

IS4102 – Advanced Software Quality Assurance

Assignment II

TestNG – Selenium

Instructions: You need to develop a software test automation framework using Selenium WebDriver and TestNG, following the Page Object Model (POM) design pattern.

Scope of the assignment:

1. Framework Structure:

The automation framework should be modular and well-organized. Key components should include:

Page Objects: Represent web pages, encapsulating element locators and actions.

Test Scripts: Test cases written using TestNG that interact with page objects.

Utility Classes: For common functionality like browser setup, logging, and reporting.

Test Data: Test data should be externalized (e.g., using properties or Excel files).

Configuration Files: For storing environment settings (e.g., browser type, base URL).

2. Page Object Model (POM) Implementation:

The core of the assignment is to implement POM for separating page-related functionality from test scripts.

Page Classes: Each class should correspond to a specific webpage or component and should contain locators and methods to interact with elements.

Action Methods: Define user actions such as clicks, form inputs, and validations within page classes.

Abstraction: Ensure that the test logic does not include direct WebDriver calls, which should only be in the page classes.

3. TestNG Features to Implement:

- **Annotations:** Utilize `@BeforeSuite`, `@BeforeTest`, `@BeforeClass`, `@BeforeMethod`, and their counterparts for proper test setup and teardown.
- **Parallel Execution:** Optionally configure parallel test execution to demonstrate scalable test automation.
- **Assertions:** Include assertions to validate test outcomes.
- **Reporting:** Generate comprehensive reports using TestNG's built-in or third-party tools (e.g., Allure Reports).

4. Cross-Browser Testing:

- Implement tests that can run across different browsers (e.g., Chrome, Firefox, Edge) using WebDriver.

5. Test Data Management:

- Use a test data file (e.g., in Excel) for managing input data.
 - Implement data-driven testing using TestNG's `@DataProvider` to validate multiple scenarios.
-

Requirements:

1. Programming Language:

- Students must use **Java** as the programming language.

2. Project Setup:

- Use **Maven** as a build tool for dependency management.
- Include Selenium and TestNG as dependencies.
- Optionally integrate libraries like Apache POI for handling Excel files and ExtentReports for enhanced reporting.

3. Documentation:

- Provide clear documentation explaining the project structure, setup instructions, and how to execute the tests.
- The framework should have well-documented code, especially for page classes and utility functions.

4. Website Choice:

Students can choose any website to implement the framework (e.g., e-commerce site, news portal, or educational site). The website must contain dynamic elements such as forms, buttons, and validation messages.

5. Bonus Points:

- Integration with **CI/CD** tools like Jenkins for automated test execution.
 - Implementation of **screenshot capturing** on test failures.
 - Use of **headless browser** modes for faster execution.
-

Deliverables:

1. **Source Code:**
 - A complete, working test automation framework hosted on GitHub or any other version control platform.
 2. **Test Report:**
 - Test execution report showcasing pass/fail results for each test case.
 3. **Presentation:**
 - A short presentation explaining the framework design, choice of tools, and demonstration of the test execution.
-

Grading Criteria:

1. **Design and Structure (25%)**
 - Proper application of the Page Object Model.
 2. **Test Coverage (30%)**
 - Extent of test coverage and the number of test cases implemented.
 3. **Reporting and Logging (20%)**
 - Quality of test reports, logging, and error-handling mechanisms.
 4. **Documentation and Usability (15%)**
 - Clarity of the documentation and ease of use of the framework.
 5. **Bonus Points (10%)**
 - Advanced features such as cross-browser testing or additional utilities.
-

Submission Deadline:

- Deadline will be published on the VLE soon.

Additional Notes:

- Ensure to run all tests before submission and provide proof of test results in the form of screenshots or logs.
- Any plagiarism in code or documentation will result in zero marks.