Cypress

Course Duration: 40 Hours (5 Days)

Overview

The Cypress course is a comprehensive training program designed for developers and QA engineers who want to master end-to-end testing using Cypress. This course covers everything from the basics of how Cypress works to advanced techniques for testing modern web applications. Learners will understand Cypress's unique architecture, how it differs from Selenium, and explore its rich set of features through hands-on exercises. Starting with installation and setup, learners will become familiar with Cypress's folder structure and configuration. The course then dives into interacting with web elements, understanding Cypress commands, and its asynchronous nature, which are crucial for writing robust tests. As the course progresses, participants will learn about Cypress basics, such as running tests in headless mode and debugging, before moving on to more advanced topics, including environment variables, custom commands, and API testing. By obtaining Cypress certification, participants will demonstrate their expertise in this powerful testing tool. This Cypress course is ideal for those looking to enhance their testing skills and ensure the reliability of their web applications, ensuring they are well-equipped to integrate Cypress into their continuous integration environment.

Audience Profile

The Cypress course by Koenig Solutions is designed for professionals seeking to master end-to-end testing for web applications. It is suitable for:

- QA Engineers and Testers
- Software Developers in Test (SDET)
- Frontend Developers
- Full Stack Developers
- DevOps Professionals
- Technical Project Managers
- Automation Test Engineers
- QA Leads and Managers
- Anyone interested in learning Cypress for web application testing

Course Syllabus

Introduction to Cypress

- What is Cypress?
- Cypress Architecture
- How is Cypress different from Selenium?
- Cypress.io Website: "How it Works?"

- Cypress.io Website: "Features"
- Running Sample Code and Explaining Test Runner & Playground

Installation & Setup

- Prerequisites
- VS Code
- Node.js 12+
- Creating a Project
- Cypress Installation
- Cypress Configuration
- Understanding cypress.json and package.json Files
- Explaining Cypress Folder Structure

Interaction with Web Elements

- DOM Terminology
- Test Structure
- Types of Locators
- Writing the First Sample Test
- Launching a URL
- Finding Web Elements
- Implicit Waits, Pause, and Debugging
- Saving the Subject of a Command
- Using the Invoke Command
- Handling UI Elements
- Checkboxes & Radio Buttons
- Lists & Dropdowns
- Web Tables
- Popups & Tooltips
- Handling Child Windows
- Handling Frames

Cypress Assertions

- **Chaining Assertions**
- Working with **Shadow DOM**

Deep Dive into Cypress Commands & Asynchronous Nature

- Understanding get and find Commands
- Grabbing Text for Validations using the Cypress Text Command

- Difference Between jQuery Methods and Cypress Commands
- Handling Asynchronous Promises in Cypress
- Handling Invisible Elements in Cypress

Cypress Basics

- Running Tests in Headless Mode
- Using Implicit Waits, Pause, and Debug
- Validating the Count of Elements on a Page
- Understanding Cypress Logs
- Performing Browser Refresh & Reload
- Working with Variables and Aliases
- Capturing Screenshots

Cypress Advanced Features

- Environment Variables
- Using Fixtures
- Creating Custom Commands
- Working with Hooks
- Implementing Page Object Model (POM) in Cypress
- BDD with Cucumber
- Using Mocks and Stubs
- API Testing with Cypress
- Cypress Retries
- Using Cypress Reporters
- Running Tests on Different Browsers
- Cross-Browser Testing
- Performing Visual Testing
- Exploring the Cypress Dashboard

Cypress End-to-End Testing

- Running Cypress Tests in a CI Environment
- Introduction to Git
- Introduction to Jenkins
- Running Cypress Tests in Different Browsers from Jenkins