Playwright

Day 1: Basic Web Development Knowledge

• HTML/CSS:

- o Introduction to HTML: Elements, attributes, and structure.
- o Introduction to CSS: Styling elements, selectors, properties.
- o Building a basic webpage.

• JavaScript:

- o Variables, data types, and operators.
- o Functions, loops, and conditionals.
- Basic DOM manipulation.

• DOM (Document Object Model):

- o Understanding the DOM tree structure.
- o Selecting and manipulating DOM elements.
- Event handling.

• Introduction to Playwright:

- o Overview of Playwright and its features.
- o Comparison with other automation tools (Selenium, Puppeteer).

Day 2-3: Playwright Fundamentals

• Installation and Setup:

- o Installing Playwright on different OS (Windows, macOS, Linux).
- o Setting up a project (Node.js environment, npm packages).
- o Configuring Playwright (launch options, browser configurations).

Creating and Running Tests:

- Writing a basic test script.
- o Running tests using the Playwright test runner.
- Understanding test results and logs.

Day 4-5: Core Playwright Concepts

Selectors:

- Using CSS selectors.
- Using XPath.
- Playwright-specific selectors (e.g., text=, role=).

• Interacting with Elements:

- o Clicking buttons and links.
- Typing into input fields.
- Selecting options from dropdowns.
- o Drag and drop operations.

• Assertions:

- Using Playwright's assertion library.
- o Common assertions (e.g., toBeVisible, toHaveText).
- o Custom assertions.

• Navigation:

- Handling page navigation and redirects.
- Waiting for elements to appear.
- Handling page load and AJAX requests.

Day 6: Frames and Iframes & Handling Dialogs

- Frames and Iframes:
 - Selecting and interacting with frames.
 - Handling nested iframes.
- Handling Dialogs:
 - o Managing alerts, prompts, and confirmations.
 - Automating dialog interactions.

Day 7-8: Advanced Playwright Features I

- Network Interception:
 - o Mocking network requests and responses.
 - o Monitoring network activity.
- Emulating Devices:
 - o Simulating different devices (e.g., mobile, tablet).
 - Configuring screen sizes and input methods.
- Visual Comparisons:
 - o Taking screenshots.
 - o Comparing screenshots for visual regressions.

Day 9: Advanced Playwright Features II

- Multi-browser Testing:
 - Running tests across Chromium, Firefox, and WebKit.
 - o Configuring browser-specific options.
- Parallel Test Execution:
 - o Running tests in parallel.
 - o Configuring parallelism in Playwright.

Day 10: Playwright Integration with Various Test Frameworks

- Playwright Test (Native Test Runner):
 - o Writing and running tests with Playwright Test.
- Jest:
 - o Integrating Playwright with Jest.
 - Writing Jest test cases using Playwright.
- Mocha:
 - Setting up Playwright with Mocha.
 - Writing Mocha test cases.
- Cucumber.js (BDD):
 - o Setting up Playwright with Cucumber.js.
 - Writing BDD test cases using Playwright.
- Allure Report Integration:
 - o Generating test reports with Allure.

Customizing Allure reports.

Day 11: Handling Object Recognition Problems I

- Robust Selectors, Chained Selectors, Playwright-Specific Selectors:
 - o Creating robust and reliable selectors.
 - Using chained selectors for complex elements.
- Locators API:
 - o Using the Locators API for efficient element selection.
- Handling Dynamic Elements:
 - o Strategies for dealing with dynamic content.
- Text and Attribute Assertions:
 - o Validating text content and attributes.
- Retry Mechanism for Flaky Elements:
 - o Implementing retry logic for unstable elements.

Day 12: Handling Object Recognition Problems II

- Handling Frames and Iframes:
 - o Techniques for working with frames and iframes.
- Shadow DOM Handling:
 - o Accessing and interacting with Shadow DOM elements.
- Using expect for Enhanced Assertions:
 - o Leveraging the expect library for more powerful assertions.

Day 13: Error & Exception Handling

- Try-Catch Blocks:
 - o Implementing try-catch for error handling.
- Playwright Test Hooks:
 - o Using hooks for setup and teardown.
- Event Handling:
 - o Managing events during test execution.
- Custom Error Handling Functions:
 - o Creating custom error handling logic.
- Retry Logic:
 - o Implementing retries for failed tests.
- Assertions and Expect Assertions:
 - Advanced usage of assertions and expect statements.

Day 14: Test Organization and Management

- Test Suites and Test Cases:
 - Structuring tests into suites and cases.
- Data-driven Testing:
 - o Parameterizing tests with multiple data sets.
- Test Fixtures:
 - Setting up and tearing down test preconditions.
- Custom Test Reports:

o Generating and customizing test reports.

Day 15: Integration with CI/CD, Debugging, and Best Practices

• Continuous Integration:

Integrating Playwright tests with CI tools (Jenkins, GitHub Actions, GitLab CI).

• Continuous Deployment:

- o Running tests in the deployment pipeline.
- Ensuring test quality in CI/CD workflows.

• Debugging and Troubleshooting:

- Using Playwright's debugging tools (e.g., debug).
- Analyzing test executions step-by-step.

• Best Practices:

- Writing clean, maintainable, and reusable code.
- Ensuring tests are reliable and not flaky.
- o Optimizing test performance for faster execution.

• Hands-on Practice & Community Resources:

- Web-based application automation using Playwright.
- o Leveraging official Playwright documentation and API references.
- Engaging with the Playwright community on platforms like GitHub, Stack Overflow, and Slack.