

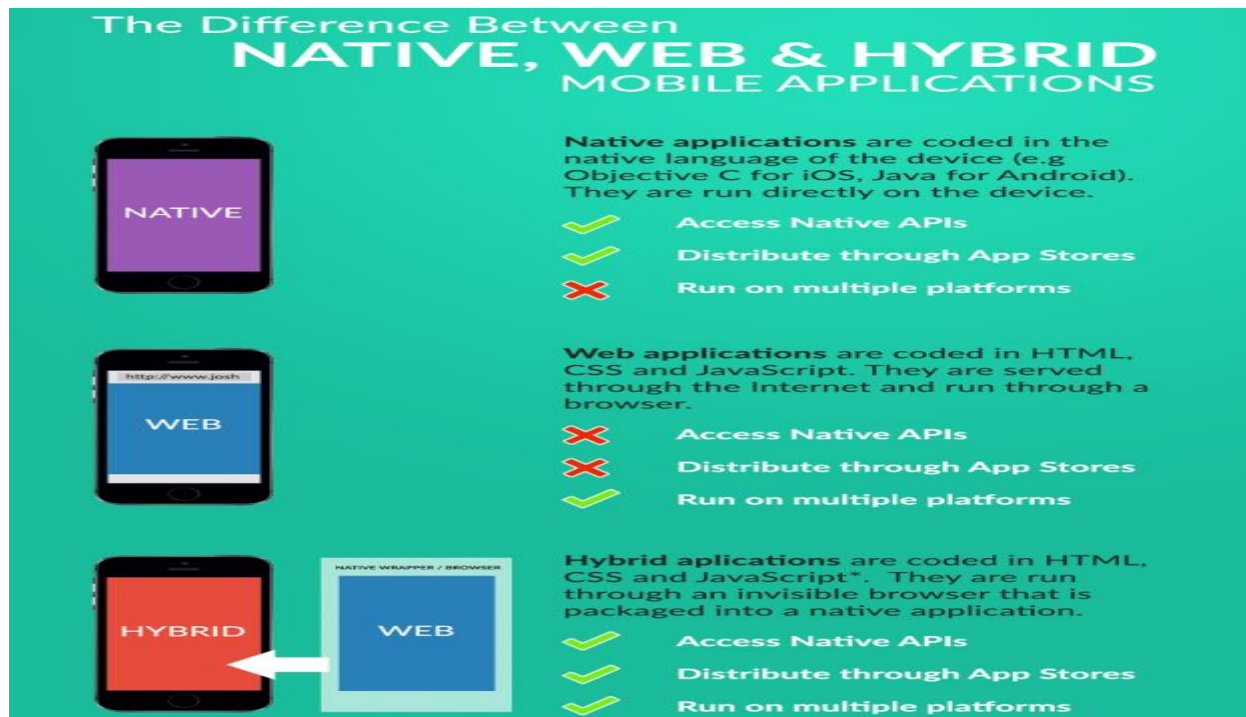
Date: 25-04-20
and 26-04-20

Facilitator: Mr. Manikandan, software Engineer, Rapid Acceleration Partners and Mr. Joel Kingsley, Associate Consultant, Zilker Technologies

Topic : Creating Web and Mobile Apps using Node.js and Firebase as backend

There are 6.8 billion people on the planet, 5.1 billion of whom own a cell phone. And today, an ever-growing percentage of these devices are smartphones. As a mobile device user, few things are as frustrating and difficult to fat-finger-navigate as a poorly designed mobile web app, or even a native app. If you're reading this you would probably already have a decent idea of what an HTML5 mobile application is, but you might not know how to actually get one on the Apple App Store or Google Play.

1. Understand the Difference Between Native, Hybrid and Web Applications



Creating mobile applications between options like:

- Native iOS/Android
- Ionic
- React Native
- Xamarin
- Flutter
- NativeScript
- Node.js

Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient.

Tools
Storage
Interface
Technology



VS Code



Firebase



2. Design Your Application

- Your app is more like a web page than a mobile application
- You don't handle online/offline states
- Your app is slow or unresponsive
- Your application doesn't make use of native features (i.e. there isn't really a reason for it to be distributed as a native app instead of a standard website)

3. Build Your Application Using a Framework and a UI Kit/Design System

```
<ion-list>
  <ion-item>
    Apple
  </ion-item>
  <ion-item>
    Banana
  </ion-item>
  <ion-item>
    Coconuts
  </ion-item>
  <ion-item>
    Durian
  </ion-item>
</ion-list>
```

4. Test Your Application Through the Browser and Package Your Application

The basic role that Cordova/Capacitor performs is this:

- Cordova/Capacitor creates a native application that contains a web view
- All of the resources for your application are stored inside of this native application

- Cordova/Capacitor loads your web-based application into the web view (the web view cannot be seen by the user)
- The web view displays your application to the user.

Advantages of Web Apps:

- Flexible Access – Employees can work from anywhere with internet access.
- Client Secure Login – Impress clients with a modern web portal and improve customer service with automated processes.
- Easy Setup – It takes a couple of minutes to setup a new user; provide a URL, username and password and they're away.
- Always Up To Date – As everyone is accessing the same version of the web app via a URL, they will always be accessing the most up-to-date version of the software.
- Storage Increase – With the availability of the cloud, storage space is virtually infinite.

Disadvantages Of Web Apps:

- Internet reliance – Whilst 4G & Wi-Fi internet access is available in many locations, if you happen to lose connection you will not be able to access your web app.
- Security – Whilst many business people may believe that data is less secure in a cloud environment. There are ways in which you can reduce risk of a data breach, such as SSL enforcement for a secure HTTPS access to your app.
- Reduced Speed – It's likely that a web app will operate at a slightly slower speed than one hosted on a server locally.
- Browser Support – Unfortunately, we don't all use the same browser. This means during development you'll need to ensure your app is supported across a variety of browsers.