**TEST CASE GENERATORBOT TEST REPORT**

#### ****Test Suite**: Login Test Case Generator**

##### ****Test Case 1: Generate Test Automation Code for Login Function****

* **Test Case ID**: TC-001
* **Test Case Title**: Generate Test Automation Code for Login Function
* **Preconditions**:
  + User is on the page.
  + User has typed the code snippet for login test cases.
  + User has selected a language.
  + User has entered the total number of test cases, positive and negative test cases.
* **Test Steps**:
  + User navigates to the page for generating test cases.
  + User types the code snippet for the login function.
  + User selects the language.
  + User enters the total number of test cases, positive and negative test cases.
  + User clicks on the circleArrowButton to generate the output.
* **Expected Result**:
  + The test automation code is generated and displayed in the output section.
* **Status**: **Pass**
* **Comments**: Test automation code is successfully generated for the login function.

#### ****Test Case 2: Enter Irrelevant Data****

* **Test Case ID**: TC-002
* **Test Case Title**: Enter Irrelevant Data in Test Case Generation
* **Preconditions**:
  + User is on the page.
  + User has entered data that is irrelevant to generating login test cases.
  + User has selected the language.
  + User has entered the total number of test cases, positive and negative test cases.
* **Test Steps**:
  + User navigates to the page for generating test cases.
  + User enters data that is not related to login function testing.
  + User selects the language.
  + User enters the total number of test cases, positive and negative test cases.
  + User clicks on the circleArrowButton to generate the output.
* **Expected Result**:
  + The output should be a message: "IRRELEVANT DATA."
* **Status**: **Pass**
* **Comments**: The system correctly handles irrelevant data input by providing an appropriate message.

#### ****Test Case 3: Generate Test Automation Code with Input Scenario****

* **Test Case ID**: TC-003
* **Test Case Title**: Generate Test Automation Code with Input Scenario
* **Preconditions**:
  + User is on the page.
  + User has typed the code snippet for login test cases.
  + User has selected the language.
  + User has entered the total number of test cases, positive and negative test cases.
  + User has entered the input scenario for the test cases.
* **Test Steps**:
  + User navigates to the page for generating test cases.
  + User types the code snippet for login test cases.
  + User selects the language.
  + User enters the total number of test cases, positive and negative test cases.
  + User enters the input scenario for the test cases.
  + User clicks on the circleArrowButton to generate the output.
* **Expected Result**:
  + The test automation code is generated and displayed in the output section with the provided input scenario.
* **Status**: **Pass**
* **Comments**: Test automation code generated as expected with the input scenario.

#### ****Test Case 4: Missing Total Test Cases****

* **Test Case ID**: TC-004
* **Test Case Title**: Missing Total Test Cases, Positive, and Negative Test Cases
* **Preconditions**:
  + User is on the page.
  + User has typed the code snippet for login test cases.
  + User has selected the language.
  + User has not entered the total number of test cases, positive and negative test cases.
* **Test Steps**:
  + User navigates to the page for generating test cases.
  + User types the code snippet for login test cases.
  + User selects the language.
  + User does not enter the total number of test cases, positive and negative test cases.
  + User clicks on the circleArrowButton to generate the output.
* **Expected Result**:
  + The test automation code should not be generated in the output section due to missing test case details.
* **Status**: **Pass**
* **Comments**: System does not generate output as expected when essential details are missing.

#### ****Test Case 5: Missing Language Selection****

* **Test Case ID**: TC-005
* **Test Case Title**: Missing Language Selection for Test Case Generation
* **Preconditions**:
  + User is on the page.
  + User has typed the code snippet for login test cases.
  + User has not selected the language.
  + User has entered the total number of test cases, positive and negative test cases.
* **Test Steps**:
  + User navigates to the page for generating test cases.
  + User types the code snippet for login test cases.
  + User does not select the language.
  + User enters the total number of test cases, positive and negative test cases.
  + User clicks on the circleArrowButton to generate the output.
* **Expected Result**:
  + The test automation code should not be generated in the output section due to missing language selection.
* **Status**: **Pass**
* **Comments**: The system correctly prevents output generation if the language is not selected.

#### ****Test Case 6: Missing Code Snippet for Test Case Generation****

* **Test Case ID**: TC-006
* **Test Case Title**: Missing Code Snippet for Test Case Generation
* **Preconditions**:
  + User is on the page.
  + User has not typed the code snippet for login test cases.
  + User has selected the language.
  + User has entered the total number of test cases, positive and negative test cases.
* **Test Steps**:
  + User navigates to the page for generating test cases.
  + User does not type the code snippet for login test cases.
  + User selects the language.
  + User enters the total number of test cases, positive and negative test cases.
  + User clicks on the circleArrowButton to generate the output.
* **Expected Result**:
  + The test automation code should not be generated in the output section due to missing code snippet.
* **Status**: **Pass**
* **Comments**: System prevents output generation when no code snippet is provided.

### Test Case ID: TC-007

**Test Case Title**: Backend API Communication with GROQ Cloud

#### Preconditions:

* The backend server is in an "on" state.
* The frontend is sending data to the backend.

#### Test Steps:

1. **GIVEN** the backend server is in an "on" state.
2. **WHEN** the data is received from the frontend.
3. **THEN** the backend API should properly communicate with the GROQ cloud.
4. **AND** the backend API should return the proper response data.

#### Expected Result:

* The backend API successfully communicates with the GROQ cloud and returns the correct response data.

#### Status:

* **PASSED**

#### Comments:

* The backend API successfully handled the data and returned the expected results from the GROQ cloud.

### Summary of Test Results:

* **Test Case 1**: **Pass**
* **Test Case 2**: **Pass**
* **Test Case 3**: **Pass**
* **Test Case 4**: **Pass**
* **Test Case 5**: **Pass**
* **Test Case 6**: **Pass**
* **Test Case 7: Pass**