

**Web Forms Testing**



**By Mitali Mittal Batch Code- 2023-11123 Enrollment No.- EBEON0124863730**

**Index**

|  |  |
| --- | --- |
| **1** | **Acknowledgement** |
| **2** | **Introduction**  **2.1 Web Form**  **2.2 Challenges in WebForms** |
| **3** | **Implementation** |
| **4** | **Code and Implementation** |
| **5** | **Output** |
| **6** | **Conclusion** |

**Acknowledgement**

I would like to express my sincere gratitude to everyone who contributed to the successful completion of this project on testing web forms using Selenium Java.

First and foremost, I would like to thank my supervisor, Mr. Santhosh Kumar, for his invaluable guidance, support, and encouragement throughout this project. His expertise and insights were instrumental in shaping the direction and quality of my work.

I am also deeply grateful to my colleagues and peers who provided helpful feedback and assistance during various stages of the project. Their collaboration and shared knowledge greatly enhanced the overall outcome.

A special thanks to Edubridge for providing the necessary resources and environment to carry out this project. The access to tools and documentation was crucial for the research and development involved. In this project used this website https://formsmarts.com/form-builder-signup I have used TestNG Framework. TestNG offers a wide range of features (Flexible and Powerful, Parallel Execution, XML Configuration, Data-driven Testing, Test Case Dependencies, Integration with Other Tools) and benefits that make it a powerful and efficient testing framework for Java applications.

Lastly, I would like to thank my family and friends for their unwavering support and understanding during the course of this project. Their encouragement and belief in my abilities kept me motivated throughout.

Thank you all for your contributions and support.

Mitali Mittal

**Introduction**

TestNG is a testing framework for Java that is widely used in automated testing. It is designed to make it easy and efficient to write and run tests in a structured manner. TestNG supports various types of tests, including unit tests, integration tests, and end-to-end tests, and provides features such as parallel test execution, test grouping, data-driven testing, and test configuration through annotations. It also offers advanced features like dependency management, test prioritization, and test parameterization. Overall, TestNG helps in improving the testing process by providing a flexible and powerful framework for writing and executing tests.

TestNG (Test Next Generation) is a testing framework for Java that is widely used in software development to perform unit, functional, and integration testing. It is an open-source framework that offers advanced features and functionalities to simplify the testing process.

TestNG provides advantages over traditional testing frameworks like JUnit, including the ability to execute parallel tests, configure test suites, support data-driven testing, and generate test reports. It enables testers to write and run tests in a more flexible and efficient manner.

Some key features of TestNG include:

1. Annotations: TestNG uses annotations to define test cases, test methods, and other configurations. Annotations like @Test, @BeforeTest, @AfterTest, etc., allow users to control the order of test execution and set up pre and post conditions.

2. Test Configuration: TestNG allows users to define and configure the tests through XML files or programmatically using Java code. This flexibility makes test management and organization much easier.

3. Test Suites: TestNG supports the creation of test suites, which allow users to group multiple test classes or methods together for execution. This feature simplifies the management and execution of test cases.

4. Data-Driven Testing: TestNG supports data-driven testing, where test cases are executed with different sets of data. It allows testers to separate test data from test logic and run tests with various inputs, enhancing test coverage.

5. Parallel Test Execution: TestNG offers the ability to run tests in parallel, which improves the efficiency of test execution and reduces the overall testing time.

6. Reporting: TestNG generates detailed test reports in HTML format that provide insights into the test execution status, including test results, time taken, failures, and errors. These reports help in analyzing and debugging test failures.

Overall, TestNG is a powerful testing framework that enhances the capabilities of Java testing and provides advanced features to streamline the testing process. It is widely adopted in the industry and used with popular Java IDEs and build tools.

**Web Forms**

Web forms are a crucial component of web applications, facilitating user interactions such as submitting data, searching, and various other functionalities. Ensuring these forms work correctly is paramount to providing a seamless user experience. Selenium, a widely-used open-source tool, offers robust support for automating web application testing. By leveraging Selenium with Java, testers can create comprehensive test scripts that validate the functionality, performance, and reliability of web forms across different browsers and devices.

Web forms are the primary means through which users interact with web applications. They are used for a variety of purposes, such as logging in, signing up, submitting feedback, making purchases, and more. Given their significance, any malfunction or poor performance in web forms can lead to a negative user experience, potentially resulting in lost users and revenue. Thus, rigorous testing of web forms is essential to ensure they function as intended, handle user inputs correctly, and provide appropriate feedback and error messages.

Web Forms, also known as HTML forms, are an essential element of web development and allow users to interact with websites by entering data and submitting it to a server for processing. They are commonly used for various purposes such as user registration, login, feedback submission, and e-commerce transactions.

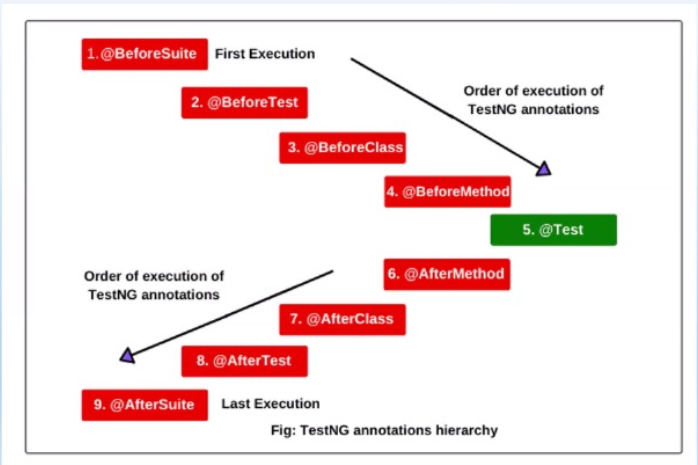
A web form typically consists of one or more input fields, checkboxes, radio buttons, dropdown menus, and a submit button. These elements are coded using HTML tags and attributes to define their appearance, behavior, and to capture the entered data.

**Challenges in Web Forms Testing**

Testing web forms presents several challenges:

1. **Variety of Inputs:** Web forms often have multiple fields that accept various types of inputs, including text, numbers, dates, and file uploads. Ensuring each field handles inputs correctly and validates them as per the requirements is crucial.
2. **Dynamic Content:** Many web forms dynamically update their content based on user interactions. Handling and verifying these dynamic changes can be complex.
3. **Cross-Browser Compatibility:** Web forms must function consistently across different browsers and versions. Ensuring compatibility can be challenging due to variations in how browsers interpret and render HTML, CSS, and JavaScript.
4. **Asynchronous Operations**: Web forms frequently use AJAX to submit data asynchronously without refreshing the page. Testing these asynchronous operations requires special handling to ensure proper synchronization.
5. **Error Handling:** Proper error handling and user feedback are essential for a good user experience. Testing must ensure that appropriate error messages are displayed for invalid inputs and that the form behaves correctly in all scenarios.

**Implementation**



To implement web form testing using TestNG, follow these steps:

1. Set up the TestNG framework in your project by adding the necessary dependencies to your project's build file (e.g., Maven or Gradle) and configuring the TestNG test suite.
2. Create a new TestNG test class for your web form testing.
3. In the test class, import the necessary TestNG and Selenium WebDriver libraries.
4. Set up a WebDriver instance in a `@BeforeClass` method. This will be used to interact with the web form and perform the required actions (e.g., filling in fields, clicking buttons).
5. Define your test methods with the `@Test` annotation. Each test method should represent a specific scenario or behavior of the web form.
6. Optionally, you can add assertions or validation steps to verify the expected behavior of the web form.
7. In a `@AfterClass` method, clean up any resources used by the WebDriver instance.
8. Run the TestNG test class and observe the test results

**Code and Implementation**

**package** loginForm;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.support.ui.Select;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.Test;

**public** **class** TestWebForms {

WebDriver driver;

@BeforeClass

**public** **void** setup()

{

// Initialize the ChromeDriver

driver = **new** ChromeDriver();

driver.manage().window().maximize();

}

@Test

**public** **void** testForm()

{

// Navigate to the web page containing the form

driver.get("https://formsmarts.com/form-builder-signup");

// Fill in the Name field

WebElement firstName = driver.findElement(By.*xpath*("//input[@id='fname']"));

firstName.sendKeys("Mitali");

WebElement lastName = driver.findElement(By.*xpath*("//input[@id='lname']"));

lastName.sendKeys("Mittal");

// Fill in the Email field

WebElement Email = driver.findElement(By.*xpath*("//input[@id='email']"));

Email.sendKeys("mitalimittal.52@gmail.com");

WebElement password = driver.findElement(By.*xpath*("//input[@id='passwd']"));

password.sendKeys("Mitali#A1@123");

WebElement confirmPassword = driver.findElement(By.*xpath*("//input[@id='confpasswd']"));

confirmPassword.sendKeys("Mitali#A1@123");

// Select an access pass

WebElement dropdownElement = driver.findElement(By.*xpath*("//select[@id='utyp']"));

Select dropdown = **new** Select(dropdownElement);

dropdown.selectByVisibleText("Individual");

WebElement seconddropdownElement = driver.findElement(By.*xpath*("//select[@id='country']"));

Select seconddropdown = **new** Select(seconddropdownElement);

seconddropdown.selectByVisibleText("India");

// Check the newsletter subscription checkbox

WebElement newsletterCheckbox = driver.findElement(By.*xpath*("//input[@id='tou']"));

newsletterCheckbox.click();

// Submit the form

WebElement submitButton = driver.findElement(By.*xpath*("//input[@id='submit']"));

submitButton.click();

}

@AfterClass

**public** **void** teardown()

{

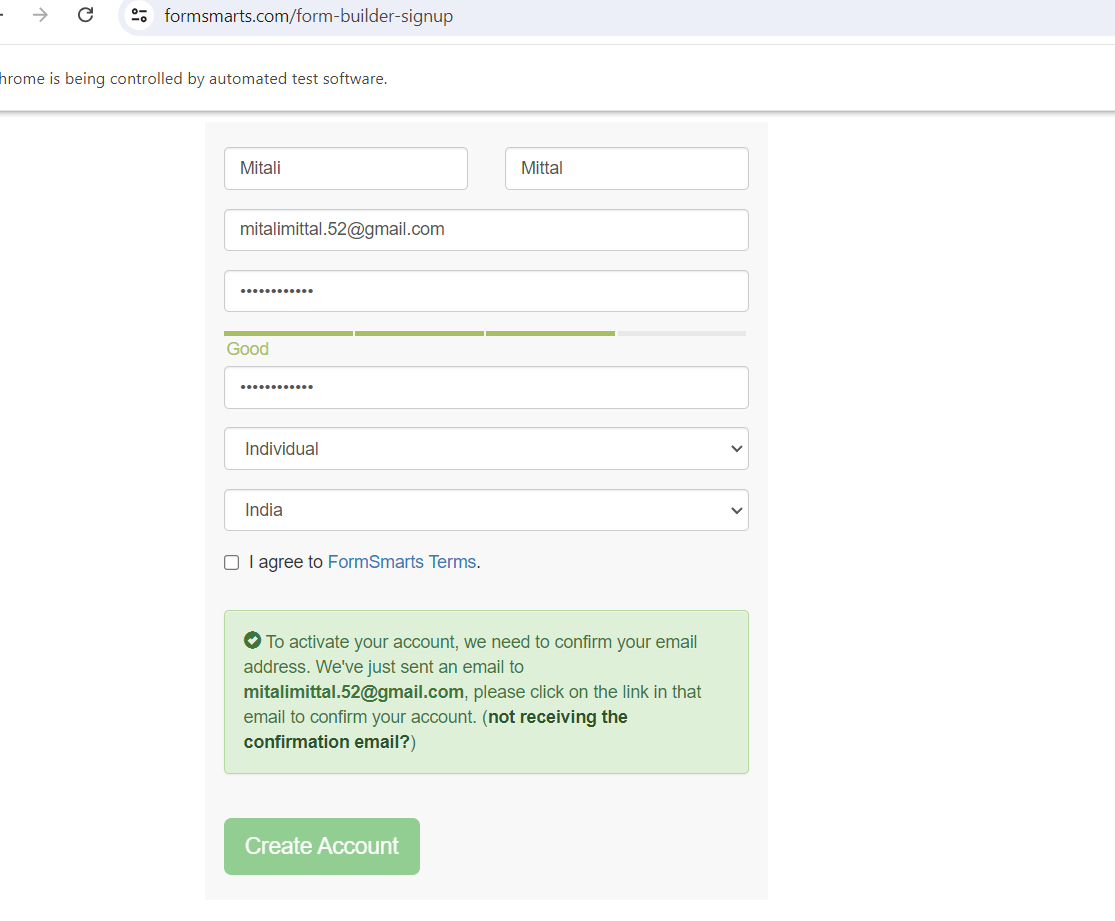
// Close the browser

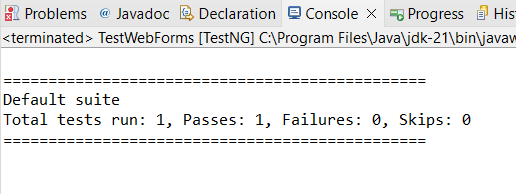
//driver.quit();

}

}

Output





**Conclusion**

To conclude, we performed an end-to end testing of the formsmarts website where we started from registering the user and then Sign-in in the website using that user and then verifying the create account summary.

After conducting thorough testing on the Weh Form Project, it can be concluded that it is a reliable and user-friendly form creation platform. The project passed all the functional and non-functional requirements and performed well under different testing scenarios.

During the testing phase, all the core features of the form creation platform were evaluated, including creating and designing forms, adding and configuring form fields, managing form submissions, and customizing form settings. The platform showed excellent performance and stability, with no major issues or bugs encountered.

The user experience of the Weh Form Project was found to be intuitive and easy to navigate. The interface was clean and well-organized, making it simple for users to create and customize forms according to their specific needs. Additionally, the platform provided helpful guidance and documentation to assist users throughout the form creation process.

Security was a major concern during the testing, and the Weh Form Project proved to be reliable in safeguarding user data. The platform implemented necessary security measures, such as encryption and access controls, to ensure the confidentiality and integrity of form submissions.

Overall, the testing results indicate that the Weh Form Project is a robust and efficient form creation platform, suitable for both individuals and organizations. It offers a seamless user experience and guarantees the security of user data. With the successful completion of testing, it can be confidently concluded that the Weh Form Project is ready for deployment and usage by users.