Software Testing

Introduction

Creating a black-box test case involves testing a system or component without knowing the internal workings of the system. Black-box testing is focused on the system's inputs, outputs, and behaviour.

Problem Statement

Use Selenium to automate the test case wherever possible. Please consider the feature that you are implementing in your Capstone Project to identify the test cases.

Here are step-by-step instructions to create a black-box test case:

- Understand the Requirements
- Identify Test Scenarios
- Define Test Inputs
- Determine Expected Outputs
- Create Test Cases
- Include Preconditions and Postconditions
- Execute the Test Case
- Compare Actual vs. Expected
- Document Test Results
- Automate using Selenium (identify the case)
    - Experiments attached**:**
        - Record and Playback using Selenium IDE
        - Validating Elements using Selenium IDE
        - Validating Elements using Assert Mechanism
        - Validating Elements using Verify Mechanism
        - Set up Selenium and Perform a Simple Browser Interaction
        - Locating Web Elements
        - Handling Alerts and Pop-ups
        - Working with Frames and iFrames
        - Data-driven testing with Excel
        - Selenium exception handling**>**

Additional Specifications

- Filled Test case as per attached template.
- Specify the test case that has been automated using selenium along with documentation.

Expected Outcomes

- Test case for the module (use the template shared).
- Capture the outcomes of the exercise:
    - Selenium script for identified test case.
    - Test outcome of the script.
- PowerPoint presentation (use the template shared)
- The above documents to be submitted in Lumen for the assignment.

Evaluation Rubric

Identifying Test Cases	Module for automation	Outcomes of the Test Script	Summary Presentation	Additional feature implemented in Selenium	Total
30	20	20	20	10	100