TEST CASE STEPS AND SUMMARY

1

def test_login ():

1. The function navigates to the login page of the web application.

2. It enters the login credentials (email and password) into the respective input fields.

3. The code clicks on the login button.

4. After a brief delay for the page to load, it compares the current URL with the expected URL of the home page.

5. If the URLs match, the login test is considered successful and the function returns "Passed". Otherwise, it returns "Failed".

2

def test_logout ():

1. The code represents a test case for logging out of a browser application.

2. It uses XPath expressions to locate and interact with specific elements on the web page, such as the "Log Out" link and the "Ok" button.

3. By simulating click actions on these elements, the code automates the logout process for testing purposes.

3

def test_dashboard ():

1. Find and click on the "Analytics and Reports" element on the webpage.

2. Select the "AUTO" option from the first dropdown menu.

3. Select the "CLASS 1" option from the second dropdown menu.

4. Select the "B" option from the third dropdown menu.

5. Click on the "Download" element.

6. Set the download directory path to "C:\\Users\\rupes\\Downloads" and define the file name as "FeesNotPaid.xlsx".

7. Wait until the file with the specified name is downloaded to the download directory.

8. Return a Boolean indicating whether the downloaded file exists in the download directory.

4

def rprt_feesnotpaid_usingbus ():

1. Find and click on the "Analytics and Reports" element on the webpage.

2. Click on the first dropdown menu and select the option "AUTO".

3. Click on the second dropdown menu and select the option "CLASS 1".

4. Click on the third dropdown menu and select the option "B".

5. Set the download directory path to "C:\\Users\\rupes\\Downloads".

6. Get the initial count of files in the download directory.

7. Click on an element within the webpage.

8. Pause the execution for 4 seconds.

9. Get the updated count of files in the download directory.

10. If the updated count is the same as the initial count, return "Failed".

11. If the updated count is greater than the initial count, return "Passed".

5

def get_student_by_name(student_name):

1. Sleep for 4 seconds.

2. Click on the "Manage Student" element on the webpage.

3. Clear the "nameorId" input field.

4. Enter the provided student name into the input field.

5. Get the number of rows as text from a specific element.

6. Sleep for 10 seconds.

7. Extract the last character from the text and convert it to an integer, assigning it to "t".

8. If "t" is greater than 0, return "Passed"; otherwise, return "Failed".

6

def add_student ():

1. Open the browser and go to "https://testbams.web.app/dashboard/managestudent".

2. Wait for 10 seconds.

3. Click on "Manage Student".

4. Click on "Add student using Form".

5. Fill in the admission number, first name, last name, father's name, and select the class.

6. Provide the student's gender, mobile number, bus ID, vehicle type, date of birth, and select other relevant options.

7. Enter the student's address, student ID, admission date, mother's name, and mother's mobile number.

8. Specify the hostel or day scholar status.

9. Click on "Add Student Details".

10. Click on "Confirm Add student Details".

11. Validate the details of the added student using the "get_student_by_name" function with the name "Yuvraj Patare".

7

def update_info(student_name,admission_no):

1. Call the "get_student_by_name" function with the argument "Yuvraj Patare".

2. Click on the admission number

3. Click on the "Edit Details" button.

4. Clear the "admissionNo" input field.

5. Enter the value of the "admission_no" variable into the "admissionNo" input field.

6. Click on the "Confirm Add Student" button.

7. Navigate to the URL "https://test-bams.web.app/dashboard/managestudent".

8. Wait for 10 seconds.

9. Click on the "Manage Student" element.

10. Enter the value "student_name" into the "nameorId" input field, clear it again, then call the "get_student_by_name" function with the argument "Yuvraj Patare" once more.

11. Retrieve the text from a button element(admission no) and convert it to an integer, storing it in the variable "get_value_after_update".

12. Check if the value of "get_value_after_update" is equal to the value of "admission_no".

13. If they are equal, print "Passed"; otherwise, print "Failed".

8

def manage_licence_update_superviser_info():

1. Click on "Manage Licenses" on the webpage.

2. Select the "AUTO" option from a dropdown menu.

3. Click a Emp-id-Number of selected person.

4. Click on "Edit Details".

5. Enter "Yuvrajs" in an input field.

6. Enter "Patare" in another input field.

7. Enter "YPBandhu125" in a third input field.

8. Click on "Update License".

9. Navigate to a specific URL.

10. Wait for 15 seconds.

11. Click on "Manage Licenses" again.

12. Enter ID ("YPBandhu125") in an input field.

13. Retrieve the emp-id, name

16. validate the Retrieve information with input information

9

def view_qr ():

1. Calls a function `get_student_by_name` with the argument "Mishika Luthra" to retrieve student information.

2. Clicks on a button located in a specific element on the webpage.

3. Retrieves the source attribute of an image located in a specific element on the webpage.

4. Checks if the length of the source is greater than 1.

5. If the length of the source is greater than 1, clicks on the button again and retrieves the source attribute of the image

6. Compares the lengths of the two sources.

7. If the length of the first source is less than or equal to the length of the new source, returns "passed".

8. If the length of the first source is greater than the length of the new source, returns "failed".

9. If the length of the source is not greater than 1, returns "passed".

10

def download_qr():

1. Calls a function `get_student_by_name` with the argument "Mishika Luthra"

2. Clicks on a check box of searched person

3. Click on download QR button

4. Validation :

 a] extract the latest zip file where assumed that folder must be empty before download

 b] then iterate on current folder check whether png is present in that or not

11

def test_table_dashboard():

1] select date on dashboard

2] then Retrieve then information of present, absent, total_student

3] if sum of total_student is equal to the present + absent then return paased else return failed