

# Adv. Selenium - Web Driver Advance

## Automation Testing Assessment (Java + Selenium)

- **Case Study: “ShopEZ” – Online Retail Website**  
(<https://www.shopezusa.com/>)
- **Login functionality.**
- **Navigate to <https://shop-ez.com>**
- **Enter username and password**
- **Click on “Login”**
- **Assert that the user is redirected to the home page**

```
public class LoginTest {
    WebDriver driver;
    ExtentReports extent;
    ExtentTest test;

    @BeforeClass
    public void setUp() {
        driver = WebDriverFactory.createDriver("chrome");
        extent = ExtentManager.getReporter();
        test = extent.createTest("ShopEZ Login Test");
        driver.manage().window().maximize();
    }

    @Test
    public void testValidLogin() {
        test.info("Navigating to ShopEZ login page");
        LoginPage loginPage = new LoginPage(driver);
        loginPage.goTo("https://www.shopezusa.com/");

        test.info("Entering credentials");
        loginPage.enterUsername("testuser");
        loginPage.enterPassword("testpass");
        test.info("Clicking Login");
        loginPage.clickLogin();
    }
}
```

```

        test.info("Verifying user is redirected to home page");
        Assert.assertTrue(loginPage.isAtHomePage(), "Home page not displayed after login");
        test.pass("Login successful – Home page displayed");
    }

    @AfterClass
    public void tearDown() {
        extent.flush();
        driver.quit();
    }
}

```

## 2) Automate the product search and add-to-cart workflow.

- **Search for a product (e.g., “Laptop”)**
- **Click on a product from the search results**
- **Add it to the cart**
- **Assert that the cart count increases**

```

package com.shopez.tests;

import com.shopez.pages.*;
import com.shopez.reports.ExtentManager;
import com.shopez.utils.WebDriverFactory;
import com.aventstack.extentreports.*;
import org.testng.Assert;
import org.testng.annotations.*;

public class ProductSearchAndCartTest {

    WebDriver driver;
    ExtentReports extent;
    ExtentTest test;

    @BeforeClass
    public void setUp() {
        driver = WebDriverFactory.createDriver("chrome");
        extent = ExtentManager.getReporter();
        test = extent.createTest("ShopEZ Product Search & Add-to-Cart Test");
    }
}

```

```

        driver.manage().window().maximize();
    }

    @Test
    public void testSearchAndAddToCart() {
        test.info("Navigating to ShopEZ home page");
        HomePage homePage = new HomePage(driver);
        homePage.goTo("https://www.shopezusa.com/");

        int initialCount = homePage.getCartCount();
        test.info("Initial cart count: " + initialCount);

        test.info("Searching for product 'Laptop'");
        homePage.searchProduct("Laptop");

        SearchResultsPage resultsPage = new SearchResultsPage(driver);
        test.info("Clicking on the first product in search results");
        resultsPage.clickFirstProduct();

        ProductPage productPage = new ProductPage(driver);
        test.info("Adding product to cart");
        productPage.addToCart();

        int updatedCount = homePage.getCartCount();
        test.info("Updated cart count: " + updatedCount);

        Assert.assertTrue(updatedCount > initialCount, "Cart count did not increase after adding product");
        test.pass("Product successfully added to cart. Cart count increased.");
    }

    @AfterClass
    public void tearDown() {
        extent.flush();
        driver.quit();
    }
}

```

### 3) Describe or design the folder structure of a basic Selenium framework using

- Maven

- **TestNG**

- **Utility classes (e.g., Waits)**

### 1. Maven (pom.xml)

- Manages dependencies:
    - Selenium Java
    - TestNG
    - WebDriverManager (for driver binaries)
    - ExtentReports (for reporting)
  - Manages plugins:
    - Surefire plugin (to run TestNG tests).
- 

### 2. TestNG (testng.xml)

Defines:

- Which tests to run.
- Parallel execution.
- Groups (Smoke, Regression, etc.).

Example:

```
<suite name="ShopEZ Automation Suite">
  <test name="E2E Tests">
    <classes>
      <class name="com.shopez.tests.LoginTest"/>
      <class name="com.shopez.tests.ProductSearchAndCartTest"/>
    </classes>
  </test>
</suite>
```

### 3. Utilities

- **WebDriverFactory.java** → Creates driver instances (Chrome, Firefox).
- **WaitUtils.java** → Contains explicit waits (waitForVisibility, waitForClick).
- **ConfigReader.java** → Loads config.properties for URL, credentials.

- **TestDataUtils.java** → Reads external test data (Excel/CSV).

#### **4) Integrate your login test with a simple reporting tool (Extent Reports).**

- **Initializing report**
- **Create Framework**
- **Logging test status & Generating report after execution**

```
package com.shopez.reports;
```

```
import com.aventstack.extentreports.ExtentReports;
```

```
import com.aventstack.extentreports.reporter.ExtentHtmlReporter;
```

```
public class ExtentManager {
```

```
    private static ExtentReports extent;
```

```
    public static ExtentReports getReporter() {
```

```
        if (extent == null) {
```

```
            ExtentHtmlReporter html = new ExtentHtmlReporter("test-  
output/ShopEZ_Login_Report.html");
```

```
            html.config().setDocumentTitle("ShopEZ Automation Report");
```

```
            html.config().setReportName("Login Test Report");
```

```
            extent = new ExtentReports();
```

```
            extent.attachReporter(html);
```

```
            extent.setSystemInfo("Project", "ShopEZ");
```

```
            extent.setSystemInfo("Tester", "QA Analyst");
```

```
        }
```

```
        return extent;
```

```
    }
```

```
}
```