

race.jsx is loading...



Check out:

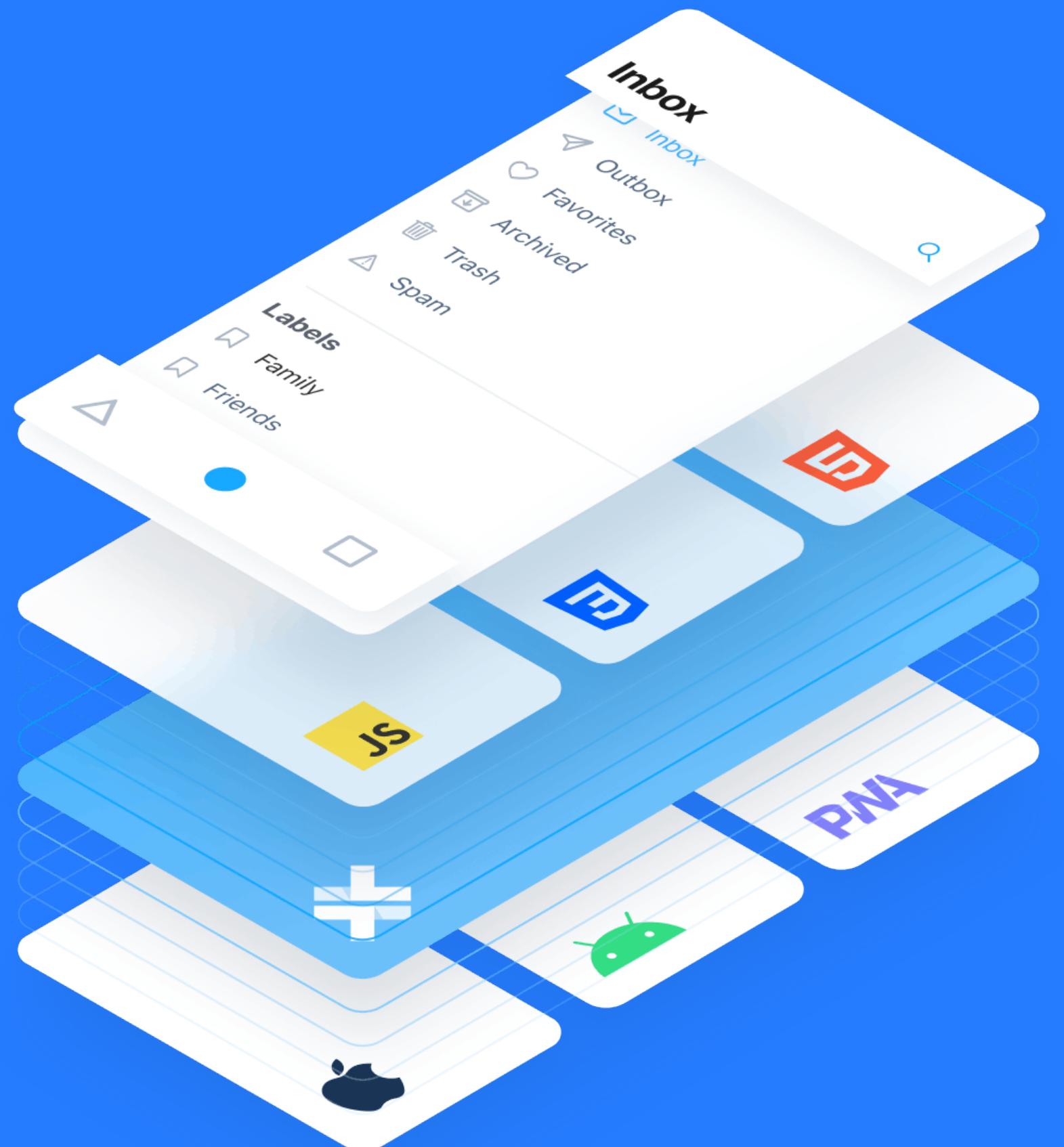
[react-cdn-firebase-rest-post-app](#) ([code](#))

Web Native, Tools & UI Components





Agenda



Ionic React (& TypeScript)
Ionic Core Concepts
UI Components
Mobile Navigation & Routing
(Mobile UI & UX)

Uge 36				
	Mandag d. 05 - 09	Tirsdag d. 06 - 09	Onsdag d. 07 - 09	Fredag d. 09 - 09
08:30 - 10:00	Advanced Web Programming Bemærk: AWP Project	Mobile RACE eaa-R104-1.12	Progressive Web Apps DOB eaa-R104-1.12	Advanced Web Programming DOB eaa-R104-1.12
10:30 - 12:00	Advanced Web Programming Bemærk: AWP Project	Mobile RACE eaa-R104-1.12	Progressive Web Apps DOB eaa-R104-1.12	Advanced Web Programming DOB eaa-R104-1.12
12:30 - 14:00	Advanced Web Programming Bemærk: AWP Project	Mobile RACE eaa-R104-1.12	Progressive Web Apps DOB eaa-R104-1.12	Advanced Web Programming DOB eaa-R104-1.12
14:15 - 15:45				
Uge 37				
	Mandag d. 12 - 09	Tirsdag d. 13 - 09	Onsdag d. 14 - 09	Fredag d. 16 - 09
08:30 - 10:00	Mobile Bemærk: Ionic Project	Mobile Bemærk: Ionic Project	Progressive Web Apps DOB eaa-R104-1.12	Advanced Web Programming DOB eaa-R104-1.12
10:30 - 12:00	Mobile Bemærk: Ionic Project	Mobile Bemærk: Ionic Project	Progressive Web Apps DOB eaa-R104-1.12	Advanced Web Programming DOB eaa-R104-1.12
12:30 - 14:00	Mobile Bemærk: Ionic Project	Mobile Bemærk: Ionic Project	Progressive Web Apps DOB eaa-R104-1.12	Advanced Web Programming DOB eaa-R104-1.12
14:15 - 15:45				
Uge 38				
	Mandag d. 19 - 09	Tirsdag d. 20 - 09	Onsdag d. 21 - 09	Fredag d. 23 - 09
08:30 - 10:00	Advanced Web Programming Bemærk: AWP Project	Mobile RACE eaa-R104-1.12	Progressive Web Apps DOB eaa-R104-1.12	Advanced Web Programming DOB eaa-R104-1.12
10:30 - 12:00	Advanced Web Programming Bemærk: AWP Project	Mobile RACE eaa-R104-1.12	Progressive Web Apps DOB eaa-R104-1.12	CV Workshop Bemærk: Online: 11.00-12.30
12:30 - 14:00	Advanced Web Programming Bemærk: AWP Project	Mobile RACE eaa-R104-1.12	Progressive Web Apps DOB eaa-R104-1.12	CV Workshop Bemærk: Online: 11.00-12.30
14:15 - 15:45		Internship info LSKJ eaa-R104-1.12		
Uge 39				
	Mandag d. 26 - 09	Tirsdag d. 27 - 09	Onsdag d. 28 - 09	Fredag d. 30 - 09
08:30 - 10:00	Mobile Bemærk: Ionic Project	Mobile RACE eaa-R104-1.12	Progressive Web Apps DOB eaa-R104-1.12	Advanced Web Programming DOB eaa-R104-1.12
10:30 - 12:00	Mobile Bemærk: Ionic Project	Mobile RACE eaa-R104-1.12	Progressive Web Apps DOB eaa-R104-1.12	Job Application Workshop Bemærk: Online: 11.00-12.30
	Mobile	Mobile	Progressive Web Apps	Job Application Workshop
				Advanced Web Programming

Your First Ionic Apps

Ionic User List App

Thinking in React?

Components, components, components

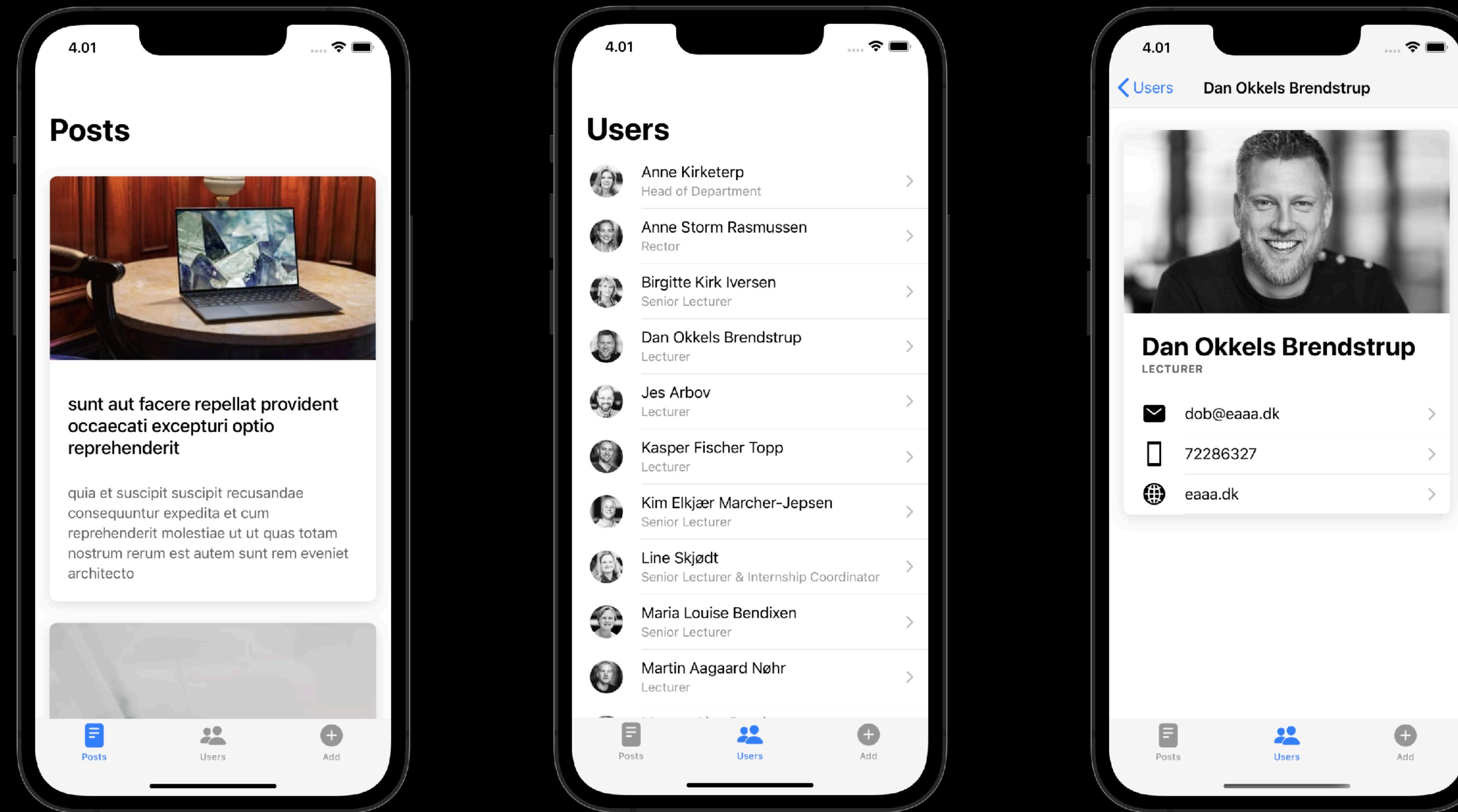
Why lifecycle methods?

Ionic Post App w/ Firebase REST

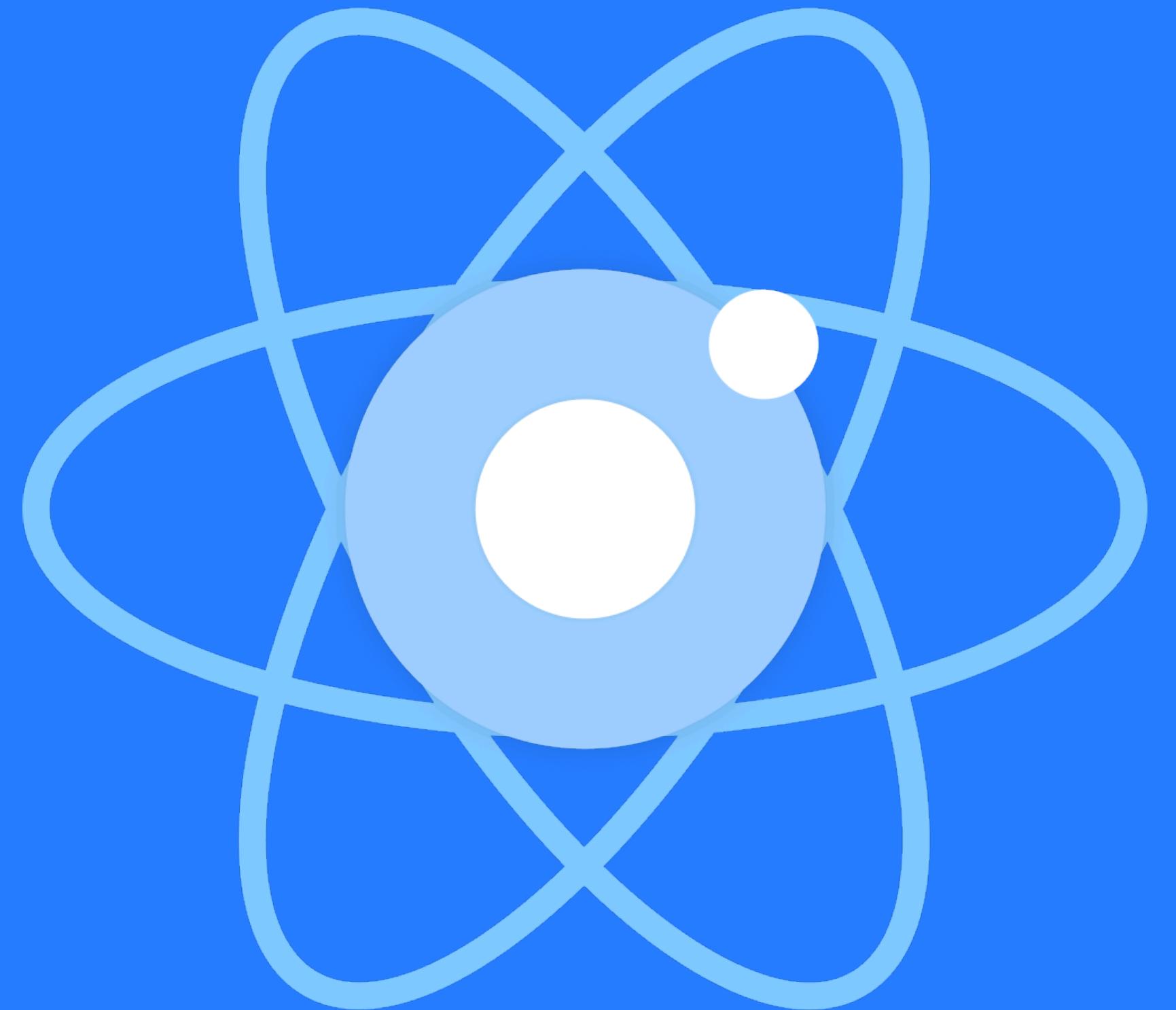
Ionic Camera App



What we are developing...



ionic-post-app-firebase-rest-v6



Key Takeaways?

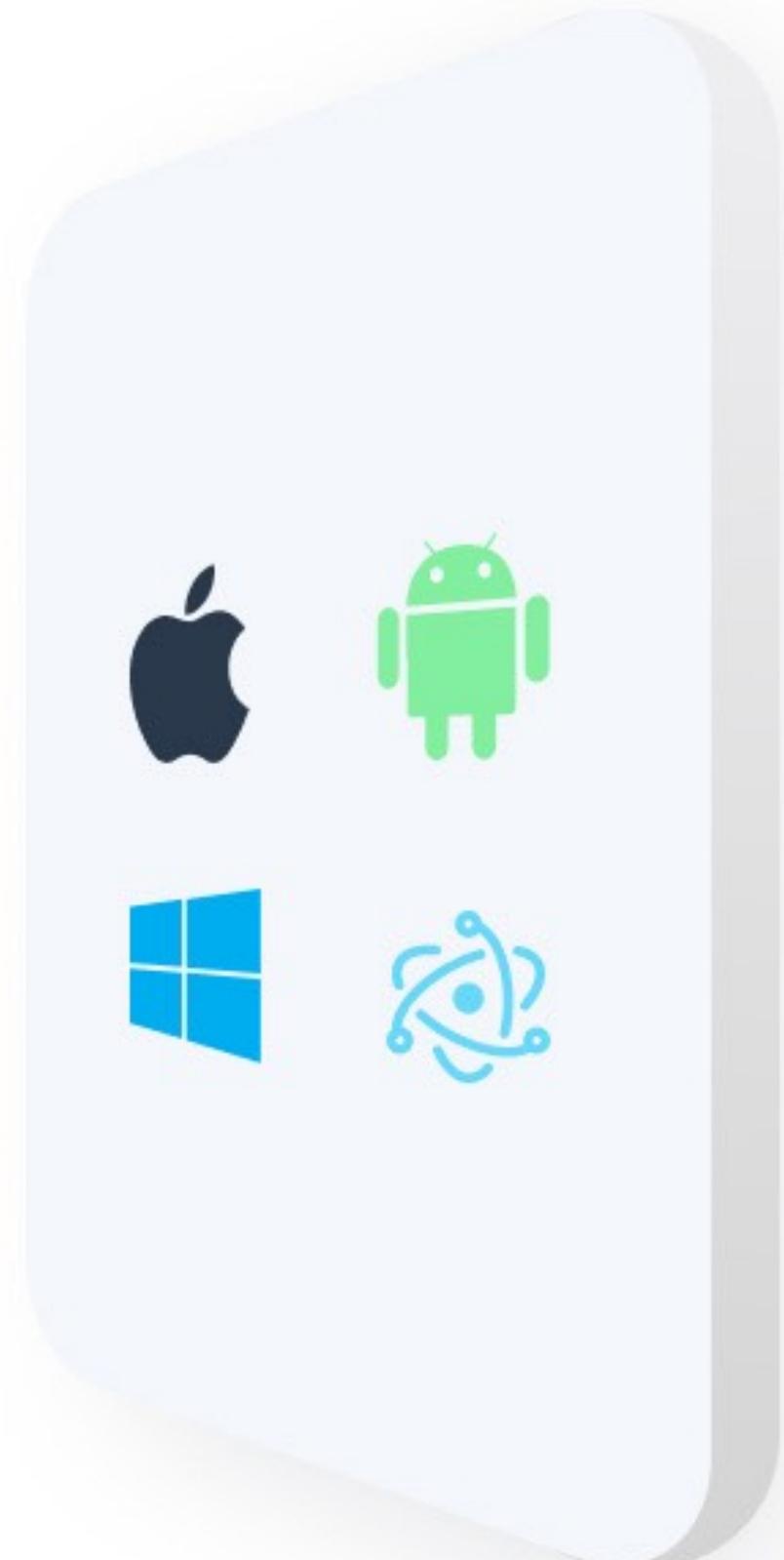
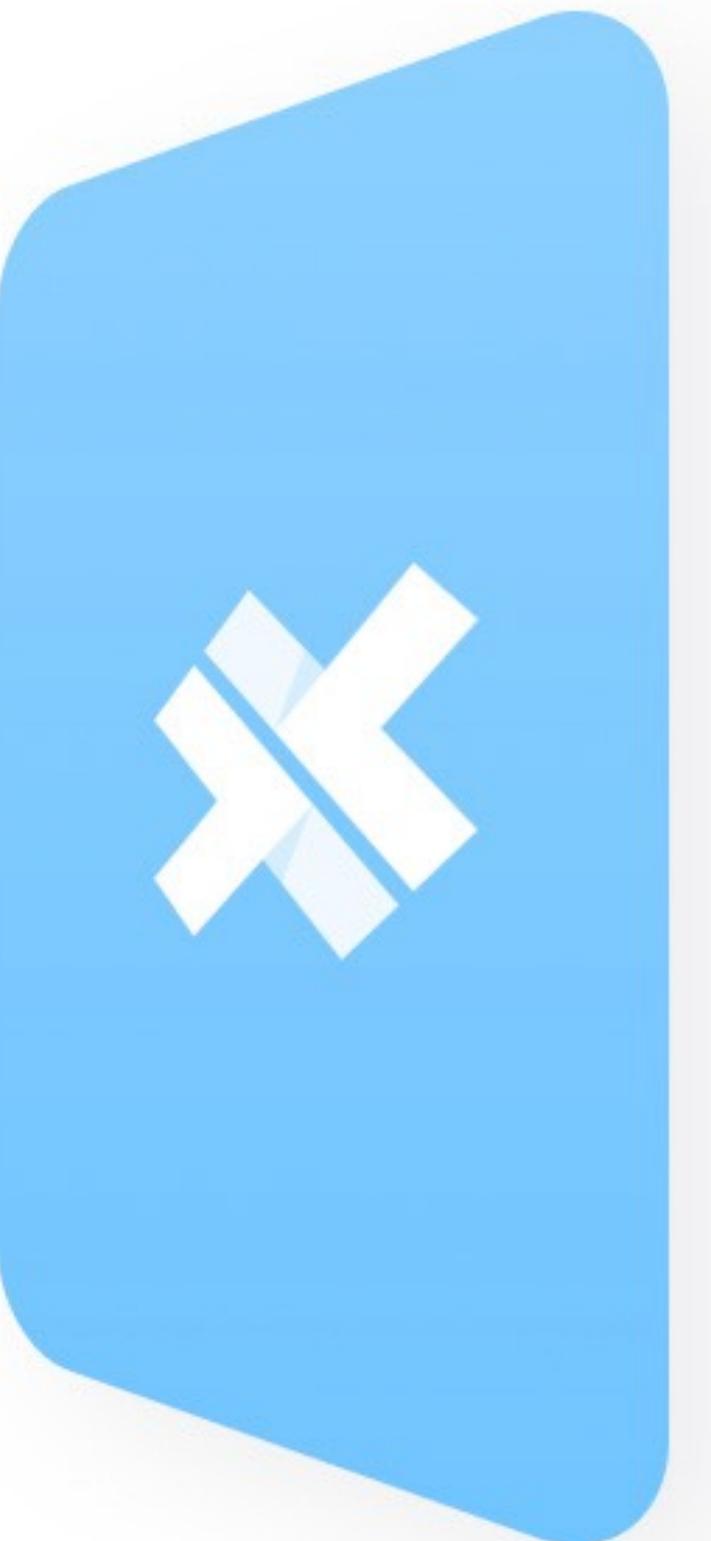
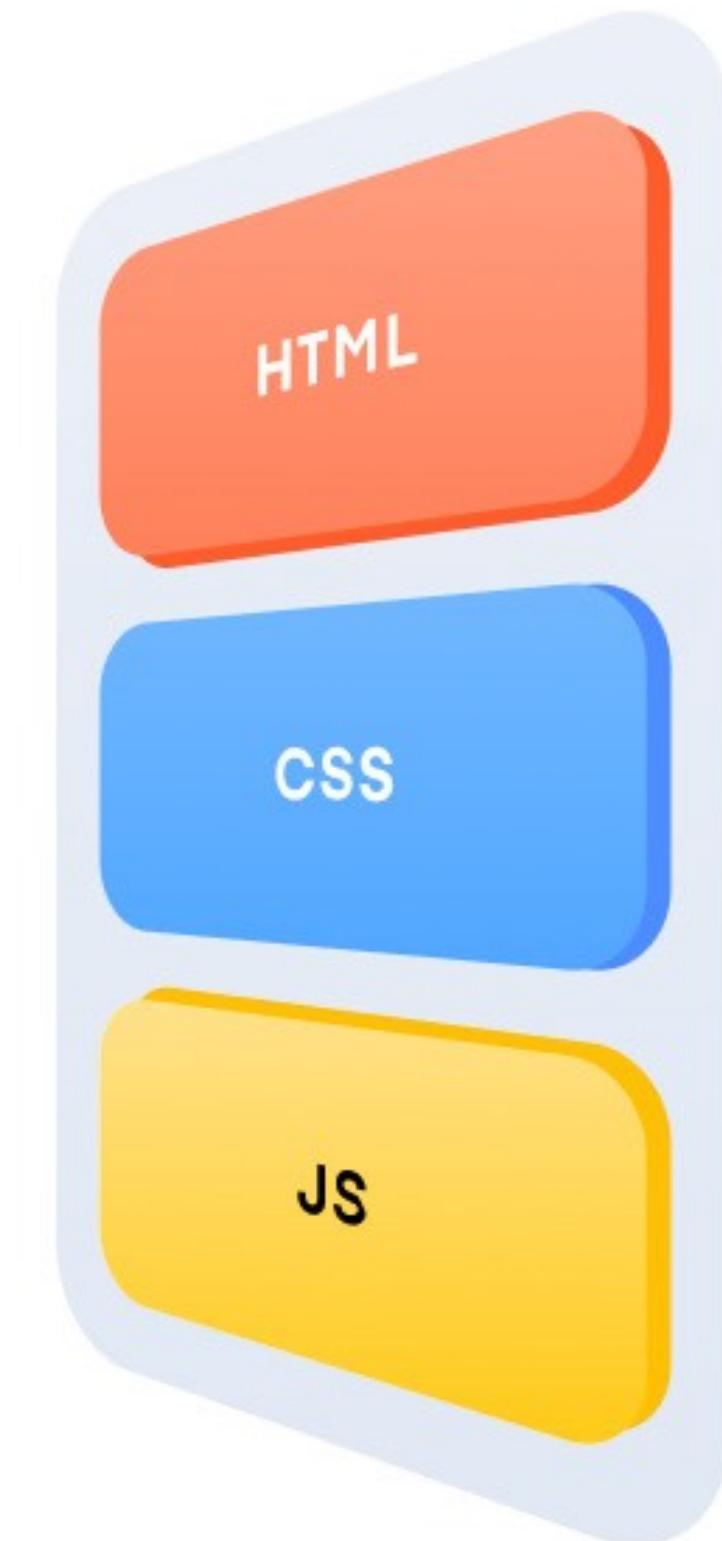
... from your first week on 2nd
semester?

when you ask Rasmus
for help and he says
"Read documentation"



Please, do not try to
understand everything

But practice and gain a deeper knowledge
over time. Accept the fact that you do not
understand all at the moment.



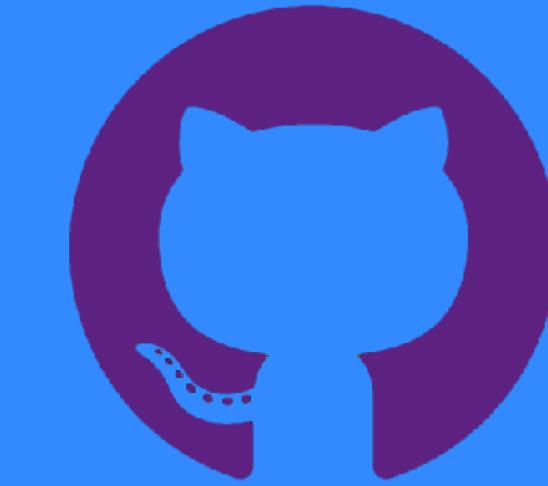
YOUR APP (ANGULAR, REACT, VUE...)

UI CONTROLS (IONIC)

NATIVE ACCESS (CAPACITOR)

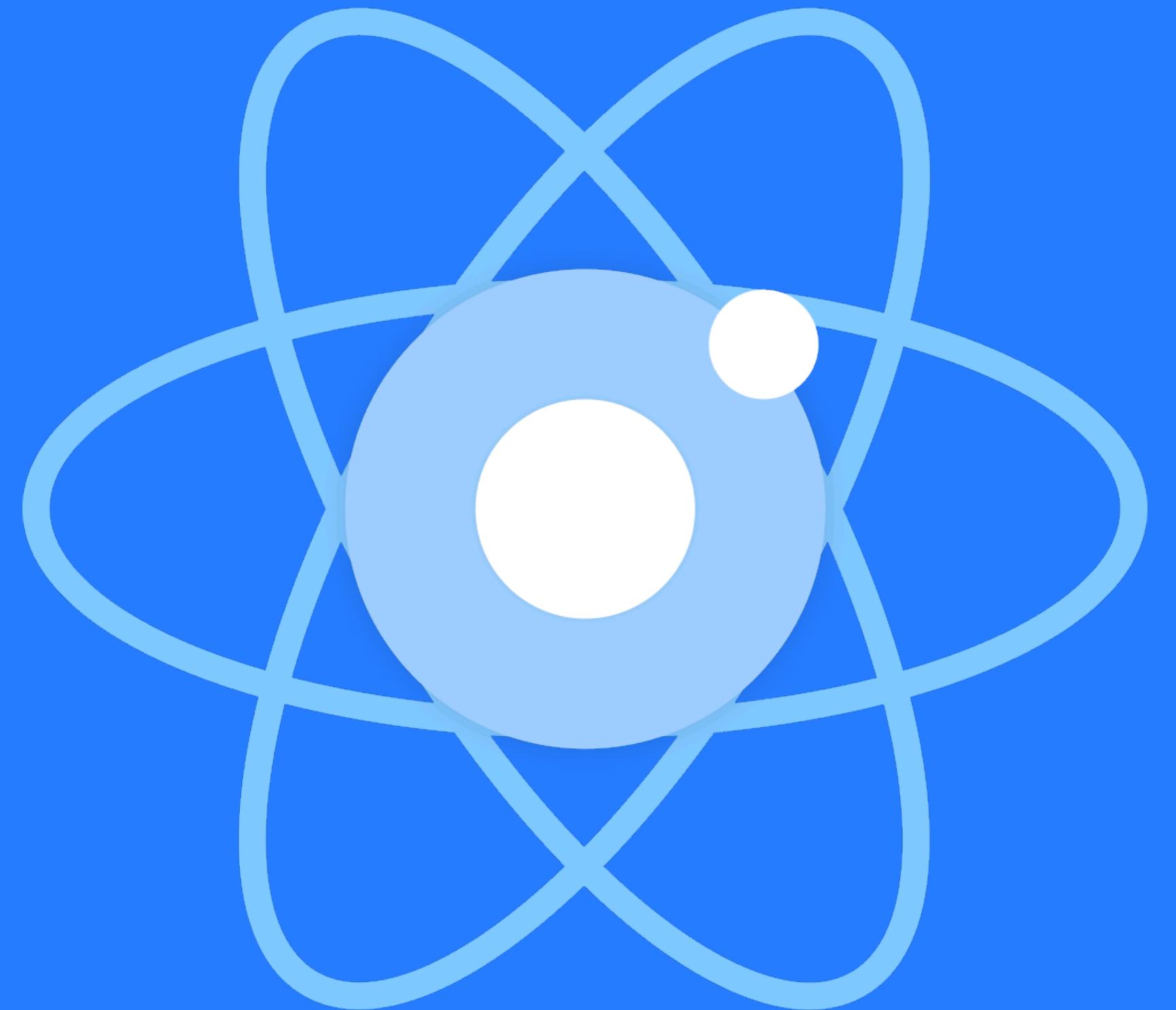
DISTRIBUTION PLATFORMS

CLI & Tools



You develop 90% in the
browser

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

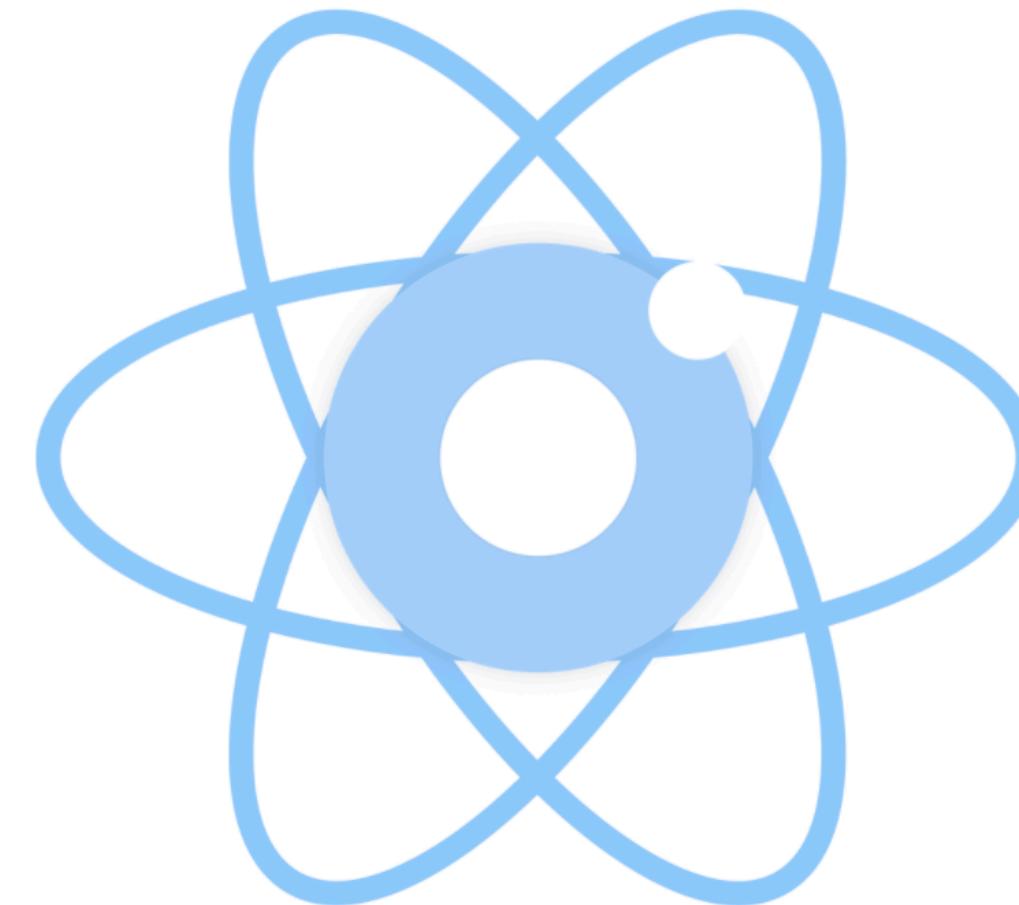


Ionic React

Build Native and Progressive
Web Apps from a single code
base with Ionic React.

Ionic is a Cross-platform

**One Codebase
Any Platform
Just React**



- ✓ 100+ mobile optimized React UI components
- ✓ Standard React tooling with react-dom
- ✓ iOS / Android / Electron / PWA

**Build with whatever
you prefer**
Angular, React, Vue, or
Vanilla JavaScript



Ionic Framework



Installation Guide

Step-by-step guides to setting up your system and installing the framework.



Native Functionality

Integrate native device plugins, like Bluetooth, Maps, HealthKit, and more.



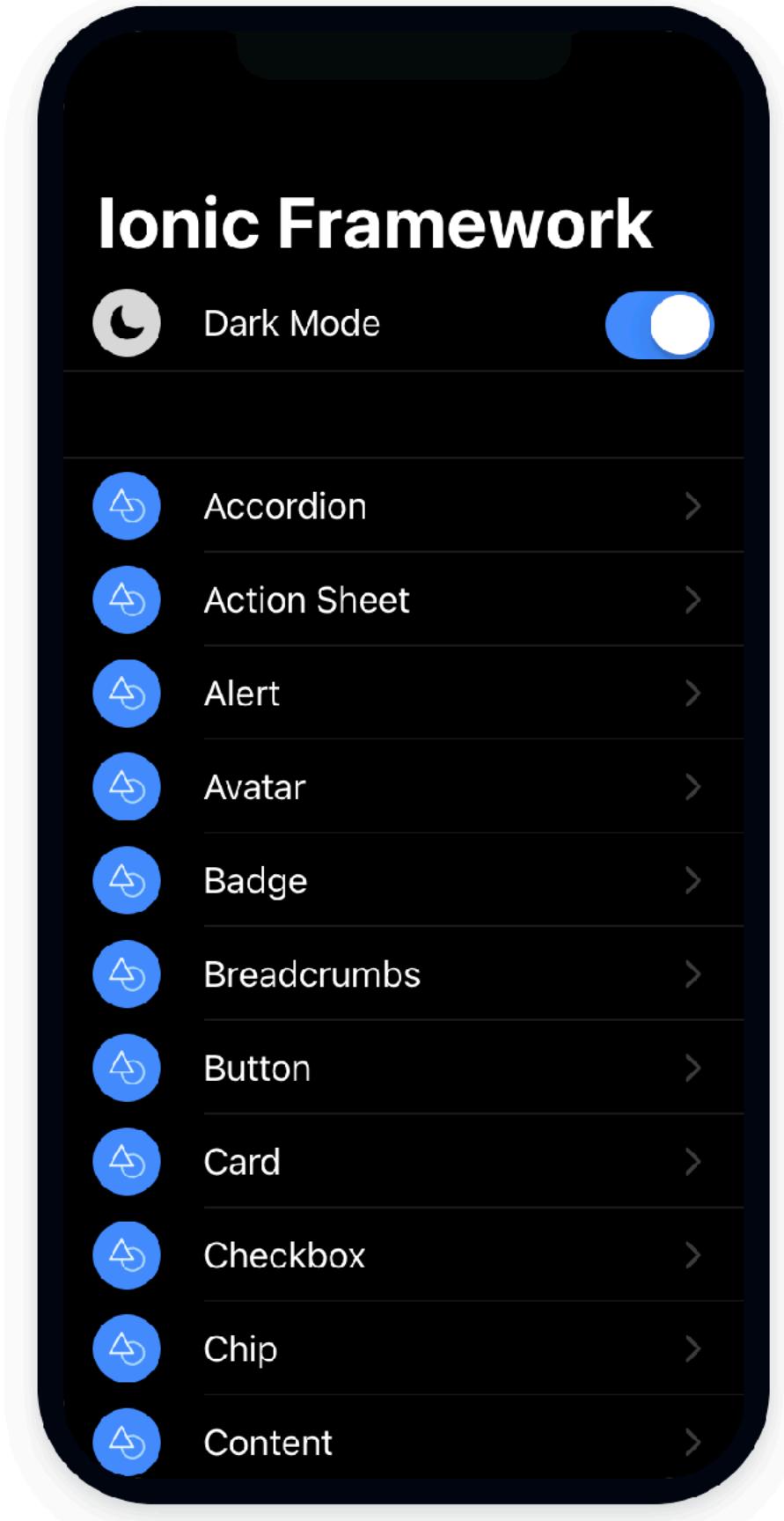
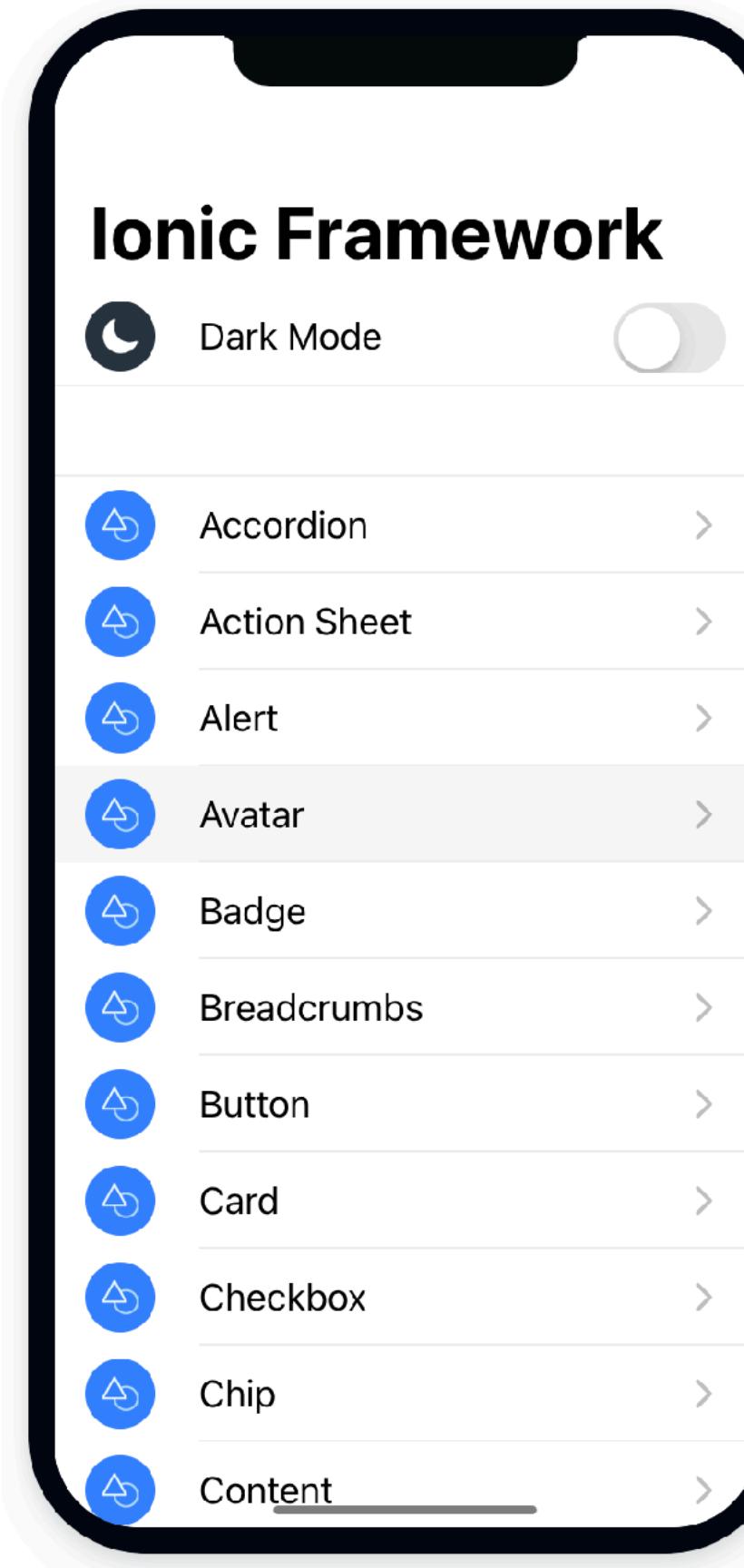
UI Components

Dive into Ionic beautifully designed UI component library.



Theming

Learn to easily customize and modify your Ionic app's visual design to fit your brand.



<https://ionicframework.com/>

Web Native

“Web Native is the idea that teams should build modern Web Apps and combine them with tools like Capacitor that unlock powerful Native APIs to the app for the platform its running on.”

“Web Native is the full capability and access to developers of the Web Platform, with the full functionality and performance benefits of traditional native apps.”

“Web Native is "hybrid" done right, and it's the future of mobile app development.”

<https://webnative.tech/>

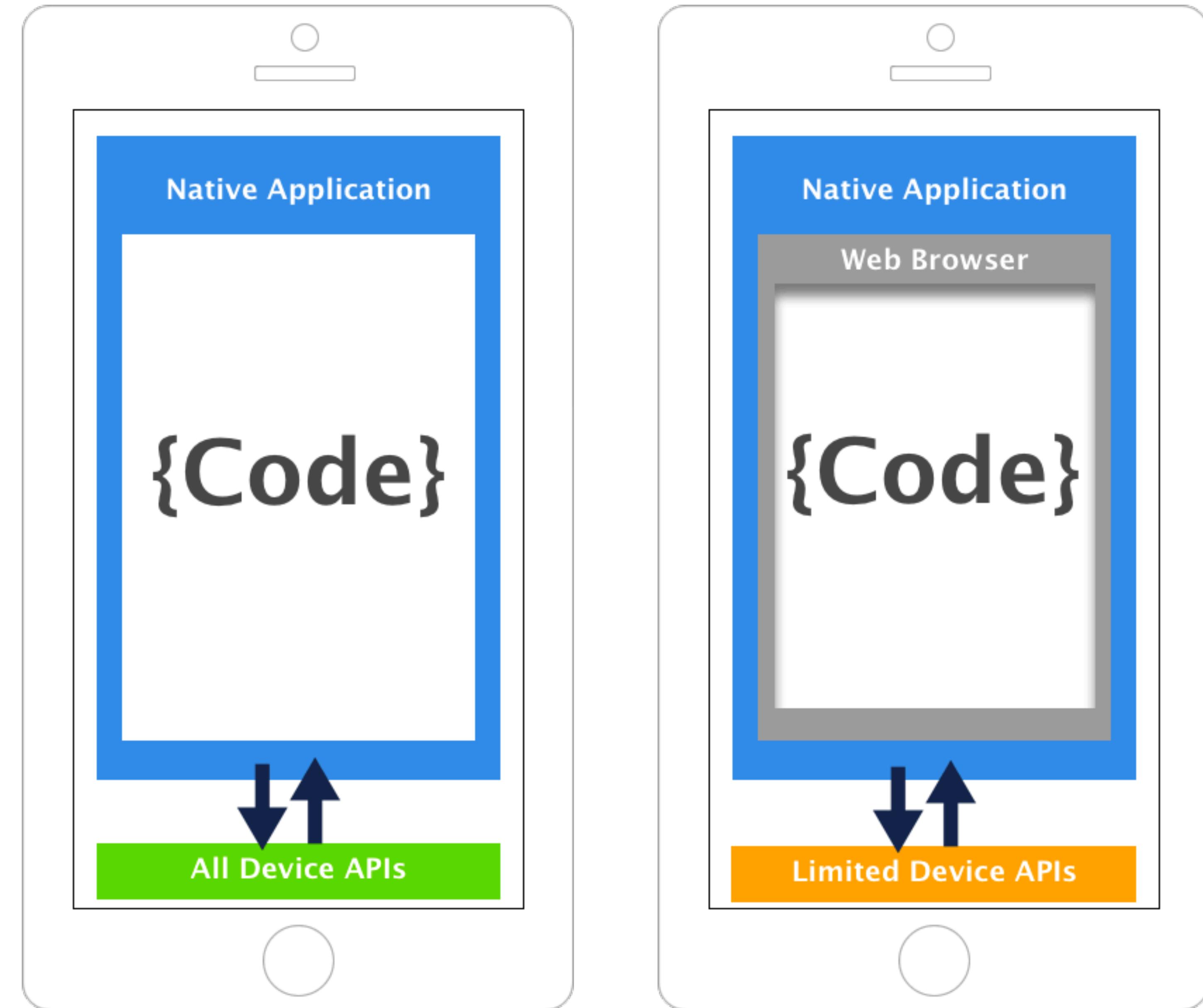


Capacitor

A cross-platform native runtime for Web Native 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.





Back in the 2010s (Cordova)

Feature	Native	Web-only	Hybrid
Device Access	Full	Limited	Limited
Performance	High	Medium to High	Low
Development Language	Platform Specific	HTML, CSS, Javascript	HTML, CSS, Javascript
Cross-Platform Support	No	Yes	Yes
User Experience	High	Medium to High	Low
Code Reuse	No	Yes	Yes



Runs web apps
across multiple
platforms

Hybrid Apps

Today (Capacitor)

Feature	Native	Web-only	Hybrid
Device Access	Full	Limited	Full (with plugins)
Performance	High	Medium to High	Medium to High
Development Language	Platform Specific	HTML, CSS, Javascript	HTML, CSS, Javascript
Cross-Platform Support	No	Yes	Yes
User Experience	High	Medium to High	Medium to High
Code Reuse	No	Yes	Yes



Runs modern web apps
natively on iOS, Android,
Electron and Web
multiple platforms.

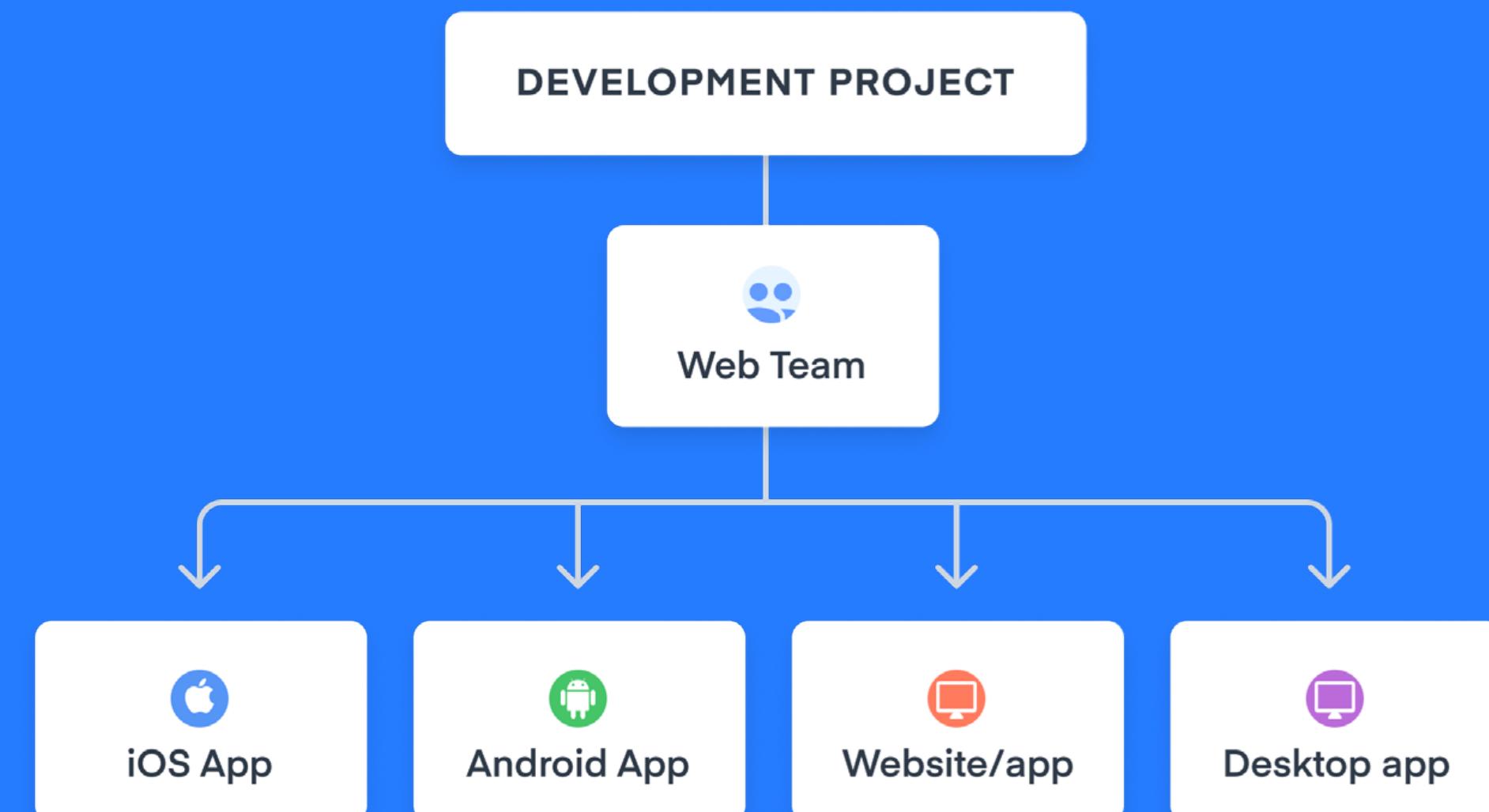
Web Native Apps & PWA

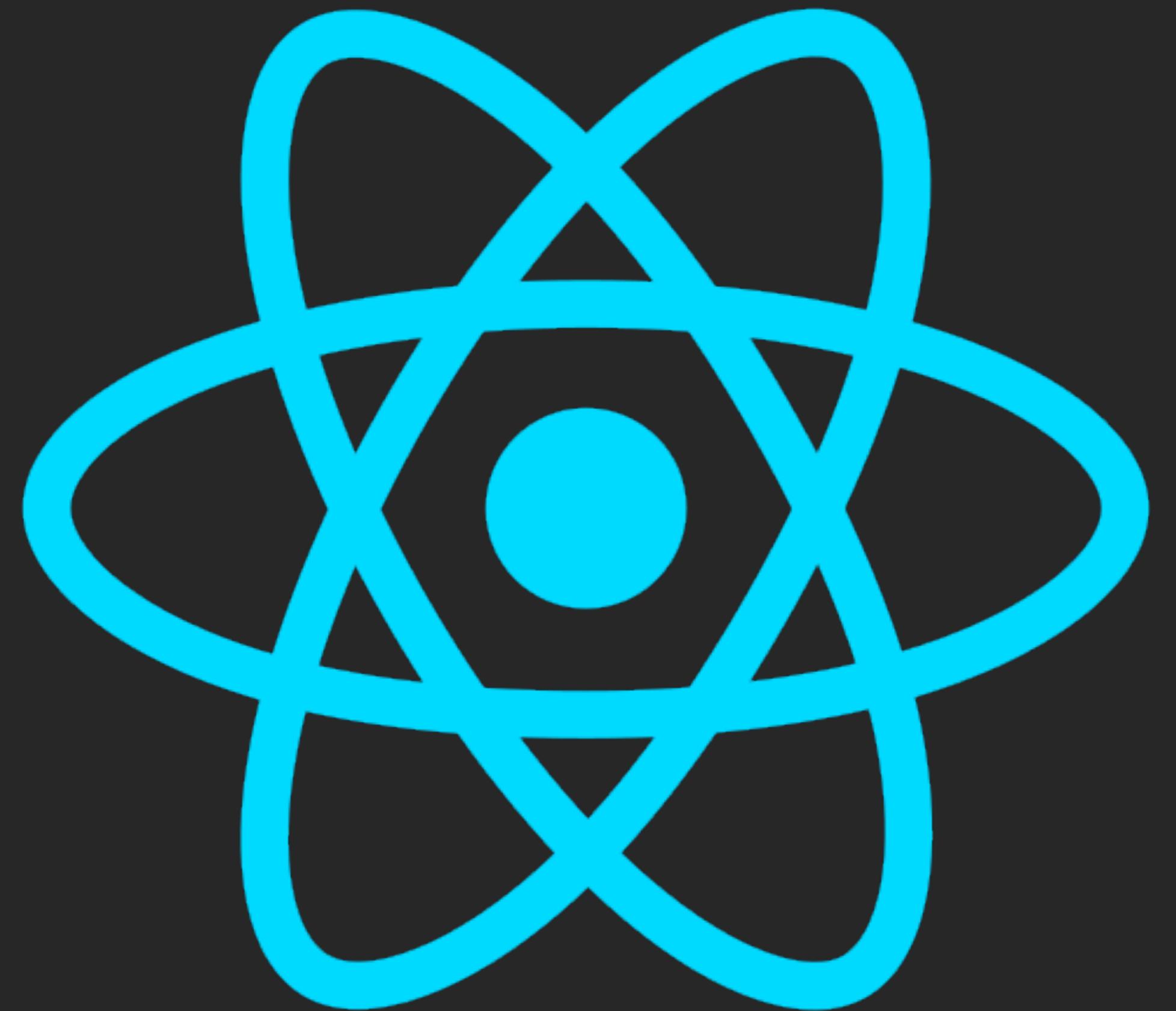
(Re)Use the expertise you already have in-house

Traditional Approach



Ionic (Web Native) Approach





It's Just React
Ionic React is just React
components

JS

Well, at the end of
the day, it's just
JavaScript





WHAT IS TYPESCRIPT?

A screenshot of a web browser window. The title bar says "API Glossary: Terminology and". The address bar shows "ionicframework.com/docs/reference/glossary#typ...". Below the address bar is a navigation bar with icons for search, refresh, and other functions. The main content area has a sidebar icon and a search bar. The main text is about TypeScript being a superset of JavaScript with type declarations and interfaces. It also mentions that using TypeScript is optional. There is a section titled "Unit Tests" with a note about testing small pieces of code.

TypeScript

TypeScript is a superset of JavaScript, which means it gives you JavaScript, along with a number of extra features such as [type declarations](#) and [interfaces](#). Although Ionic is built with TypeScript, using it to build an Ionic app is completely optional.

Unit Tests

Unit Tests and unit testing are a way to test small pieces of code to see if

<https://ionicframework.com/docs/reference/glossary#typescript>

TypeScript is **JavaScript with syntax for types**.

TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale.

<https://www.typescriptlang.org/>



Usage

Angular Javascript **React** Stencil Vue

ios Android

Action Sheet

ion-action-sheet

Accordion

ion-accordion

ion-accordion-group

Alert

ion-alert

Badge

ion-badge

Breadcrumb

ion-breadcrumb

ion-breadcrumbs

Button

ion-button

ion-ripple-effect

Card

ion-card

ion-card-content

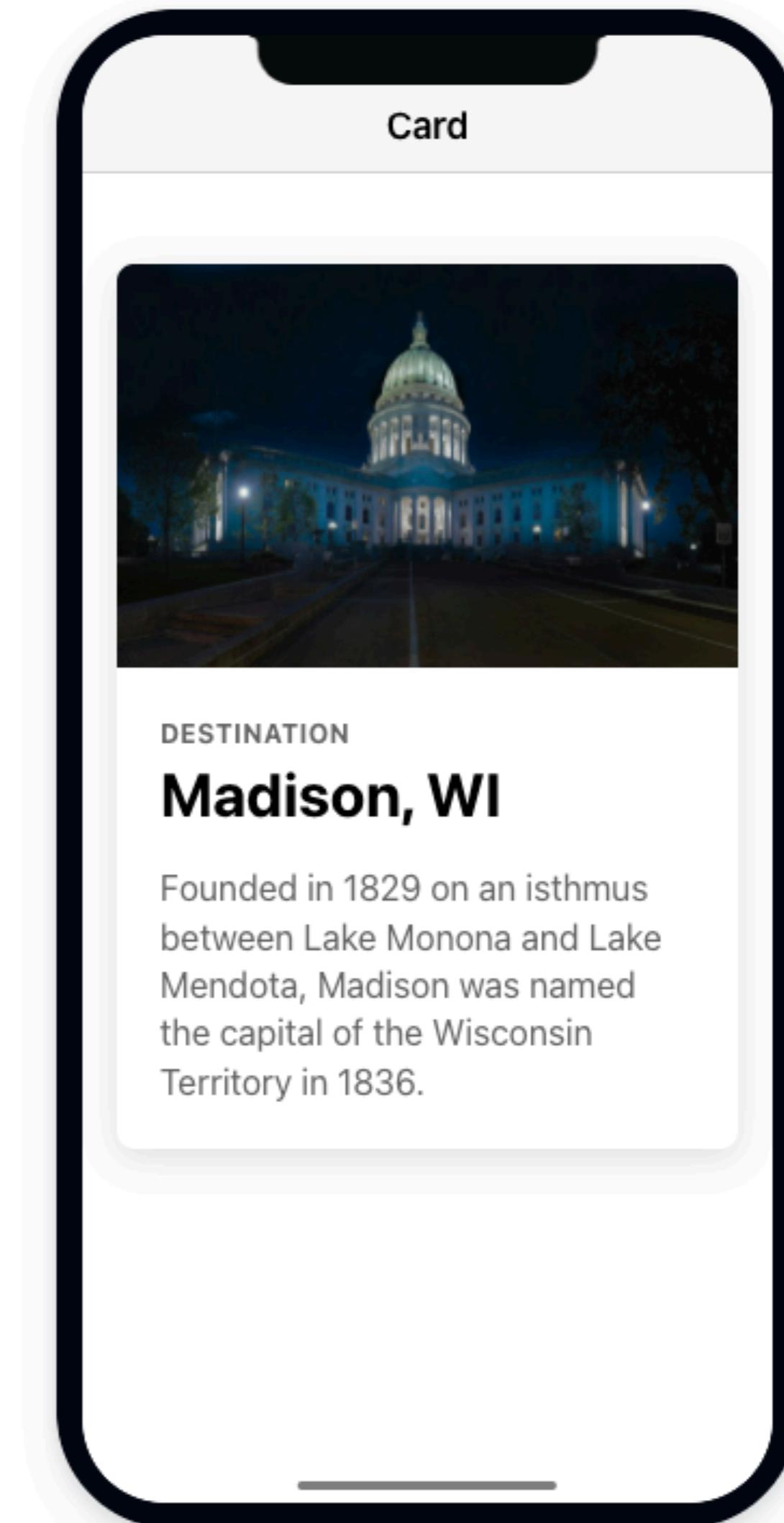
ion-card-header

ion-card-subtitle

```
import React from 'react';
import { IonContent, IonHeader, IonPage, IonTitle, IonToolbar, IonCard, IonCardContent } from '@ionic/react';
import { pin, wifi, wine, warning, walk } from 'ionicons/icons';
```

```
export const CardExamples: React.FC = () => {
  return (
    <IonPage>
      <IonHeader>
        <IonToolbar>
          <IonTitle>Card Examples</IonTitle>
        </IonToolbar>
      </IonHeader>
      <IonContent>
        <IonCard>
          <IonCardHeader>
            <IonCardSubtitle>Card Subtitle</IonCardSubtitle>
            <IonCardTitle>Card Title</IonCardTitle>
          </IonCardHeader>
          <IonCardContent>
            Keep close to Nature's heart... and break clear away, once in awhile  

            and climb a mountain or spend a week in the woods. Wash your spirit
          </IonCardContent>
        </IonCard>
        <IonCard>
          <IonItem>
            <IonIcon icon={pin} slot="start" />
            <IonLabel>ion-item in a card, icon left, button right</IonLabel>
            <IonButton fill="outline" slot="end">View</IonButton>
          </IonItem>
        </IonCard>
      </IonContent>
    </IonPage>
  );
}
```



TypeScript for JavaScript Programmers

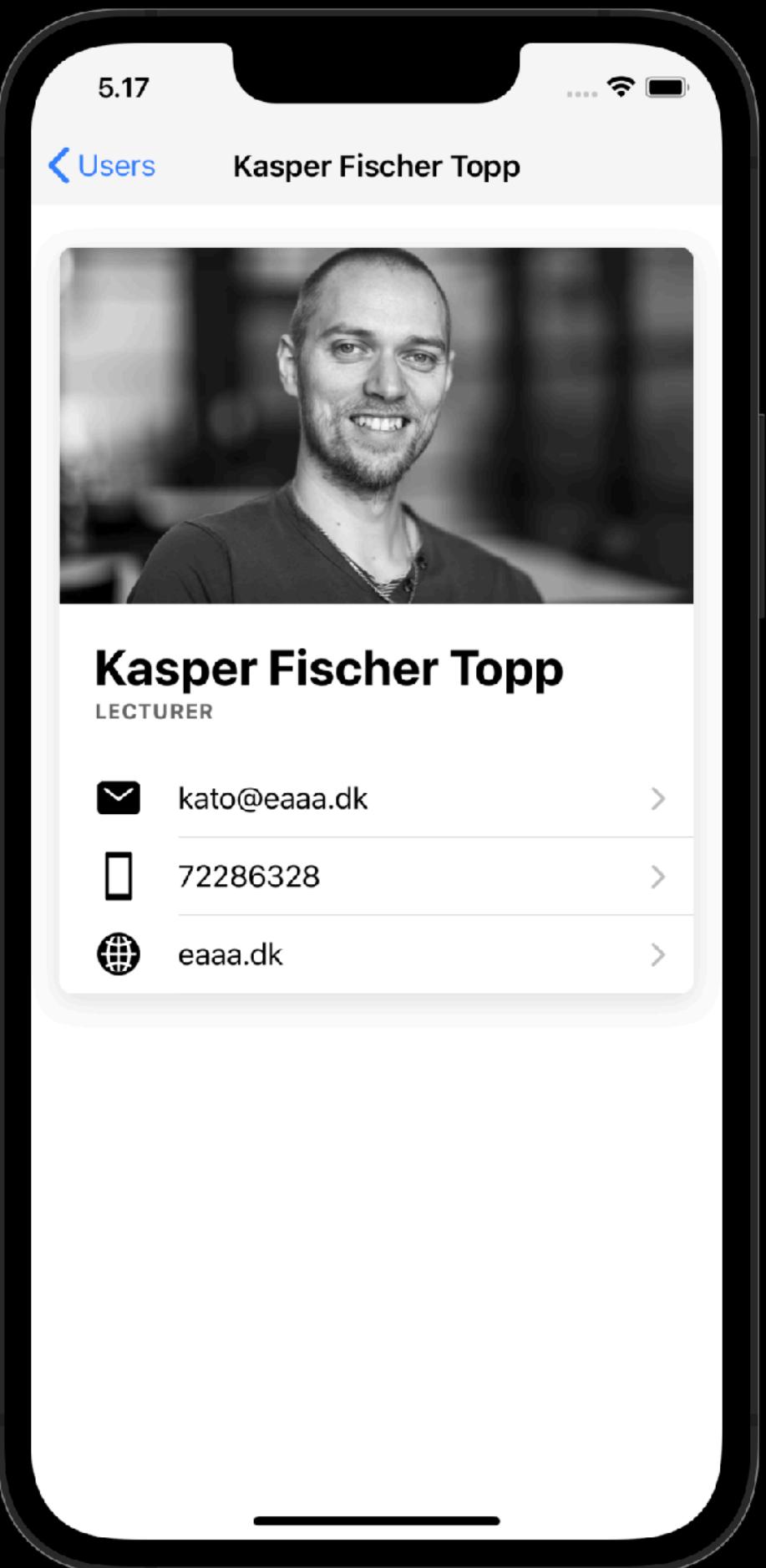
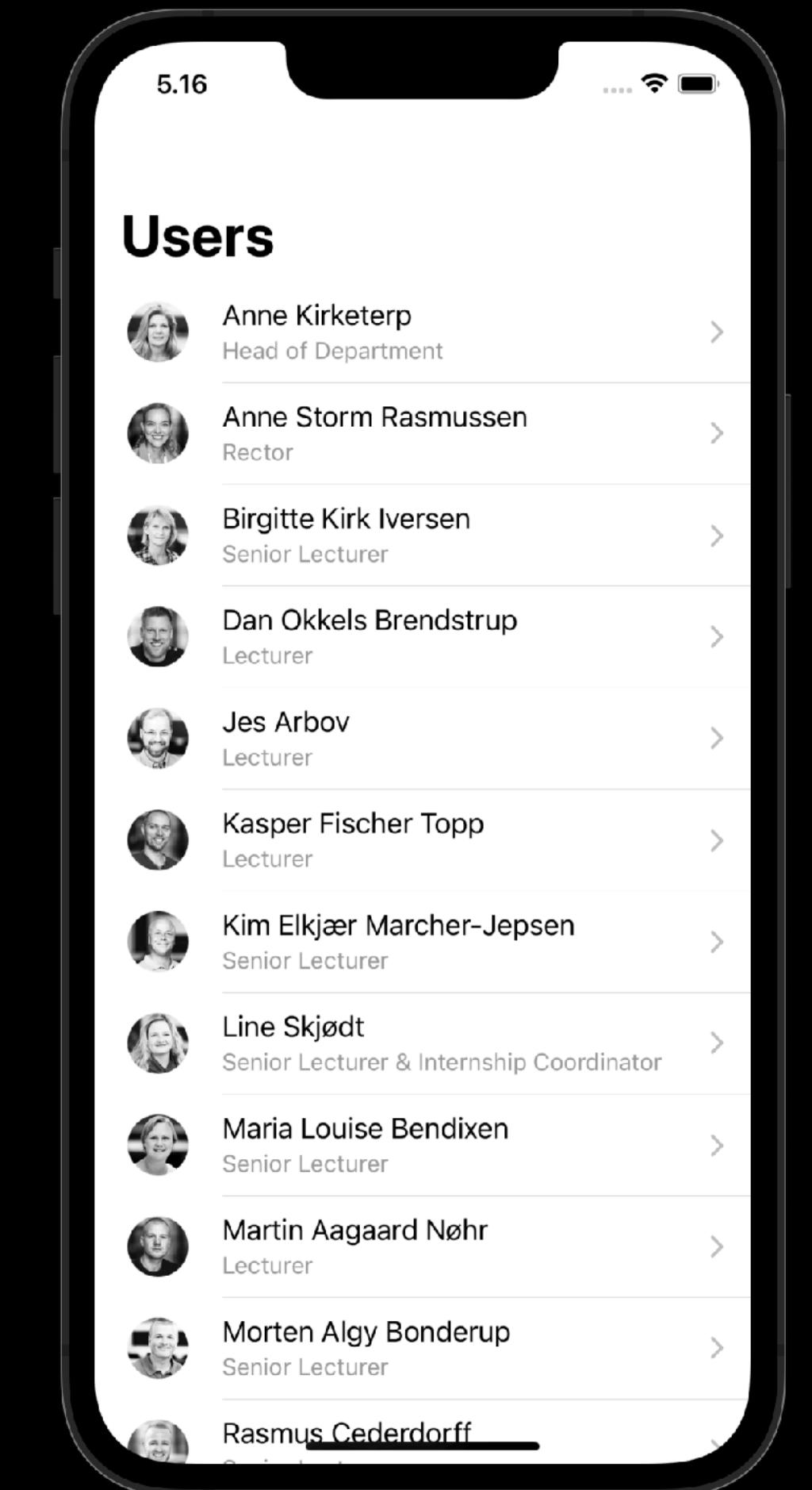
<https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>

Ionic User List App

<https://race.notion.site/User-List-e4d5af63b2de443f946cc2f5a1227716>

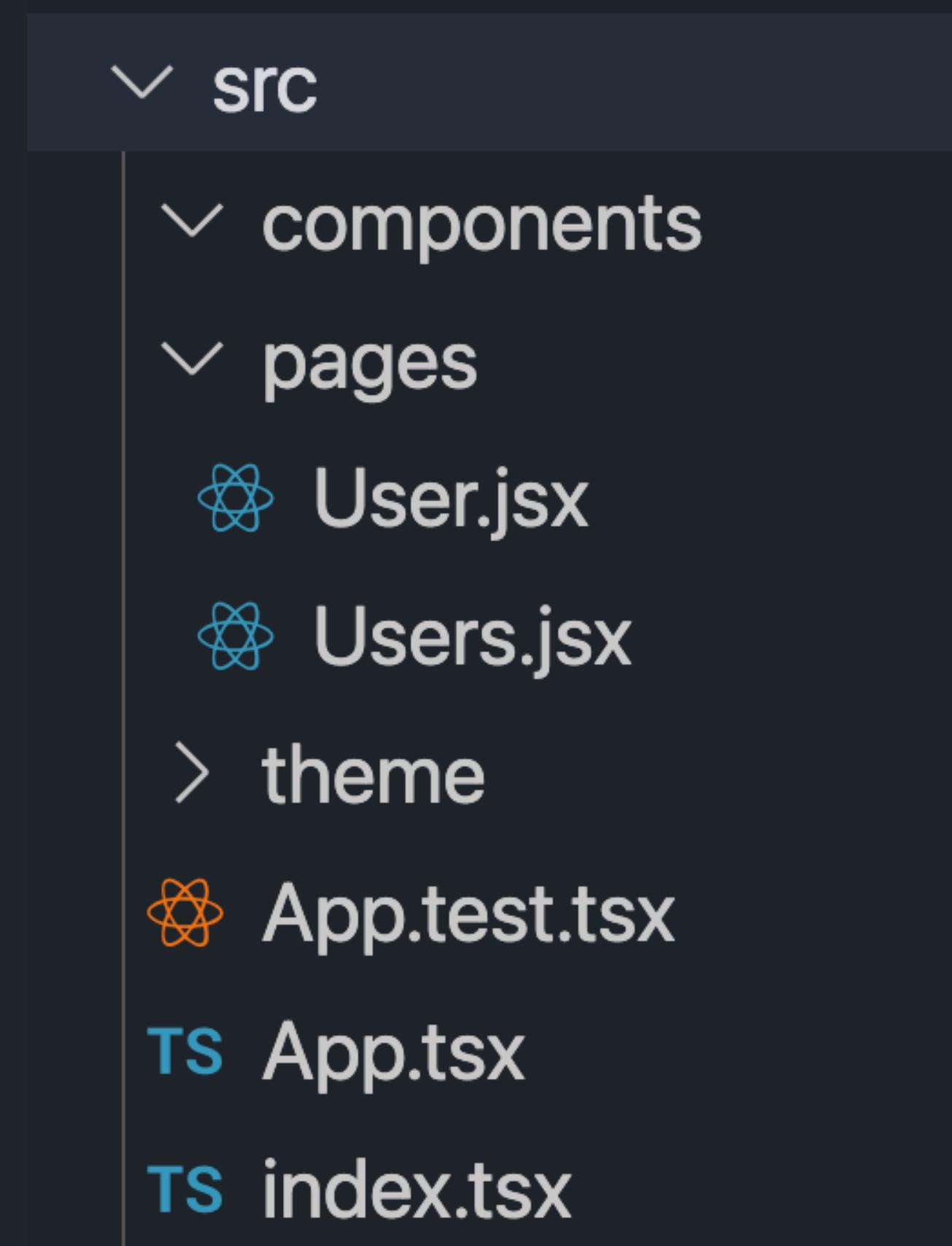
[ionic-user-list](#)

[ionic-user-list-no-ts](#)

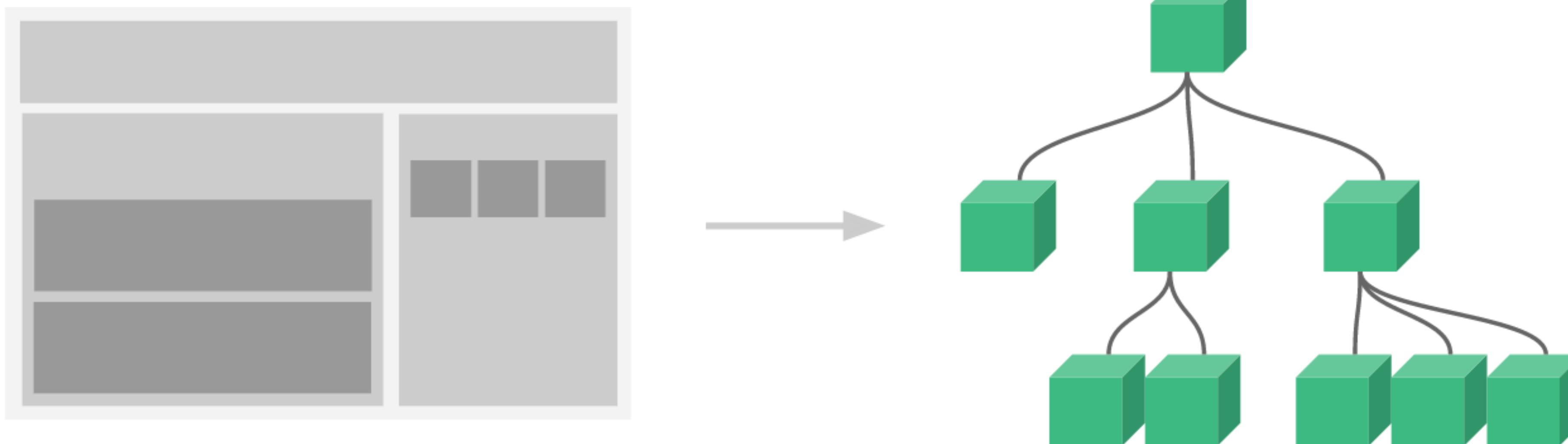


Thinking in React?

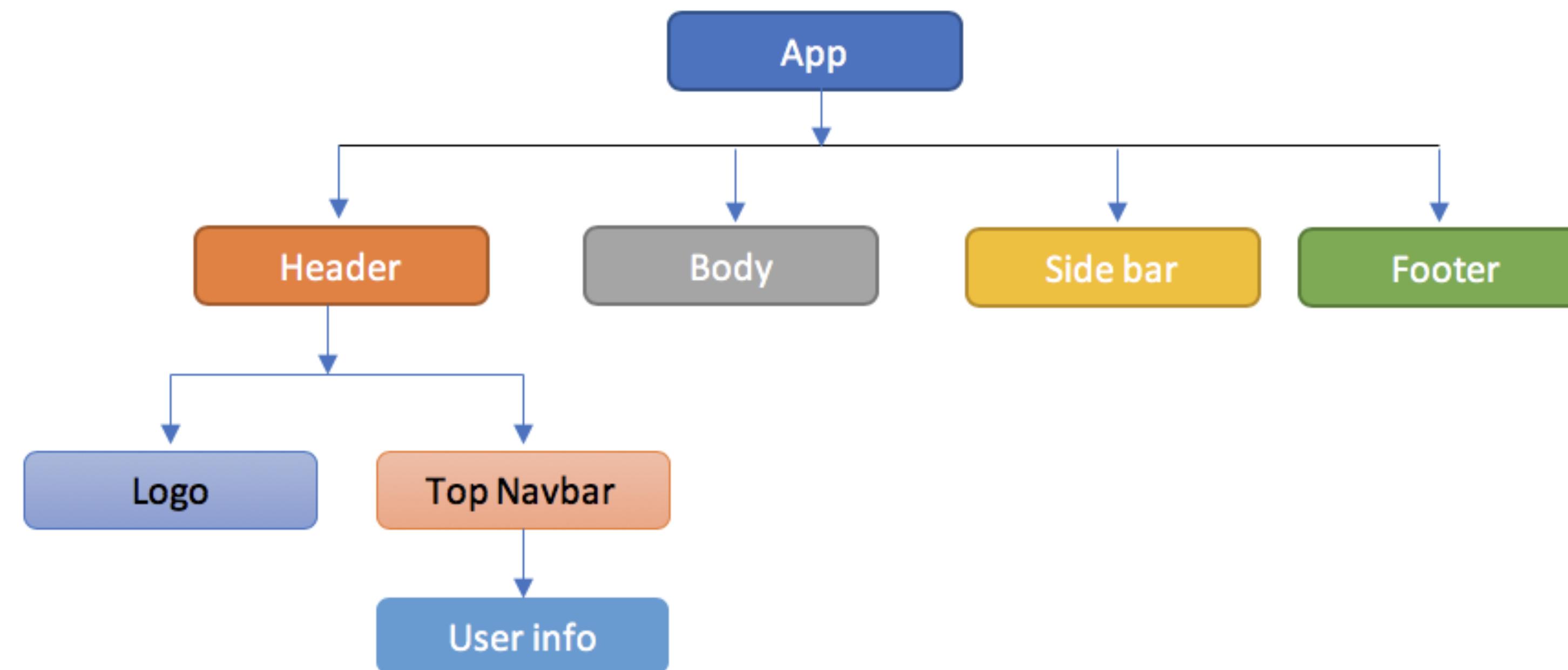
[https://beta.reactjs.org/learn/
thinking-in-react](https://beta.reactjs.org/learn/thinking-in-react)



Components



App structured in components



Structure

It's all components

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

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

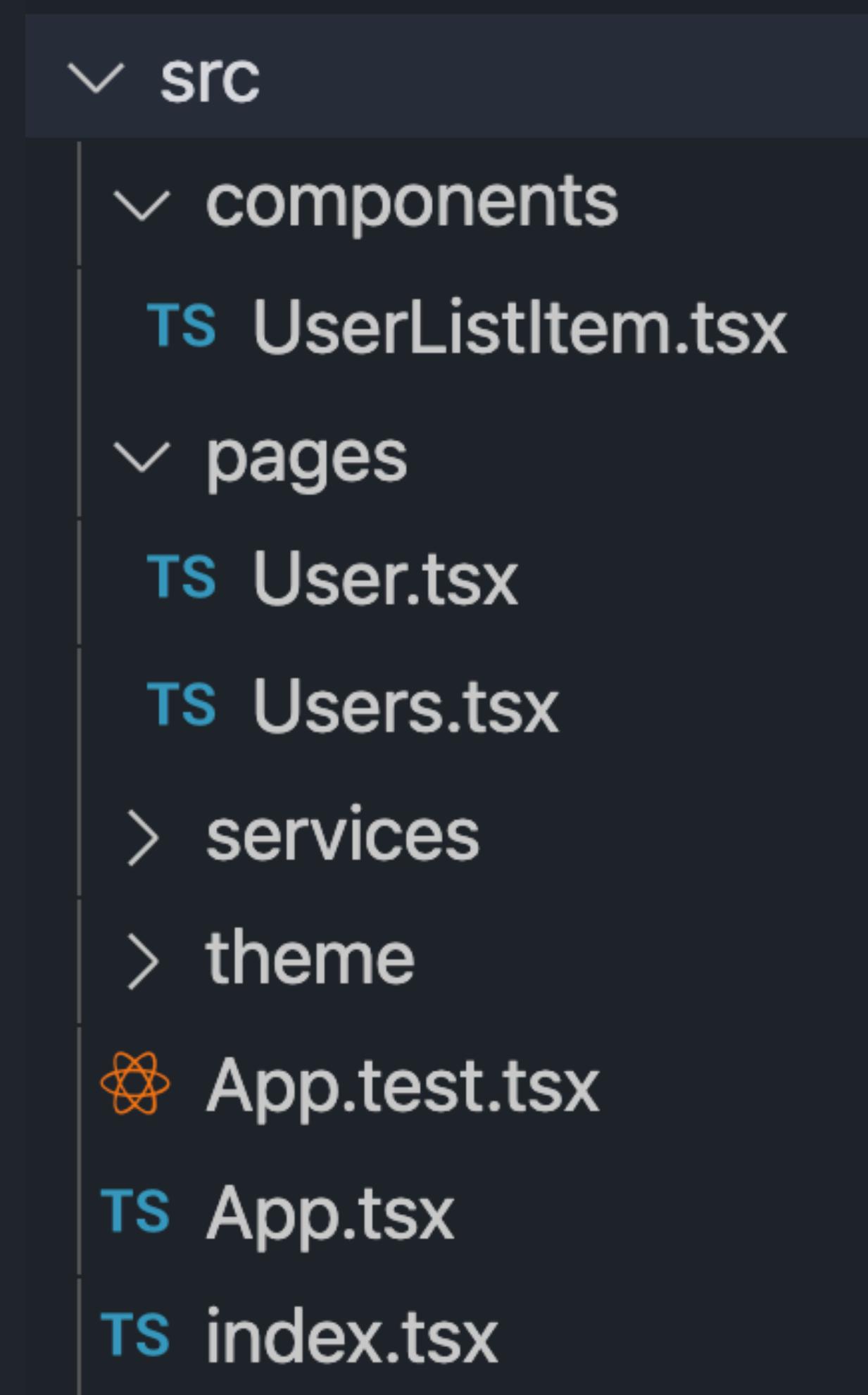
```
import React from 'react';
import { Route } from 'react-router-dom';
import { IonApp, IonRouterOutlet } from '@ionic/react';
import { IonReactRouter } from '@ionic/react-router';
import Home from './pages/Home';

/* Core CSS required for Ionic components to work properly */
import '@ionic/react/css/core.css';

const App: React.FC = () => (
  <IonApp>
    <IonReactRouter>
      <IonRouterOutlet>
        <Route path="/home" component={Home} exact={true} />
        <Route exact path="/" render={() => <Redirect to="/home" />} />
      </IonRouterOutlet>
    </IonReactRouter>
  </IonApp>
)
```

Components, components, components

1. Review your Users component.
2. Think of how to restructure the component and divide it into several components.
3. Create a file and a new component named UserListItem or whatever you prefer.
4. Extract the IonItem to the newly created component. Refactor what's needed.
5. Import and use the new component in Users and make it pass the needed props.



Ionic Core Concepts

<https://ionicframework.com/docs/core-concepts/fundamentals>

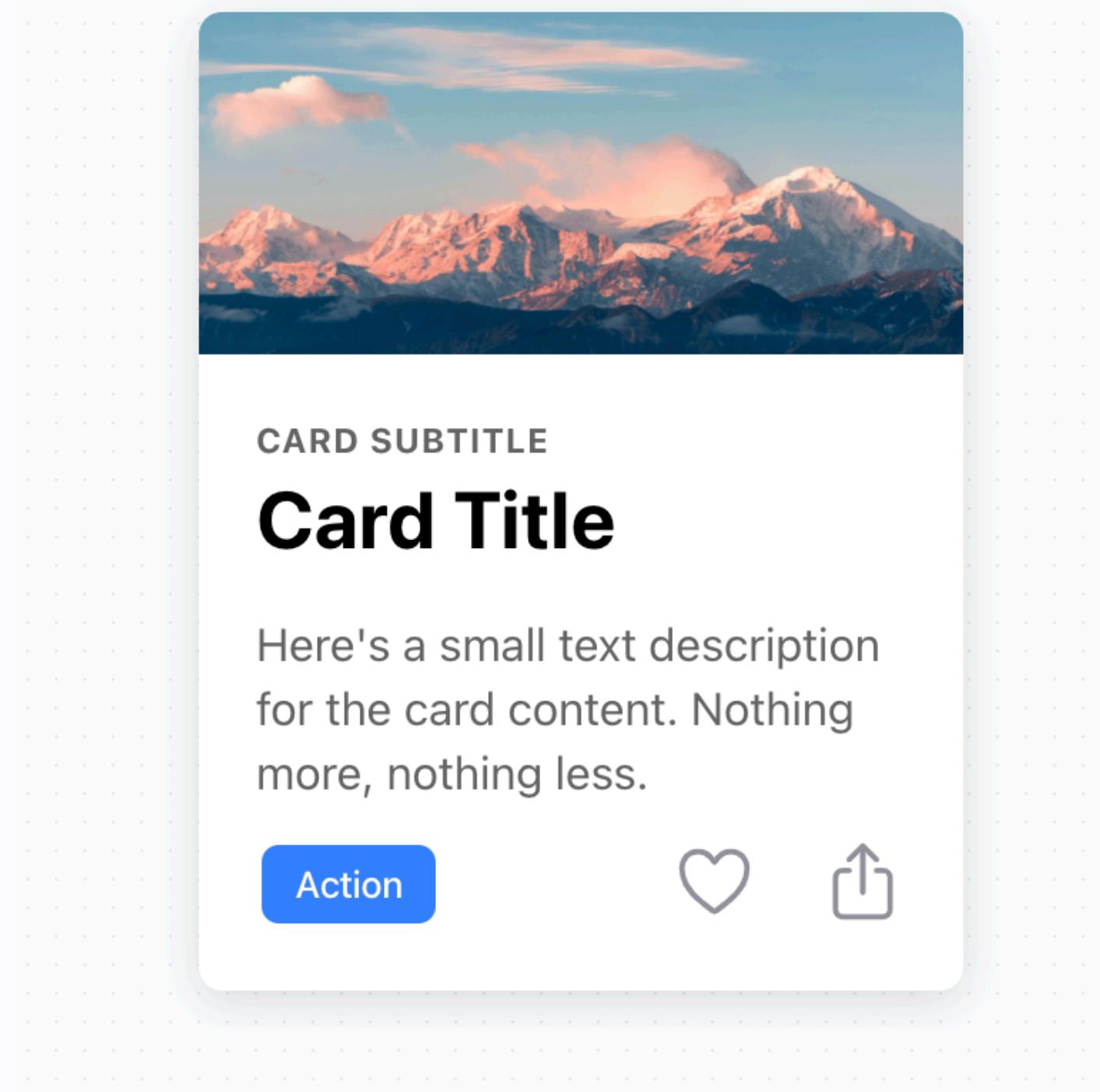
UI Components

The screenshot shows a web browser window displaying the Ionic UI Components documentation at ionicframework.com/docs/components. The page has a dark header with the Ionic logo and navigation links for Guide, Components (which is the active tab), CLI, Native, and an Upgrade Guide. The main content area features a large title "UI Components" and a paragraph explaining that Ionic apps are built from high-level components like cards, lists, and tabs. Below this, there's a grid of cards, each representing a different component:

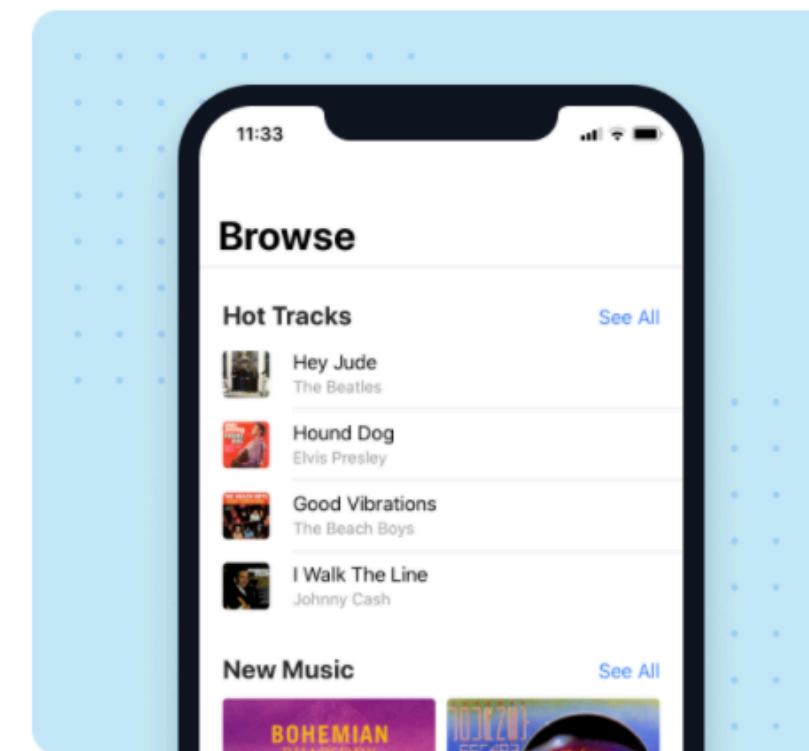
- Action Sheet**: Shows a blue modal with three horizontal buttons.
- Alert**: Shows a circular icon with a list icon and a red badge with the number 2.
- Badge**: Shows a circular icon with a red badge with the number 2.
- Breadcrumb**: Shows a blue card with a list of items.
- Button**: Shows a blue button icon.
- Card**: Shows a blue card icon.
- Checkbox**: Shows a blue checkbox icon.
- Chip**: Shows a blue chip icon.
- Content**: Shows a blue smartphone icon.

The sidebar on the left contains a navigation tree for "Getting Started" under "Components", listing categories like Action Sheet, Accordion, Alert, Badge, Breadcrumb, Button, Card, and Checkbox, each with its corresponding component name (e.g., ion-action-sheet, ion-accordion) listed below it.

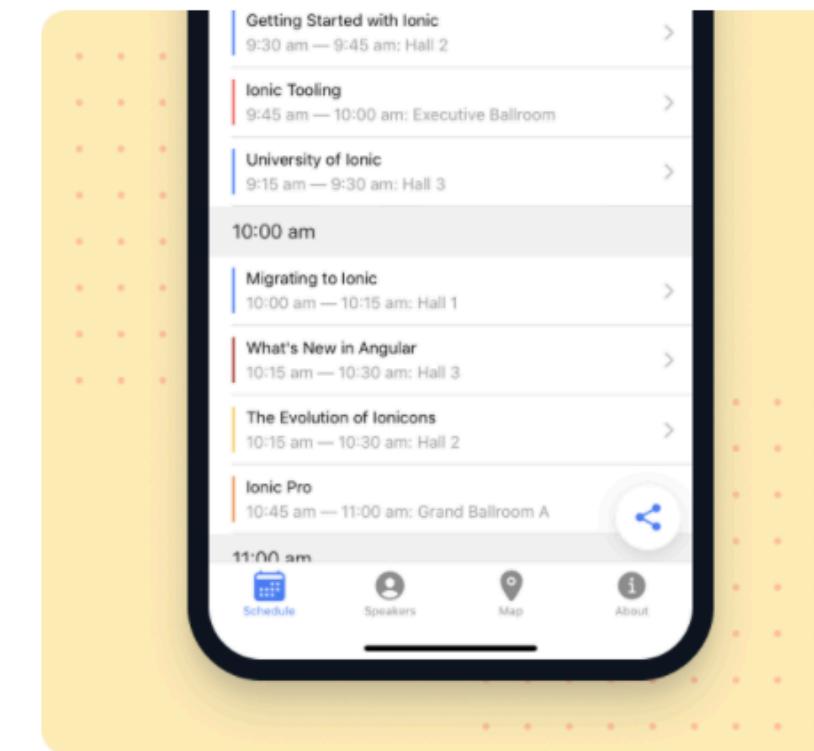
Predefined UI Components



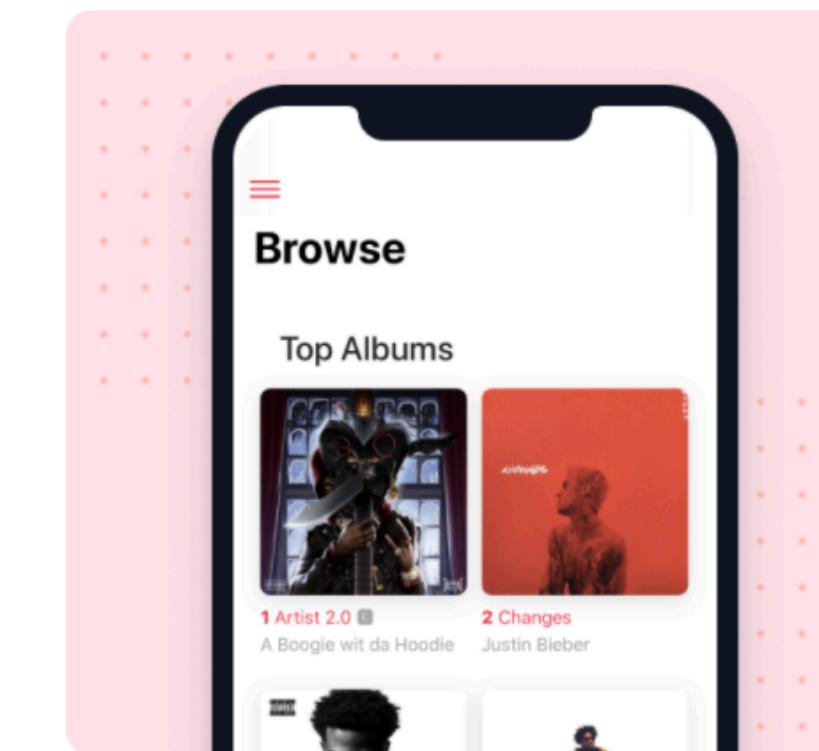
Awesome docs with many examples
and sample apps



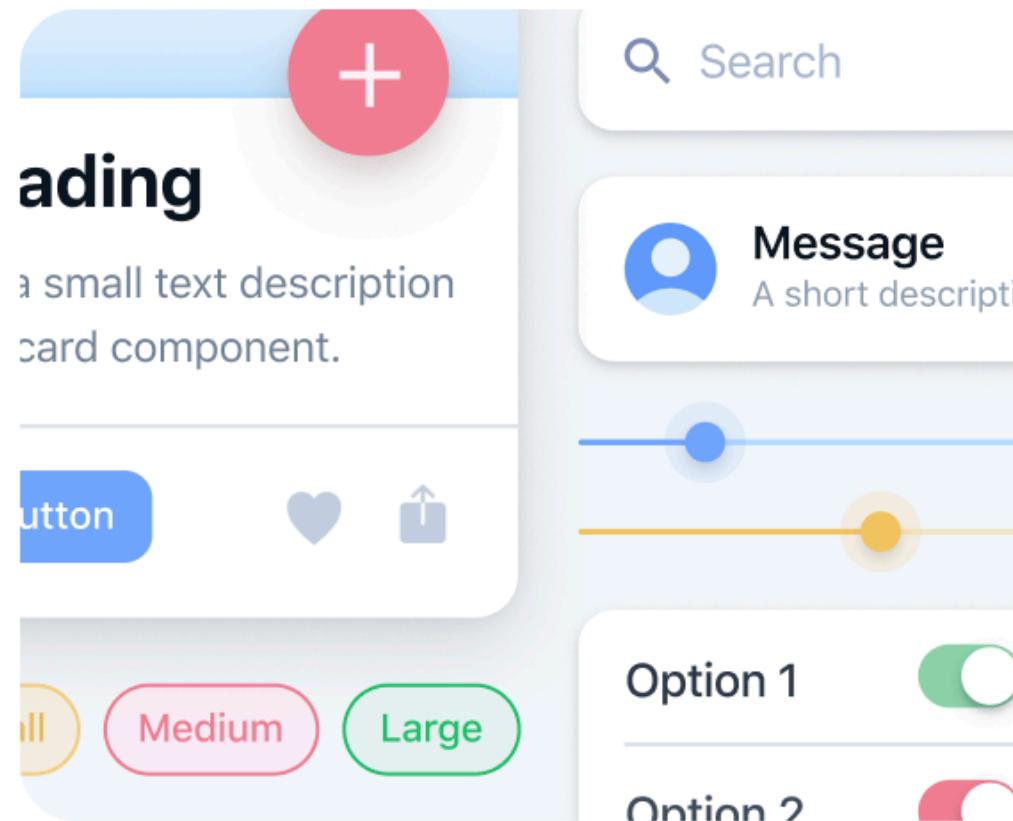
Music Player



Conference App



StarTrack



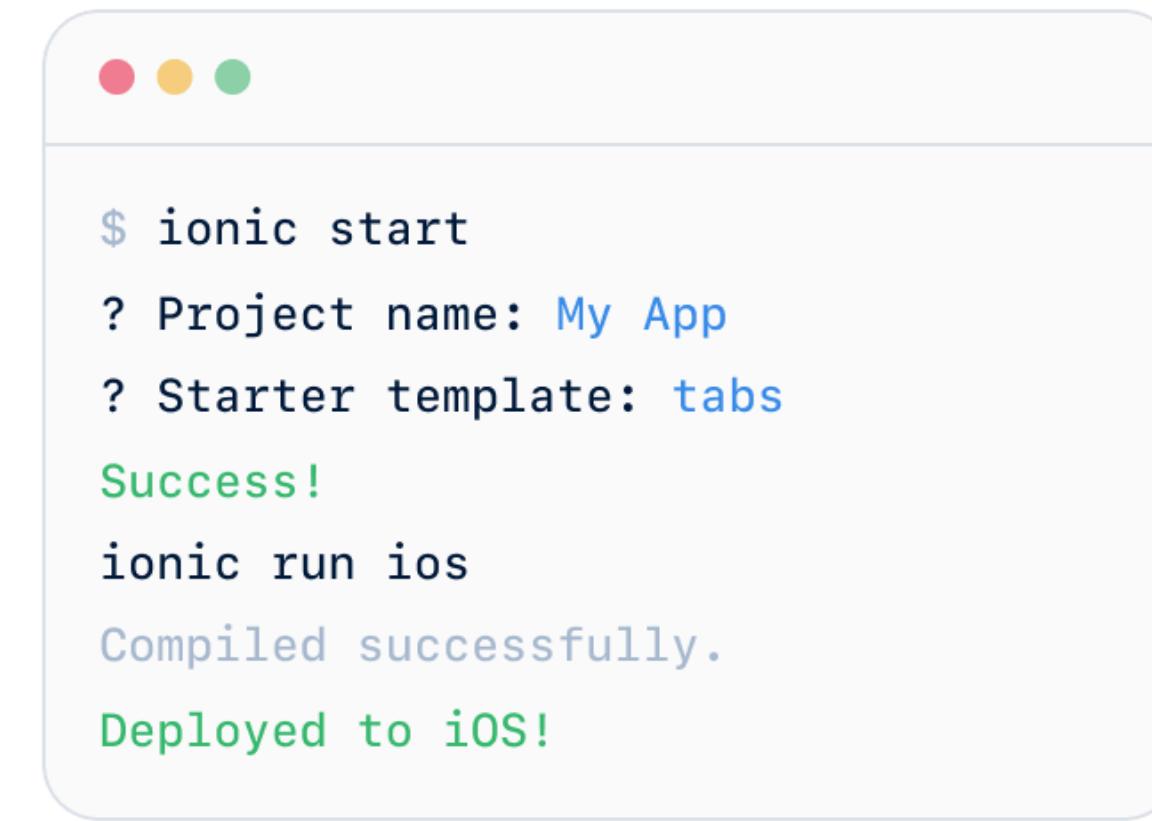
Pre-designed UI components

Ionic's [UI components](#) look great on all mobile devices and platforms. Start with pre-made components, typography, and a base theme that adapts to each platform.



Write once, run anywhere

Ionic lets developers to ship apps to the app stores and as a PWA with a single code base. With [Adaptive Styling](#), apps look and feel at home on every platform.



Developer-friendly tooling

Create, build, test, and deploy your app with the [Ionic CLI](#). Take advantage of Live Reload, deployments, integrations, and even use your favorite JS framework's CLI.

Lifecycle of ion (React) Component

Lifecycle hooks: <https://ionicframework.com/docs/react/lifecycle>

Why lifecycle methods?

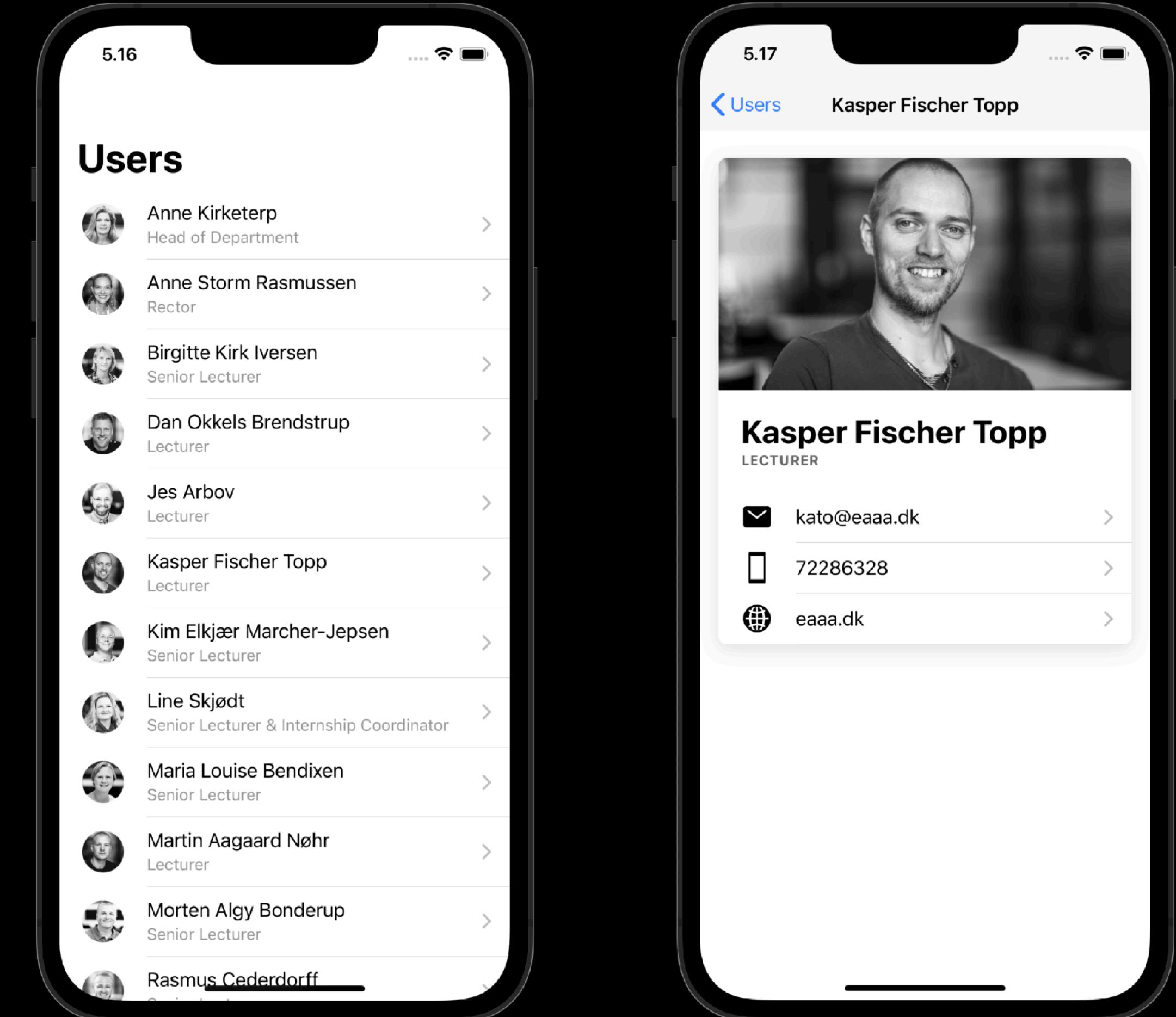
<https://ionicframework.com/docs/react/lifecycle#lifecycle-methods-in-functional-components>

1. Reimplement Users and User component using one of the Ionic React Lifecycle Hooks.
2. Consider the use and differences between `useEffect(...)` and `useIonViewWillEnter(...)`.

Why lifecycle methods?

<https://ionicframework.com/docs/react/lifecycle#lifecycle-methods-in-functional-components>

1. Reimplement Users and User component using one of the Ionic React Lifecycle Hooks.
2. Consider the use and differences between `useEffect(...)` and `useIonViewWillEnter(...)`.



```
<IonApp>
  <IonReactRouter>
    <IonTabs>
      <IonRouterOutlet>
        <Route exact path="/posts">
          <PostsPage />
        </Route>
        <Route exact path="/add">
          <AddPage />
        </Route>
        <Route path="/users/:id">
          <UserPage />
        </Route>
        <Route exact path="/">
          <Redirect to="/posts" />
        </Route>
      </IonRouterOutlet>
      <IonTabBar slot="bottom">
        <IonTabButton tab="Posts" href="/posts">
          <IonIcon icon={reader} />
          <IonLabel>Posts</IonLabel>
        </IonTabButton>
        <IonTabButton tab="add" href="/add">
          <IonIcon icon={add} />
          <IonLabel>Add</IonLabel>
        </IonTabButton>
      </IonTabBar>
    </IonTabs>
  </IonReactRouter>
</IonApp>
```

Navigation & Routing

<https://ionicframework.com/docs/react/navigation>

ion-router

<https://ionicframework.com/docs/api/router>

ion-nav

<https://ionicframework.com/docs/api/nav>

Add New Page

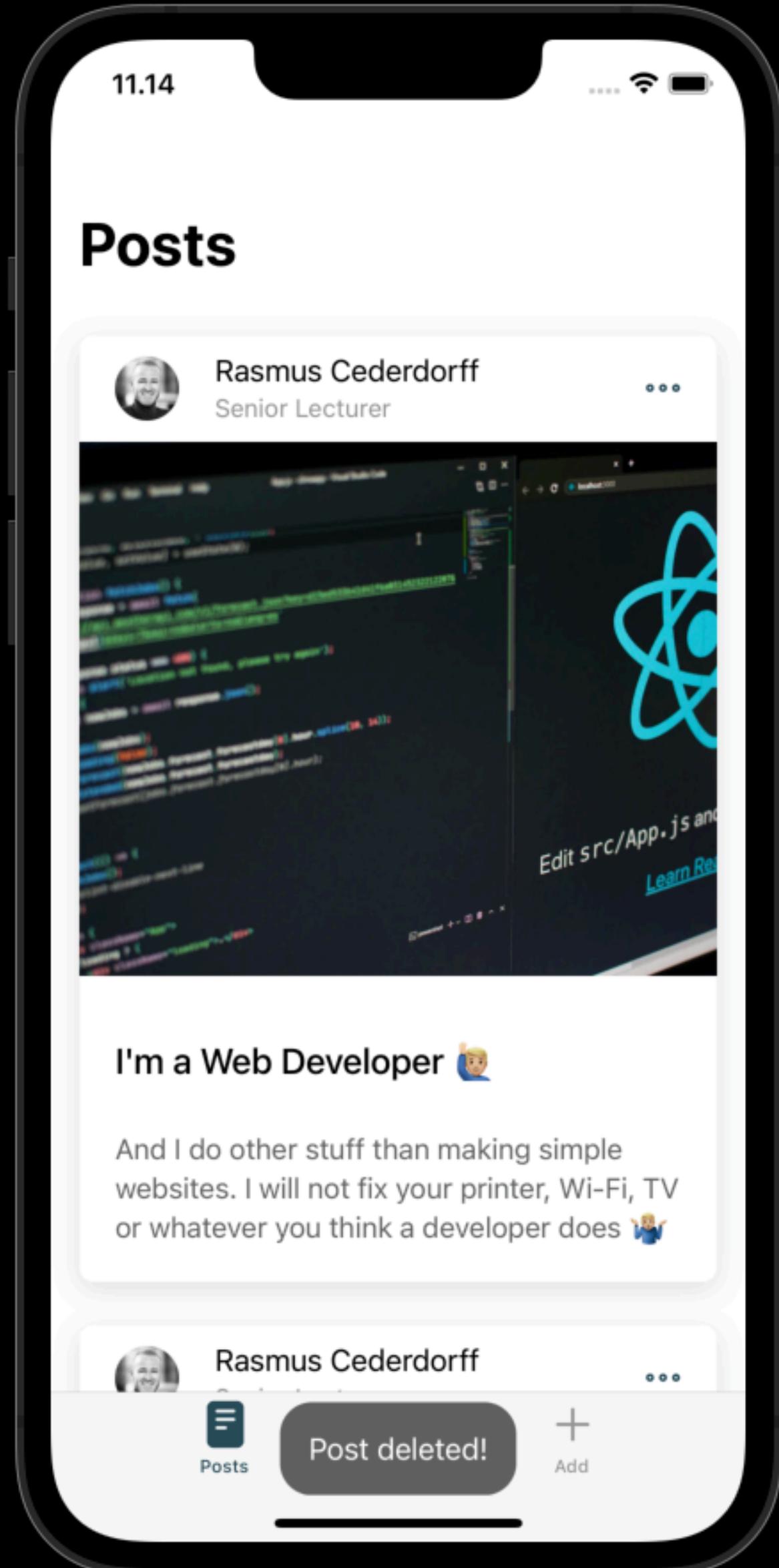
1. Add a new function component (a new page).
2. Add a new route in App.tsx to handle routing of your new page.
3. Test and make sure the navigation works as expected.
4. Next, change the redirect so your new page is the default page of your app.

Types of Mobile Navigation

What kind of types do you know?

Add a modal

<https://ionicframework.com/docs/api/modal#usage>

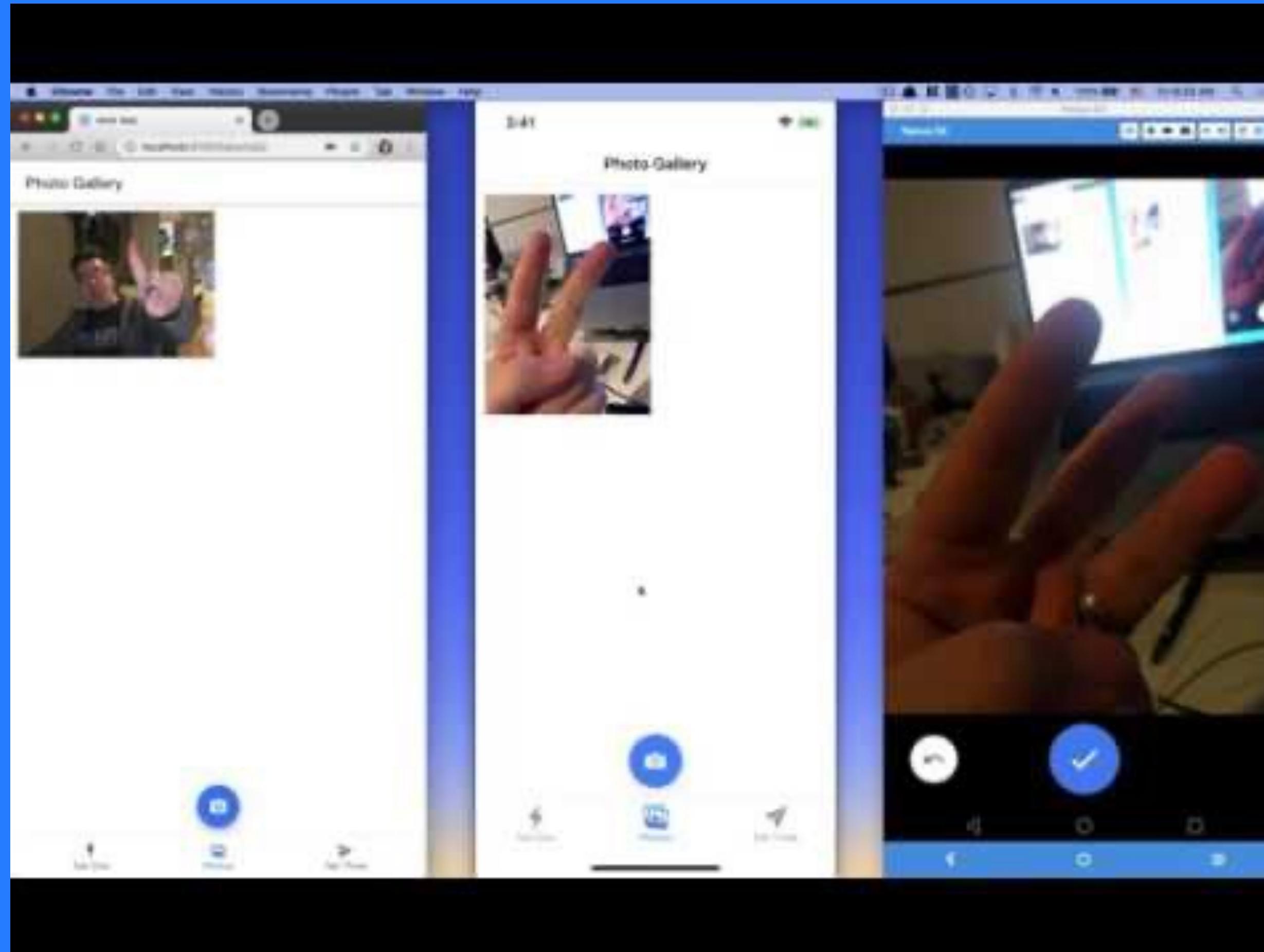


Ionic Post App w/ Firebase REST

Step 1 - 5

Ionic Camera App

<https://ionicframework.com/docs/react/your-first-app#create-an-app>



Next Tuesday

The screenshot shows a Notion page with a dark blue header and a white content area. The title '3. Native Device Features' is at the top. Below it is a section titled 'Themes' with a bulleted list of topics. To the left of the themes is a sidebar with various metadata fields.

3. Native Device Features

Lecturer: RACE
Date: 06/09/2022
Notes: Ionic Project Kick Off
Course: MAD

Themes

- Build, emulate & compile for the Web, Android & iOS
- Capacitor Setup
- Native Device Features
- Using the Native APIs
- Android Studio, Xcode & Livereload
- Ionic CLI & Tools