

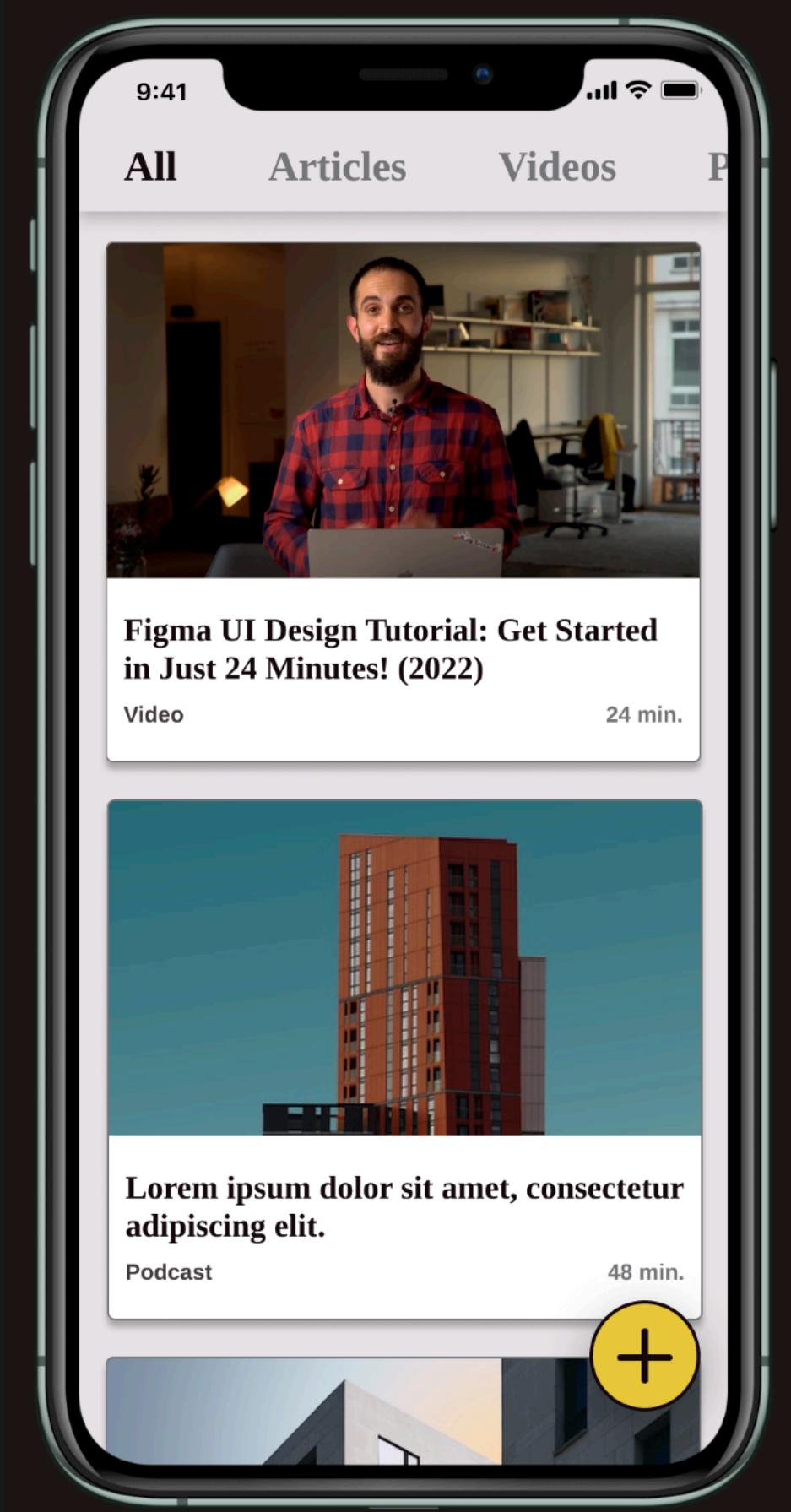
# STORAGE & NATIVE APIS



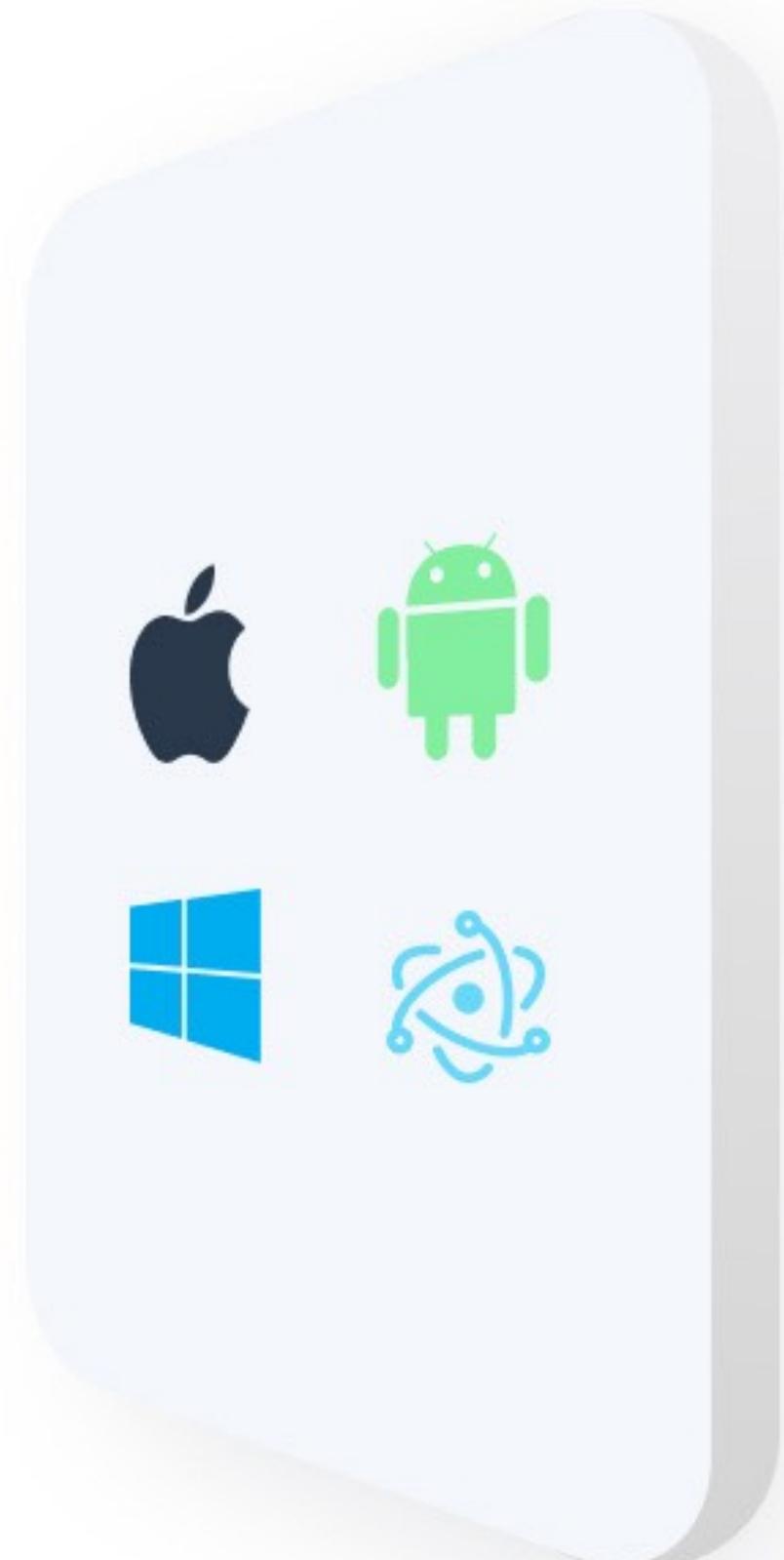
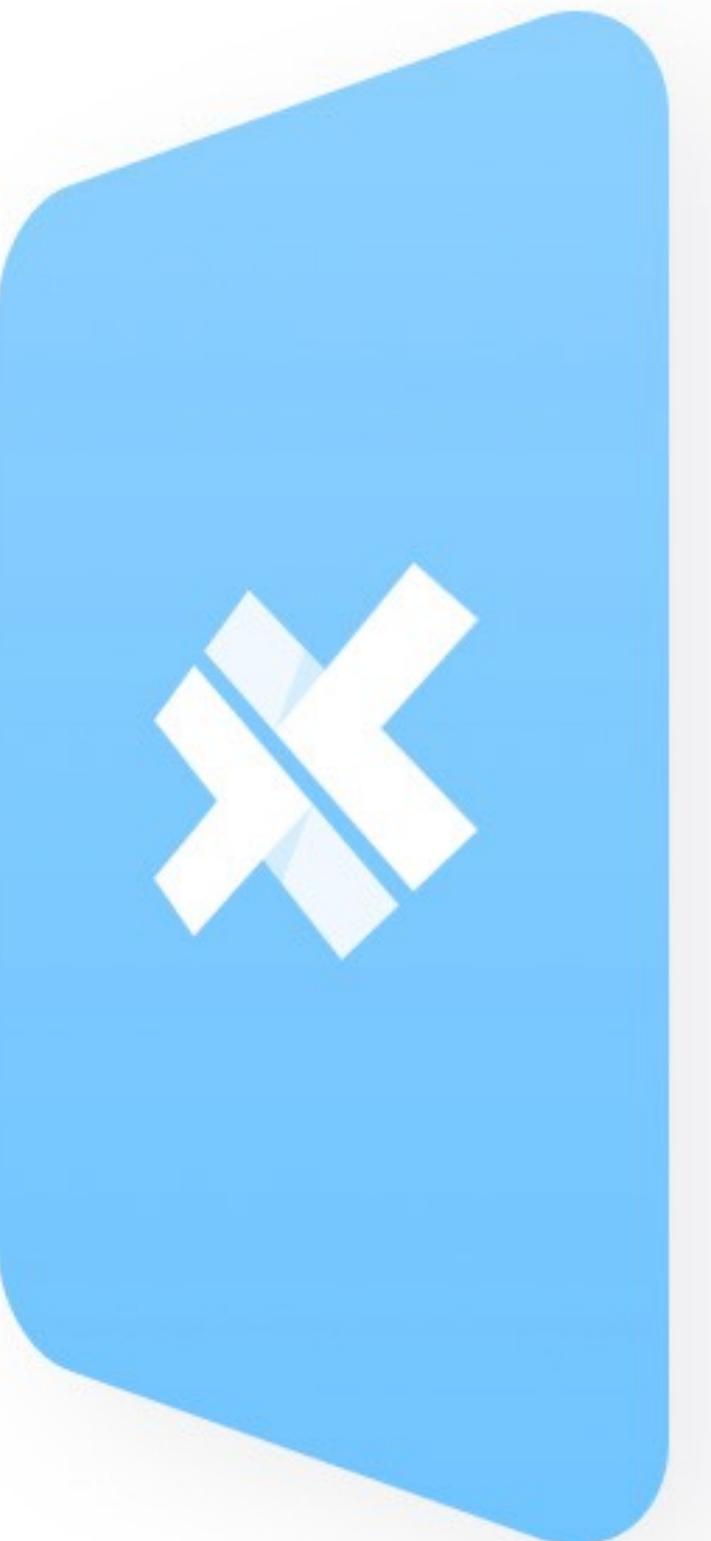
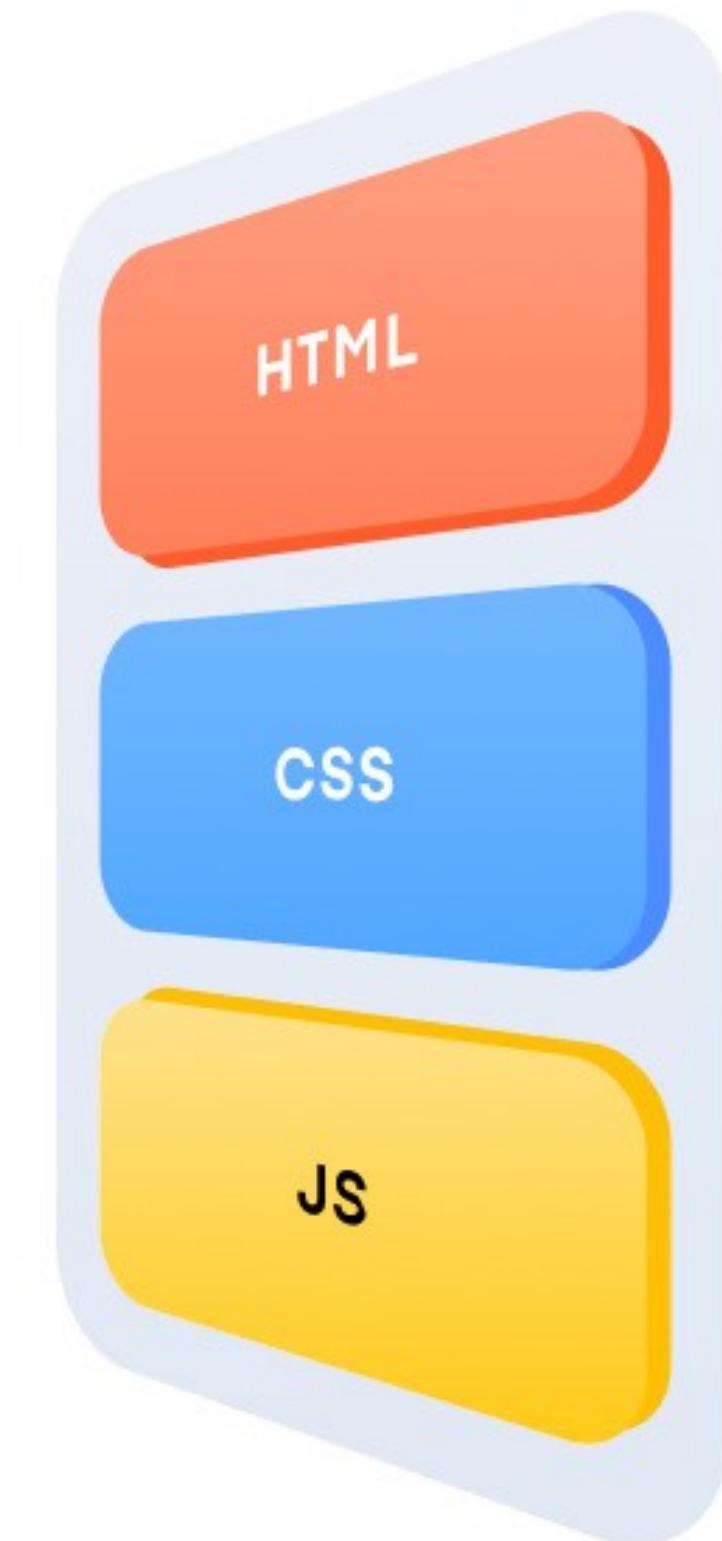
# Agenda

- Ionic Later List App w/ Firebase REST
  - Recap on Ionic React & UI Component
  - Native APIs & Native Device Features
  - Storage & Firebase
- Internship Info w/ Line Skjødt

# Ionic Later List App



<https://race.notion.site/Ionic-Later-List-App-866a28f6bf284b109183863d41bdd0ee>



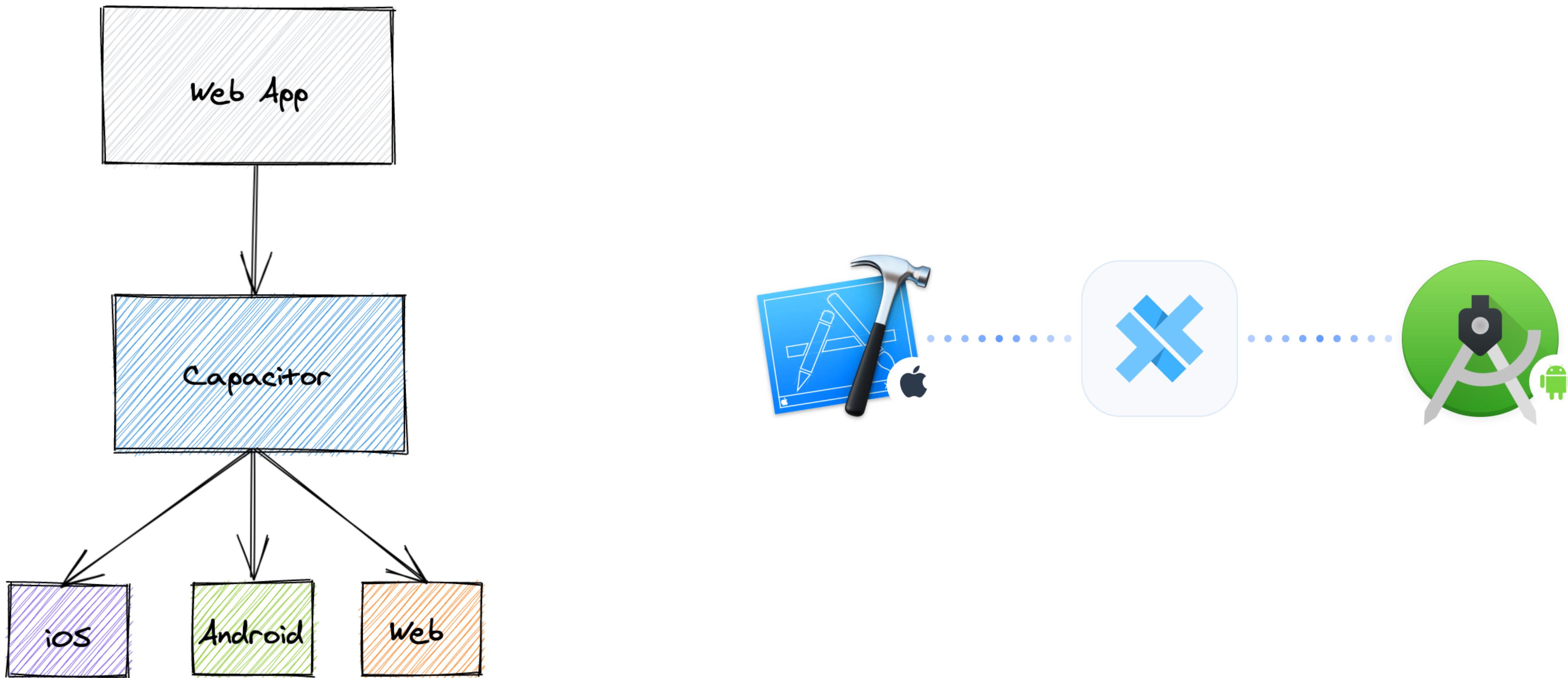
YOUR APP (ANGULAR, REACT, VUE...)

UI CONTROLS (IONIC)

NATIVE ACCESS (CAPACITOR)

DISTRIBUTION PLATFORMS

# App Dev w/ Capacitor





App Dev can be tough 🙃

You develop 90% in the  
browser

... and then you test on devices (physical devices!)

# Deployment



1. \$ ionic build
2. \$ ionic cap add [platform]
3. \$ ionic cap copy or \$ ionic cap sync (plugins)
4. \$ ionic cap open [platform]
5. \$ ionic cap run [platform]
6. \$ ionic cap run [platform] -l --external (Livereload)

Deploying to iOS and Android

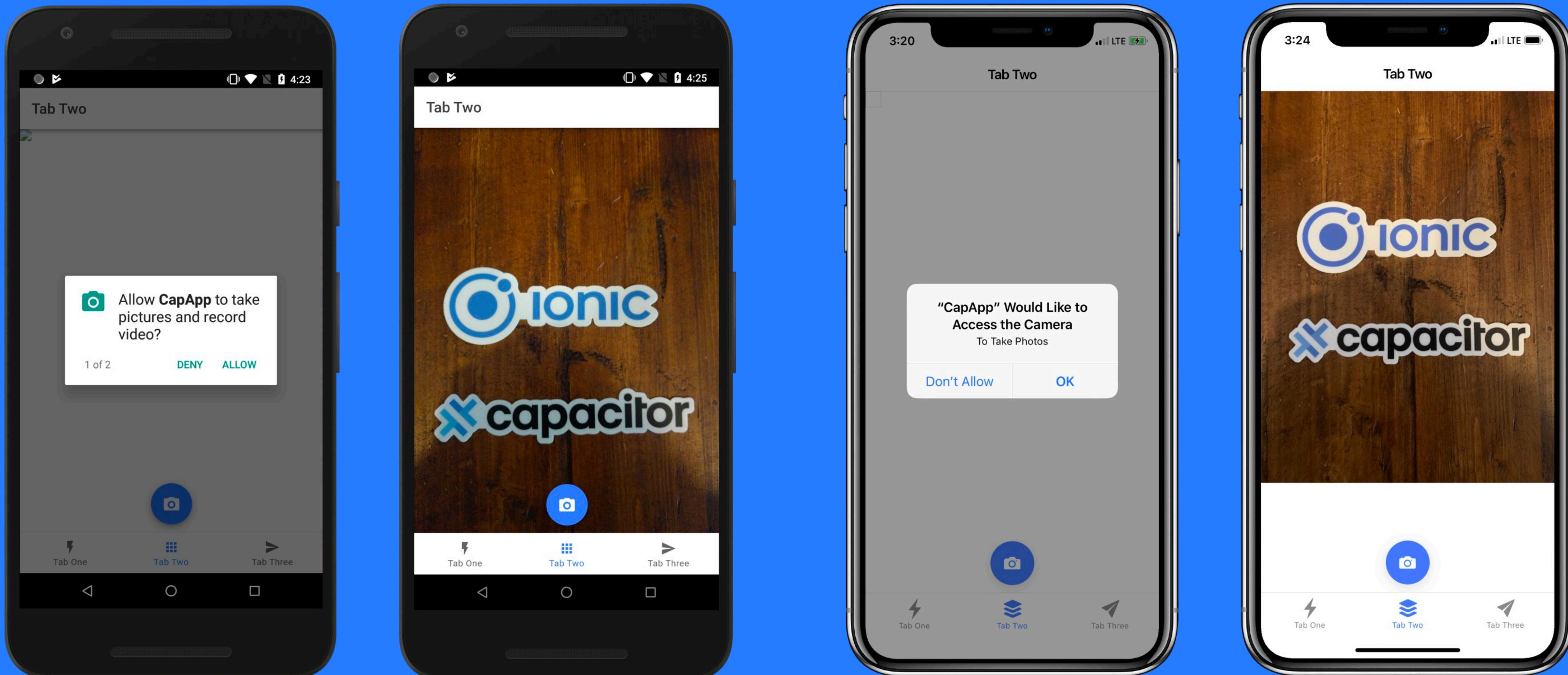
# Android & Livereload



1. Update Java Version
2. Android Studio -> SDK Manager -> Android 9.0(Pie)
3. Turn on developer mode on your phone, Allow USB Debugging and connect your device via USB.
4. Through Android Studio, choose your physical device and run project.
5. Run `$ ionic capacitor run android -l --external` and choose your device.

# User Permissions

<https://ionicframework.com/docs/react/your-first-app/deploying-mobile>



# Configuring Android

<https://capacitorjs.com/docs/android/configuration>

The screenshot shows a web browser window with the title 'Configuring Android - Capacitor'. The URL in the address bar is <https://capacitorjs.com/docs/android/configuration>. The page content is titled 'Setting Permissions'. It explains that permissions are defined in `AndroidManifest.xml` inside the `<manifest>` tag. An example code snippet shows how to add Network permissions:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.getcapacitor.myapp">
  <activity>
    <!-- other stuff -->
  </activity>

  <!-- More stuff -->

  <!-- Your permissions -->

  <!-- Network API -->
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
</manifest>
```

Below the code, a note states: 'Generally, the plugin you choose to use will ask you to set a permission. Add it in this file.'

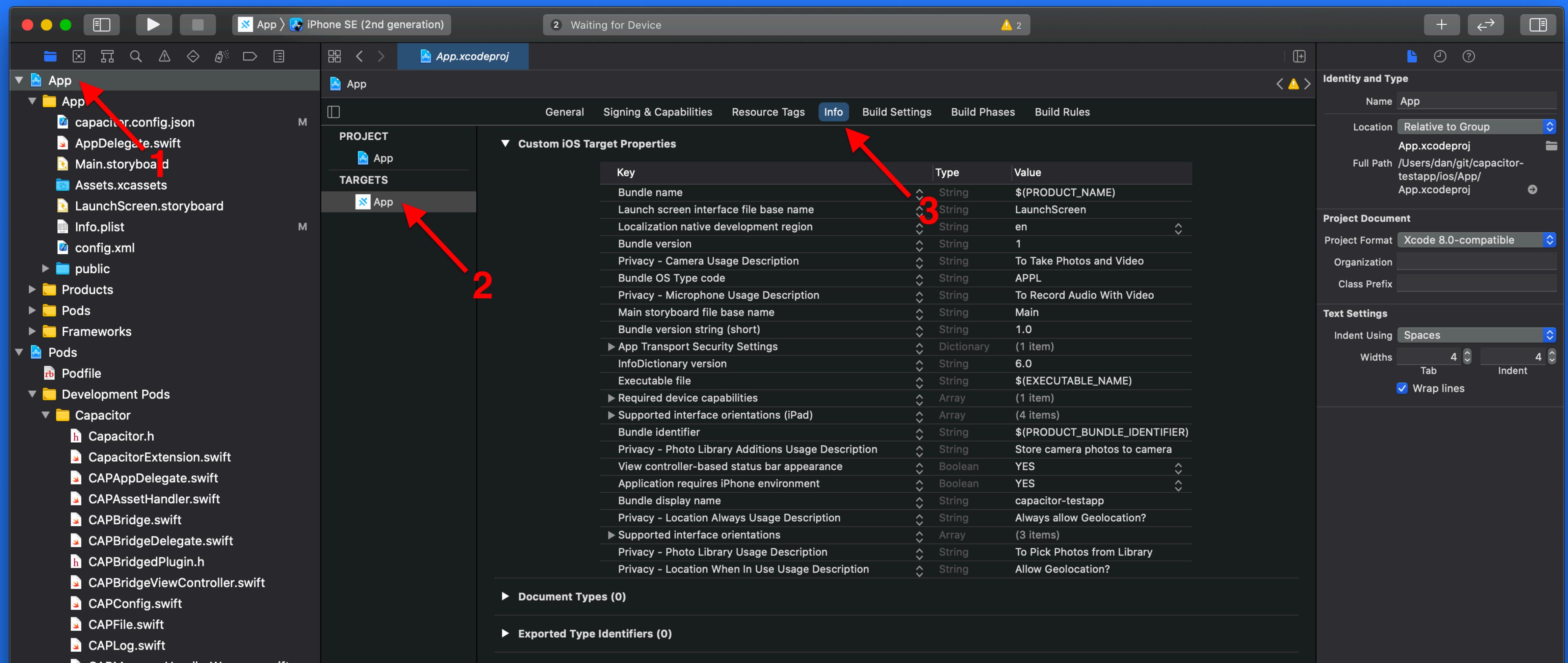
The left sidebar contains navigation links for 'Environment Setup', 'Installation', 'Using with Ionic Framework', 'Basics' (with sub-links for Development Workflow, Using Plugins, Native Project Configuration, and Utilities), 'Upgrade Guides', 'Cordova/PhoneGap', 'Concepts', and 'iOS' (with sub-links for Getting Started, Configuration, Custom Native Code, Deploying to App Store, Custom ViewController, and Troubleshooting).

The top right features a search bar, navigation icons, and links for 'Docs', 'Plugins', 'CLI', 'Community', 'Blog', 'Enterprise', and social media sharing.

The right side includes a 'CONTENTS' sidebar with links to 'Configuring AndroidManifest.xml', 'Changing the Package ID', 'Changing the App Name', 'Deeplinks (aka Android App Links)', 'URL Schemes', and 'Setting Permissions'. There is also a 'Submit an edit' button and an 'appflow' logo with the tagline 'Continuous app delivery made easy. Build, publish, and update from the cloud.'

# Configuring iOS

<https://capacitorjs.com/docs/ios/configuration>



# Common Commands

<https://race.notion.site/Ionic-CLI-Common-Commands-7715d0d0c3754bbc97bd3f0a47ddb975>

# Dev & deployment tips

<https://ionicframework.com/docs/developing/tips>

<https://ionicframework.com/docs/cli/livereload#tips>

<https://ionicframework.com/docs/react/your-first-app/deploying-mobile>

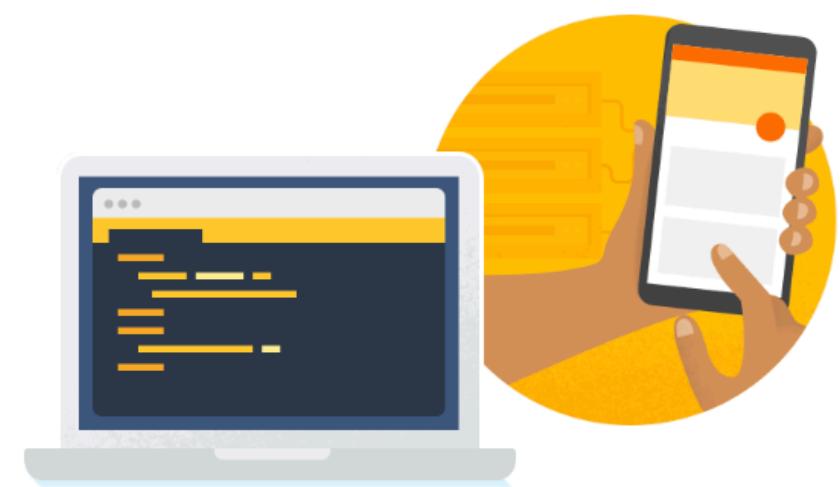


# Introducing Firebase



# What is Firebase?

Platform & Backend-as-a-Service  
for Web & App Development



## Build better apps



### Cloud Firestore

Store and sync app data at global scale



### ML Kit BETA

Machine learning for mobile developers



### Cloud Functions

Run mobile backend code without managing servers



### Authentication

Authenticate users simply and securely



### Hosting

Deliver web app assets with speed and security



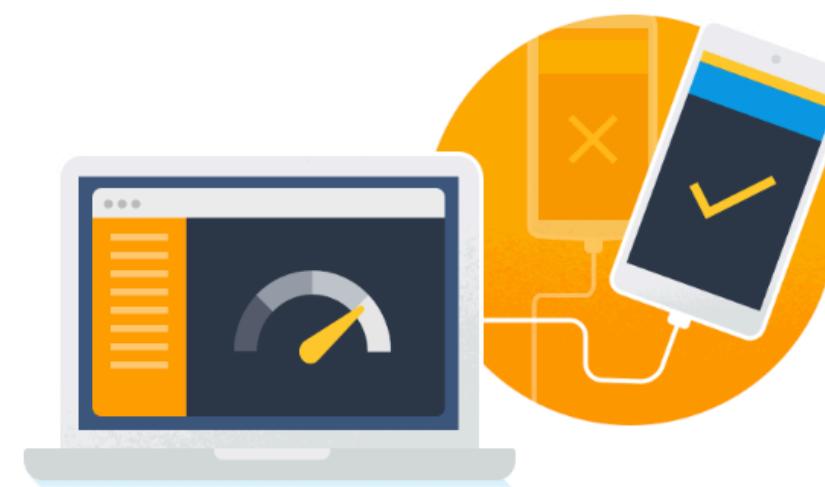
### Cloud Storage

Store and serve files at Google scale



### Realtime Database

Store and sync app data in milliseconds



## Improve app quality



### Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



### Performance Monitoring

Gain insight into your app's performance



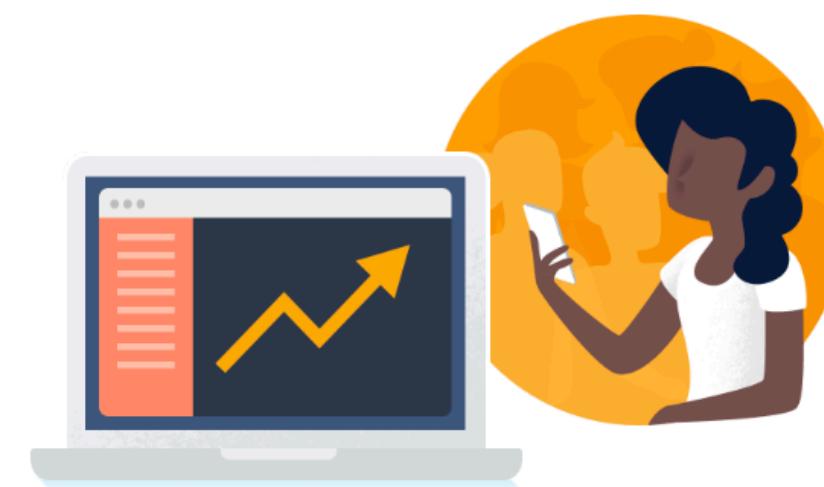
### Test Lab

Test your app on devices hosted by Google



### App Distribution BETA

Distribute pre-release versions of your app to your trusted testers



## Grow your business



### In-App Messaging BETA

Engage active app users with contextual messages



### Google Analytics

Get free and unlimited app analytics



### Predictions

Smart user segmentation based on predicted behavior



### A/B Testing BETA

Optimize your app experience through experimentation



### Cloud Messaging

Send targeted messages and notifications



### Remote Config

Modify your app without deploying a new version



### Dynamic Links

Drive growth by using deep links with attribution



# Realtime Database & REST API

Store and sync data in real time

<https://firebase.google.com/products/realtime-database>

# CRUD, CRUD, CRUD

... data, file storage, push notifications,  
authentication, hosting, etc.

```
import { Storage } from '@capacitor/storage';

const setName = async () => {
  await Storage.set({
    key: 'name',
    value: 'Max',
  });
};

const checkName = async () => {
  const { value } = await Storage.get({ key: 'name' });

  alert(`Hello ${value}!`);
};

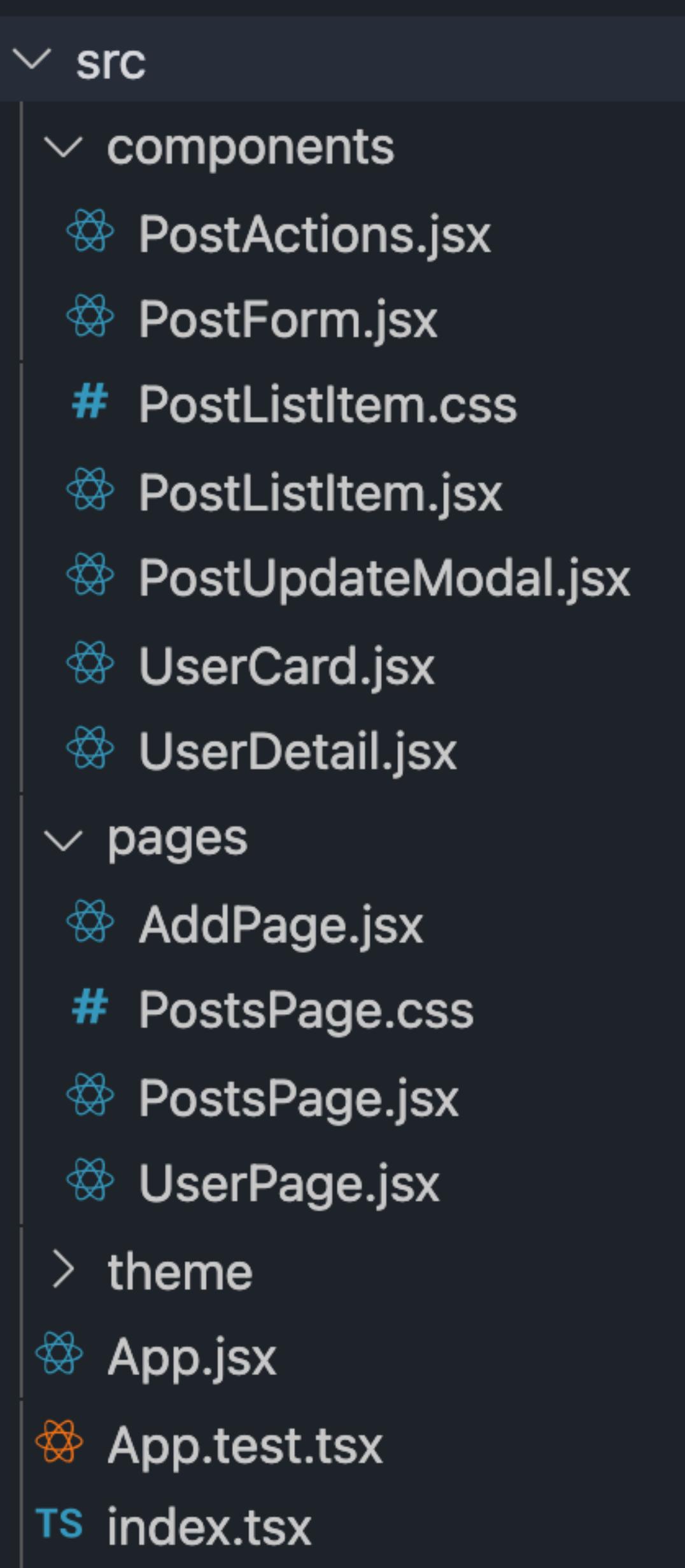
const removeName = async () => {
  await Storage.remove({ key: 'name' });
};
```

## Ionic Post App w/ @capacitor/storage

1. Explore and install the @capacitor/storage plugin.
2. Reimplement the services with @capacitor/storage. Use the docs to explore how to get and set data.

# Thinking in React

In React, UI is a **function of props & states**. The UI is built with (function) **Components**.



# Capacitor

A cross-platform native runtime for web apps.



Capacitor is a cross-platform native runtime that makes it easy to build modern web apps that run natively on iOS, Android, and the Web.

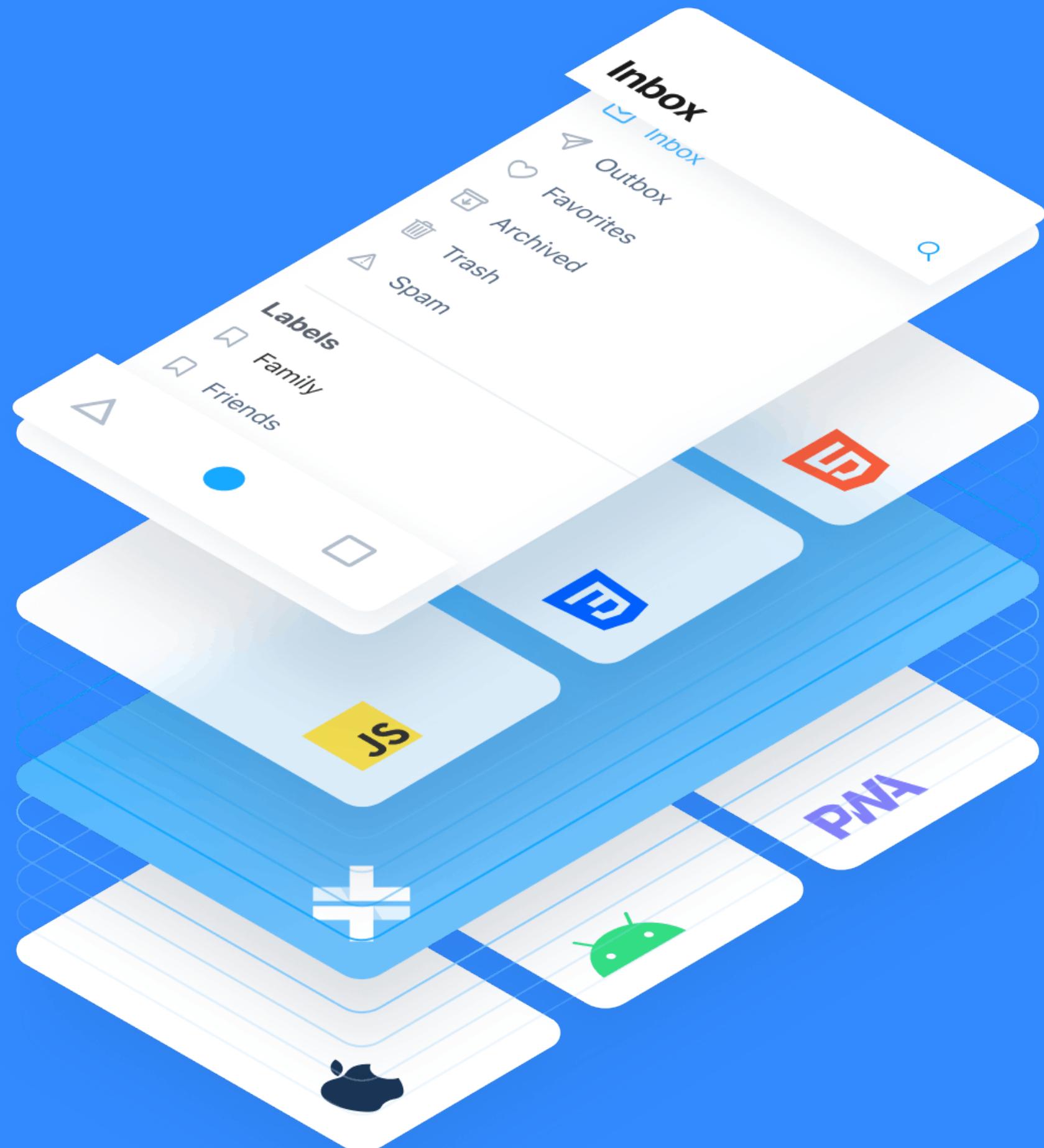
Representing the next evolution of Hybrid apps, Capacitor creates Web Native apps, providing a modern native container approach for teams who want to build web-first without sacrificing full access to native SDKs when they need it.

<https://capacitorjs.com/>

# Capacitor

A cross-platform native runtime for web apps.

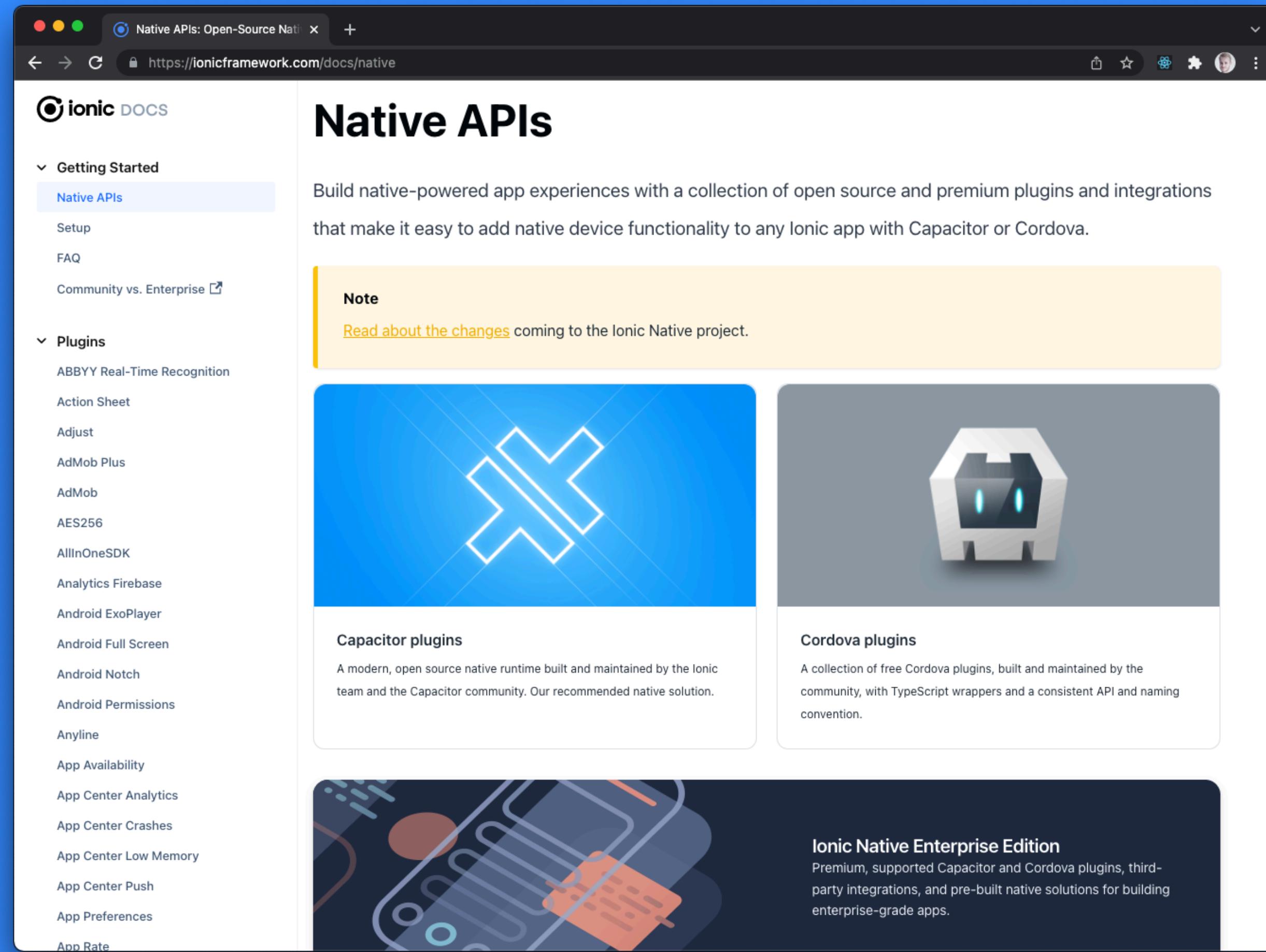
The Web View and the native app communicate through the use of Capacitor or Cordova plugins. Plugins provide native APIs such as camera, geolocation, and filesystem access to your web app.



<https://capacitorjs.com/>

# Capacitor Plugins

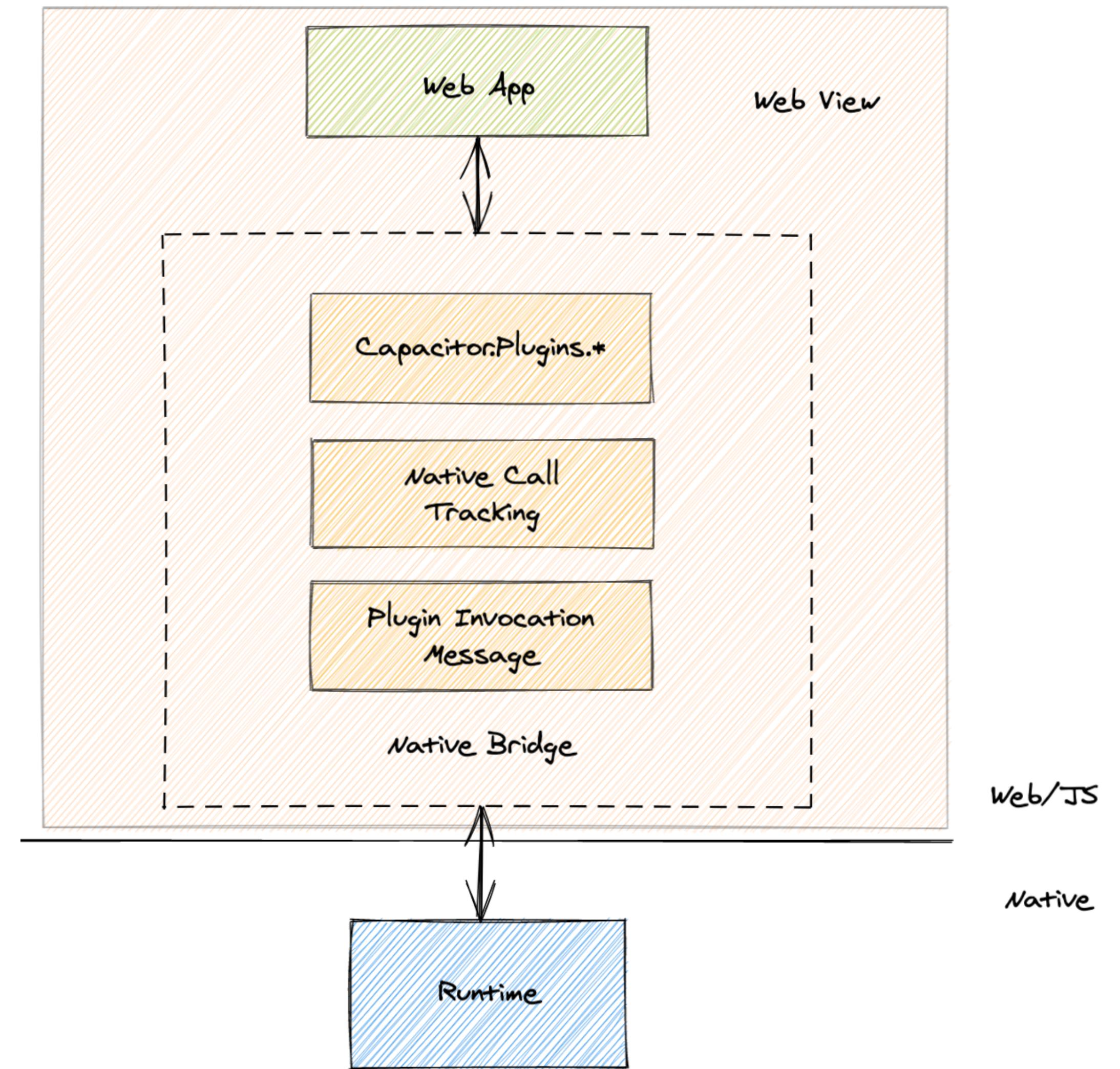
Enables directly access to native devices features with Native APIs



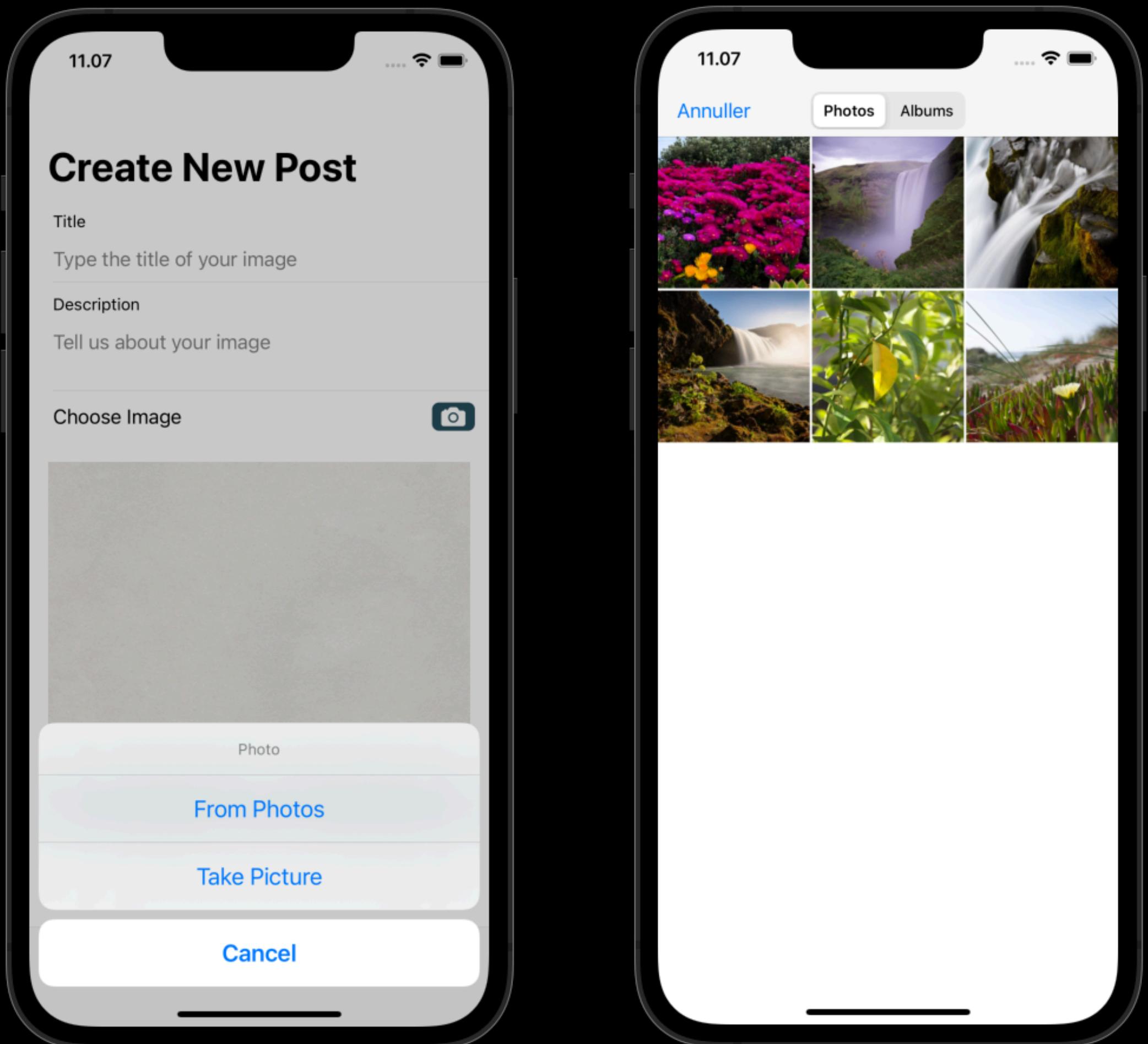
<https://ionicframework.com/docs/native>

# Native APIs

... provides access to native features such as camera, geolocation, and filesystem in your Web Native App.



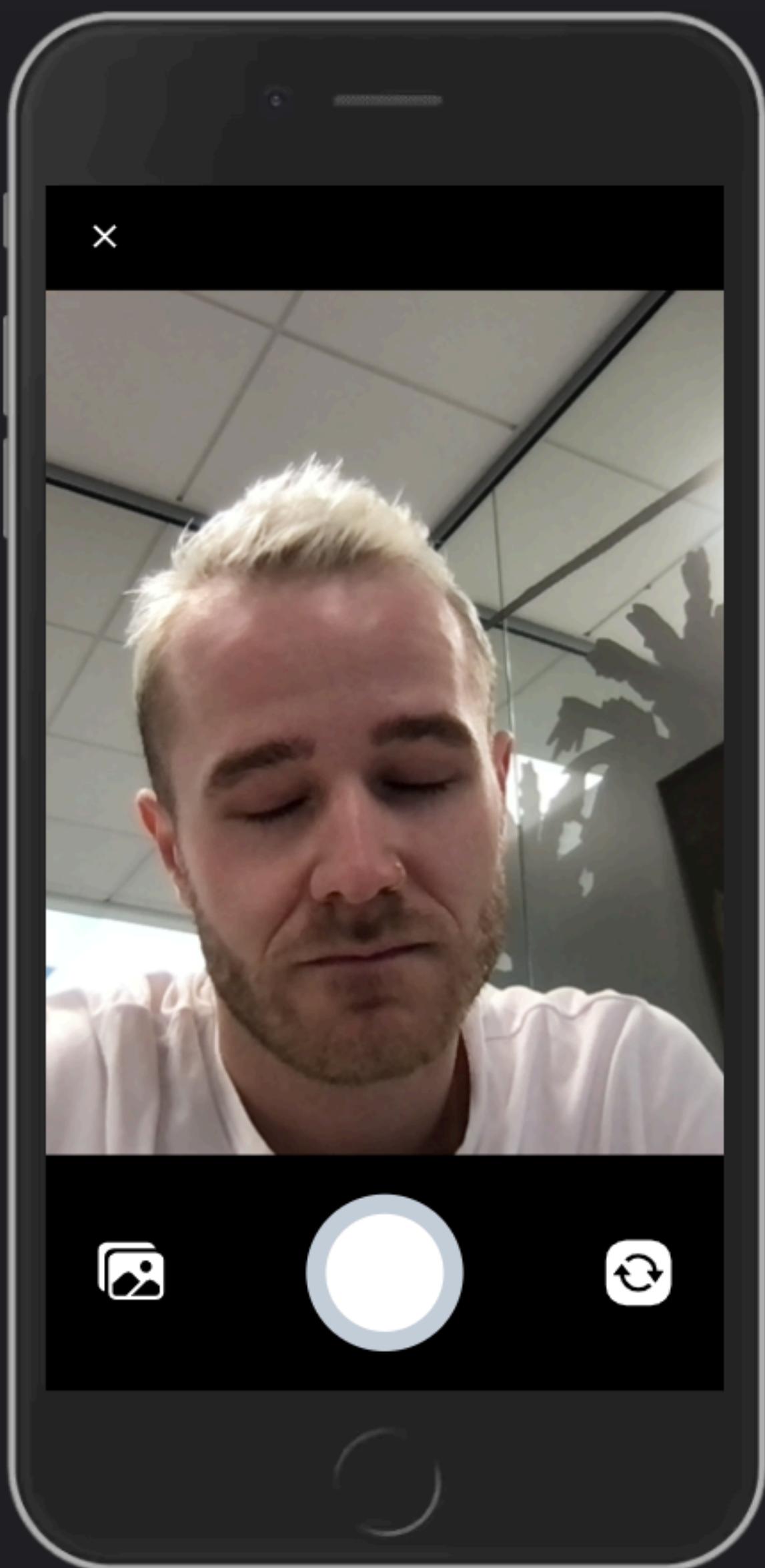
# @capacitor/camera



```
async function takePicture() { ←  
  const imageOptions = {  
    quality: 80,  
    width: 500,  
    allowEditing: true,  
    resultType: CameraResultType.DataUrl  
  };  
  const image = await Camera.getPhoto(imageOptions);  
  const imageUrl = image.dataUrl;  
  setImage(imageUrl);  
  
  return (  
    <form onSubmit={submitEvent}>  
      <IonItem>  
        <IonLabel position="stacked">Title</IonLabel>  
        <IonInput value={title} placeholder="Type the title" />  
      </IonItem>  
      <IonItem>  
        <IonLabel position="stacked">Description</IonLabel>  
        <IonTextarea value={body} placeholder="Tell us about your image" />  
      </IonItem>  
      <IonItem onClick={takePicture} lines="none">  
        <IonLabel>Choose Image</IonLabel>
```

# @ionic/pwa-elements

Web-based UI for PWAs



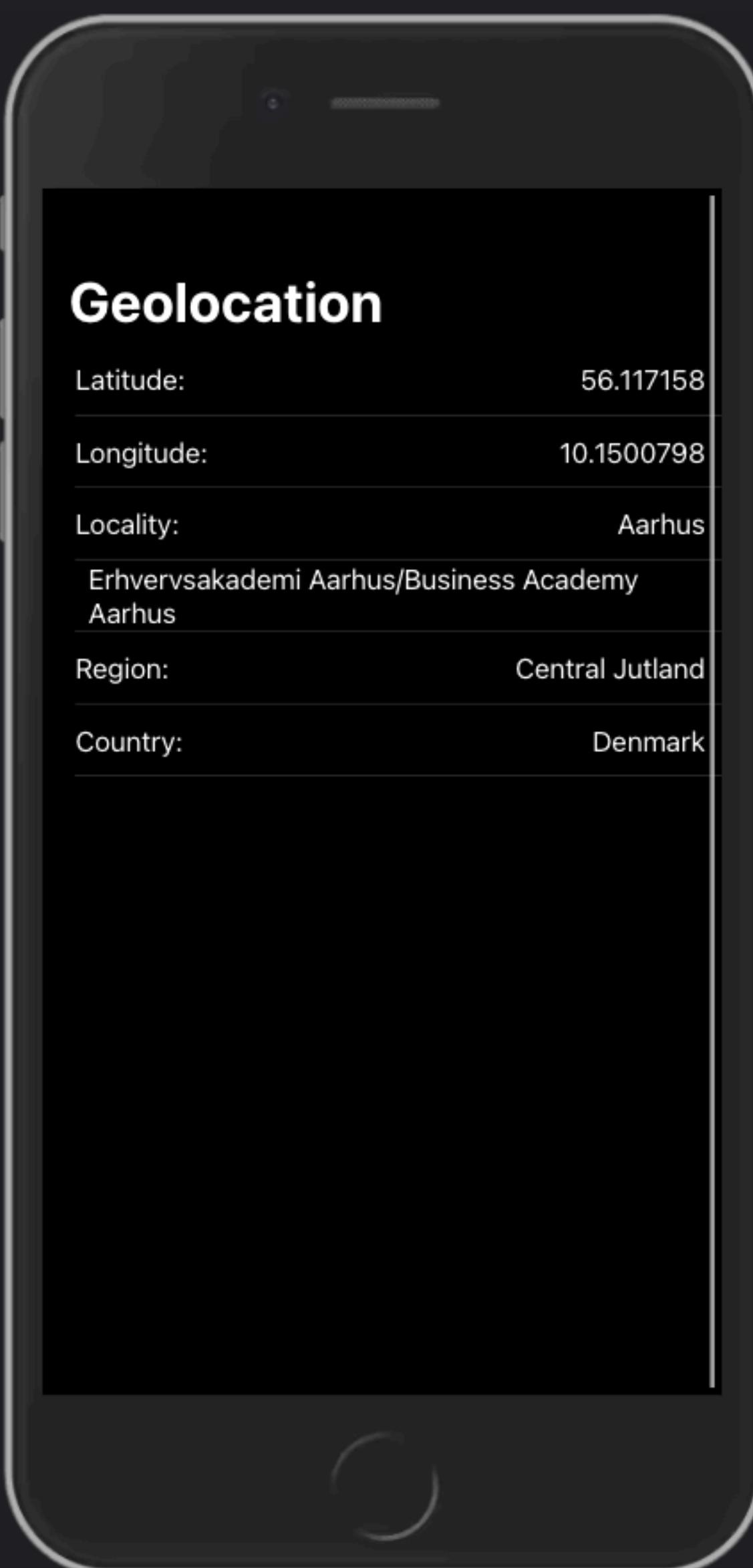
```
npm install @ionic/pwa-elements
```

```
import { defineCustomElements } from '@ionic/pwa-elements/loader';

ReactDOM.render(<App />, document.getElementById('root'));

// Call the element loader after the app has been rendered the first time
defineCustomElements(window);
```

# @capacitor/geolocation



## Track the current position of the device

```
import { IonContent, IonHeader, IonPage, IonTitle, IonToolbar, IonItem, IonLabel } from "@ionic/react"
import { Geolocation } from "@capacitor/geolocation";
import { useEffect, useState } from "react";

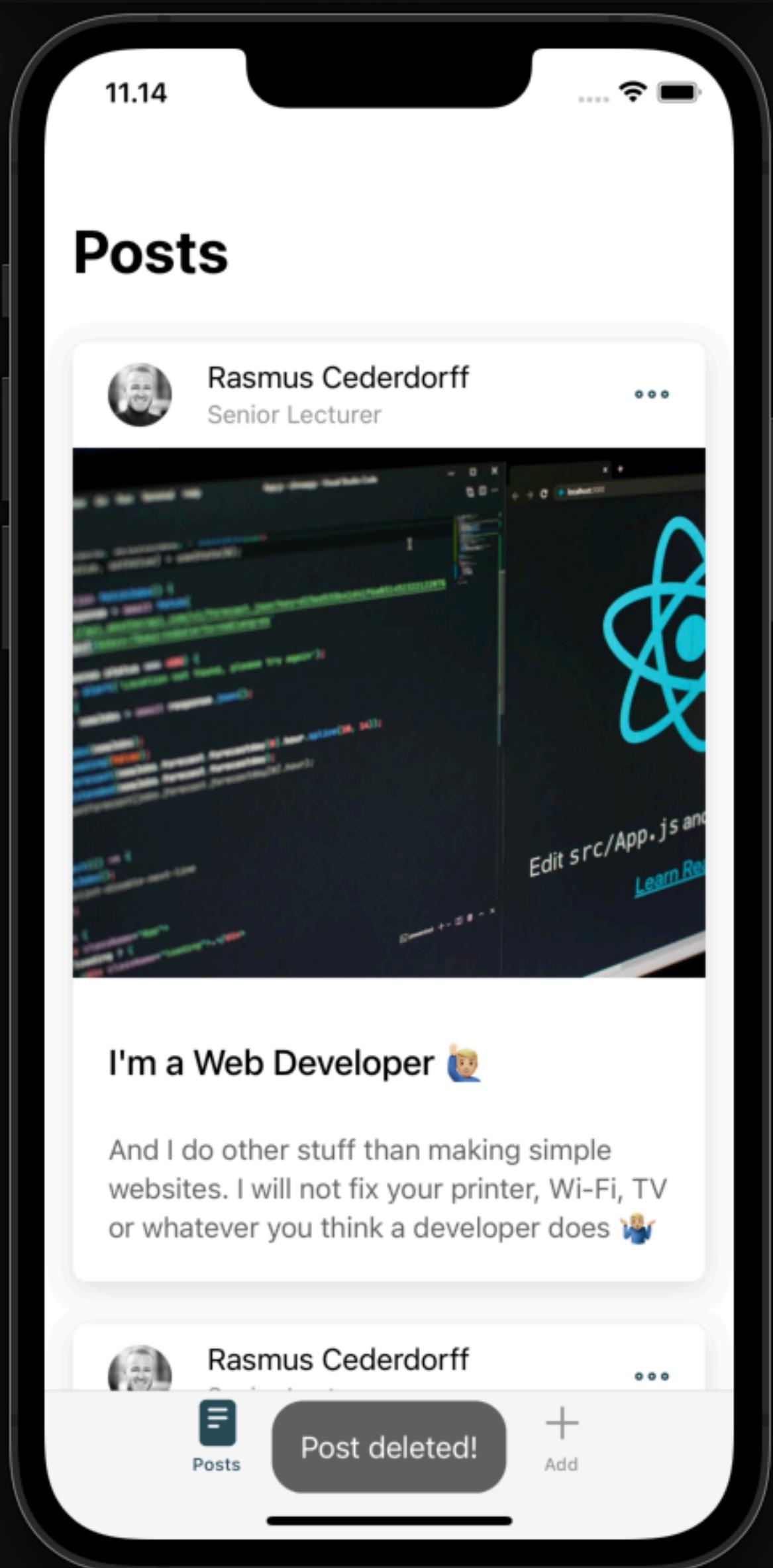
const Home = () => {
  const [lat, setLat] = useState("");
  const [long, setLong] = useState("");
  const [location, setLocation] = useState("");

  async function getCurrentPosition() {
    const coordinates = await Geolocation.getCurrentPosition();
    console.log("Current position:", coordinates);
    setLat(coordinates.coords.latitude);
    setLong(coordinates.coords.longitude);
    getLocation(coordinates.coords.latitude, coordinates.coords.longitude);
  }

  async function getLocation(latitude, longitude) {
    const key = "14de60fd94d4d8753fe8a277ba88667a";
    const res = await fetch(`http://api.positionstack.com/v1/reverse?access_key=${key}&lat=${latitude}&lon=${longitude}`);
    const result = await res.json();
    console.log(result);
    const loc = result.data[0];
  }
}
```

# @capacitor/toast

Pop up notification



```
import { Toast } from '@capacitor/toast';

const showHelloToast = async () => {
  await Toast.show({
    text: 'Hello!',
  });
};
```

# Icons & Splash Screens

<https://race.notion.site/Splash-Screen-and-Icons-be395d6b4b8b49e896797d1eb36e2fb2>

Ionic App Project Description x +

https://race.notion.site/Ionic-App-Project-Description-6dc82eb47b30480c908e40c141326426 Incognito

MAD - Ionic React / Ionic App Project Description Search Duplicate ... Try Notion

# Ionic App Project Description

Develop a Mobile App based on themes, concept and terms from the first part of the Mobile App Dev Course.

The Mobile App is developed in groups of 2-3 students. The project is based on a self-chosen theme and must meet the below technical requirements, demonstrating your learned skills and competencies from the course.

There are no requirements in addition to research and project management but feel free to use known project tools. However the main purpose is to practice programming and development of mobile apps with React, Ionic React and Capacitor JS.

The project is finalised by handing in a link to a GitHub repository including a README.md file with a short description of your app.

[Ionic App Project Description](#)

# Next Tuesday

The screenshot shows a Notion page with a yellow header featuring the Notion logo and several small icons related to app development. The main title of the page is "5. Firebase & App Dev". Below the title, there are three data cards: Date (03/03/2022), Teacher (RACE), and Course (MAD). The "Themes" section lists the following topics:

- Firebase
- Authentication & User Management
- Cloud Storage & Push Notifications
- Ionic Project
- Recap on chosen themes: <https://eaaa.padlet.org/race/mad>
- Next Steps