Steps to create the Starter Kit

Make sure the below softwares are already installed in your system and your using open network without any restriction (good bandwidth).

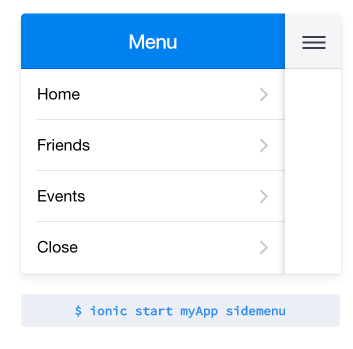
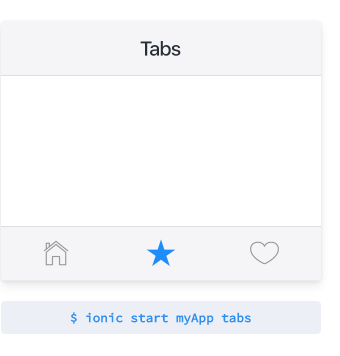
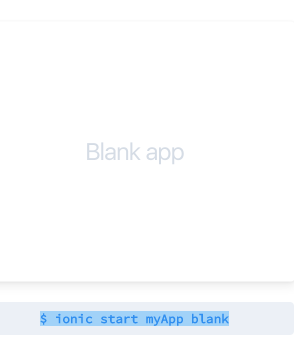
* **Node Js**
* **Java version Sdk 8**
* **Cordova and ionic**
* **Visual Studio code**
* **Android Studio 3.0**
* **Chrome browser**

1. **Start creating an application**

Use the command below to create a new mobile application.

$ ionic start myApp blank

( myApp is the name of your application and ‘blank’ is a template used)



1. **RUN Your Application**

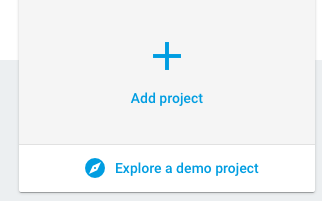
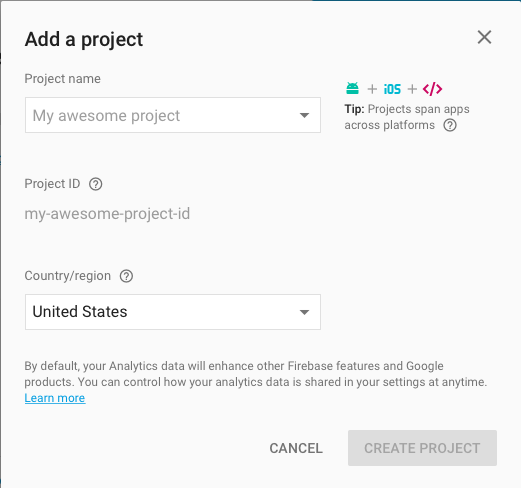
**To run your application, use the below command.**

*cd myApp*

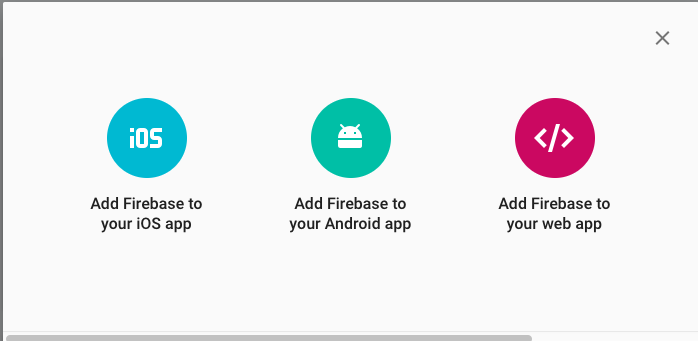
*ionic serve*

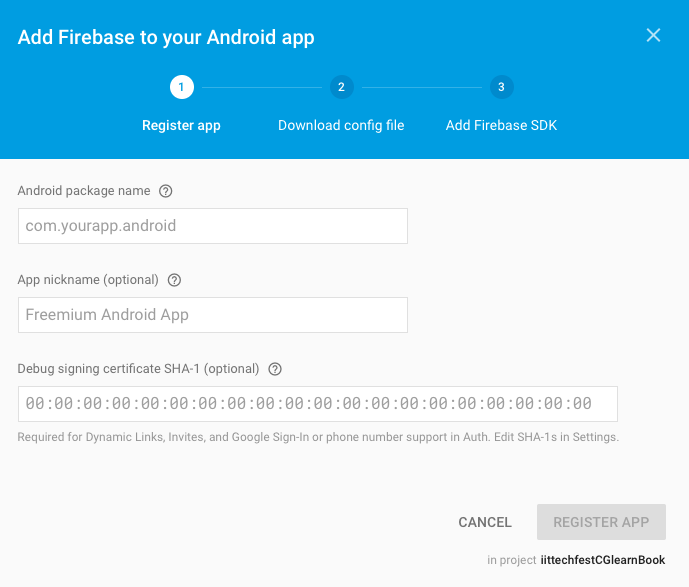
{Now your blank project is successfully created}

1. **Open firebase base console** [**https://console.firebase.google.com**](https://console.firebase.google.com) **from valid gmail account . Click on “Add Project “ create project , project id ,select country and click on Create project.**

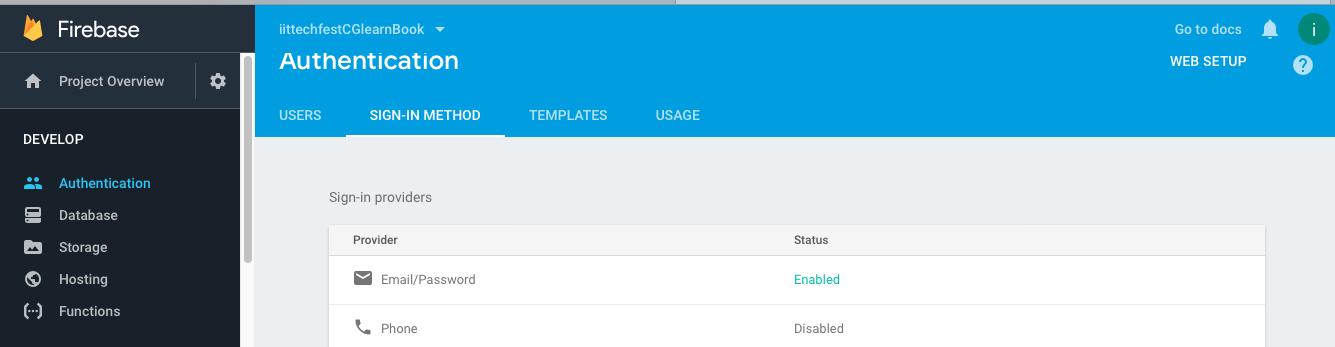
1. **After creating select “add app” Select “ Add Firebase to your Android app ” for android platform. Similarly for iOS and web app platform select respective options.**





1. **Enable Sign-in Authentication :**

Open your Firebase console account click on “Authentication” click “Sign in Method “ Enable the “Email/Password” Authentication



In Step 1:

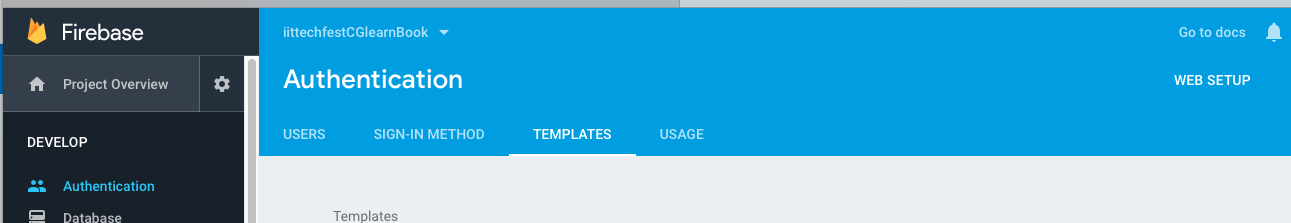
Add the package name (mandatory) other are optional then click on Register APP

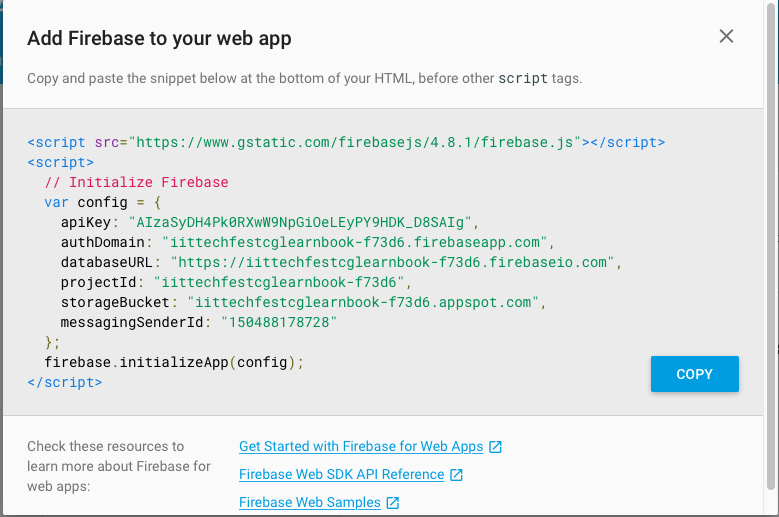
Step 2:

Download the google-service-json file and paste it in *your {project folder path }>platform > Android folder*

Step 3:

Go to google firebase console account and click Authentication followed by Web Setup and Copy the code





1. **Paste the following code App > app.component.ts**

Mention the code inside the

“constructor (

) {

}”

this is how it should look like at the end

constructor(private platform: Platform, private statusBar: StatusBar, private splashScreen: SplashScreen) {

// Initialize Firebase

var config = {

apiKey: "AIzaSyDH4Pk0RXwW9NpGiOeLEyPY9HDK\_D8SAIg",

authDomain: "iittechfestcglearnbook-f73d6.firebaseapp.com",

databaseURL: "https://iittechfestcglearnbook-f73d6.firebaseio.com",

projectId: "iittechfestcglearnbook-f73d6",

storageBucket: "iittechfestcglearnbook-f73d6.appspot.com",

messagingSenderId: "150488178728"

};

firebase.initializeApp(config);

}

ngAfterViewInit() {

}

1. **To add plugins to the project, use the below command**

ionic cordova plugin add < plugin name >

plugin name :

*cordova-plugin-browsertab*

*cordova-plugin-buildinfo*

*cordova-plugin-compat*

*cordova-plugin-console*

*cordova-plugin-customeurlscheme*

*cordova-plugin-device*

*cordova-plugin-firebase-analytics*

*cordova-plugin-inappbrowser*

*cordova-plugin-ionic-webview*

*cordova-plugin-splashscreen*

*cordova-plugin-statusbar*

*cordova-plugin-whitelist*

*cordova-support-google-services*

*cordova-universal-links-plugin*

*ionic-plugin-keyboard*

Your Project files should have the following codes :

1. **App > app.component.ts**

import { Component, ViewChild } from '@angular/core';

import { Platform } from 'ionic-angular';

import { StatusBar } from '@ionic-native/status-bar';

import { SplashScreen } from '@ionic-native/splash-screen';

import { LoginPage } from '../pages/login/login';

import \* as firebase from 'firebase';

import { FirebaseAnalytics } from '@ionic-native/firebase-analytics';

@Component({

templateUrl: 'app.html'

})

export class MyApp {

@ViewChild('myNav') nav

rootPage:any;

firstRun: boolean = true;

constructor(private platform: Platform, private statusBar: StatusBar, private splashScreen: SplashScreen) {

// Initialize Firebase

var config = {

apiKey: "AIzaSyDH4Pk0RXwW9NpGiOeLEyPY9HDK\_D8SAIg",

authDomain: "iittechfestcglearnbook-f73d6.firebaseapp.com",

databaseURL: "https://iittechfestcglearnbook-f73d6.firebaseio.com",

projectId: "iittechfestcglearnbook-f73d6",

storageBucket: "iittechfestcglearnbook-f73d6.appspot.com",

messagingSenderId: "150488178728"

};

firebase.initializeApp(config);

}

ngAfterViewInit() {

firebase.auth().onAuthStateChanged((user) => {

if (user) {

// User is authenticated.

this.setRootPage('HomePage');

} else {

// User is not authenticated.

this.setRootPage(LoginPage);

}

});

}

setRootPage(page) {

if (this.firstRun) {

// if its the first run we also have to hide the splash screen

this.nav.setRoot(page)

.then(() => this.platform.ready())

.then(() => {

// Okay, so the platform is ready and our plugins are available.

// Here you can do any higher level native things you might need.

this.statusBar.styleDefault();

this.splashScreen.hide();

this.firstRun = false;

});

} else {

this.nav.setRoot(page);

}

}

}

1. **App > app.html**

<ion-nav #myNav [root]="rootPage"></ion-nav>

1. **App > app.module.ts**

import { BrowserModule } from '@angular/platform-browser';

import { ErrorHandler, NgModule } from '@angular/core';

import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';

import { SplashScreen } from '@ionic-native/splash-screen';

import { StatusBar } from '@ionic-native/status-bar';

import { MyApp } from './app.component';

import { LoginPage } from '../pages/login/login';

import { FirebaseuiProvider } from '../providers/firebaseui/firebaseui';

import { FirebaseAnalytics } from '@ionic-native/firebase-analytics';

@NgModule({

declarations: [

MyApp,

LoginPage

],

imports: [

BrowserModule,

IonicModule.forRoot(MyApp)

],

bootstrap: [IonicApp],

entryComponents: [

MyApp,

LoginPage

],

providers: [

StatusBar,

SplashScreen,

{provide: ErrorHandler, useClass: IonicErrorHandler},

FirebaseuiProvider,

FirebaseAnalytics

]

})

export class AppModule {}

1. **App > main.ts**

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app.module';

platformBrowserDynamic().bootstrapModule(AppModule);

1. **Home > home.html**

<!--

Generated template for the HomePage page.

See http://ionicframework.com/docs/components/#navigation for more info on

Ionic pages and navigation.

-->

<ion-header>

<ion-navbar>

<ion-title>Home</ion-title>

</ion-navbar>

</ion-header>

<ion-content padding>

<h2>Welcome!</h2>

<button ion-button (click)="logout()">

Log out

</button>

</ion-content>

1. **Home >** [**home.module.ts**](http://home.module.ts)

import { NgModule } from '@angular/core';

import { IonicPageModule } from 'ionic-angular';

import { HomePage } from './home';

@NgModule({

declarations: [

HomePage,

],

imports: [a

IonicPageModule.forChild(HomePage),

],

exports: [

HomePage

]

})

export class HomePageModule {}

1. **Home** **> home.scss**

page-home {

}

1. **Home > home.ts**

import { Component } from '@angular/core';

import { IonicPage, NavController, NavParams } from 'ionic-angular';

import \* as firebase from 'firebase';

/\*\*

\* Generated class for the HomePage page.

\*

\* See http://ionicframework.com/docs/components/#navigation for more info

\* on Ionic pages and navigation.

\*/

@IonicPage()

@Component({

selector: 'page-home',

templateUrl: 'home.html',

})

export class HomePage {

constructor(public navCtrl: NavController, public navParams: NavParams) {

}

ionViewDidLoad() {

console.log('ionViewDidLoad HomePage');

}

logout() {

firebase.auth().signOut();

}

}

1. **Login > login.html**

<ion-header>

<ion-navbar>

<ion-title>

Login

</ion-title>

</ion-navbar>

</ion-header>

<ion-content padding>

<div id="firebaseui-auth-container"></div>

</ion-content>

1. **Login > login.sccss**

page-login {

}

1. **Login > login.ts**

import { Component } from '@angular/core';

import { NavController } from 'ionic-angular';

import { FirebaseuiProvider } from '../../providers/firebaseui/firebaseui';

import \* as firebase from 'firebase';

import \* as firebaseui from 'firebaseui';

@Component({

selector: 'page-login',

templateUrl: 'login.html'

})

export class LoginPage {

constructor(public navCtrl: NavController, public uiProvider: FirebaseuiProvider) {

}

ionViewDidLoad() {

console.log('ionViewDidLoad LoginPage')

// The start method will wait until the DOM is loaded.

this.uiProvider.ui.start('#firebaseui-auth-container', this.getUiConfig());

}

getUiConfig() {

// FirebaseUI config.

return {

callbacks: {

signInSuccess: (currentUser, credential, redirectUrl) => {

// Do something.

// Return type determines whether we continue the redirect automatically

// or whether we leave that to developer to handle.

return false;

}

},

credentialHelper: firebaseui.auth.CredentialHelper.NONE,

signInOptions: [

// Leave the lines as is for the providers you want to offer your users.

{

provider: firebase.auth.GoogleAuthProvider.PROVIDER\_ID,

customParameters: {

// Forces account selection even when one account

// is available.

prompt: 'select\_account'

}

},

//firebase.auth.FacebookAuthProvider.PROVIDER\_ID,

//firebase.auth.TwitterAuthProvider.PROVIDER\_ID,

//firebase.auth.GithubAuthProvider.PROVIDER\_ID,

//firebase.auth.EmailAuthProvider.PROVIDER\_ID,

// firebase.auth.PhoneAuthProvider.PROVIDER\_ID // not available for Ionic apps

],

// Terms of service url.

tosUrl: '<your-tos-url>'

};

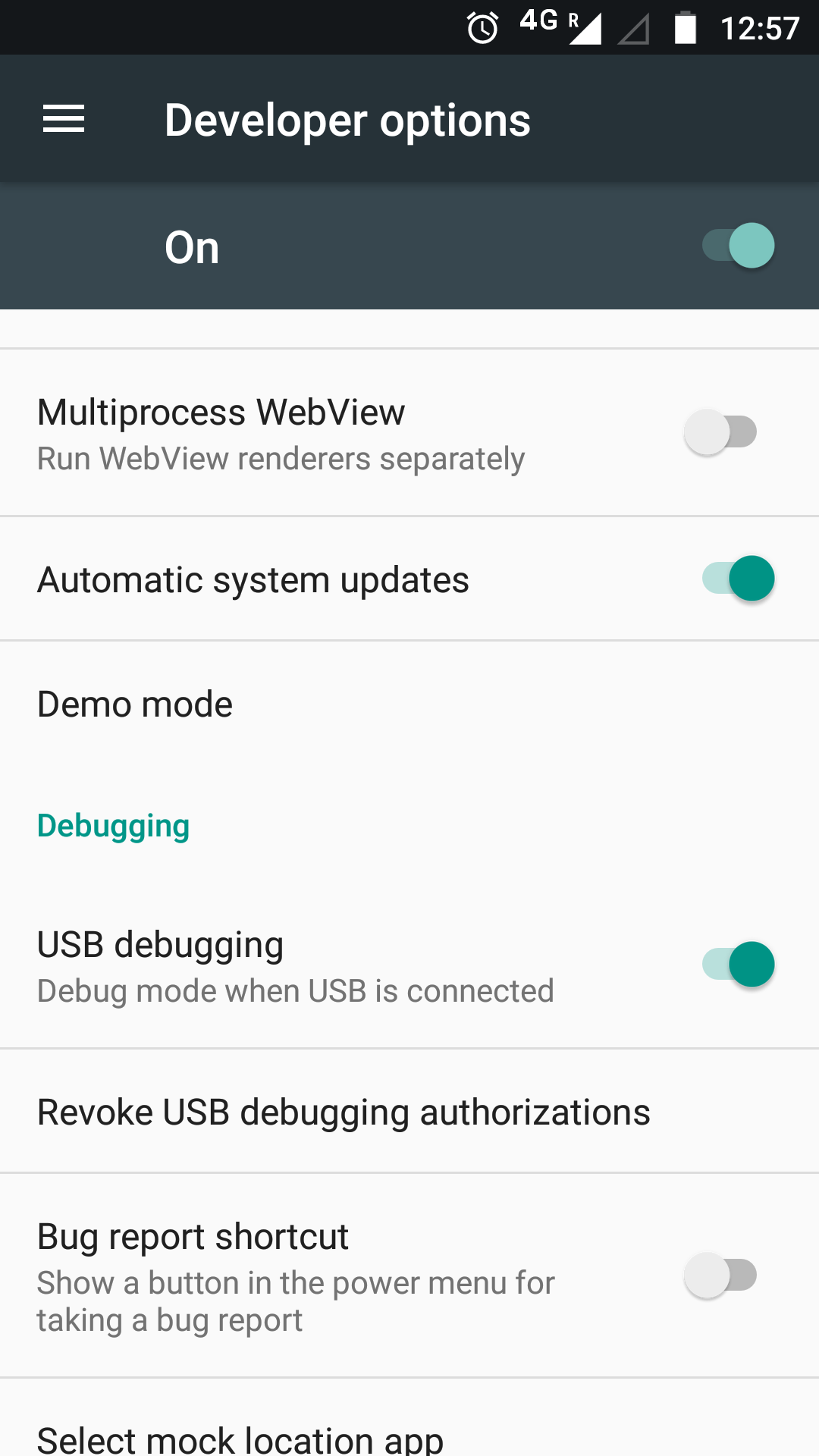
}

}

1. **Add Android Platform to your project by adding the command “ionic cordova platform add android” {wait till the command executes completely}**

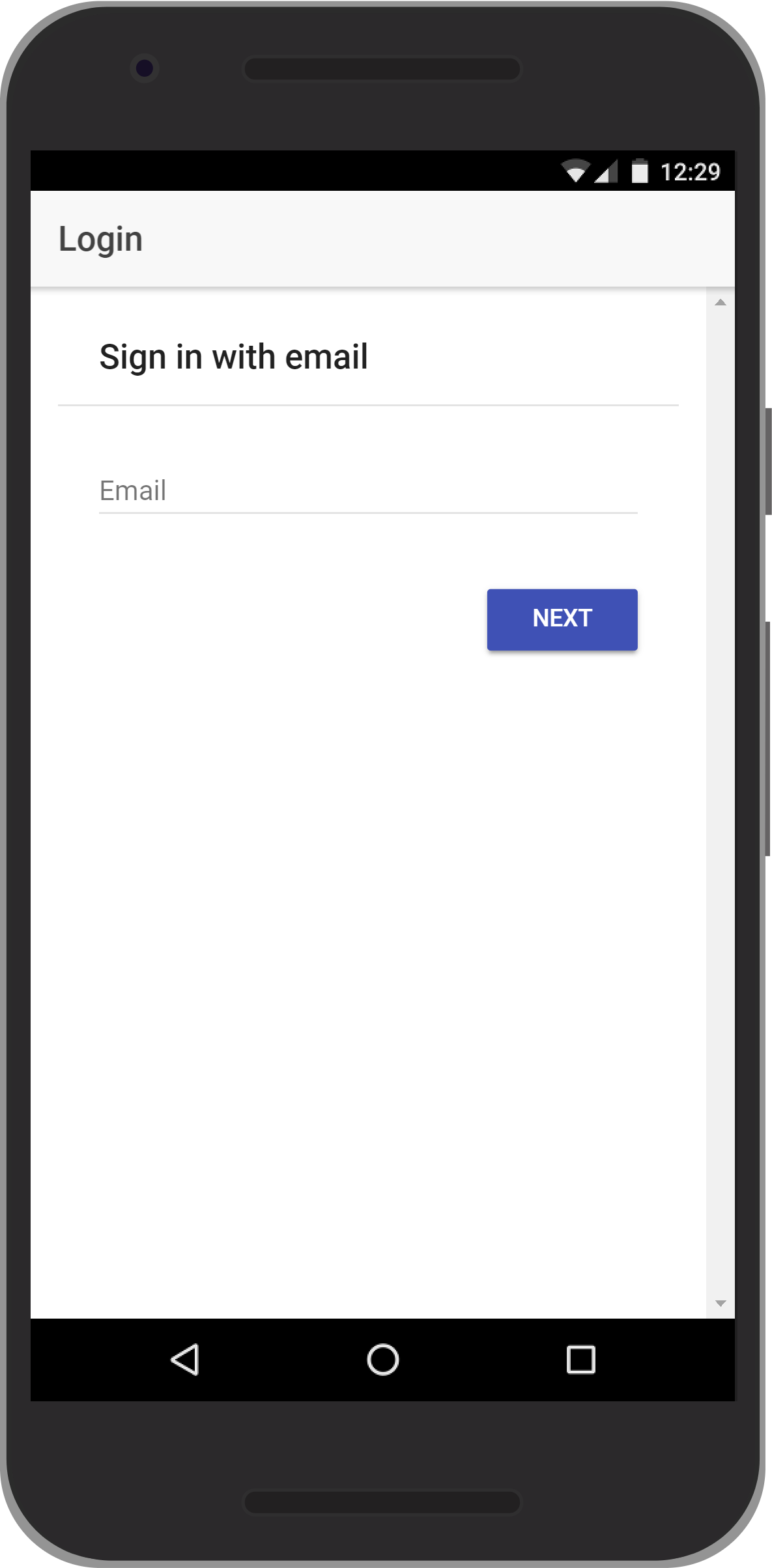


Make sure your android device is connected to your system (also the usb debug option of your android device should be enabled). You can do that by going to the “Developer Option” under “Settings”. Else run step 7 directly



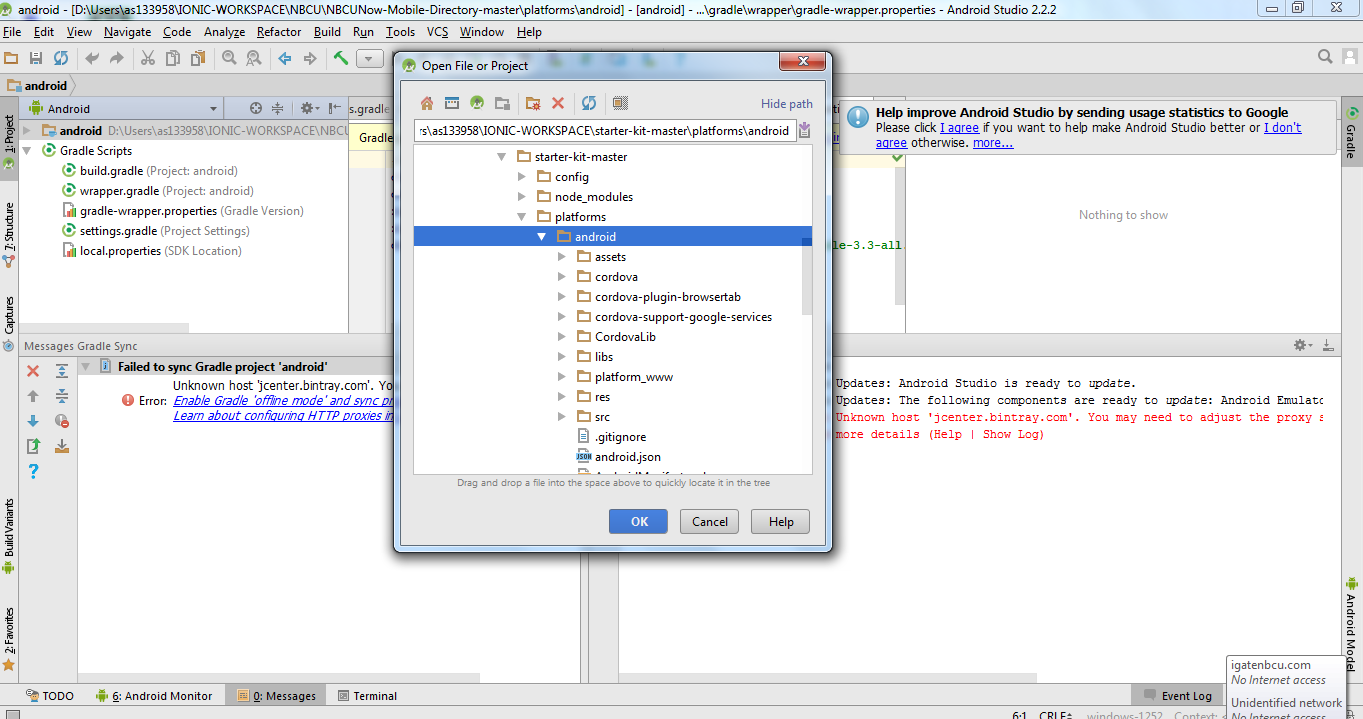
1. **Add command “ionic cordova build android”; after the build is successful add command “ionic cordova run android” (make sure the android device is already connected to your system) check the output. If android device is not connected then got to below path <project folder> /platforms / android / builds / outputs /apk/ debug and copy the android-debug.apk file and paste it manually in your device.**

If the output is “Success” the app will run in your device shortly, and you will be able to see an output like below.



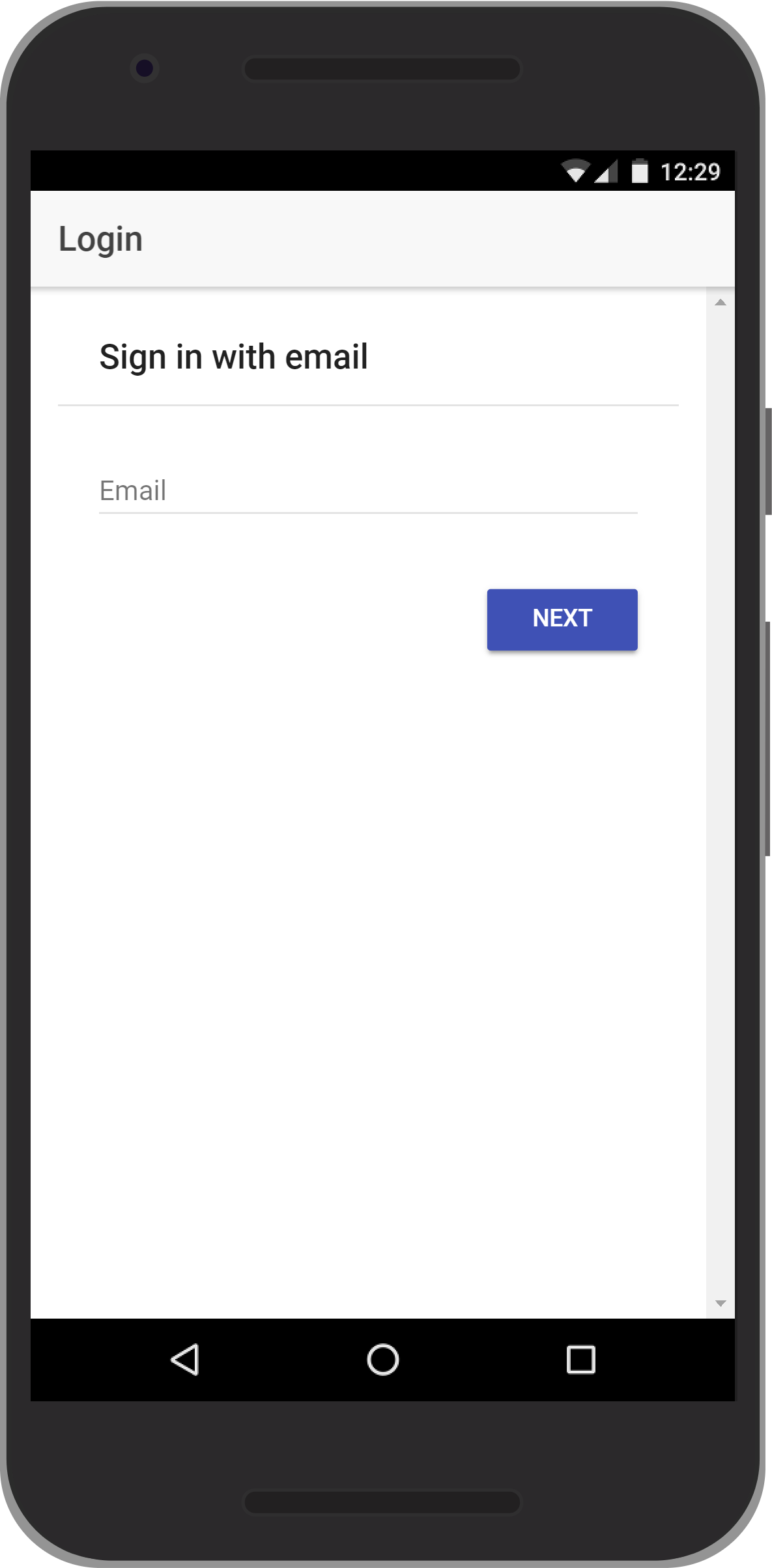
Else need to solve the issue.

There is also an option of opening and running the app in Android Studio. Open Android Studio. Click on “file Open” and select the below path: << Project folder/ platforms /android >>



Once open click on the green play icon in the Android studio” (make sure the android device is already connected to your system) (attach screenshot of showing mobile run)

Display the final output page.



**About Capgemini**

**A global leader in consulting, technology services and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients’ opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2016 global revenues of EUR 12.5 billion.**

**Learn more about us at** [**www.capgemini.com**](http://www.capgemini.com)

This message contains information that may be privileged or confidential and is the property of the Capgemini Group.  
Copyright © 2017 Capgemini. All rights reserved.