Full Stack
AI-Powered Testing
With
Live Projects & Placements

Courses Content :

[Manual Testing + Agile Testing + Jira + Domain Testing + Linux Testing + Playwright + Typescript + API Testing + Database Testing + AI Testing + Mock Interviews + Placements]

Manual Testing

Software Testing Fundamentals

- Introduction to Software Testing
- Software Development Process
- SDLC Real Time process Steps
- What is Software & Software Testing?
- Define QA Process
- History of Software testing
- Objective of Testing
- Why Testing required?
- When to start Testing
- Testing importance
- Definition, Basics & Types
- Software Testing as a Career Path (Skills, Salary, Growth)
- Why software has Defects
- Services based vs Product based Companies

Testing Roles and Responsibilities

- Software Test Engineer
- Real Time Job Role of Tester
- Senior Software Test Engineer
- Test Lead
- Test Manager

Software Testing Methods

- White Box Testing
- Black Box Testing
- Gray Box Testing
- Difference of Whitebox & Blackbox Testing

Software Development Life Cycle - SDLC

- What is SDLC?
- SDLC Phases
- SDLC Models
- Waterfall model
- V model
- Verification & Validation
- Agile Model

Agile Process Concepts

- What is agile?
- Why Agile is important
- Agile Testing principles
- What is mean by scrum master
- Roles of Scrum Master
- Sprint Planning
- Sprint Release
- Product Backlog
- What is Epic
- Concept of User Stories
- Defect Backlog
- Standup meeting
- Status meeting
- Scrum meeting

Agile Testing

- Agile Overview
- What is Agile Testing
- Agile Principles
- Agile Methodologies/Frameworks
- Agile Testing Life Cycle
- Agile Terms

Software Testing Life Cycle - STLC

Understanding the requirements

- Requirements Specification
- Business requirement specification
- Software requirement specification
- Functional requirement specification

Test Plan Preparation

- Overview of Test Plan
- Entry and Exit criteria
- Test Plan template

Test Engineer Responsibilities

- LAB Checklist

Creation and working with Folder Structure

Test Scenarios

- Test Scenario Entry and Exit Criteria
- Test Scenario Template
- Test Scenarios Identification
- Writing Test Scenarios for application

Test Cases

- Test cases Entry and Exit Criteria
- Test cases Template
- Test cases Identification
- Test Design Guidelines
- Writing Test cases for application
- Good Test Case design steps
- Test Data Preparation

Test Case Design Techniques

- Equivalence Class Partitioning
- Boundary Value Analysis
- State Transition
- Decision Table
- White box Testing Techniques

Software Testing Types

- Smoke Testing
- Sanity Testing
- Re-Testing
- Regression Testing
- Static Testing
- Dynamic Testing
- Ad-hoc Testing
- Functionality Testing
- Usability Testing
- Compatibility Testing
- Data Base Testing
- Interface Testing
- Performance Testing
- Security Testing

User Acceptance Testing

- Alpha Testing
- Beta Testing
- UAT Testing

Test Execution

- When to start Test Execution
- Process to start test execution
- What is build
- Build Release process
- Executing Test Cases on multiple builds
- Test Cases Execution Status

Bug/Defect management

- Defect/Bug Life Cycle
- Defects Reporting
- Defects Reporting Template
- Defects Reporting & Re-Testing
- Defects Closing
- Severity and Priority
- Defect /Bug/Error/Failure
- Defects in Real Time application
- How to find more bugs

Test/Project management Tool: JIRA

- Introduction to Jira
- Features of Jira
- Test Case Design in Jira
- Creating Test Cycle
- Test Execution in Jira
- Bug Reporting using Jira
- Jira Dashboard

Status Reports Process

- Daily Status Report
- Daily Defect Report
- Weekly Status Report
- Monthly Status Report

Test Closure

- Criteria for Test Closure
- Test Closure process
- Test Summary Reports
- When testing need to be stopped

Domain Testing (Banking/Insurance)

- Domain overview
- Why Domain knowledge is Required ?
- Types of Domains
- Domain Terms
- Domain importance

- Types of domain application
- Finding bugs using Domain Testing

Linux Testing – For Testers

- What is Linux
- Key Features of Linux
- Key Components of Linux
- Why Linux for Testers
- Linux File Structure
- Linux Commands for Tester

Playwright Automation With AI TESTING

[Playwright + Type Script / Java Script + Frameworks + API Testing + AI Testing -Gen AI + Prompt Engineering + Agentic AI]

Module 1: Automation Testing Foundations & Playwright Overview

- Introduction to Automation Testing
- Manual vs Automation Testing
- Automation Testing Life Cycle (ATLC)
- Limitations of Selenium
- What is Playwright?
- Why Companies Prefer Playwright
- Key Playwright Features
    - Auto Waiting & Stability
    - High-Speed Execution
    - Cross-Browser Support
    - Parallel Execution
- Playwright vs Selenium – Practical Comparison
- JavaScript vs TypeScript – Industry Preference
- Playwright Architecture & Execution Flow

Module 2: JavaScript + TypeScript

Core Programming

- Variables (var, let, const)
- Data Types
- Operators
- Conditional Statements
- Loops , Keywords
- Functions & Arrow Functions
- Arrays & Array Methods
- Strings & String Methods
- Objects & JSON

Asynchronous Programming

- Callbacks
- Promises
- Async & Await

TypeScript Concepts

- Why TypeScript?
- Type Inference & Explicit Types
- Function Type Annotations
- Array & Object Type Annotations
- Enums ,Type Assertion
- Interfaces ,Modules

Module 3: OOPS Concepts for Playwright

- Classes methods & Objects
- Access Modifiers (public, private, protected)
- Inheritance
- Polymorphism
- Encapsulation & Abstraction
- interface
- Practical OOPS implementation in Playwright Framework

Module 4: Playwright Setup & Core Concepts

- Playwright Installation
- Project Structure & Best Practices
- Playwright Test Runner
- Browser, Context & Page
- Writing First Playwright Test
- Headed vs Headless Execution
- Playwright Configuration File (playwright.config.ts)
- Running Tests using CLI & npm scripts

Module 5: Locators & User Actions

Locators

- Built-in Playwright Locators
- CSS Selectors (Basic to Advanced)
- XPath (Functions, Operators, Axes)
- Role, Label, Text & Placeholder Locators
- Handling Dynamic Elements
- Best Locator Strategies

User Actions

- Input Fields & Buttons

- Checkboxes & Radio Buttons
- Dropdowns
- Mouse Actions (Hover, Right Click, Drag & Drop)
- Keyboard Actions

Module 6: Advanced UI Handling

- Auto Waiting & Smart Synchronization
- Explicit Timeouts
- Handling Alerts & Dialogs
- Handling Frames & iFrames
- Shadow DOM
- Web Tables
- File Upload Automation
- Multiple Tabs & Windows

Module 7: Assertions, Debugging & Stability

- Assertions using expect
- Hard Assertions vs Soft Assertions
- Debugging Techniques
  - page.pause()
  - Playwright Inspector
- Handling Flaky Tests
- Retry Mechanism
- Screenshots on Failure
- Video Recording
- Trace Viewer

Module 8: Test Suite Management & Execution Control

- Test Suite Structure
- Grouping Tests
- Tags & Annotations
- Skip / Only / Fixme
- Test Parameterization
- Parallel Execution
- Cross-Browser Execution
- Device Emulation
- Global Setup & Global Teardown

Module 9: Excel file operation

Reading Test Data

- Reading test data from **Excel (.xlsx)** using Playwright
- Reading test data from **CSV files**
- Writing **data to excel**
- Excel with Playwright

Module 10: Automation Frameworks with Playwright

Hybrid Automation Framework

- Test Scripts creation
- Test Suite Management
- Base Page Design
- Reusable Common Actions
- Config & Environment Management
- Allure reports
- CI/CD Ready Framework Structure
- Git hub actions

Page Object Model (POM) Framework

- Introduction to POM
- Separating Page Locators & Page Actions
- Test Script Design using POM
- Best Practices for POM
- Maintenance & Scalability Advantages

Cucumber Framework with Playwright (BDD)

- Introduction to BDD (Behavior Driven Development)
- Cucumber Architecture
- Feature Files & Gherkin Syntax
- Scenarios & Scenario Outlines
- Step Definition Implementation using Playwright

- Hooks in Cucumber
- Mapping Feature → Step Definition → Page Classes
- Running Cucumber Tests with Playwright

Module 11: API Testing with Playwright

- API Testing Fundamentals
- HTTP Methods (GET, POST, PUT, DELETE)
- API Request Context
- Request Headers & Payload Handling
- Response Validation
- API Authentication
  - Token
  - Bearer
  - Basic
- API Chaining
- Schema Validation using AJV
- UI + API End-to-End Scenarios

Module 12: Reporting & CI/CD Integration

Reporting

- Playwright HTML Reports
- Allure Report Setup & Configuration
- Allure Annotations
- Attaching Screenshots & Logs
- CI-Friendly Reporting

Git & GitHub

- Git Basics (CLI)
- Repository Management
- Branching Strategy
- Pull Requests & Merging

Jenkins

- Jenkins Installation & Setup
- Creating Jenkins Jobs
- Integrating Playwright Framework with Jenkins
- Scheduled & Nightly Builds
- Jenkins + Allure Report Integration

Module 13: AI-Powered Automation with Playwright

- GitHub Copilot for Automation
- AI-Based Test Case Generation
- Smart Script Suggestions
- AI-Assisted Debugging
- Playwright Codegen (Record & Playback)
- VS Code Playwright Extension
- Playwright with MCP (AI + Automation)

Module 14: Data Base Testing using SQL

- What is DataBase
- Types of DataBase
- What is SQL
- Writing SQL queries
- Types of commands
- *DDL: Data definition language
- *DML: Data manipulation language
- *DCL: Data control language
- AND / OR clauses
- Where Clauses
- Update & Delete DataBase
- Primary Key VS Foreign Key
- Joining Tables
- Right/Left joins
- joining multipul tables
- Roles and Responsibility of Tester
- Differences between User-Interface Testing and Data Testing
- Sample TestScenarios/test case for database testing

AI – TESTING

With

[Gen AI + Prompt Engineering + Agentic AI]

Module 1: Foundations of AI for Testers

- What is AI? – Overview & Real-world Relevance
- How AI Works – Data, Algorithms, and Learning Loop
- Key Components of AI
  - Machine Learning (ML)
  - Deep Learning (DL)
  - Neural Networks
  - Cognitive Computing
  - NLP (Natural Language Processing)
  - Computer Vision

Module 2: Types of AI & Their Relevance in Testing

- Narrow AI vs General AI vs Super AI
- Reactive Machines vs Limited Memory vs Theory of Mind vs Self-Aware AI
- Practical Use Cases for QA/Testers

Module 3: Generative AI in QA

- What is GenAI? – Why It's a Game Changer for Testers
- Applications in QA
  - Test Case Generation
  - Test Data Generation
  - Test Documentation (Reports, Plans, Closure)
- Practical Hands-on Prompts
  - "Generate 20 negative test cases for a login page"
  - "Create 100 fake user profiles for testing"

Module 4: Prompt Engineering for Testers

- What is Prompt Engineering & Why It Matters in QA
- Types of Prompting
  - Zero-Shot, Few-Shot, Role-Based, Format-Guided
- Crafting Effective Prompts: Formula (Role + Task + Context + Constraints + Format)
- Best Practices & Pitfalls
- Hands-on Labs
  - Writing test cases with structured prompts
  - Test Data generation prompts
  - Bug report optimization with AI

Module 5: Agentic AI for Test Automation

- What is Agentic AI? (AI + LLM + Autonomy + Actions)
- Why It's Powerful in QA
- Applications in QA
  - Self-Healing Test Automation
  - Smart Regression Testing
  - Autonomous Test Execution & Bug Logging
- Demo: How an AI Agent Automates Login Test Flow

Module 6: Gen AI Across STLC (Software Testing Life Cycle)

- Requirement Analysis with AI
- Test Planning with AI Assistance
- Test Case Design – Manual & Automation Scripts
- Test Execution – Self-healing Automation
- Defect Reporting & RCA with AI
- Test Closure – Automated Reports & Insights

- Hands-on: Using ChatGPT/Claude to assist in STLC

Module 7: AI Tools for Testers

- **Bottest.ai** – Automated QA for Conversational AI
- **TestCraft** – No-code AI-powered Testing
- **Testron AI** – Agentic Test Generation & Maintenance
- **Testers.ai (Checkie.ai)** – AI Agents for End-to-End Testing
- Hands-on Setup & Execution
    - Installing & Running Testers.ai
    - Generating Reports & Analysis

Module 8: Testing AI Systems (LLMs & Chatbots)

- How AI Tools Work (NLP + ML Flow)
- Testing Methodologies
    - Functional Testing
    - Usability Testing
    - Performance Testing
    - Security Testing
- Scenario Testing: Positive, Negative, Edge Cases
- Using RAG (Retrieval-Augmented Generation) in Testing
- Practical Lab: Testing a Chatbot with AI

Module 9: Advanced Prompt Engineering in QA

- Optimizing Test Case Design with AI
- Test Data Generation for Complex Scenarios (Banking, E-commerce, Insurance)
- Test Case Optimization & Prioritization
- API Testing with AI-generated Scripts (Postman, REST Assured)
- Automating Documentation (Strategy, Traceability Matrix, Release Notes)

Capstone Project

Learners will:

- Use GenAI to generate test cases for a real application
- Apply prompt engineering for bug reporting & test data
- Use an Agentic AI tool (Testers.ai / Testron) in automation
- Submit final test execution report generated by AI

Outcome of this Course:

By the end of this program, participants will be able to:

- Leverage **GenAI** for faster test case & data creation
- Master **Prompt Engineering** to maximize AI testing efficiency
- Use **Agentic AI Tools** for autonomous, self-healing test automation
- Confidently test both **traditional applications** & **AI-driven systems**

==Training Highlights:

- Flexible Learning Options [**Offline/Online/Hybrid**]
- **Offline / Online / Hybrid** lab support
- Real Time **Live projects** job oriented Trainings
- Real Time Trainers with more than **15+** experience
- Free access for **Chat GPT & N8N** Tools
- **Mock Interviews** till getting placed
- **100%** Placement Assistance