Introduction:

The methodology, goals, and parameters of testing the registration form are described in this test plan. To ensure the form works properly and satisfies the requirements, the plan will include information on the testing approach, environment, test data, and timeframe.

Objectives :

The following are this test plan's main goals:

1- to confirm that the registration form accurately captures and processes user data.
2- to guarantee the validation of all required fields.
3- to verify that the form functions properly in a range of settings and input scenarios.
4- to find and fix any issues before the form is used in production.

Scope :

This test plan's scope consists of:

Testing the registration form's functionality.
input field validation (such as text boxes, dropdown menus, and radio buttons).
confirmation of data storage and form submission.
Validating field data and error messages.
To make sure the form is user-friendly, conduct usability testing.
compatibility testing across devices and browsers.

Test Environment :

The testing ground will comprise:

Browsers: Chrome, Firefox, Safari, Edge
Operating Systems: Windows, macOS, and Linux
Devices: Mobile, Tablet, and Desktop
Test server running the XYZ application in its staging version.
Test data is used to configure the database.

Test Data :

To cover a range of input circumstances, test data will be generated, such as:

sets of valid data necessary for a successful registration.
Erroneous data sets (e.g., erroneous email formats, and missing mandatory fields) are used to evaluate error handling.
data sets with boundary values (such as the maximum and minimum input lengths).
SQL injection attempts and special characters.

Test scenarios:

Testing for Field Validation:
Check sure the obligatory fields are all marked as such.
Verify that the input fields (such as email and phone number) accept the correct data formats.
Verify that fields reject data that isn't legitimate (such as an improper email format).

Tests for Form Submission:
Examine a successfully submitted form with accurate data.
When submitting the form with inaccurate or missing information, check the error messages.
Verify how duplicate registrations are handled.

Tests of User Experience:
Make sure it's simple to use and navigate the form.
Check to make sure error messages are understandable and informative.
Verify the form's functionality across a range of screens and devices.

Tests for security:
Check for vulnerabilities related to SQL injection.
Make sure that the data is being sent securely (via HTTPS, for example).
Tests for Device and Browser Compatibility:

To ensure consistency, test the signup form across a range of browsers and devices.

Test scenarios:

Test scenarios will include:

1- Testing for Field Validation:

Check sure the obligatory fields are all marked as such.
Verify that the input fields (such as email and phone number) accept the correct data formats.
Verify that fields reject data that isn't legitimate (such as an improper email format).

2- Tests for Form Submission:
Examine a successfully submitted form with accurate data.
When submitting the form with inaccurate or missing information, check the error messages.
Verify how duplicate registrations are handled.

3- Tests of User Experience:
Make sure it's simple to use and navigate the form.
Check to make sure error messages are understandable and informative.
Verify the form's functionality across a range of screens and devices.

4- Tests for security:
Check for vulnerabilities related to SQL injection.
Make sure that the data is being sent securely (via HTTPS, for example).

5- Tests for Device and Browser Compatibility:
To ensure consistency, test the signup form across a range of browsers and devices.

Test Execution Schedule:

Test Phase	**Start Date**	**End Date**	**Responsible Team**
Test planning	y-m-d	y-m-d	QA Team
Test case Development	y-m-d	y-m-d	QA Team
Test Environment setup	y-m-d	y-m-d	DevOps Team
Test execution	y-m-d	y-m-d	QA Team
Defect reporting and fixes	y-m-d	y-m-d	QA and Dev Team
Regression testing	y-m-d	y-m-d	QA Team
Test Closure	y-m-d	y-m-d	QA Team

Risk and Assumptions:

Risk:

- Possible hold-ups in the test environment setup.
- Inability to obtain the necessary test data.
- Alterations were made to the registration form specifications throughout the test cycle.

Assumptions:

- The registration form's requirements are consistent and well-documented.
- Before the test execution phase starts, the test environment setup will be finished.
- Every resource that is required (people, software, and hardware) will be accessible on schedule.

Documentation Task B:

Test Case 1: Field Validation for Mandatory Fields

ID for Test Case: TC_REG_01

Test Case Description: Confirm that all required fields are filled in and that the right error messages appear if a field is left empty.

Prerequisites: The test environment has access to the registration form.

Steps in the Test:
Get the registration form open.
All required fields, such as Name, Email, and Password, should be left empty.
Press the "Submit" button.

Anticipated outcomes:
It is not appropriate to submit the form.
Each mandatory field should have an appropriate error message stating that it is required next to it.

Following conditions:
The user is still seeing the problem warnings on the registration form page.

Test Case 2: Successful form Submission With Valid Data

ID for Test Case: TC_REG_02
Test Case Description: Confirm that the registration form was filled out correctly and submitted with accurate information.

Prerequisites: The test environment has access to the registration form.

Steps in the Test:
Get the registration form open.
All mandatory fields (e.g., Name: "John Doe", Email: "john.doe@example.com", Password: "Password123") should have valid data entered in them.
The final step is to press the "Submit" button.

Anticipated outcomes:
The form ought to be successfully submitted.
Either a success message or a referral to the success page should be sent to the user.

Following conditions:
The database should contain the user's data.
The registered credentials ought to allow the user to log in.

Test Case 3: Cross-Browser Compatability

ID for Test Case: TC_REG_03
Test Case Description: Check that the signup form works properly in various web browsers.

Prerequisites: The test environment has access to the registration form.

Steps in the Test:
Use the Chrome browser to access the registration form.
Fill out all the relevant fields with accurate data.
Press the "Submit" button to see how it behaves.

With the Firefox, Safari, and Edge browsers, repeat steps 1-3.

Anticipated outcomes:
All tested browsers should be able to use the form in the same way.
In every browser, the form should be correctly submitted with accurate data.
All browsers should provide the same user experience, with no differences in layout or functionality.

Following conditions:
Regardless of the browser being used, the user's data ought to be kept in the database.
The user ought to have access to log in.

Scenario Identification for the Registration Form:

Positive Scenarios

Positive Scenario 1: Accomplished Registration with Accurate Data Procedures:

Get the registration form open.
All mandatory fields (e.g., Name: "Jane Doe," Email: "jane.doe@example.com," Password: "SecurePass123") must include valid data.
Press the "Submit" button.

Anticipated Result:
The form ought to be successfully submitted.
The user ought to be taken to a success or confirmation page.
You can specify an email address to receive a confirmation email.
The database should contain the user's data.

Positive Scenario 2: Filling Out the Form with Optional Information Actions:
Get the registration form open.
Complete all mandatory fields with accurate data, and don't forget to fill in the optional ones (such as address and phone number).
Press the "Submit" button.

Anticipated Result:
The form ought to be successfully submitted.
The user ought to be taken to a success or confirmation page.

Negative Scenarios:

Steps in the Negative Scenario 1: Missing Required Fields

Get the registration form open.
Leave one or more mandatory fields (password, email, etc.) blank.
Press the "Submit" button.

Anticipated Result:
It is not appropriate to submit the form.
Every mandatory field that is left empty should have an error message next to it stating that it is required.
The user is still on the page with the registration form.

Steps to Take in Negative Scenario 2: Invalid Email Format

Get the registration form open.
All fields require correct data to be entered; however, the email field contains invalid data (such as "invalid-email-format").
Press the "Submit" button.

Anticipated Result:
It is not appropriate to submit the form.
The email field should have an error notice stating that the email format is incorrect next to it.
The user stays on the registration form page.

Test Writing:

This is the process of creating automated test scripts with Cypress for the two previously specified positive scenarios.

Make sure you install Cypress locally in your project by using this command.

```
npm install cypress --save-dev
```

Cypress Test Script for Positive Scenario 1: Successful Registration with Valid Data

```
describe('Registration Form', () => {

  beforeEach(() => {
    cy.visit('/registration') // Adjust the URL to your registration form
  });

  it('should successfully submit the form with valid data', () => {
    cy.get('input[name="name"]').type('Jane Doe');
    cy.get('input[name="email"]').type('jane.doe@example.com');
    cy.get('input[name="password"]').type('SecurePass123');
    cy.get('form').submit();

    // Adjust the selector to the actual success message or redirection confirmation
    cy.url().should('include', '/success');
    cy.contains('Registration successful').should('be.visible');
  });
});
```

Cypress Test Script for Positive Scenario 2: Registration with Optional Fields

```
describe('Registration Form with Optional Fields', () => {

  beforeEach(() => {
    cy.visit('/registration') // Adjust the URL to your registration form
  });

  it('should successfully submit the form with valid data including optional fields', () => {
    cy.get('input[name="name"]').type('Jane Doe');
    cy.get('input[name="email"]').type('jane.doe@example.com');
    cy.get('input[name="password"]').type('SecurePass123');
    cy.get('input[name="address"]').type('123 Main St');
    cy.get('input[name="phone"]').type('555-555-5555');
    cy.get('form').submit();

    // Adjust the selector to the actual success message or redirection confirmation
    cy.url().should('include', '/success');
    cy.contains('Registration successful').should('be.visible');
  });
});
```

To run the test, you can use the following command:

```
npx cypress open
```