Software testing and Automation

Course Learning Objectives	● Introduction to Software Testing Principles: Covering product design and web/mobile UX aspects. ● Differentiate between Black-Box and White-Box Testing methods, comprehending their applications in software testing. ● Create effective test cases, emphasize bug reporting, and utilize metrics and statistics for evaluation. ● Utilize model-driven test design, define test procedures, and organize test cases effectively. ● Conduct various types of advanced testing, including performance, usability, security, and compatibility testing. ● Apply testing methodologies specific to the agile environment and grasp testing techniques for web and mobile applications. ● Understand automated software testing, particularly focusing on Selenium WebDriver for web application testing. ● Implement test automation tools effectively, comprehend their functionality, and generate comprehensive test reports.
Course Outcomes	On successful completion of the course, students will be able to: ● Design and Identify the fundamental concepts of software testing and the prerequisites for software testing. ● Create test planning plans and the various processes involved in testing. ● Design test cases that are effective in identifying critical defects in the application ● Execute advanced testing methods. ● Test software through automation with Selenium and Testing.

Course Duration: 45 Hours

UNIT I FOUNDATIONS OF SOFTWARE TESTING

Why do we test Software? Black-Box Testing and White-Box Testing, Software Testing Life Cycle, V- model of Software Testing, Program Correctness and Verification, Reliability versus Safety, Failures, Errors and Faults (Defects), Software Testing Principles, Program Inspections, Stages of Testing: Unit Testing, Integration Testing, System Testing

UNIT II TEST PLANNING

The Goal of Test Planning, High-Level Expectations, Intergroup Responsibilities, Test Phases, Test Strategy, Resource Requirements, Tester Assignments, Test Schedule, Test Cases, Bug Reporting, Metrics and Statistics

UNIT III TEST DESIGN AND EXECUTION

Test Objective Identification, Test Design Factors, Requirement identification, Testable Requirements, Modeling a Test Design Process, Modeling Test Results, Boundary Value Testing, Equivalence Class Testing, Path Testing, Data Flow Testing, Test Design Preparedness Metrics, Test Case Design Effectiveness, Model-Driven Test Design, Test Procedures, Test Case Organization and Tracking, Bug Reporting, Bug Life Cycle.

UNIT IV ADVANCED TESTING CONCEPTS

Performance Testing: Load Testing, Stress Testing, Volume Testing, Fail-Over Testing, Recovery Testing, Configuration Testing, Compatibility Testing, Usability Testing, Testing the Documentation, Security testing, Testing in the Agile Environment, Testing Web and Mobile Applications.

UNIT V TEST AUTOMATION AND TOOLS

Automated Software Testing, Automate Testing of Web Applications, Selenium: Introducing Web Driver and Web Elements, Locating Web Elements, Actions on Web Elements, Different Web Drivers, Understanding Web Driver Events, Testing:

Understanding Testing.xml, Adding Classes, Packages, Methods to Test, Test Reports.

Test Projects:

Use Cases

1. Develop the test plan for testing an e-commerce web/mobile application (www.amazon.in).

Task 1: The scope of testing specifying test scenarios

Task 2: Identifying testing objectives,

Task 3: Outlining test cases

Task 4: Detailing the testing environment

2. Design the test cases for testing the e-commerce application

Task 1: Validate the application functionality

Task 2: Validate the application usability Task

3: Validate the application security, Task 4:

Validate the application performance

3. Test the e-commerce application and report the defects in it.

Task 1: Verify that users can successfully register for a new account

Task 2: Confirm that users can search for products successfully Task

3: Confirm that users can view their order history.

Task 4: Confirm that payments are processed successfully.

Task 5: Validate the functionality of the checkout process

4. Develop the test plan and design the test cases for an inventory control system.

Task 1: Inventory Control System including inventory management

Task 2: Inventory Control System including order processing

Task 3: Inventory Control System including user authentication

Task 4: Inventory Control System including reporting

5. Execute the test cases against a client-server or desktop application and identify the defects.

Task 1: Develop comprehensive test cases based on the requirements

Task 2: Prepare the test environment

Task 3: Run the test cases systematically

Task 4: Document the results of each test cases

Task 5: Identify any defects during the testing process, log them in a defect tracking system

6. Test the performance of the e-commerce application.

Task 1: Identify performance metrics

Task 2: Select performance testing tools based on your application and testing requirement

Task 3: Develop realistic test scenarios

Task 4: Set up a test environment

Task 5: Run the performance tests

7. Automate the testing of e-commerce applications using Selenium.

Task 1: Install Selenium

Task 2: Download a WebDriver compatible with your browser

Task 3: Use appropriate locators to locate web elements Task

4: Testing framework

8. Integrate TestNG with the above test automation.

Task 1: Install TestNG using your preferred build tool or directly in your IDE

Task 2: Update the Selenium script

Task 3: Create a TestNG XML file to define the test suite and test classes

Task 4: Run the TestNG suite using your preferred method, either through your IDE or command line

9. Mini Project:

a) Build a data-driven framework using Selenium and TestNG

b) Build Page object Model using Selenium and TestNG

c) Build BDD framework with Selenium, TestNG, and Cucumber