

# Introduction

The Internet has become the de facto standard means by which most people interact with a vast majority of software applications. This move away from desktop computing to server-side computing with HTML as the front-end technology has created a whole new class of applications to be built and delivered to customers.

However, it has also created a new set of challenges in terms of developing and testing these applications. It is now assumed your customer may be sitting thousands of miles away and working in a language you are not familiar with.

A second aspect of modern software development methodology is to start testing the software early and often. Most software developers do not follow the cascading model of development whereby testing only starts after development ends. Further, developers are being increasingly encouraged to create vast arrays of unit tests to ensure quality.

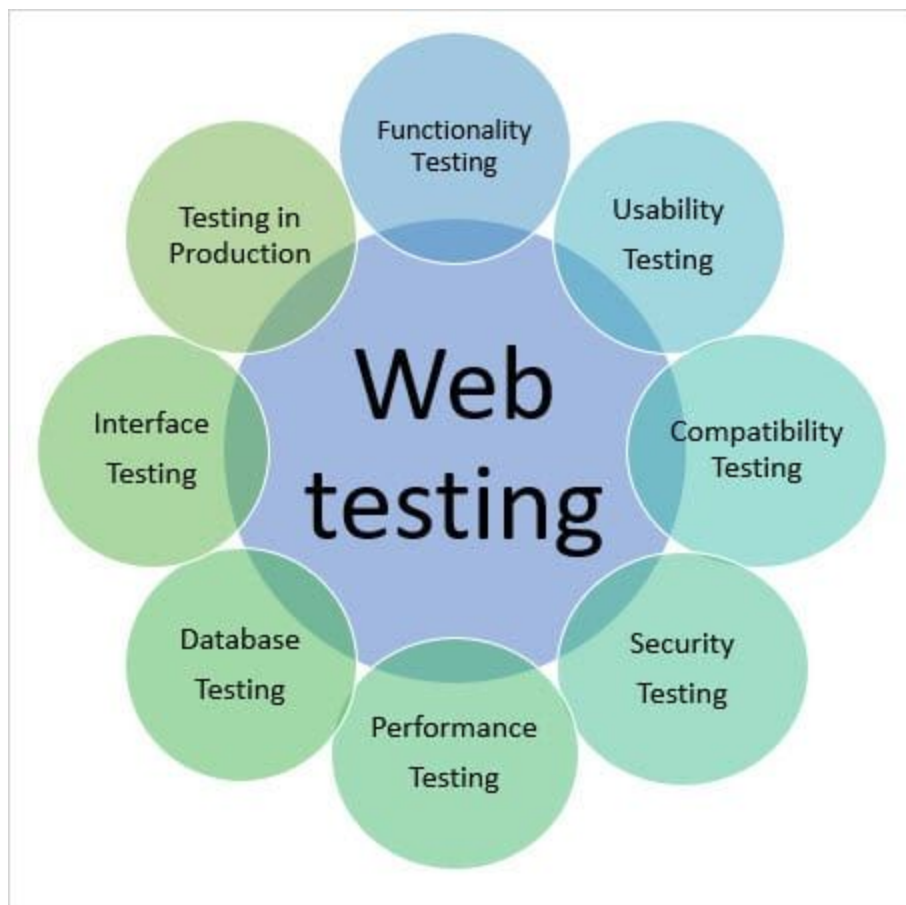
# 1.0.0 Web Testing

Web testing is a software testing practice to test websites or web applications for potential bugs. It's a complete testing of web-based applications before making live.

A web-based system needs to be checked completely from end-to-end before it goes live for end users.

By performing website testing, an organization can make sure that the web-based system is functioning properly and can be accepted by real-time users.

The UI design and functionality are the captains of website testing.



## **1.1.0 Need For Web Test**

All software is designed and developed to meet and satisfy certain "functional" needs. A functional need may be technical, business, or process. A key component of testing a software is to ensure that it functionally does what it is supposed to do. Functional Testing is the process by which expected behavior of an application is tested.

Most software undergo changes over a period of time. These changes may occur during initial development of the software, or after the release of the software. Any change to the software carries a potential of errors or bugs being introduced. Regression Testing is the process of frequently re-testing a software to ensure known existing behavior or functionality is not impacted. By impacted, we mean either modified or broken. If behavior is modified, the developer or tester must ensure that it is expected. Otherwise, it is called a "Regression".

As the size and complexity of applications have grown over the years, it has become increasingly difficult to ensure functional compliance and to ensure no regressions over time. Even a small size project can generate excessive permutations and combinations of test cases that are too great for a human to test consistently and continually. Hence, it is critical to find a testing system which can be automated.

The AppPerfect Web Test was created to assist users with such challenges. Since the mechanics of Functional Testing and Regression Testing are similar, they can be addressed with a single testing tool. AppPerfect Web Test provides support for "recording" a test once and then "replaying" it when desired without additional manual intervention in an automated manner.

## **1.2.0 Web Test Language**

- a. .NET
- b. Javascript
- c. C++
- d. Python

## **1.3.0 Web Test Tools**

- a. Selenium
- b. Scrapy
- c. Beautiful Soup

# 2.0.0 Web Testing Checklist

- 2.1 Functionality Testing
- 2.2 Usability testing
- 2.3 Interface testing
- 2.4 Compatibility testing
- 2.5 Performance testing
- 2.6 Security testing

## 2.1.0 Functionality Testing

Test for – all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages, Cookie testing, etc.

```
import scrapy
from ..items import WebtestItem

class QuoteSpiderSpider(scrapy.Spider):
    name = 'webtests'
    page=2
    start_urls = [
        'http://quotes.toscrape.com/page/1/'
    ]

    def parse(self, response):
        items=WebtestItem()

        title=response.css('.text::text').extract()
        author=response.css('.author::text').extract()
        tag=response.css('.tag::text').extract()
        link=response.css('.header-box a, .tag').css('::attr(src)').get()

        items['title']= title
        items['author']=author
        items['tag']=tag
        items['link']=link

        yield items

    next_page = 'http://quotes.toscrape.com/page/' + str(QuoteSpiderSpider.page) + '/'
    if QuoteSpiderSpider.page <= 11:
        QuoteSpiderSpider.page += 1
        yield response.follow(next_page, callback=self.parse)
```

### 2.1.1 Check all the links

- Test the outgoing links from all the pages to the specific domain under test.
- Test all internal links.
- Test links jumping on the same pages.
- Test links used to send email to admin or other users from web pages.
- Test to check if there are any orphan pages.
- Finally, link checking includes, check for broken links in all the above-mentioned links.

→ ↻ ⓘ Not secure | quotes.toscrape.com/page/2/

```
class QuoteSpider(scrapy.Spider):
    name = 'quotes'
    page_number=2
    #allowed_domains = ['quotes.toscrape.com']
    start_urls = ['http://quotes.toscrape.com/page/1/']
```

## 2.1.2 Test forms on all pages

Forms are an integral part of any website. Forms are used for receiving information from users and to interact with them. So what should be checked in these forms?

- First, check all the validations on each field.
- Check for default values of the fields.
- Wrong inputs in the forms to the fields in the forms.
- Options to create forms if any, form delete, view or modify the forms.

Let's take an example of the search engine project I am currently working on, in this project we have advertiser and affiliate signup steps. Each sign-up step is different but dependent on other steps.

So sign up flow should get executed correctly. There are different field validations like email Ids, User financial info validations, etc. All these validations should get checked in manual or automated web testing.

## Quotes to Scrape

[Login](#)

"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."

by [Albert Einstein](#) (about)

Tags: [change](#) [deep-thoughts](#) [thinking](#) [world](#)

"It is our choices, Harry, that show what we truly are, far more than our abilities."

by [J.K. Rowling](#) (about)

Tags: [abilities](#) [choices](#)

"There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle."

by [Albert Einstein](#) (about)

Tags: [inspirational](#) [life](#) [live](#) [miracle](#) [miracles](#)

"The person, be it gentleman or lady, who has not pleasure in a good novel, must be intolerably stupid."

by [Jane Austen](#) (about)

Tags: [aliteracy](#) [books](#) [classic](#) [humor](#)

"Imperfection is beauty, madness is genius and it's better to be absolutely ridiculous than absolutely boring."

by [Marilyn Monroe](#) (about)

Tags: [be-yourself](#) [inspirational](#)

## Top Ten tags

[love](#)

[inspirational](#)

[life](#)

[humor](#)

[books](#)

[reading](#)

[friendship](#)

[friends](#)

[cute](#)

[quote](#)

### 2.1.3 Cookies Testing

Cookies are small files stored on the user's machine. These are basically used to maintain the session- mainly the login sessions. Test the application by enabling or disabling cookies in your browser options.

Test if the cookies are encrypted before writing to the user machine. If you are testing the session cookies (i.e. cookies that expire after the session ends) check for login sessions and user stats after the session ends. Check the effect on application security by deleting the cookies. (I will soon write a separate article on cookie testing as well)

### 2.1.4 Validate your HTML/CSS

If you are optimizing your site for Search engines then HTML/CSS validation is the most important one. Mainly validate the site for HTML syntax errors. Check if the site is crawlable to different search engines.

#### Quotes to Scrape

Login

*"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."*

by **Albert Einstein** (about)

Tags: [change](#) [deep-thoughts](#) [thinking](#) [world](#)

*"It is our choices, Harry, that show what we truly are, far more than our abilities."*

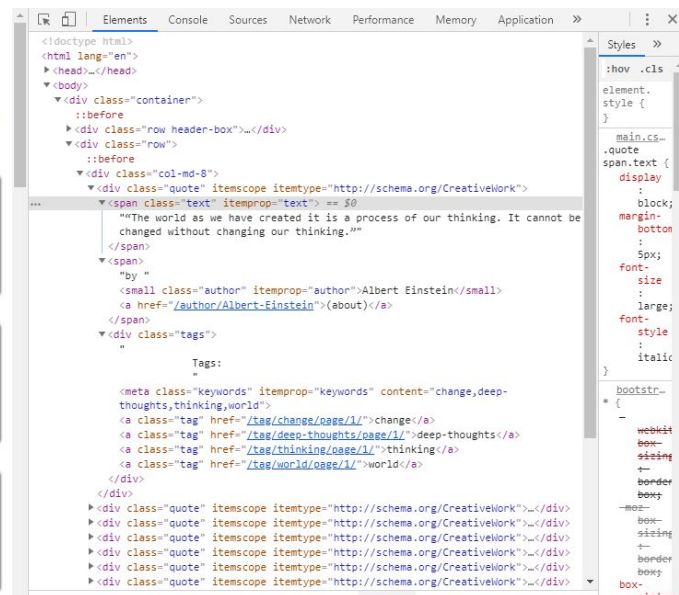
by **J.K. Rowling** (about)

Tags: [abilities](#) [choices](#)

*"There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle."*

by **Albert Einstein** (about)

Tags: [inspirational](#) [life](#) [live](#) [miracle](#) [miracles](#)



### 2.1.5 Database Testing

Data consistency is also very important in a web application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality.

Check if all the database queries are executing correctly, data is retrieved and also updated correctly. More on database testing could be load on DB, we will address this in web load or performance testing below.

```
#
# Don't forget to add your pipeline to the ITEM_PIPELINES setting
# See: https://docs.scrapy.org/en/latest/topics/item-pipeline.html

import sqlite3

class WebtestPipeline(object):

    def __init__(self):
        self.create_connection()
        self.create_table()

    def create_connection(self):
        self.connn= sqlite3.connect('myquotes.db')
        self.curre= self.connn.cursor()

    def create_table(self):
        self.curre.execute("""create table quotes_tb(
                                title text,
                                author text,
                                tag text,
                                link text
                            )""")

    def process_item(self, item, spider):
        self.store_db(item)
        return item

    def store_db(self, item):
        self.curre.execute("""insert into quotes_tb values (?, ?, ?)""", (
            item['title'][0],
            item['title'][0],
            item['title'][0],
            item['title'][0]
        ))
        self.connn.commit()
```

## 2.2.0 Usability Testing

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.

- Ease of learning
- Navigation
- Subjective user satisfaction
- General appearance

### 2.2.1 Test for Navigation

Navigation means how a user surfs the web pages, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages.

### 2.2.2 Usability Testing includes the following

- The website should be easy to use.
- The instructions provided should be very clear.
- Check if the instructions provided are perfect to satisfy its purpose.
- The main menu should be provided on each page
- It should be consistent enough.

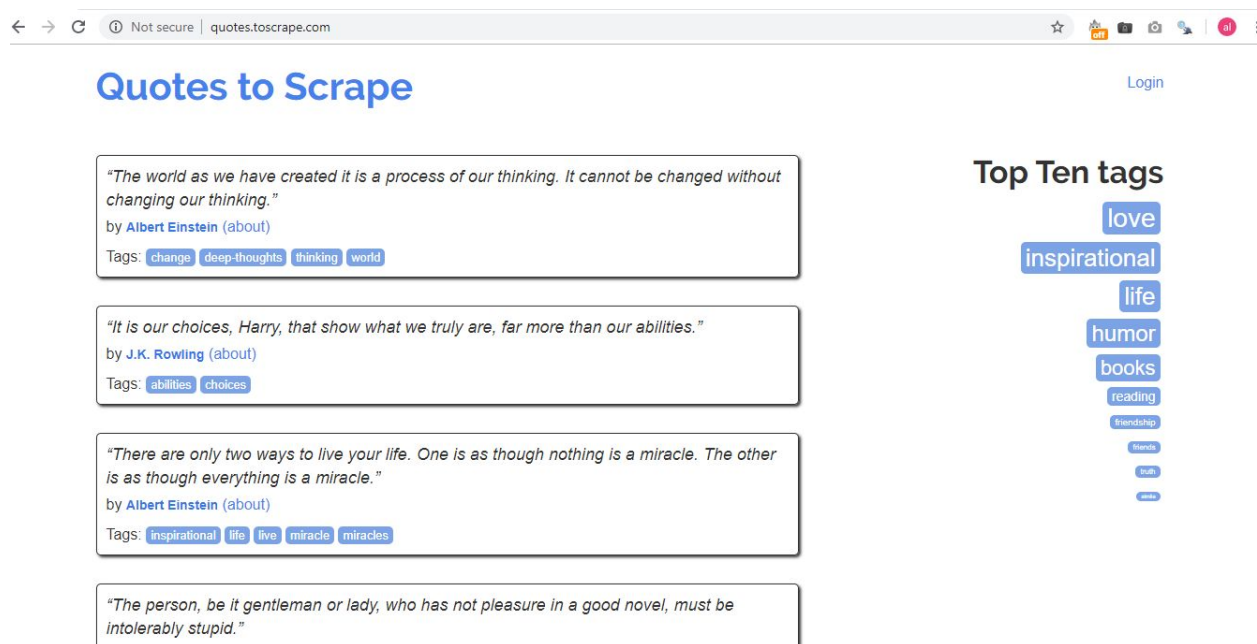
### 2.2.3 Content Checking

Content should be logical and easy to understand. Check for spelling errors. The usage of dark colors annoys the users and should not be used in the site theme.

You can follow some standard colors that are used for web pages and content building. These are the commonly accepted standards like what I mentioned above about annoying colors, fonts, frames, etc.

Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes.

These are some of the basic important standards that should be followed in web development. Your task is to validate all for UI testing.



### 2.2.4 Other user information for user help

Like the search option, the sitemap also helps files, etc. The sitemap should be present with all the links on websites with a proper tree view of navigation. Check for all links on the sitemap.

"Search in the site" option will help users to find content pages that they are looking for easily and quickly. These are all optional items and if present they should be validated.



## 2.3.0 Interface Testing

In web testing, the server-side interface should be tested. This is done by verifying that communication is done properly. Compatibility of the server with software, hardware, network, and the database should be tested.

**The main interfaces are:**

- Web server and application server interface
- Application server and Database server interface.

Check if all the interactions between these servers are executed and errors are handled properly. If the database or web server returns an error message for any query by application server then the application server should catch and display these error messages appropriately to the users. Check what happens if the user interrupts any transaction in-between? Check what happens if the connection to the web server is reset in between?

## 2.4.0 Compatibility Testing

The compatibility of your website is a very important testing aspect. See which compatibility test to be executed:

- Browser compatibility
- Operating system compatibility
- Mobile browsing
- Printing options

### 2.4.1 Browser Compatibility

In my web-testing career, I have experienced this as most influencing part of website testing.

Some applications are very dependent on browsers. Different browsers have different configurations and settings that your web page should be compatible with.

Your website coding should be a cross-browser platform compatible. If you are using java scripts or AJAX calls for UI functionality, performing security checks or validations then give more stress on browser compatibility testing of your web application.

Test web application on different browsers like Internet Explorer, Firefox, Netscape Navigator, AOL, Safari, Opera browsers with different versions.

## 2.4.2 OS Compatibility

Some functionality in your web application is that it may not be compatible with all operating systems. All new technologies used in web development like graphic designs, interface calls like different API's may not be available in all Operating Systems. Hence test your web application on different operating systems like Windows, Unix, MAC, Linux, Solaris with different OS flavors.

## 2.4.3 Mobile Browsing

We are in the new technology era. So in future Mobile browsing will rock. Test your web pages on mobile browsers. Compatibility issues may be there on mobile devices as well.

## 2.4.4 Printing Options

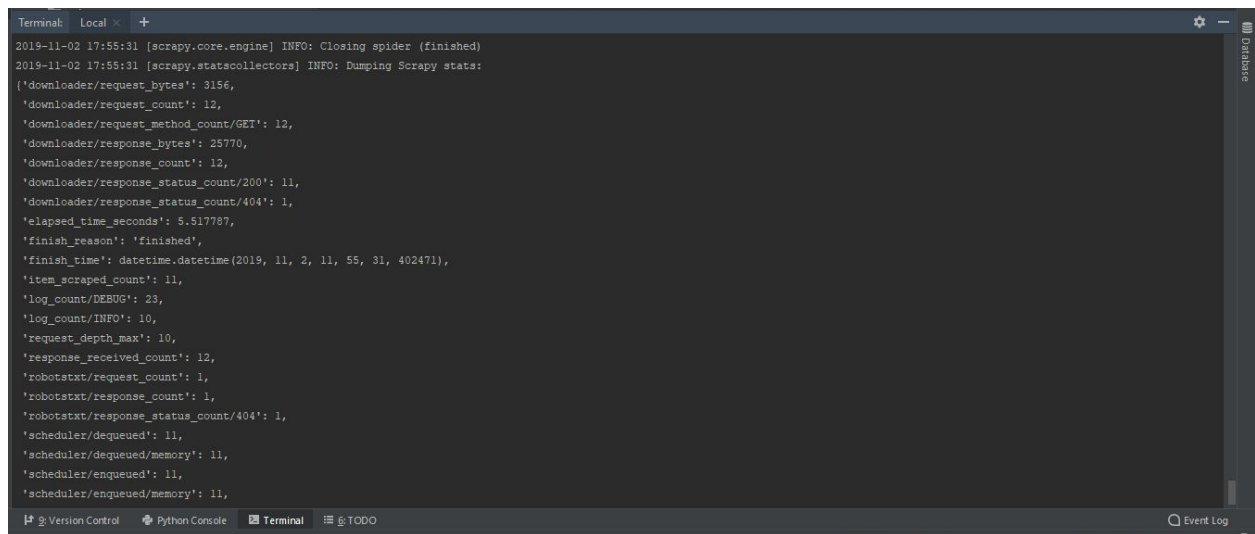
If you are giving page-printing options then make sure fonts, page alignment, page graphics, etc. are getting printed properly. Pages should fit the paper size or as per the size mentioned in the printing option

## 2.5.0 Performance Testing

The web application should sustain to heavy load. Web performance testing should include:

- Web Load Testing
- Web Stress Testing

Test application performance on different internet connection speeds.



```
Terminal: Local x +
2019-11-02 17:55:31 [scrapy.core.engine] INFO: Closing spider (finished)
2019-11-02 17:55:31 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
{'downloader/request_bytes': 3156,
 'downloader/request_count': 12,
 'downloader/request_method_count/GET': 12,
 'downloader/response_bytes': 25770,
 'downloader/response_count': 12,
 'downloader/response_status_count/200': 11,
 'downloader/response_status_count/404': 1,
 'elapsed_time_seconds': 5.517787,
 'finish_reason': 'finished',
 'finish_time': datetime.datetime(2019, 11, 2, 11, 55, 31, 402471),
 'item_scraped_count': 11,
 'log_count/DEBUG': 23,
 'log_count/INFO': 10,
 'request_depth_max': 10,
 'response_received_count': 12,
 'robotstxt/request_count': 1,
 'robotstxt/response_count': 1,
 'robotstxt/response_status_count/404': 1,
 'scheduler/dequeued': 11,
 'scheduler/dequeued/memory': 11,
 'scheduler/enqueued': 11,
 'scheduler/enqueued/memory': 11,
}
```

### **2.5.1 Web Load Testing**

You need to test if many users are accessing or requesting the same page. Can the system sustain in peak load times? The site should handle many simultaneous user requests, large input data from users, simultaneous connection to DB, heavy load on specific pages, etc.

### **2.5.2 Web Stress Testing**

Generally stress means stretching the system beyond its specified limits. Web stress testing is performed to break the site by giving stress and checked as to how the system reacts to stress and how it recovers from crashes. Stress is generally given on input fields, login and sign up areas.

In web performance, testing website functionality on different operating systems and different hardware platforms is checked for software and hardware memory leakage errors.

Performance testing can be applied to understand the web site's scalability or to benchmark the performance in the environment of third-party products such as servers and middleware for potential purchase.

### **2.5.3 Connection Speed**

Tested on various networks like Dial-Up, ISDN, etc

#### **Load**

- i. What is the no. of users per time?
- ii. Check for peak loads and how the system behaves
- iii. A large amount of data accessed by the user

#### **Stress**

- i. Continuous Load
- ii. Performance of memory, CPU, file handling, etc

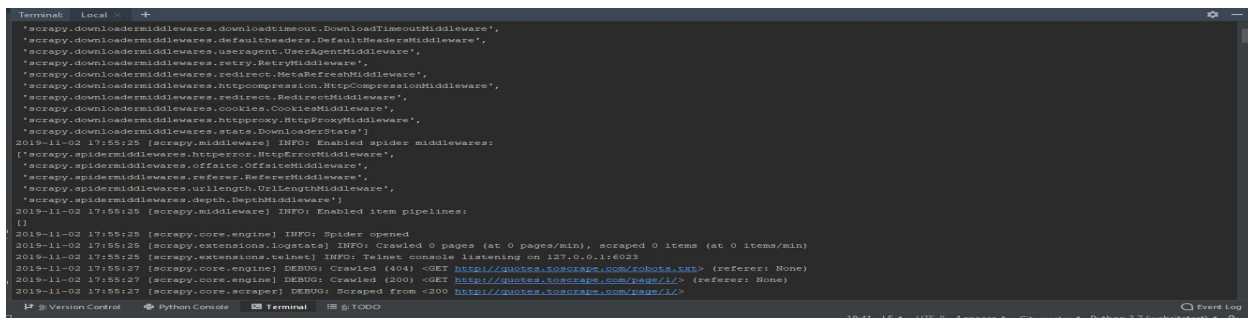
## 2.6.0 Security Testing

Following are some of the test cases for web security testing:

- Test by pasting the internal URL directly into the browser address bar without login. Internal pages should not open.
- If you are logged in using username and password and browsing internal pages then try changing URL options directly. I.e. If you are checking some publisher site statistics with publisher site ID= 123. Try directly changing the URL site ID parameter to different site ID which is not related to the logged-in user. Access should be denied for this user to view other's stats.
- Try some invalid inputs in input fields like login username, password, input text boxes, etc. Check the system's reaction to all invalid inputs.
- Web directories or files should not be accessible directly unless they are given download option.
- Test the CAPTCHA for automating script logins.
- Test if SSL is used for security measures. If it is used, the proper message should get displayed when users switch from non-secure HTTP:// pages to secure HTTPS:// pages and vice versa.
- All transactions, error messages, security breach attempts should get logged in log files somewhere on the web server.

The primary reason for testing the security of a website is to identify potential vulnerabilities and subsequently repair them.

- Network Scanning
- Vulnerability Scanning
- Password Cracking
- Log Review
- Integrity Checkers
- Virus Detection



```
Terminal Local +
"scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware",
"scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware",
"scrapy.downloadermiddlewares.useragent.UserAgentMiddleware",
"scrapy.downloadermiddlewares.retry.RetryMiddleware",
"scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware",
"scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware",
"scrapy.downloadermiddlewares.redirect.RedirectMiddleware",
"scrapy.downloadermiddlewares.cookies.CookiesMiddleware",
"scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware",
"scrapy.downloadermiddlewares.state.DownloaderState"]
2019-11-02 17:55:25 [scrapy.middleware] INFO: Enabled spider middlewares:
["scrapy.spidermiddlewares.httperror.HttpErrorMiddleware",
"scrapy.spidermiddlewares.offsite.OffsiteMiddleware",
"scrapy.spidermiddlewares.referrer.ReferrerMiddleware",
"scrapy.spidermiddlewares.urllength.UrlLengthMiddleware",
"scrapy.spidermiddlewares.depth.DepthMiddleware"]
2019-11-02 17:55:25 [scrapy.middleware] INFO: Enabled item pipelines:
[]
2019-11-02 17:55:25 [scrapy.core.engine] INFO: Spider opened
2019-11-02 17:55:25 [scrapy.extensions.logstats] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)
2019-11-02 17:55:25 [scrapy.extensions.telnet] INFO: Telnet console listening on 127.0.0.1:6023
2019-11-02 17:55:27 [scrapy.core.engine] DEBUG: Crawled (404) <GET http://quotes.toscrape.com/robots.txt> (referrer: None)
2019-11-02 17:55:27 [scrapy.core.engine] DEBUG: Crawled (200) <GET http://quotes.toscrape.com/page/1/> (referrer: None)
2019-11-02 17:55:27 [scrapy.core.spider] DEBUG: Scraped from <200 http://quotes.toscrape.com/page/1/>
```

# 3.0.0 Types Of Web Testing

A website is classified into many types, it is about 20 types. All these are shrinking under static and dynamic type. Among them let's discuss 4 types and its testing methods in a detailed manner. Before that, I just want to bullet those types.

- Simple static website testing
- Dynamic web application testing
- E-commerce website testing
- Mobile website testing

## 3.1.1 Simple Static Website

A simple static website will display the same content for all visitors who are visiting the website at different times. It is also known as an informational website. In a static website, the only developer can do changes that too in code only. This type of website will not have any major functionalities and it purely depends on UI design.

Testing a simple static website is very easy, you have to consider only a few things while testing. Some of them are mentioned below.

### Quotes to Scrape

Login

*"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."*

by [Albert Einstein](#) (about)

Tags: [change](#) [deep-thoughts](#) [thinking](#) [world](#)

*"It is our choices, Harry, that show what we truly are, far more than our abilities."*

by [J.K. Rowling](#) (about)

Tags: [abilities](#) [choices](#)

*"There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle."*

by [Albert Einstein](#) (about)

Tags: [inspirational](#) [life](#) [live](#) [miracle](#) [miracles](#)

*"The person, be it gentleman or lady, who has not pleasure in a good novel, must be intolerably stupid."*

by [Jane Austen](#) (about)

Tags: [literacy](#) [books](#) [classic](#) [humor](#)

*"Imperfection is beauty, madness is genius and it's better to be absolutely ridiculous than absolutely boring."*

by [Marilyn Monroe](#) (about)

Tags: [be-yourself](#) [inspirational](#)

### Top Ten tags

[love](#)

[inspirational](#)

[life](#)

[humor](#)

[books](#)

[reading](#)

[friendship](#)

[travel](#)

[kids](#)

[style](#)

### 3.1.2 Points to Remember

- a. Testing the GUI design is a must because a static website purely depends on it. You need to compare the approved PSD files with web page developed. Check all the elements in the design should present on the developed page
- b. The other part of GUI design is to check the font size, font style, spacing, and color everything has been reproduced.
- c. Secondly, you need to check the links (page links) whether it is properly working or not? And also find, is there any broken link?
- d. Verify the spelling and content in all web pages by comparing the content given by the client.
- e. In some cases image will not display properly, it may break or sometimes images gets duplicated, wrong images may display. It has to be checked keenly. Because for a static website, only content and images will give lives.
- f. Check the scroll bar carefully, in my experience, I have faced issues with the scrollbar. The issue you will face is unwanted scrolling appears or scroll gets hidden (it may hide the contents). The above issues are applicable to both horizontal and vertical scrolls.
- g. If there is a contact form check it is working properly by sending some dummy messages.
- h. Check whether it is an error-free web page, validate it with W3 validator or other related software.

Some constant things to be checked in a static website,

- Check favicon is present on the tab bar
- URL should contain the correct page title
- If copyright information is there, it should be displayed
- If there is a contact form, Captcha is a must. [It prevents junk email]
- Check the loading speed of the website. [A static website should not take much time for loading]. If a gif image is used while loading then track its functionality

Apart from these, there are huge things that have to be tested at the backend of every website that is system testing, security testing, interface testing, compatibility testing, and performance testing, etc. For this, you need to have technical knowledge. In a simple static website, you will not find more functionalities if there you need to do functionality testing too.

### 3.2.0 Dynamic Web Application (CMS Website)

It is the type where the user can update and change their website content regularly. From here I am going to use the word “web application testing” instead of dynamic website testing. The web application is a *combination of front-end and back-end programming*.

The front-end will be HTML and CSS whereas back-end uses programming languages like PHP, Javascript, and ASP etc. With this backend, user/client can add or change the content on the website.

Testing a web application is not easy than testing a static website but not much difficult than testing an e-commerce website. Functionality testing is the most important thing to be performed while testing a web application. The web application may contain much-complicated functionality so tester needs to be very careful while testing.

There are two different types of web applications are there, one is no action will be carried out by the user in front-end (i.e. only back-end changes will reflect in front-end) the other is end-user will work in front-end itself (for example login, signup, newsletter subscription, and other similar actions). So testing should be done according to it.

#### Points to Remember

The points which I mentioned in static website testing are to be included while testing a web application also. In addition to that, the following things are to be noted.

- a. In GUI section, *tooltip is compulsory* for all fields and buttons, field alignment (spacing) should be done properly, disabled field/ buttons should be greyed out, fields/ buttons should be in standard format as in SRS, error message should be displayed if something goes wrong, pop-up message should only display at the center of the web page, drop-down menu should not be truncated.

Tab shortcut key should work in all fields and more.

- a. In functionality section, if your web application is having login or sign up functionality then check the *mandatory field validation*, form validation (i.e. number fields should accept only numbers, not alphabets ), character restriction on fields (i.e. only these many characters can be entered) Special characters and negative numbers restriction on fields, testing the email functionality, testing the document upload (i.e. only *specified document type can be uploaded*), timeout functionality, sorting functionality, javascript is working on compatible browsers etc should be tested.

- b. When coming to back-end functionality section, test image uploading for broken images, text entering in the fields is working or not. Back-end update should *reflect on front-end, database testing* (i.e. whether you can add new fields or deleting unwanted fields) all these things are to be performed. Special characters and negative numbers restriction on fields, testing the email functionality, testing the document upload (i.e. only *specified document type can be uploaded*), timeout functionality, sorting functionality, javascript is working on compatible browsers etc should be tested.
- c. Performance is not necessary for a web application (dynamic website) since it has very less content. If you need you can do with the tools with which you are familiar. Pick-up some standard online performance tool, if you want to do simple performance testing.

### 3.3.0 E-commerce Website

An e-commerce website is somewhat complicated when compared to the above two. The tester needs to be very cautious while testing an e-commerce site. There are huge things to be checked in e-commerce sites out of them I just cover some of my experienced issues on e-commerce website testing.

In the GUI section, you need to check all the features as in SRS and the same with the functionality. The functionality will be almost the same for all commercial websites. Functionality-wise you need to check all pages such as the main page (includes featured products, special offers display, log in details, search functionality) product detail page, category page, placing an order, payment gateway everything has to be tested.

The screenshot shows the Amazon Bangladesh homepage with a search bar at the top. Below the search bar, there's a navigation bar with links like 'Today's Deals', 'Help', 'Registry', 'Gift Cards', 'Sell', and 'Your Amazon.com'. The main content area displays search results for 'Accessories & Supplies'. On the left, there's a 'Department' sidebar with a list of categories under 'Electronics' and 'Accessories & Supplies'. The main product list shows two items: 'AmazonBasics AA 1.5 Volt Performance Alkaline Batteries - Pack of 48' and 'AmazonBasics AAA 1.5 Volt Performance Alkaline Batteries - Pack of 36'. Both products are marked as 'Best Seller' and have star ratings and review counts. The first product is priced at \$13.97 (\$0.29/Count) and the second at \$9.98 (\$0.28/Count). Both offer a 5% discount with 'Subscribe & Save' and ship to Bangladesh.



## Points to Remember

- a. Check if the shopping cart is getting updated when you buy or increase the quantity. Check this functionality in all the pages and circumstances
- b. Check if special coupons and *offers are applied to correct orders* and you see the discounted price is displaying or not
- c. Sometimes while updating a single product it will get multiplied by considering the number of variations in the product. So check whether the single product is displayed and its variations are displayed correctly. (I faced this problem)
- d. Check if the filter option whether is working exactly. If filtering is been done, based on the category & pricing chosen?
- e. While sign up, super validation should be done. Only the new user can sign up.
- f. If an existing user, added a product to the shopping basket, the wishlist section during their previous login should be saved and displayed during the next login too.
- g. Compare products should work by comparing the products based on some specifications assigned in the back-end.
- h. Check whether the Currency converter is working fine. Based on the country chosen the currency converter should display the relevant price and tax rates.
- i. Generally many Plug-ins are used in an e-commerce (WordPress & similar) website, you need to be very careful. The plug-in installation may conflict or affect any other major functionality. So follow up with the plug-ins installation and its usage
- j. Check whether the social sharing option is working on the individual product or not.
- k. Shipping cost should be generated based on the region selected. And also check the tax rate generation. (It may cause some legal problems, during the end-users purchase).
- l. Payment gateway should work only if valid card details are given. Validation should apply to the Card number and CVV code number. [It is better to keep validation on the card number field itself]
- m. Email generation on each and every process during purchase should happen (sign up, product ordering, payment successful, order canceled, order received and other email triggers if any).

### 3.4.0 Mobile Website

First of all, let's be clear about a mobile website. Generally people think both a mobile website and mobile application to be the same, but in reality, a mobile website is developed with HTML pages and can be viewed only with an internet connection. But the mobile app is nothing but an application that can be downloaded and used later without an internet connection. Here many of us get confused and raise a question What is the difference between mobile website & responsive website?

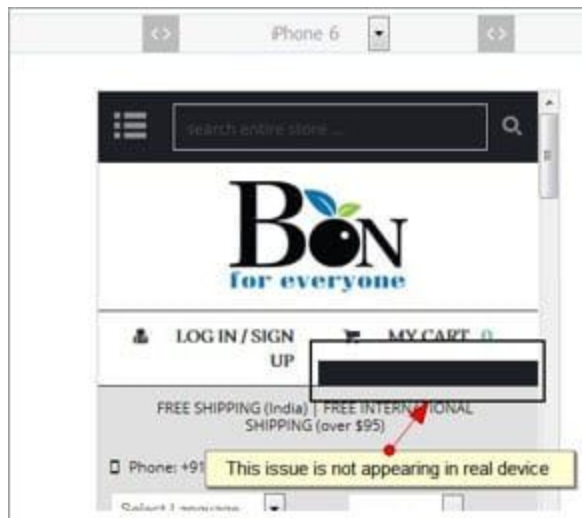
A responsive website means making the content fit into the mobile device size instead of creating a version whereas a mobile website is creating a new version that is not a reflection desktop version. In the mobile website, you will have only limited pages, and unwanted functionalities are removed here.

Testing a mobile website is somewhat tedious rather than other types of websites. It will have separate designs and you need to be careful while testing the functionalities.

#### Points to Remember:

#### Important points to consider while testing a mobile website:

- Usually, we will use an emulator for testing a mobile website and we can get ideal results but I always prefer you to test on real devices. I have faced many issues when I tested in real devices [Especially apple devices]. Real device specifications may conflict with the web pages developed.



**This image explains about simulator testing and the backline issue appearing in it.**

- GUI & usability testing are more important as it is not the reflection of the desktop version.
- Performance is another important factor to be considered for mobile website testing. Performance-related issues can be tracked when you test on the real devices.
- Check whether browsing normal web links from mobile is getting triggered by a mobile link.
- Check page scrolling, page navigation, text truncation, etc on the mobile website.
- 

### **Best Web Testing Tools**

There is a wide range of testing tools that are available for web app testing.

=> Check this comprehensive list of Most Popular Web Application Testing Tools.

#### **3.4.1 Points to be considered while testing a Website**

The websites are essentially client/server applications – with web servers and ‘browser’ clients.

Consideration should be given to the interactions between HTML pages, TCP/IP communications, Internet connections, firewalls, applications that run in web pages (such as applets, javascript, plug-in applications) and applications that run on the server-side (such as CGI scripts, database interfaces, logging applications, dynamic page generators, asp, etc.).

Additionally, there are a wide variety of servers and browsers with various versions of each. They include small but sometimes significant differences between them in terms of variations in connection speeds, rapidly changing technologies, and multiple standards & protocols. The end result of which testing for websites can become a major ongoing effort.

### **3.4.2 Sample test scenarios for testing a web application**

**Few other considerations to be included while testing a website are given below.**

- What is the expected load on the server (e.g., number of hits per unit time)?
- What kind of performance is required under each load condition (such as web server response time, database query response times)?
- What kind of tools will be required for performance testing (such as web load testing tools, other tools already in-house that can be adapted, web robot downloading tools, etc.)?
- Who is the target audience? What kind of browsers will they be using? What kind of connection speeds will they be using? Are they intra-organizations (thus likely with high connection speeds and similar browsers) or Internet-wide (thus with a wide variety of connection speeds and browser types)?
- What kind of performance is expected from the client-side (e.g., how fast should pages appear, how fast should animations, applets, etc. load and run)?
- Will the downtime for server and content maintenance/upgrades be allowed? If so, then how much?
- What kind of security (firewalls, encryption, passwords, etc.) will be required and what is it expected to do? How can it be tested?
- How reliable are the site's Internet connections required to be? And how does that affect the backup system or redundant connection requirements and testing?
- What process will be required to manage updates to the web site's content?
- What are the requirements for maintaining, tracking, and controlling page content, graphics, links, etc.?
- Which HTML specification will be adhered to? How strictly? What variations will be allowed for targeted browsers?
- Will there be any standard requirements for page appearance and/or graphics throughout a site or parts of a site??
- How will internal and external links be validated and updated? And how often? will it happen?
- Can testing be done on the production system, or will a separate test system be required?
- How are browser caching, variations in browser option settings, dial-up connection variability, and real-world internet 'traffic congestion' problems to be accounted for testing?

- How extensive or customized are the server logging and reporting requirements; are they considered as an integral part of the system and do they require testing?
- How are CGI programs, applets, javascript,python, ActiveX components, etc. to be maintained, tracked, controlled, and tested?
- Pages should be 3-5 screens max unless the content is highly focused on a single topic. If larger, provide internal links within the page.
- The page layouts and design elements should be consistent throughout a site so that it's clear to the user that they are still on a site.
- Pages should be as browser-independent as possible, or pages should be provided or generated based on the browser type.
- All pages should have links external to the page; there should be no dead-end pages.
- The page owner, revision date, and a link to a contact person or organization should be included on each page.

## 4.0.0 Web Testing FAQs

Below mentioned should be the various questions coming to a tester's mind while thinking of a website that is already developed and can be exposed to the public

- Is the website functioning as expected?
- Will the end-user find the website easy to browse?
- Is the website accessible on different devices possessed by end-users?
- Is the website secured enough?
- Is the website performance up to the mark?
- Is the data entered on a website stored accurately and persist across sessions?
- Is the website integrated well with other interfaces in the workflow?
- Will the website perform as expected even after going live?

To answer these questions, different testing techniques have been identified that can be used to test a web application.

*Let's take an example of an e-commerce website that has been recently released to the QA team for testing.*

*We'll go through each one of the above-specified questions in detail to understand the scope of the test and see how website testing can be performed.*

## Is the website functioning as expected?

confirm that the website is functioning well, QA needs to perform functional testing. During functional testing, different features of an application needs to be validated against the requirements mentioned in the functional specification document. Below are a few generic scenarios, a QA is expected to cover while performing functional testing of any website even if they are not mentioned in functional specifications:

- User navigation to different pages of the website and completing the end-to-end workflow
- If the user can select/deselect checkboxes
- If the user can select values from Dropdown fields
- If the user can select/deselect Radio buttons
- Different navigation buttons like Submit, Next, Upload etc. buttons are working well
- Calendars are loading properly and allowing the user to select a date
- Calculations are happening as implemented
- Search functionality is working if any
- Correct Information display
- Various internal & external links to other pages
- Correct Tab Order of the fields on web pages
- Mandatory and Optional fields should be verified for the positive and negative inputs
- Default values for each web field should be verified
- Email functionality is implemented for some action on the website

It's important for websites to be compatible with search engines. Hence we should review websites for HTML syntax correctness, format & compliance standards like WS-I, ISO & ECMA.

Considering cookies, which are used to maintain login sessions, the website should be tested by enabling/disabling cookies or by using the mismatched domain. Testing can also be performed across sessions by resetting cookies to bring browsers back to the vanilla state.

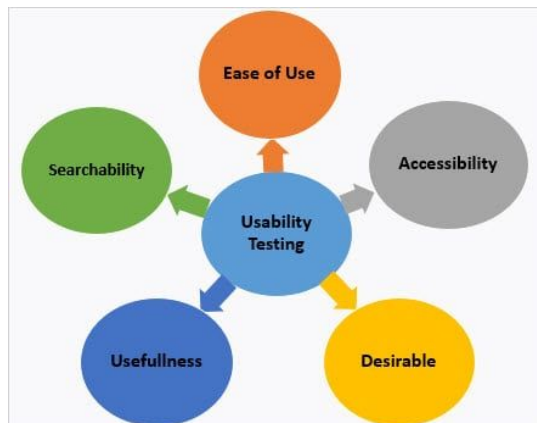
QA should also validate that website cookies are always stored locally in an encrypted format.

Considering our e-commerce website, various links like Men's Fashion, Women's Fashion, Kid's Fashion, Home Accessories, Electronic Appliances, Books, Movies & Music, etc. available on a web page should be clicked on and verified if the user navigates to the expected page.

Similarly, different functionalities like Login, Signup, Search Option, Filters, Sort Order, Add to Cart, etc. should be verified on different web pages like Login Page, Sign up Page, Product Details Page, Shopping Cart, Order Review, Payment, etc. The website should be checked for session/cookie management like session expiration and session storage etc.

## **Will the end-user find the website easy to browse?**

Usability testing has to be performed to measure the website's ease of use for an end-user in the context of accessibility, searchability, and usefulness, etc.



Below mentioned are few of the test scenarios that should be verified while performing usability testing for a website:

- Website content should be informative, structured and linked logically so that user can understand easily
- Web page controls should be easy for users to navigate
- The website should have Help & Instruction documents uploaded
- The website should have the Search feature for end-user convenience
- Access to/from the Main menu to all pages should be there
- Website content should be verified for any spelling mistakes
- The website should follow defined guidelines in the context of background color, pattern, style, fonts, image placements, frames, borders, etc.

The website should be accustomed to the translation feature considering the fact that it can be accessed by users from different nations with different languages, currencies, etc.

Few tools that can be used to perform usability testing are User Zoom and Reflector. An e-commerce website should be customer-friendly, easy-to-navigate and attention-grabbing. All web pages should be verified for accessibility, fonts, styling,

images, spelling mistakes and product relevant information. A website should be equipped with relevant help documents and customer support facilities. Considering the increase in touchscreen-based interfaces we need to validate the accessibility of both key inputs and touch screen inputs. Similarly, images and website content should be validated for usability on different screen sizes (mobiles, laptops, and tabs, etc.).



## Is the website accessible on different devices possessed by end-users?

Assuming that our website can be accessed by a range of users with a different set of devices, we need to ensure that the website runs well on all of them without any glitches.

To ensure the same, website compatibility checks should be done which comes with **Compatibility Testing**. During compatibility testing of a website, it is ensured that the website runs well on different browsers, Operating Systems & Devices like laptops, mobile phones, tablets, printers, etc.

**Browsers Compatibility (Cross Browser Testing):**

The website should work well with different browsers like Microsoft Internet Explorer, Microsoft Edge, Firefox, Google Chrome, Safari, and Opera. All active versions of these browsers should be verified with different browser features turned ON/OFF.

Also, while performing **cross-browser testing**, QA should also check for the optimal website performance across browsers.



**Operating System Compatibility (Cross Platform Testing):**

In order to identify potential user experience issues, a website should be tested on various platforms like Windows, Linux, Unix, MAC, Solaris, etc. so as to be sure of the OS compatibility.

**Device Compatibility (Cross-Device Testing):**

A website can be browsed through different devices like laptops, mobiles, tablets, etc. with different OS available like iOS, Android, Windows, etc. Hence, testing should be performed on the devices too covering the below scenarios.

- Website screen size should be adjustable as per the device
- A device should be screen rotation featured
- The website should not show up any loading issues on different devices with different network speeds
- Verify the website behavior when the device is in/out of network range
- Verify the website behavior on low CPU and Memory to support different form factors

For an e-commerce website, the compatibility check is one of the most important testing types. The customer base will be large and will access our website from different browsers, operating systems & devices.

Considering mobile platforms are becoming popular, we should ensure website load on small form factor under acceptable load time. It is also important to validate the use of different network speed to ensure it is usable for all customers.

**Is the website secured enough?**

Security testing is performed to uncover vulnerabilities in a system and ensure a website is secured.



**Below is the checklist that can be verified while performing security testing:**

- The website should be accessible to only authenticated users
- Website users should be able to perform only those tasks for which they are authorized
- The website should be verified for CAPTCHA fields for user identification
- Browser security settings should be verified while moving from secure to insecure pages
- Web Server protection should be there for inaccessible web directories or files
- Ensure restricted files should not download without appropriate access
- Sessions which got inactive should automatically get killed after a certain period of time
- All invalid and unauthorized attempts by end-users or intermittent system errors/failures should get logged for analysis purpose

Tools like Retina CS Community, Veracode, and SQL Map can be used to perform security testing of your website.

**As a part of security testing, an e-commerce website should be validated for**

1. Website Access Controls.
2. Any leakage of user's personal info.
3. Secured Payment Methods.

**Is the website performance up to the mark?**



To check the performance of a website, performance testing can be done. It will evaluate the behavior of an application under a variety of workload conditions which could be a realistic scenario. If the system goes live without conducting performance tests, it may end up with issues like a slow running system or poor usability which likely will affect the brand image as well as market sales.

A website can be tested against load & stress.

**Below given is the checklist for web performance testing:**

- Website behavior should be observed under normal and peak load conditions
- Website's performance should be examined by measuring response time, speed, scalability and resource utilization
- Proper RCA (root cause analysis) should be done with a solution if a system breaks down or gets unstable at any point in time
- Network latency issues should be identified if any

An e-commerce website should be tested thoroughly using a set of simulated users during normal as well as peak load conditions which can be during 'Sale Season'. During the sale, users accessing the website would multiply. Also, website behavior should be examined while multiple concurrent users accessing the same items or performing the same actions (like transactions or placing orders) on the website. There are various tools available in the market for performance testing. Few of them are LoadRunner, WinRunner, Silk Performer, JMeter, etc.

**Is the data entered on a website stored accurately and persist across sessions?**

The database is one of the critical components of a web application that holds the complete information entered through a website. Hence, to make sure that correct user data is getting saved in database tables without any manipulations and to maintain data integrity below verifications should be performed.



- Verify data consistency across user interface i.e. Website UI and Database
- Verify that DB tables are updating properly whenever insert/update/delete actions are performed by a website application
- Verify the response time of technical queries and fine-tune them if required
- Check for DB connectivity and access permissions

As a QA team member testing e-commerce website, you can perform below activities and validate the changes each time in the corresponding database tables. This will make sure that website UI and DB, both are consistent.

- a. Placing an Order for a product.
- b. Canceling Product.
- c. Opt to Exchange Product.
- d. Opt to Return Product.

**Is the website integrated well with other interfaces in the workflow?**

Interface level testing is performed to check on the smooth interaction of the website with different interfaces like Web Server & Database Server.

During interface testing, the tester needs to make sure if the application requests are being sent properly to the database and correct information is displayed to the client as output. A webserver should not throw any denial exceptions at any point in time and the database should always stay in sync with the application.

**Will the website perform as expected even after going live?**

Once a product moves into a production environment, the regular inspection should be done to keep a check on quality control.

**Below scenarios can be considered while verifying product in production:**

- Web application tests should be executed periodically and test logs should be saved as proof of Service Level Agreement (SLA) compliant
- Auto-scaling systems and load balancers should be checked if in place and functioning
- Keep a check the end-user experiences and try to uncover defects or malicious attacks which typically goes unnoticed during QA testing
- Monitor the product response time during peak loads
- Execute edge level test cases in real-time to identify network failures, connection failures or interruption by an unexpected call

## Reference

