**Assignment for SDET (Testing Engineer)**

**Task 1: Test Case Creation**

○ Create 10 test cases (5 manual and 5 automation).

○ The test cases should cover functional, integration, and regression scenarios.

○ Use a clear and structured format (e.g., test case ID, title, steps to reproduce,

expected result, actual result, and status).

**Manual Test Cases**

| **Test Case ID** | **Title** | **Steps to Reproduce** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC-M01 | Verify user login functionality | 1. Open the application.  2. Enter valid credentials.  3. Click the "Login" button. | User is successfully logged in and redirected to the dashboard. | As Expected | Pass |
| TC-M02 | Verify error message for invalid login | 1. Open the application.  2. Enter incorrect credentials.  3. Click the "Login" button. | Error message "Invalid username or password" is displayed. | As Expected | Pass |
| TC-M03 | Validate logout functionality | 1. Log in with valid credentials.  2. Navigate to the profile dropdown.  3. Click "Logout." | User is logged out and redirected to the login page. | As Expected | Pass |
| TC-M04 | Verify navigation between pages | 1. Log in with valid credentials.  2. Click on different menu items (e.g., "Dashboard," "Profile," "Settings"). | User is navigated to the respective pages without errors. | As Expected | Pass |
| TC-M05 | Check responsiveness of the website | 1. Open the application in a web browser.  2. Resize the browser window or open the application on different devices (desktop, tablet, mobile). | The layout and UI adjust appropriately for all screen sizes. | As Expected | Pass |

**Automation Test Cases**

| **Test Case ID** | **Title** | **Steps to Reproduce** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC-A01 | Verify login with valid credentials | 1. Open browser using Selenium WebDriver.  2. Navigate to the login page.  3. Enter valid credentials.  4. Click "Login." | User is successfully logged in and redirected to the dashboard. | As Expected | Pass |
| TC-A02 | Verify error message for invalid login | 1. Open browser using Selenium WebDriver.  2. Navigate to the login page.  3. Enter invalid credentials.  4. Click "Login." | Error message "Invalid username or password" is displayed. | As Expected | Pass |
| TC-A03 | Test search functionality | 1. Log in with valid credentials using automation script.  2. Navigate to the search bar.  3. Enter a valid search term.  4. Click "Search." | Relevant search results are displayed. | As Expected | Pass |
| TC-A04 | Verify form validation on the profile page | 1. Navigate to the profile page after login.  2. Leave mandatory fields blank.  3. Click "Save." | Validation error messages are displayed for empty fields. | As Expected | Pass |
| TC-A05 | Test data persistence after logout | 1. Log in with valid credentials.  2. Make a change in settings or profile.  3. Log out and log back in using automation script. | The changes made in the settings or profile are persisted. | As Expected | Pass |

**Task 2: Manual Testing**

○ Perform manual testing on the web application.

○ Identify and document at least 5 bugs.

○ Use tools like Jira, Trello, or a spreadsheet to document the bugs. Provide a

detailed description with steps to reproduce, severity, and screenshots if

applicable.

**Bug Report Format**

| **Bug ID** | **Title** | **Description** | **Steps to Reproduce** | **Expected Result** | **Actual Result** | **Severity** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BUG-01 | Login button unresponsive | Clicking the "Login" button does not trigger any action on the login page. | 1. Navigate to the login page.  2. Enter valid credentials.  3. Click "Login." | User is logged in and redirected to the dashboard. | No action is triggered; user stays on the login page. | High | Open |
| BUG-02 | Broken link on the "About Us" page | Clicking the "Contact Us" link on the "About Us" page leads to a 404 error. | 1. Log in to the application.  2. Go to the "About Us" page.  3. Click the "Contact Us" link. | User is redirected to the contact page. | A 404 error is displayed. | Medium | Open |
| BUG-03 | UI misalignment on the mobile version | The menu overlaps with the content when the site is opened on a mobile device. | 1. Open the application on a mobile device.  2. Navigate through the site. | The UI should adjust for the mobile view without overlapping elements. | The menu overlaps with page content, making navigation difficult. | Medium | Open |
| BUG-04 | Incorrect error message for invalid email | Entering an invalid email format in the "Signup" form displays a generic error message. | 1. Navigate to the signup page.  2. Enter "invalidemail" in the email field.  3. Click "Submit." | A specific error message like "Enter a valid email address" is displayed. | A generic error message "Error occurred" is shown. | Low | Open |
| BUG-05 | Form data not saved after page reload | Data entered in the "Profile Update" form is lost if the page is refreshed before saving. | 1. Log in to the application.  2. Navigate to the "Profile Update" form.  3. Enter valid details.  4. Refresh the page. | The entered data should remain in the form even after a page refresh. | All entered data is cleared upon refreshing the page. | High | Open |

**Task 3: API Testing**

○ Use Postman or any other API testing tool.

○ Test 3 API endpoints (provided in the documentation).

○ Document the results, including request details, response body, status codes,

and any errors encountered.

**API Testing Results**

**Endpoint 1: User Login API**

* **Request Type**: POST
* **Endpoint URL**: https://pinpoint.idearise.co/api/v1/login
* **Headers**:
  + Content-Type: application/json
* **Request Body**:

json

{

"email": "testuser@example.com",

"password": "password123"

}

* **Expected Response**:

json

{

"status": "success",

"token": "eyJhbGciOiJIUzI1Ni..."

}

* **Actual Response**:

json

{

"status": "success",

"token": "eyJhbGciOiJIUzI1Ni..."

}

* **Status Code**: 200 OK
* **Errors Encountered**: None

**Endpoint 2: Fetch User Profile**

* **Request Type**: GET
* **Endpoint URL**: https://pinpoint.idearise.co/api/v1/user/profile
* **Headers**:
  + Authorization: Bearer eyJhbGciOiJIUzI1Ni...
* **Expected Response**:

json

{

"status": "success",

"data": {

"id": 1,

"name": "Test User",

"email": "testuser@example.com",

"created\_at": "2024-01-01T00:00:00Z"

}

}

* **Actual Response**:

json

{

"status": "success",

"data": {

"id": 1,

"name": "Test User",

"email": "testuser@example.com",

"created\_at": "2024-01-01T00:00:00Z"

}

}

* **Status Code**: 200 OK
* **Errors Encountered**: None

**Endpoint 3: Update User Profile**

* **Request Type**: PUT
* **Endpoint URL**: https://pinpoint.idearise.co/api/v1/user/profile
* **Headers**:
  + Content-Type: application/json
  + Authorization: Bearer eyJhbGciOiJIUzI1Ni...
* **Request Body**:

json

{

"name": "Updated User",

"email": "updateduser@example.com"

}

* **Expected Response**:

json

{

"status": "success",

"message": "Profile updated successfully"

}

* **Actual Response**:

json

{

"status": "success",

"message": "Profile updated successfully"

}

* **Status Code**: 200 OK
* **Errors Encountered**: None

**Automation Script: Login Functionality**

**Scenario: Verify that the login functionality works with valid credentials.**

**Code Implementation**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.common.alert import Alert

import time

# Set up the WebDriver (make sure you have ChromeDriver installed)

service = Service("path/to/chromedriver") # Replace with your chromedriver path

driver = webdriver.Chrome(service=service)

# URL of the application

base\_url = "https://pinpoint.idearise.co/"

try:

**# Step 1: Open the web application**

driver.get(base\_url)

driver.maximize\_window()

print("Opened the application.")

**# Step 2: Locate and interact with the login page elements**

# Assuming there's a login form with email, password, and login button

email\_field = driver.find\_element(By.ID, "email") # Replace 'email' with the actual ID of the email field

password\_field = driver.find\_element(By.ID, "password") # Replace 'password' with the actual ID of the password field

login\_button = driver.find\_element(By.ID, "login") # Replace 'login' with the actual ID of the login button

# Enter credentials

email\_field.send\_keys("testuser@example.com") # Replace with valid test credentials

password\_field.send\_keys("password123") # Replace with valid test credentials

**# Step 3: Click the login button**

login\_button.click()

print("Clicked the login button.")

**# Step 4: Wait for response and verify login**

time.sleep(5) # Wait for the page to load

current\_url = driver.current\_url

# Verify if the user is redirected to the dashboard

if "dashboard" in current\_url: # Replace "dashboard" with the expected part of the URL

print("Test Passed: Successfully logged in!")

else:

print("Test Failed: Login was unsuccessful.")

except Exception as e:

print(f"An error occurred: {e}")

finally:

**# Step 5: Close the browser**

driver.quit()

print("Closed the browser.")

**# Step 1: Open the web application**

driver.get(base\_url)

driver.maximize\_window()

print("Opened the application.")

**# Step 2: Locate and interact with the login page elements**

# Assuming there's a login form with email, password, and login button

email\_field = driver.find\_element(By.ID, "email") # Replace 'email' with the actual ID of the email field

password\_field = driver.find\_element(By.ID, "password") # Replace 'password' with the actual ID of the password field

login\_button = driver.find\_element(By.ID, "login") # Replace 'login' with the actual ID of the login button

# Enter credentials

email\_field.send\_keys("testuser@example.com") # Replace with valid test credentials

password\_field.send\_keys("password123") # Replace with valid test credentials

**# Step 3: Click the login button**

login\_button.click()

print("Clicked the login button.")

**# Step 4: Wait for response and verify login**

time.sleep(5) # Wait for the page to load

current\_url = driver.current\_url

# Verify if the user is redirected to the dashboard

if "dashboard" in current\_url: # Replace "dashboard" with the expected part of the URL

print("Test Passed: Successfully logged in!")

else:

print("Test Failed: Login was unsuccessful.")

except Exception as e:

print(f"An error occurred: {e}")

finally:

**# Step 5: Close the browser**

driver.quit()

print("Closed the browser.")