Test Automation Framework for the Web Application

- **Test Automation tool - Selenium WebDriver**

Why Selenium Webdriver?

- Selenium is used for automating testing across various web browsers such as Chrome, Mozilla Firefox, Safari and Internet Explorer.
- The selenium automation framework is very easy to use as it provides a user-friendly interface and the availability of open source code is one of the benefits of Selenium.
- Selenium supports all major programming languages, like Java, Python, JavaScript, C#.

- **Programming Language: Java**

Why Java?

- One of the major reasons for using Java with Selenium is that it has vast open-source libraries that can automate different testing processes, including unit and browser testing.
- Selenium with Java offers good IDE support like IntelliJ IDEA and Eclipse for easy test scripting and maintenance

- **Test Framework: TestNG (Java)**

Why TestNG?

- Its ability to write powerful test cases with the help of annotations, grouping, and parametrizing and it covers all classifications of test automation like Unit testing, Functional testing, End-to-End, and integration testing.

- **Implement the Page Object Model (POM)**

  Page Objects: Create classes for each significant page or component (e.g., LoginPage, ProductPage, CartPage, CheckoutPage). Each class should include the elements of the page and the actions that can be performed on them.

  Actions: Implement methods in each page object class to interact with the page elements (e.g., enterUsername, clickLoginButton, searchProduct).

Why POM as design pattern?

As it helps reduce code duplication and improves test case maintenance and could achieve code reusability by writing the code once and use it in different tests

- **Design Test Cases**

1. **Functional Test Cases:**
    - User Login/Registration: Test cases for successful and unsuccessful logins/registrations, password resets, and validations.
    - Product Search: Validate search functionalities with different categories.
    - Add to Cart: Ensure products can be added/removed from the cart and that quantities and prices are correctly updated.
    - Checkout Process: Verify the entire checkout workflow, including payment methods, address validation, and order summary.
    - Order History: Test order history retrieval, order details, and status updates.

2. **Non-Functional Test Cases:**

    - Performance Testing: Test page load times , response times
    - Security Testing: Verify security aspects such as authentication and authorization.