Automation Testing using Selenium and NUnit

Prerequisites

- Basic understanding of programming (preferably in C# or Java)
- Familiarity with HTML and DOM structure
- Experience in manual testing (optional, but beneficial)
- Understanding of basic software testing concepts (unit testing, functional testing, etc.)

1. Introduction to Software Testing and Selenium Basics

Lecture Topics

- Overview of Software Testing
    - Types of testing (manual, automation)
    - Importance of automation testing
- Introduction to Selenium
    - What is Selenium?
    - Selenium WebDriver Architecture
    - Selenium vs Other Automation Tools
- Setting up the environment
    - Installation of JDK, Eclipse/IntelliJ, Visual Studio (for C#)
    - Installing Selenium WebDriver
    - Installing necessary browser drivers (ChromeDriver, GeckoDriver)

Lab Details

- **Lab 1.1:** Setting up Selenium with Visual Studio/Eclipse
    - Install Selenium WebDriver
    - Configure browser drivers
    - Write a basic Selenium test script to open a browser and navigate to a webpage
- **Lab 1.2:** Running and troubleshooting Selenium scripts
    - Fix basic issues like path errors or missing drivers

---

2. Selenium WebDriver in Depth

Lecture Topics

- WebDriver API Overview
    - Interacting with elements (click, sendKeys, etc.)
    - Element locators (ID, Name, XPath, CSS Selector)
    - Handling dynamic elements
- Working with different browsers (Chrome, Firefox, etc.)
- Synchronization techniques
    - Implicit Wait vs Explicit Wait
    - Handling pop-ups, alerts, and frames

Lab Details

- **Lab 2.1:** Writing Selenium tests using different locators
    - Create test cases using ID, Name, CSS, and XPath selectors
- **Lab 2.2:** Implementing waits in Selenium
    - Use implicit and explicit waits in test cases for handling page load times
- **Lab 2.3:** Cross-browser testing
    - Run tests across Chrome, Firefox, and other browsers

---

3. Advanced Selenium Features

Lecture Topics

- Handling Advanced Scenarios
    - File uploads and downloads
    - Handling JavaScript pop-ups
    - Capturing screenshots
    - Managing browser cookies and sessions
- Working with data-driven testing
    - Reading data from external files (Excel, CSV)
    - Parameterization using TestNG/NUnit

Lab Details

- **Lab 3.1:** Working with file uploads and pop-ups

    - Write a script to handle file upload dialogs and pop-up alerts
- **Lab 3.2:** Capture screenshots for failed test cases
    - Implement a function to take screenshots on failure
- **Lab 3.3:** Data-driven testing with external files
    - Create a test case that reads data from an Excel sheet and performs form submission

---

4. Introduction to NUnit for Software Testing

Lecture Topics

- Overview of Unit Testing with NUnit
    - What is NUnit?
    - NUnit vs other testing frameworks (MSTest, xUnit)
    - Setting up NUnit in Visual Studio
- Writing and Running Tests with NUnit
    - NUnit attributes ([Test], [SetUp], [TearDown], etc.)
    - Assertions in NUnit
- Structuring Unit Test Projects
    - Best practices for organizing tests

Lab Details

- **Lab 4.1:** Setting up NUnit for Unit Testing
    - Install NUnit and create a simple test project in Visual Studio
- **Lab 4.2:** Writing and executing basic NUnit tests
    - Create and run test cases using NUnit attributes ([Test], [SetUp], etc.)
- **Lab 4.3:** Assertions and test validation
    - Write test cases that use different types of assertions for validation (e.g., Assert.AreEqual, Assert.IsTrue)

---

5. Advanced NUnit and Integration with Selenium

Lecture Topics

- Parameterized and Data-Driven Testing in NUnit

    - How to pass parameters to NUnit test cases
    - Using NUnit's [TestCase], [TestCaseSource] attributes for data-driven tests
- Test Fixtures and Test Suites
    - Organizing tests into suites
    - Running tests in parallel using NUnit
- Integrating NUnit with Selenium
    - Using NUnit to structure Selenium test cases
    - Running Selenium test suites with NUnit

Lab Details

- **Lab 5.1:** Data-driven testing with NUnit
    - Implement NUnit test cases that accept parameters and run multiple scenarios
- **Lab 5.2:** Creating and executing Selenium test suites with NUnit
    - Organize multiple Selenium test cases into a test suite and run using NUnit
- **Lab 5.3:** Parallel test execution
    - Set up NUnit to run Selenium tests in parallel on multiple browsers