**Read Me File for ECS- Digital Test Automation.**

**Test Scenarios:**

1. Valid Submission :

**Automated**, this test is covered in src/test/ECSRequirement.java This test is just automated as per the requirement from ECS.

1. Invalid Submission:

This scenario covers both positive and negative scenarios and is **partly automated** (details below), this test is covered in src/test/ValidAndInvaidSubmission.java

a) Blank fields validation:

**Automated**. This covers name, email and message fields in the contact us page.

b) Invalid email ID validation:

**Automated** – This covers invalid email formats.

c) Validate maximum length of text fields:

**Not Automated**.

d) Test send button (email acknowledgement):

**Automated** to check for the confirmation message displayed in the UI. The successful submission of details results in a database entry and hence, the database entry or the email server is **not automated**.

e) Test Call Us button:

**Not Automated.**

f) Test Email Us button:

**Not Automated.**

g) All the text in the contact us page (for spelling and grammar)

**Not Automated.**

1. Cross Browser Testing – **Not Automated.** This is not automated because there is an issue with gecko driver in my laptop (Ubuntu platform).
2. Mobile Testing – **Not Automated.**

***Instructions to build and run:***

1. *Link to the git repo -*
2. *Import the code into Eclipse (or similar IDE) and build and Run (Before executing check the platform limitations section on chrome driver details.)*

***Platform limitations:***

*None. However, the path for the chrome driver is provided in src/lib/BrowserFunctions.java*

*This needs to be changed to point to the correct location in the machine where the test is executed.*

***Dependencies:***

1. *The below external Jar files need to be imported in Eclipse project.*
2. *Selenium WebDriver*
3. *Junit*
4. *Cucumber*
5. *JRE, JDK*
6. *Chrome driver (exe file) and chrome browser*

***Improvements:***

1. *Currently the test for checking the valid submission is done based on the alert message received after a failure to send the contact details (THERE WAS AN ERROR TRYING TO SEND YOUR MESSAGE. PLEASE TRY AGAIN LATER.) I believe this test should be based on the successful message in the UI. Since the Qaworks website was not able to send the details (maybe the email server is down), and failed with an error message, I used the failure message.*
2. *I have used system.out.println in my code, this is because I am assuming the regression pack will be integrated with jenkins and will be executed as a over night build. When this happens, all a automated tester has to do is check the console messages in jenkins if a automatic mail is triggered for failure. If this is setup with jenkins is not present, I can change these to log messages or something similar to fit in to the available frameworks within the organization.*
3. *I have also not used assert function in any of the tests because, the assert function will break the execution when there is a failure. Again if there is a requirement to use assert, I can modify the code.*

***Observations:***

1. *In the name field a user can enter only numeric values, I am not sure if this a part of the requirements. If not then this is a defect.*
2. *When there is a failure to send the contact details, an alert message is displayed, which is correct. However, the font colour is not bright enough to read it quickly.*
3. *In the Contact us page, there is no captcha and hence, anyone can automate stimulating a significant number of users and bring the server down. I am not sure if there is a validation for a user with same subject or message to send more than a few times.*

***Folder structure in Git:***

1. ***src:***

*This folder contains all the codes and tests except for the feature files.*

1. *core:*

*This folder contains the test runner. This is the starting point for this project.*

1. ***lib:***

*This folder contains alll the library functions. There are a few common functions (like navigating to different menus within qaworks and a few browser functions like navigating, refreshing, closing) in this folder which can be used by other teams/tests.*

1. ***pageObjects:***

*This folder contains all the page objects for the contact us page that are required by other tests.*

1. ***test:***

*This folder contains the valid and invalid tests.*

1. ***Features:***

*This folder contains the feature files.*