**SAPIA Chat Bot Test Scenarios**

1. Test a Happy Path:

**Description:**

Test the chatbot's ability to handle a successful conversation and provide correct responses to user queries.

**Steps:**

* Enter First Name and Last Name to start the conversation.
* Ask a series of valid questions related to the chatbot's domain.
* Verify that the chatbot responds appropriately and provides correct answers.

1. Test for Invalid Inputs:

**Description**:

Test how the chatbot handles invalid or unexpected user inputs.

**Steps**:

* Enter random characters or symbols as input.
* Verify that the chatbot responds with an error message or a message indicating that the input is invalid.

**Example1:** Provide the email address without domain name, it should display the appropriate error message to the user

Provide the input email address as **example** instead of [**example@gmail.com**](mailto:example@gmail.com)

**Example2:** Provide invalid mobile number, verify that chat bot it should displays the appropriate error message to the user

1. Test for Free Text Questions:

**Description**:

Test a failed path when typing less than 50 words for any free text question, you need to see the pop up window

**Steps**:

* Provide the less than 50 words to the free text questions that require longer answers.
* Verify that the chatbot should display the appropriate pop-up window along with **EDIT** and **Continue** buttons in pop up window

1. Test Scenario: Edit Given Answer

**Description**:

Test the chatbot's ability to allow the user to edit their given answer.

**Steps**:

* User Input: Enter the First name and last name for bot question
* Trying to click on edit option on given answer
* Trying to change the “First Name” and “Last Name”

**Expected Result**:

The chatbot should allow the user to edit their given answer and update the response accordingly.

1. Test Scenario: Emoji Options Interaction

**Description**:

Test the chatbot's ability to provide emoji options to users and handle their selections.

**Steps**:

* User clicks on emoji options
* User should able to see the emoji option to use the emoji’s

**Expected Result**:

* The chatbot should present emoji options to the user after the initial greeting.
* The user should be able to click on an emoji option to indicate their mood or feeling.

**Feedback on chatbot application:**

1. Positive Feedback: positive feedback for aspects of the chatbot that worked well and provided a good user experience.
2. Suggestions for Improvement:
3. One observation during the testing the chatbot application, below is the scenario

**Description**:

Test a failed path when typing less than 50 words for any free text question, you need to see the pop up window

**Steps**:

* Provide the less than 50 words to the free text questions that require longer answers.
* Verify that the chatbot should display the appropriate pop-up window along with **EDIT** and **Continue** buttons in pop up window
* **User should not able to type in the input field until user clicks on “EDIT” and “Continue” button in pop up window**

1. Taking more time to give the response to the users

**About Tasks**

1. I have created the README.md to follow the steps to execute the test script
2. I created the **utils** folder to maintaining the below utils files to use for entire automation framework
3. assertUtils.js: To maintain the all the asserts methods in common util file instead of using them directly in the test script.
4. browserUtils.js: To maintain the all the browser actions on elements in application that can be useful to the entire framework

**Example**:

* clickOnElement
* setInputField

1. commonUtils.js: To maintain the common methods in common file to reduce the duplicate code in many places

**Example**:

* generateRandomEmail
* createReportDirectory

1. Below are changes modified in the “playwright.config.js”
2. **Please update the below piece of code in playwright.config.js**

Here Defining the BaseURL for accessing the globally in the framework

use: {

/\* Base URL to use in actions like `await page.goto('/')`. \*/

// baseURL: 'http://127.0.0.1:3000',

baseURL: 'https://fi.sandbox.sapia.ai/ap-southeast-2/cohorts/623a9ab72ad0b2561fecc7ae'

},

1. **Increate the timeout in playwright.config.js**

I face the timeout issues in test execution as chatbot app lil bit slow while giving the response

Earlier it had "30000" changed to "300000"

// Global test timeout (in milliseconds)

timeout: 300000,

1. Executed the test cases in sequential mode, I set it the below flag to flase
2. /\* Run tests in files in parallel \*/
3. fullyParallel: false,