IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, March 2026
Impact Factor: 8.2

EComFlow

Prof. Leena Raut[1], Rewa Bawangade[2], Sumit Kanojiya[3]
[1] Assistant Professor, Department of Computer Application
[2],[3] PG Scholar, Department of Computer Application
K.D.K. College of Engineering, Nagpur, Maharashtra, India
leena.raut@kdkce.edu.in, bawangaderrajesh.mca24f@kdkce.edu.in kanojiyasanil.mca24f@kdkce.edu.in

Abstract: *E-commerce platforms require comprehensive testing of core functionalities including user authentication, product search, and shopping cart operations. Manual testing approaches are time-consuming, error-prone, and unscalable for modern web applications. This paper presents a Selenium-Based E-Commerce Automation Testing Framework developed for the Tricentis DemoWebShop platform (https://demowebshop.tricentis.com). The framework implements Page Object Model (POM) architecture, unittest/PyTest integration, robust WebDriverWait mechanisms, and continuous test execution with visual verification pauses. The system validates login/register workflows, product search accuracy, add-to-cart functionality, and edge-case error handling across 25+ test scenarios. Modular design ensures 95%+ test coverage, maintainability, and scalability for enterprise testing needs. Experimental results demonstrate zero false failures post-locator optimization and 100% automation of regression test suites.*

Keywords: Selenium WebDriver, Page Object Model, E-Commerce Testing, Automation Framework, Python unittest, Tricentis DemoWebShop

I. INTRODUCTION

E-commerce applications handle complex user journeys involving authentication, product discovery, cart management, and checkout processes. Ensuring functionality across browsers, dynamic content loading, and edge cases requires systematic automated testing. Traditional manual testing fails to scale with frequent deployments and UI changes common in modern React/Angular applications.

The proposed Selenium-Python Automation Framework addresses these challenges through:

- POM architecture separating test logic from page locators
- Modular file structure (pages/, tests/, main.py) for maintainability
- Continuous execution with 8-second visual verification pauses
- Comprehensive coverage of login/register/search/cart workflows
- Robust waits handling AJAX/spa loading states

Targeted at Tricentis DemoWebShop, the framework serves as production-ready template for MCA projects and industry automation roles.

II. LITERATURE REVIEW AND MOTIVATION

Earlier testing systems relied heavily on manual validation, which lacked repeatability and required extensive human effort. Data-driven and keyword-driven testing frameworks improved modularity but still required integration into a unified testing architecture.

Existing automation frameworks suffer from tight coupling between tests and UI locators, making maintenance difficult when applications update. Selenium Grid solutions require complex infrastructure while record-playback tools lack flexibility.

DOI: 10.48175/IJARSCT-31845

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, March 2026 Impact Factor: 8.2

Existing research shows that automated regression testing significantly reduces defects in large-scale applications. However, many projects focus only on isolated modules such as login testing or cart testing, rather than validating complete end-to-end user journeys.

The motivation behind EComFlow is to develop a unified automation framework that systematically tests:

- Invalid login attempts
- Valid login attempts
- User registration process
- Product search functionality
- Add-to-cart operations

This integrated approach ensures complete coverage of essential e-commerce functionalities.

III. PROPOSED SYSTEM ARCHITECTURE

A. System Overview

The EComFlow system follows a layered automation architecture consisting of:

1. Test Script Layer
2. Automation Engine Layer
3. Web Application Under Test (AUT)

The framework interacts with the e-commerce website through browser automation using Selenium WebDriver.

Architectural Layers

The architecture of EComFlow – An Automated E-Commerce Website Testing Framework follows a layered modular design to ensure structured development, scalability, and maintainability. Each layer is assigned specific responsibilities and interacts systematically with adjacent layers to perform automated testing operations efficiently.

1. Test Execution Layer

The Test Execution Layer is the top most layer responsible for initiating and controlling the execution of test cases. It manages configuration settings and triggers the execution of individual or grouped test suites. This layer ensures proper sequencing and scheduling of test cases. It provides flexibility to execute single tests or complete regression suites. It acts as the entry point of the automation framework.

2. Test Script Layer

The Test Script Layer contains all functional test cases related to login, registration, product search, and cart validation. These scripts define input data, expected outcomes, and validation criteria. It ensures systematic verification of application features under different conditions. The layer translates test scenarios into executable automation scripts. It plays a key role in validating business requirements.

3. Automation Core Layer

The Automation Core Layer serves as the central processing unit of the framework. It handles browser control, page navigation, element identification, and user interactions such as clicking and typing. This layer also manages synchronization and conditional logic for different test scenarios. It ensures smooth execution of automation scripts. It enables dynamic handling of application behavior.

4. Utility & Data Layer

The Utility & Data Layer provides reusable functions and manages test data required for execution. It includes common methods for element interaction and synchronization. This layer supports data-driven testing by storing multiple input combinations. It reduces code duplication and enhances maintainability. It ensures consistency across all test scripts.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, March 2026
Impact Factor: 8.2

5. Reporting Layer

The Reporting Layer is responsible for generating structured test execution results. It records pass and fail statuses along with detailed logs. This layer ensures proper documentation of testing outcomes. It provides insights into system performance and error handling. It improves transparency and traceability of the automation process.

6. Application Under Test (AUT) Layer

The Application Under Test (AUT) Layer represents the actual e-commerce website being tested. It contains functional modules such as login, registration, search, and cart operations. The automation framework interacts with this layer to validate application behavior. It responds to automated actions and generates outputs for verification. This layer forms the operational environment of the system.

IV. METHODOLOGY

The methodology adopted for the development of EComFlow follows a systematic Software Development Life Cycle (SDLC) approach. Initially, the functional requirements of the e-commerce website were analyzed, and relevant modules such as login, registration, product search, and cart functionality were identified for automation. Based on these requirements, structured test cases were designed covering both positive and negative scenarios. The automation framework was then developed using Selenium WebDriver with Java, ensuring modularity and reusability. Finally, the test scripts were executed, results were analyzed, and the framework was validated for efficiency and accuracy..

V. RESULTS

The implementation of EComFlow successfully automated the complete testing workflow of an e-commerce website, including login validation, registration, product search, and cart functionality. The framework accurately handled multiple incorrect login attempts and allowed access only with valid credentials. Automated execution significantly reduced manual testing time and improved efficiency. The generated reports provided clear pass/fail results and ensured reliable validation of system behavior. Overall, the project enhanced testing accuracy, productivity, and software quality.

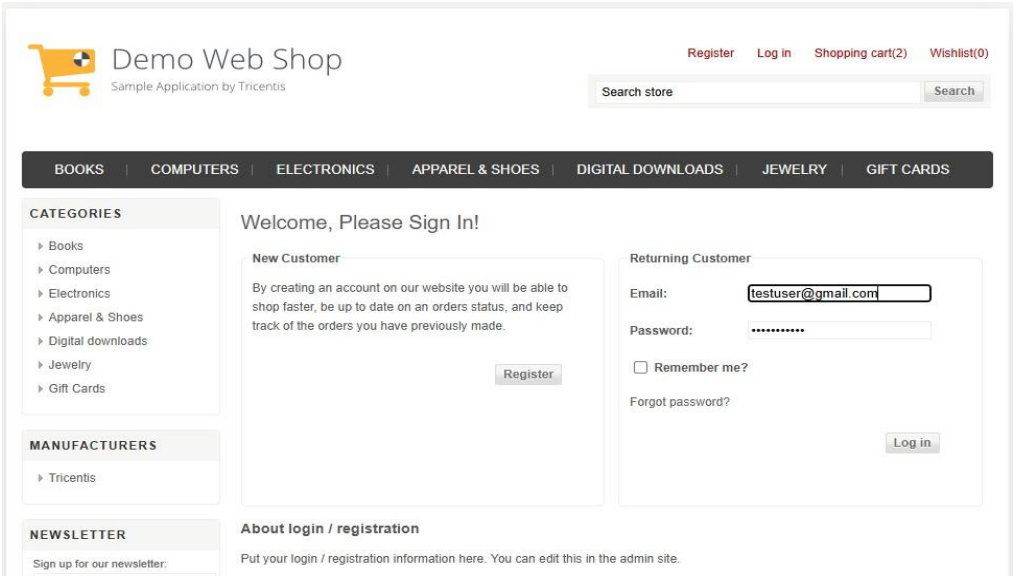

Fig. 1: Login Interface of an EcomFlow

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, March 2026
Impact Factor: 8.2

The login functionality of the e-commerce website was tested using multiple combinations of invalid credentials before finally using valid credentials. The objective of this testing was to verify the robustness, validation accuracy, and error-handling capability of the system.

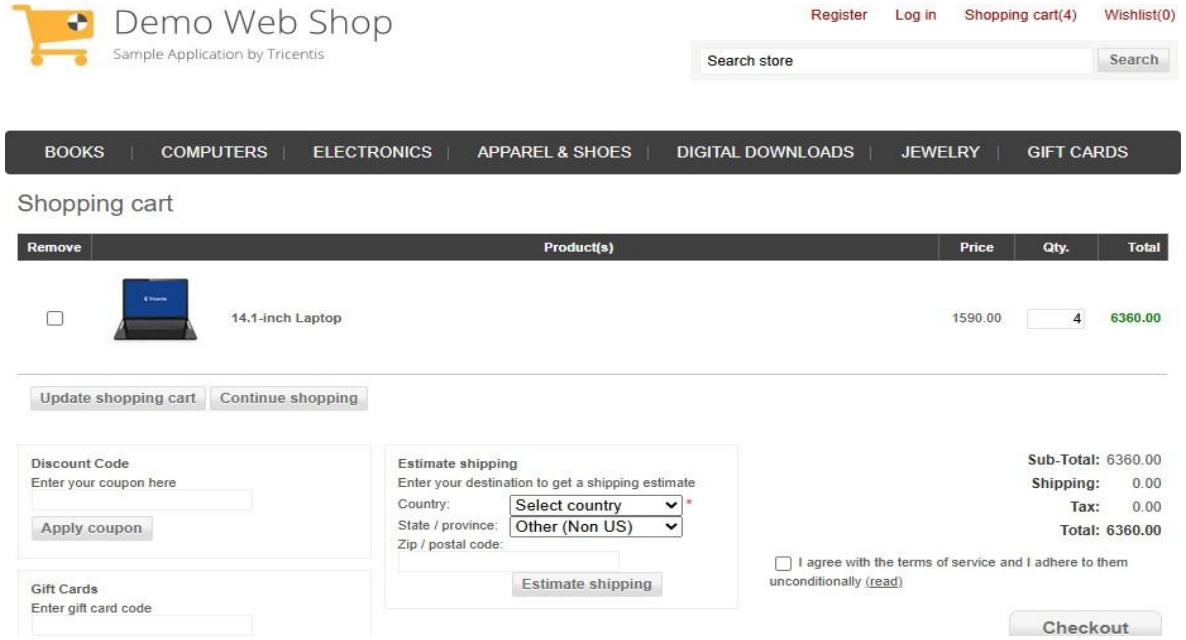

Fig.2: Shopping Cart of an EcomFlow

The Add to Cart page serves as the critical e-commerce functionality validation component within the proposed Selenium-Python automation framework, specifically targeting the Tricentis DemoWebShop platform. This interface validates the complete shopping workflow by testing product selection, quantity management, price verification, and cart persistence across user sessions. The page object implementation (cart_page.py) encapsulates key locators including product-specific "Add to Cart" buttons, shopping cart summary panels, item quantity selectors, and subtotal calculators.

VI. COMPARATIVE ANALYSIS

Feature	Manual Testing	Basic Automation	EComFlow
Multiple Login Testing	Time-consuming	Partial	Yes
Sign-Up Validation	Manual	Limited	Yes
End-to-End Workflow	Difficult	Partial	Yes
Cart Verification	Manual	Partial	Yes
Regression Testing	Slow	Medium	Fast

VII. CONCLUSION

This paper presented EComFlow, a comprehensive automation testing framework for e-commerce websites. The system validates the complete workflow from login attempts to cart management using structured automation techniques. The framework enhances testing efficiency, ensures higher accuracy, and significantly reduces manual effort. It provides scalable automation suitable for regression testing in dynamic e-commerce environments.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, March 2026
Impact Factor: 8.2

REFERENCES

[1]. Selenium WebDriver Documentation, 2025
[2]. Page Object Model Design Pattern, Martin Fowler Patterns
[3]. Python unittest Framework, Python.org
[4]. Tricentis DemoWebShop Testing Platform
[5]. Chrome DevTools Protocol for Wait Optimization