Module 1: Introduction to Playwright

- What is Playwright?
- Playwright vs Cypress vs Selenium
- Supported browsers & platforms
- Installation and setup

Module 2: Setting Up Your Project

- Installing Playwright with Node.js
- Initializing your test project
- Understanding the folder structure
- Installing Playwright Test Runner

Module 3: Writing Your First Test

- Creating your first test file
- Using test(), expect(), and page object
- Running tests using the CLI
- Viewing reports

Module 4: Core Playwright Features

- page.goto(), page.click(), page.fill(), etc.
- Element selectors
- Assertions (toBeVisible, toHaveText, etc.)
- Screenshots and video recordings

Module 5: Browser and Contexts

- Launching different browsers (Chromium, Firefox, WebKit)
- Working with browser contexts (like incognito mode)
- Emulating devices and geolocation

Module 6: Advanced Interactions

- Dropdowns, checkboxes, radio buttons
- File uploads/downloads
- Handling popups, alerts, and dialogs
- Handling iframes

Module 7: Network Control and API Testing

- Intercepting network requests
- Mocking and modifying API responses
- Using page.route() and page.on('request')
- API testing with request.newContext()

Module 8: Test Hooks and Fixtures

- beforeAll, beforeEach, afterEach, afterAll
- Custom test fixtures
- Sharing setup and teardown

Module 9: Parallel and Cross-Browser Testing

- Running tests in parallel
- Configuring multiple projects in playwright.config.ts
- Running tests across Chrome, Firefox, and Safari

Module 10: Handling Authentication

- UI-based login
- Saving and reusing login sessions
- API-based login for faster tests

Module 11: Page Object Model (POM)

- Why use POM
- Structuring your code using classes
- Reusability and maintainability

Module 12: Playwright Test Reports and Debugging

- HTML and JSON reports
- Trace Viewer
- Using debug mode for slow test inspection

Module 13: CI/CD Integration

- Running Playwright in GitHub Actions / GitLab / Jenkins
- Headless testing in CI
- Storing reports and artifacts

Bonus: Real-World Project

- Automate a real eCommerce or login application
- Write tests for full user flows
- Use mocks and authentication