

Why  
Katalon ▾

Platform ▾

Solutions ▾

Integrations

View  
Pricing  
demoDownload  
Resources ▾  
Studio

All

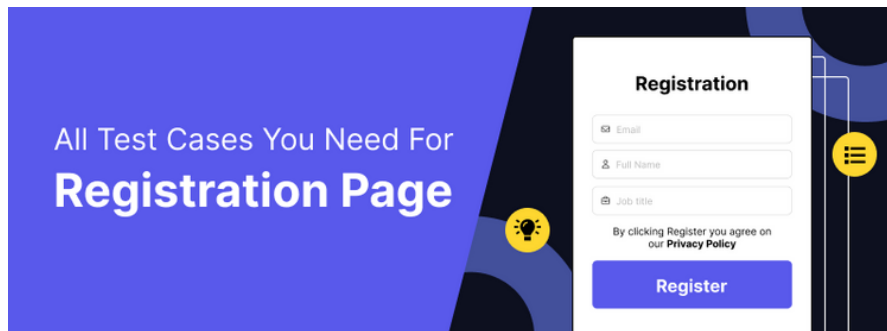
News

Products

Insights

DevOps and  
CI/CD**TABLE OF CONTENTS**Free Test Case  
Template To Download1. Functional Test  
Cases For Registration  
Page2. Security Test Cases  
For Sign-up Page3. Performance Test  
Cases On The  
Registration Page4. Cross-browser Test  
Cases For The  
Registration Page4. Integration Test  
Cases5. Usability Test Cases  
On Registration PageHow To Write Test  
Cases For Registration  
Page Faster and Better  
With Katalon Platform

# 100 Test Cases For Registration Page (With Template + Detailed Guide)



A Registration page is more than just a place for users to create their account. It is a process designed to protect the backend, ensure compliance with privacy and security standards, as well as verify the users before granting them certain access to the digital services. If you are trying to test your Registration page and don't know where to start, read on and discover **100 test cases for Registration Page** that you can use for references.

In this article, we will list out the most common and essential test cases for that page and categorize them in groups. We also include a **sample test case template for**

**this page** so that you can get started more quickly.

**Read More:** [A Complete Web Testing Checklist](#)

## Free Test Case Template To Download

To best write your test cases, you should always have a test case template, which we have prepared in PDF, Doc, and Excel sheet formats for you to download. Simply hit the button below and start noting down your test cases right away.

[PDF](#) | [Doc](#) | [Excel](#)

Or Simply Manage All of Your Test Cases With  
Katalon TestOps

## 1. Functional Test Cases For Registration Page

[Functional testing](#) is a must-have for any areas of the website or software, not just the registration page. Essentially these tests check if the Registration page is doing what it is supposed to be doing: verify that new users can successfully register with valid information.

**Read More:** [Top Functional Testing Tools To Significantly Enhance Your Testing Activities](#)

### 1. Mandatory Fields and Instructions:

1. Verify the presence of all mandatory fields on the registration page.



2. Ensure clear and concise instructions are provided on how to fill out the registration form.
3. Test the registration page's behavior when trying to submit the form with just the email field filled out.
4. Test the registration page's behavior when trying to submit the form with just the password field filled out.
5. Ensure that error messages are displayed specifically for each empty mandatory field when attempting to submit the form with missing information.
6. Check if the registration page provides inline validation for each field, indicating whether the provided information is valid or not as the user fills out the form.
7. Test the behavior of the registration page when submitting the form with valid information in all mandatory fields.
8. Verify that the registration page prevents submission if any mandatory field is left empty.
9. Confirm that the registration page maintains the data entered in the fields even after a submission attempt with missing information, so users don't need to re-enter valid data.
10. Ensure that the registration page does not allow leading or trailing spaces in mandatory fields (e.g., name, email).
11. Test the case where the email field is pre-filled with an example email address or placeholder text, ensuring that it is cleared when the user begins typing.
12. Verify that the registration page provides tooltips or help icons near mandatory fields to explain any specific requirements or constraints.

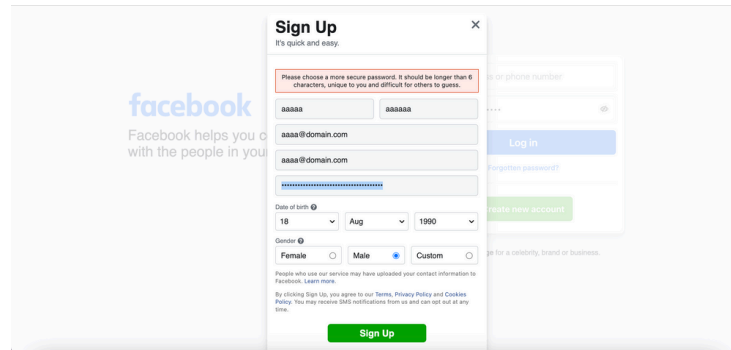





## 2. Field Validation:

1. Test the behavior when submitting the form with empty or blank fields individually.
2. Test the response to submitting the form with all fields empty.
3. Validate email format and response for invalid email addresses.
4. Verify the enforcement of password strength criteria (length, special characters, numbers, etc.).
5. Ensure the password and password confirmation fields match.
6. Verify that the registration page correctly identifies and handles email addresses with multiple "@" symbols (e.g., "user@domain@domain.com").
7. Test the registration page's handling of excessively long email addresses, ensuring it enforces a reasonable character limit and displays an error message if exceeded.
8. Check the registration page's response to entering a password without any alphanumeric characters (e.g., "\*\*\*\*\*"), ensuring it enforces the inclusion of letters and numbers.
9. Verify that the registration page properly handles passwords with spaces, special characters, or non-alphanumeric characters and validates them according to the specified criteria.
10. Test the behavior when the password and password confirmation fields contain leading or trailing spaces, and ensure they are correctly handled and trimmed.


11. Verify that the registration page detects and prevents the use of commonly used or easily guessable passwords (e.g., "password," "123456," "admin").
12. Test the registration page's response when the password and password confirmation fields contain long strings of the same character (e.g., "aaaaaaaaaaaa").



A Facebook password containing only the "a" characters returns the error message as shown

### 3. Error Handling:

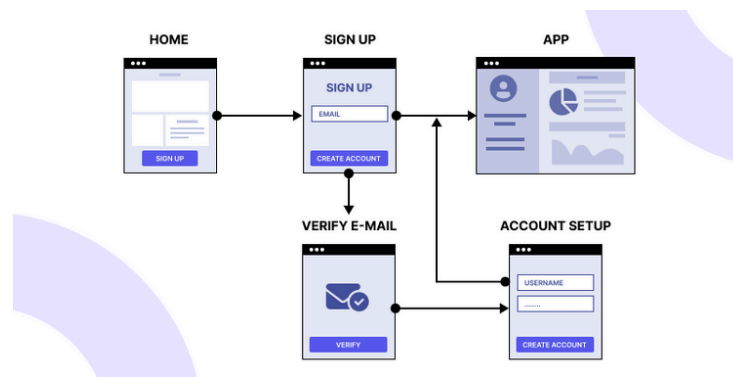
1. Check the response when providing an email address already registered.
2. Validate the CAPTCHA feature's effectiveness in distinguishing users from bots.
3. Verify that error messages are clear and user-friendly for various issues (e.g., invalid email, password mismatch).
4. Test the response to using a username that is already taken.
5. Verify that the registration page gracefully handles any internal server errors or exceptions that may occur during the registration process and displays a user-friendly error message.
6. Test the registration page's response when the server or network experiences high traffic or slow response times, ensuring it doesn't display misleading error messages.



7. Verify that the registration page provides context-specific error messages for different issues (e.g., username already taken, invalid email format) rather than generic error messages.
8. Test how the registration page handles the case when a user with a previously locked account tries to register again, ensuring it provides instructions for account recovery instead of displaying a vague error.
9. Check the registration page's response when a user submits a registration form with a username that is a reserved keyword or term (e.g., "admin," "root"), ensuring it prevents registration and explains the restriction.
10. Verify that the registration page effectively handles cases where a user's session expires during the registration process, prompting them to log in again and preserving their entered data.

**Read More:** [A Complete Guide on Database Testing](#)

#### 4. Email Verification:



Basic email verification workflow

1. Verify the sending of a verification email to the provided address upon successful registration.
2. Test the ability to resend verification emails if needed.
3. Check that the registration page handles cases where a user attempts to use an expired verification link, providing instructions for re-sending the verification email.

4. Test the behavior when a user clicks on the verification link multiple times, ensuring that additional clicks do not cause any unintended side effects.
5. Verify that the registration page clearly communicates the need to check the spam or junk folder for the verification email, ensuring users are aware of this possibility.
6. Test the registration page's handling of cases where the email verification link has been tampered with or contains invalid characters, ensuring it detects and responds to such situations securely.
7. Check the registration page's response when the user enters the verification code manually, ensuring it correctly validates the code and confirms the email address.
8. Verify that the registration page enforces a time limit for completing the email verification process and prevents users from using outdated verification links.
9. Test how the registration page handles the situation when a user attempts to verify an email address that is already associated with an active account, ensuring it provides appropriate guidance.

## 5. Media:

1. Confirm that users can upload a profile picture during registration (if applicable).
2. Verify that users can successfully upload a valid image file (e.g., JPEG, PNG) as their profile picture during registration.
3. Test the registration page's response when users attempt to upload an image in an unsupported format (e.g., GIF, BMP). Ensure that it displays an appropriate error message.
4. Check if the registration page enforces a maximum file size for profile pictures and accurately displays an error message if the size exceeds the limit.



5. Verify that uploaded profile pictures are appropriately resized or cropped to generate thumbnails for user profiles.
6. Test the ability to cancel a photo upload in progress and ensure it stops the upload process without causing errors.
7. Ensure the registration page allows optional profile information input (e.g., bio).

## 6. Terms and Policies:

**Typeform**

Get better data with conversational forms, surveys,  
quizzes & more.

randomemail@random.com

.....

☒ I agree to Typeform's [Terms of Service, Privacy Policy and Data Processing Agreement](#).

[See options](#)

Create my free account

### Typeform sign-up page with Terms and Policies acceptance section

1. Verify that users are presented with the terms of service and privacy policy on the registration page.
2. Check that users are required to accept the terms of service and privacy policy before proceeding with registration.
3. Ensure that there is a link provided for users to access the full terms of service and privacy policy documents.
4. Test that screen readers and assistive technologies can properly read and navigate the terms and policy text.
5. Verify that an appropriate error message is displayed if a user attempts to register without accepting the terms.





6. Check that the terms and policy text is presented in a clear and readable format, avoiding dense legal language when possible.
7. Ensure that the system records the date and time of a user's acceptance of the terms and privacy policy, associating it with the user's account.
8. If the application serves a global audience, verify that terms and policy documents are available in multiple languages, and users can choose their preferred language.
9. Test the process for users to revoke their consent to the terms and privacy policy after registration and how it affects their account.

## 7. Duplicate Account Prevention:

1. Test the ability to prevent duplicate accounts using the same email address.
2. Verify that prohibited characters and spaces in usernames are disallowed.
3. Check if the registration page treats email addresses with different casing (e.g., "User@email.com" and "user@email.com") as unique or not.
4. Verify correct handling of very long usernames within input length limitations.
5. Test the registration page's response when a user tries to register with a username that is already in use by another account, ensuring it prevents registration and displays a relevant error message.
6. If the registration page allows users to sign up using social media accounts (e.g., Google, Facebook), verify that it prevents the creation of duplicate accounts for the same social media profile.
7. Test the behavior when a user deactivates their account and later attempts to re-register with the same email address or username.



**Read More:** [100 Test Cases For Login Page](#)

## 2. Security Test Cases For Sign-up Page

The Registration page is also one of the first layers of security for the system. If not properly developed with data security best practices in mind, the Registration page can be vulnerable to cyber attacks, especially when the information stored in your system is sensitive. Here are some security test cases you need to consider for your Registration page:

### 1. Input Validation:

1. Verify that the registration page properly validates user input and rejects any data that contains potentially harmful characters (e.g., SQL injection attempts, script tags).
2. Test for Cross-Site Scripting (XSS) vulnerabilities by attempting to inject malicious scripts into input fields and ensure that the page sanitizes and escapes user-generated content.
3. Test for Cross-Site Request Forgery (CSRF) vulnerabilities by simulating unauthorized form submissions and confirming that the registration page employs CSRF tokens to prevent such attacks.
4. Check if the registration page correctly handles input containing HTML entities or URL-encoded characters and ensures they are properly decoded.



### 2. Password Security:

1. Verify that the registration page enforces strong password policies, including requirements for length, special characters, and a mix of upper and lower case letters.
2. Test for password hashing and encryption by registering and checking how the password is stored in the database, ensuring it is not stored in plaintext.



3. Confirm that the registration page does not display the entered password in plaintext, even momentarily, during the registration process.
4. Test for password brute-force attacks by attempting to register with commonly used passwords and verifying that the system detects and prevents such attempts.
5. Check if the registration page provides password strength feedback to users, guiding them on creating strong passwords.

### 3. Session Management and Authentication:

1. Verify that the registration page enforces secure session management practices, including using secure cookies and properly expiring sessions after a period of inactivity.
2. Test the registration page's response when attempting to register with an existing, but locked or banned, user account, ensuring it provides appropriate feedback.
3. Test the behavior when attempting to register with a previously registered email address and ensure that it does not reveal whether an email is registered or not.
4. Check if the registration page uses account lockout mechanisms after multiple failed registration attempts to prevent brute-force attacks.
5. Verify that the registration page employs rate limiting to protect against registration attempts from the same IP address at an unusually high frequency.



### 4. Data Privacy and Protection:

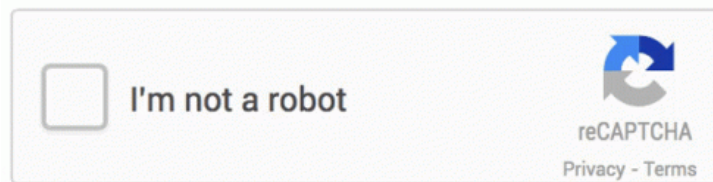
1. Ensure that the registration page complies with data privacy regulations (e.g., GDPR) by obtaining explicit consent from users to collect and process their personal data.
2. Test the registration page's response to user requests to delete their accounts and personal data,



confirming that it handles data deletion requests securely and effectively.

3. Check that the registration page stores sensitive user data, such as passwords and email addresses, securely using strong encryption and hashing techniques.
4. Verify that the registration page does not expose sensitive information in error messages or response headers, protecting against information disclosure.=

## 5. Account Recovery and Verification:



1. Test the security of the email verification process by attempting to manipulate the verification link to gain unauthorized access or verifying with expired links.
2. Verify that the registration page employs CAPTCHA challenges to mitigate automated bot registration attempts.
3. Test the password recovery feature to ensure it securely allows users to reset forgotten passwords without exposing sensitive information.
4. Confirm that the registration page does not provide hints or notifications about the existence of specific user accounts during the password recovery process.
5. Test for user enumeration vulnerabilities by attempting to reset the password for both registered and non-registered email addresses and verifying that the system does not reveal which addresses are valid.

**Read More:** [What is Penetration Testing? A Complete Guide](#)



### 3. Performance Test Cases On The Registration Page

Performance testing helps determine how well the registration page can handle an increasing number of concurrent users. During peak hours, or any special event, the system must be able to handle a sudden influx of traffic and registrations, and performance testing provides insights into the system threshold so that the team can optimize their resources for it better.

**Read More:** [Top 15 Performance Testing Tools For Your QA Team](#)

#### 1. Load Testing:

- Test the registration page with a low load (e.g., 10 concurrent users) to ensure basic functionality and responsiveness.
- Gradually increase the load on the registration page to simulate medium traffic (e.g., 100 concurrent users) and assess its performance.
- Conduct a peak load test with a high number of concurrent users (e.g., 500+) to determine how the page behaves under maximum expected load.
- Test the registration page's ability to handle a sudden spike in traffic by rapidly increasing and decreasing the number of concurrent users.
- Evaluate the registration page's performance during a sustained load over an extended period (e.g., 24 hours) to identify any memory leaks or resource issues.



**Read More:** [Load Testing vs Performance Testing: A Complete Guide](#)

#### 2. Scalability Testing:



1. Assess the registration page's ability to scale horizontally by adding more server instances and ensuring that the load is distributed evenly.
2. Test the registration page's ability to scale vertically by increasing the system's resources (e.g., CPU, RAM) and measuring its improved performance.
3. Verify that the system can automatically provision additional resources in response to increased load (auto-scaling) without causing downtime or performance degradation.
4. Measure the registration page's performance when dealing with different volumes of user registrations (e.g., 100, 1,000, 10,000 registrations per hour).

### 3. Response Time and Throughput:

1. Measure the average response time of the registration page under normal load conditions to establish a baseline.
2. Assess the response time of the registration page as the load increases, ensuring it remains within acceptable limits.
3. Calculate the throughput by determining the number of registrations processed per second under various load levels.
4. Test the registration page's response time during periods of peak activity to identify performance degradation or bottlenecks.
5. Verify that the registration page can handle bursts of registrations without a significant increase in response time.



### 4. Stress Testing:

1. Test the registration page's performance under extreme conditions, such as a significantly higher load than expected during peak usage.
2. Verify the registration page's resilience by gradually increasing the load beyond its capacity until it reaches a breaking point, noting how it fails.



3. Evaluate how the system recovers from a stress test, including the time it takes to return to normal performance levels once the stress is removed.
4. Assess how the registration page handles long-duration stress tests (e.g., 48 hours) to detect potential resource leaks or gradual performance degradation.
5. Verify that the registration page gracefully degrades under stress, prioritizing essential functionality and maintaining core registration capabilities.

### 5. Resource Utilization and Efficiency:

1. Monitor CPU utilization during performance testing to ensure it stays within acceptable levels.
2. Evaluate memory usage to identify potential memory leaks or excessive consumption during registration.
3. Test the registration page's database queries and assess database response times under varying loads.
4. Check network bandwidth usage to ensure that the registration page does not cause network congestion.
5. Measure the impact of the registration process on server disk I/O, ensuring that it does not strain storage resources.



## 4. Cross-browser Test Cases For The Registration Page

Cross-browser testing is when you assess the compatibility and functionality of a website or web application across browsers, platforms, and versions. Currently there are about [63000+ possible browser - device - OS combinations](#) that testers must consider when doing this, which is why they usually leverage [automation testing tools](#) to handle the more repetitive test cases.



**Read More:** [Top 8 Cross-browser Testing Tools For Your QA Team](#)

Some common test cases for cross browser testing include:

### 1. Basic Cross-browser Functionality Testing:

1. Verify that the registration page loads correctly in the latest versions of popular browsers (e.g., Chrome, Firefox, Safari, Edge).
2. Check that placeholders or labels within input fields are displayed correctly and aligned consistently across browsers.
3. Test keyboard navigation to ensure users can move through form elements using the keyboard tab key.
4. Ensure that the registration page is usable and displays appropriately on both desktop and mobile browsers.
5. Test the registration form's submission process in each browser, ensuring users can submit their registration details successfully.
6. Test how each browser handles form validation attributes (e.g., required, minlength, maxlength) and ensures consistency.
7. Check for discrepancies in date and time format input fields (if applicable) across browsers.
8. Verify that special characters (e.g., &, <, >) in input fields are properly encoded and displayed consistently.
9. Test the behavior of browser autofill features when filling out the registration form.
10. Ensure that date pickers and calendar widgets (if used) work consistently across browsers.

### 2. Layout and Responsiveness:

1. Verify that the registration page layout remains consistent across browsers, including correct placement of form elements and labels.





2. Test how the registration page responds to changes in browser window size and ensure elements are appropriately resized or repositioned.
3. Check that media queries and responsive design elements work as expected on various screen sizes and resolutions.
4. Test how the registration page handles viewport changes when users zoom in or out.
5. Verify that the page maintains readability and usability at various zoom levels.
6. Test the behavior of touch events and gestures on mobile browsers, ensuring smooth interactions and form input.
7. Ensure that font rendering, text alignment, and spacing remain consistent across browsers and devices.

### 3. Cross-Device and Cross-Platform Testing:

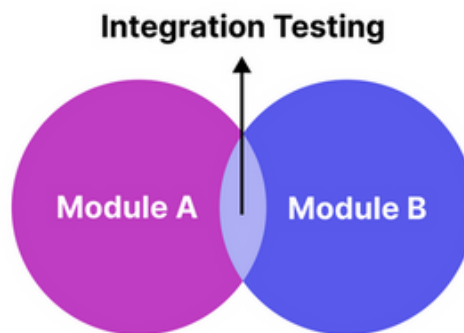
1. Test the registration page on different operating systems (e.g., Windows, macOS, iOS, Android) to ensure cross-platform compatibility.
2. Verify that the page is usable on both touch and non-touch devices (e.g., desktops, tablets, smartphones).
3. Test how the registration page handles device-specific input methods (e.g., virtual keyboards, hardware keyboards, touchscreens, mouse).
4. Check for differences in browser behavior between desktop and mobile versions (e.g., mobile browsers may have different default font sizes).
5. Test the registration page's functionality on different browsers available on mobile devices, such as Chrome for Android and Safari for iOS.



Level Up Your Cross-browser Testing With  
Katalon TestCloud

## 4. Integration Test Cases

[Integration testing for a registration page](#) is about checking to see how that registration page interacts with other system components and external services. Common systems to be connected with the Registration page include the backend database, email sending service (for verification purposes), social media signup integration, and even APIs (such as APIs for localization services to serve registrations of international users/visitors).



Some important integration test cases for this page include:

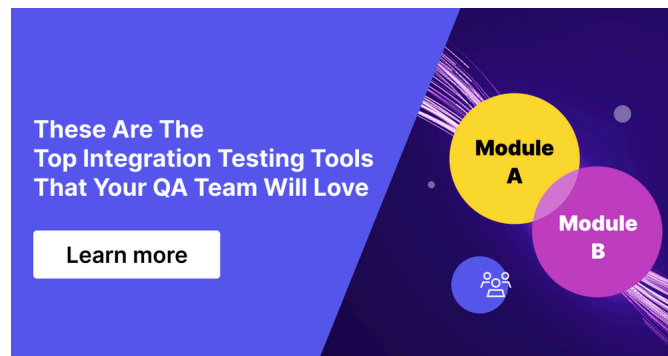
### 1. Database Integration:

1. Verify that user registration data is correctly inserted into the database, including fields like username, email, and hashed password.
2. Test database integration for unique constraints, ensuring that duplicate registrations are prevented based on email addresses or usernames.
3. Check that the registration page properly handles database connection errors and provides informative error messages.

### 2. Email Integration:



1. Test the registration page's integration with an email service to ensure that verification emails are sent upon successful registration.
2. Verify that email content (subject, body) contains the correct user-specific information, such as the verification link.
3. Test email delivery failure scenarios, such as when the email service is unavailable or returns an error, and ensure that the registration process handles them gracefully.



### 3. Third-Party Authentication Integration:

1. Verify that users can register using third-party authentication providers (e.g., Google, Facebook, Twitter) by clicking on their respective authentication buttons.
2. Test registration through multiple third-party providers to ensure consistency in the registration process.
3. Verify the registration page's response when the third-party authentication service is temporarily unavailable or returns an error.
4. Test how the page informs users about third-party authentication failures and provides guidance on resolving them.
5. Test whether users with existing accounts on the registration system can link their accounts to third-party authentication providers for simplified login.
6. Ensure that the linking process works smoothly without causing data conflicts.



7. Test the third-party authentication integration to ensure it complies with user privacy preferences and data security regulations (e.g., GDPR).
8. Verify that user data from third-party providers is handled securely and with respect to user consent.

**Read More:** [Top Test Cases For API Testing You Will Need](#)

## 5. Usability Test Cases On Registration Page

These test cases tend to check the more abstract aspects of the software. It delves into how users interact with the software and how they perceive the overall experience. For example:

1. Verify that the registration form is clearly labeled, and users can easily understand the purpose of each field.
2. Check that field labels are descriptive, placed correctly, and visible to users.
3. Test that placeholder text within input fields provides helpful hints but does not disappear when users start typing.
4. Ensure that input validation messages for errors (e.g., missing fields, invalid email) are displayed clearly and prominently.
5. If the registration process has multiple steps, confirm that users are provided with clear progress indicators.
6. If applicable, assess the password strength indicator to help users create secure passwords.
7. Verify that the cursor focus is set to the first input field, making it easy for users to start typing.
8. Test the registration page's layout and usability on various mobile devices, ensuring that it is mobile-responsive and all form elements are easily accessible.

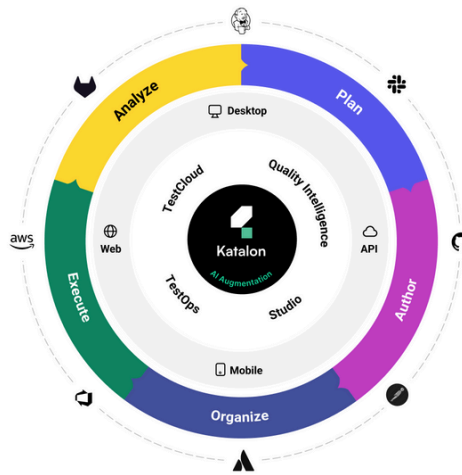

9. Ensure that users can interact with form elements (e.g., buttons, checkboxes) on touchscreen devices without issues.
10. Confirm that the "Register" or similar call-to-action button is prominently displayed and stands out on the page.
11. Test the flow of the registration process to ensure that users can move smoothly from one step to the next without confusion.
12. Verify that users can easily navigate back to the previous step or correct errors without losing their data.
13. If applicable, assess the usability of the confirmation page, including its clarity and the presence of a successful registration message.

## How To Write Test Cases For Registration Page Faster and Better With Katalon Platform



Testing for your signup page is not challenging, but it is highly repetitive by nature. Automating test cases on Registration page is a recommended strategy, but writing test scripts from scratch then continuously maintaining them after each code change is actually quite counterproductive.

With [Katalon Platform](#), you can orchestrate the entire web/software testing lifecycle from test creation, execution, maintenance, to reporting, without having to write a single line of code.



For example, thanks to its **Record-and-Playback** features, you can easily capture test objects and record a sequence of activities. It can then be re-executed in regression testing sessions, saving QA teams a lot of time and effort.

If testers wrote those test scripts from scratch, they also have to actively maintain them, and as the application grows, the number of test scripts to create and maintain grows accordingly, and it is quite a challenge to keep up with that.

Meanwhile, with Katalon, you can simply let [the Self-healing feature](#) do the work. It detects changes in the system and automatically update the test scripts, freeing up resources for your team to focus on other high-value activities.

Katalon Platform also supports running scripts [on a wide range of devices, browsers, and testing environments](#), allowing QA teams to perform a lot of testing activities in

just 1 place instead of having to spend time configuring environments and constantly switching tools.

After executing tests, Katalon Platform enables teams to review test results with the comprehensive and customizable test reports in LOG, HTML, CSV, and PDF formats before forwarding them as email attachments. There is a debugging mode for testers to conduct a root cause analysis for a specific failed case.

Not that just, Katalon is also a forerunner in the [AI testing](#) space. You can leverage a lot of [AI-powered features](#) built into the platform to supercharge your testing.

Download and Witness The Power of Katalon  
(Free) in Action

## Related Articles

Top Regression  
Testing Tools  
You Should Know



Troubleshooting  
Katalon Web Recorder  
A Comprehensive Guide



Ensuring Visibility  
of Screenshots in  
Katalon Reports



Top 10 Regression Testing Tools and Frameworks That Boosts Automation

Troubleshooting Katalon Web Recorder: A Comprehensive Guide

Ensuring Visibility of Screenshots in Katalon Reports



Why Katalon	Platform	Solutions	Resources	Docs	Company
Business Outcomes	Platform Overview	Visual Testing	Academy	Documentation	About Katalon
Speed	Katalon Studio	Web Testing	Blog	Getting Started	Careers
Flexibility & Scalability	Katalon Runtime Engine	API Testing	Community	Image Library on Docker	Events
Visibility	Katalon TestCloud	Mobile Testing	Technical Support	Sample Projects on GitHub	News
Innovation & AI	Katalon TestOps	Desktop Testing	Webinars		Partners
Affordability & ROI	TrueTest	Integrations	Download Katalon Studio		
Success Stories	Pricing		Onboarding Hub		

1720 Peachtree Street NW, Suite 870, Atlanta, GA 30309

General inquiries: [business@katalon.com](mailto:business@katalon.com)  
Legal inquiries: [legal@katalon.com](mailto:legal@katalon.com)  
Partnership inquiries: [partner@katalon.com](mailto:partner@katalon.com)



Terms Privacy License & Security Policy Agreement

Copyright © 2024 Katalon, Inc. All rights reserved.

