



Universal Apps With Expo

Writing truly universal code
and deploying it everywhere

Poland / 2019

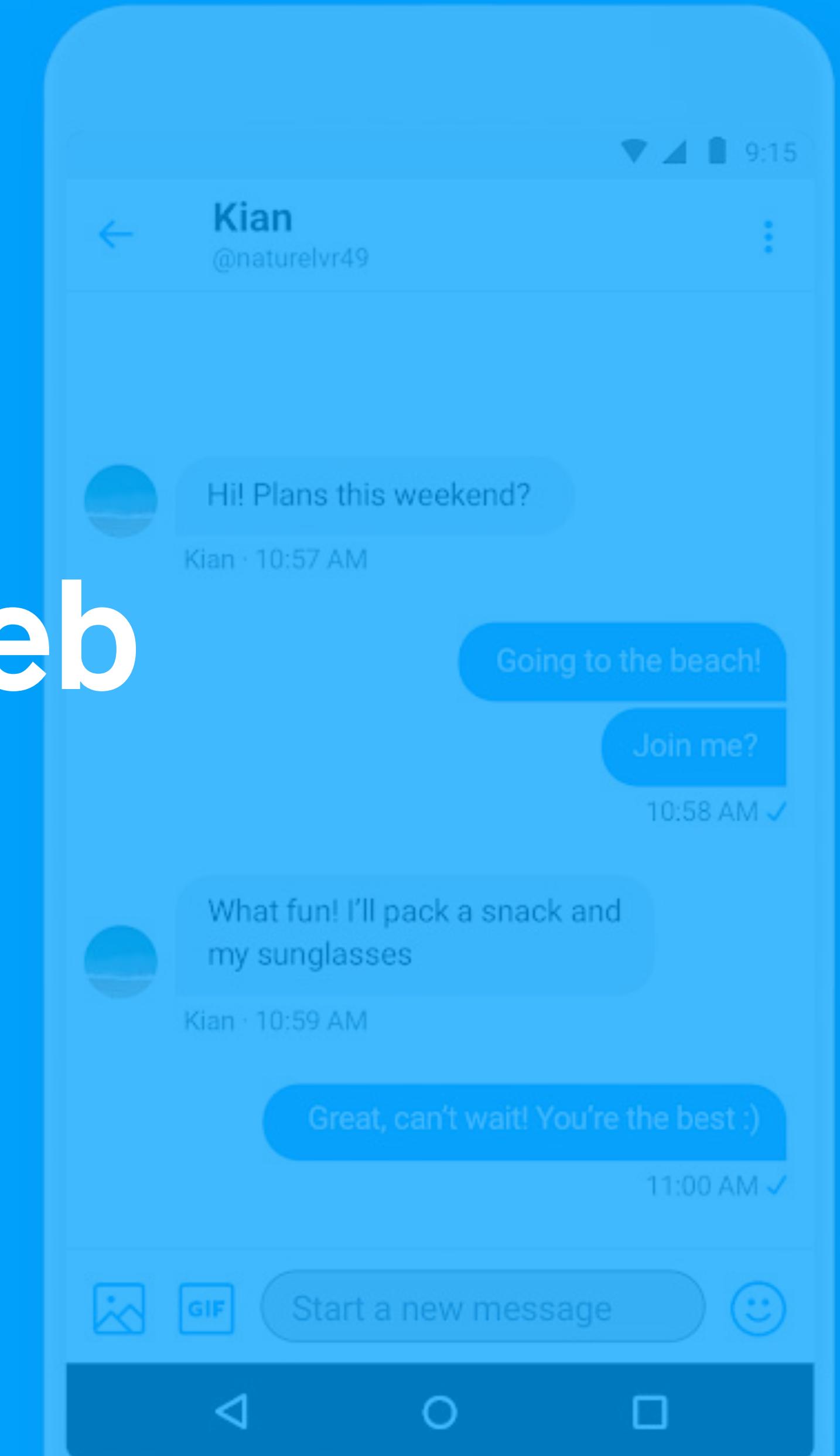
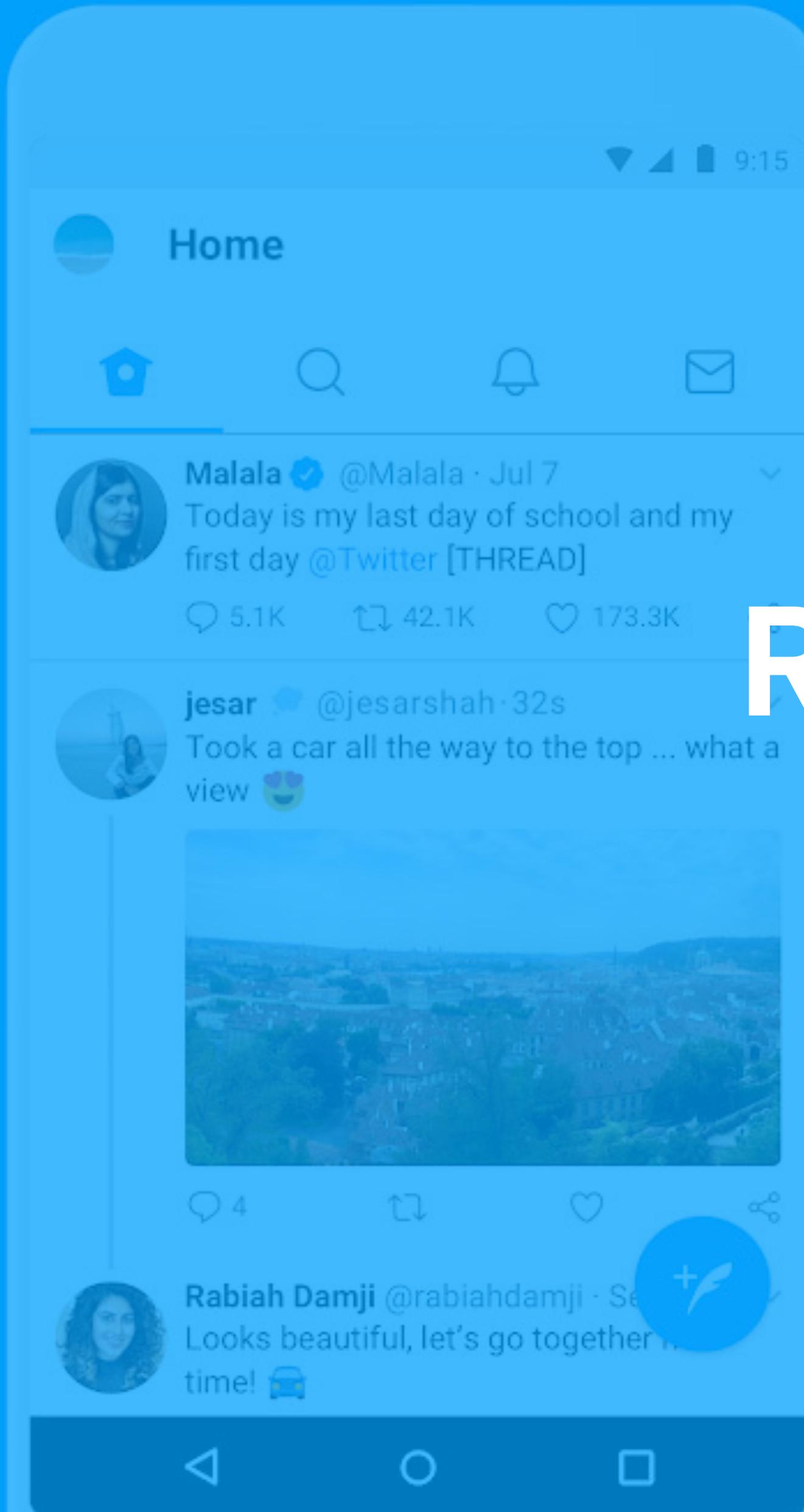


Expo

Universal Apps



React Native for Web



Used By

- Major League Soccer
- Flipkart
- Uber
- The Times
- DataCamp

React Native Web

Nicolas Gallagher

- React Rally '17
- Styles
- Accessibility



React Native Web

The Catch

React Native Web

- Not enough libraries
- Tree shaking was too fragile
- React Native Bad Practices
- Native / API conflicts

React Native Web

Fonts

React Native Web

CSS

```
/* src/index.css */
@font-face {
  font-family: 'MyFont-Book';
  font-style: normal;
  font-weight: 400;
  src: url('./MyFont-Book.ttf');
}
```

Config

```
// webpack.config.js
{
  test: /\.ttf$/,
  loader: "url-loader",
  include: path.resolve(__dirname, "src")
}
```



Closed

Custom (web) fonts?



@necolas would you accept a PR to add an expo-like API to load custom fonts?
<https://docs.expo.io/versions/latest/sdk/font.html>



Custom (web) fonts?



@necolas would you accept a PR to add an expo-like API to load custom fonts?
<https://docs.expo.io/versions/latest/sdk/font.html>



necolas commented on Jan 2, 2018

Owner

No, that belongs in a tool like Expo. Thanks for asking :)



necolas commented on Jan 2, 2018

Owner



...

"Expo for Web" is an idea that's been kicking around. And making a combination of CRA and CRNA.



8



necolas comme

"Expo for Web"

Expo for Web

First Look

Expo for Web

- Unimodules
- Babel Preset
- Webpack Config
- CLI

Expo for Web

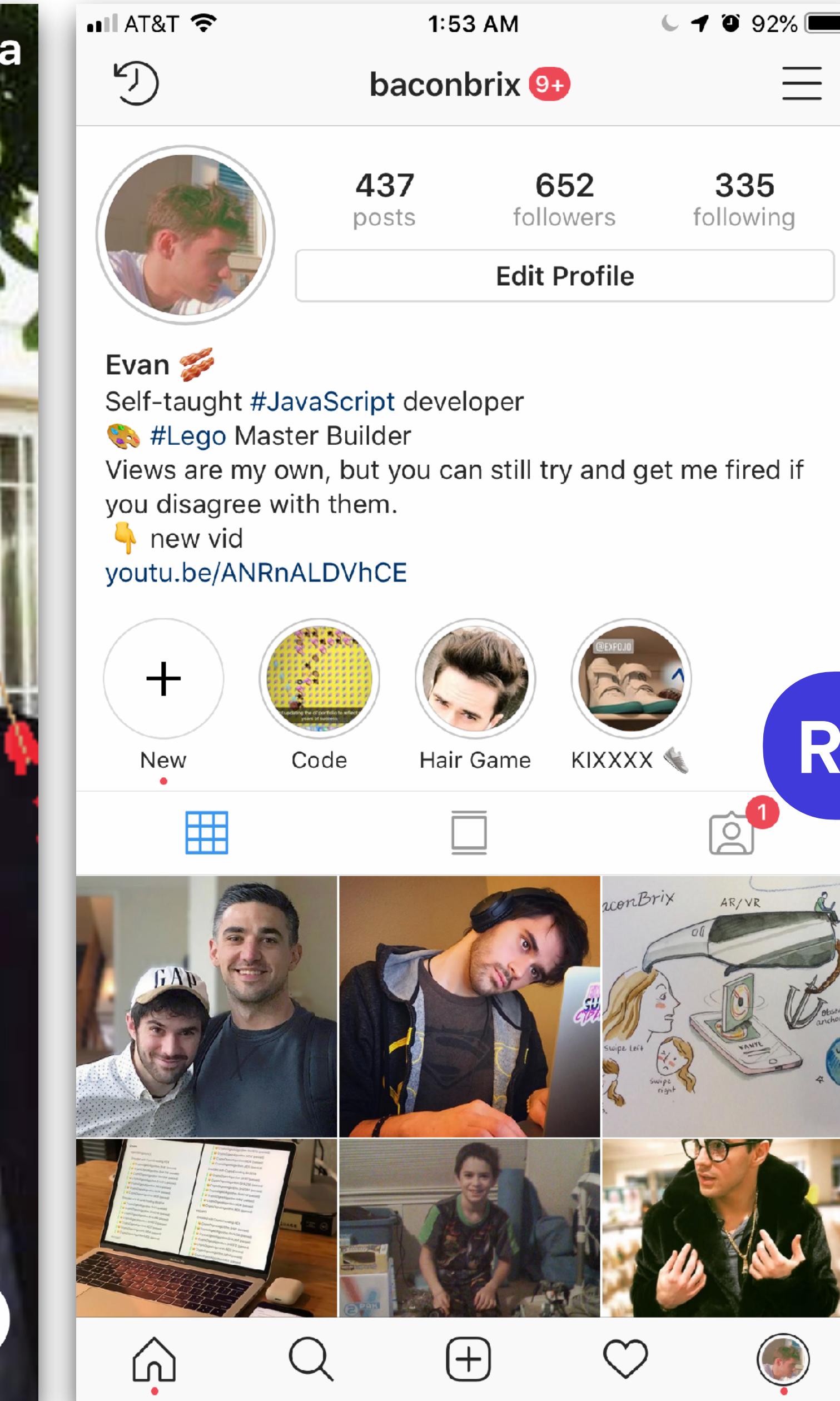
Unimodules



Unimodules



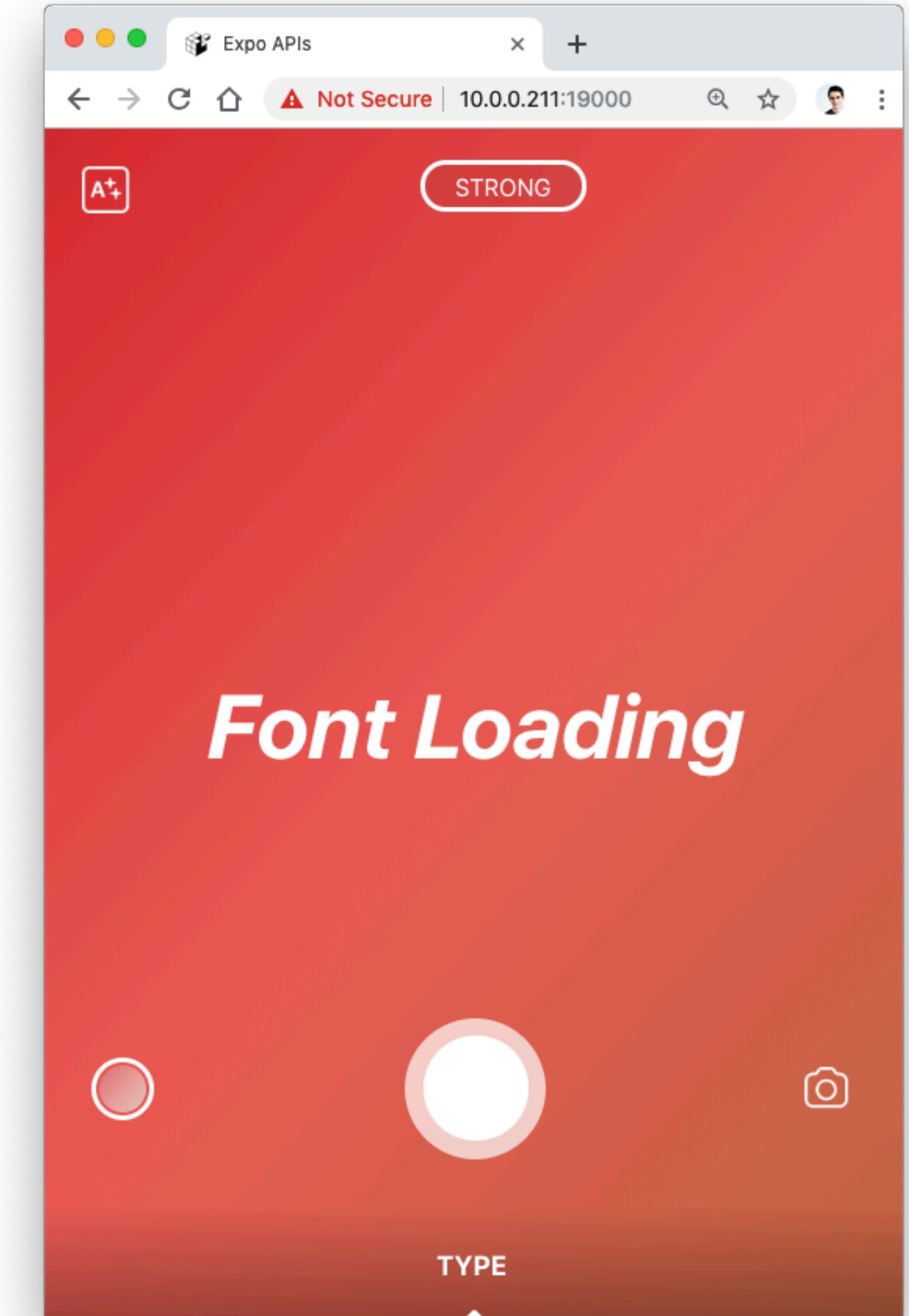
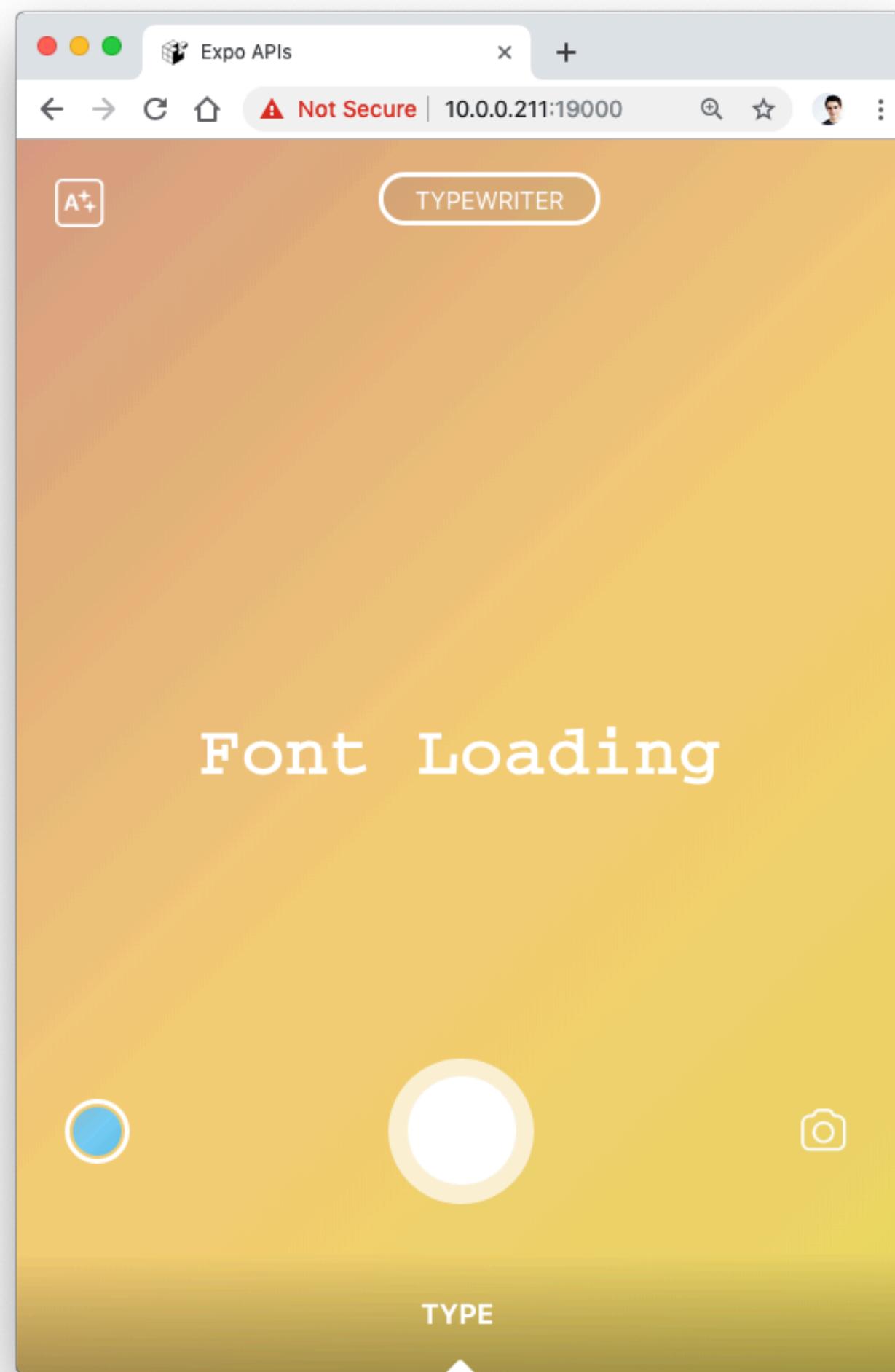
Expo



React Native Web

- Direct Abstractions
- Emulated
- Synthetic
- Unsupported

Font Loading



Font Loading

```
import { Font } from 'expo'

Font.loadAsync({
  'SF-Book': require('./SF-Book.ttf')
})
```

Expo for Web

- SVG
- SQLite
- Printing
- GL View
- Gradients

Expo for Web

Emulated Functionality

Image Picker

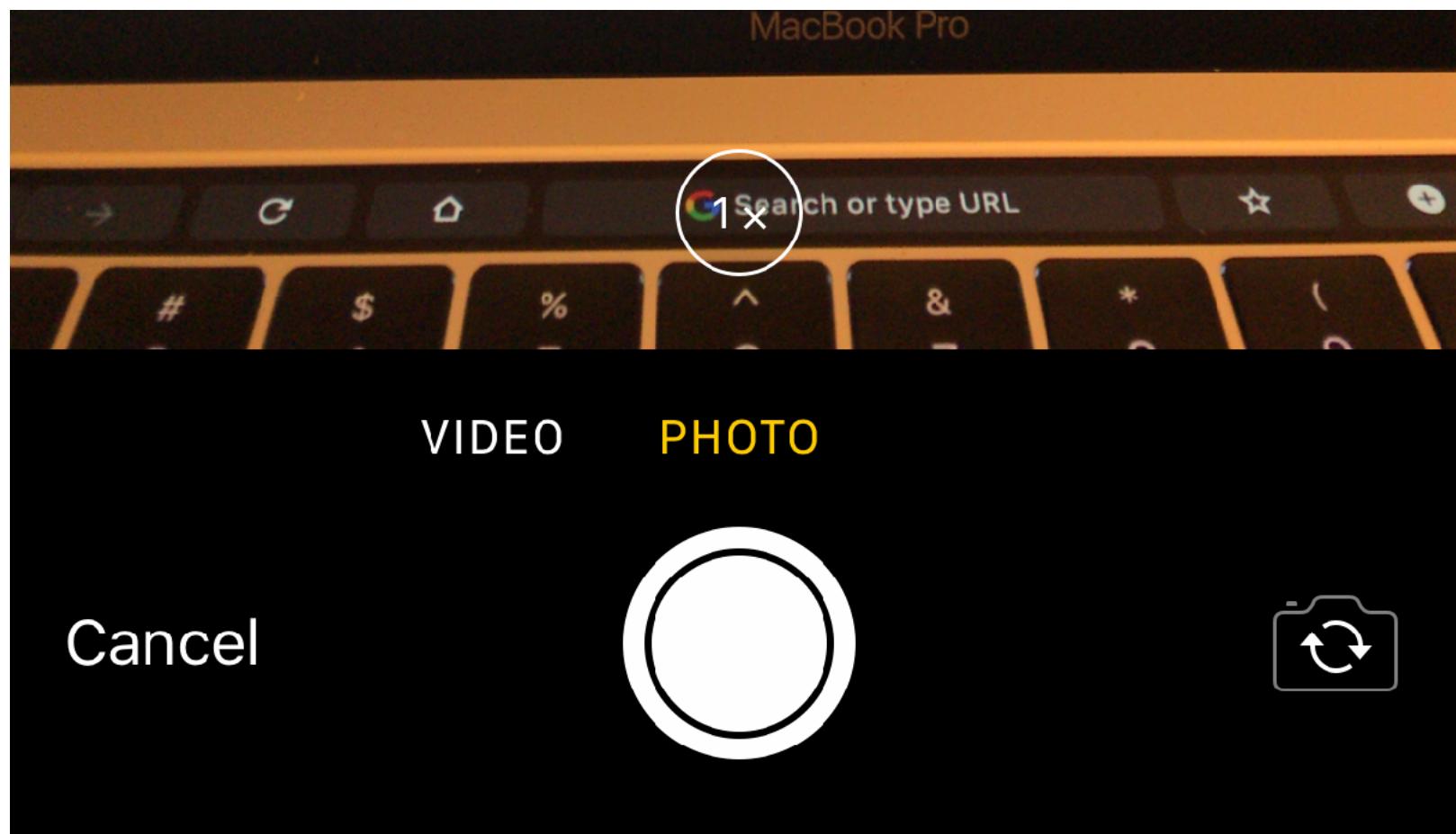
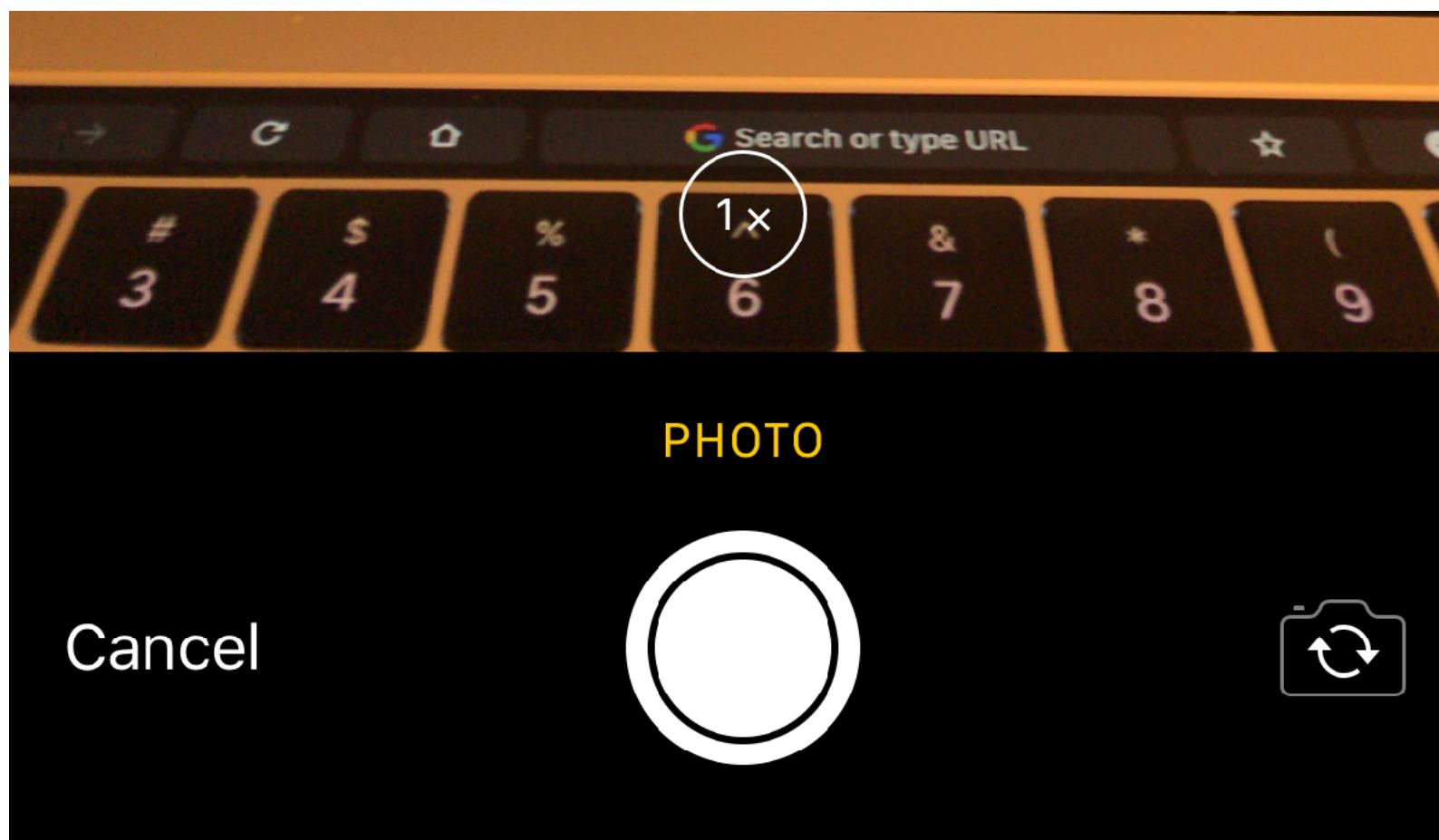
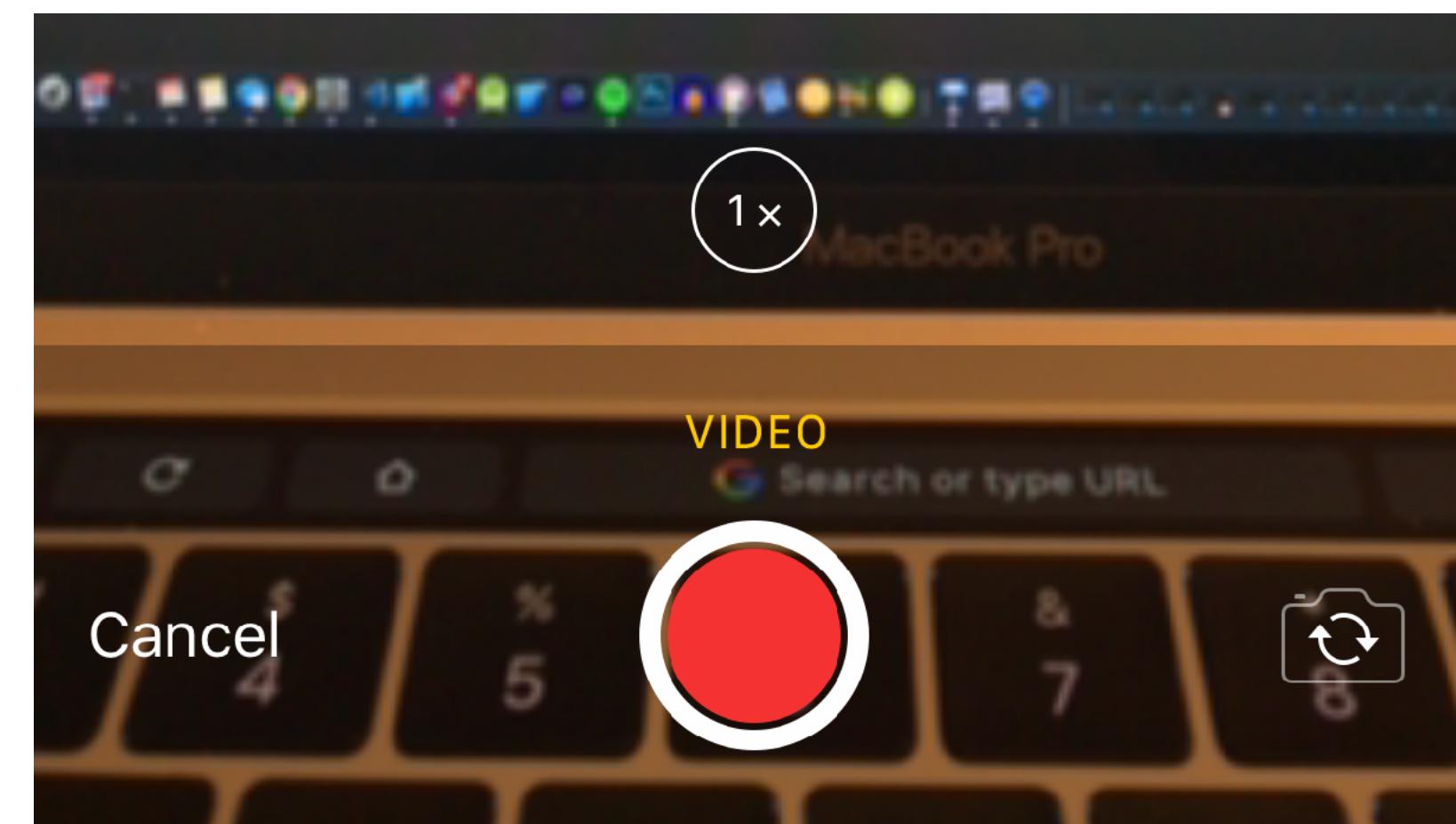


Photo & Video



Photo



Video

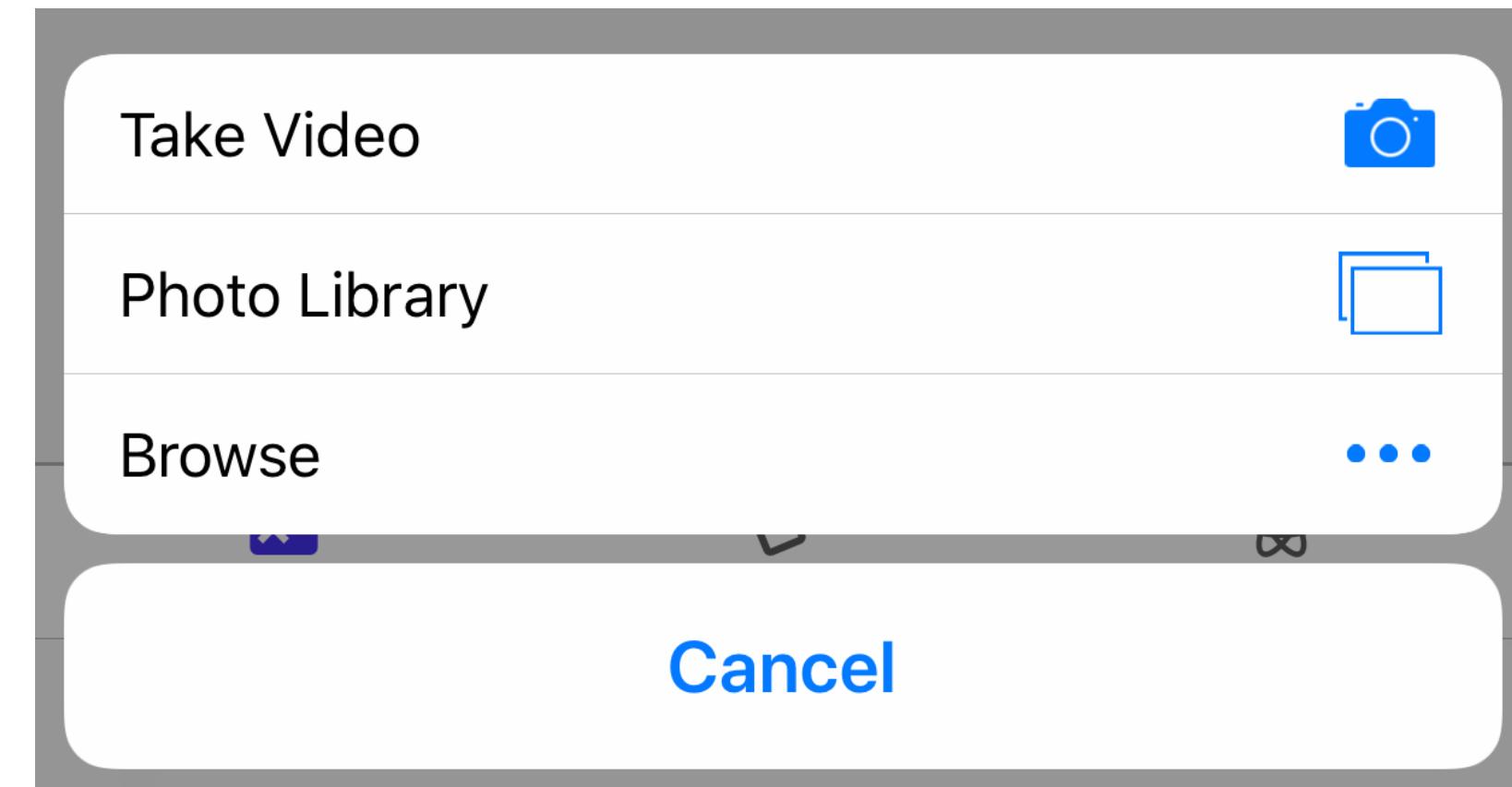
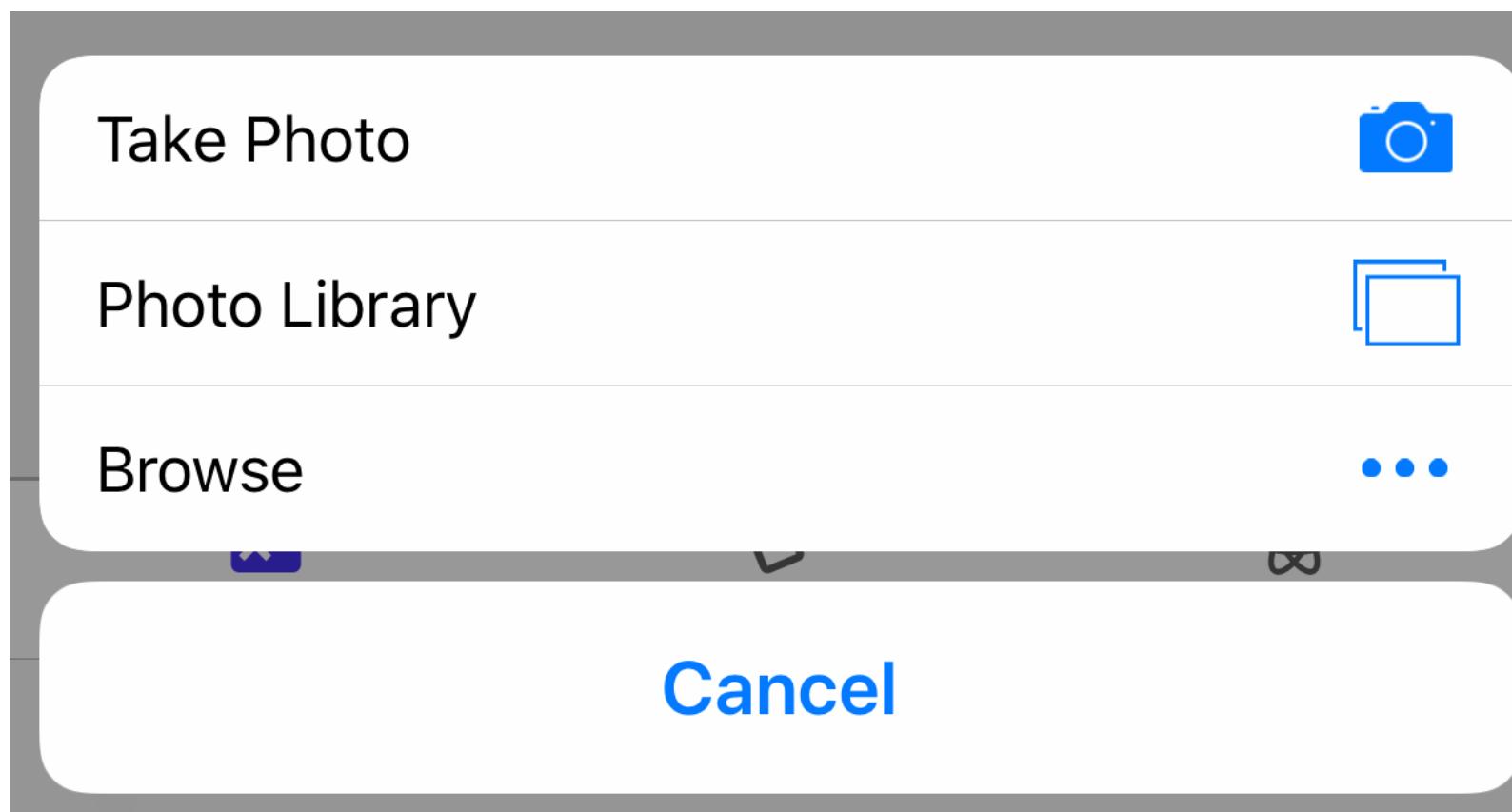
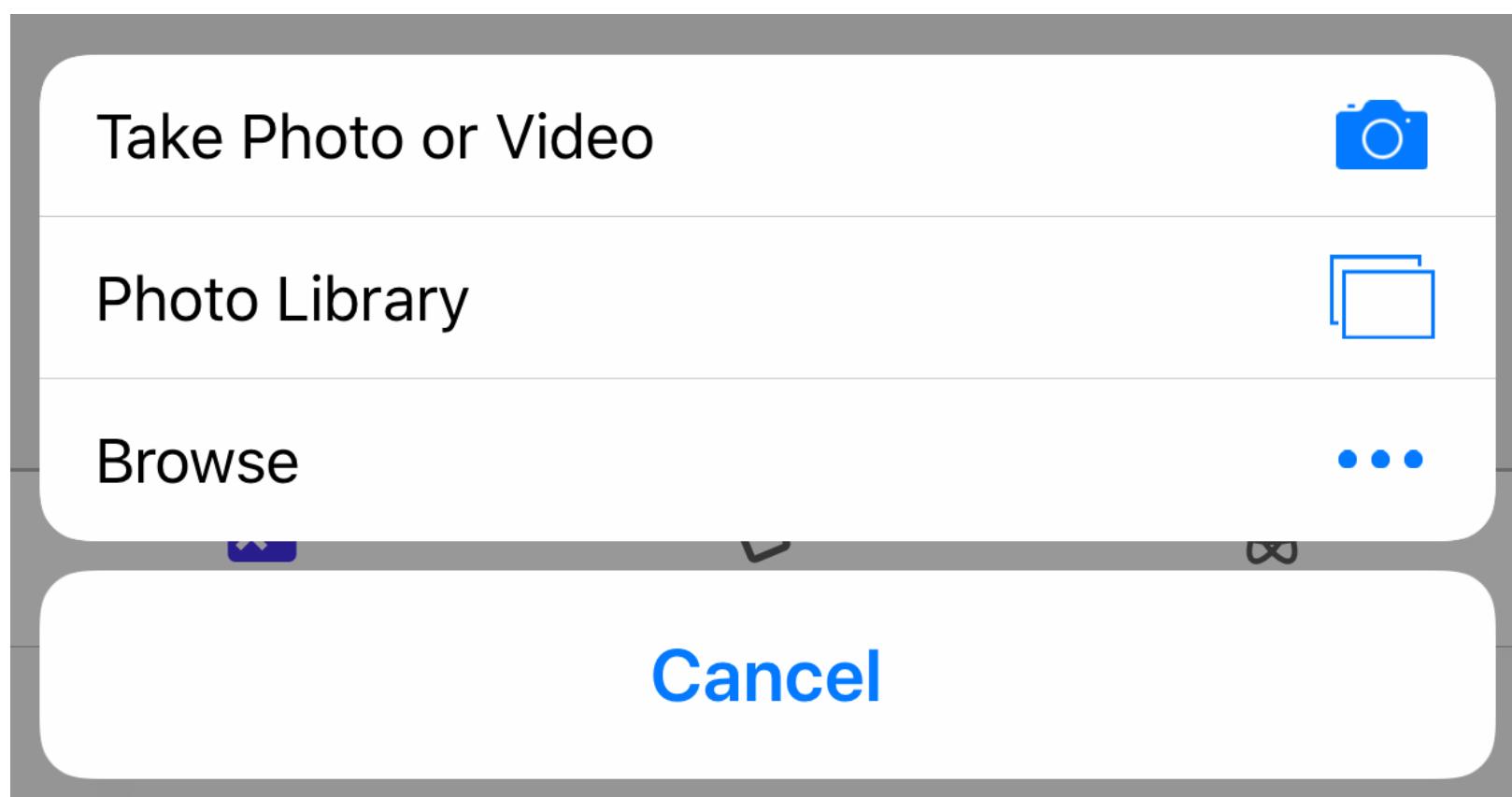


Image Picker

```
<body>
+  <input accept="" multiple="" />
// After interaction
-  <input accept="" multiple="" />
</body>
```

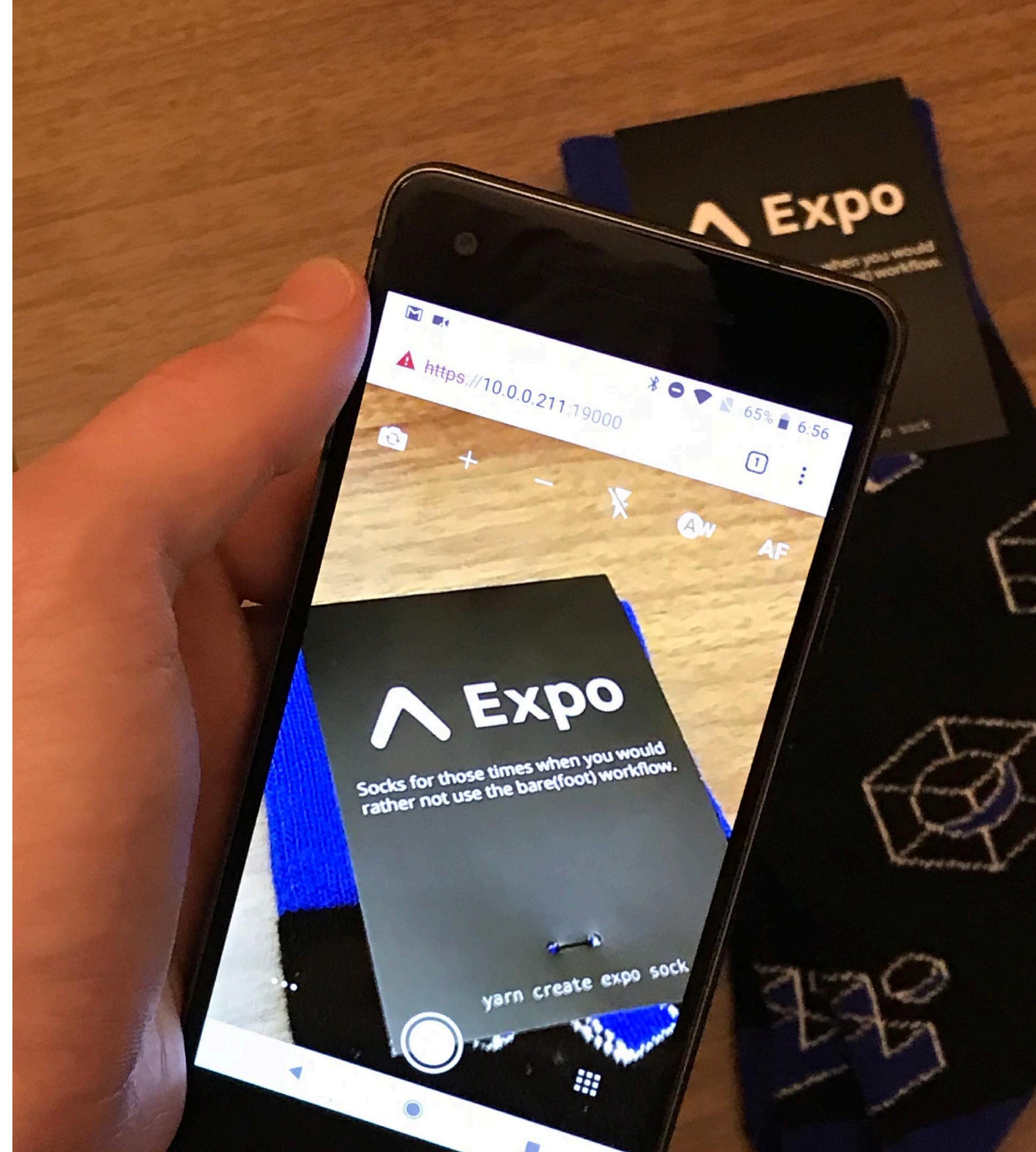
Image Picker

```
<Button accessibilityLabel="Pick an Image" />
```

Camera

Mobile
Camera Flipping
Picture Size

Android
White Balance
Torch
Auto Focus
Zoom



Expo for Web

Synthetic Functionality

View Shot API

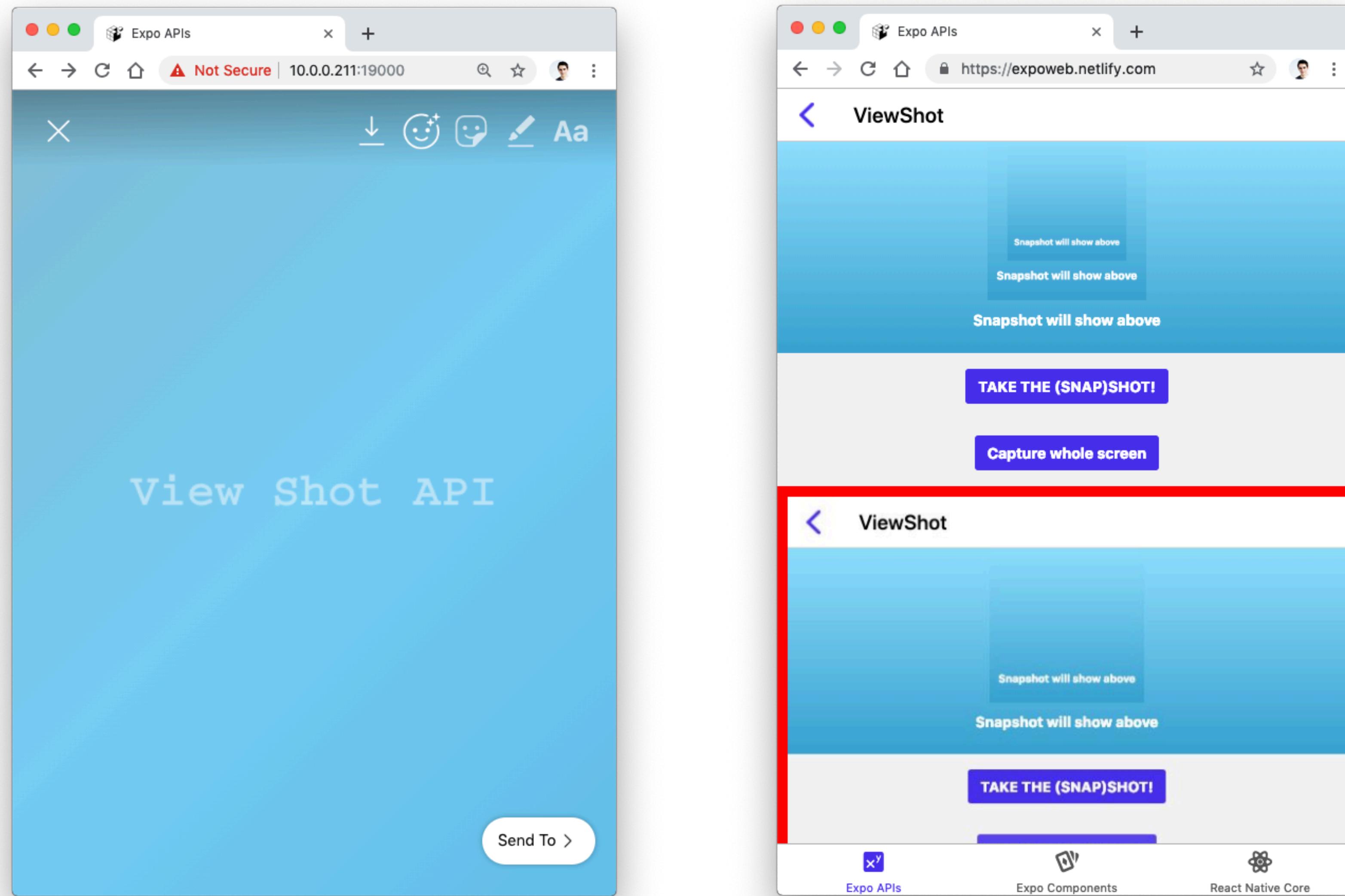
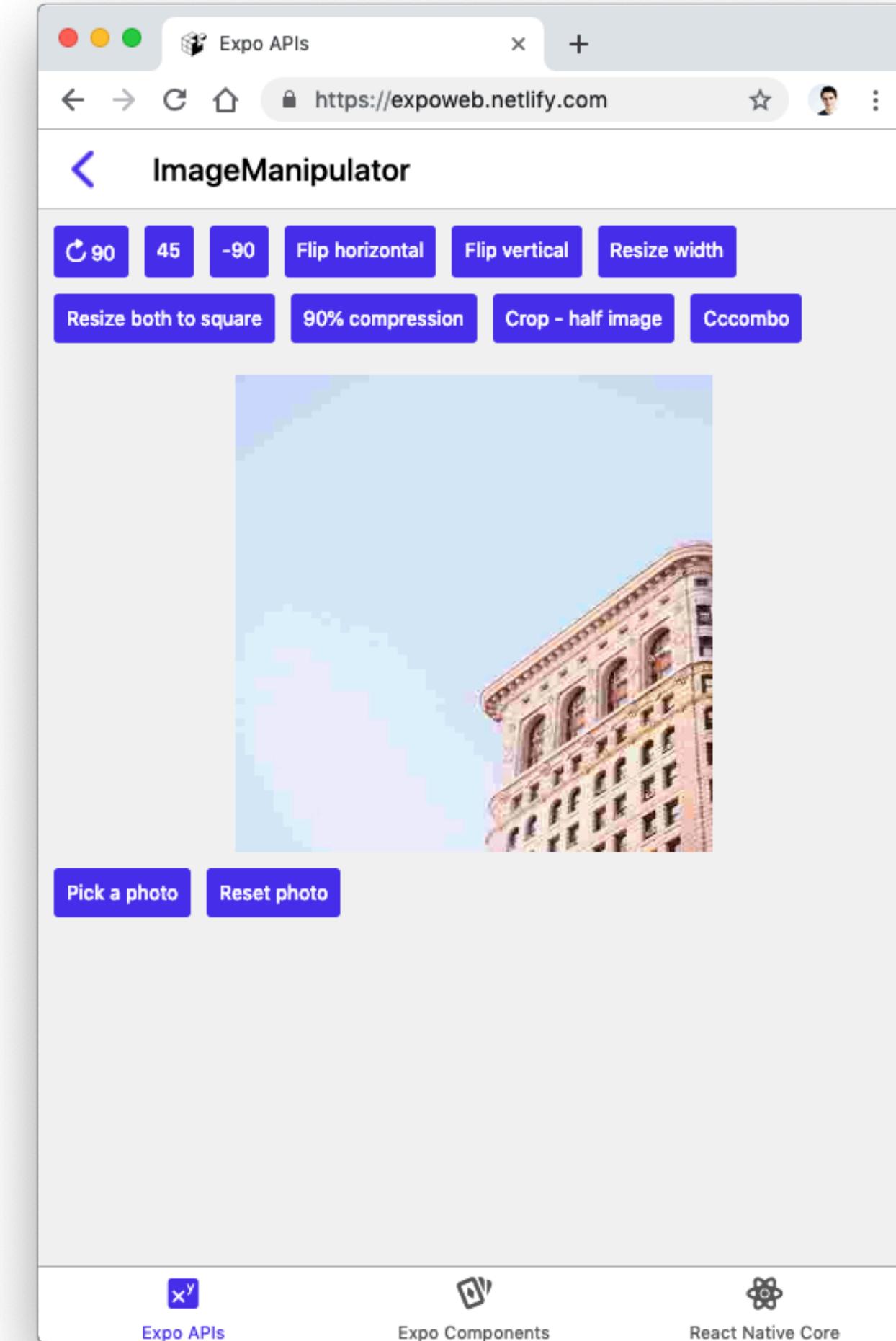
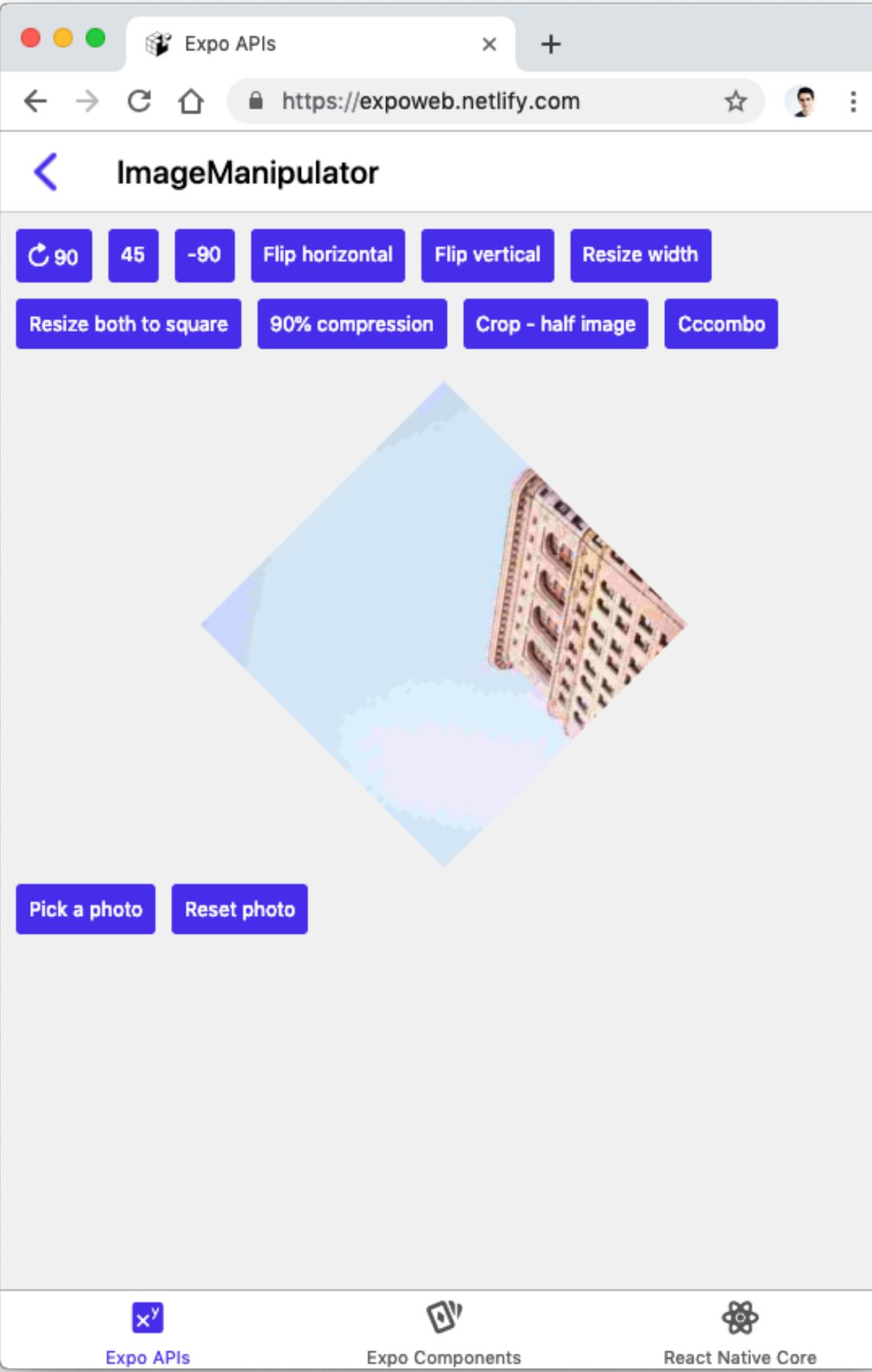
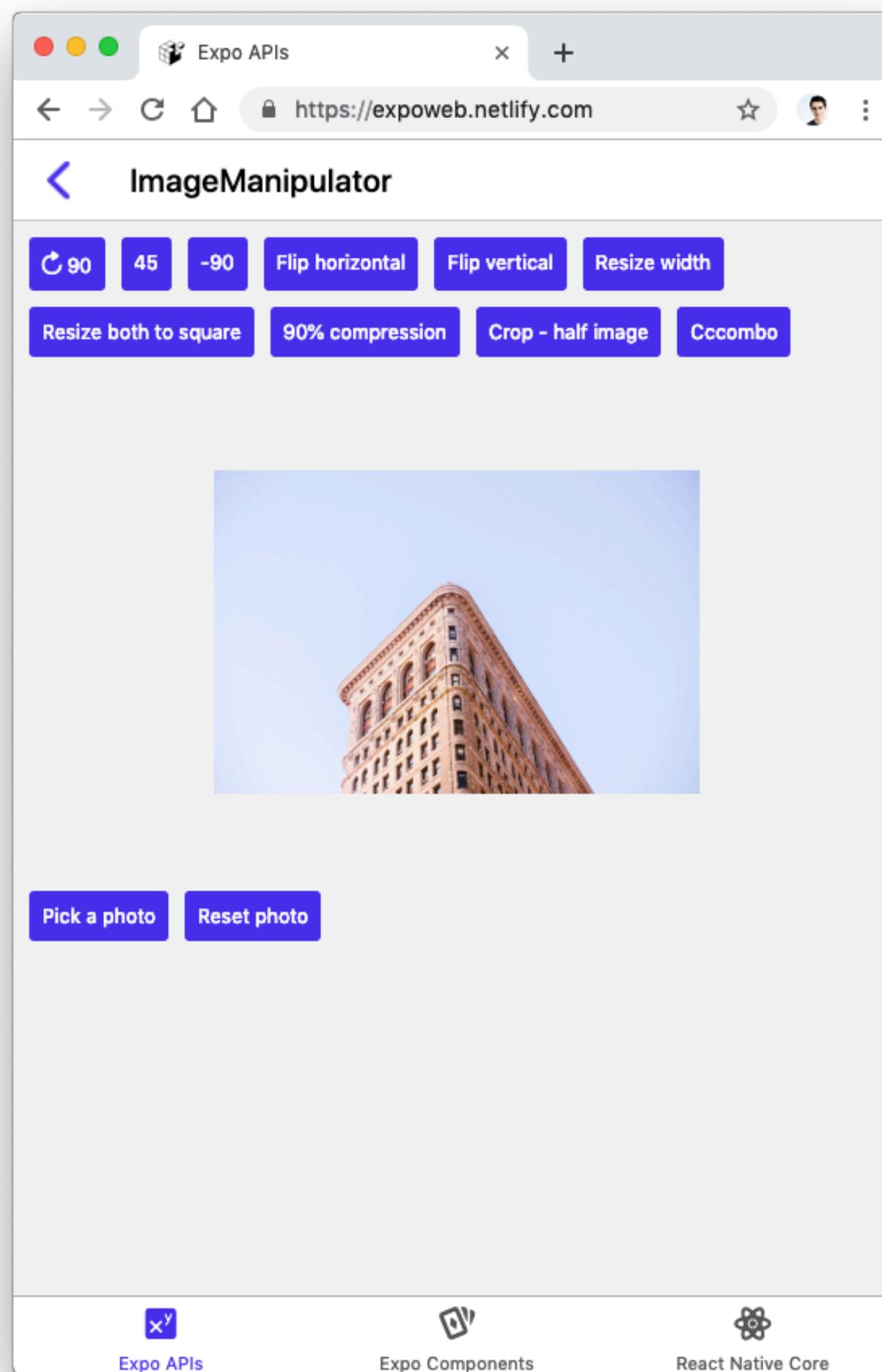
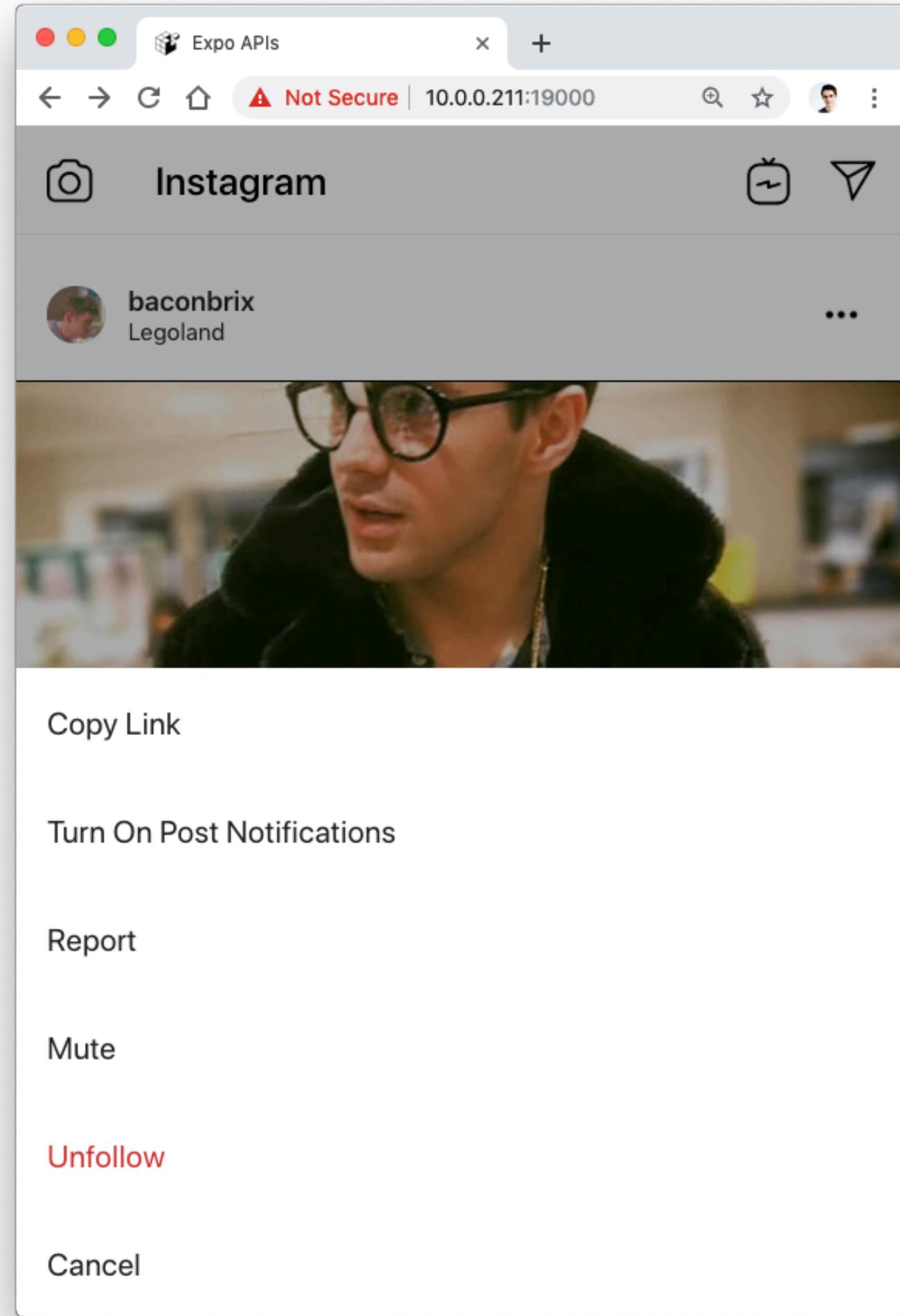


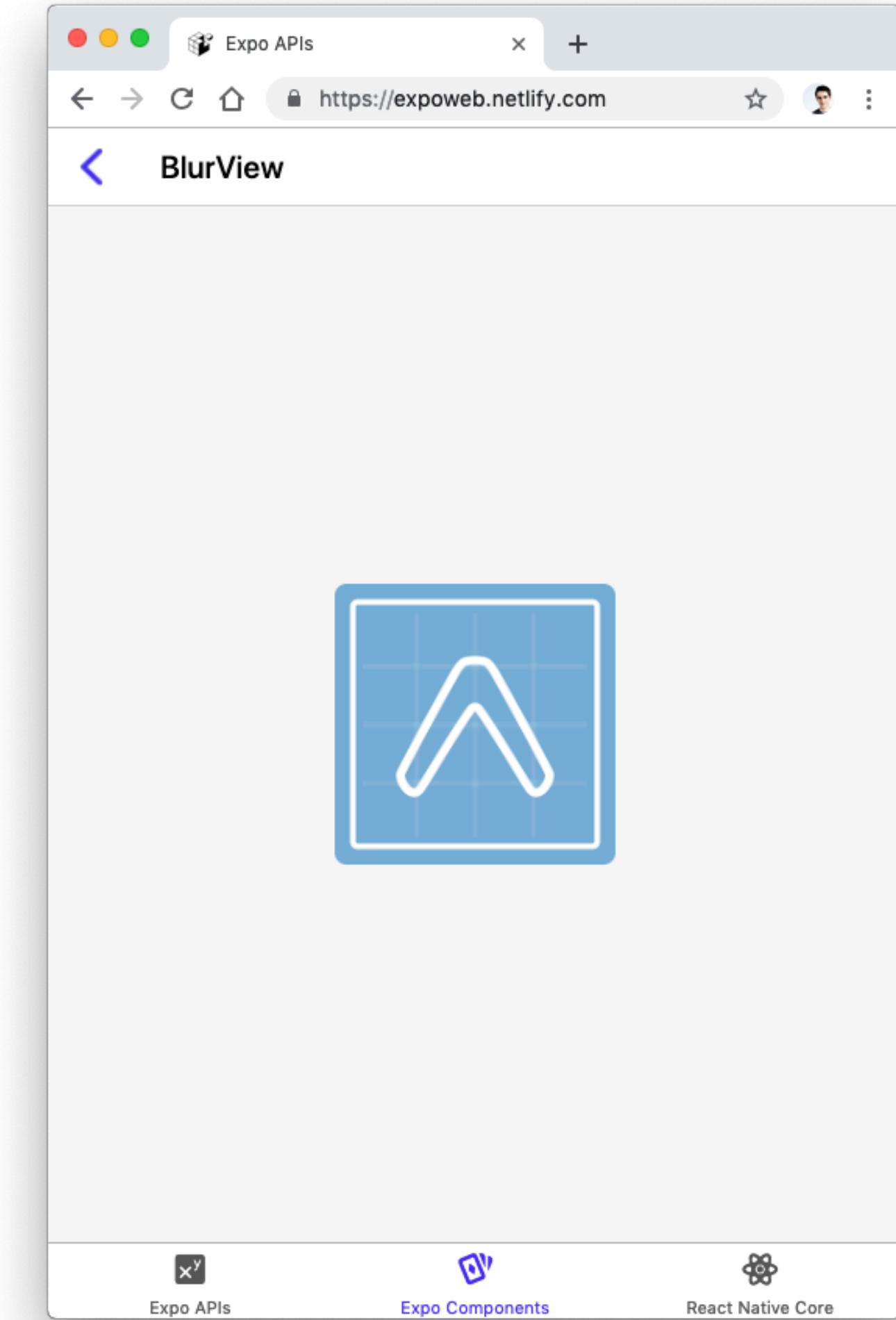
Image Manipulation



Fallback



Action Sheet



Blur View

Unsupported Features

Expo for Web

- Calendar
- Contacts
- Media Library
- Task Manager

Expo for Web

```
// 😁  
await Feature.isAvailableAsync()  
  
// 😁  
const { status } await Permissions.getAsync(/*...*/);  
// status === 'unavailable'  
  
// 😊  
Feature.doThing();  
// throws UnavailabilityError  
// Feature.doThing() is not supported on web
```

Expo for Web

Babel Preset

Babel Preset Expo

metro:react-native-babel-preset
babel-plugin-react-native-web

Babel Preset Expo



Webpack



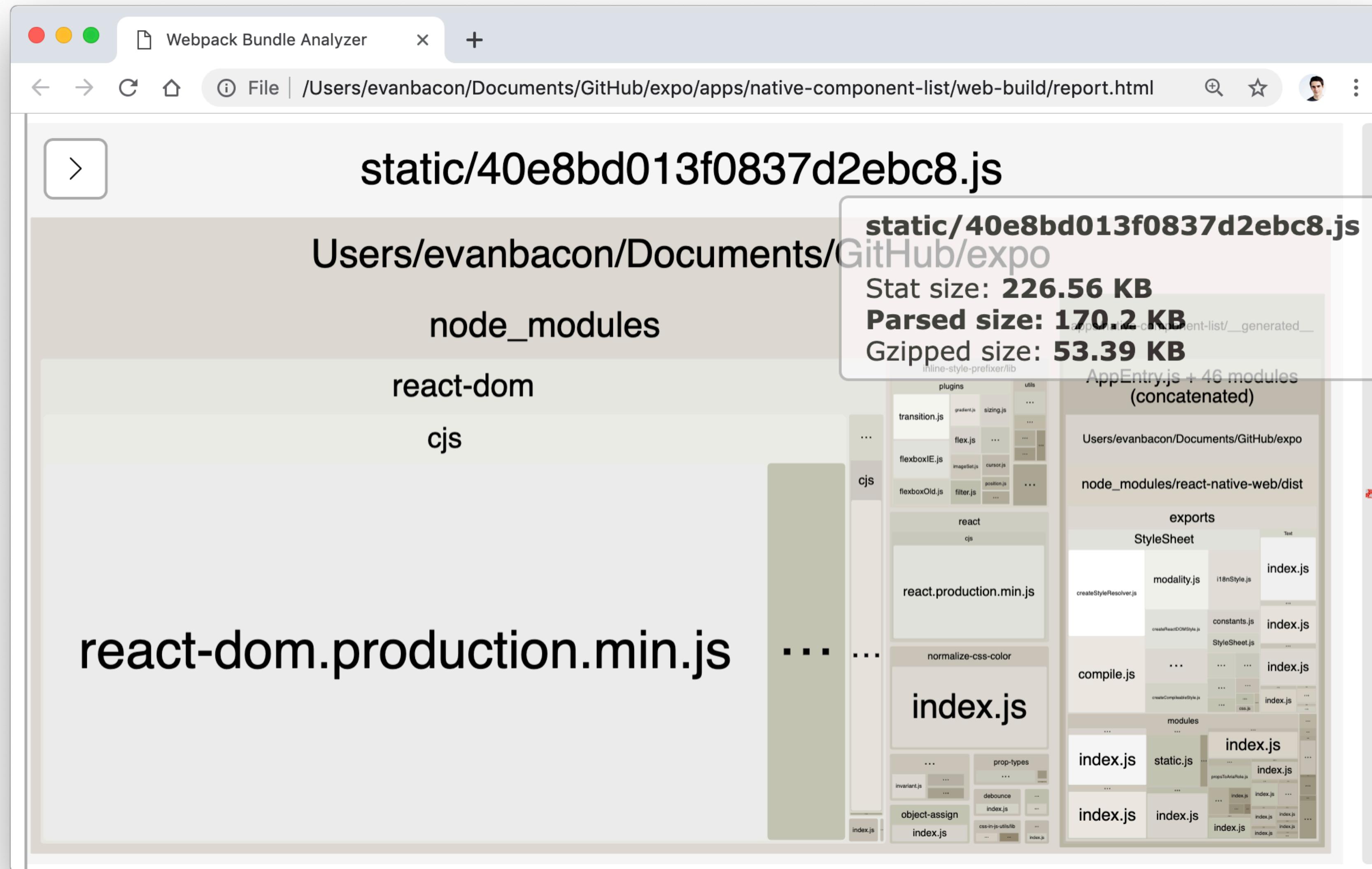
Metro

Tree Shaking

```
import { Text } from 'react-native';

export default () => <Text>😊 notice me senpai</Text>
```

Tree Shaking



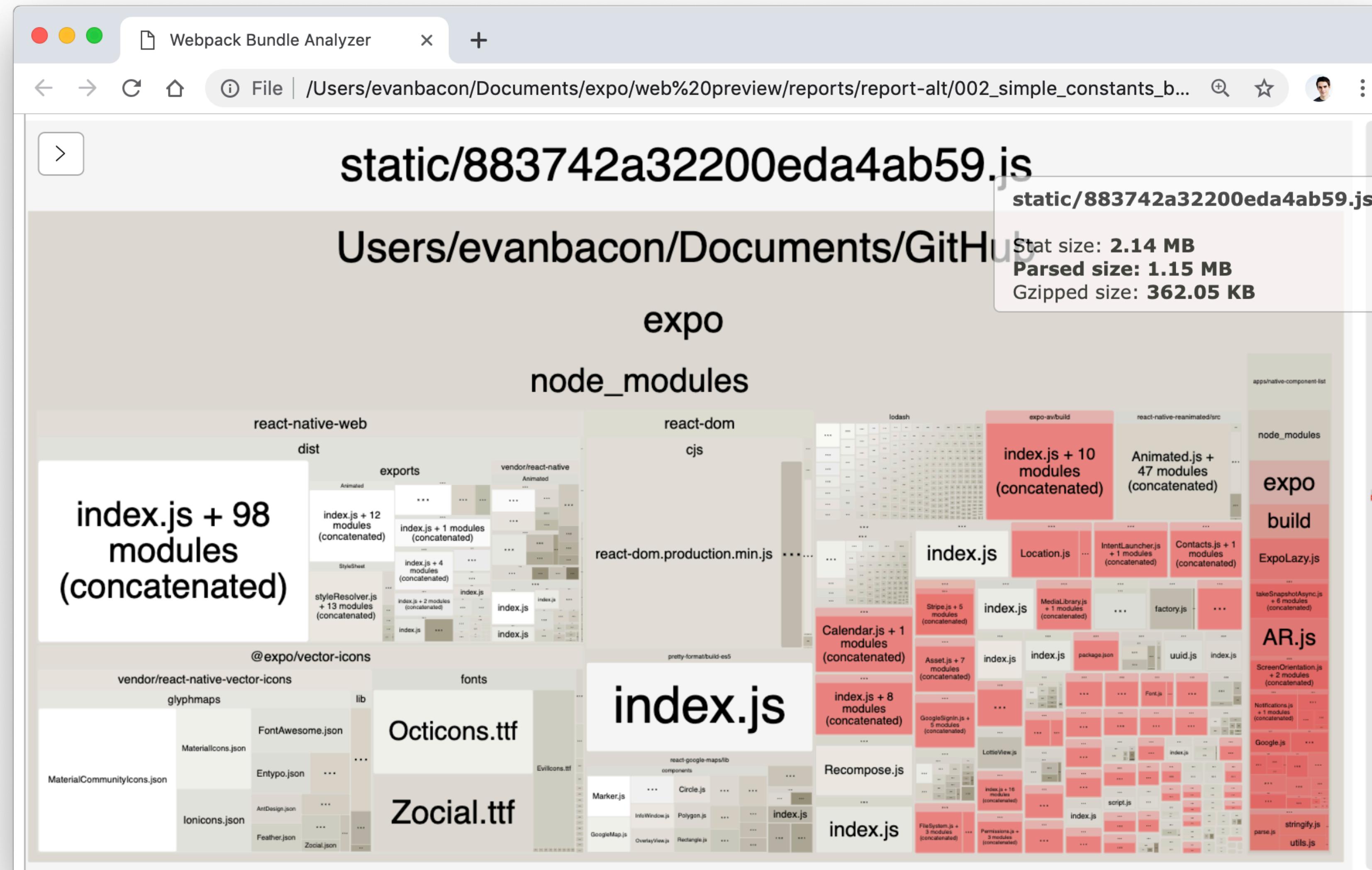
Tree Shaking

```
import { Text } from 'react-native';

import { Constants } from 'expo';

export default () => <Text>😢 I don't feel so good</Text>
```

Tree Shaking



+308.66kb



Tree Shaking



```
module.exports = {  
  get Constants() {  
    return require('./Constants')  
  }  
}
```

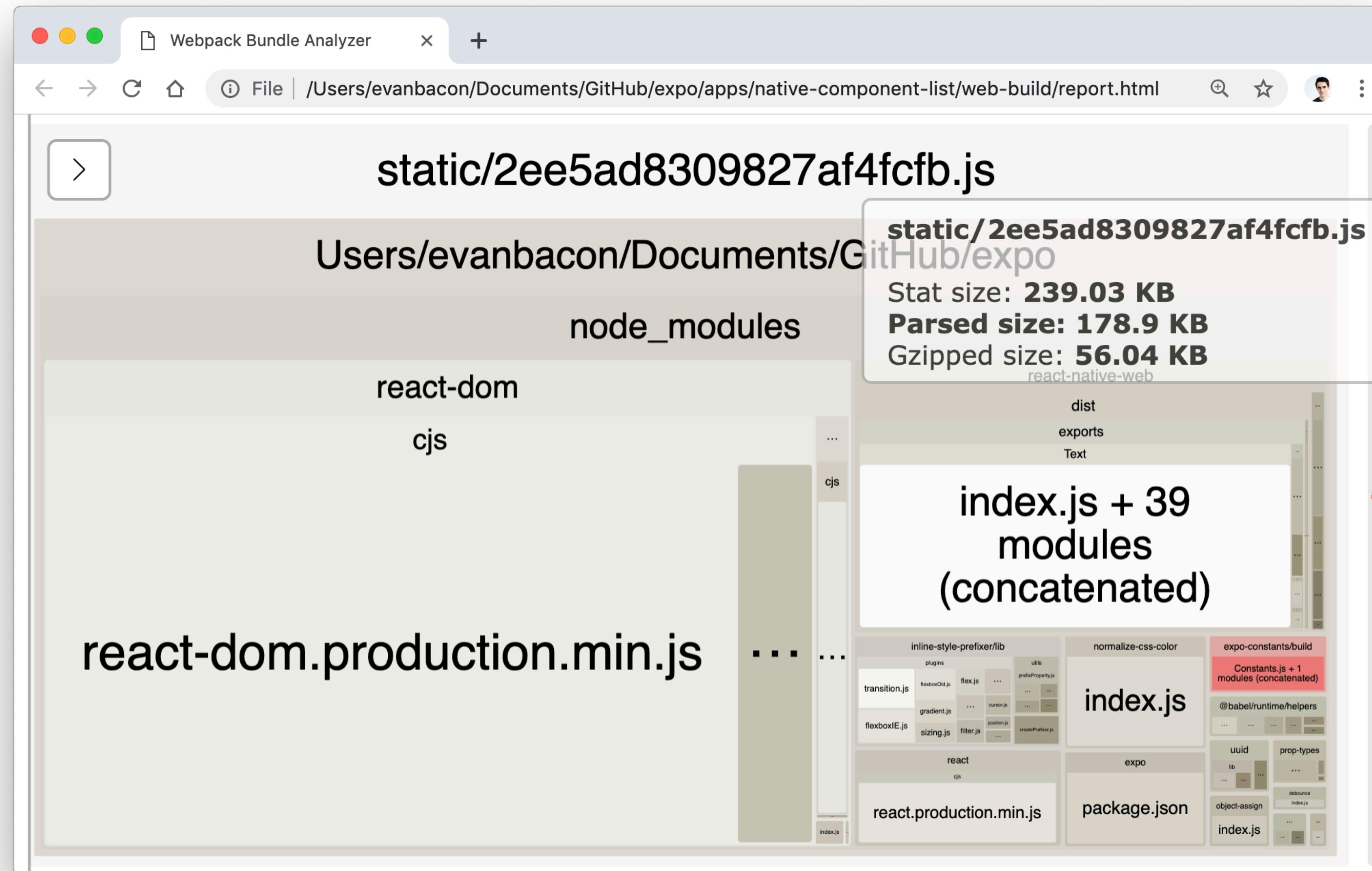
```
import Constants from './Constants'  
  
export {  
  Constants  
}
```

Tree Shaking

```
import { Text } from 'react-native';
import { Constants } from 'expo';

const App = () => (
  <Text>
    (J•ワ•)/*:・°✧ ✧°・: *＼(•ワ•＼)
  </Text>
);
```

Tree Shaking



Expo for Web

Expo CLI



Expo CLI

expo start

expo build:web

github.com/expo/web-examples



Expo CLI

The screenshot shows a browser window displaying the performance audit results for a managed workflow. The URL is <https://expomanaged.netlify.com>. The audit was run on a Nexus 5X device with dimensions 412 x 732 at 100% scale. The audit results are as follows:

Category	Score
Performance	100
Progressive Web App	100
Accessibility	100
Best Practices	100

Score scale: 0-49 (red), 50-89 (orange), 90-100 (green).

Performance

Metrics

First Contentful Paint	1.4 s ✓	First Meaningful Paint	1.4 s ✓
Speed Index	1.4 s ✓	First CPU Idle	1.4 s ✓
Time to Interactive	1.4 s ✓	Estimated Input Latency	10 ms ✓

Values are estimated and may vary.

[View Trace](#)

Diagnostics

More information about the performance of your application.

1	Minimize Critical Requests Depth	3 chains found
---	----------------------------------	----------------

Publishing

- AWS Amplify
- Now
- Netlify
- Surge
- Github Pages



Publishing

AWS Amplify

- Connect Repo
- Auto Detect Expo
- Deploy and host with
Zero Config

Expo for Web

Webpack Config

Webpack Config

```
// app.json
{
  ...
  "web": {
    "build": {
      "brotli": true,
    },
    "serviceWorker": {
      ...
    }
  }
}
```

Webpack Config

```
// webpack.config.js

const config = require('@expo/webpack-config')
const merge = require('webpack-merge')

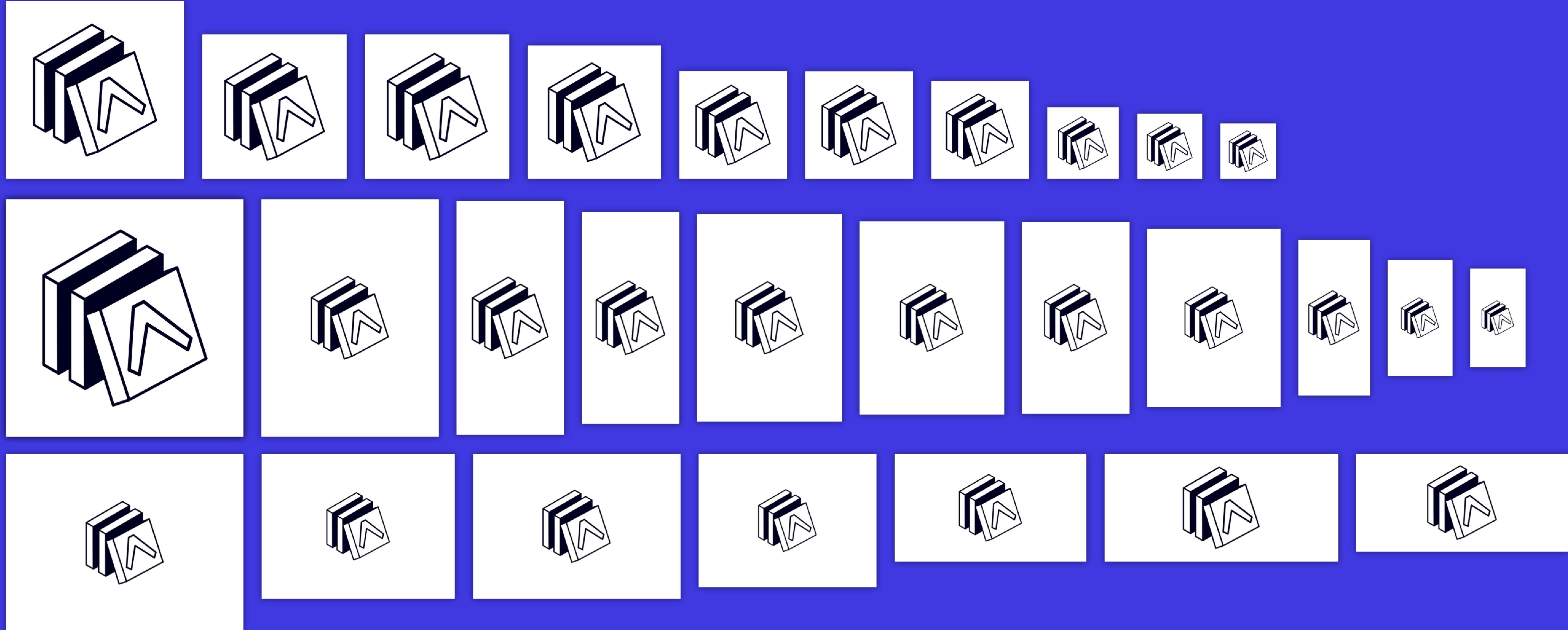
module.exports = env => merge(
  config(env), {
    // custom config...
  }
)
```

Steps to Make a Progressive Web App in Expo

Expo PWA

Make a
App in Expo

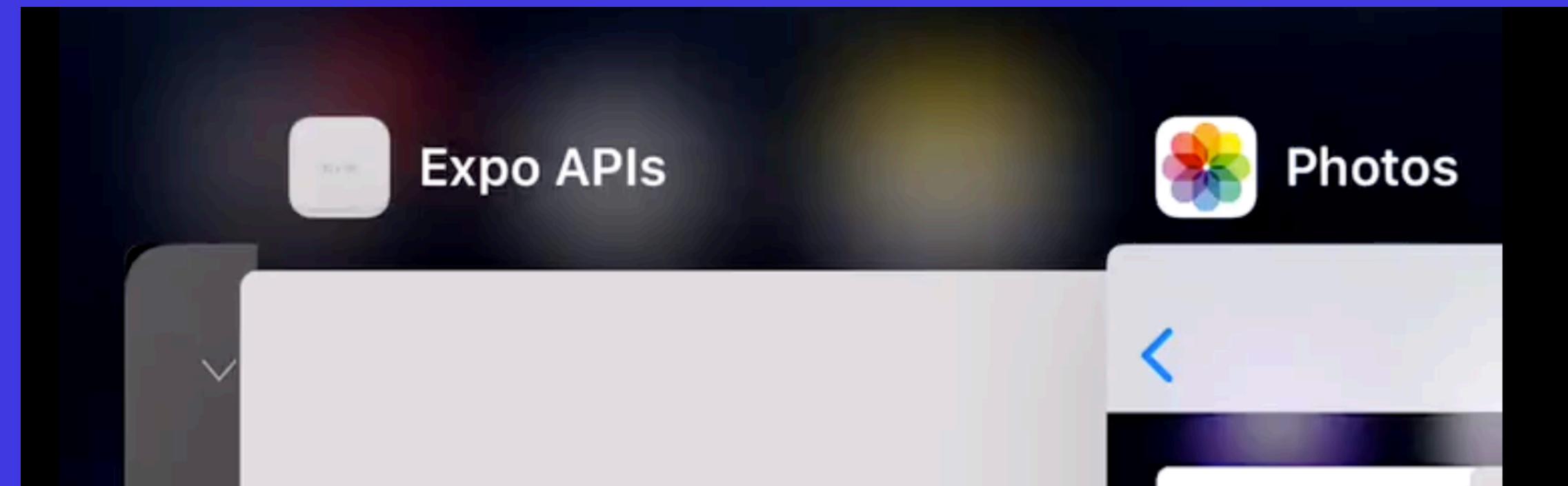
Icon Generation



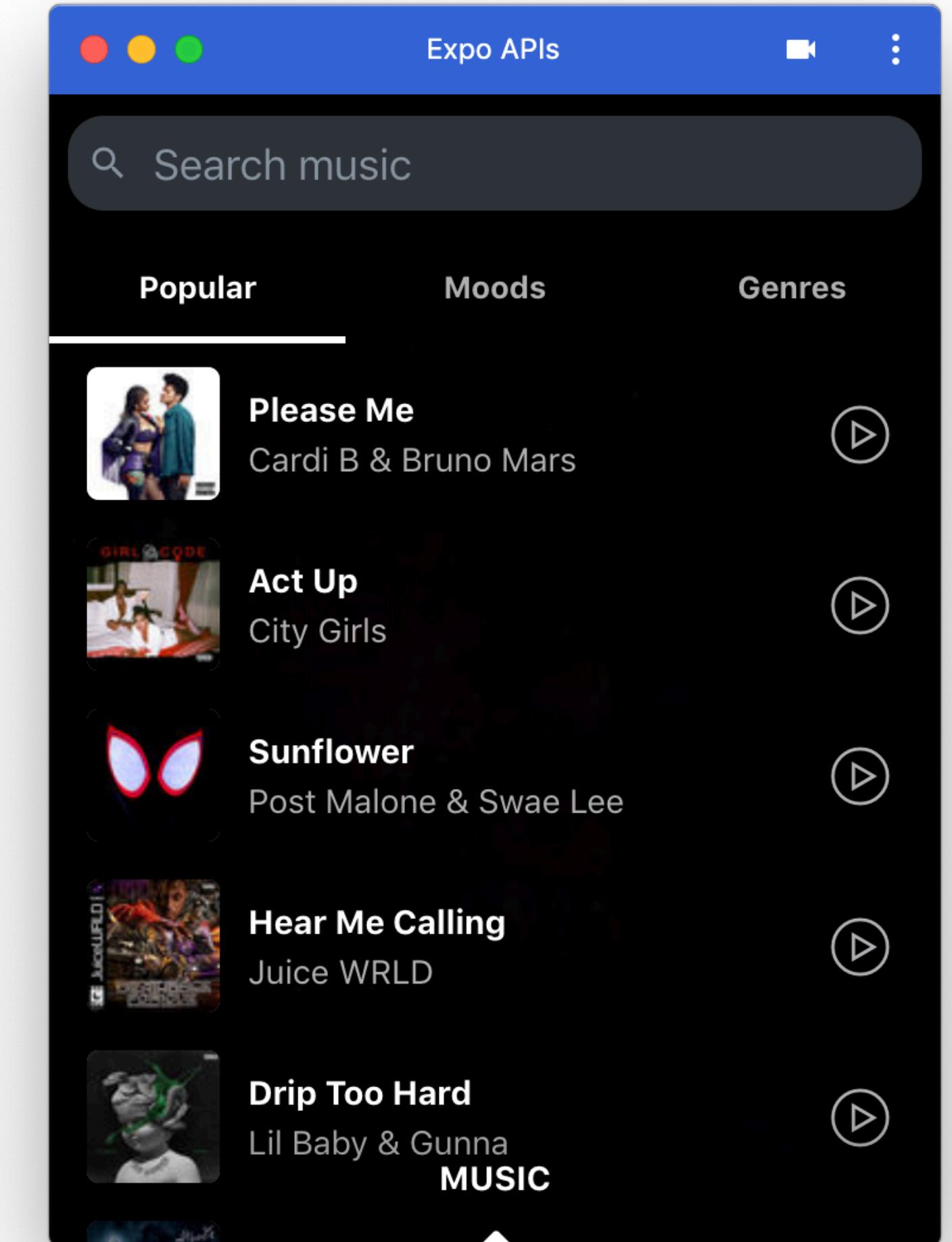
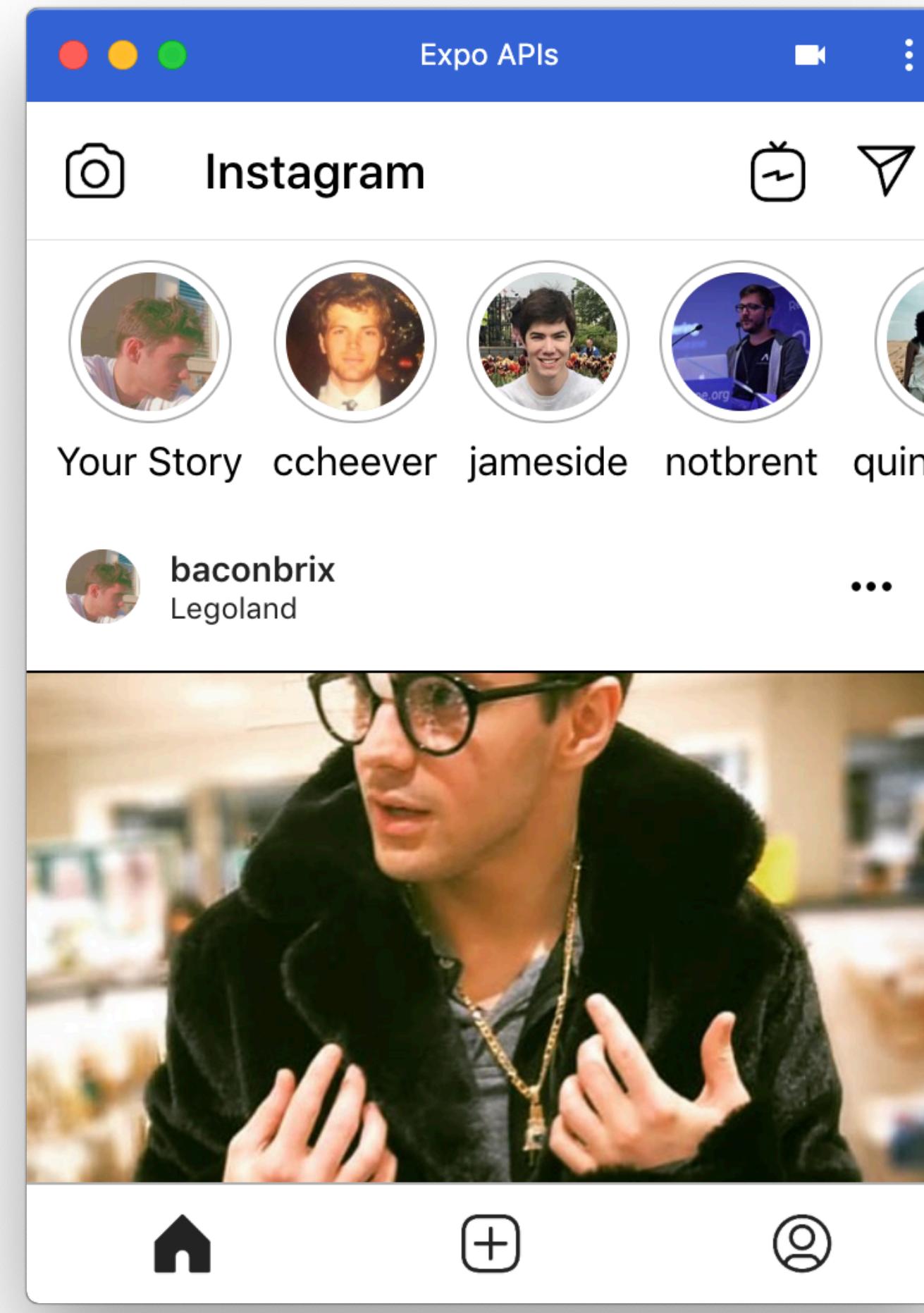
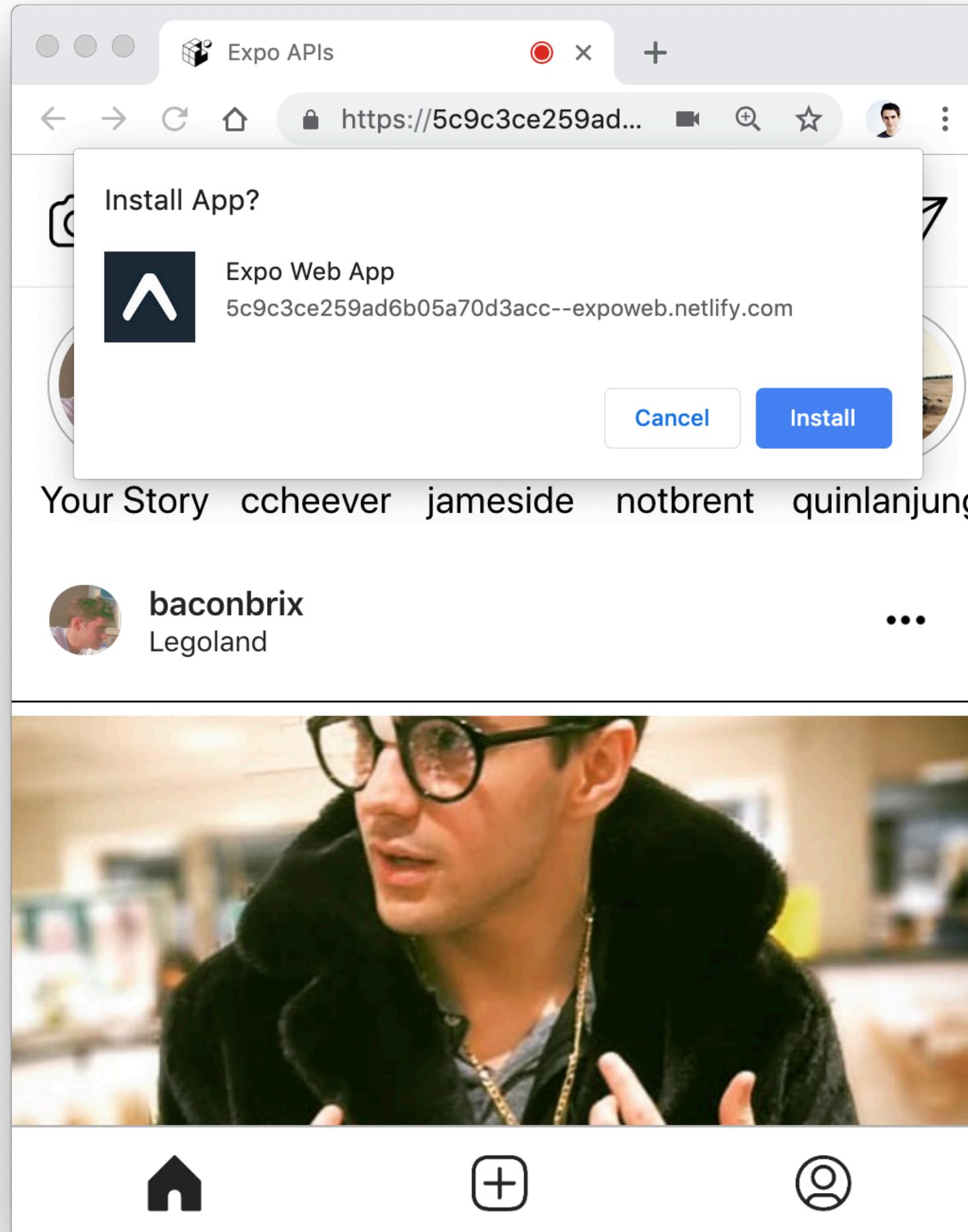
PWA Splash Screens

- Restart the Webpack Server
- Remove Existing PWA From Multi-Tasker
- Delete Existing PWA From Home Screen
- Open New Incognito Window in Safari
- Create PWA
- Return to Home Screen
- Open PWA (Load Times May Very)

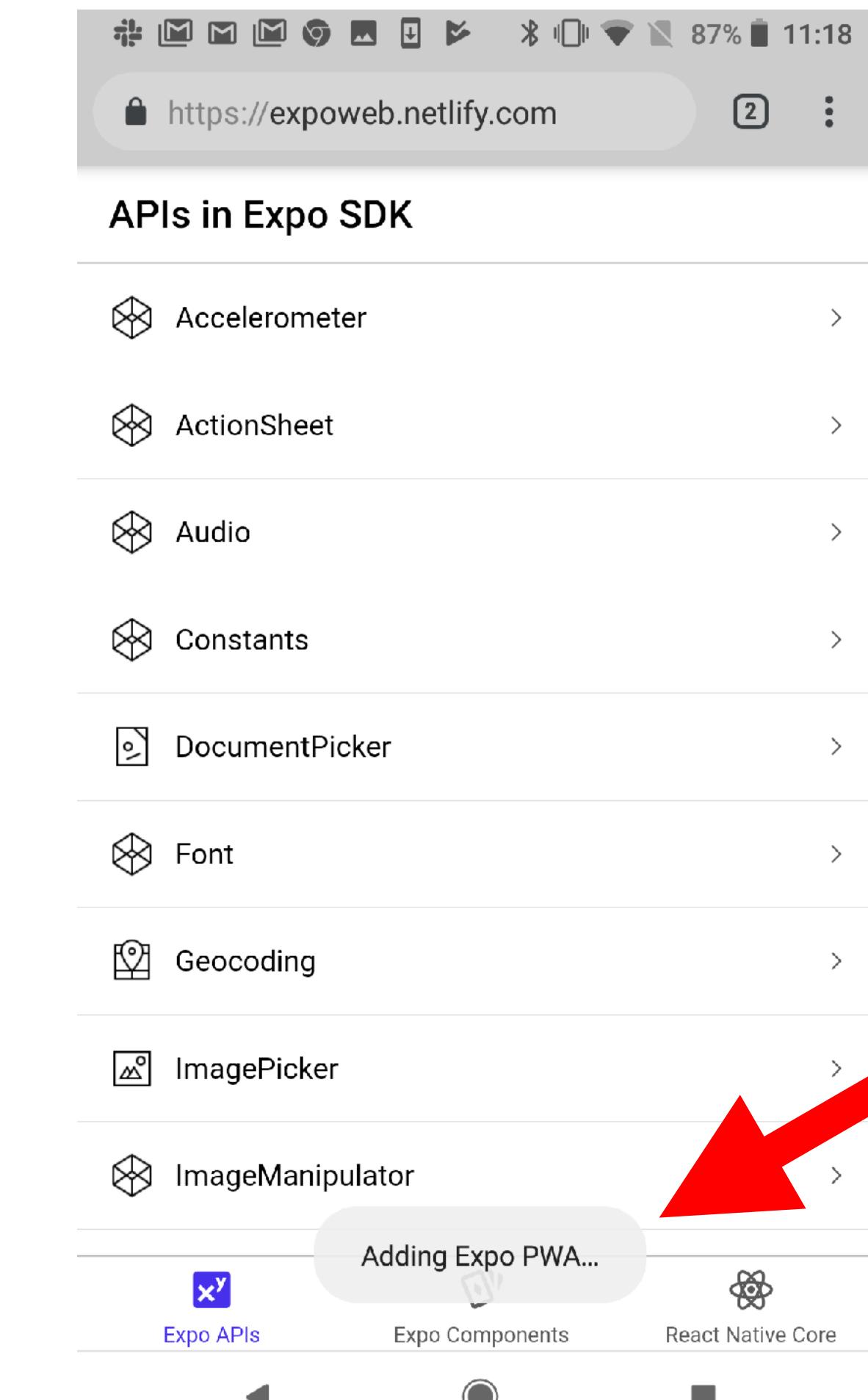
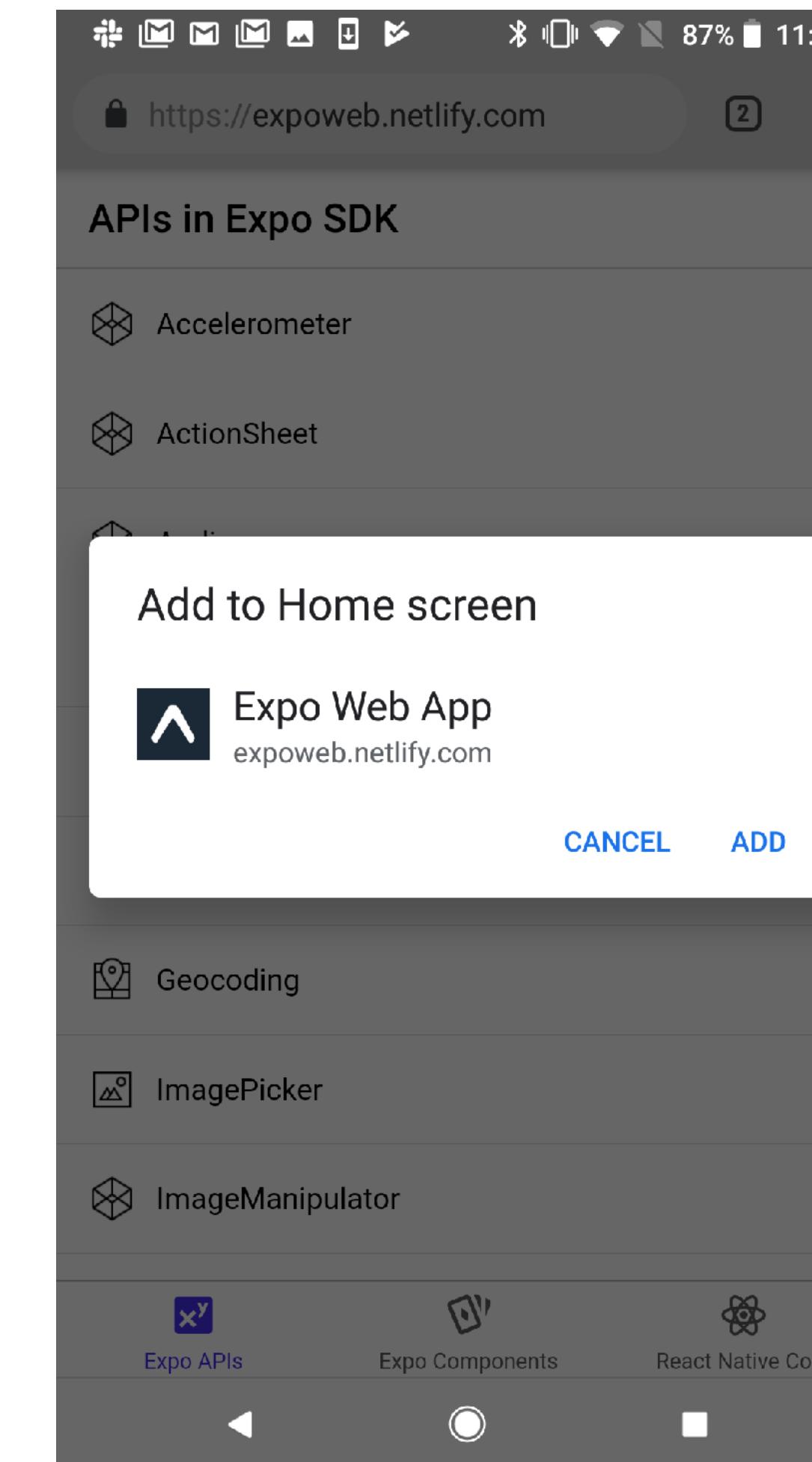
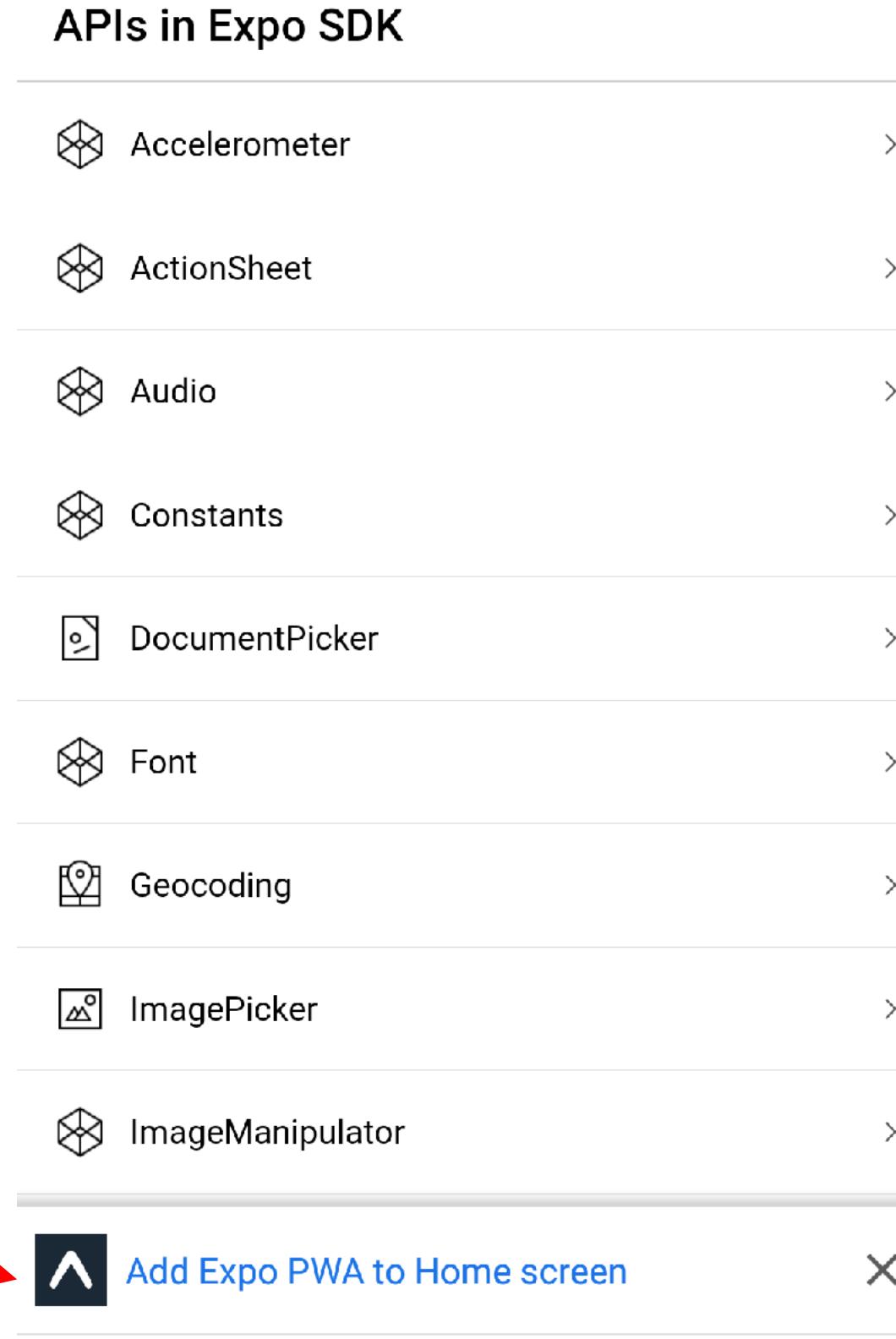
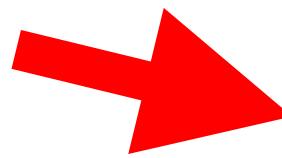
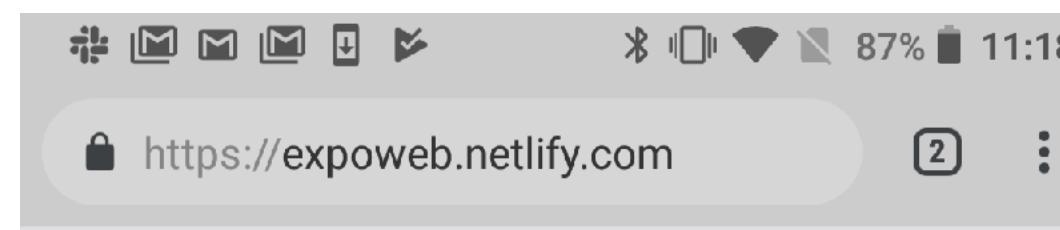
PWA Splash Screens



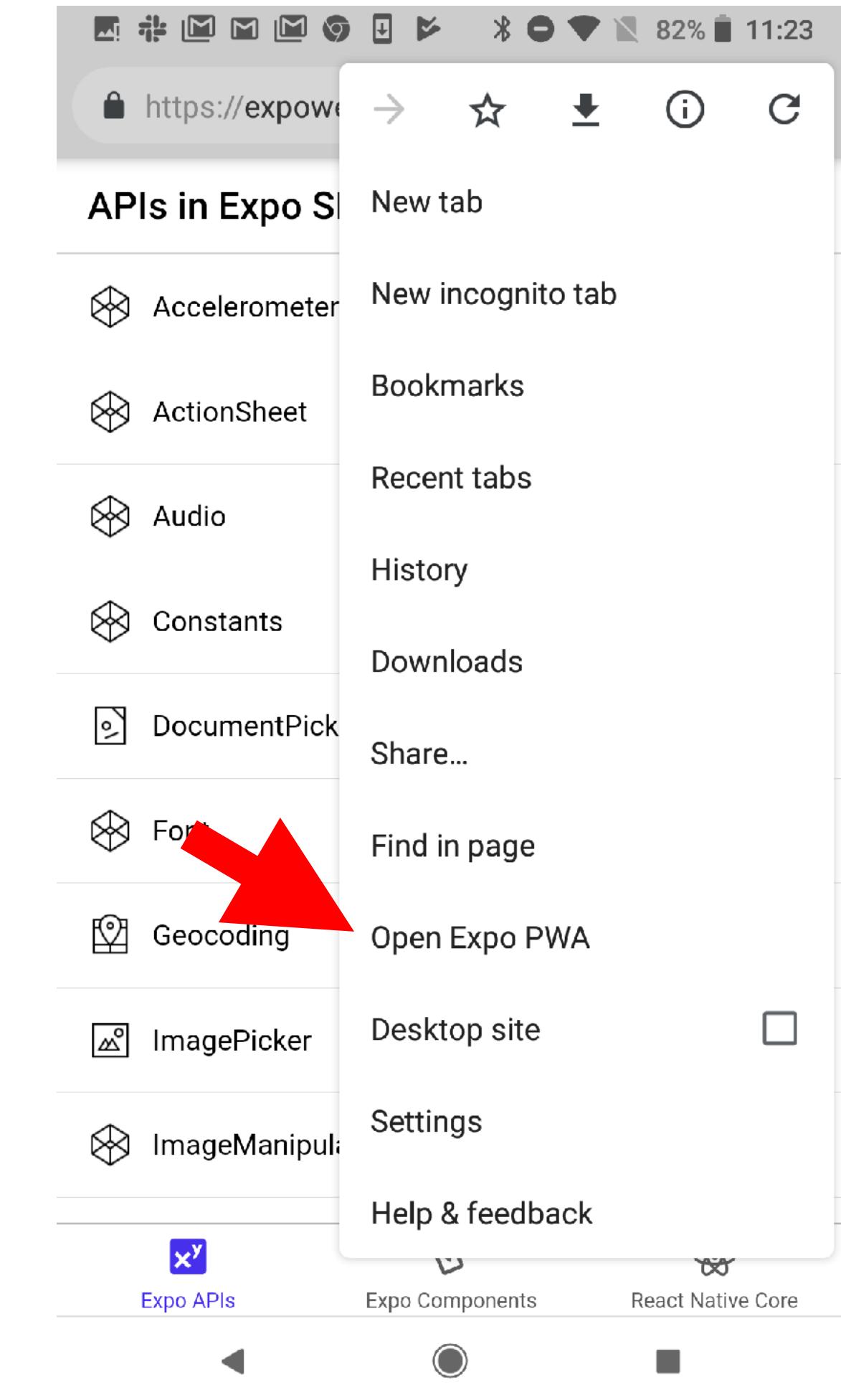
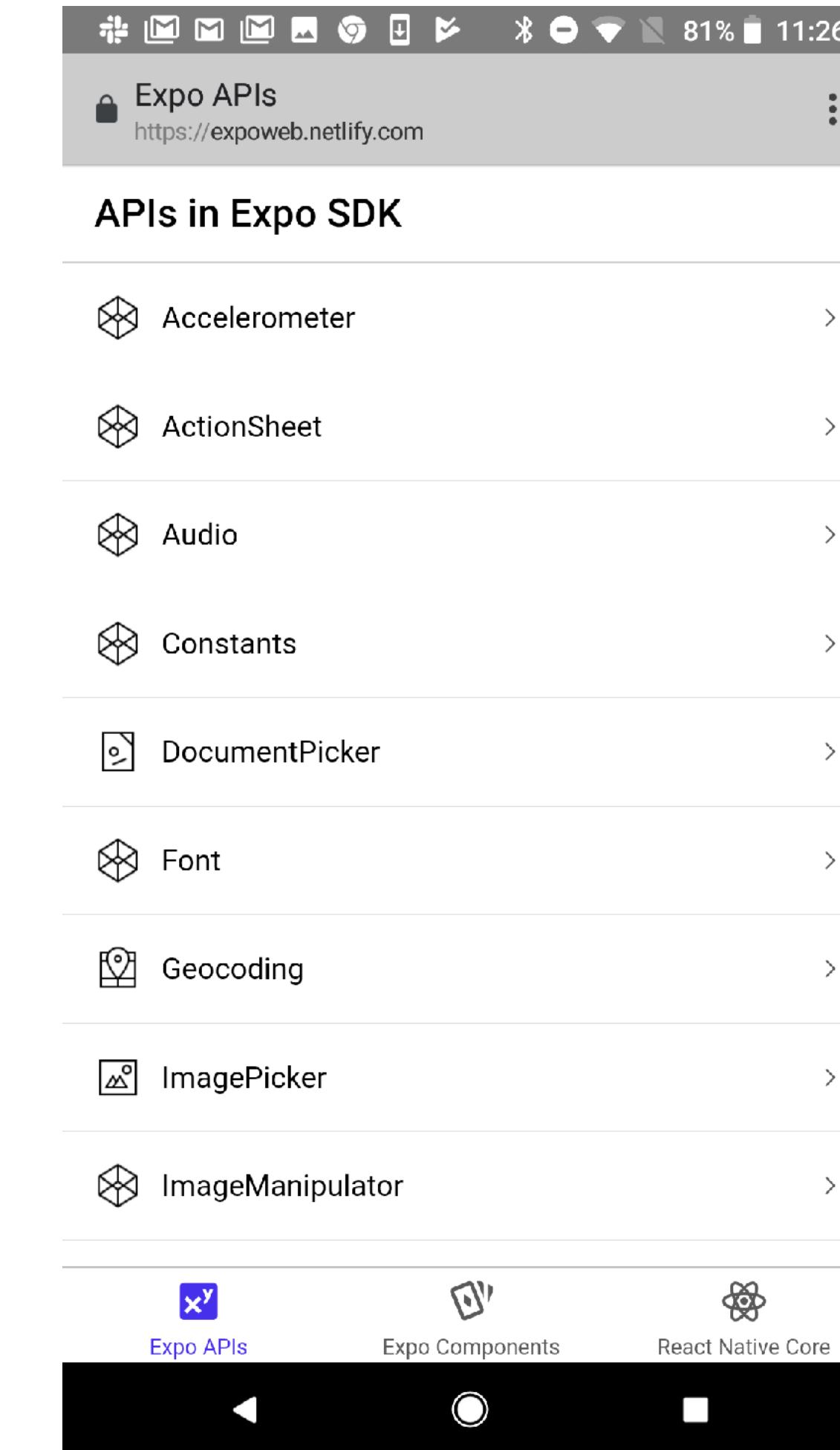
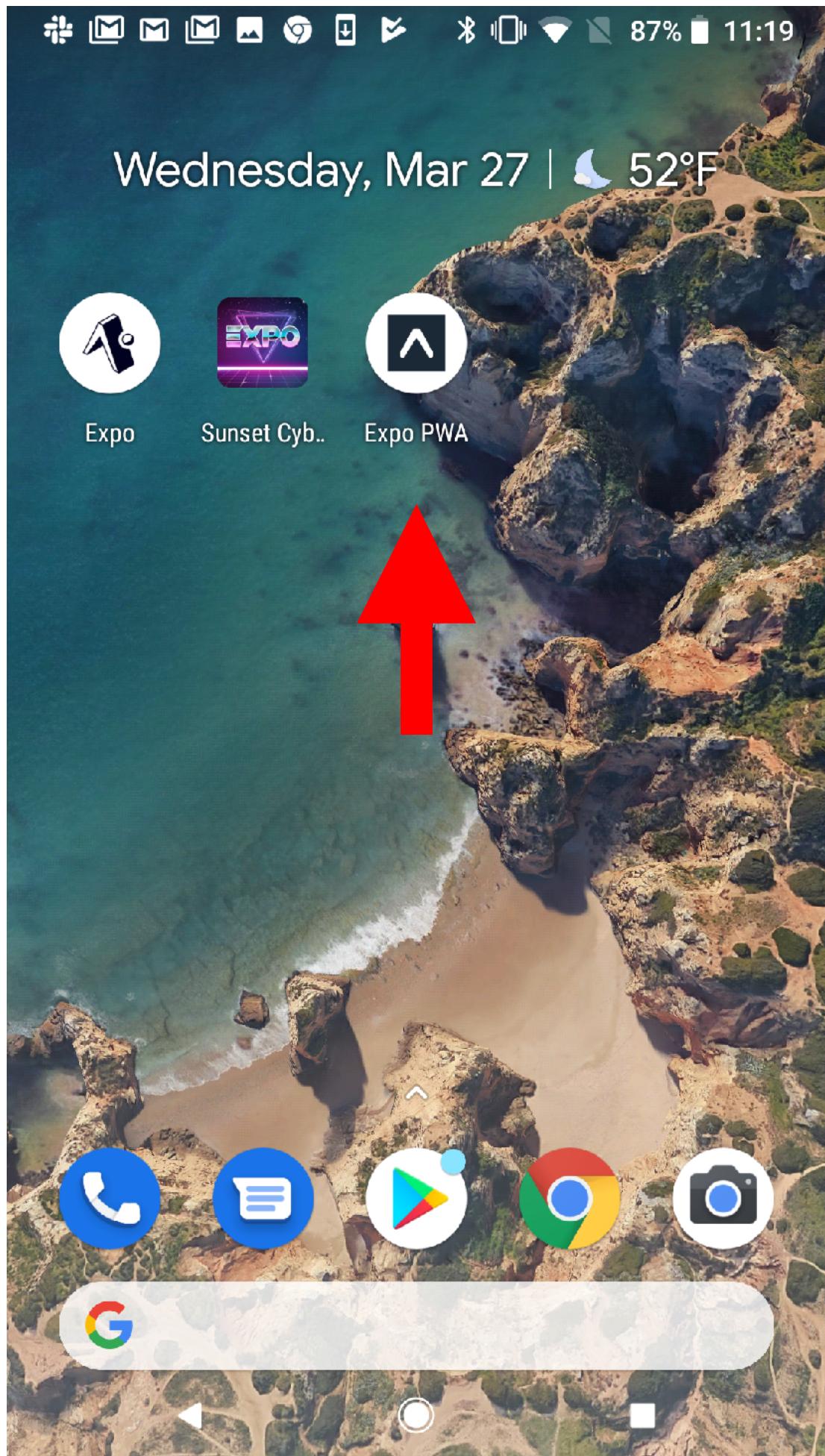
Desktop PWA



Android PWA



Android PWA





PROBLEMS

Animations



Problems

react-native-gesture-handler

Declarative API exposing platform native touch and gesture system to React Native.

 JavaScript  1.9k  211

react-native-reanimated

React Native's Animated library reimplemented

 JavaScript  1k  72

Cube Transition

Instagram

Your Story ccheever jameside notbrent quinlanjung tz

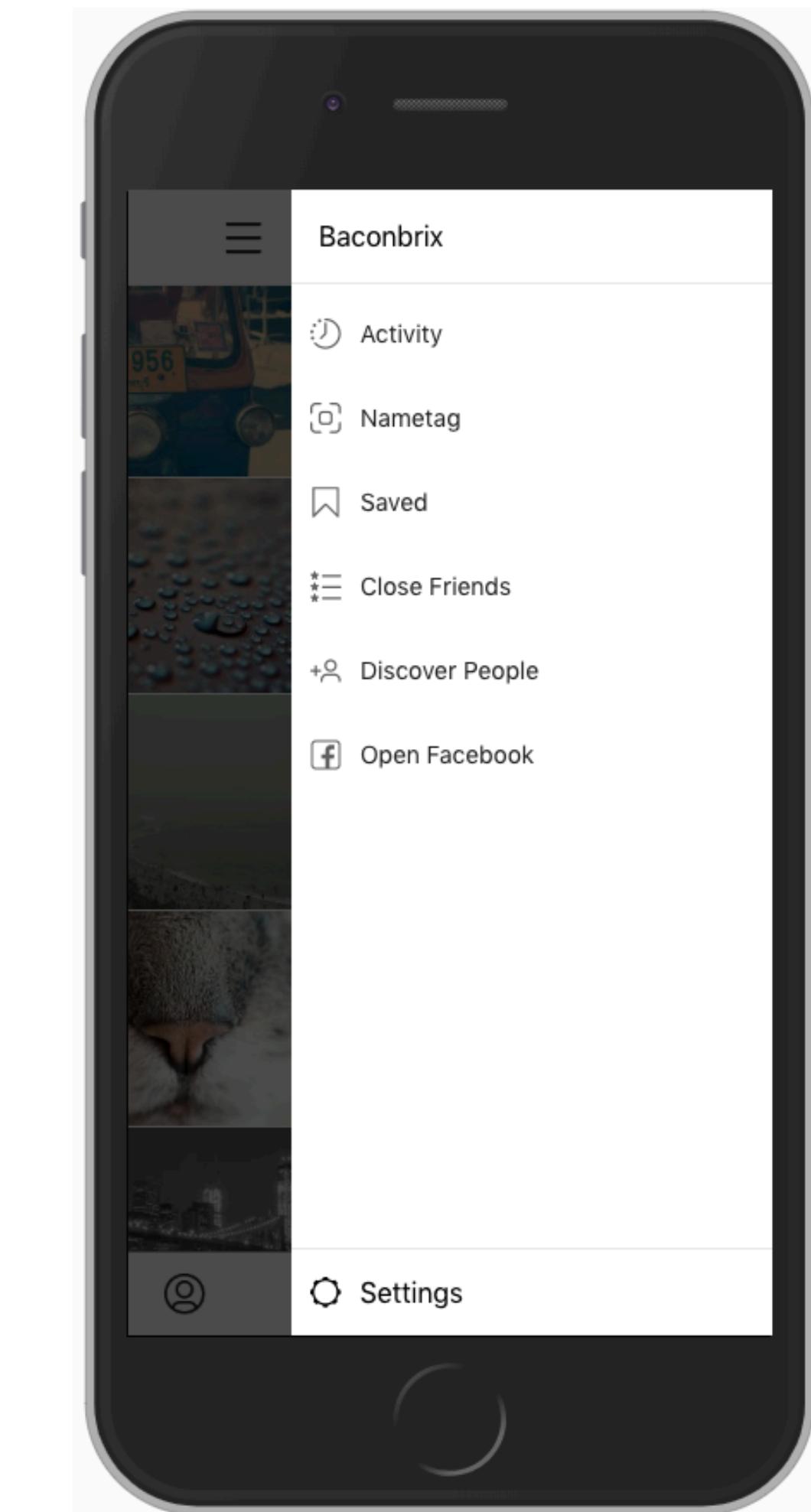
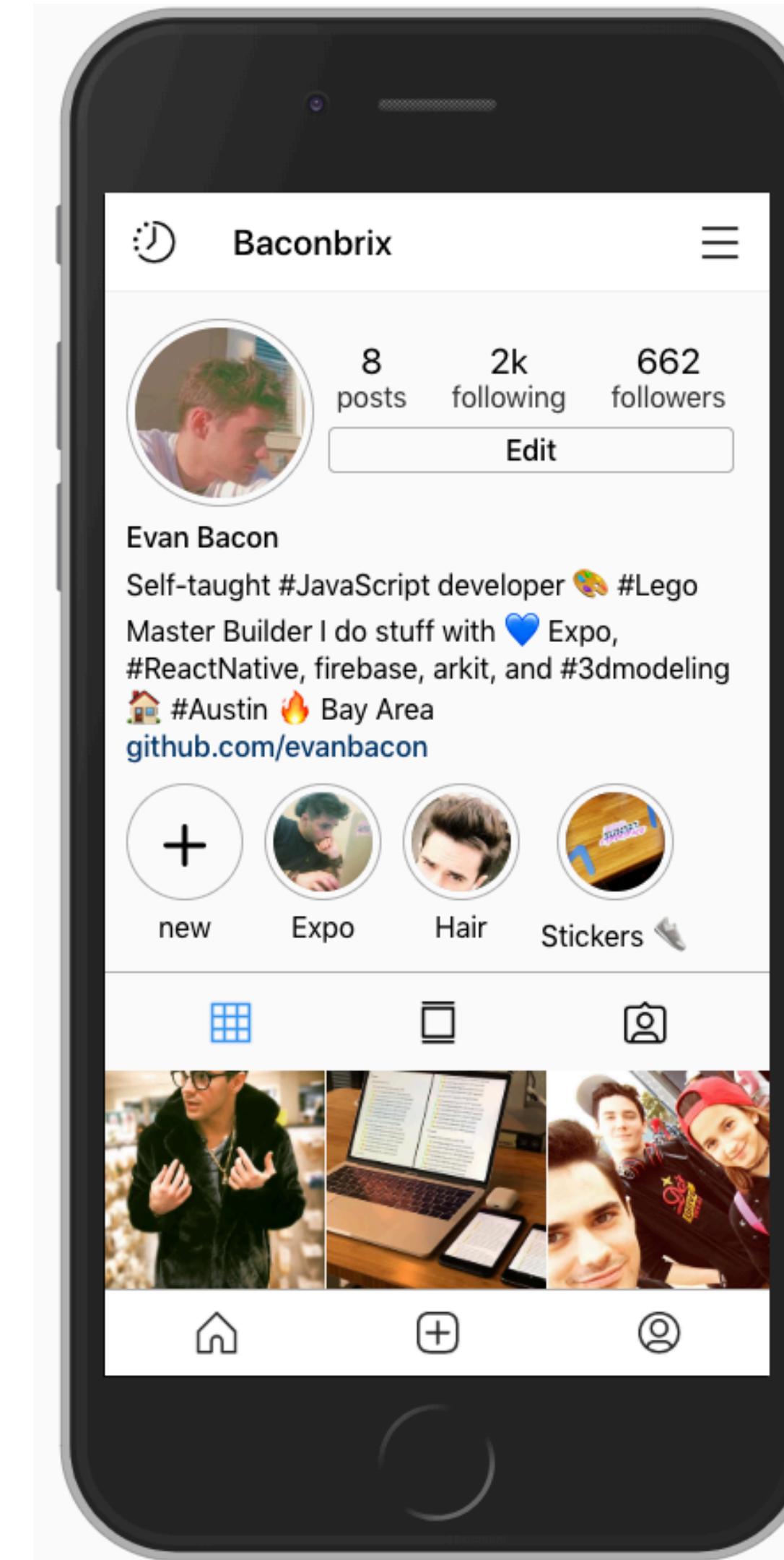
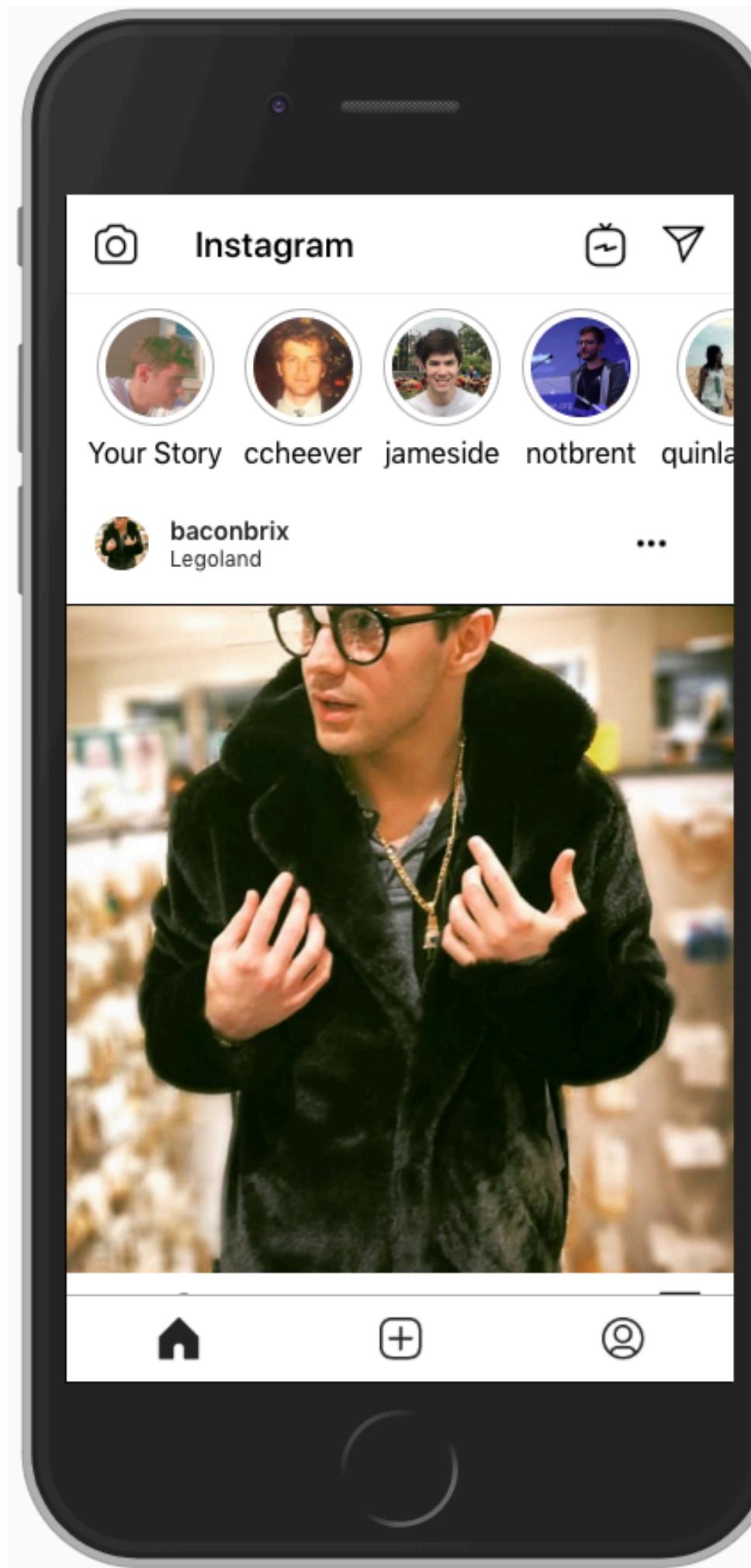
baconbrix
Legoland

...

The image shows a screenshot of the Instagram mobile application. At the top, there's a navigation bar with icons for camera, profile, direct messages, and search. Below the bar, a row of six circular profile pictures is displayed, with the second one from the left having a cursor over it. Underneath these are the handles: 'Your Story', 'ccheever', 'jameside', 'notbrent', 'quinlanjung', and 'tz'. A post by 'baconbrix' titled 'Legoland' is shown below, featuring a large video thumbnail of a man in a black hoodie and glasses gesturing. At the bottom of the screen are the standard Instagram navigation icons: home, add post, and profile.

Third Party

React Navigation



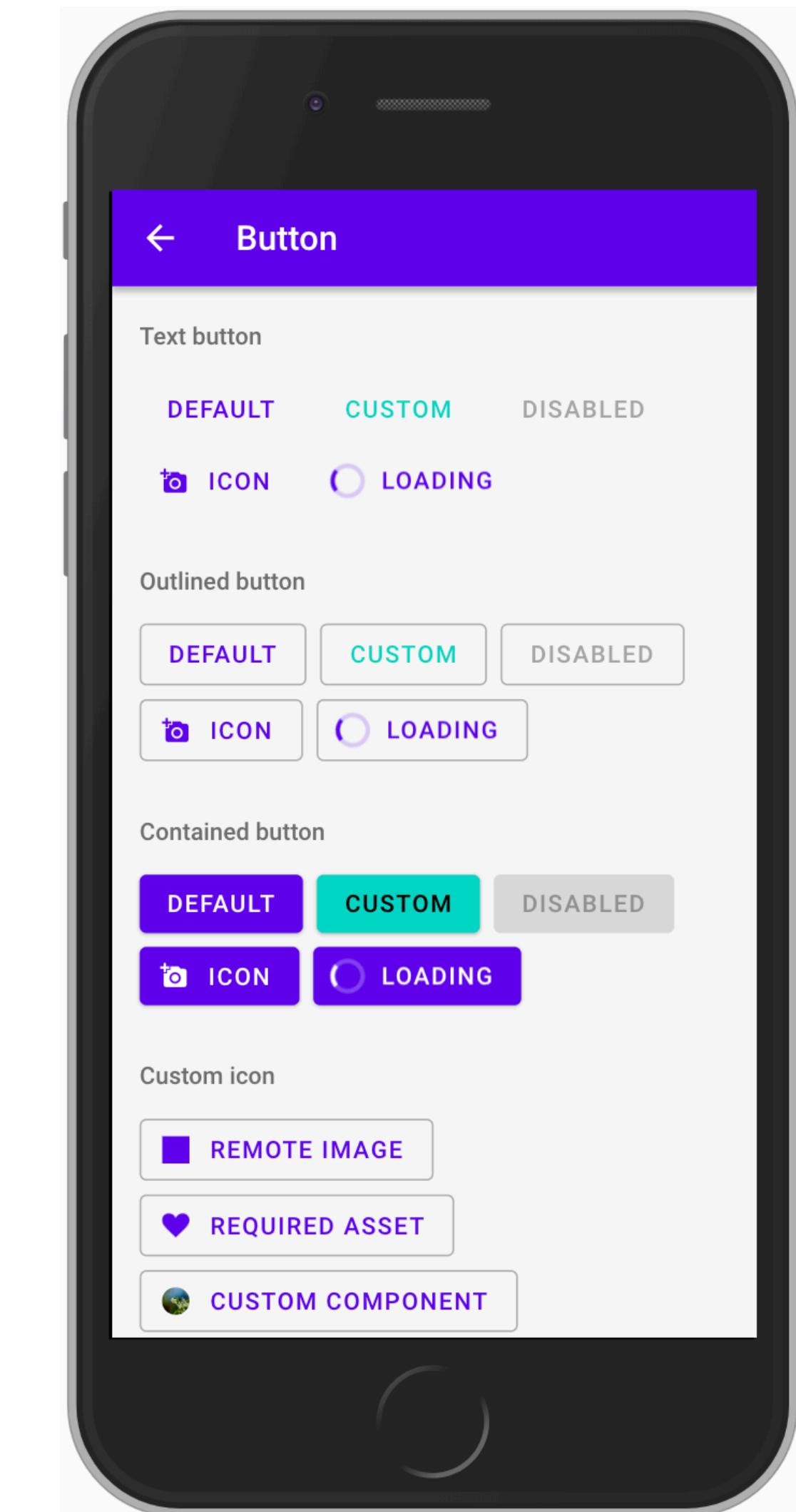
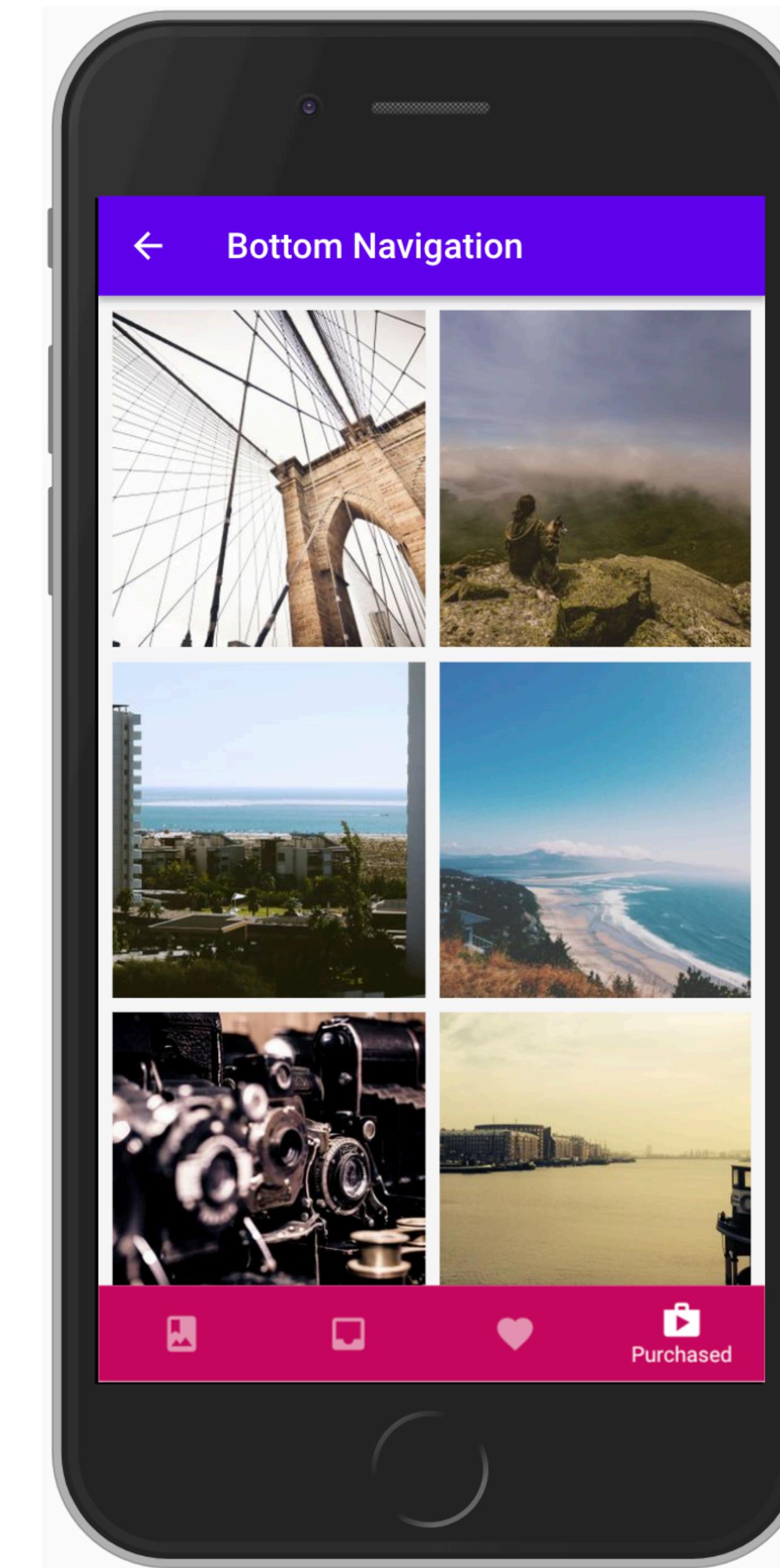
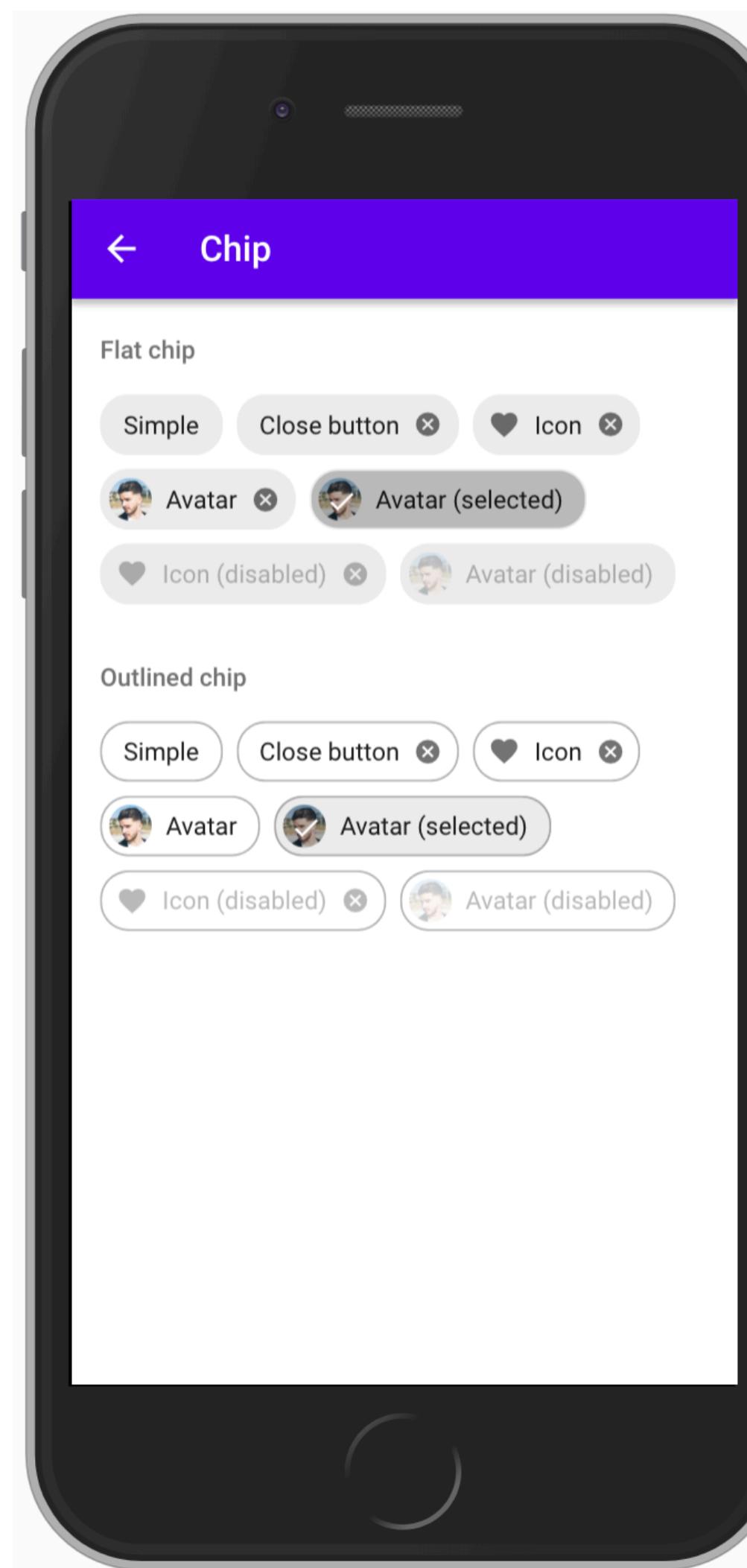
 Expo

Stack Nav

Tab Nav

Drawer Nav

React Native Paper





λ Expo

Keeping Up With the Code-Ashians

@Expo

@necolas

@baconbrix

