



SOFTWARE TESTING AND QUALITY ASSURANCE

EXPERIMENT 9

NAME: Shashwat Shah

SAP ID: 60004220126

BRANCH: COMPUTER ENGINEERING

BATCH: C22

Aim: To study software Automation Testing tool WinRunner for Checking GUI Objects

Theory:

WinRunner is a functional testing tool developed by **Mercury Interactive** (now owned by **Micro Focus**). It is used to automate the testing of software applications by recording and replaying user interactions with the GUI. WinRunner helps testers perform regression testing, detect defects, and ensure the stability of software products.

Key Features of WinRunner:

- 1. GUI Testing:**
WinRunner allows testers to check the presence, properties, and behavior of graphical user interface (GUI) objects like buttons, text fields, checkboxes, etc.
- 2. Test Script Language (TSL):**
It uses a scripting language called TSL to create automated test scripts that can be customized and reused.
- 3. Record and Playback:**
WinRunner provides both **context-sensitive** and **analog recording** modes to capture user actions for automation.
- 4. Object Repository:**
It stores GUI object properties in a **GUI Map**, allowing testers to reference objects by logical names in scripts.
- 5. Synchronization:**
WinRunner supports synchronization points to handle application delays, ensuring scripts wait for the expected conditions.
- 6. Verification Points:**
It includes built-in functions for checking GUI object properties, such as whether a button exists, is enabled, or displays the correct label.



Importance of GUI Object Testing in WinRunner:

- Ensures all visible elements of the application work as intended.
- Automates repetitive UI testing tasks to save time and effort.
- Detects layout, property, and functionality issues early in development.

Code:

1. Login page

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Page</title>
</head>
<body>
  <h1>Login to Your Account</h1>

  <!-- Login Form -->
  <form id="login-form">
    <!-- Username input field -->
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" placeholder="Enter
your username" required><br><br>

    <!-- Password input field -->
    <label for="password">Password:</label>
    <input type="password" id="password" name="password"
placeholder="Enter your password" required><br><br>

    <!-- Submit Button -->
    <button type="submit" id="submit-button">Login</button>
  </form>
</body>
</html>
```



2. GUI test page

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
import time

# Set up the WebDriver
service = Service("D:\\hi\\STQA\\exp_9\\chromedriver-
win64\\chromedriver.exe") # Use double backslashes or raw string
driver = webdriver.Chrome(service=service)

# Open the login page
driver.get("file:///D:/hi/STQA/exp_9/login.html") # Use file:// for local
HTML files

# Maximize the window
driver.maximize_window()

# GUI Object Checks
try:
    # Check if the heading is present
    heading = driver.find_element(By.TAG_NAME, "h1")
    print("Heading found:", heading.text)
    time.sleep(2) # Wait to screenshot heading

    # Check if the submit button exists and is clickable
    button = driver.find_element(By.ID, "submit-button")
    if button.is_displayed() and button.is_enabled():
        print("Submit button is visible and clickable.")
    else:
        print("Submit button not available.")
    time.sleep(2) # Wait to screenshot button state

    # Enter username
    username_textbox = driver.find_element(By.NAME, "username")
    username_textbox.send_keys("test_user")
    print("Username entered successfully.")
    time.sleep(4) # Wait to screenshot username

    # Enter password
    password_textbox = driver.find_element(By.NAME, "password")
    password_textbox.send_keys("test_password")
    print("Password entered successfully.")
    time.sleep(4) # Wait to screenshot password

    # Click the submit button
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



```
button.click()
print("Submit button clicked.")
time.sleep(5) # Wait to screenshot post-click

except Exception as e:
    print("Error:", e)

# Final pause before closing the browser
time.sleep(3)
driver.quit()
```

Output:

Login to Your Account

Username:

Password: