

Adding Firebase Authentication to an Ionic 2 Mobile App

DS '17 Hands-On Cloud Lab

Venkatesh Voona

Lead Researcher (MIL)- Mobile/NextGen
Miracle Software Systems, Inc.

December 9th, 2017

Adding Firebase Authentication to an Ionic 2 Mobile App

Goal

In this lab we will guide you how to add firebase authentication to an Ionic 2 Mobile App. here you are going to create a mobile app with signup and signin screens and apply authentication by using firebase.

Pre-Requisites

The following installations will need to be completed for this lab to be run successfully,

- Google Account
- Firebase Plugin to be Installed
- Node JS and npm installed
- Cordova
- Ionic Framework
- Text editor (Visual Studio Code is preferable)

Technology Involved

- Ionic Framework
- Cordova
- HTML / CSS / Javascript
- Typescript
- Angular 2/4

Labs Steps

Let's get started with the lab!

#1 | Create a new Ionic project

To create/start a new project in Ionic 2 framework, we need to give a below command.

`ionic start <application name> <starter template name>`

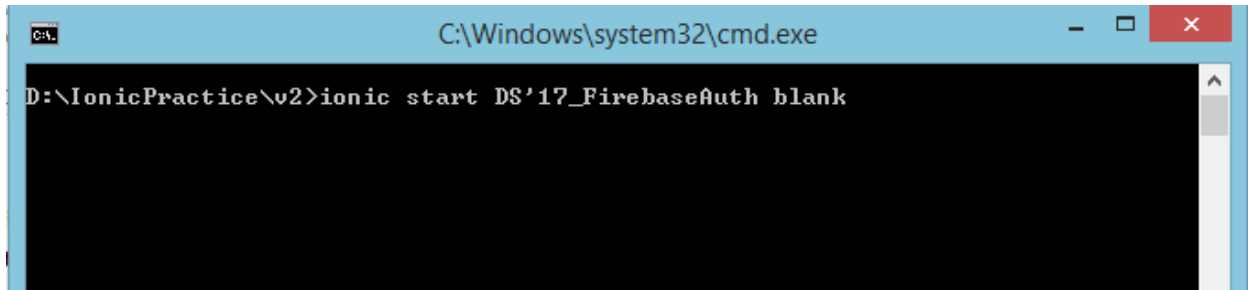
We will be having following starter templates in ionic 2 framework,

tabs	A starting project with a simple tabbed interface
blank	A blank starter project
Sidemenu	A starting project with a side menu with navigation in the content area
Super	A starting project complete with pre-built pages, providers and best practices for Ionic development
Conference	A project that demonstrates a realworld application
Tutorial	A tutorial based project that goes along with the Ionic documentation
Aws	AWS Mobile Hub Starter

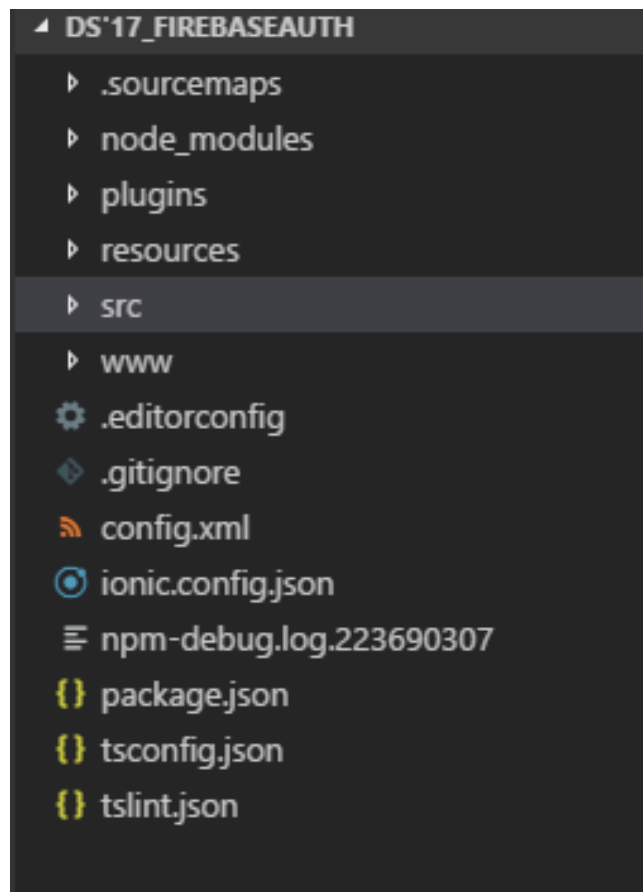
In this lab we are going to use blank template and start implementing the actual application.

Eg : `ionic start DS17_FirebaseAuthentication blank`

Give the above command in the command prompt as shown in the below figure.



Once you give the above command, the ionic application will be created with the following folder structure as shown in the figure.

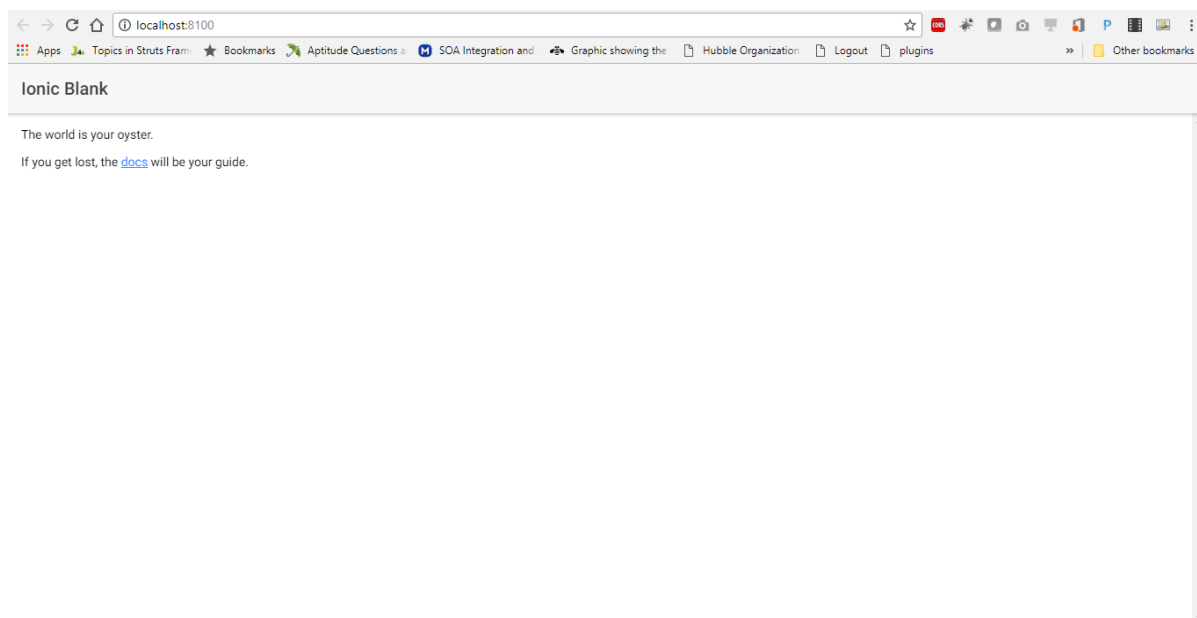


#2 | Running the application

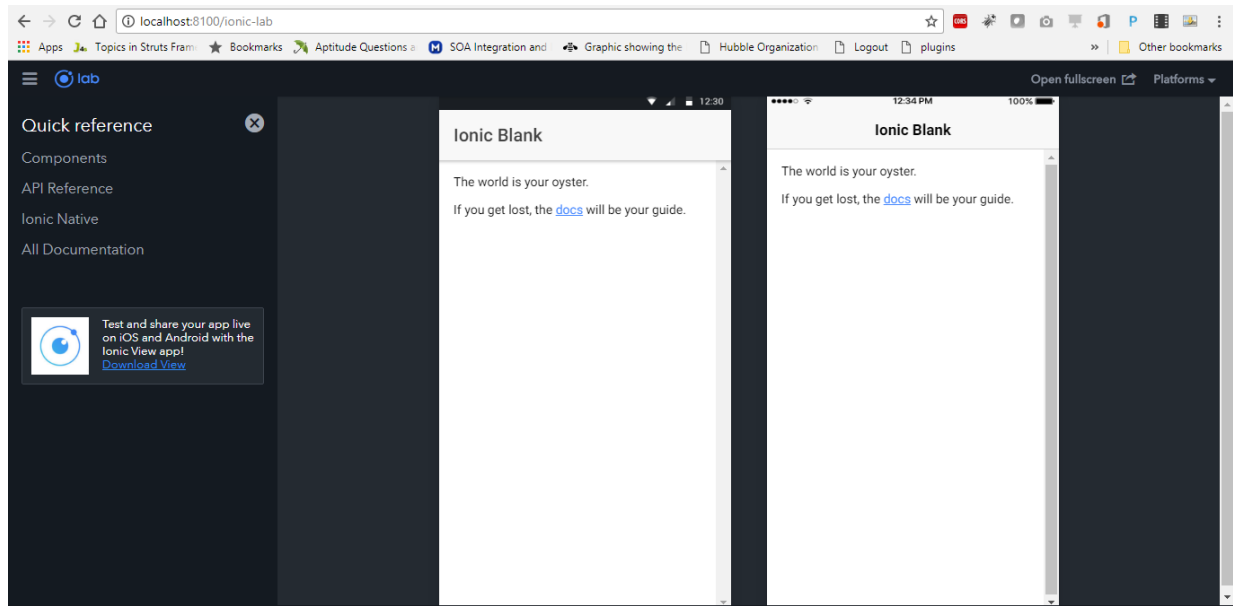
Before going to give run command, we need to route to the current application folder and then give the following commands.

Run on Browser	-	ionic serve
Run on Mobile simulator	-	ionic serve --lab

If you run the application on browser, you can see the output as shown in the below figure.

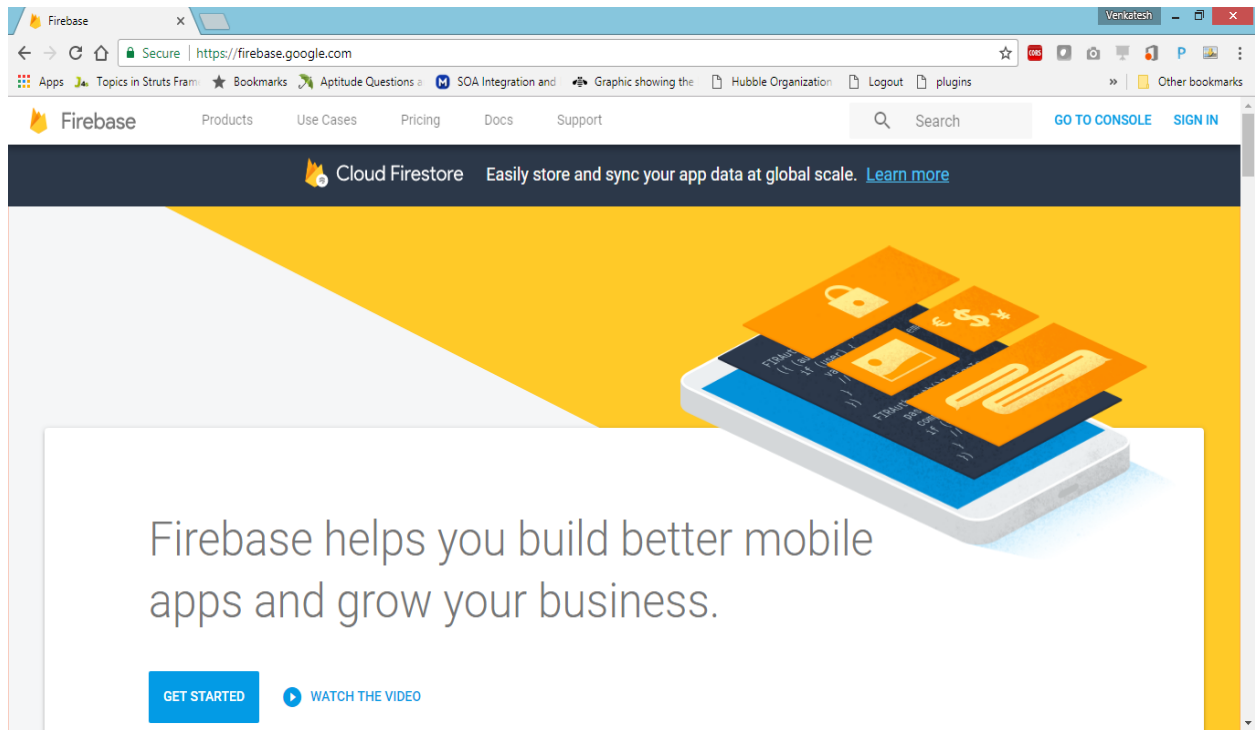


If you run the application on mobile browser simulator , you can see the output as shown in the below figure. Here if you observe you can see the output in two different platforms one is Android and another one is IOS.

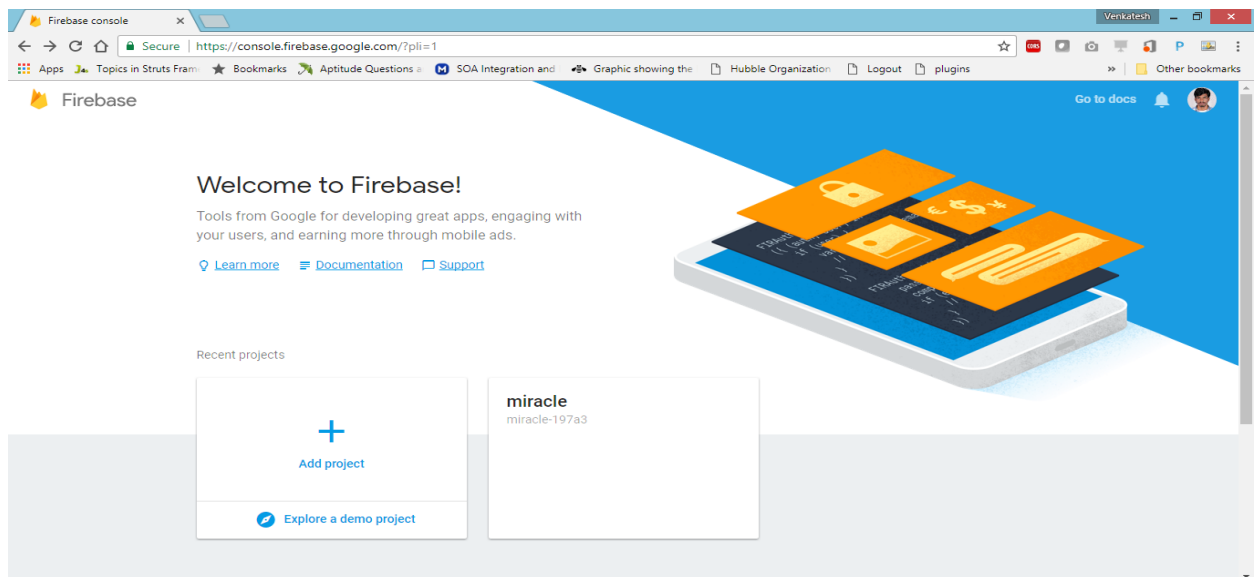


#3 | Create a project in Firebase Console

To work with firebase authentication, you need to create a project in the firebase console. Go to this link <https://firebase.google.com/> you will be get the following page as shown in the figure.

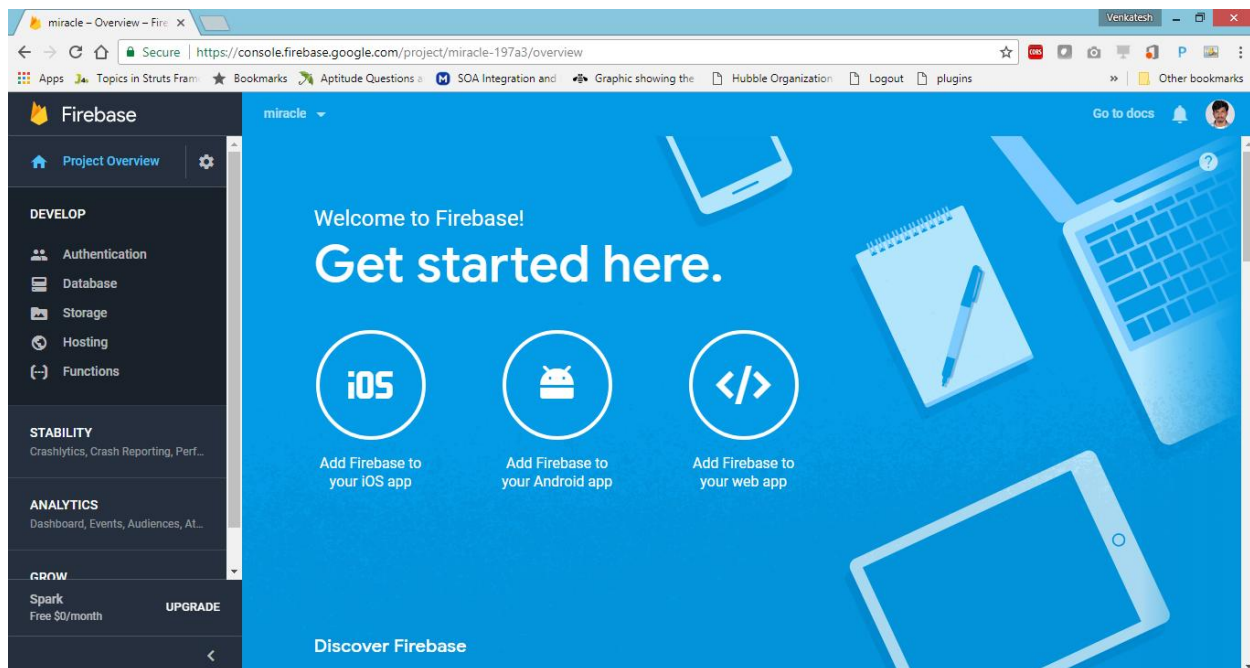


Click on **Go to Console** button which is on top right corner of the above figure then you will get sign-in prompt if you are not logged in. Log on with your gmail credentials to get into the firebase console. Below is the dashboard of firebase console.



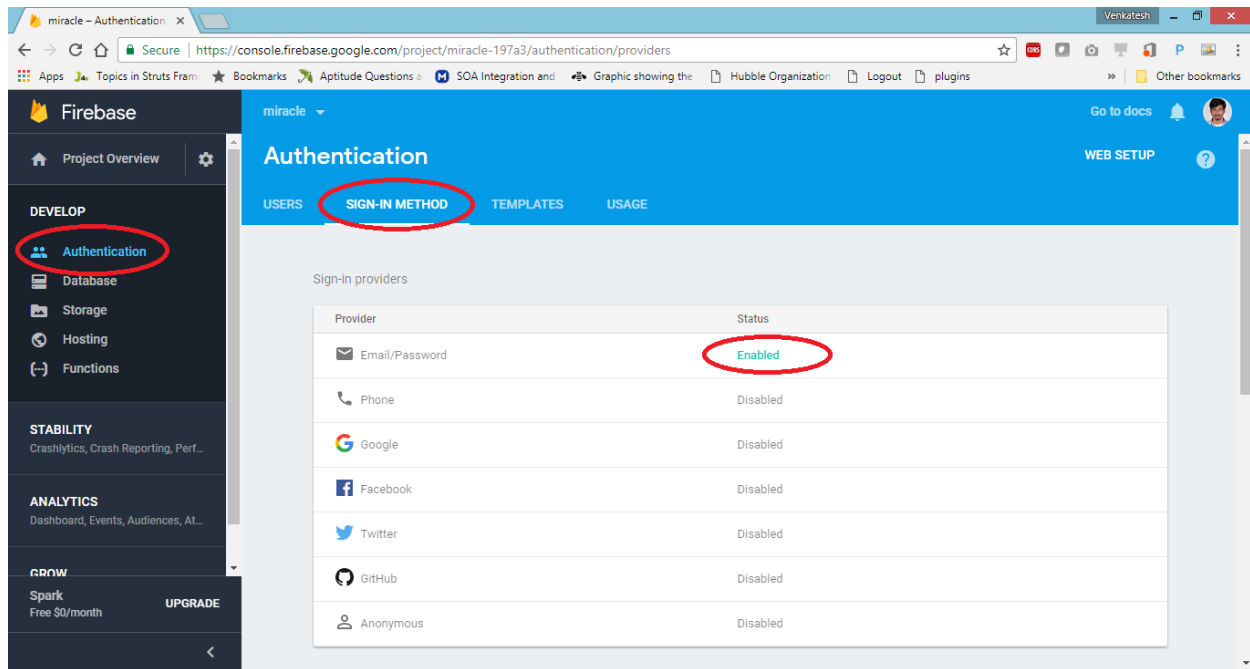
In order to work on firebase authentication, you need create a new project or you can add firebase authentication to the existing project. To add new project, click on Add project tile which is on above figure.

Once you create the project, new tile will create with your project, just click on that tile, it will redirects to your project dashboard.



#4 | Enable Sign-In Method

In firebase, you can select which sign-in method that is required for you. In this scenario, you are going to choose Email/Password method. To work on Email/Password, you need enable it. As shown in the below figure.

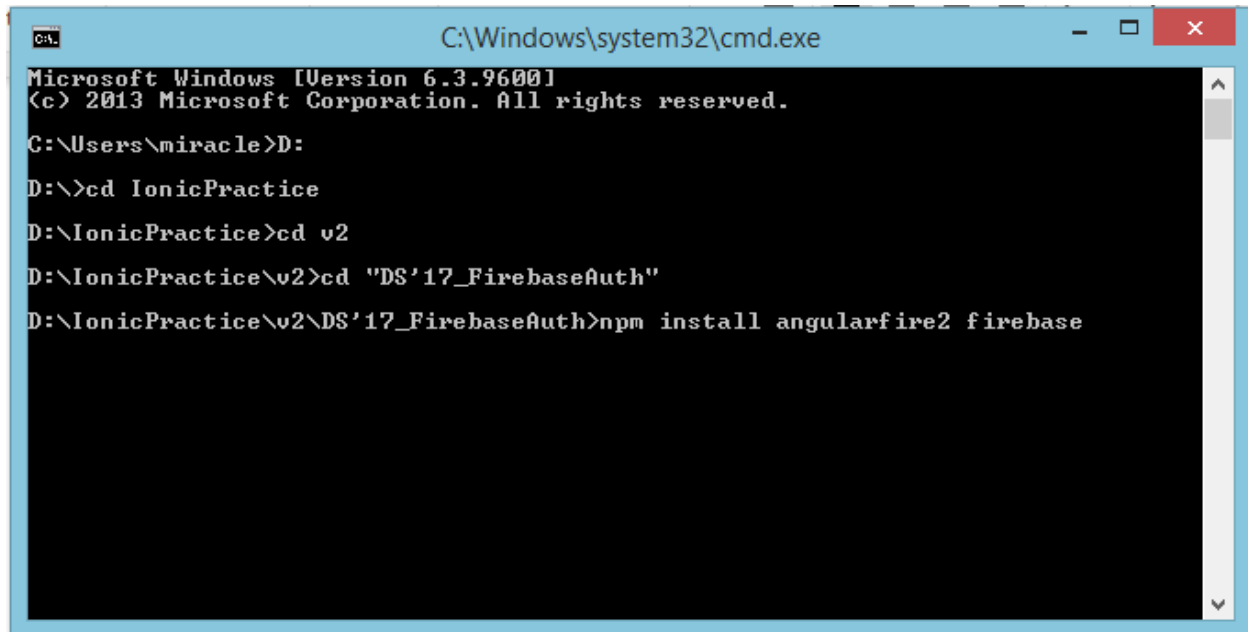


As you can observe in the figure, click on Authentication tab on the left and then select Sign- In Method tab on the right panel. Now you can enable your option called Email/Password.

#5 | Install Firebase plugin

In order work with firebase, you need to have the angularfire2 plugin installed in your ionic application. Change the director to where your ionic application has been created and then give the following command,

npm install angularfire2 firebase



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\miracle>D:
D:\>cd IonicPractice
D:\IonicPractice>cd v2
D:\IonicPractice\v2>cd "DS'17_FirebaseAuth"
D:\IonicPractice\v2\DS'17_FirebaseAuth>npm install angularfire2 firebase
```

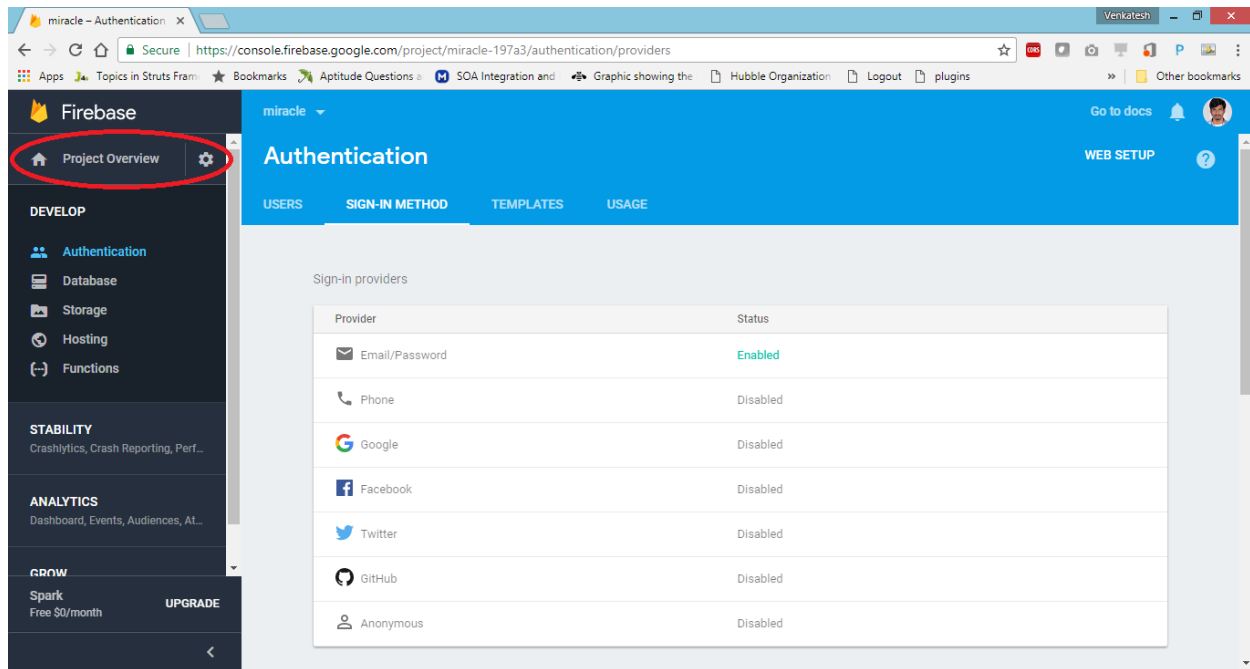
#6 | Import and Initialize

After installation of angularfire2 firebase plugin, you need to import required modules, here you should add the following statements in app.module.ts file to import modules.

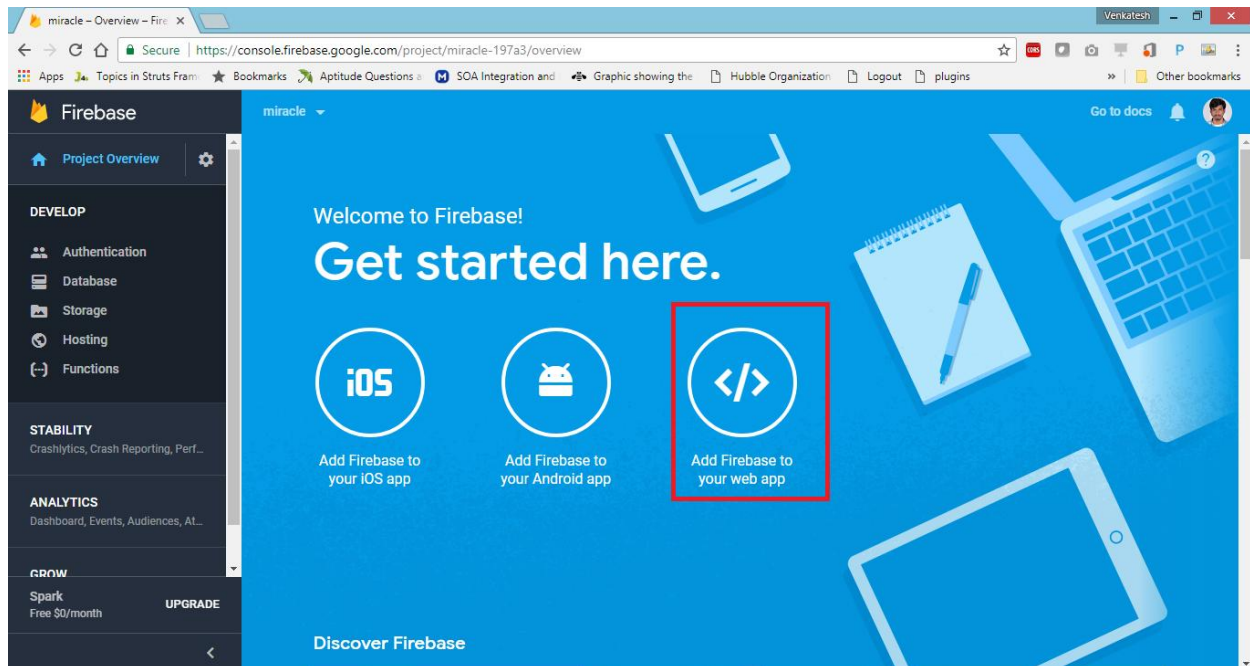
```
import { AngularFireAuthModule } from 'angularfire2/auth';
import { AngularFireDatabaseModule } from 'angularfire2/database';
import { AngularFireModule } from 'angularfire2';
```

#7 | Configure Firebase Authentication

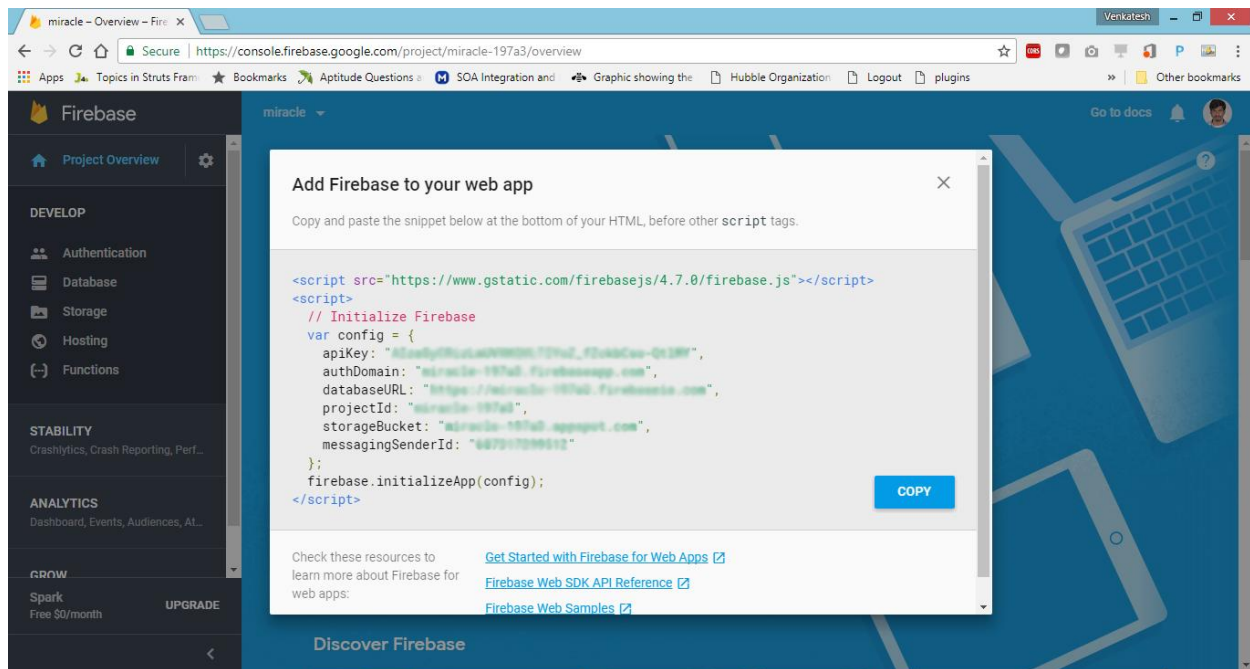
Go to home page (Project Overview) in the firebase project console as shown in the figure.



Once you click on the project overview then the following page will display.



Click on Add firebase to your web app, then a popup window will open with configuration details as shown in the figure.

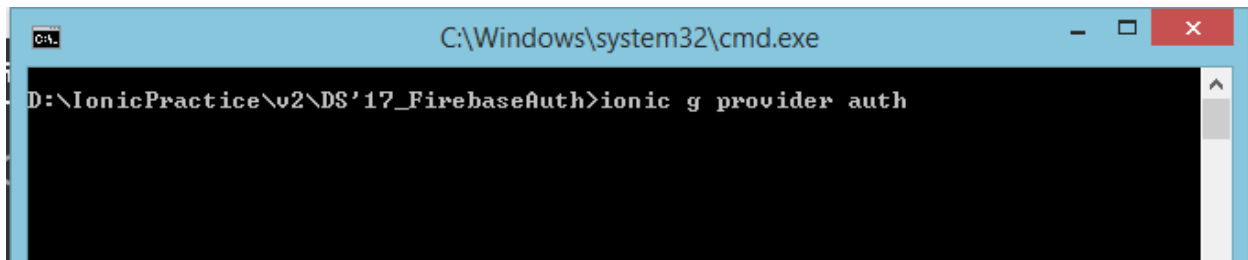


Just click on copy option as shown in the above page and paste it in your ionic application. Open the **app.module.ts** file and paste the below code your configuration details.

```
export const firebaseConfig = {  
  apiKey: "<Your apikey >",  
  authDomain: "<Your authDomain>",  
  databaseURL: "<Your databaseURL>",  
  projectId: "<Your ProjectID>",  
  storageBucket: "<Your storageBucket>",  
  messagingSenderId: "<Your MessagingSenderId>"  
};
```

#8 | Create a Authentication Service

In order to create a service provider in your ionic application, give this command `ionic g provider <ServiceName>` in the command prompt as shown in the figure.



Right after the service provider creation, you need to add this service in the providers array in **app.module.ts**. Also you have to import the service.

Import Statement - `import { Auth } from '../providers/auth';`

Provider Array -

```
providers: [  
  StatusBar,  
  SplashScreen,  
  Auth,  
  {provide: ErrorHandler, useClass: IonicErrorHandler}  
]
```

#9 | Service Implementation for Sign-Up and Sign-In

Here is code snippet for create user with email and password and signin with email and password.

Code Snippet

```
import { Injectable } from '@angular/core';
import { FirebaseApp } from 'angularfire2';
import * as firebase from 'firebase/app';

import { AngularFireModule } from 'angularfire2';
import { AngularFireAuthModule, AngularFireAuth } from
'angularfire2/auth';

import { Observable } from 'rxjs/Observable';

@Injectable()
export class Auth {
  user: Observable<firebase.User>;

  constructor(afAuth: AngularFireAuth) {
    this.user = afAuth.authState;
  }

  login(credentials: { email: string, password: string }) : Promise <any>{

    return  firebase.auth().signInWithEmailAndPassword(credentials.email,
credentials.password);
  }

  register(credentials: { email: string, password: string }) : Promise<any> {
    return
firebase.auth().createUserWithEmailAndPassword(credentials.email,
credentials.password);
  }
}
```

#10 | Create Sign-In and Sign-Up pages

To create a new page in your ionic application, give the following command,

ionic g page <page name>

After creating new page, you have to update the **app.module.ts** with this new page details like importing page, adding the **SecondPage** in the **declarations** and **entryComponents** arrays in **app.module.ts** file as shown in the below code.

Code Snippet : 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 { HomePage } from '../pages/home/home';
import { Register } from '../pages/register/register';
import { Auth } from '../providers/auth';
import { AngularFireAuthModule } from 'angularfire2/auth';
import { AngularFireDatabaseModule } from 'angularfire2/database';
import { AngularFireModule } from 'angularfire2';
import { SuccesPage } from '../pages/succes-page/succes-page';
```

```
export const firebaseConfig = {
  apiKey: "<Your apikey >",
  authDomain: "<Your authDomain>",
  databaseURL: "<Your databaseURL>",
  projectId: "<Your ProjectID>",
  storageBucket: "<Your storageBucket>",
  messagingSenderId: "<Your MessagingSenderId>"
};
```

```
@NgModule({
  declarations: [
    MyApp,
    HomePage,
    Register,
```

```
SuccesPage
],
imports: [
  BrowserModule,
  IonicModule.forRoot(MyApp),
  AngularFireModule.initializeApp(firebaseConfig),
  AngularFireDatabaseModule,
  AngularFireAuthModule
],
bootstrap: [IonicApp],
entryComponents: [
  MyApp,
  HomePage,
  Register,
  SuccesPage
],
providers: [
  StatusBar,
  SplashScreen,
  Auth,
  {provide: ErrorHandler, useClass: IonicErrorHandler}
]
})
export class AppModule {}
```

#11 | Designing the screens

In this lab you need to design 3 screens, one is Sign-up screen, second one is Sign-in screen and finally third one is success screen. Below are the code snippets for each screen.

Code Snippet : Home.html

```
<ion-header>
```



```
<ion-navbar color="headerColor">
  <ion-title text-center>
    DS'17 Firebase Authentication
  </ion-title>
</ion-navbar>
</ion-header>

<ion-content padding text-center>

<!-- DS'17 Logo -->
<div >
  
</div>
<!-- Login Form -->
<ion-item >
  <ion-label floating><ion-icon name="person" ></ion-icon> Username</ion-label>
  <ion-input type="text" [(ngModel)]="uname1" ></ion-input>
</ion-item>
<ion-item >
  <ion-label floating>
    <ion-icon name="lock"></ion-icon> Password
  </ion-label>
  <ion-input type="password" [(ngModel)]="pass1"></ion-input>
</ion-item>

<div text-center padding>
  <button type="submit" ion-button class="lButton" block
(click)="login(uname1,pass1)">Login</button>
<h4 (click)="register()" class="txtNewUser">Register Here</h4>
<div text-center>
  
</div>
</div>
</ion-content>
```

```
<ion-footer>
<h6 class="copyRight" text-center> Miracle Software Systems, Inc. 2017 </h6>
</ion-footer>
```

Code Snippet: Home.scss

```
.txtNewUser {
  color: #00aae7;
}
.lButton{
  width: 105%;
  text-transform: none;
  background-color: #d33257
}
```

Code Snippet : Register.html

```
<ion-header>

<ion-navbar color="headerColor">
  <ion-title >
    Register
  </ion-title>
</ion-navbar>

</ion-header>

<ion-content padding>
<div text-center>
  
</div>
<!-- Login Form -->
<ion-item >
  <ion-label floating><ion-icon name="person" ></ion-icon> Username</ion-
label>
```

```
<ion-input type="text" [(ngModel)]="uname" ></ion-input>
</ion-item>
<ion-item class="login-signup">
  <ion-label floating>
    <ion-icon name="lock"></ion-icon> Password
  </ion-label>
  <ion-input type="password" [(ngModel)]="pass" ></ion-input>
</ion-item>

<div text-center padding>
  <button          type="submit"          ion-button          class="lButton"
(click)="register(uname,pass)" >Register Here</button>
</div>
</ion-content>
```

Code Snippet : success-page.html

```
<ion-header>
<ion-navbar color="headerColor">
  <ion-title text-center>
    Successfully Login
  </ion-title>
</ion-navbar>
</ion-header>

<ion-content padding>
<h4 text-center>Logged Succesfully</h4>
<div text-center padding>
  <button type="submit" ion-button class="lButton" (click)="goBack()" >Go
Back</button>
</div>
</ion-content>
```

Code Snippet : theme/variable.scss

```
$font-path: "../assets/fonts";  
$app-direction: ltr;
```

```
@import "ionic.globals";
```

```
$colors: (  
  primary: #488aff,  
  secondary: #32db64,  
  danger: #f53d3d,  
  light: #f4f4f4,  
  dark: #222,  
  headerColor: #d33257  
);
```

```
@import "ionic.theme.default";  
@import "ionic.ionicons";
```

```
@import "roboto";  
@import "noto-sans";
```

The output of the above screens are shown in the figure. No need to run the execution command again and again, if you made any changes then the application will automatically builds.

Fig :

DS'17 Firebase Authentication



 Username

 Password

Login

[Register Here](#)

MIRACLE
SOFTWARE SYSTEMS

Miracle Software Systems, Inc. 2017

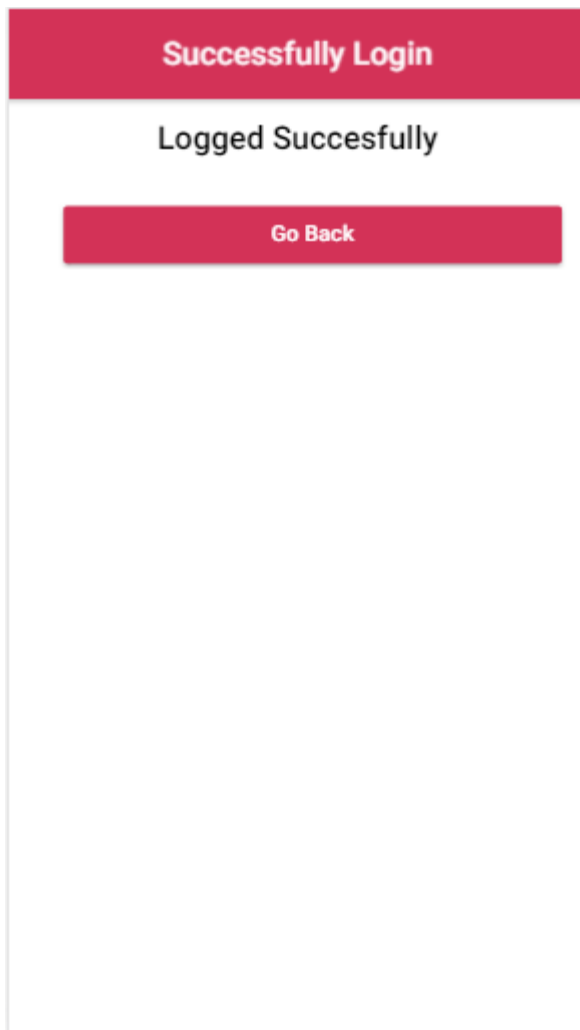
← Register



 Username

 Password

Register Here



#12 | Accessing service from component

In this lab, you are going to use home screen as a sign-in and register screen as a sign-up. Below are the code snippets for accessing service from the component

Code snippet : home.ts

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { Register } from '../register/register';
```

```
import { Auth } from '../providers/auth';
import { SuccesPage } from '../succes-page/succes-page';

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {

  constructor(public navCtrl: NavController, public auth: Auth) {

  }
  register() {
    this.navCtrl.push(Register);
  }

  login(uname:string,pass:string){

    let credentials={
      "email":uname,
      "password":pass
    }
    this.auth.login(credentials)
    .then((data) => {
      console.log(data);
      this.navCtrl.setRoot(SuccesPage);
    });
  }
}
```

Code Snippet : register.ts

```
import { Component } from '@angular/core';
import { NavController, NavParams, Loading, AlertController } from 'ionic-
angular';
```

```
import { Auth } from '../providers/auth';
import { HomePage } from '../home/home';

@Component({
  selector: 'page-register',
  templateUrl: 'register.html',
})
export class Register {

  loading: Loading;

  regResult;
  constructor(public navCtrl: NavController, public navParams: NavParams, public
auth:Auth, public alertCtrl: AlertController){
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad Register');
  }
  register(uname:string,pass:string){
    let credentials={
      "email":uname,
      "password":pass
    }
    this.auth.register(credentials).then((data) => {
      console.log(data);
      this.navCtrl.push(HomePage);
    });
  }
}
```

Code Snippet: success-page.ts

```
import { Component } from '@angular/core';
import { NavParams, NavController } from 'ionic-angular';
```



```
import { HomePage } from '../home/home';

@Component({
  selector: 'page-succes-page',
  templateUrl: 'succes-page.html',
})
export class SuccesPage {

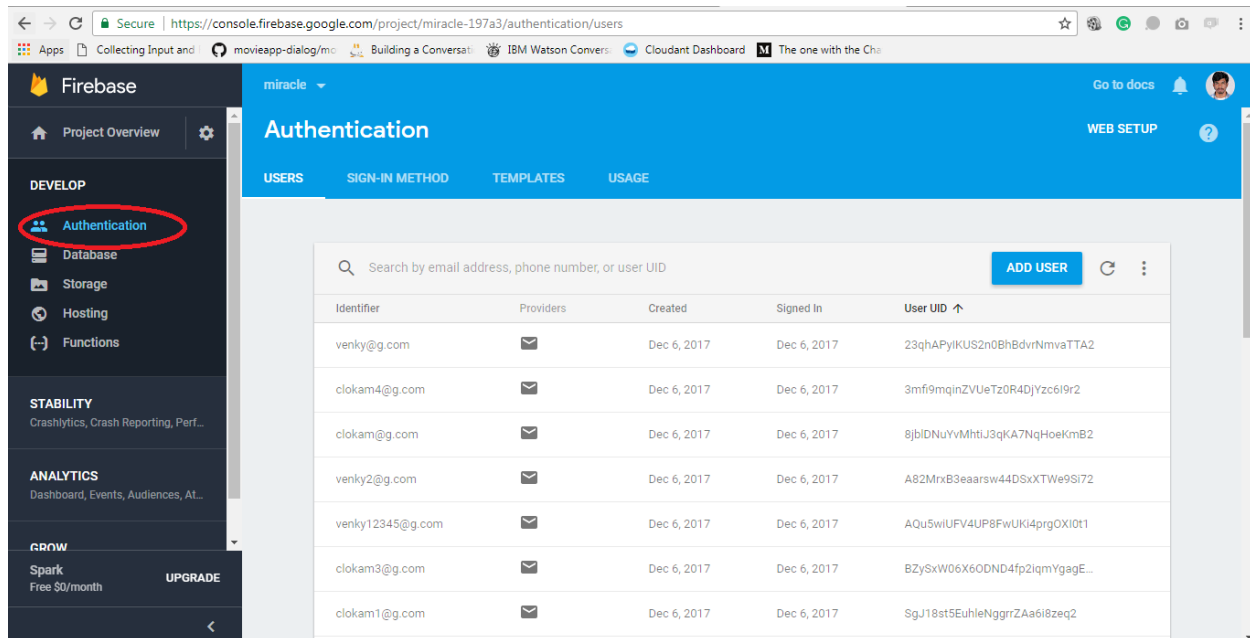
  constructor(public navCtrl: NavController, public navParams: NavParams) {
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad SuccesPage');
  }

  goBack(){
    this.navCtrl.push(HomePage);
  }
}
```

#13 | Checking User Data in Firebase console

In order to check the user data in firebase console, you need to Click on authentication tab on left panel as shown in the figure



From the above screen you can find the user data in the tabular form, you can also add the user manually by clicking on **Add User** option