**TESTING PLAN**

1. Functional Testing.

It is one of the most common testing techniques to check if the product works as the customer intended for and fulfils the requirements recorded in the developer’s documentation. It includes the following tasks.

#### **1.1. Testing UI Workflows.**

Every web application has several business workflows which a tester can know from the requirement specification document. However, in the Agile model, there occur product grooming meetings to discuss the functionality. So, the tester here can write test cases to cover different scenario and set success criteria.

#### **1.2. Testing Hyper Links.**

A web page may contain many types of links like the alternate, archives, external, help, icon, search, and tags. We need to ensure all of them are working fine or else to report any dead link.

#### **1.3. Input Field Validation.**

Input fields mostly appear on web forms to ask information from users. Some of them can be left blank and some can’t. A tester has to verify the right behaviour associated with them.

* If there are text fields on the page, then check if they have a default value or not. If it is a drop-down list, then ensure it is getting populated with options.
* There could even be AJAX fields to verify that change values at runtime.
* Also, a tester should not forget to check the error messages that appears on the screen.

#### **1.4. Cookie Testing.**

When a user accesses any website, the browser caches it session information inside a cookie. It is to save him from the hassle to log in every time he visits the site. A couple of things to verify here are.

* Make sure the cookie gets cleaned after the clearing the browser cache or after its expiry.
* A tester should also check by clearing the cookie and see if the website is asking for credentials or not.

#### **1.5. Validate HTML. CSS & XPath.**

To ensure a website will run smooth, it should have a clean HTML structure complemented with optimized CSS and unique XPath. If it lacks in any of that, would result in incorrect workflows and bad user experience. There are standard W3C practices for using HTML and CSS which the website should adhere firmly.

### 2. Usability Testing.

Usability testing confirms that the web application provides a pleasing user experience. Not only the testers but also the actual users or the customers of the product perform the usability testing. While the traditional testing carries out by a developer, designer or project manager, the usability testing avoids any bias by taking feedback from the end user. There are usually three types of usability testing to conduct usability analysis.

#### **2.1. Comparative Usability Testing.**

Intends to analyse the usability of one website with another. Such tests are usually run to compare a website against peer or competitor sites.

#### **2.2. Explorative Usability Testing.**

Here, the users test a range of different services where they verify possible end to end scenarios. It helps in highlighting any gaps and points out where to focus the design efforts.

#### **2.3. Usability Evaluation.**

It is suitable for testing a new or updated product at the pre or post-launch phase. This test makes the users aware of the new design to ensure it is easy to use and brings a positive user experience. Its aim is to identify any potential issues before the product launch.

### 3. Interface Testing.

Whether in waterfall or agile, interface testing is essential for ensuring a positive user experience. There are mainly three areas that a tester should target.

#### **3.1. Application.**

An application may give access either through the UI or via the SOAP/REST APIs. So, both the interfaces need a through validation. Testers should ensure that all the requests reach the database and the response renders correctly at the client-end.

#### **3.2. Web Server**

Web server is the backend processing all the client requests. The Proper checks should be conducted to ensure that it should not decline any request made from either via UI or the REST API.

#### **3.3. Database.**

First, the database should respond to all queries sent via UI or APIs. And any change in data should not violate the data integrity. Also, need to check if the data returned from the database is displaying correctly or not. Next, it should not permit any direct access instead should return a proper access denial message.

### 4. Compatibility Testing.

Compatibility testing confirms the website design is compatible across different browsers and also on a variety of devices. It includes following tests.

#### **4.1. Browser Compatibility Testing.**

This test makes sure that the web application can render successfully across different browsers like Chrome, Firefox, Safari, Internet Explorer. It helps in find out HTML, CSS, JavaScript, and AJAX related issues.

#### **4.2. Device Compatibility Testing.**

This test confirms that the web application is responsive and works on devices of different shapes and sizes. Please note that it’s not a native application test. Instead, it runs using the built-in device browser.

### 5.Performance Testing.

Performance testing intends to trespass the boundaries of standard testing and aims to verify the application response time and throughput under various load conditions. Usually, we can group these tests in four categories.

#### **5.1. Load Testing.**

It is the most common form of performance testing that observes the system under specific load conditions. Load testing helps in measuring the response time of business-critical transactions and the capacity of the underlying database and application server.

#### **5.2. Stress Testing.**

Stress testing aims to test the application beyond the expected maximum load. With it, we can determine the threshold limit of the system and also document the behaviour when it goes past the limit.

#### **5.3. Soak Testing.**

Another name that we call soak testing with is the endurance testing. It proposes putting the application continuously under the high-load conditions and suggests monitoring parameters like CPU and memory utilization.

#### **5.4. Spike Testing.**

Spike testing goes by its name and proposes to measure the application performance when there is a sudden increase in the volume of users accessing it. Its purpose is to check if the application would handle the sudden spike.

### 6. Security Testing.

Security testing is crucial for all type of web applications. It makes sure the application won’t allow unauthorized access to any of its resources that could break the site itself. And more importantly, it ensures that the sensitive information is safe and won’t cause any social or financial damage to its users. Here, we are sharing some of the best security testing techniques which a tester can apply to uncover the security issues.

#### **6.1. Privilege Escalation.**

It’s an act of exploiting a bug, design flaw or a misconfiguration in the web application to gain direct access to the reserved resources.

#### **6.2. SQL Injection.**

It is a technique which takes undue advantage of ambiguities present in the web application and bypasses ambient security checks. The input fields (text boxes) are the prime targets for implanting SQL injection. To prevent such attacks, the application should either not allow Special characters or use them properly.

#### **6.3. Unauthorized Data Access.**

It is a technique which proposes to gain unauthorized access to data within an application. Such data leaks take place on servers or launched via networks.

#### **6.4. URL Manipulation.**

This method modifies the website URL and steals important information. It happens when the application uses the HTTP GET method to exchange data between the client and the server. The information is passed in parameters in the query string. The tester can change a parameter value in the query string to check if the server accepts it.

#### **6.5. Denial of Service.**

It is a method to prepare conditions that make a machine or network resource unavailable to its legitimate users.

#### **6.6. Identity Spoofing.**

Identity spoofing is a method where the tester uses the credentials of a user or device to launch attacks against network hosts, steal data or bypass access controls.

#### **6.7. Cross-Site Scripting (XSS)**

Cross-site scripting is a computer security flaw found in web applications. It enables attackers to inject client-side script into Web pages viewed by other users.

#### **6.8. Vulnerability Scanning.**

Testers can use an automated computer program to identify security loopholes of the web applications and determine where it can be exploited and/or threatened.