**Test Task: Documentation and Test Scenario Identification**

**Task:**

**Documentation:**

#### Introduction:

This document outlines the test plan for the online registration form of a website. The form allows users to register by entering their personal information, including full name, email address, password, confirm password, date of birth, gender, and newsletter subscription preference. The form should validate inputs, display appropriate error messages, and successfully submit the data to the server upon correct completion.

#### Objectives:

* To verify that the registration form functions as expected.
* To ensure that all input fields are validated correctly.
* To confirm that appropriate error messages are displayed for invalid inputs.
* To check that the data is successfully submitted to the server upon correct completion of the form.

#### Scope:

The scope of this test plan includes:

* Functional testing of each field in the registration form.
* Validation testing to ensure correct error messages are displayed.
* Usability testing to ensure a smooth user experience.
* Security testing to ensure data is handled securely.

#### Test Environment:

* Operating System: Windows 10, macOS, Linux
* Browsers: Chrome, Firefox, Safari, Edge
* Tools: Cypress, Javascript

#### Test Data:

* Valid and invalid full names
* Valid and invalid email addresses
* Passwords with less than 8 characters
* Matching and non-matching passwords
* Various dates of birth
* Different gender selections
* Newsletter subscription preferences

#### Test Scenarios:

1. Successful registration with valid data.
2. Error messages for empty required fields.
3. Error message for invalid email format.
4. Error message for password less than 8 characters.
5. Error message for non-matching passwords.
6. Successful data submission to the server.

#### Test Execution Schedule:

* Week 1: Test planning and preparation of test data.
* Week 2: Development of test cases and automated test scripts.
* Week 3: Execution of test cases and defect reporting.
* Week 4: Retesting and regression testing.

#### Risks and Assumptions:

* Risks: Potential delays in test execution due to unavailability of test environment.
* Assumptions: The registration form has been implemented as per the requirements and is ready for testing.

**Test Cases:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Preconditions** | |  | | --- | |  |  |  | | --- | | **Test Steps** | | **Expected Results** | **Post-conditions** |
| TC\_RF\_001 | Verify successful registration with valid data | User is on the registration form page | 1. Enter a valid full name.  2. Enter a valid email address.  3. Enter a valid password (at least 8 characters).  4. Enter the same password in the confirm password field.  5. Select a date of birth.  6. Select a gender.  7. Select 'Yes' for newsletter subscription.  8. Click the 'Submit' button. | User should be successfully registered, and a confirmation message should be displayed. | User's data is stored in the server database. |
| TC\_RF\_002 | Verify error message is displayed for empty required fields | User is on the registration form page | 1. Leave all fields empty.  2. Click the 'Submit' button. | Error messages should be displayed for each empty required field. | User is prompted to fill in the required fields. |
| TC\_RF\_003 | Verify error message is displayed for invalid email format | User is on the registration form page | 1. Enter a valid full name.  2. Enter an invalid email address (e.g., 'user@domain').  3. Enter a valid password (at least 8 characters).  4. Enter the same password in the confirm password field.  5. Select a date of birth.  6. Select a gender.  7. Select 'Yes' for newsletter subscription.  8. Click the 'Submit' button. | An error message should be displayed indicating the invalid email format. | User is prompted to enter a valid email address. |

**Scenario Identification**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario Type** | **Scenario Description** | **Steps** | **Expected Outcome** |
| Positive | Successful Registration | 1. Enter a valid full name.  2. Enter a valid email address.  3. Enter a valid password (at least 8 characters).  4. Enter the same password in the confirm password field.  5. Select a date of birth.  6. Select a gender.  7. Select 'Yes' for newsletter subscription.  8. Click the 'Submit' button. | User should be successfully registered, and a confirmation message should be displayed. |
| Positive | Opt-out of Newsletter Subscription | 1. Enter a valid full name.  2. Enter a valid email address.  3. Enter a valid password (at least 8 characters).  4. Enter the same password in the confirm password field.  5. Select a date of birth.  6. Select a gender.  7. Select 'No' for newsletter subscription.  8. Click the 'Submit' button | User should be successfully registered, and a confirmation message should be displayed |
| Negative | |  | | --- | | Empty Required Fields |  |  | | --- | |  | | 1. Leave all fields empty.  2. Click the 'Submit' button. | Error messages should be displayed for each empty required field. |
| Negative | Invalid Email Format | 1. Enter a valid full name.  2. Enter an invalid email address (e.g., 'user@domain').  3. Enter a valid password (at least 8 characters).  4. Enter the same password in the confirm password field.  5. Select a date of birth.  6. Select a gender.  7. Select 'Yes' for newsletter subscription.  8. Click the 'Submit' button. | An error message should be displayed indicating the invalid email format. |

**Test Writing:**

**Test Script for Successful Registration**

**describe('Registration Form Tests', () => {**

**it('should successfully register with valid data', () =>** {

cy.visit('https://example.com/registration');

cy.get('input[name="fullName"]').type('John Doe');

cy.get('input[name="email"]').type('john.doe@example.com');

cy.get('input[name="password"]').type('password123');

cy.get('input[name="confirmPassword"]').type('password123');

cy.get('select[name="dob"]').select('1990-01-01');

cy.get('input[name="gender"][value="Male"]').check();

cy.get('input[name="newsletter"][value="Yes"]').check();

cy.get('button[type="submit"]').click();

cy.contains('Registration successful').should('be.visible');

});

**it('should successfully register without subscribing to the newsletter', () =>** {

cy.visit('https://example.com/registration');

cy.get('input[name="fullName"]').type('Jane Doe');

cy.get('input[name="email"]').type('jane.doe@example.com');

cy.get('input[name="password"]').type('password123');

cy.get('input[name="confirmPassword"]').type('password123');

cy.get('select[name="dob"]').select('1992-02-02');

cy.get('input[name="gender"][value="Female"]').check();

cy.get('input[name="newsletter"][value="No"]').check();

cy.get('button[type="submit"]').click();

cy.contains('Registration successful').should('be.visible');

});

});

### Additional Information:

* The registration form is assumed to follow standard HTML5 input validation for email and password fields.
* Error messages are assumed to be displayed as inline text next to the respective input fields.