# **🛣️ WebdriverIO (WDIO) Roadmap:**

## **🧩 Phase 1: Absolute Beginner**

1. **Understand What WDIO Is**
   1. What is WebDriverIO?
   2. Difference between WebDriver protocol and DevTools protocol.
2. **Setup Your Environment**
   1. Install Node.js and npm.
   2. Install VS Code or any good code editor.
3. **Install WDIO**
   1. Learn how to install WDIO via npm install @wdio/cli.
   2. Understand WDIO configuration wizard (npx wdio config).
4. **Understand WebDriverIO Project Structure**
   1. Learn about folders like /tests, /pageobjects, /wdio.conf.js.
5. **Write Your First Test**
   1. Open a browser.
   2. Navigate to a URL.
   3. Validate the title.
6. **Locators & Selectors**
   1. id, class, css selector, xpath.
   2. Best practices for finding elements.
7. **WDIO Basic Commands**
   1. browser.url(), $(selector), .click(), .setValue(), .getText(), .waitForExist().
8. **Assertion Libraries**
   1. Install and use expect-webdriverio.
   2. Basic assertions: toBeDisplayed(), toHaveTextContaining().
9. **Synchronization in WDIO**
   1. Understand implicit waits.
   2. Use explicit waits with waitUntil().
10. **Run Tests Locally**
    1. Understand how to run tests using npx wdio run wdio.conf.js.

## **🛠️ Phase 2: Intermediate**

1. **Learn Page Object Model (POM)**
   1. Why use POM?
   2. Create pageobject classes for maintainable tests.
2. **Advanced WDIO Commands**
   1. scrollIntoView(), isEnabled(), keys() for keyboard events.
3. **WDIO Hooks**
   1. before(), after(), beforeTest(), afterTest() for setup/teardown.
4. **Handle Different Browser Windows/Tabs**
   1. newWindow(), switchWindow(), getWindowHandles().
5. **Handle Alerts/Popups**
   1. acceptAlert(), dismissAlert(), getAlertText().
6. **Data-Driven Testing**
   1. Run the same test with different data using arrays/JSON files.
7. **Parallel Test Execution**
   1. Learn about maxInstances in WDIO config.
8. **Cross-Browser Testing**
   1. Setup tests for Chrome, Firefox, Safari using services.
9. **Take Screenshots**
   1. browser.saveScreenshot(), automatic screenshots on test failures.
10. **Reporting Tools**
    1. Use Allure Reporter or WDIO’s built-in reporters (spec, dot, junit).

## **🚀 Phase 3: Advanced/Expert**

1. **Advanced Configuration**
   1. Environment-specific config files.
   2. Command-line options overrides.
2. **Cucumber Framework Integration**
   1. Use Gherkin syntax (Feature, Scenario, Given, When, Then).
   2. @wdio/cucumber-framework setup.
3. **Mocha Framework Advanced**
   1. describe, it, beforeEach, afterEach.
   2. Retry mechanism for flaky tests.
4. **API Testing with WDIO**
   1. Use WDIO + Axios/Node-fetch for API calls.
   2. Validate API responses inside your tests.
5. **Service Integrations**
   1. Integrate Browserstack, SauceLabs for cloud-based cross-browser testing.
   2. Learn about @wdio/browserstack-service.
6. **Visual Regression Testing**
   1. Install and configure WDIO Visual Regression Service.
   2. Compare screenshots for UI differences.
7. **Continuous Integration**
   1. Run WDIO tests in GitHub Actions, Jenkins, GitLab CI/CD.
8. **Build a WDIO Test Framework from Scratch**
   1. Logging, Utilities, Test Data management.
   2. Create reusable functions and custom commands.
9. **Best Practices for Test Automation**
   1. Test pyramid: UI vs API vs Unit.
   2. Keeping tests independent.
   3. Mocking external dependencies when needed.
10. **Contribute to Open Source / Real-World Projects**
    1. Contribute to WDIO GitHub.
    2. Start building your portfolio with real-world projects.
    3. Mentor juniors once you're solid. Teaching is mastery!

# **🧠 Bonus Resources:**

* **Official Docs:** <https://webdriver.io/docs/gettingstarted>
* **YouTube Channels:**
  + Naveen AutomationLabs
  + LambdaTest tutorials
* **Free Practice Sites:**
  + <https://the-internet.herokuapp.com/>
  + <https://demoqa.com/>