, and optimize scripts.
2. **Improve Engagement:** Add more interactive exercises and gamification features.
3. **Enhance SEO Strategy:** Focus on backlinking and long-tail keywords.
4. **Fix Broken Links:** Regularly check and update internal and external links.
5. **Increase Conversion Rates:** Improve call-to-action placement and A/B test landing pages.

1. **Testing Scope**

2.1 Scope

The scope of this testing effort includes the evaluation of key features and pages on **AutomationExercise.com**, such as:

* Home Page
* Login/Signup Page
* Products Page
* Product Details Page
* Cart Page
* Checkout Page
* Order History/Confirmation Page
* Contact Us Page
* Footer and Miscellaneous Links
* Error Handling and Edge Cases

The testing will cover functional, usability, and edge-case scenarios to ensure the website performs as expected under various conditions. This scope ensures comprehensive coverage of the platform's core functionalities, enabling the team to deliver a high-quality user experience.

2.2 Testing Scope ( Features & Pages ) To Be Tested

The testing scope for **AutomationExercise.com** will focus on the following key features and pages to ensure the website functions as expected and delivers a seamless user experience:

**1. Home Page**

* Verify the layout, design, and responsiveness across devices.
* Test navigation links (e.g., Home, Products, Cart, Login/Signup).
* Check for the presence of key elements (e.g., header, footer, banners, recommended items).

**2. Login/Signup Page**

* Validate user registration functionality (valid/invalid inputs, error messages).
* Test user login functionality (valid/invalid credentials, error handling).
* Check "Forgot Password" functionality (if applicable).

**3. Products Page**

* Verify product listing, sorting, and filtering options.
* Test product search functionality.
* Validate product details (e.g., images, descriptions, pricing).

**4. Product Details Page**

* Check the accuracy of product information.
* Test "Add to Cart" functionality.
* Verify quantity selection and updates.

**5. Cart Page**

* Validate the items added to the cart.
* Test quantity updates and item removal.
* Verify the total price calculation.

**6. Checkout Page**

* Test the checkout process (e.g., address, payment details).
* Validate error handling for incomplete/incorrect inputs.
* Verify order confirmation and success messages.

**7. Order History/Confirmation Page**

* Check if the order details are displayed correctly.
* Verify the ability to view past orders (if applicable).

**8. Contact Us Page**

* Test the contact form submission (valid/invalid inputs, error messages).
* Verify email notifications (if applicable).

**9. Footer and Miscellaneous Links**

* Test links to social media, privacy policy, terms of service, etc.
* Verify the functionality of newsletter subscription (if applicable).

**10. Error Handling and Edge Cases**

* Test for 404 pages, broken links, and invalid URLs.
* Validate behavior for edge cases (e.g., empty cart, out-of-stock items).

This scope ensures comprehensive coverage of the website's core functionalities, providing a robust evaluation of its performance, usability, and reliability. The testing will include functional, usability, and edge-case scenarios to ensure the platform meets its objectives and delivers a high-quality user experience.

2.3 Testing Role

Testing Roles

1. Mohamed Mondy (Team Leader)

* Responsibilities:
  + Oversee the entire testing process.
  + Assign tasks to the team and monitor progress.
  + Review final reports and ensure testing quality.
  + Communicate with stakeholders (e.g., developers or management).

2. Omar Ahmed

* Role: Manual Testing
  + Test the website's UI and functionality manually.
  + Focus on user flows, navigation, and edge cases.
  + Verify responsiveness and usability across devices.
  + Multilingual Testing (Language switching, text alignment)
  + Performance Testing (Page load speed, slow internet behavior)

3. Suzana

* Role: API Testing
  + Test the backend APIs for functionality, performance, and security.
  + Validate endpoints, request/response structures, and error handling.
  + Boundary Testing (Edge cases & invalid values).
  + Environment Testing (Staging, Production, Development).
  + Failure Testing (Server crashes & request overload).
  + Concurrency Testing (Handling multiple users simultaneously).
  + Use tools like Postman or Swagger for API testing.

4. Mariam Darwish

* Role: Automation Testing
  + Develop and execute automated test scripts for critical workflows.
  + Use tools like Selenium, Cypress, or Playwright.
  + Focus on regression testing and repetitive test cases.
  + Cross-browser Testing – Ensure compatibility across different browsers and devices.

5. Mohamed Ayman

* Role: Manual Testing
  + Test specific features like login/signup, cart, and checkout.
  + Validate error messages, form submissions, and user inputs.
  + Perform exploratory testing to identify unexpected issues.
  + Session Timeout Testing (Automatic logout after inactivity).
  + Payment & Subscription Testing (Transaction success & failure scenarios).

6. Mina Magdy

* Role: Automation Testing
  + Collaborate with Mariam on automation scripts.
  + Focus on automating test cases for product pages and order history.
  + Ensure scripts are maintainable and reusable.
  + CI/CD Integration – Automate test execution within the CI/CD pipeline upon code updates.

Workflow

1. Manual Testing Team (Omar & Mohamed Ayman):

* Start with exploratory testing and basic functionality checks.
* Document bugs and share them with the team.

2. API Testing Team (Suzana):

* Test backend APIs independently and ensure they meet requirements.
* Share API test results with the automation team for integration.

3. Automation Testing Team (Mariam & Mina):

* Develop automated scripts for repetitive and critical test cases.
* Run regression tests after each release or update.

4. Team Leader (Mohamed Mandy):

* Coordinate between teams and ensure deadlines are met.
* Review test reports and communicate progress to stakeholders.

Testing Roles Assignment Table

|  |  |  |
| --- | --- | --- |
| Team Member | Role | Responsibilities |
| Mohamed Mondy | Team Leader | Oversee testing process, assign tasks,  review reports,  communicate with stakeholders. |
| Omar Ahmed | Manual Testing | Test UI, navigation, responsiveness, and edge cases manually. |
| Suzana | API Testing | Test backend APIs for functionality, performance, and security using tools like Postman.  Assist with API testing, validate frontend-backend integration, and performance. |
| Mariam Darwish | Automation Testing | Develop and execute automated test scripts using tools like Selenium or Cypress. |
| Mohamed Ayman | Manual Testing | Test specific features (login, cart, checkout) and validate error handling. |
| Mina Magdy | Automation Testing | Collaborate on automation scripts, focus on product pages and order history. |

Workflow Table

|  |  |  |
| --- | --- | --- |
| Phase | Team Members Involved | Tasks |
| Manual Testing | Omar Ahmed, Mohamed Ayman | Test UI, navigation, forms, and edge cases. Document bugs. |
| API Testing | Suzana | Test backend APIs, validate endpoints, and ensure integration. |
| Automation Testing | Mariam Darwish, Mina Magdy | Develop and run automated scripts for regression and critical workflows. |
| Reporting & Review | Mohamed Mondy | Review test results, communicate progress, and ensure quality. |