Simplilearn Automation Suite Project Write-Up

Overview

The Simplilearn Automation Suite is a sophisticated project aimed at automating various user interactions on the Simplilearn online learning platform. Utilizing Java as the programming language, the suite employs Selenium WebDriver for web automation and JUnit for implementing Test-Driven Development (TDD) methodologies. This suite is designed to streamline processes such as logging in, course navigation, searching for courses, and verifying user profile information.

Objectives

The primary objective of this project is to create an efficient and reliable automation suite for Simplilearn that enhances user experience and aids in quality assurance testing. Specific goals include:

Automated Login: Facilitate automated login processes for different user accounts.

Course Page Navigation: Automate the process of navigating to various course pages.

Course Search Functionality: Implement automation for searching courses using keywords.

Profile Verification: Automate checks for user profile information accuracy.

Tools and Technologies

Programming Language: Java

Web Automation Tool: Selenium WebDriver

Testing Framework: JUnit

Project Management Tool: Maven

Browser Driver: ChromeDriver (can be extended to other browsers)

Project Structure

Project Name: SimplilearnAutomationSuite

Package Name: CourseCrafter

Key Classes:

LoginTest.java: Handles automated testing of the login process.

CourseNavigationTest.java: Automates and verifies navigation to different course pages.

CourseSearchTest.java: Tests the course search functionality.

ProfileVerificationTest.java: Automates the verification of user profile information.

Development Approach

The project follows the Test-Driven Development (TDD) approach, ensuring that testing drives the development process. This methodology enhances the reliability and robustness of the automation suite.

Write Test Cases: Define test scenarios for each functionality.

Implement Automation: Develop automation scripts to fulfill test requirements.

Refactor and Optimize: Continuously improve the code for efficiency and maintainability.

Continuous Integration: Implement CI pipelines for automated testing.