Playwright Automation Testing

Track -1 Manual Testers
Duration – 10 days / 4 Hrs each day

Module 1: JavaScript, TypeScript, and Playwright Setup

Day 1 Topics:

- Introduction to JavaScript for testers
  - Variables, arrays, objects, and loops
  - Conditional statements (if, switch)
  - Functions: declaration, parameters, return types
- Introduction to async/await and Promises

Day 2 Topics:

- Introduction to TypeScript
  - Static typing, enums, interfaces, type inference
  - TypeScript vs JavaScript: key differences
- Introduction to Node.js & npm
  - Installing Node.js
  - npm init, package.json
  - Installing dependencies
- Installing and configuring Playwright CLI
  - Setting up VS Code for Playwright development

Labs:

- Write JS/TS functions using variables, arrays, loops
- Set up a TypeScript-based Playwright test project
- Install required dependencies and run the first test

Module 2: Writing Tests and Element Interactions

Day 3 Topics:

- Writing Basic Tests in Playwright
  - Launching browser
  - Navigating to URL
  - Locating elements using:
    - Text selectors
    - Role selectors
    - CSS/XPath selectors
  - Performing user actions:
    - Click
    - Fill input fields
    - Press keyboard keys
    - Select dropdown options

Day 4 Topics:

- Writing basic assertions
    - toBeVisible, toHaveText, toBeEnabled
- Handling dynamic elements (waits, retries)
- Working with frames and iframes

Labs:

- Write tests for login or signup page
- Automate clicking buttons and filling forms
- Validate element visibility and state

Module 3: Page Object Model, Screenshots, and Test Hooks

Day 5 Topics:

- Implementing Page Object Model (POM)
    - Structuring project with POM pattern
    - Creating reusable classes for pages
- Managing selectors using locators file
- Capturing screenshots on test failure

Day 6 Topics:

- Recording videos of test runs
- Using Hooks and Fixtures
    - beforeEach, afterEach
    - Setup and teardown for tests
- Test data management
    - Using JSON or static data files

Labs:

- Create POM for a sample application (e.g., login page)
- Implement tests using POM structure
- Add screenshot and video capture logic
- Setup hooks for repeated setup/cleanup

Module 4: Test Flows, Debugging, and Authentication

Day 7 Topics:

- Test Organization and Grouping
    - Organizing tests by feature
    - Using test tags, filenames, and folders
- Handling Forms, Popups, Alerts
    - Interacting with modals, alerts, confirm boxes
    - Uploading and downloading files

Day 8 Topics:

- Debugging Techniques
    - Console logs and browser logs
    - Debug mode and trace viewer
- Authentication Scenarios
    - Login flows with and without UI
    - Token/cookie-based authentication
    - Preserving session state

Labs:

- Automate a complex form with validations
- Use console and trace viewer to debug test
- Implement authentication script with session storage

Module 5: Cross-Browser, Parallel Execution, Reporting, CI/CD

Day 9 Topics:

- **CI/CD & Headless Execution**
    - GitHub Actions / Jenkins / Azure pipelines setup
    - Containerized test execution (Dockerfile, Playwright Docker image)
    - Run Playwright in Kubernetes or cloud agents
- **Advanced Reporting & Automation Governance**
    - Allure Testops report with Jira, HTML report, CI test artifacts
    - Creating dashboards using custom JSON output
    - Code reviews, linting, pull request automation
    - Branch strategy and test coverage integration

Day 10 Topics:

- CI/CD Integration Concepts
    - Overview of GitHub Actions, GitLab CI, Jenkins
    - Running Playwright tests in CI pipeline
- Reporting using Allure TestOps with Jira
- Automation Governance
    - Folder structure, code reviews
    - Naming conventions and documentation

Labs:

- Execute tests in different browsers
- Run tests in parallel using CLI config
- Generate and interpret HTML and trace reports
- Simulate tests on mobile viewport
- Review CI logs and test execution results (in sandbox/demo)