Program-2

Step1: Recording the Login Test Case

1.Navigate to the Swag Labs login page

2. Manually test the login by entering a registered username and password.

3. Automate this process using TOSCA.

Step2: Creating a New Test Module in TOSCA

1.Create a new module under a folder named "Login Demo".

2. Scan Application

3.XScan dashboard:

Username field

Password field

Login button

4.Click "Save", then "Finish Screen".

Step3: Creating the Login Test Case

1.Navigate to the Test Cases tab and create a new folder named "Login Demo".

2.Drag and drop the Swag Labs module into this folder.

3.Assign test values

Username: (e.g., standard_user).

Password

Login Button

Step4: Executing the Login Test Case

1.TOSCA will automatically enter the username and password.

2.A green checkmark indicates that the test has passed successfully

.Experiment No. 3: Object Identification Methods in TOSCA

Objective: Use different methods (Property, Anchor, Image, Index) to identify objects in a web app.

1. Open TOSCA Commander

- Create a new project using the standard template.

- Name it "Experiment3".

2. Open Google in Chrome

- Visit: https://www.google.com

3. Create a Module for Google Page

- In TOSCA Go to Modules Create a folder named "Google".

- Scan the Google homepage.

4. Add Controls to Module

- Select and add: Search Bar, Google Search Button.

5. Try Different Object Identification Methods

- By Property (default): If object is green, it's uniquely identified.

- By Anchor: Drag Search Bar as anchor if not unique.

- By Image: Capture the image of the button if anchor fails.

- By Index: Enable index-based identification as a last resort.

6. Save Module

7. Create a Test Case

- Folder & test case: 'GoogleSearchTest'.

- Add module: Search Bar = "TOSCA", Button = Click.

8. Run the Test in ScratchBook.

- Should perform a Google search for "TOSCA".

Result: Test confirms all object identification methods work.

Experiment No. 4: Buffer Operations and Product Validation

Objective: Learn and apply buffer operations like Set, Delete, Partial Buffer, and use them in product validation.

1. Open TOSCA Commander

- Go to Modules TBox Automation Tools Buffer Operations.

2. Set Buffer

- Create test case 'Buffer_Test'. Add 'Set Buffer' with Buffer Name: B1, Value: ABCDE.

- Run and check in Buffer Viewer.

3. Name to Buffer

- Add 'Name to Buffer'. Use {B[B1]} to fetch value.

4. Partial Buffer

- Add 'Partial Buffer'. New Buffer: B10, Value: {B[B1]}.

- Start = 2, End = 4 Result = BCD. Optional: Last = 2 DE.

5. Delete Buffer

- Add 'Delete Buffer'. Name: B10. Run and confirm deletion.

6. Real-Time Use Case: Product Name Validation

- Scan product (e.g., Sauce Labs Backpack) and store in buffer.

- Scan Add to Cart, Cart link, and Cart product name.

7. Create Test Case: Product_Buffer - Add all modules. Set actions:

Product Name: Buffer = Buffer1

Add to Cart & Cart Link: Click

Cart Product Name: Verify = {B[Buffer1]}

8. Run the Test Case from ScratchBook on SauceDemo site.

- Product in cart should match buffer value.

Result: Test passes if cart item equals original product name.

PROGRAM 5:

Step 1: Create Test Case

Name the test case: Window Operations.

Step 2: Add Start Program Module

1. Right-click on the test case and select Search and Add Test Step.

2. Search for Tbox Start Program.

3. Add it to the test case.

4. In the value field, enter the full path of the application to be opened.

Example: D:\test.docx

Note: Ensure the file test.docx exists in the given location before executing the test.

Step 3: Add Tbox Window Operation Module

1. Again, search for Tbox Window Operation and add it below the start program.

2. In the Caption, enter: test* to identify the Word file.

3. In the Operation, choose an action (e.g., Minimize).

Step 4: Add Additional Operations

Maximize: Copy the window operation step, change the operation to Maximize.

Close: Add another step and choose the Close operation.

Execution:

Right-click the test case and select Run in ScratchBook.

Observe that the Word file:

Opens *minimize * Maximizes* Closes

Part B: Send Key Operations

Step 1: Create New Test Case

1. Create a test case named Send Key Operations.

2. Add Tbox Start Program to open the existing Word file test.docx.{path =D:\Test.docx}

Step 2: Add Send Key Module

1. Search and add Tbox Send Keys.

2. Set Caption to: test*

3. Under Input, enter the text to be typed in the Word file.

Example: TOSCA Automation Software Testing

Step 3: Add More Send Key Actions

Select All Text:

o Add another Tbox Send Keys.

o Use key command: (CAPS)A to simulate Ctrl+A.

Delete All Text:

o Add another Tbox Send Keys.

o Use key command: {DEL} to delete selected text.(delete "{DEL"}

Save the File:

o Use key command: (CAPS)S to simulate Ctrl+S.(^(S))

Close the Application:

o Use key command: %{F4} to simulate Alt+F4.

Execution:

Run the test case and observe:

Text is typed into the file. All text is selected. Text is deleted. File is saved and closed.

Part C: Mathematical Operations Using Buffer

Step 1: Create New Test Case

1. Create a test case named Math Operations.

Step 2: Add Set Buffer Module

1. Search and add Tbox Set Buffer.

Step 3: Perform Mathematical Calculations

Addition Example

Buffer Name: add

Value: {Math[2+3]} → Output: 5

Experiment 6: Record and Playback Feature

1. Enable Recorder in Settings

Go to Project > Settings > TBox > Execution Recorder.

Set to:

Enabled or

Enabled only on failure

Leave output path as default.

2. Run a Test Case

Open a test case (e.g., Send Key Operation).

Run in ScratchBook.

After execution, check the recording in:

C:\ToscaProjects\<ProjectName>\ToscaCommander\Recordings

3. Playback the Recording

Open .mp4 file in any media player.

Review the recorded execution.

4. Optional: Exclude Specific Test

Right-click a test case (e.g., Login Page).

Create Test Configuration Parameter:

Name = AvoidExecutionRecorder

Value = True

Result

TOSCA records the execution as a video file. Playback allows easy review of each step. If

AvoidExecutionRecorder = True is set, that test won’t be recorded.

PROGRAM 7:->

A.)Scanning and Module Creation

1. Open Tosca Commander.

2. Create a new folder named Login.

3. Navigate to Scan → Scan Application, and select the Swag Labs login page.

4. Add attributes:

o Username

o Password

o Login button

5. Save and close the scan, and name the module as LoginPage.

B.) Creating Test Case

6. Create a new Test Case folder named TestCaseDriven.

7. Inside it, create a test case named LoginPageTest.

8. Drag and drop the LoginPage module into the test case.

9. Add two built-in modules:

o Open URL (set to https://www.saucedemo.com)

o Close Browser (title = Swag*)

10. Arrange the steps as:

0. Open URL

1. LoginPage (Username, Password, Login)

2. Close Browser

C.) Convert Test Case to Template

11. Right-click on the test case LoginPageTest → Select Convert to Template.

12. Ensure the test case icon changes to show it is now a template.

D.) Creating TestCase-Design Sheet

13. Navigate to TestCase-Design tab.

14. Right-click → Create TestSheet → Name it as LoginTestSheet.

Add Instances:

15. Right-click on LoginTestSheet → Click on I (Create Instance) → Create 3 instances.

Add Attributes:

16. Right-click on the sheet → Select O (Create Attribute):

o Attribute 1: Username

o Attribute 2: Password

Provide Values:

17. Under each attribute, right-click to create values:

Username Values:

1. standard_user

2. problem_user

3. performance_glitch_user

Password Value:

o secret_sauce (used commonly for all)

Map Instances to Values:

18. Click on each instance column and assign username and password values appropriately using dropdowns.

E.)Mapping Data Sheet to Template

19. Drag and drop the LoginTestSheet over the LoginPageTest (template).

20. Right-click on template → Select Check Template (no errors should be shown).

21. Right-click → Select Jump to Schema Definition to verify the mapping.

F.) Map Attribute Values

22. In the template test case, open the LoginPage module step.

23. Drag the Username attribute from the TestSheet and drop it onto the username field.

o Confirm ExcelReference appears as Excel:Username

24. Similarly, drag the Password attribute to the password field.

G.)Generate Test Cases

25. Right-click on the LoginPageTest (template).

H.)Execute Test Cases

Program 8:

Step 1: Create a Module Folder

Step 2: Scan the Application

2. Open the obstacle scenario page (Obstacle No: 92248) in a browser.

3. In XScan, switch to Advanced Mode.

4. Select the entire table that contains user data (not individual cells).

5. Also select any one Edit button from the table.

Step 3: Organize Scanned Elements

1. In the scanned module, drag and drop the selected Edit button into the corresponding table row where user details are displayed.

2. Ensure the hierarchy is: Table → Row → Edit Button.

Step 4: Create a Test Case and Drag the scanned module into the test case.

Step 5: Set Constraints to Identify the Dynamic Row

1. In the test case, locate the row object under the table.

2. Right-click the First Name cell and change Action Mode to Constraint.

3. Enter the Value as John.

Step 6: Add Additional Constraint for Unique Identification

1. Right-click the Last Name cell and set Action Mode to Constraint.

2. Enter the Value as Doe.

Now TOSCA will identify the row where First Name = John and Last Name = Doe.

Step 7: Execute the Test Case

1. Run the test case in ScratchBook.

2. Observe that the Edit button next to "John Doe" is clicked, regardless of row position.

Step 8: Validate Dynamic Handling

1. Refresh the obstacle page multiple times (observe row positions change).

2. Each time, re-run the test case in ScratchBook.

3. Verify that TOSCA consistently identifies the correct row using the constraints and clicks the right Edit button.

program 9: Part A: Static Synchronization using TBox Wait Module

2. Create a new Module named Wait.

o Add "Calculate" and "Send" buttons to the module.

o Finish and close the scanner.

4. Create a new Test Case and name it StaticWaitTest.

5. Drag and drop the created module into the test case.

6. Add steps as follows:

o Step 1: Click on the Calculate button.

o Step 2: Insert a TBox Module TBox Wait On.

Duration: Set a value such as 20,000 ms (or your preferred wait time).

o Step 3: Click on the Send button (note: this step will execute regardless of the button's state).

o Navigate to Project > Settings > TestConfiguration > TBox > Synchronization:

Synchronization.DefaultTimeout = 10,000 ms (default for finding elements).

Synchronization.WaitOn = 20,000 ms (default max time for wait on actions).

You may increase this value if needed.

8. Execute the Test Case (StaticWaitTest) in ScratchBook:

Part B: Dynamic Synchronization using Wait On Action Mode

1. Create another Test Case and name it DynamicWaitTest.

2. Drag and drop the same module containing Calculate and Send buttons.

3. Step 1: Click on Calculate button (same as before).

4. Step 2: For the Send button:

o Change the ActionMode to WaitOn.

o Set the Condition: Enabled = True.

5. Step 3: Copy and create a new step to Click the Send button using Input ActionMode.

6. Execute the Test Case (DynamicWaitTest) in ScratchBook:

o Observe that the test waits until the Send button is actually enabled (progress bar completes).

o Once the condition is met, the button is clicked immediately.

program 10: Step 1: Identify Reusable Steps

1. Choose a common scenario such as login functionality.

2. Identify the repetitive steps:

o Open the website

o Enter username

o Enter password

o Click Login

Step 2: Create a Reusable Module

3. Open TOSCA Commander.

4. Navigate to the Modules section and click Create Module.

5. Name it as reuse or any other relevant name.

6. Record the following steps using Scan:

o Enter Username

o Enter Password

o Click Login

7. Save and close the module.

Step 3: Create a Test Case Using the Module

8. Go to TestCases section and create a new test case named Login TestCase.

9. Drag the module created in Step 2 into this test case.

10. Optionally rename the module instance to Login Page.

11. Create or reuse an existing Open URL module and add it to the test case to open the

SwagLabs site.

Step 4: Create a Test Step Library Folder

12. Right-click on the workspace and select Create TestStepLibrary Folder.

13. Move both the Open URL and Login Page modules into this library folder.

14. Verify that the moved modules show an arrow icon indicating library reference.

Step 5: Parameterize the Reusable Steps

15. Right-click on the Open URL step in the library folder → Select Create Business Parameter (P icon).

16. Right-click on the created Business Parameter → Select Create Parameter and name it URL.

17. Drag and drop the URL parameter to the actual URL value field.

18. Repeat the same for the Login Page:

• Create Business Parameters for Username and Password.

• Drag and drop each parameter to the respective fields in the module.

Step 6: Use Reusable Steps in New Test Cases

19. Create a new test case (e.g., Login 1).

20. Drag and drop the Open URL and Login Page blocks from the library.

21. Enter values for:

o URL: https://www.saucedemo.com

o Username: standard_user

o Password: secret_sauce

22. Execute the test case and verify successful login.

Step 7: Repeat for Other Test Scenarios

23. Create new test cases (e.g., Login 2, Login 3 for locked_out_user, problem_user, etc.).

24. Drag and drop the reusable steps again from the library.

25. Change only the input values (username and password) as required.

program 11: Step 1: Launch and Scan the Application

1. Open Tosca Commander. 2. Create a new folder named ConditionalStatements.

3. Start scanning the Swag Labs application.

4. On the products page, scan:

o "Add to cart" button for Sauce Labs Backpack o"Add to cart" button for Sauce Labs Bike Light

5. Since both buttons have the same label, use Identify by Anchor:

o Drag the Sauce Labs Backpack label as the anchor for the first button.

o Drag the Sauce Labs Bike Light label as the anchor for the second button.

6. Click Finish Screen and close the scanner

Step 2: Create a Test Case

1. Create a new Test Case under ConditionalStatements.

2. Drag the scanned module into the test case twice:

o First instance for Backpack

o Second instance for Bike Light

3. Set both step values to Click.

4. Run the test once to confirm both items are added when available.

Step 3: Introduce Conditional Logic

1. Right-click on the test case and select Create If Statement.

2. An "If" block with Condition and Then branches will appear.

3. Drag the first product test step inside the Then section.

4. Click the blue arrow next to Condition, and:

o Change ActionMode to Verify o Set Property to Enabled *Set Property to Enabled

o Set Value to True

5. This means: If the "Add to cart" button for Sauce Labs Backpack is enabled, then click it.

Step 4: Add Fallback Step

1. Outside the If block (after the condition), add the test step for Bike Light.

2. This step will always run regardless of the condition.3.This ensures that if the first product is already in cart, the second product will still be added.