MASTER TEST AUTOMATION (PRO) - PLAYWRIGHT JS

WEB + API

HIGH-LEVEL TECH STACK:



JAVASCRIPT - MODULE 1

01 FUNDAMENTALS OF JAVASCRIPT

Introduction to JavaScript:

- What is JavaScript?
- Key Features & Benefits
- JS Versions & History
- JavaScript Component
- Memory Management
- Tools & Installations

JavaScript Data Types:

- Primitive vs Non-Primitive Types
- Pass by Value vs Pass by Reference
- Mutable vs Immutable Types
- Literal vs Non-Literal Values
- Type Conversion - Type Casting & Coercion

Variables in JavaScript:

- var, let, const Explained
- Differences & Scoping Rules
- When to Use Which & Why
- Hoisted & TDZ
- Best Practices for Variable Declaration

JavaScript Operators:

- Assignment operators
- Arthimethic Operators
- Comparison Operators
- Logical Operators
- Concatenation Operator
- Type Operators typeOf & instanceOf

02 CONTROL STATEMENTS

Conditional Statements (Decision-making)

- if Statement
- if...else Statement
- if...else if...else Statement
- Ternary Operator (Short-hand if...else)
- Switch Statement

Looping / Iteration Statements

- for Loop
- while Loop
- do-while Loop

Loop Control Statements

- break
- continue

03 STRINGS

- Basic String Properties
- Accessing Characters
- Search & Check Methods
- Extracting Substrings
- Case Conversion
- String Replacement & Repeat
- Trimming & Padding
- Splitting and Joining
- Other Useful Methods - concat, valueOf etc.

JAVASCRIPT - MODULE 2

04 ARRAYS

- Creating & Accessing Arrays
- Modifying Arrays
- Iterating Arrays

Common Array Methods:

- Adding & Removing Elements
- Finding & Checking Values
- Transforming Arrays
- Advanced Operations

Higher-Order Array Methods:

- Iterating Through Arrays
- Transforming & Filtering Data
- Reducing & Aggregating Values
- Searching & Checking Conditions
- Creating User-Defined Arrays
- User-Defined Arrays

08 FUNCTIONS

- Function Declaration
- Function without Parameters
- Function with Parameters
- Function with Return Value

Other topics:

- Default Parameters
- Function Hoisting

Arrow Functions (=>):

- Basic Arrow Function
- With Parameters and Return
- Shorter Version (Implicit Return)
- Single Parameter
- No Parameters

Function Expression:

- Basic Function Expression
- With Parameters and Return Value

Other topics:

- Not Hoisted
- Can Be Anonymous or Named

Basics of Function Flow:

- Function without callback
- Function with callback (callback accepted, higher-order function)
- Function returning another function
- Function with callback and parameter
- Function with nested callback

05 SET & MAP

Set Operations:

- Creating & Accessing Sets
- Modifying Sets
- Iterating Sets
- Common Set Methods
- User-Defined Set

Map Operations:

- Creating & Accessing Maps
- Read Values
- Modifying Maps
- Deleting & Clearing
- Iterating Maps
- Common Map Methods
- User-Defined Map

06 ERROR HANDLING

- JS Exceptions - Runtime errors.
- Exception Types - ReferenceError, TypeError, SyntaxError, etc.
- Handling Mechanism - try, catch, finally, and throw
- Custom Exceptions - using throw new Error("message")

07 OOPS

- Class, Method & Object.
- Inheritance and its types.
- Polymorphism.
- Abstraction & Encapsulation.

09 CONSTRUCTOR

- Defining Constructor

Types of Constructor:

- Default Constructor
- User-defined Constructor
- Paramaterized Constructor

Examples:

- Constructor Using a Regular Function
- Using class Syntax with Construtor
- Constructor with Class, method and Objects
- Use of static keyword - Method & Variables.

JAVASCRIPT - MODULE 3

10 JAVASCRIPT OBJECTS

- Javascript Objects
- JavaScript Arrays / Nested Arrays.
- Javascript Objects Destructing
- JavaScript Arrays Destructing

Import / Export:

- Import named function
- Import default class/function
- Import everything
- export default - Export from module

Math Functions:

- Round numbers
- Find maximum and minimum
- Generate random numbers
- Exponentiation & roots
- Absolute values & signs

Regular Expressions:

- Define patterns to match text
- Test and find matches in strings
- Use methods to manipulate string data

11 ASYNC JAVASCRIPT / AWAIT & PROMISES

1. Introduction to Asynchronous Thinking

- Function without setTimeout
- Function with setTimeout
- Function with setTimeout inside a callback
- Function with setTimeout inside a callback with parameter

2. Promises - Modern Asynchronous Flow:

- Function without Promise
- Function with Promise - using resolve and reject
- Function with Promise - only using resolve
- Function with Promise - return value
- Function with Promise - involving multiple functions

3. async & await - Cleaner Async Code

- Function without async/await
- Function with async/await
- Using await on a function that doesn't return a Promise
- Function using try...catch with async/await

4. Combining Concepts (Advanced Practice)

- Function using async, await, callback, setTimeout, and Promise
- Chained Promise functions using .then() and .catch()

12 BUILT-IN MODULES & UTILITY FUNCTIONS

File Operations (with .txt files):

- Write data to a .txt file
- Read data from a .txt file
- Append data to a .txt file
- Delete a .txt file

Date Functions:

- Create date object
- Get year, month, date
- Get time
- Format date

TYPE-SCRIPT

Basic Types & Functions

- Basic Types: string, number, boolean, any, unknown, void.
- Arrays & Tuples
- Enums
- Functions: typed parameters & return types, optional/default and function overloading

Generics

- Generic Functions
- Constraints & Default Generics

Type Aliases & Advanced Types

Classes & OOP

- Classes: properties, methods
- Inheritance
- Implementing Interfaces
- Implementing in Errors.

Modules & Utilities

- Modules: import & export
- Utility Types: Partial, Required, Readonly, Pick, Omit, Record, Exclude, Extract, NonNullable
- Key Types: keyof, typeof, indexed access types

PLAYWRIGHT AUTOMATION

01 INTRODUCTION TO PLAYWRIGHT
- What is Playwright?
- Comparison with Selenium & Cypress
- Advantages of Playwright
- Supported browsers & platforms
- Installation & Setup (Node.js + Playwright)
- Project structure overview

02 PLAYWRIGHT TEST RUNNER
- Introduction to Playwright Test
- test(), expect(), describe()
- beforeEach, afterEach, beforeAll, afterAll
- Parallel execution
- Running tests via CLI
- Configuration using playwright.config.js

03 BROWSER & CONTEXT
- Launching browsers: Chromium, Firefox, WebKit
- Browser types: headed vs headless
- Creating browser contexts
- Playwright show-report
- Managing multiple contexts & incognito mode

04 PAGE INTERACTION
- Navigating to URLs
- Click, type, fill, press, dblclick, check/uncheck
- Waits: auto-wait, explicit wait, waitForSelector
- Handling popups, dialogs, alerts

05 SELECTORS IN PLAYWRIGHT
- Locator Strategies: CSS & X-path
- Playwright Locators vs Selectors
- page.locator() vs direct handling
- Best Practices in Element Selection

06 ASSERTIONS & EXPECT API
- Using expect() for assertions
- toHaveText, toBeVisible, toBeChecked, etc.
- Soft assertions
- Debugging failed assertions

07 FRAMES & IFRAMES
- Identifying and switching frames
- Interacting inside frames
- Nested frames handling

08 HANDLING FORMS & INPUTS
- Working with input fields, textareas
- Dropdowns: selectOption() & evaluate()
- File uploads
- Form submission and validation

09 KEYBOARD AND MOUSE EVENTS
- Simulating keyboard input
- Special key presses (Enter, Tab, and many more.)
- Mouse movement, hover, drag-and-drop

10 SCREENSHOTS & VIDEO RECORDING
- Capture full-page and element screenshots
- Record test video
- Trace viewer & debugging with trace file

11 ADVANCED PLAYWRIGHT FEATURES
- Multi-tab/multi-window handling
- Handling file downloads & uploads
- Extracting data from html webtables.
- Working with browser storage (cookies, localStorage)
- Capturing console logs, performance metrics

12 CROSS-BROWSER TESTING
- Running tests across Chromium, Firefox, WebKit
- Setting browser-specific capabilities
- Mobile emulation (iPhone, Pixel, etc.)

FRAMEWORK & IPT CLASSES

01 PAGE OBJECT MODEL (POM) IN PLAYWRIGHT

- Creating reusable page classes
- Implementing POM structure
- Separating test logic from test data
- Best practices for maintainability

02 BASE CLASS CREATION FOR PAGE ACTIONS

- Centralizing page object
- Reusable functions in base class
- Reducing duplication in page classes

03 TEST DATA MANAGEMENT

- Using excel files for data-driven testing
- Reading, Writing & Updating data from Excel.

04 CUCUMBER - BDD

Cucumber + Playwright Setup:

- Project structure
- Installing dependencies & configuration
- Folder structure

Cucumber Basics:

- What is BDD ?
- Gherkin Syntax & its keyword.
- Writing Feature Files (.feature)

Writing Step Definitions:

- Mapping Gherkin steps to code
- Reusable & Parameterized steps
- Async/await in step definitions

Test Execution & Reports:

- Running tagged scenarios
- Parallel execution setup
- Generating HTML Reports
- Screenshots integration

Hooks:

- Before, After, BeforeAll, AfterAll hooks
- Creating custom world
- Attaching screenshots, logs, etc. in hooks

Advanced Concepts:

- Data-driven Testing using Examples/Data Tables
- Custom Parameters & Transformations
- Using Background for shared steps
- Retry logic in steps

PROJECTS

E-commerce Domain - Grocery Application

- End-to-end order placement, including the cancellation process

Hospitality Domain - Hotel Booking

- End-to-end workflow from booking to cancellation

Travel Domain - Flight Booking

- Airline ticketing, travel management, and cancellation

GROCERY PROJECT - WEB AUTOMATION

Modules covered in the Grocery Application:

- User Authentication (Login & Logout)
- Product Search & Menu Navigation
- Cart Management (Add/Remove items)
- Checkout Process (Address selection & order summary)
- Payment Integration (Debit/Credit card transactions)
- Order Management (Order placement, history & tracking)
- Favorites Module (Save and manage favorite items)
- User Profile (Upload profile picture & manage details)

API AUTOMATION

06 API TESTING WITH PLAYWRIGHT

- Sending HTTP requests (GET, POST, PUT, DELETE)
- Parameters - Query & Path
- Headers & Authorizations
- Using request fixture
- Assertions on API response
- Status Code
- Serialization & De-Serialization
- Intercepting network requests and responses
- Modifying API data (mocking)
- Integration with Cucumber Project
- Reporting & Logs.

HOTEL BOOKING PROJECT - WEB AUTOMATION

Modules covered in the Hotel Booking Application:

- User Authentication (Login & Logout)
- Hotel Search (Search hotels based on location, date, and preferences)
- Search Results (Sorting & filtering options)
- Booking Page
- Enter user information
- Add special requests
- Payment module (Credit/Debit card & UPI)
- Hotel Booking Management
- Search existing bookings
- Update booking details
- Cancel booking

MODULES COVERED IN THE API AUTOMATION

- Authentication - Basic, Bearer, OAuth 2.0, API
- Account - Profile management, personal details- Upload Profile Pic
- Cart & Checkout - Add, remove, and update items, address & payment service
- Orders - Order placement, history tracking, and status updates.
- Booking - Search, book, modify, and cancel reservations.

MODULES COVERED IN THE WEB AUTOMATION

- Base Class for API
- Support / Utils
- Playwright Test
- Cucumber BDD
- Config json
- Allure Reporting

MODULES COVERED IN THE API AUTOMATION

- Base Class for API
- POJO Class for payloads
- Playwright Test
- Cucumber BDD
- JSON Path Expressions
- Allure Reporting

PROCESS-ORIENTED

DEVOPS

DevOps Integration - Git & Jenkins

Git & GitHub -

- Version control, Push & Pull, branching, and merging basics
- Git Commands

Jenkins CI/CD -

- Running Playwright tests in pipelines with reports - Freestyle Jobs
- Pipeline jobs

AGILE-SCRUM

- Agile Methodologies
- Scrum Model - Incremental & Iterative Approach
- Sprint / Timebox

Scrum Meetings -

- DSU
- Sprint Planning / Grooming
- Sprint Demo
- Sprint Retrospective
- Scrum Artifacts - Product Backlog & Sprint Backlog

Scrum Roles:

- Development Team - Dev & QA, Product Owner, Scrum Master

SQL

- SQL query Basics- select/ filtering/ sorting/group by
- Advanced SQL Querying - Joins / Subqueries / Aggregate functions / Analytical functions
- Data Manipulation and Management
- MySQL - DBMS

JIRA

- Scrum Board
- User Story
- Product Backlog & Sprint Backlog
- Acceptance Criteria
- Task & Bug Creation
- Other Integrations.

MANUAL TESTING

- 1. SDLC & STLC
- 2. Bug / Defect and Its Lifecycle
- 3. Waterfall, V Model & Iterative Model
- 4. Testing techniques - a. Boundary Value Analysis, b. Equivalence Partitioning, c. State Transition, d. Decision Table, e. Error Guessing Technique
- 5. Test Strategy & Test Plan document & Test case template
- 6. Functional and Non-functional testing
- 7. Testing Types - Smoke, Regression, Sanity, E2E, Integration, Retesting, Adhoc Testing
- 8. Effective bug reporting - Priority & Severity

PLAYWRIGHT - GEN AI

GEN AI - PLAYWRIGHT MCP - COPOLIT - OPEN-AI

GENAI FOR TEST AUTOMATION

- INTRODUCTION TO AI - MACHINE LEARNING, DEEP LEARNING & GENERATIVE AI IN TEST AUTOMATION
- CONFIGURING & SETUP OPEN-AI & OTHER TOOLS
- BASICS OF PROMPT ENGINEERING FOR QA
- AI-BASED LOCATOR SUGGESTIONS & SELF HEALING SCRIPTS
- USING GEN AI FOR TEST DATA GENERATION (FAKER, AI MODELS)
- AI-POWERED DEFECT PREDICTION & ROOT CAUSE ANALYSIS
- REAL-TIME CASE STUDIES & PROJECTS
- FUTURE OF TESTING WITH GEN-AI

PLAYWRIGHT MCP + COPILOT

- GENERATE MEANINGFUL TESTS FROM NATURAL LANGUAGE REQUIREMENTS
- RECORD AND SCAFFOLD ROBUST AUTOMATION SCRIPTS WITH CODEGEN
- SIMULATE ADVANCED USER INTERACTIONS PRECISELY USING EXTENAL TOOLS
- EXECUTE, DEBUG, AND OPTIMIZE TESTS EFFICIENTLY
- INTEGRATE CHATGPT / COPILOT AI SUGGESTIONS SAFELY
- AUTOMATE DOCUMENTATION AND REPORTING FOR FASTER DELIVERY