

Complete Web Application Testing Checklist



While testing the web applications, one should mention the below mentioned checklist. The below mentioned checklist is applicable for all types of web applications depending on their business requirements.

The web application testing checklist consists of the following:

- Usability Testing
- Functional Testing
- Compatibility Testing
- Database Testing
- Security Testing
- Performance Testing

Now let's look at each checklist in detail:

Usability Testing

What is Usability Testing?

- Usability testing is nothing but the User-friendliness check.
- In Usability testing, the application flow is tested so that a new user can understand the application.
- Basically, system navigation is checked in Usability testing.

What is the purpose or Goal of Usability testing?

A Usability test establishes the ease of use and effectiveness of a product using a standard set of practices.

Usability Test Scenarios:

- Web page content should be correct without any spelling or grammatical errors
- All fonts should be same as per the requirements.
- All the text should be properly aligned.
- All the error messages should be correct without any spelling or grammatical errors and should match with the field label.

- Tool tip text should be there for every field.
- All the fields should be properly aligned.
- Enough space should be provided between field labels, columns, rows, and error messages.
- All the buttons should be in a standard format and size.
- Home link should be there on every single page.
- Disabled fields should be grayed out.
- Check for broken links and images.
- Confirmation message should be displayed for any kind of update and delete operation.
- Check the site on different resolutions (640 x 480, 600x800 etc.?)
- Check the end user can run the system without frustration.
- Check the tab should work properly.
- Scroll bar should appear only if required.
- If there is an error message on submit, the information filled by the user should be there.
- Title should display on each web page.
- All fields (Textbox, dropdown, radio button etc) and buttons should be accessible by keyboard. The user should be able to perform all operations by using keyboard.
- Check if the dropdown data is not truncated due to the field size and also check whether hardcoded or managed via administrator.

Functional Testing:

What is Functional Testing?

- Testing the features and operational behavior of a product to ensure they correspond to requirements.
- Testing that ignores the internal mechanism of a system or component and focuses solely on the output generated in response to selected inputs and execution conditions.

What is the purpose or Goal of Functional testing?

- The goal of Functional testing is to verify whether your product meets the intended functions mentioned in your development documentation.

Functional Test Scenarios:

- Test all the mandatory fields should be validated.
- Test the asterisk sign should display for all the mandatory fields.
- Test the system should not display the error message for optional fields.
- Test that leap years are validated correctly & do not cause errors/miscalculations.
- Test the numeric fields should not accept the alphabets and proper error message should be displayed.
- Test for negative numbers if allowed for numeric fields.
- Test division by zero should be handled properly for calculations.
- Test the max length of every field to ensure the data is not truncated.

- Test the pop up message ("This field is limited to 500 characters") should display if the data exceeds the maximum size of the field.
- Test that a confirmation message should display for update and delete operations.
- Test the amount values should display in currency format.
- Test all input fields for special characters.
- Test the timeout functionality.
- Test the Sorting functionality.
- Test the functionality of the buttons available
- Test the Privacy Policy & FAQ is clearly defined and should be available for users.
- Test if any functionality fails the user gets redirected to the custom error page.
- Test all the uploaded documents are opened properly.
- Test the user should be able to download the uploaded files.
- Test the email functionality of the system.
- Test the JavaScript script is properly working in different browsers (IE, Firefox, Chrome, Safari and Opera).
- Test to see what happens if a user deletes cookies while in the site.
- Test to see what happens if a user deletes cookies after visiting a site.
- Test all the data inside combo/list box is arranged in chronological order.

Compatibility Testing:

What is Compatibility testing?

- Compatibility testing is used to determine if your software is compatible with other elements with which it should operate, e.g. Browsers, Operating Systems, or hardware.

What is the purpose or Goal of Compatibility testing?



Faster. Easier. Better.

TRY qTEST FREE



- The purpose of Compatibility testing is to evaluate how well software performs in a particular environment, such as Operating Systems, hardware or software.

Compatibility Test Scenarios:

- Test the website in different browsers (IE, Firefox, Chrome, Safari and Opera) and ensure it displays properly.
- Test the HTML version being used is compatible with appropriate browser versions.
- Test the images display correctly in different browsers.
- Test the fonts are usable in different browsers.
- Test the JavaScript code is usable in different browsers.

- Test the Animated GIF's across different browsers.

Tool for Compatibility Testing:

Spoon.net: Spoon.net provides access to thousands of applications (Browsers) without any helps you to test your application on different browsers on one single machine.

Database Testing:

What is Database Testing?

- In Database testing backend records are tested which have been inserted through the web applications. The data which is displaying in the web application should match with the data in the Database.

To perform the Database testing, the tester should be aware of the below mentioned points:

- The tester should understand the functional requirements, business logic, application flow and design thoroughly.
- The tester should figure out the tables, triggers, store procedures, views and cursors used in the application.
- The tester should understand the logic of the triggers, store procedures, views and cursors.
- The tester should figure out the tables which get affected when insert, update and delete are performed through the web or desktop applications.

With the help of the above mentioned points, the tester can easily write the test scenarios for Database testing.

Test Scenarios for Database Testing:

- Verify the database name: The database name should match with the specifications.
- Verify the Tables, columns, column types and defaults: All things should match with the specifications.
- Verify whether the column allows a null or not.
- Verify the Primary and foreign key of each table.
- Verify the Stored Procedure:
 - Test whether the Stored procedure is installed or not.
 - Verify the Stored procedure name
 - Verify the parameter names, types and number of parameters.
 - Test the parameters if they are required or not.
 - Test the stored procedure by deleting some parameters
 - Test when the output is zero, the zero records should be affected.
 - Test the stored procedure by writing simple SQL queries.
 - Test whether the stored procedure returns the values

- Test the stored procedure with sample input data.
- Verify the behavior of each flag in the table.
- Verify the data gets properly saved into the database after the each page submission.
- Verify the data if the DML (Update, delete and insert) operations are performed.
- Check the length of every field: The field length in the back end and front end must be same.
- Verify the database names of QA, UAT and production. The names should be unique.
- Verify the encrypted data in the database.
- Verify the database size. Also test the response time of each query executed.
- Verify the data displayed on the front end and make sure it is same in the back end.
- Verify the data validity by inserting the invalid data in the database.
- Verify the Triggers.



QASymphony

Faster. Easier. Better.

TRY qTEST FREE



What is Security Testing?

Security Testing involves the test to identify any flaws and gaps from a security point of view

Test Scenarios for Security Testing:

1. Verify the web page which contains important data like password, credit card numbers, security question etc should be submitted via HTTPS (SSL).
2. Verify the important information like password, credit card numbers etc should display in a secure manner.
3. Verify password rules are implemented on all authentication pages like Registration, forgot password, change password.
4. Verify if the password is changed the user should not be able to login with the old password.
5. Verify the error messages should not display any important information.
6. Verify if the user is logged out from the system or user session was expired, the user should not be able to navigate the site.
7. Verify to access the secured and non secured web pages directly without login.
8. Verify the "View Source code" option is disabled and should not be visible to the user.
9. Verify the user account gets locked out if the user is entering the wrong password several times.
10. Verify the cookies should not store passwords.
11. Verify if, any functionality is not working, the system should not display any application, security information. Instead, it should display the custom error page.
12. Verify the [SQL](#) injection attacks.
13. Verify the user roles and their rights. For Example The requestor should not be able to access admin page.

14. Verify the important operations are written in log files, and that information should be tr
15. Verify the session values are in an encrypted format in the address bar.
16. Verify the cookie information is stored in encrypted format.
17. Verify the application for Brute Force Attacks

What is Performance Testing?

Performance testing is conducted to evaluate the compliance of a system or component with performance requirements.

General Test scenarios:

- To determine the performance, stability and scalability of an application under different
- To determine if the current architecture can support the application at peak user levels.
- To determine which configuration sizing provides the best performance level.
- To identify application and infrastructure bottlenecks.
- To determine if the new version of the software adversely had an impact on response time
- To evaluate product and/or hardware to determine if it can handle projected load volume

How to do Performance testing? By Manual Testing or by Automation

Practically it is not possible to do the performance testing manually because of some drawbacks

- More number of resources will be required.
- Simultaneous actions are not possible.
- Proper system monitoring is not available.
- Not easy to perform the repetitive task.

Hence to overcome the above problems we should use Performance testing tool. Below are the popular testing tools.

- Apache JMeter
- Load Runner
- Borland Silk Performer.
- Rational Performance Tester
- WAPT
- NEO LOAD

[< Prev](#)

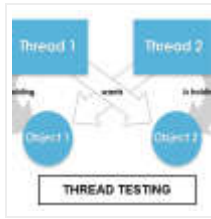
YOU MIGHT LIKE:

SAP - CRM



SAP CRM Business Transaction Process

SOFTWARE TESTING



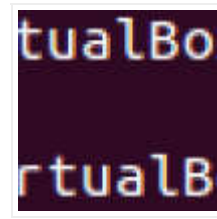
Thread Testing: Types, Strategy & Multithreading

SOFTWARE TESTING



Conformance Testing (Compliance Testing) - Complete Guide

BIGDATA



Hadoop & Mapreduce Examples: Create your First Program

About

[About us](#)
[Advertise with Us](#)
[Jobs](#)
[Privacy Policy](#)

Contact Us

[Contact us](#)
[FAQ](#)
[Write For Us](#)

Follow Us

Certificat

[ISTQB Cert](#)
[MySQL Cer](#)
[QTP Certifi](#)
[Testing Cer](#)
[CTAL Exam](#)