EVERYTHING
YOU NEED
TO KNOW TO MAKE
AN EDUCATED
DECISION ABOUT
YOUR TECHNOLOGY
STACK

IDEAS2IT

# angular vs react.js vs vue.js

+ Our evaluation of UI Frameworks, Tools & Technologies

2019

# What's inside?

# Preface

The only constant in the web frontend development landscape is that it is in constant flux year after year. So, it becomes paramount to reevaluate the tools, frameworks, libraries and practices after every few quarters. As of this writing most of the web frontend development happens in either Angular, ReactJS or VueJS. Compared to our previous evaluation in 2018 which only had Angular and ReactJS as major players, now, we also have VueJS with significant traction.

As always a direct comparison between Angular, ReactJS and VueJS alone will not be sufficient. So, it will not be an individual comparison but a comparison of their respective ecosystems on the whole.

# Angular, React.JS & Vue.JS - Comparison

In this section, we compare all three frameworks using a plethora of parameters to highlight how they fare against each other.

| | Angular | ReactJS | VueJS |
|---|---|---|---|
| Type | JavaScript Framework | JavaScript Library | JavaScript Library |
| Used for | Web development and Hybrid mobile app development (Ionic) | Web development and Native mobile app development (React Native) | Web development and Hybrid mobile app development (Onsen UI) |
| Maintained by | Google & Community | Facebook & Community | Community |
| Coded in | TypeScript | Javascript (Also supports Typescript) | Javascript (Also supports Typescript) |
| Ease of Learning | Steep learning curve since it is an end to end framework. | Easier | Easiest among the three |
| Popularity | Ranked 4th | React is ranked 2nd on the list of hottest Javascript projects on GitHub | Ranked 1st, Vue.js was the trendiest project in 2017. |
| Frontend Performance | Fast (Slowest of the lot) | Faster | Faster |
| GitHub Contributors /Stars | 821 / 44408 | 1268 / 120028 | 251 / 125271 |
| Browser rendering | Shadow Dom | Virtual DOM | Virtual DOM |
| Bootstrapping | Angular-cli | CRA (Create React App) | Vue-cli |
| Debugging | Easy to debug HTML and JS | Easy to debug the JS but difficult to debug the HTML | Easy to debug HTML and JS |
| Error handling | Compile-time | Runtime | Runtime |
| Key Features | - HTML-based template<br>- RxJS<br>- Shadow DOM<br>- Service Worker Support<br>- Ivy: New Rendering Engine<br>- Bazel Compiler | - One-way data binding with props<br>- Stateful components<br>- Virtual DOM<br>- JSX (JavaScript XML)<br>- Architecture beyond HTML | - HTML-based template<br>- Reactivity<br>- Reusable Components<br>- Transitions<br>- Routing<br>- Integrations |

# Angular, React.JS & Vue.JS - Pros & Cons

Learn about the benefits of each framework. This is based on the latest release for each framework.

The latest version of Angular is 7. It comes with a new, significantly smaller rendering engine.

## Benefits of Angular

- Opinionated, full-fledged E2E framework. Easy to maintain the code.
- Two-way data binding avoids developer intervention.
- Out of the box server side rendering (SSR) with cache support reducing CPU load.
- The framework is designed to be highly testable.
- Use of directives, components and elements helps in cleanly organizing the code.
- Angular CLI makes it easy to follow the best practices as a project grows.

## Drawbacks of Angular

- It has slow initial loading times and still struggles with complex frequently changing pages. However, SSR alleviates the problem though.
- The opinionated code organization actually makes it difficult to handle the scopes of complex entities.
- The community support is slowly waning.


ReactJS has not matured any further after the previous evaluation.

## Benefits of ReactJS

- Easier to learn.
- Unopinionated and highly flexible. It even supports Typescript.
- Virtual DOM implementation makes it one of the fastest client side renderer available.
- Unidirectional data flow prevents parent from being affected by children.
- Great community provides it with a lively ecosystem of off-the-shelf components.
- It is relatively easier to migrate between versions.
- Create-react-app makes it simple to create a new application from scratch.

## Drawbacks of ReactJS

- Official documentation still remains poor.
- Since there are lots of off-the-shelf components to choose from it becomes the responsibility of the developer to evaluate them.
- Since it is a view only framework, it requires considerable effort to integrate with other layers of MVC.


VueJS is gaining lots of traction in recent months. It is usually lauded for its performance and simple lightweight APIs.

## Benefits of Vue JS

- Super small library. It is barely 8KB after Gzipping.
- Developers can separate the template-to-virtual-DOM compiler and even the run time.
- Outperforms bulky frameworks like Angular and Ember.js.
- Ease of understanding and development.
- Easier to debug, saving lot of time for developers.
- Useful for building entire applications or even replacing existing applications partially.
- Like ReactJS, its Virtual DOM implementation makes it one of the fastest client side library.

## Drawbacks of Vue JS

- Development is done by a closed community which is predominantly chinese causing language barriers.
- Being unopinionated and flexible means developer has to bear the responsibility for cleaner, maintainable code.

# Angular, React.JS & Vue.JS - Conclusion

Learn which framework wins the race and why.

# Conclusion

What we just saw is the current state of the above libraries or frameworks. Last year ReactJS was the chosen choice for frontend web development. Like we noted back then it is a temporary choice. Though ReactJS is still competitive our choice would be VueJS.

Yes, in last one year there were changes in Angular and ReactJS but they were just evolutionary and not revolutionary enough to overcome the traction which VueJS was able to garner in the same period. VueJS coupled with its performant, low-barrier-to-entry API and a rallying community is clearly the ecosystem of choice. Needless to say - for the time being.

# Our evaluation of UI Frameworks, Tools & Technologies

The following is a compendium of the current UI Frameworks, Technologies, Tools and Techniques that were most interesting to us. This choice is based on feedback from our engineers based on their daily work and experiences.

Introducing Ideas2IT's UI Technoverse →

# Ideas2IT's UI Technoverse at a glance

Open interactive version

### Frameworks

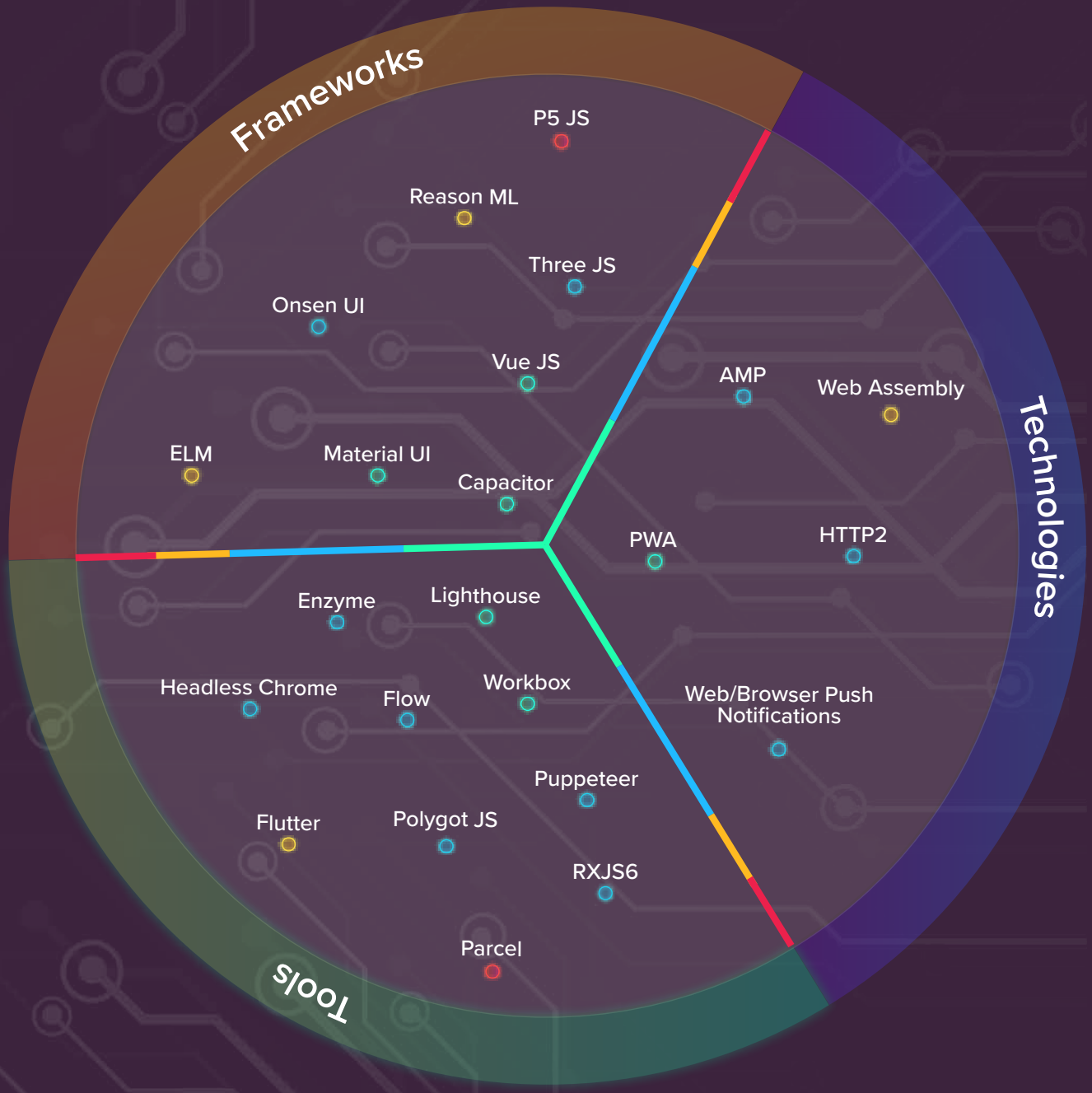
Opinionated and un-opinionated frameworks or even languages which drive the way in which an application could be developed.

### Technologies

Opinionated and un-opinionated frameworks or even languages which drive the way in which an application could be developed.

### Tools

Opinionated and un-opinionated frameworks or even languages which drive the way in which an application could be developed.




### Ideal

Why aren't you using this already?

### Useful

Use if applicable to your project

### Not ready

Promising but not there yet.

### Avoid

The heading says it all! Steer clear.

# Frameworks

Opinionated and un-opinionated frameworks or even languages which drive the way in which an application could be developed.

## 1.Capacitor IDEAL

Capacitor is a cross platform application development toolkit. It can be used to generate hybrid applications for iOS and Android; and desktop applications for Windows, MacOS and Linux using Electron. It can be safely considered as a shared superset of Cordova and Electron. Like Cordova and Electron, it provides a plugin API which in turn provides access to the native OS functionalities like camera, GPS, Push notifications, etc. As of version 4 of Ionic platform, all hybrid applications internally use Capacitor. It is highly recommended to evaluate and use Capacitor in any new hybrid development.

## 2. Material UI IDEAL

Capacitor is a cross platform application development toolkit. It can be used to generate hybrid applications for iOS and Android; and desktop applications for Windows, MacOS and Linux using Electron. It can be safely considered as a shared superset of Cordova and Electron. Like Cordova and Electron, it provides a plugin API which in turn provides access to the native OS functionalities like camera, GPS, Push notifications, etc. As of version 4 of Ionic platform, all hybrid applications internally use Capacitor. It is highly recommended to evaluate and use Capacitor in any new hybrid development.

## 3. Vue JS IDEAL

Vue JS is one of the lightweight frameworks for building progressive web applications. In web applications, smaller the payload size, better the responsiveness. Vue is sized around 20KB, which is much smaller than its counterparts like React and Angular. Though it started out slow, it has gathered enough traction to be widely supported by the community. Also, it has very good documentation. It is a good time to make use of Vue JS.

## 4. Onsen UI USEFUL

OnsenUI is a hybrid mobile app or mobile web app development framework. The USP of OnsenUI is that it allows development using different frameworks/libraries such as jQuery, Angular JS, Angular, React JS, Vue JS. This makes developing an OnsenUI app very simple because we would be leveraging our existing knowledge. Like other similar frameworks, it also provides mobile-optimized UI components which mimic the native look and feel. OnsenUI is supplemented by the commercial tool Monaca which provides all bells and whistles to build, test and deploy to end users.

## 5. Three JS USEFUL

Three JS is a lightweight open source javascript library for creating 3D web graphics and animations within the web browser. It primarily works on canvas elements, SVG elements, and uses WebGL for rendering. It supports many special effects and filters including particles, lensflare, sprites, real-time reflection and refraction, and even area-based lighting. Three JS is compatible with almost all modern browsers. Most of its

features can be leveraged using less amount of code and it also has good documentation with sufficient examples. Three JS releases a new version about once a month, and the API can change at any time. This implies a lot of outdated solutions and references. However, it is a good choice for the feature set it provides.

## 6. Elm NOT READY

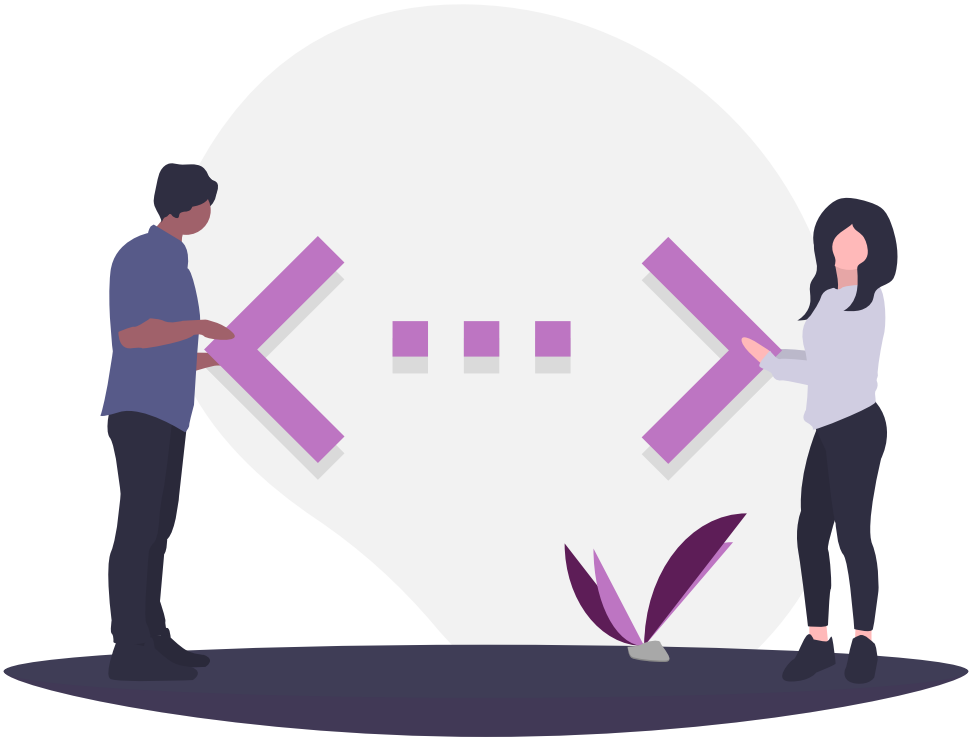
Elm is a functional programming language that promises no run-time exceptions through it's intelligent type inference and its functional programming paradigm. To draw parallels, Redux has been majorly inspired by Elm's model view architecture. It is lightweight, robust and faster with lesser asset size compared to other available frameworks. It has its own package manager which has ample amount of support. In Elm, every data is immutable and it relies on pure functions to handle this immutability. However, Elm is still in its earlier stages. It has a growing community which acknowledges the vast family of problems it can solve. Elm seems to have a good potential in future and it can be used as a gateway to master functional programming.

## 7. Reason ML NOT READY

Reason, developed by Facebook, is syntactic sugar over OCaml. Bucklescript serves as the wrapper. The USP of Reason is that Bucklescript can compile to both Javascript and assembly. It brings in all the benefits of OCaml like strong typing, immutable data and functional orientation. In the long term, Facebook wants to make use of Reason for cross-platform native development. However, in the interim, it can only be used for the Javascript generation. In theory, Reason shows promise but it is still a work in progress.

## 8. P5 JS AVOID

P5 JS a JavaScript library which has a full set of drawing functionalities. However, it is not just limited to drawing in a HTML5 canvas. A huge list of addons can be used to make the visuals interact with HTML5 objects, including text, input, video, webcam, and sound. Though P5 JS can be used for quick prototyping, it is not production ready when it comes to performance.



# Technologies

Innovations which re-imagine and rewrite the way we develop applications to achieve age-old development goals such as faster development time, better performance, ease of maintenance, etc.

## 1.PWA IDEAL

A Progressive Web App or PWA is a web app that uses modern web capabilities to deliver an app-like experience to users on the device. They are installable and live on the user's home screen, without the need for an app store. They offer an immersive, full-screen experience with help from a web app manifest file and can even re-engage users with web push notifications. When launched from the user's home screen, service workers enable a Progressive Web App to load instantly, regardless of the network state. A service worker, written in JavaScript, is like a client-side proxy and puts you in control of the cache and how to respond to resource requests. The programmatic API allows developers to decide how to handle caching and provides a much more flexible experience than other options. It definitely improves web usage in devices and the user experience. It is recommended to use PWA if an application can leverage its benefits.

## 2. AMP USEFUL

Accelerated Mobile Pages (AMP) is an initiative which envisions high-performance mobile pages. Creating an AMP page involves following a set of guidelines, including the AMP library and using some AMP-specific HTML tags. This ensures that best practices for resource

caching and delivery are followed and enforced. Also, Google provides the AMP Cache which acts as a Content Delivery Network (CDN) for AMP compatible web pages. Usually, AMP pages are well suited for static content pages like blogs. It would be highly recommended to use AMP for applicable pages and gain massive performance (and SEO) benefits.

## 3. HTTP2 USEFUL

HTTP has been the text based protocol powering the internet revolution. Internet over the decades has evolved massively. HTTP1.1, which we use by default, has started to show its age and limits. HTTP2 is a major overhaul of HTTP1.1. For starters, it is a binary protocol which brings in benefits of a TCP socket like having more than one data stream sharing a single connection, which alone single handedly makes the protocol much more efficient than the previous version. It also natively supports streaming. Most modern browsers and web servers support HTTP2. We can take up a strangler approach in moving to HTTP2, one feature at a time.

## 4. Web/Browser Push Notifications USEFUL

Web/Browser push notifications is an upcoming web feature which brings the Android and iOS push notifications to the browsers. At the moment, the browser support is limited to Chrome, Safari and Firefox. Also, the underlying implementations differ widely between the platforms and browsers, which translates to browser specific development. However, when properly used, it provides very good end user experience. So, it is recommended to use Web Push Notifications to augment the user experience wherever applicable and feasible.

## 5. Web Assembly NOT READY

Web Assembly (WASM) is a new standard for binaries which can be executed as-is in a browser. It is a precompiled binary format which brings in benefits like smaller size and 'native code'-like performance. It will execute in a sandboxed virtual machine, honouring all permissions and security model of JS execution. Also, it can be accessed from JS code. Currently, there is tooling support to compile C/C++/Rust code to WASM format. At the moment, there is not much browser support and therefore WASM is not ready for mainstream use but it can power future web applications with AR and ML use cases.

# Tools

Aids which facilitate development by providing solutions for specific use cases such as Hot Module Reloading, evaluating quality of web page, facilitating, testing, etc.

## 1. Lighthouse IDEAL

Lighthouse is a tool to evaluate the quality of a web page. Once a page is evaluated, it provides a report highlighting the various facets of the webpage like security, functionality, performance, accessibility, offline support, etc. It was originally developed to evaluate Progressive Web Apps but it can be used in all the cases. It can be executed from Chrome as an extension, from command-line and from node applications programmatically. Also, the reports can be directly saved as gists. Considering the value Lighthouse provides, it is an indispensable tool for web developers.

## 2. Workbox IDEAL

Workbox, as the name suggests is a toolbox, from Google, which makes PWA development easier. It provides a set of utility functions which could be used in a Service Worker. It makes actions like caching, prefetching, background syncing, etc, simpler. It also has a CLI which can be used to generate the service worker itself. It can be augmented with new features using plugins. Workbox is absolutely necessary if service workers are used in an application.

## 3. Enzyme USEFUL

Enzyme is an open source JavaScript testing framework developed and maintained by Airbnb. It makes testing React applications easy with a minimal setup cost. It also makes it easier to assert, manipulate, and traverse any React Components' output. It works with Browserify, Webpack, JSDOM, Jest, Karma, Mocha, etc. It keeps the code base clean with reduced boilerplate code. We would encourage all React app developers to use this library.

## 4. Flow USEFUL

Javascript is a weakly typed dynamic language. Though it makes it simple to code, it introduces complex, esoteric bugs during execution. Adding strong typing to Javascript has been attempted by many. The most famous one is Typescript from Microsoft. Flow is one such static type checker from Facebook. Similar to Typescript, using Flow means writing non-standard Javascript and using a transpiling step to generate valid javascript. However, Flow also supports type annotations being added as part of comments, which makes it an ideal choice to augment existing code bases. Also, the ideology behind Flow makes it easier to integrate with React projects. Since type checking the existing code is made easy with Flow, it is an imperative and soft option for Javascript.

## 5. Headless Chrome USEFUL

End to end automation testing is usually done in a headless mode. When trying to test Webkit/Blink based browsers like Chrome/Safari/Opera, usually some headless browsers which leverage Webkit/Blink like Phantom JS, Slimer JS, Trifle JS, etc. are used. Starting

from Chrome v59, a new headless mode is supported natively. Using Headless Chrome intuitively increases the reliability of the automation suites since it is the actual browser and not some emulation of it. It is recommended to use Headless Chrome to replace the likes of PhantomJS.

## 6. Polygot JS USEFUL

Polyglot JS is an internationalization (i18n) library developed by AirBnb. It has zero dependency and it can work in both browser and node JS. It does not provide translation but helps in managing translated literals across multiple locales. As expected in an i18n library, it provides interpolation of string literals. The specific strength of Polyglot is that it makes it easier to handle pluralization, which is one of the most common problems in i18n. It is recommended to use Polyglot JS in any application which needs multi lingual support.

## 7. Puppeteer USEFUL

Puppeteer is a chromium-only equivalent to Selenium Webdriver. It uses the Chromium debug protocol to invoke and control Chromium/Chrome instances. By default, it executes in a headless mode but it can be disabled during development/debugging of Puppet scripts. Due to its tight integration with Chromium, it can provide granular control when compared to alternatives like Selenium Webdriver. Also, the barrier to entry is low since the Puppeteer scripts are similar to Selenium scripts. However, Puppeteer is not limited to automation testing alone. It can also be used for Web Crawling.

## 8. RXJS 6 USEFUL

RxJS has been a library of choice to make use of Observables. Observables make it easy to handle multiple asynchronous events in an application. Until Observables becomes a feature of Javascript, libraries like RxJS is required. RxJS is being used internally in Angular from version 2.0. RxJS 6 comes with a newer and simpler API and also the ability to include only the part of the library which will be used. The main use cases of RxJS are streaming existing data, generating data, interoperation between existing APIs, and combining and selecting from existing streams. The disadvantage with RxJS is that it requires a paradigm shift in the way applications are designed. However, it is a good investment to make use of RxJS since the methodologies introduced are generic.

## 9. Flutter NOT READY

Flutter is a native mobile application development toolkit by Google, which is used to develop applications for Android and iOS. It is Google's answer to React Native. Flutter applications are written in the Dart language. It comes with a huge set of UI widgets which can be used to create fluid and intuitive UI. Naturally, based on the platform, the app can use different styles for the UI (like Material Design, Cupertino [iOS specific style]). Flutter provides the usual benefits to developers like common code base, Hot Module Reloading, etc. At the moment, Flutter is still in preview, but it looks promising. The official release followed by community traction could make it a go-to development platform for Mobile Development.

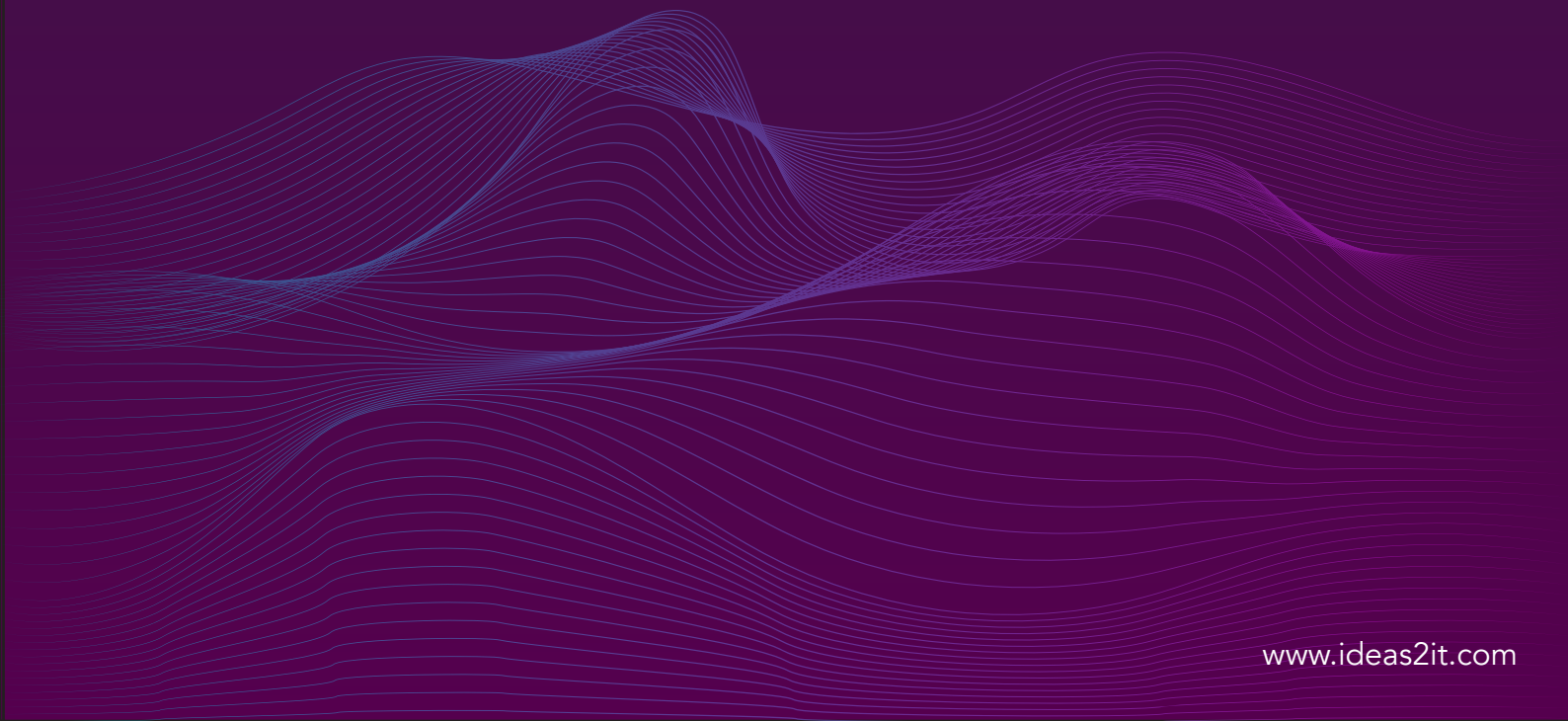
## 10. Parcel AVOID

Parcel is a web application bundler with the USP of zero configuration and fast multi-core compilation. It provides the features of most modern bundlers like Hot Module Replacement, integrations with transpilers, development server, etc. It is still being developed to become a scalable, plugin-supporting system and to allow optional configurations. Though Parcel can be used for quickly bootstrapping an application and zero configuration looks good on the paper for most applications, it is not practical enough. Moreover, Webpack 4 also promises all the USPs of Parcel. So, it is recommended to stick with existing bundlers for now.

Ideas2IT is a close-knit team of 250+ experts offering Web & mobile development for startups and businesses.

www.ideas2it.com
sales@ideas2it.com

IDEAS2IT



www.ideas2it.com