TESTRIG

QA's guide to dealing with the latest trends in web development 2022

Progressive Web Application

A Progressive Web Application has modern web capabilities to deliver an app-like user experience. They are fast, reliable, and engaging. Progressive Web Apps are built using common web technologies including HTML, CSS, and JavaScript.

By using these technologies, they can deliver a native app-like experience on the web. In addition, they can work offline and be installed on devices like phones and tablets.

As a result, Progressive Web Apps provide a unique opportunity to combine the best of the web and the best of apps.

How to Test Progressive Web Application

Progressive web applications load like regular web pages but they can offer user functionality such as working offline, push notifications, Responsiveness, Secure connections, slow network and device hardware access traditionally available only to native applications.

However, it is important to test PWA on real devices to verify functionality across devices.

Voice Search Optimization

Voice search optimization is a process of optimizing a website for voice search queries. Voice search is a speech recognition technology that allows users to search the internet by speaking instead of typing.
It is fast, convenient and accurate. It is becoming increasingly popular as more devices are equipped with this technology.

How to Test Voice based application?

While testing a voice-based application, it is important to consider both the accuracy of the speech recognition and the quality of the audio output.

To test the accuracy of speech recognition, you will need to create a list of commands that the application should be able to understand. After that, it is important to test each command by speaking it aloud into the microphone.

Few tests are necessary for your voice-based application release. These include unit, usability, and performance testing.

API First Development

API First Development is a methodology for designing and building APIs. The principle behind API First Development is that the API should be designed before the implementation. This allows for a well-designed API that meets the needs of the consumers.

It also results in less re-work, since changes in the API can be made without affecting the implementation.
In addition, API First Development makes it easier to document the API, and test and deploy it.

As a result, it is becoming increasingly popular with companies which are looking to provide APIs to their customers.

How to Test API?

One of the most important aspects of API testing is making sure that the API is stable and reliable. To do this, we need to test for both functional and non-functional requirements. Functional requirements include things like correct input/output, error handling, and so on. Non-functional requirements include things like performance, scalability, and security.

No-code/low-code Development

No-code and low-code platforms are a new type of development platform that allows businesses to create software applications without the need for traditional coding. These platforms are user-friendly and provide a Visual Development Environment (VDE) that makes it easy to develop applications without writing code.

No-code and low-code applications offer significant benefits over traditional coding methods, including reduced development time and cost, increased efficiency, and improved collaboration.

How to Test No-code/low-code application?

When it comes to testing no-code or low-code applications, there are certain best practices that should be followed in order to ensure the reliability and quality of the final product.

- ***Set a clear objective***
- ***Know your audience***
- ***Be aware of limitations.***

Server-less Application

A server-less application is a type of application that does not require a server to run. Instead, the application is run on a platform-as-a-service (PaaS) or infrastructure-as-a-service (IaaS) provider. Server-less applications are typically event-driven and can scale automatically. This allows for a more scalable and cost-effective solution than traditional server-based applications.

How to Test Server-less Application?

There are various ways to test your serverless applications. One way is by using AWS LambdaTest, which allows you to run code in a simulated environment of an actual Amazon Web Services (AWS) instance called "Lambdas."

Another method for testing serverless applications is to use a local development server. This can be helpful for testing your code's functionality without having to deploy it to a live environment. it is also important to test your application in a real-world environment. This can be done by deploying your application to a staging environment and running tests against it.

Cloud-Native Application

Cloud-native applications are built using a microservices architecture, which means that they are composed of small, independent services that communicate with each other using APIs. This type of architecture enables developers to rapidly release new features and updates without adversely affecting the overall performance of the app. In addition, cloud-native apps are designed to be scalable and resilient, allowing them to seamlessly handle increased traffic or other unexpected challenges.

How to Test Cloud-Native Application?

There are four common ways to test a cloud-native application, but we'll focus on unit, component, integration, and end-to-end testing.

<TESTRIG />

Contact us

To Align your development with the best QA practices

info@testrigtechnologies.com