

BUILD A MOBILE
APP....

IN ONE HOUR!!



David Geller

BUILD A PROFESSIONAL LOOKING MOBILE APP IN AN HOUR

- ▶ Multiple ways to build and deploy mobile apps today:
 - **Native iOS** and **Android** applications
 - **Hybrid**, cross-platform applications
 - Progressive Web Apps (**PWAs**)
- ▶ Tools, documentation & learning guides are always improving
- ▶ Always important to find a balance between project requirements, expectations and development effort (including cost)

NATIVE, HYBRID OR PWA?

- ▶ Should you develop your mobile apps natively for iOS (Swift) and/or Android (Java)?
- ▶ What are your application's requirements?
- ▶ What hardware / sensor requirements do you have?
- ▶ What performance requirements do you have?
- ▶ Do you require a desktop companion application?
- ▶ What's your company's development culture?
- ▶ What are your requirements for store deployment?

HTML / CSS / JAVASCRIPT

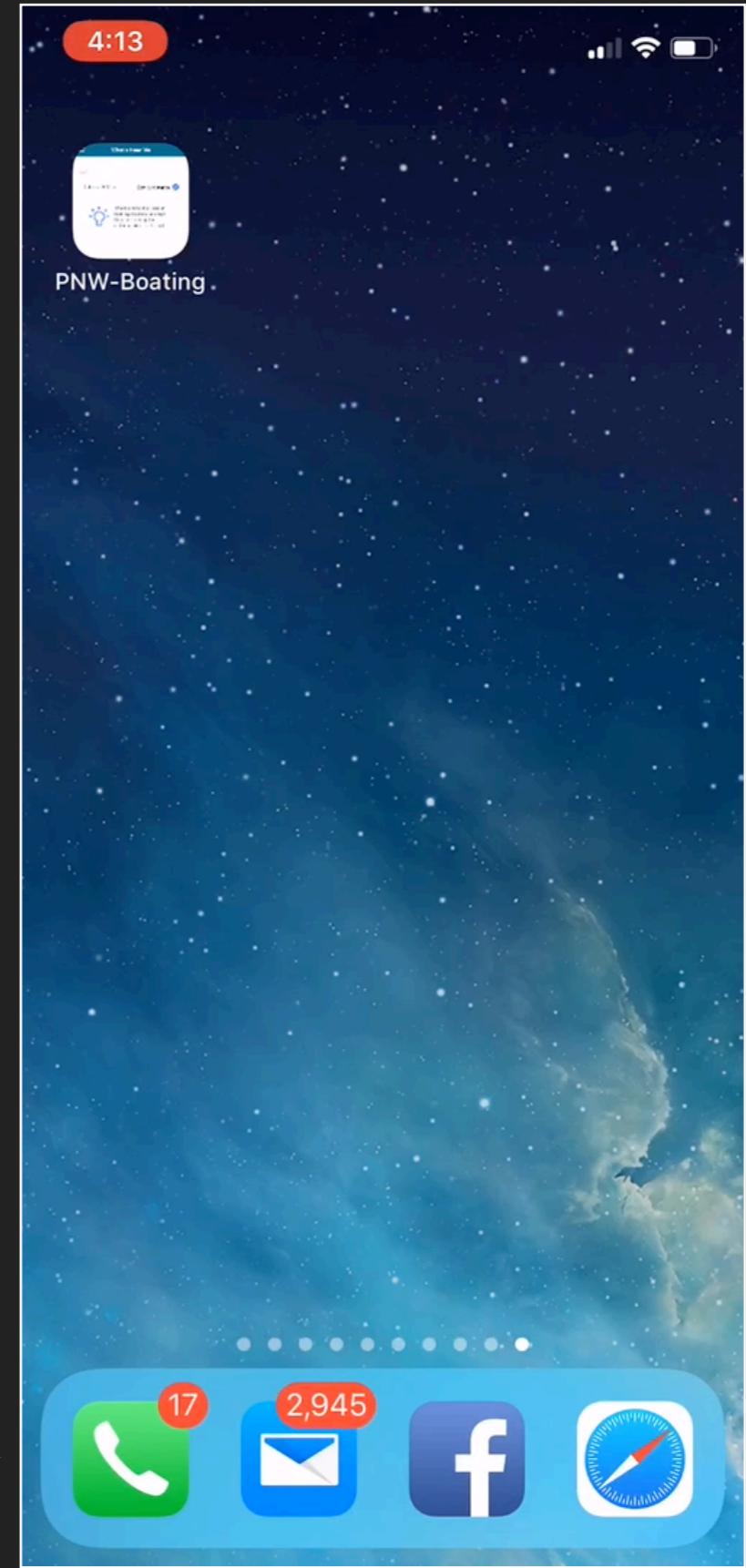
- ▶ Progressive Web Apps (PWAs) run inside [mobile] web browsers
- ▶ They're built using HTML, CSS and Javascript (Typescript)
- ▶ They're, by design, cross-platform
- ▶ They're easily deployed
- ▶ They're rapidly developed
- ▶ They don't require a store



PNWBOATING.APP

- ▶ Developed in **TypeScript**
- ▶ Uses the **Ionic Framework**
- ▶ Harnesses Google's **Firebase** platform for authentication, database and image storage
- ▶ Utilizes the camera and GPS capabilities of the phone
- ▶ Behaves like an application

BUILT WITH
IONIC 3



WHAT'S OUR MISSION TODAY?

CREATE A MOBILE APP

- ▶ Simple mobile, browser-based app that will utilize Firebase for authentication
- ▶ Allow the user to select an existing image or capture a new one - demonstrating access to device hardware (camera)
- ▶ Save that image to Firebase
- ▶ This is a sample! It's designed to give you a "taste" of creating PWAs



DEPENDENCIES

- ▶ Ionic Framework v4
(beta.ionicframework.com/docs)
- ▶ node.js
- ▶ npm
- ▶ angular-cropperjs (for image cropping)
- ▶ Firebase (firebase.google.com)
- ▶ angular-firebase (to connect to Google's Firebase)



STEP 1: INSTALL YOUR TOOLS

- ▶ Install node.js and npm (nodejs.org/en)
- ▶ Install Visual Studio Code (code.visualstudio.com)
- ▶ Install Ionic Framework

```
[ sudo ] npm install -g ionic
```

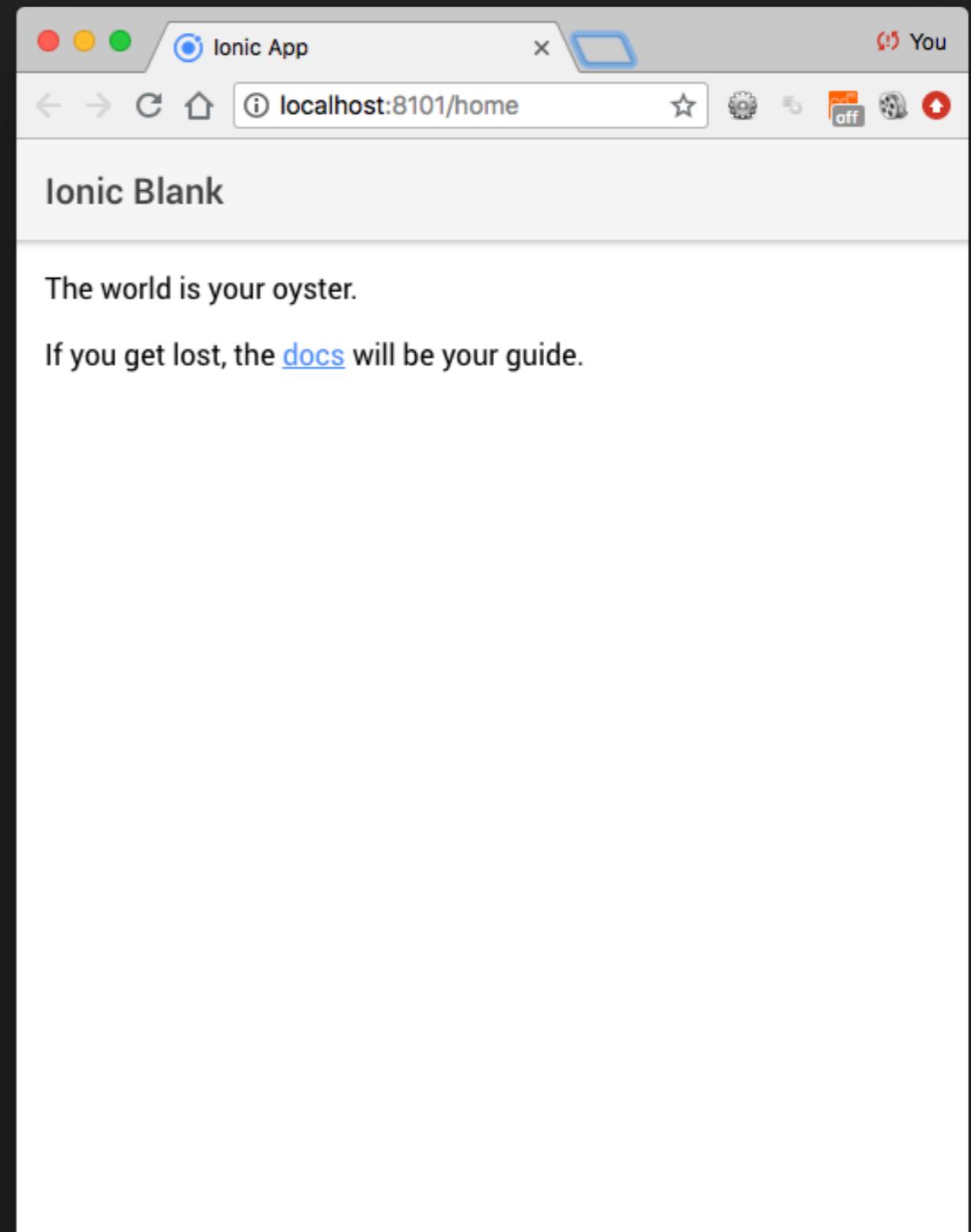
- ▶ Create a new Ionic project
- ```
ionic start PhotoGrab blank --type=angular
```

- ▶ Test our new app
- ```
cd PhotoGrab
ionic serve --browser "Google Chrome"
```

STEP 2: TEST OUR NEW APP

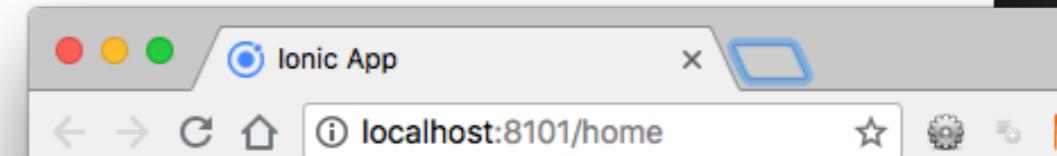
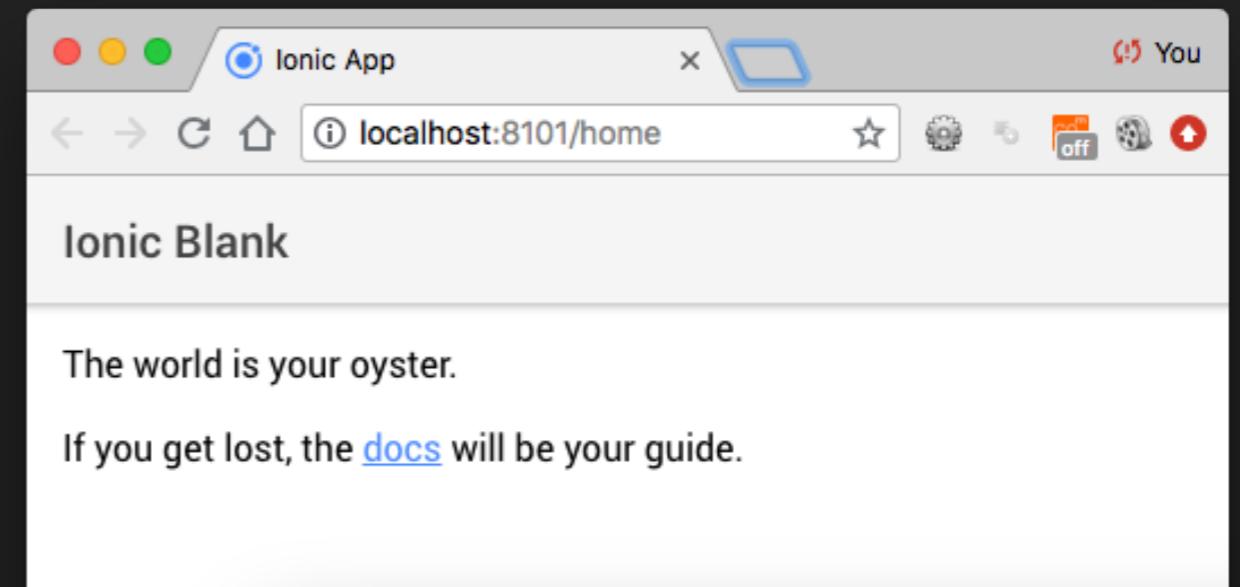
- ▶ Test our new app

```
ionic serve --browser  
"Google Chrome"
```



STEP 3: MODIFY OUR APP

- ▶ Modify the file **home.page.html** to see how Ionic changes the application in “real-time”



STEP 4: ADD PAGES TO OUR APP

- ▶ Install Ionic Framework

```
ionic g page login
```

STEP 5: INSTALL FIREBASE AND ANGULAR FIRE

- ▶ Install libraries to work with Google Firebase

```
npm i firebase @angular/fire --save
```

PACKAGE.JSON

```
"@angular/common": "~6.1.1",
"@angular/core": "~6.1.1",
"@angular/fire": "^5.0.0",
"@angular/forms": "~6.1.1",
"@angular/http": "~6.1.1",
"@angular/platform-browser": "~6.1.1",
"@angular/platform-browser-dynamic": "~6.1.1",
"@angular/router": "~6.1.1",
"@ionic-native/core": "5.0.0-beta.15",
"@ionic-native/splash-screen": "5.0.0-beta.15",
"@ionic-native/status-bar": "5.0.0-beta.15",
"@ionic/angular": "4.0.0-beta.7",
"angular-cropperjs": "^0.1.5",
"core-js": "^2.5.3",
"firebase": "^5.5.0",
"fix-orientation": "^1.0.0",
"rxjs": "6.2.2",
"zone.js": "^0.8.26"
```

STEP 6: CREATE A FIREBASE ACCOUNT / PROJECT

- ▶ Visit Firebase
firebase.google.com
- ▶ Create a new project
- ▶ Click on the Authentication link then select “sign-in method” tab and enable “Email/Password”
- ▶ Create a new user for yourself
(you can add registration features on your own)

STEP 7: ACTIVATE LOGIN PAGE

- ▶ Copy app.module.ts file
- ▶ Copy firebase configuration setting for an HTML project
- ▶ Copy login.page.html and login.page.ts files (copy all the login directory files)
- ▶ Copy app-routing.module.ts (talk about routes)
- ▶ Create assets/images directory and copy over place_holder.png
- ▶ Install Angular CropperJS
`npm i angular-cropperjs@v0.1.5 --save`
- ▶ Test app

GETTING YOUR FIREBASE CONFIG DATA

Database

Storage

Hosting

Functions

ML Kit

Quality

Crashlytics

Performance

Test Lab

Analytics

Dashboard, Events, Convers

Grow

Predictions, A/B Testing, CI

Your project

Add Firebase to your web app

Copy and paste the snippet below at the bottom of your HTML, before other script tags.

```
<script src="https://www.gstatic.com/firebasejs/5.5.0.firebaseio.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyBbx0k_-UQ2fbTgIYNx0ww9bJjPwTUCTXo",
    authDomain: "photograb-14821.firebaseio.com",
    databaseURL: "https://photograb-14821.firebaseio.com",
    projectId: "photograb-14821",
    storageBucket: "photograb-14821.appspot.com",
    messagingSenderId: "277595523941"
  };
  firebase.initializeApp(config);
</script>
```

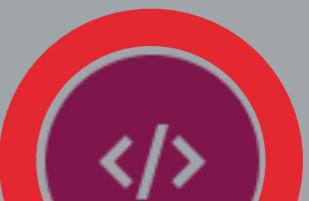
Copy

Check these resources to learn more about Firebase for web apps:

[Get Started with Firebase for Web Apps](#)

[Firebase Web SDK API Reference](#)

[Firebase Web Samples](#)



STEP 7A: UPDATE OUR HOME PAGE, CREATE GLOBAL SERVICE

- ▶ Copy over files for the home directory

home.page.html

home.page.ts

- ▶ Create a service for sharing data

ionic g service Globals

copy over Globals.service.ts

- ▶ Create loading indicator class

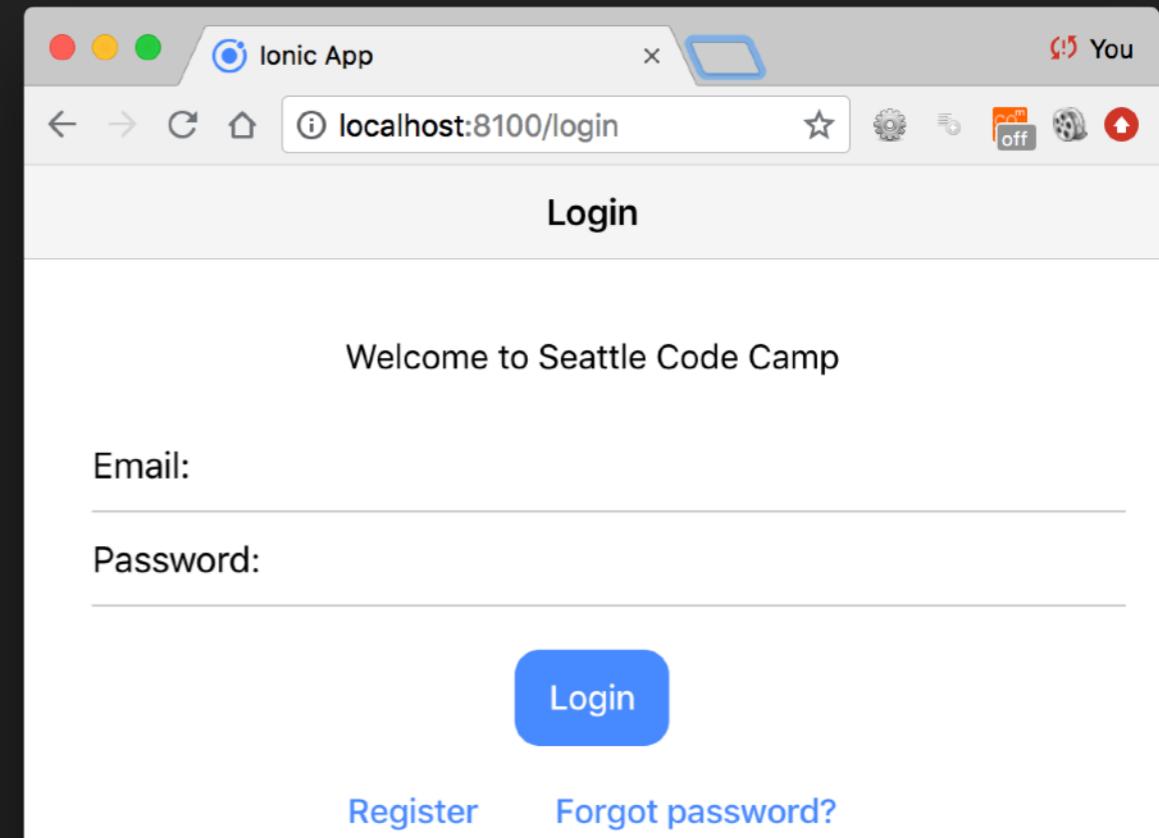
create “custom” directory under src

create loading.ts

copy contents for loading.ts

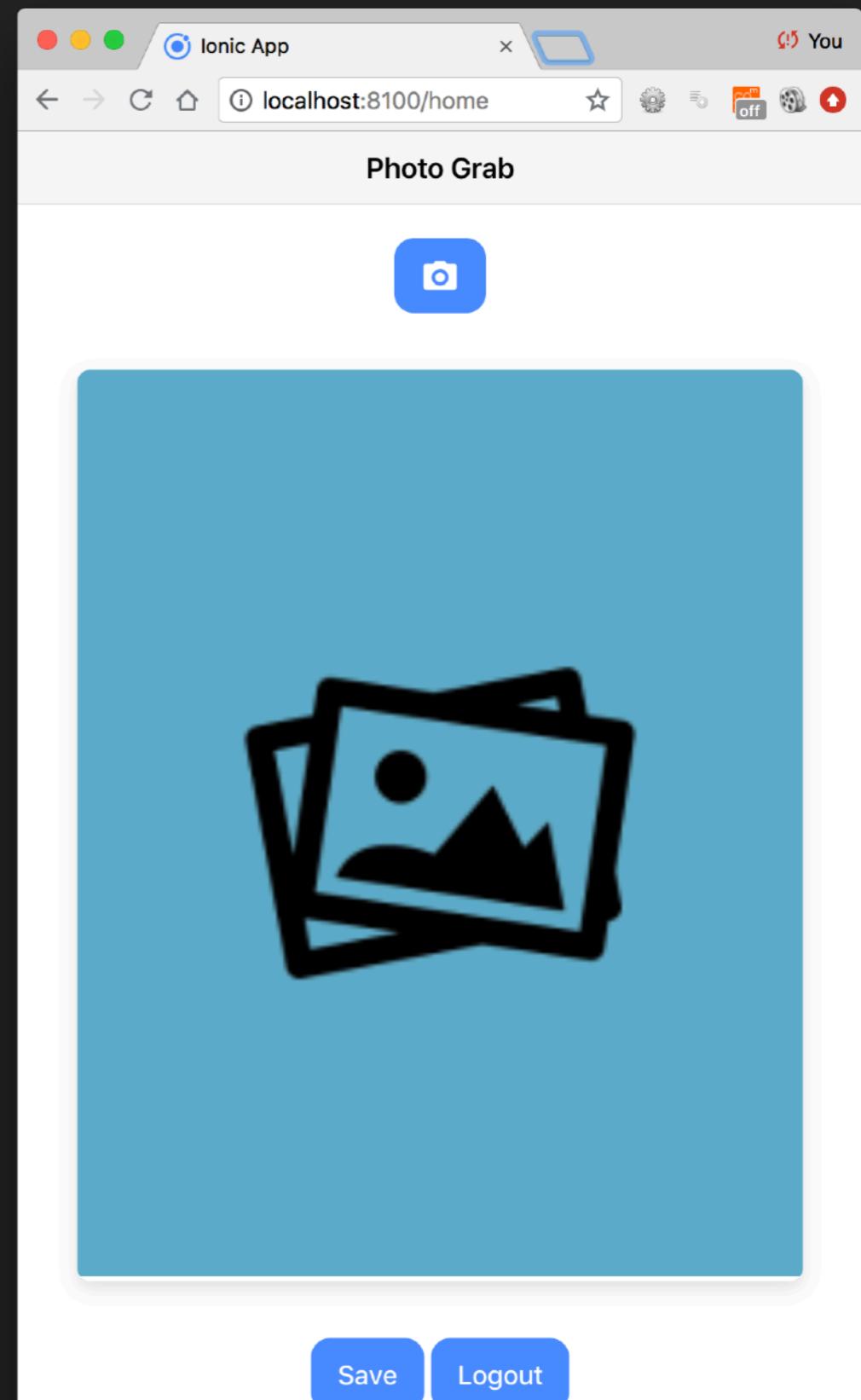
LOGGING INTO PLATFORM

- ▶ Running the app should display our login page
- ▶ Make sure our Javascript console is open:
Opt-Cmd-J
- ▶ Attempt to login
- ▶ If successful we'll land on our home page.

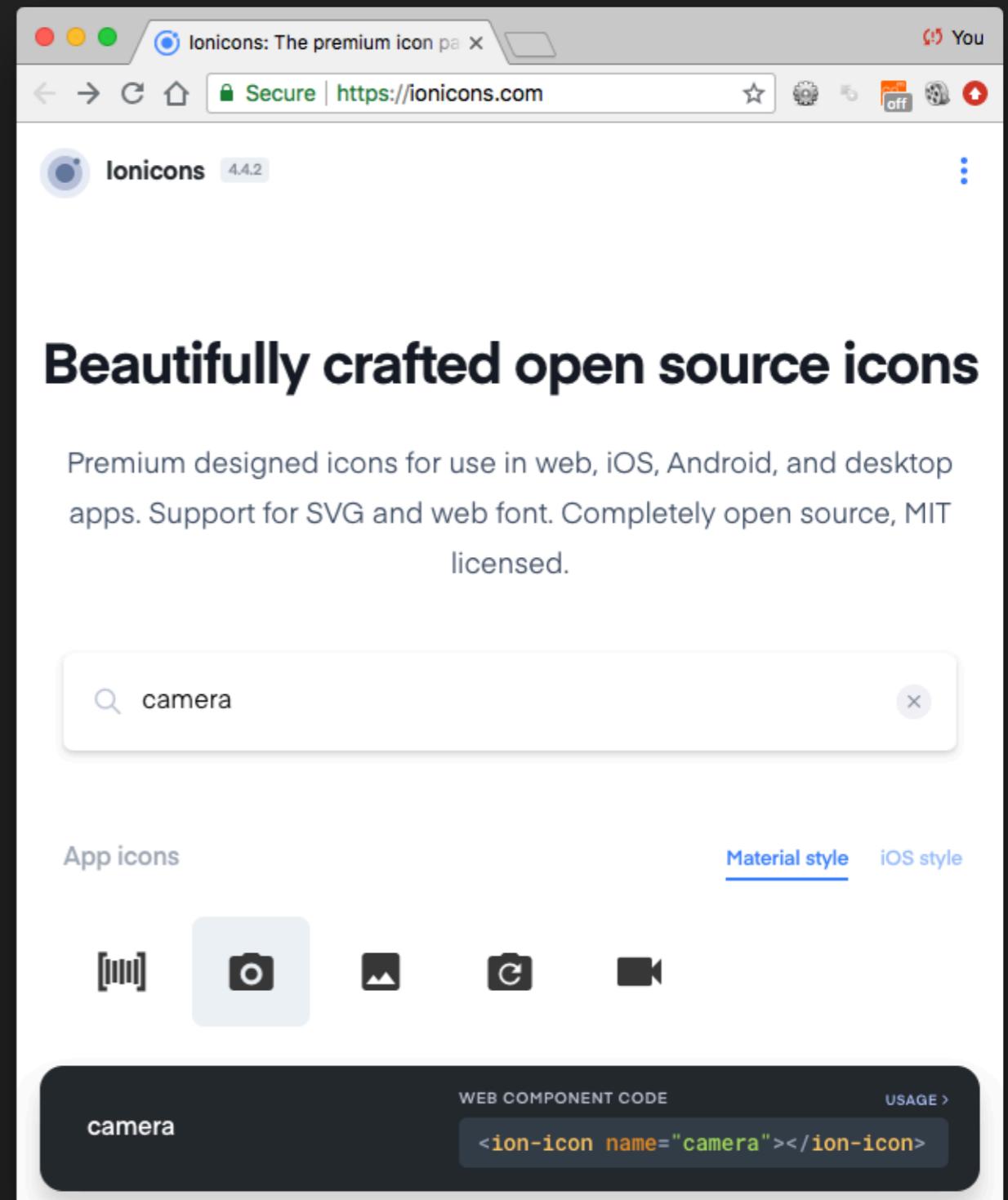
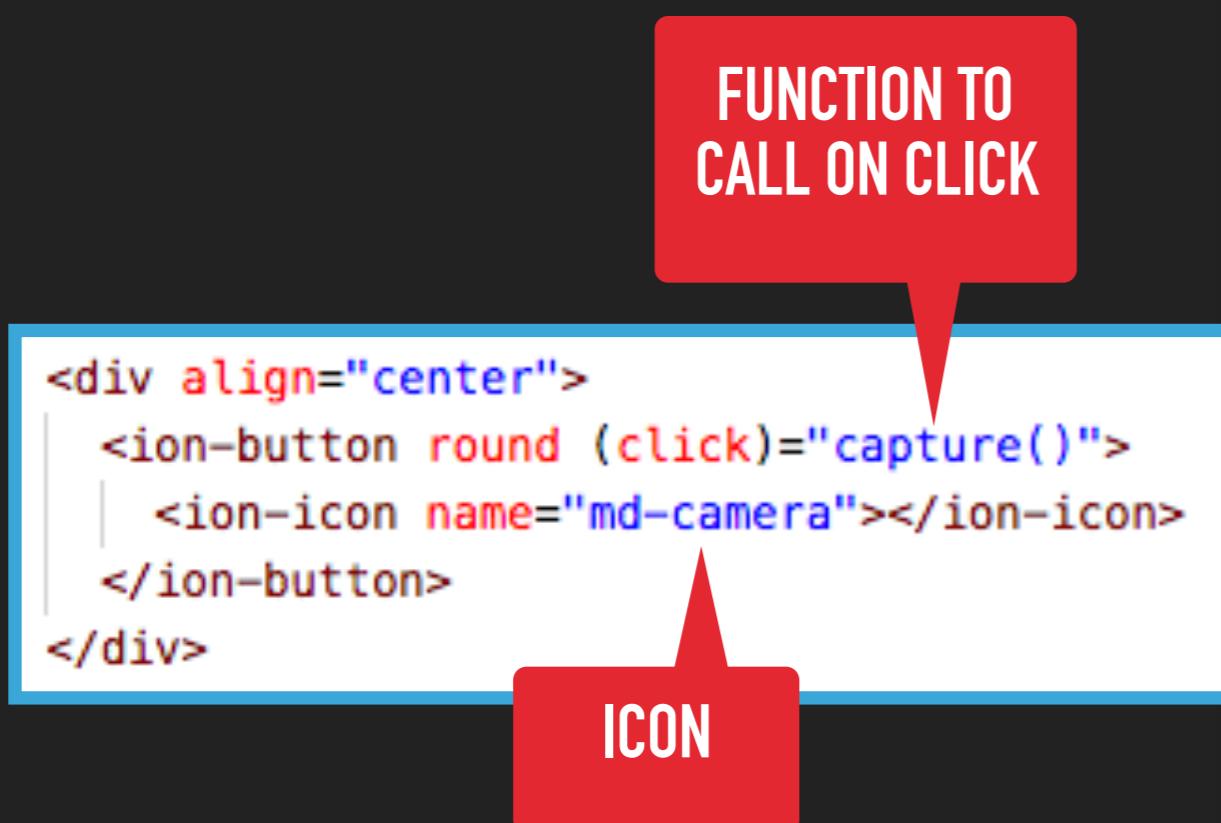


AFTER WE'RE LOGGED IN

- ▶ Our app is now running and ready to capture photos
- ▶ Place holder image is also a button



ADDING IN BUTTON ICONS



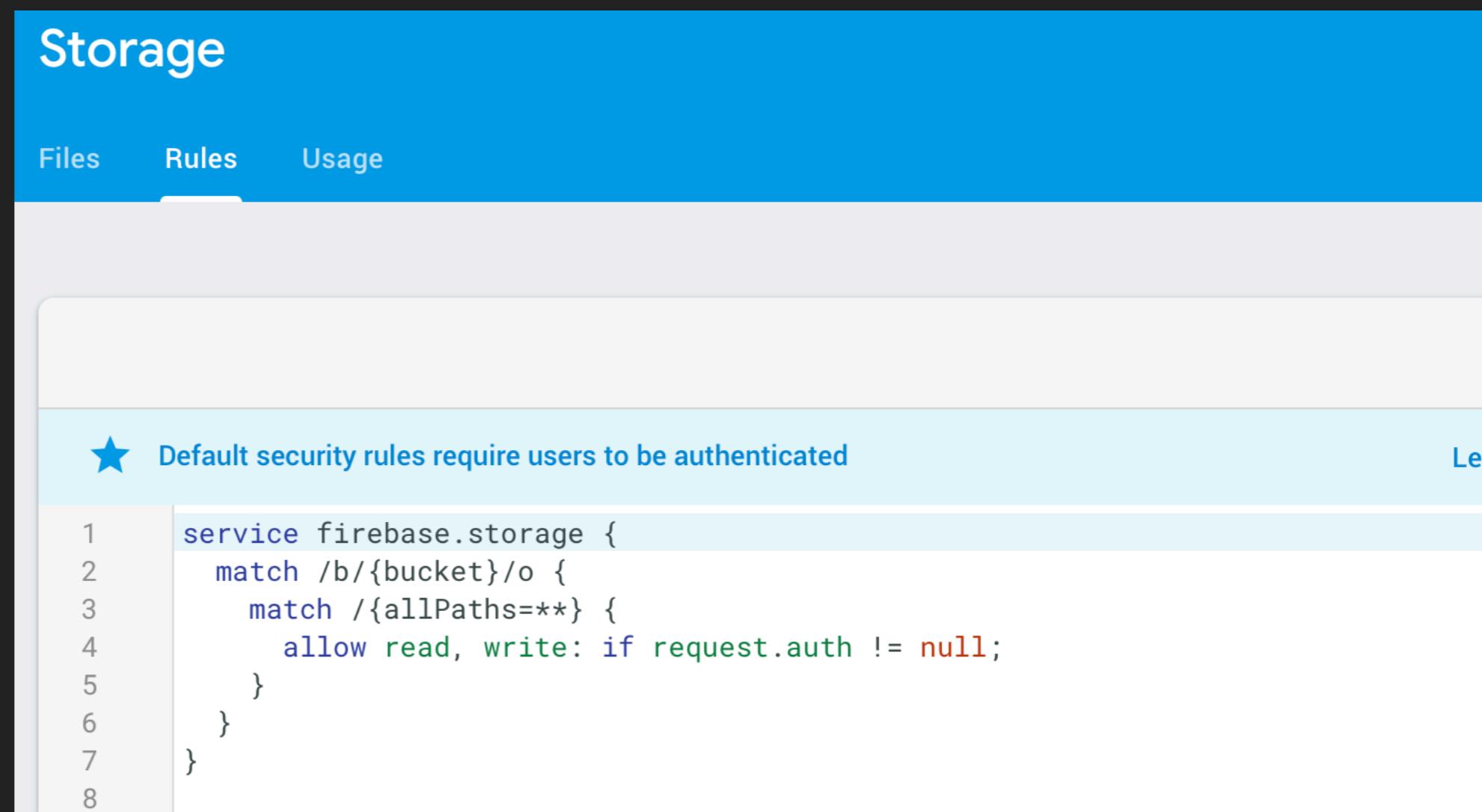
STEP 8: ADD PHOTO CAPTURE PAGE

- ▶ Create page
 - ionic g page PhotoCapture
- ▶ Copy the files from the photo-capture directory
- ▶ HTML5 takes care of connecting us to the camera (or to our photo library):

```
<div class="hidden_input">
| <input #inputcamera type="file" accept="image/*" />
</div>
```

STEP 8A: INITIALIZE THE FIREBASE STORAGE RESOURCE

- ▶ Setup storage in Firebase
- ▶ Confirm rules are setup properly



The screenshot shows the Firebase Storage Rules interface. The top navigation bar has tabs for 'Files', 'Rules', and 'Usage'. The 'Rules' tab is selected. Below the tabs, there is a note: '★ Default security rules require users to be authenticated'. The main area displays the following security rules:

```
1 service firebase.storage {  
2     match /b/{bucket}/o {  
3         match /{allPaths=**} {  
4             allow read, write: if request.auth != null;  
5         }  
6     }  
7 }  
8 }
```

On the far right edge of the screenshot, the letters 'Le' are partially visible.

STEP 9: ADD ANGULAR CROPPERJS

- ▶ Update photo-capture.module.ts

- ▶ Install fix-orientation package

```
npm i fix-orientation --save
```

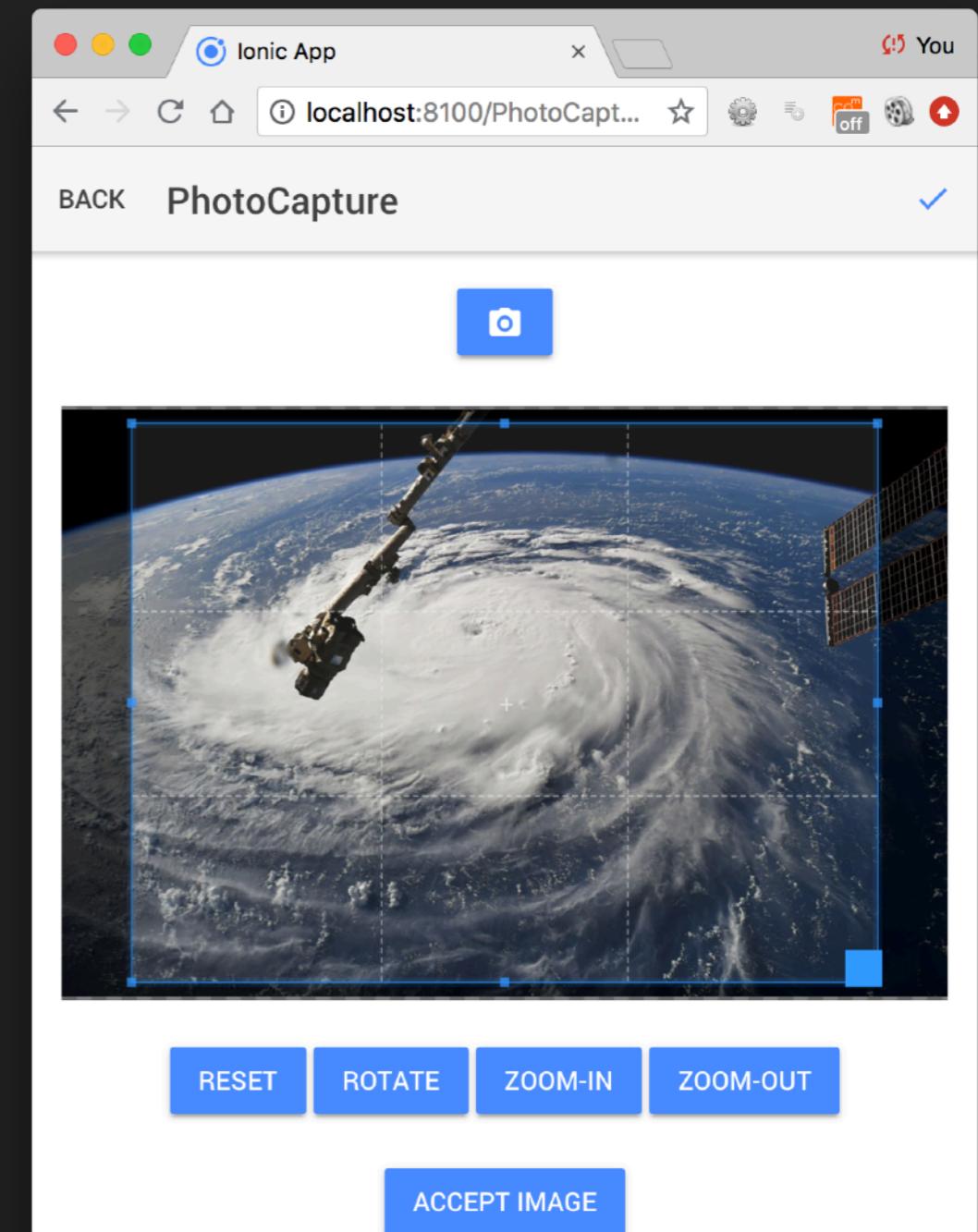
- ▶ Uncomment the following line in app.module.ts

```
import { AngularCropperjsModule } from  
'angular-cropperjs';
```

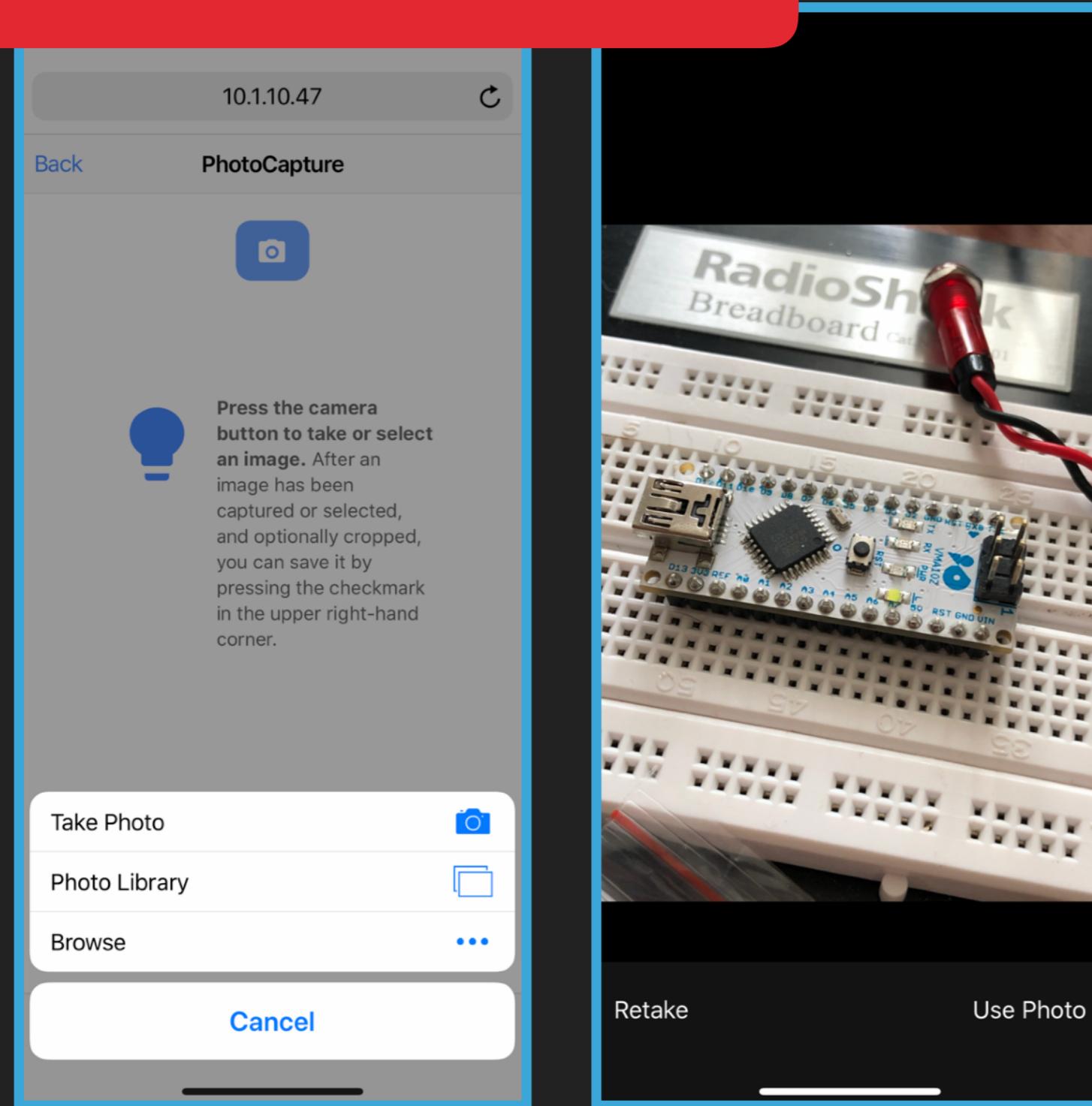
- ▶ Test app

LOADING AN IMAGE

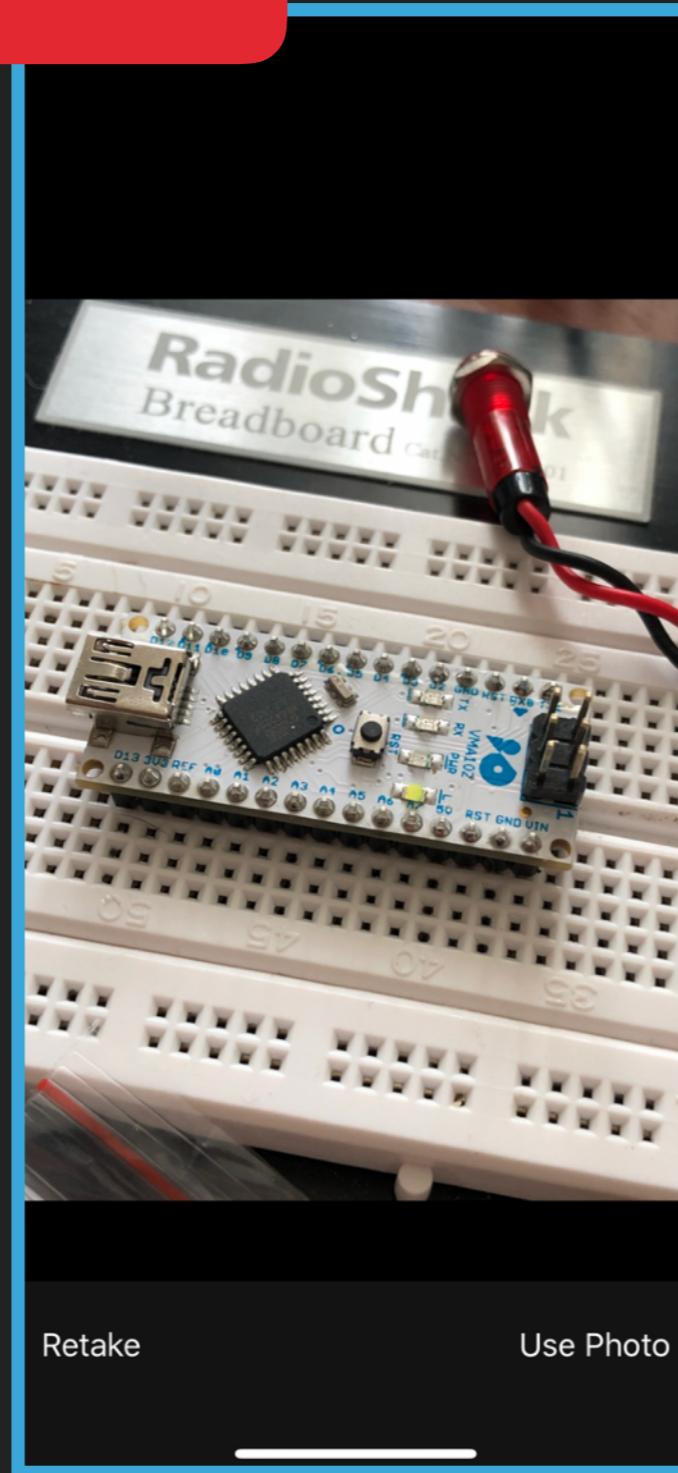
- ▶ Image is displayed with cropping control
- ▶ Press “Accept Image”



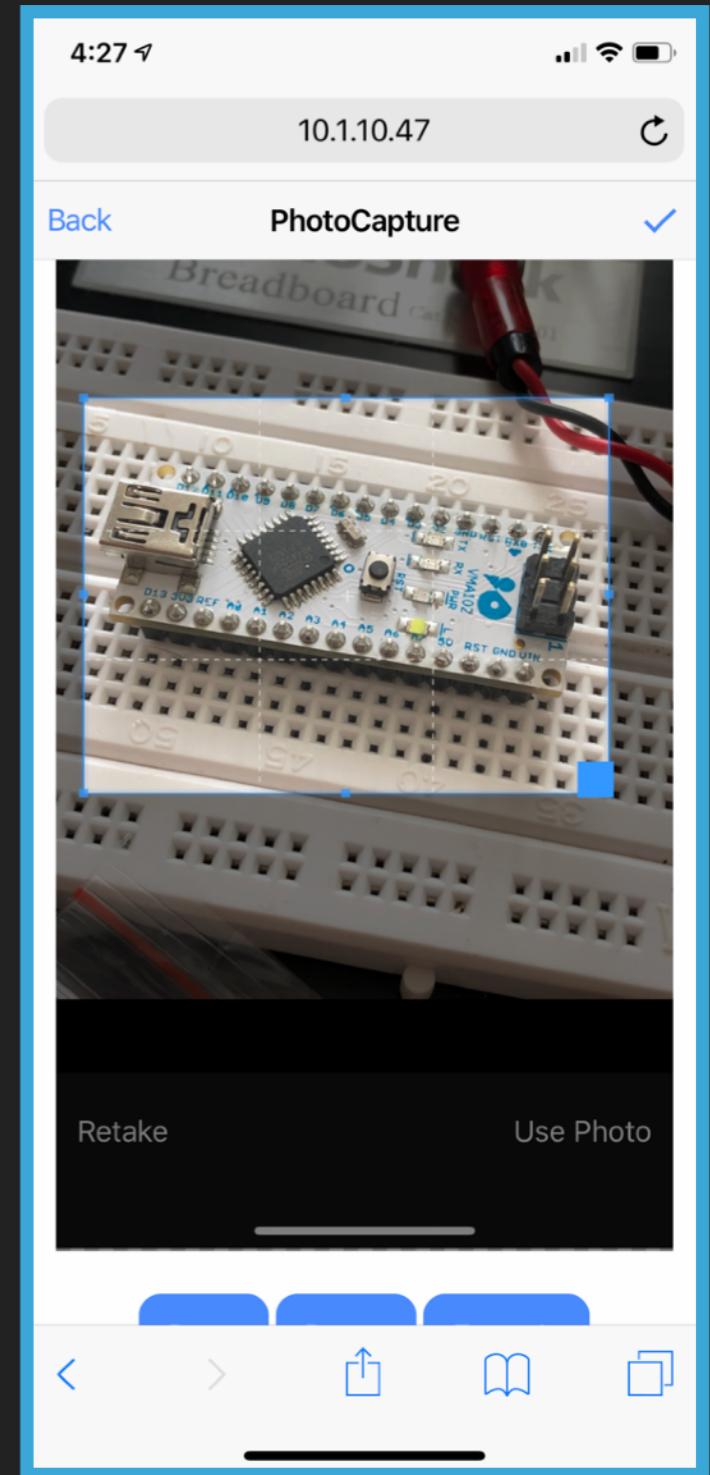
CAPTURING AN IMAGE ON AN IPHONE



1



2



3

STEP 10: INSTALL FIREBASE CLI

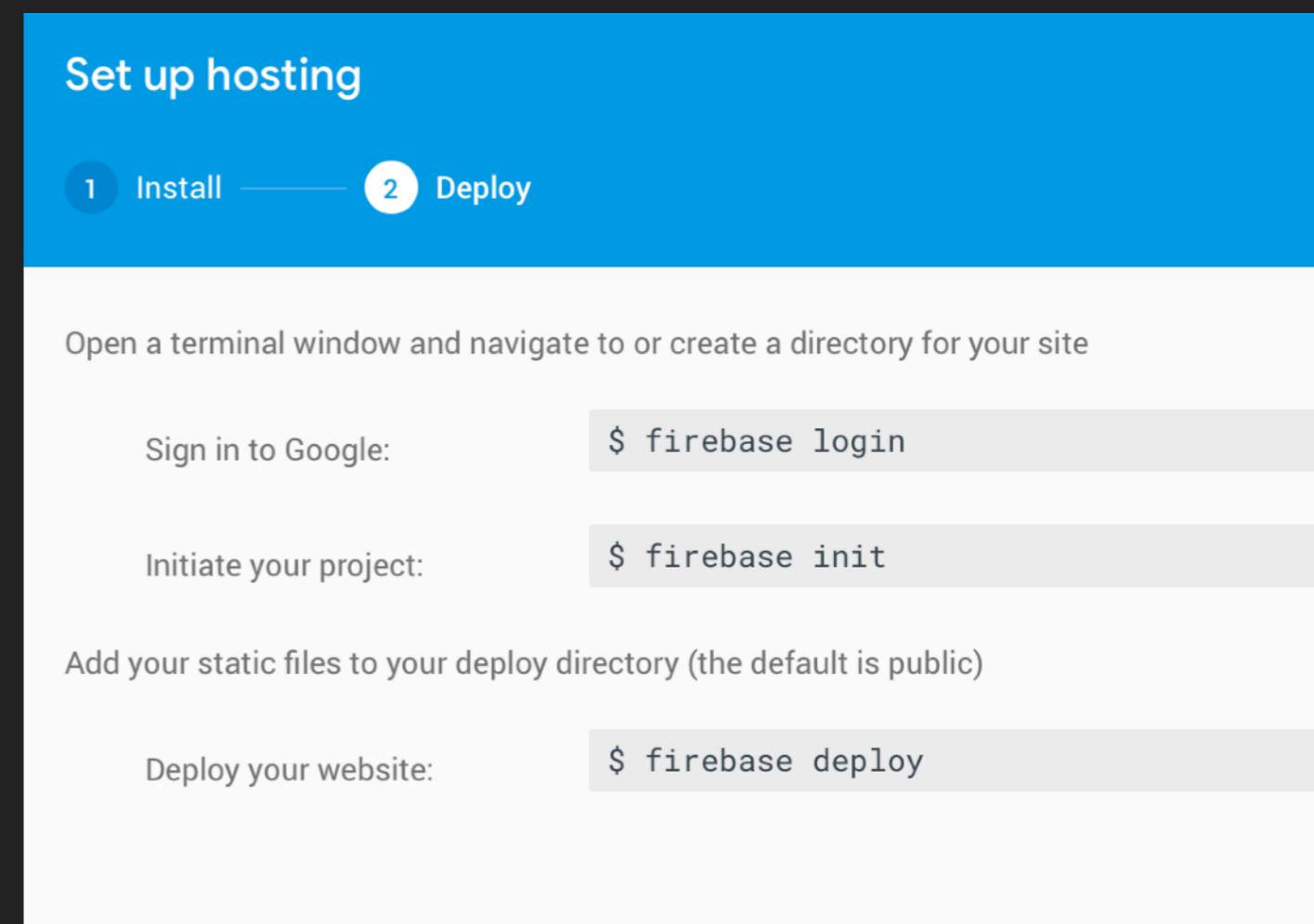
- ▶ Install the Firebase tools:

```
npm install -g firebase-tools
```

- ▶ Setup your project for hosting

```
firebase login
```

```
firebase init
```



STEP 11: DEPLOY APP AND TEST

- ▶ Build your app
`npm run build`
- ▶ Deploy to Firebase
`firebase deploy`
- ▶ Test the app in your web browser
<https://photograb-14821.firebaseio.com>

WHAT TO EXPLORE NEXT...

- ▶ Explore Firebase's database capabilities
- ▶ Explore Amazon's AWS resources and how they can be integrated into Ionic apps
- ▶ Explore the Ionic platform in greater detail

YOU CAN REACH ME AT:

davidgeller@gmail.com

github.com/davidgeller/seattlecodecamp

