



Preliminary Assessment for SDET

Report on Automation Testing and Programming Challenges

Prepared by:

Shivani Patel

Task1 – Automation testing using Selenium

Please find the java-based Selenium program for this task.

Software used: Eclipse IDE 2022-06 (4.24.0), Selenium-Java 4.3.0, testng 7.6.1

User Instructions:

1. Download the zip folder for project named “SeleniumProject1” from GitHub (Link to download the project files: https://github.com/Shivii111/PlanIT-Preliminary-Test_Shivani-Patel.git)
2. Unzip the folder and extract the project files
3. Open Eclipse IDE in your computer.
4. Import the downloaded project (SeleniumProject1)
5. Run the selected project
6. Check the results for automated testcases as mentioned below

Chosen 10 test cases for the given website are as follows.

Testcase 1

- **Test Scenario:** To test an incorrect user login on <https://jupiter.cloud.planittesting.com/#/>
- **Test Steps:**
 - Launch Chrome Browser
 - Navigate to the *jupiter.cloud.planittesting.com* URL.
 - Click on login tab
 - Enter invalid credentials
- **Browser:** Google Chrome
- **Test Data:** Incorrect credentials
- **Expected/Intended Results:** Once incorrect credentials are entered onto the login page, system should throw validation message for incorrect credentials
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 2

- **Test Scenario:** Perform product buying scenario
- **Test Steps:**
 - Click start shopping button on home page
 - Click on buy button for the items that you want to buy
 - Click on the cart tab
 - Click on checkout button
 - Fill the checkout detail form with appropriate data
 - Click on submit button
- **Browser:** Google Chrome
- **Test Data:** Contact details and payment details
- **Expected/Intended Results:** Once all the steps are followed to buy a product and submit button is clicked, system should generate a notification message for successful order placement with order number.
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 3

- **Test Scenario:** To check the functionality of navigation bar's tabs (home, shop, and contact) along with the logo icon
- **Test Steps:**
 - Click contact tab from home page
 - Click home tab from contact page
 - Click shop tab from home page
 - Click on logo icon (Jupiter Toys) from shop page
- **Browser:** Google Chrome
- **Test Data:** N/A
- **Expected/Intended Results:** As it clicks on each navigation tabs, the system should redirect it to an appropriate webpage. And as it clicks on logo icon, system should redirect it on homepage
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 4

- **Test Scenario:** Contact tab functionality testing → Fill the contact details and click on submit button should display the thank you message on screen
- **Test Steps:**
 - Click contact tab from home page
 - Fill the contact detail form with appropriate details
 - Click submit button
- **Browser:** Google Chrome
- **Test Data:** Contact details and message
- **Expected/Intended Results:** As it fills the form and clicks submit button, system should generate thank you message
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 5

- **Test Scenario:** Check the functionality of empty cart button
- **Test Steps:**
 - Click start shopping button on home page
 - Click on buy button for the items that you want to buy
 - Click on the cart tab
 - Click on empty cart button
 - Select No from empty cart confirmation alert
 - Click on empty cart button again
 - Now, Select Yes from empty cart confirmation alert
- **Browser:** Google Chrome
- **Test Data:** N/A
- **Expected/Intended Results:** If it clicks NO to empty cart confirmation pop-up, then system should redirect it to cart webpage. If it selects Yes option, then system should empty the cart and display the empty cart message on the screen
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 6

- **Test Scenario:** Can user modify the purchased items from the cart and does system update the price accordingly?
- **Test Steps:**
 - Click start shopping button on home page
 - Click on buy button for the items that you want to buy
 - Click on the cart tab
 - Increase the amount of quantity of items to buy
- **Browser:** Google Chrome
- **Test Data:** Quantity of items to buy
- **Expected/Intended Results:** If it increases the quantity of items to buy in the cart, system should update the total price value accordingly
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 7

- **Test Scenario:** Check the functionality of remove item button (RED button on left hand side of an item in the cart)
- **Test Steps:**
 - Click start shopping button on home page
 - Click on buy button for 3 items that you want to buy
 - Click on the cart tab
 - Click on remove item button for one item
 - Select YES from remove item confirmation alert
- **Browser:** Google Chrome
- **Test Data:** N/A
- **Expected/Intended Results:** Once it clicks on remove item button and select YES from remove item confirmation alert, system should remove the item from the list in the cart
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 8

- **Test Scenario:** Check validation for empty fields in checkout form
- **Test Steps:**
 - Click start shopping button on home page
 - Click on buy button for items that you want to buy
 - Click on the cart tab
 - Click on checkout button
 - Leave the fields empty in checkout detail form and click submit button
- **Browser:** Google Chrome
- **Test Data:** N/A
- **Expected/Intended Results:** Once it clicks on submit button with empty fields in the checkout detail form, system should generate validation alert
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 9

- **Test Scenario:** Check validation for empty fields in contact detail form
- **Test Steps:**
 - Click contact tab from home page
 - Leave the fields empty in contact detail form and click submit button
- **Browser:** Google Chrome
- **Test Data:** N/A
- **Expected/Intended Results:** Once it clicks on submit button with empty fields in contact detail form, system should generate validation alert
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Testcase 10

- **Test Scenario:** Check validation for incorrect input in the fields in a contact detail form
- **Test Steps:**
 - Click contact tab from home page
 - Enter inappropriate values in the fields in contact detail form and click submit button
- **Browser:** Google Chrome
- **Test Data:** N/A
- **Expected/Intended Results:** Once it clicks on submit button with inappropriate input in the fields in a contact detail form, system should generate validation alert
- **Actual Results:** As Expected
- **Test Status:** Pass/Fail: Pass

Task 2 Programming Challenge

Following programming challenges have been chosen to perform.

Challenge 2:

Write a solution to find the character that has the highest number of occurrences within a certain string, ignoring case. If there is more than one character with equal highest occurrences, return the character that appeared first within the string.

For example:

Input: "Character"

Output: c

Solution:

Please find the java-based program which provides desired results for this challenge.

Software used: Netbeans IDE 8.2, Java\jdk1.8.0_202

User Instructions:

1. Download the zip folder for project named “CharacterOccurrences” from GitHub (Link to download the project files: https://github.com/Shivii111/PlanIT-Preliminary-Test_Shivani-Patel.git)
2. Unzip the folder and extract the project files
3. Open NetBeans IDE in your computer.
4. Import the downloaded project (CharacterOccurrences)
5. Run the selected project

Challenge 4:

Build a food ordering system where the user inputs the food name into the food order and then the order gets

passed to the appropriate restaurant on a list and the restaurant prints the receipt on stdout. No need to create a

UI, a main function that receives parameters is enough.

For example:

Input: “pepperoni pizza”

Output: "Awesome pizza place, pepperoni pizza, \$20”

For example:

Input: “burger”

Output: “wild burger joint, burger, \$15”

Solution:

Please find the java-based program which provides desired results for this challenge.

Software used: Netbeans IDE 8.2, Java\jdk1.8.0_202

User Instructions:

1. Download the zip folder for project named “FoodOrderingSystem” from GitHub (Link to download the project files: https://github.com/Shivii111/PlanIT-Preliminary-Test_Shivani-Patel.git)
2. Unzip the folder and extract the project files
3. Open NetBeans IDE in your computer.
4. Import the downloaded project (FoodOrderingSystem)
5. Run the selected project

----- End of document-----