

Automation Testing Project

**Testing
ECommerce Web
Application**
(<https://automationexercise.com/>)

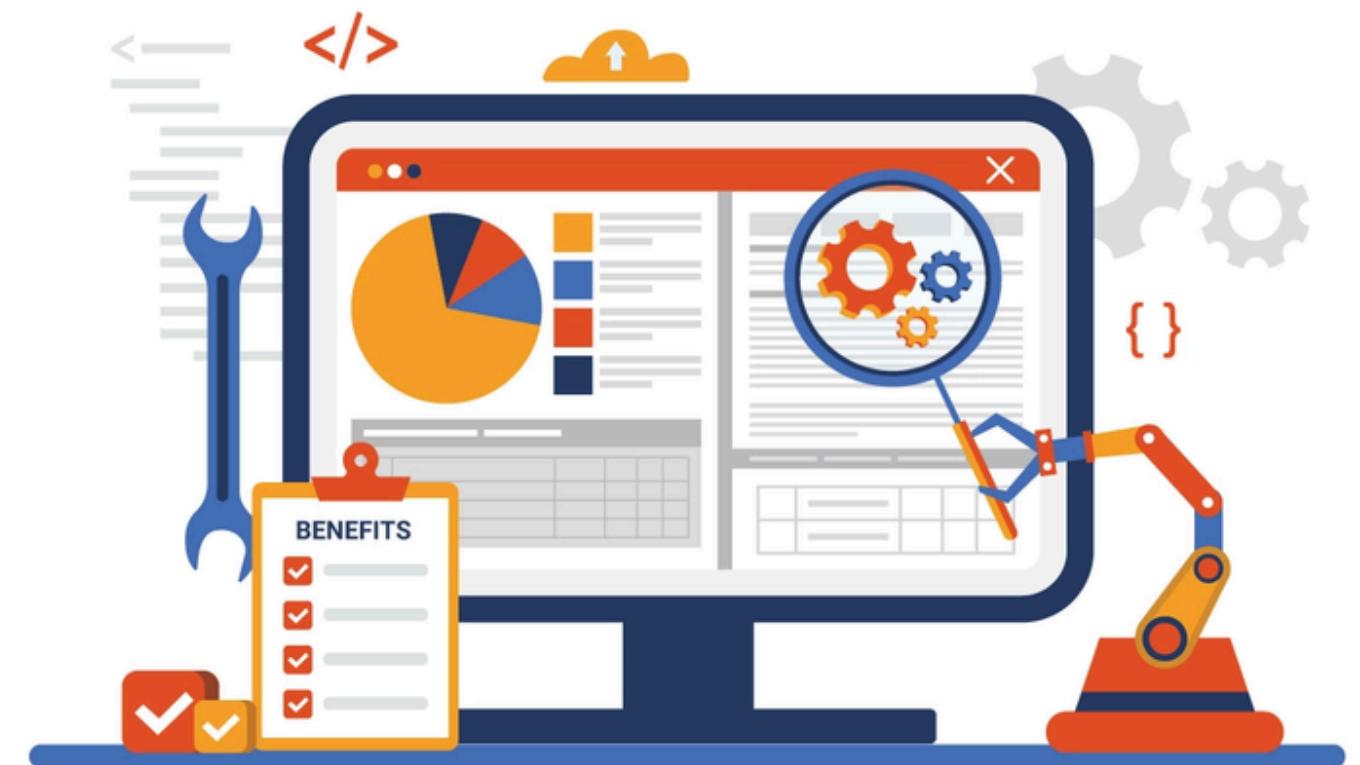
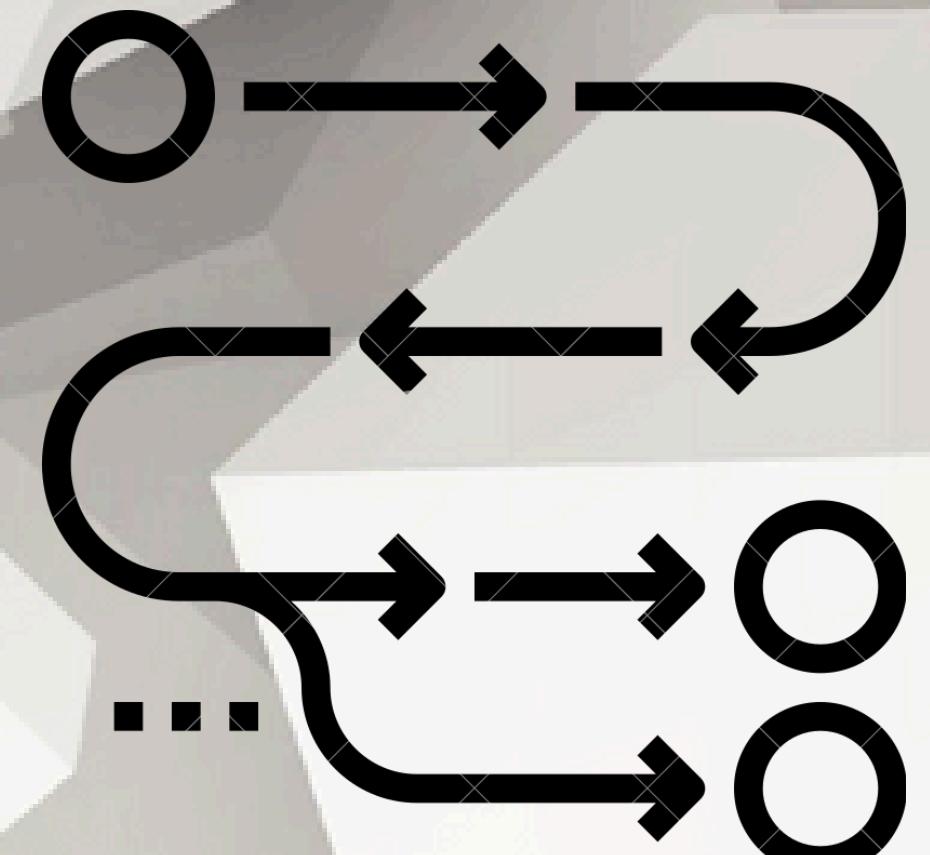


Table of contents

- 1. Problem Statement
- 2. Objective
- 3. Proposed Automation Tools
- 4. Manual test cases Demo
- 5. Automated testcase Demo
- 6. Sample Reports
- 7. Defect Report
- 8. Jenkins Integration
- 9. Conclusion



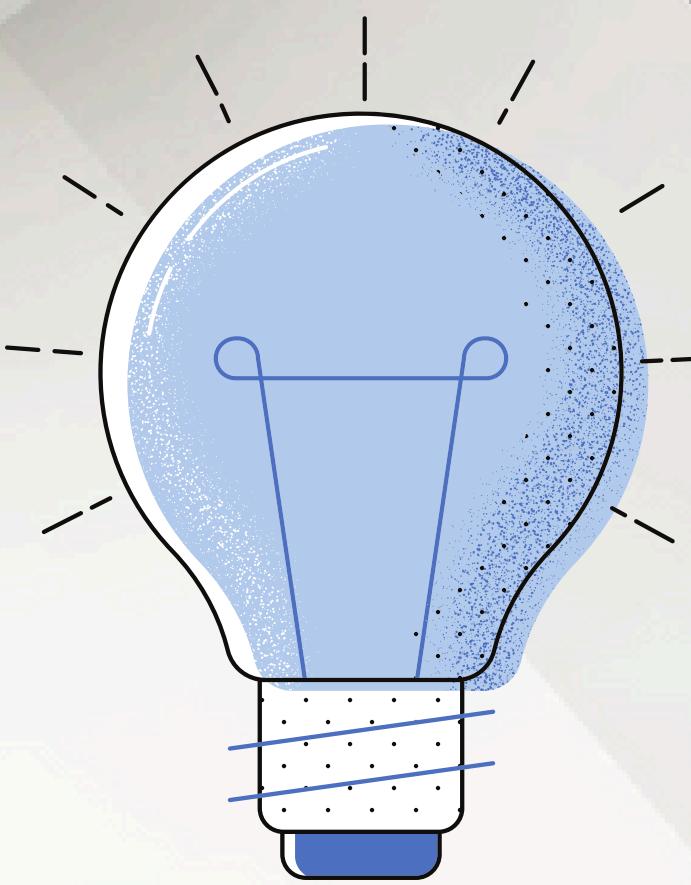
Problem Statement

- The goal of this project is to automate critical functionalities and workflows on a sample webpage used for demonstration and testing purposes.
- The webpage includes features like navigation, forms, buttons, links, and data validation, which are essential for user interaction.
- The project aims to automate key functionalities like user registration, login, product search, add to cart, and checkout using Java and Selenium Web Driver



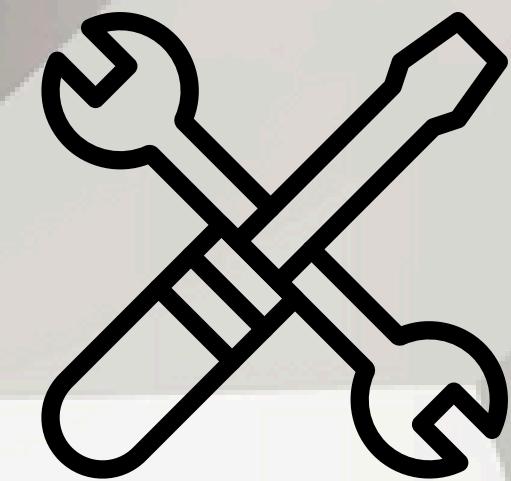
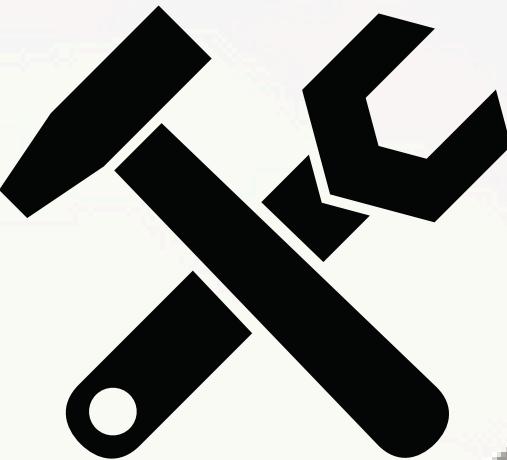
Objective

- Reduce testing time by automating repetitive web test cases.
- Improve test coverage and ensure reliability across different environments.
- Detect bugs and issues early in the development cycle.
- Enable Continuous Integration/Continuous Testing workflows.
- Provide detailed, reproducible reports for better decision-making.
- Facilitate easier maintenance and scalability as the application grows.



Proposed Automation Tools

- Selenium WebDriver – For browser automation across Chrome, Firefox, Edge, etc.
- TestNG – Framework to manage test cases, execution flow, and reporting.
- Maven – For project dependency management and build automation.
- ExtentReports – To generate user-friendly and insightful test reports.
- Git/GitHub – For version control and collaborative development.
- Browser Drivers (ChromeDriver, SafariDriver, etc.) – To interact with web elements.



Manual Test Cases

Test Case For E-Commerce Web Application HomePage												
Index	Test Case ID	Objective	Pre-requisites	Steps	Test Data	Expected Result	Actual Result	Proof	Status	Can be automated	Done By	
1	TC_ECommerce_WebApp_Home_001	To verify and validate the valid URL is working on different browsers (Chrome, Safari, Edge).	URL..	Step - 1: Open Different Browsers Step - 2: Paste URL, Step - 3 : Click enter	URL: http://automationexercise.com/0/home_safari_Edge	User should be able to see the Dashboard / home page	Working as expected		Pass	DONE	VK	
2	TC_ECommerce_WebApp_Home_002	To verify and validate the Home button functionality on the home page.	Home page should be open	Step - 1: Open Chrome Browser Step - 2: Paste URL, Step - 3 : Click enter Step - 4 : Click on Home button.	URL: http://automationexercise.com/0/home	User should be able to see the Dashboard / home page after clicking on Home button	Working as expected		Pass	DONE	VK	

[Click here to open manual test cases](#)

Automated Test Cases

Structure of our project

```
Finalproj
  src/main/java
    com.automationproj.base
    com.automationproj.config
    com.automationproj.pages
    com.automationproj.utilities
  src/main/resources
  src/test/java
    Base
    Tests
  src/test/resources
    Reports
    Screenshots
    TestData
  JRE System Library [JavaSE-17]
  Maven Dependencies
  Referenced Libraries
  TestNG
  src
  target
  test-output
  pom.xml
  README.md
  testng.xml
```

Sample code snippet

```
1 package Tests;
2
3 import com.automationproj.base.*;
4
5 import Base.BaseTest;
6
7 import org.openqa.selenium.By;
8 import org.openqa.selenium.WebElement;
9 import org.testng.Assert;
10 import org.testng.annotations.Test;
11
12 public class TestAutomation extends BaseTest {
13
14     String url = "https://automationexercise.com/";
15
16     @Test(priority = 1)
17     public void verifyUrlLoads() {
18         driver.get(url);
19         String currentUrl = driver.getCurrentUrl();
20         Assert.assertTrue(currentUrl.contains("automationexercise.com"), "URL did not load correctly");
21     }
22
23     @Test(priority = 2)
24     public void verifyHomeButton() {
25         driver.get(url);
26         WebElement homeButton = driver.findElement(By.xpath("//a[normalize-space()='Home']"));
27         homeButton.click();
28
29         String currentUrl = driver.getCurrentUrl();
30         Assert.assertTrue(currentUrl.contains("automationexercise.com"), "Home button did not work correctly");
31     }
32 }
33 }
```

Sample Reports

Tests

Test Case	Status
testCartIconOnLoginPage on chrome	Pass
testHomeIconOnLoginPage on chrome	Pass
testLoginIconOnLoginPage on chrome	Pass
testProductIconOnLoginPage on chrome	Pass
testTestCasesIconOnLoginPage on chrome	Pass

testCartIconOnLoginPage on chrome

09.03.2025 12:30:07pm 09.03.2025 12:30:10pm 00:00:03:098 · #test-id1



STATUS	TIMESTAMP	DETAILS
Pass	12:30:10 pm	Test Passed

This report reflects a successful round of automated tests for the login page. It confirms that the key elements work properly, the tests are stable, and the framework generates useful, actionable reports for developers and testers. It's a clear demonstration of how automation improves test accuracy, efficiency, and documentation.

Defect Report

Defect Report																		
#	Defect ID	Title/Summary	Description	Module/Features	Severity	Priority	Environment	Steps to Reproduce	Expected Result	Actual Result	Proof	Reported By	Reported Date	Assigned To	Status	Priorit Date	Remark	
#	1 BUG_REG_01	Input issues with Name Field	To verify and validate the name text box in the sign up page with invalid input	Registration	Medium	Medium	BIT	Step - 1: Open the below URL, https://www.automationexercise.com Step - 2: Click on login/signup Step - 3: Enter the invalid input into the name text box.	User should not able to enter invalid input in the name textbox (Eg : 12345) and it should throw a pop-up message.	User is able to input 12345.		Venkata Kalyan	13-08-2023	Developer-1	New			
#	2 BUG_REG_02	Input issues with Name Field inside registration	To verify and validate the name text box in the registration page with invalid input	Registration	Medium	Medium	BIT	Step - 1: Open the below URL, https://www.automationexercise.com/signup Step - 2: Enter the invalid name in the name text box.	User should not able to enter invalid input in the name textbox (Eg : 12345) and it should throw a pop-up message.	User is able to input 12345.		Venkata Kalyan	13-08-2023	Developer-1	New			
#	3 BUG_REG_03	Input issues with First Name Field	To verify and validate the first name text box in the sign up page with invalid input	Registration	Medium	Medium	BIT	Step - 1: Open the below URL, https://www.automationexercise.com/signup Step - 2: Enter the invalid first name in the text box.	User should not able to enter invalid input in the name textbox (Eg : 12345) and it should throw a pop-up message.	User is able to input 12345.		Venkata Kalyan	13-08-2023	Developer-1	New			
#	4 BUG_REG_04	Input issues with Second Name Field	To verify and validate the last name text box in the sign up page with invalid input	Registration	Medium	Medium	BIT	Step - 1: Open the below URL, https://www.automationexercise.com/signup Step - 2: Enter the invalid last name in the text box.	User should not able to enter invalid input in the name textbox (Eg : 12345) and it should throw a pop-up message.	User is able to input 12345.		Venkata Kalyan	13-08-2023	Developer-1	New			

A structured document that records bugs or issues identified during automated or manual testing, providing details for developers to investigate and resolve. Automated tests link defects to specific test cases and build versions, ensuring faster resolutions and improved product quality.

[Click here to open defect report](#)

Jenkins Integration



Why Jenkins?

- Automates test execution by integrating with version control, enabling Continuous Integration (CI) for faster feedback and higher reliability.

How it works:

- Pulls the latest code from GitHub
- Builds the project and sets up the environment
- Executes Selenium tests with TestNG
- Generates and shares reports after every build

Jenkins Report

The screenshot shows the Jenkins interface for a build labeled '#5'. The left sidebar includes links for Status, Changes, Console Output, Edit Build Information, Delete build '#5', Timings, Git Build Data, and TestNG Results, with 'TestNG Results' currently selected. The main content area is titled 'TestNG Results' and displays a summary: '106 failures(+106), 11 skipped(+11)' with a red progress bar indicating failure, and '426 tests(+426)' in total. Below this is a section titled 'Failed Tests' with a table listing 106 failed test methods. The table has columns for 'Test Method' (listing methods like 'Tests.ContactUsPage.verifyFileAttachment' and 'Tests.ContactUsPage.verifyScrollUpFunctionality') and 'Duration' (showing times like '00:00:02.384' and '00:00:02.547').

Test Method	Duration
>>> Tests.ContactUsPage.verifyFileAttachment	00:00:02.384
>>> Tests.ContactUsPage.verifyScrollUpFunctionality	00:00:02.547
>>> Tests.UI_Home_2.verifyChatSupportArrowButton	00:00:02.940
>>> Tests.UI_Home_2.verifyRecommendationsNextButton	00:00:06.210
>>> Tests.ALLCART_UI.verifyBreadcrumbNavigationOnCartPage	00:00:02.542
>>> Tests.ALLCART_UI.verifyCartEmptyMessageAndRedirectLink	00:00:02.343
>>> Tests.ALLCART_UI.verifySubscriptionInputFieldAndButtonVisibility	00:00:12.956
>>> Tests.ALLCART_UI.verifySubscriptionPlaceholderText	00:00:02.941
>>> Tests.ContactUs_UI_2.verifyChooseFileFieldsAcceptingInput	00:00:02.397
>>> Tests.ContactUs_UI_2.verifyContentsUnderFeedbackForUsTitle	00:00:02.485
>>> Tests.ContactUs_UI_2.verifyFeedbackForUsFontSizeCorrect	00:00:02.393
>>> Tests.ContactUs_UI_2.verifyFeedbackForUsFontStylesCorrect	00:00:03.014
>>> Tests.ContactUs_UI_2.verifyFeedbackForUsTitlesPresent	00:00:12.820
>>> Tests.ContactUsPage_UI_1.verifyNoteSectionDisplayOnContactUsPage	00:00:02.340
>>> Tests.TestCases_UI_1.verifyTechCaseTitleOnPage	00:00:02.389

- Automates test execution and generates detailed reports after each build
- Provides insights into test results, environment details, and execution timelines
- Supports continuous testing by triggering builds on code changes

CONCLUSION

- Automation drastically reduces manual effort while increasing test efficiency and reliability.
- The project validated that Selenium and TestNG are robust tools for web testing.
- Automated reports enhance transparency and support faster debugging.
- Automation enables Continuous Testing and supports Agile development.
- Future work: Expand test coverage, integrate with CI/CD pipelines, and explore performance and security testing.



Thank You



Presented by - Team A2

Venkata kalyan

K.S.Chakradhar

Balaji

Syed suhail