MIRACLE
SOFTWARE SYSTEMS

Login Functionality with Ionic and Firebase Authentication

Open Lab | Digital Summit '18

Miracle Innovation Labs

Miracle Software Systems, Inc.

Login Functionality with Ionic and Firebase Authentication

Goal

In this lab we will guide you to add Firebase authentication to an Ionic 3 mobile app. Here you will create a mobile app with SignUp and SignIn screens and apply authentication by using Firebase.

Pre-Requisites

- The following installations will needed to complete this lab and run successfully,
- Google Account
- Firebase Plug-in to be installed
- Node JS and NPM installed
- Cordova
- Ionic Framework
- Text editor (Visual Studio Code is preferable)

Technology Involved

- Ionic 3 Framework
- Cordova
- HTML/CSS/JavaScript
- Typescript
- Angular 2/4

Lab Steps

Let's get started with the lab!

Step #1 | Create a New Ionic Project

To create/start a new project in Ionic 3 Framework, we need to give a below command,

Ionic start <application-name> <starter-template-name>

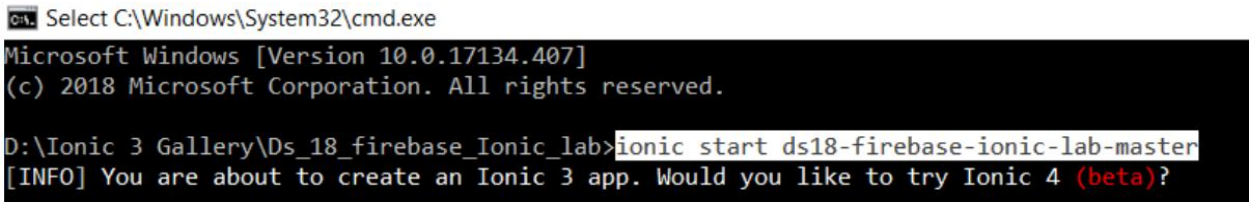
We will be having the following starter templates in Ionic 3 Framework.

tabs	A starting project with a simple tabbed interface
blank	A blank starter project
side menu	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

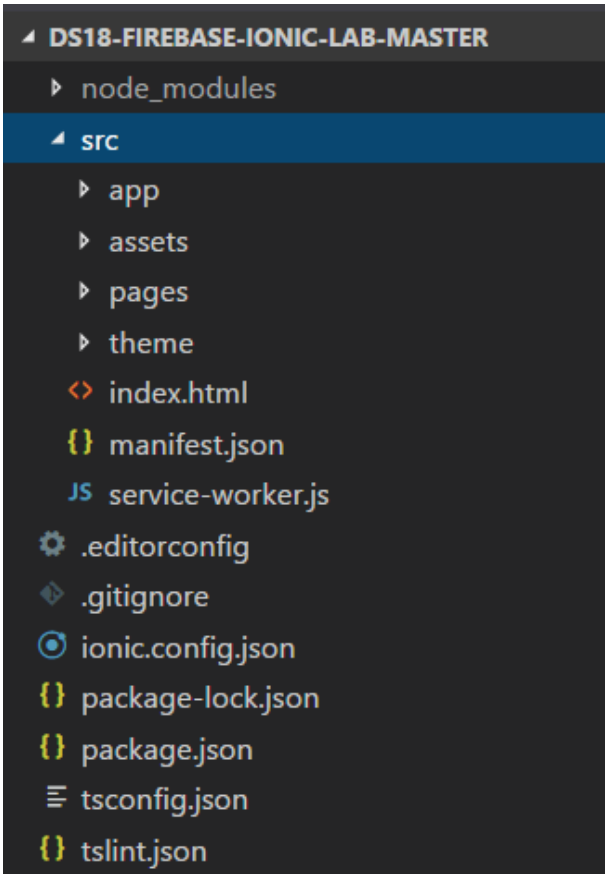
In this lab we are going to use blank template and start implementing the actual application.

ionic start DS18_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 below,



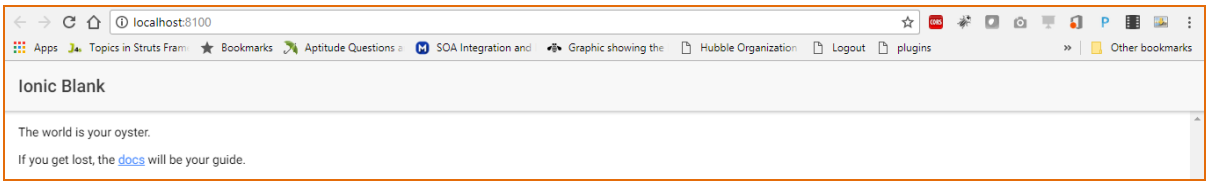
Step #2 | Running the Application

Before providing the run command, route to the current application folder and give the following commands.

Run on Browser - **ionic serve**

Run on Mobile simulator - **ionic serve -l**

Run the application in the browser, the output will be as shown below,

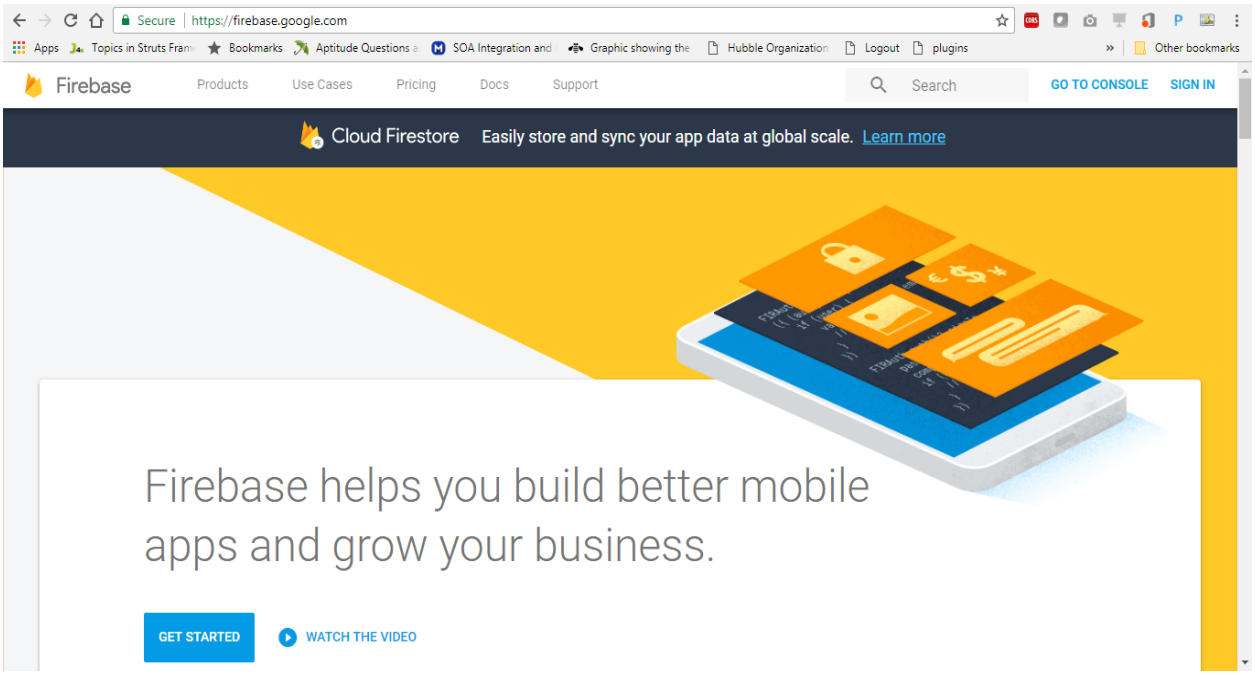


Run the application on a simulator, and the output will be as shown below. You can notice that the output is in two different platforms one is in Android and another one is in iOS.

Step #3 | Create a Project in Firebase Console

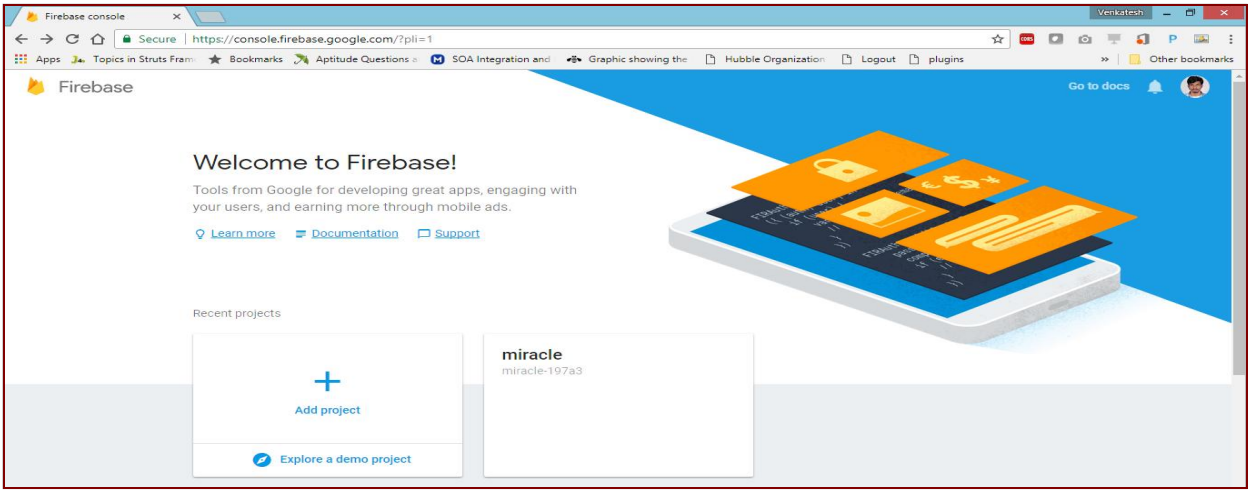
To work with Firebase authentication, you need to create a project in the Firebase console.

Open the link, https://firebase.google.com/



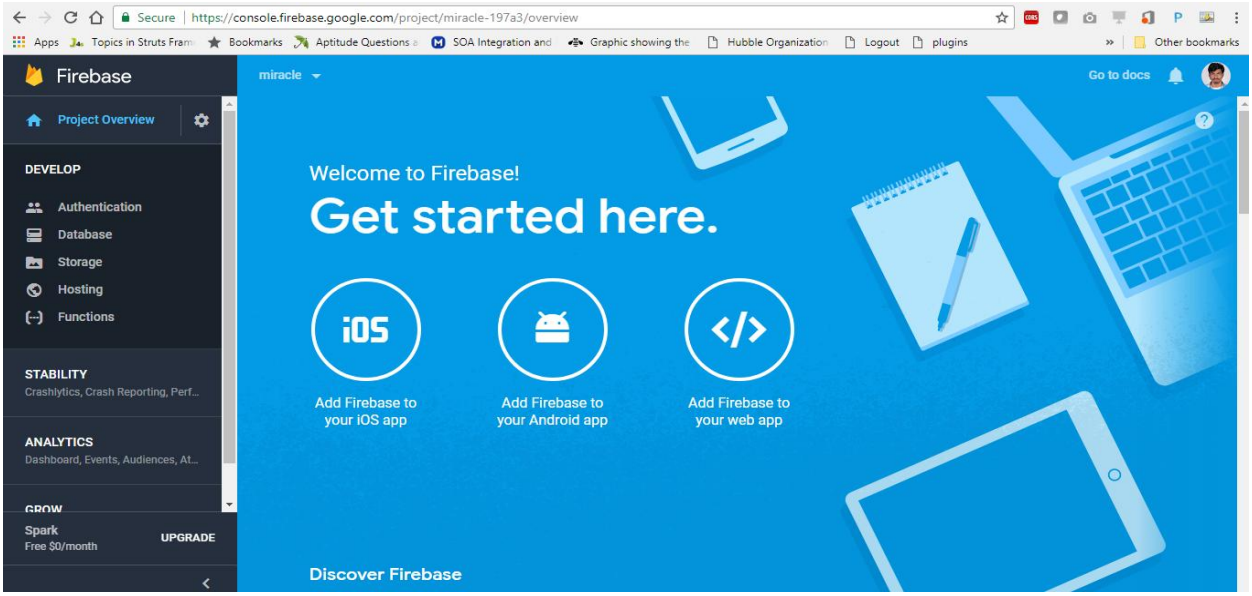
Click on **Go to Console** which is on top right corner, and it requests you to sign-in if you are not logged in.

Login with your Gmail credentials and you will be able to see the dashboard of Firebase console as below.



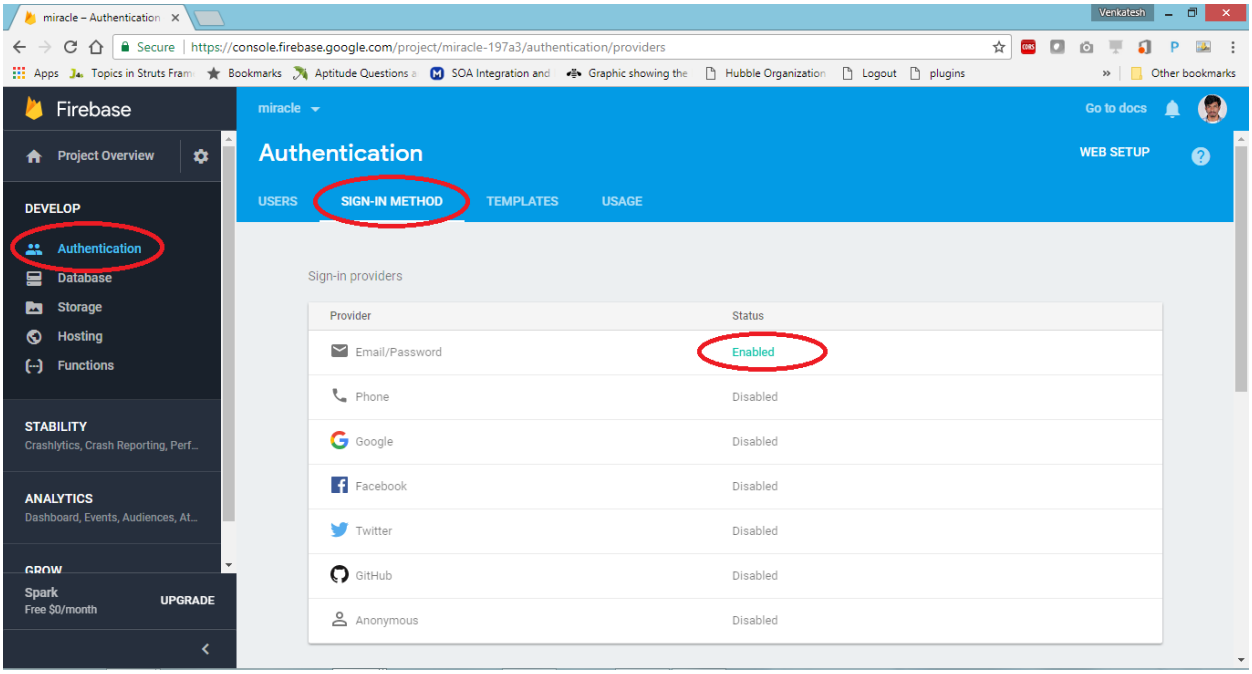
In order to work on Firebase authentication, you need to create a new project or you can add Firebase authentication to the existing project. To add new project, click on **Add project** tile.

Once you click on Add project, your project is created and will redirect to your project dashboard.



Step #4 | Enable Sign-In Method

In firebase, you can select the sign-in method as per your requirement. In this scenario, you are going to choose **Email/Password** method. To work on Email/Password, you need to enable the option as shown below.



Click on Authentication tab on the left menu and then select Sign-In method tab on the top. Now, Email/Password is successfully enabled.

Step #5 | Install Firebase Plug-in

In order to work with Firebase, you need to have the **angularfire2** plug-in installed in your Ionic application. Change the directory to where your Ionic application has been created and provide the following command,

npm install angularfire2 firebase

```
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
```

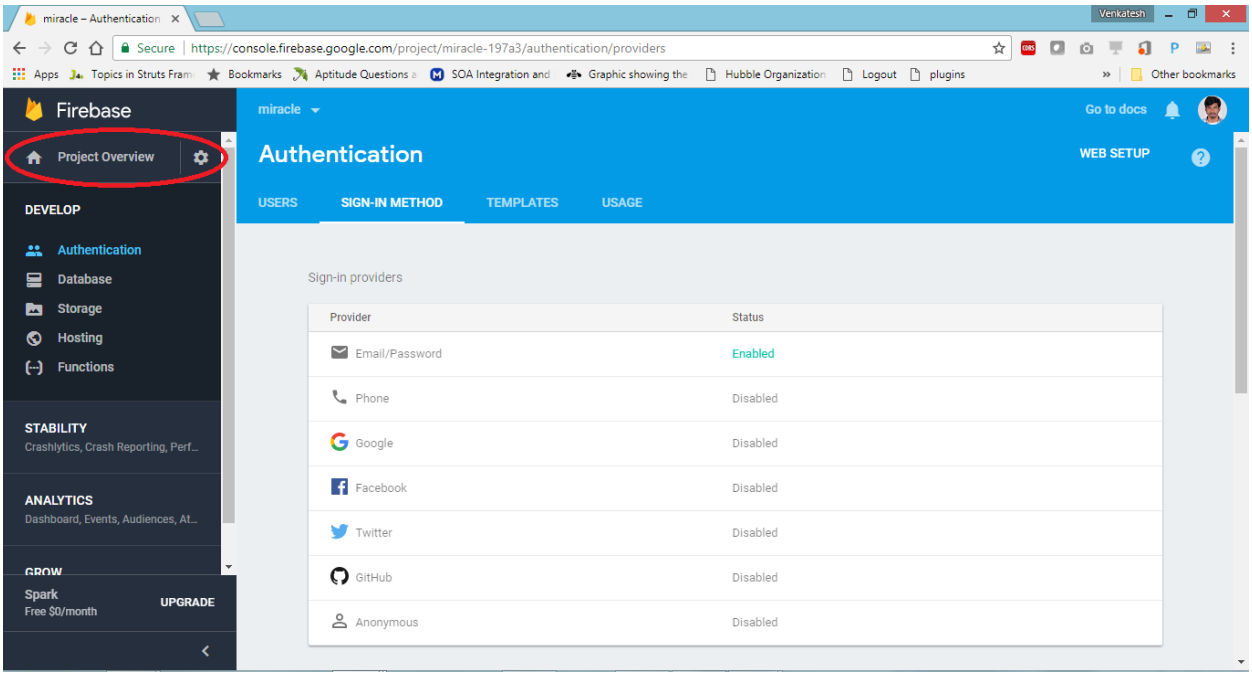
Step #6 | Import and Initialize

After installation of angularfire2 plug-in, you need to import required modules. Here, you should add the following statements in **app.module.ts** file to import modules.

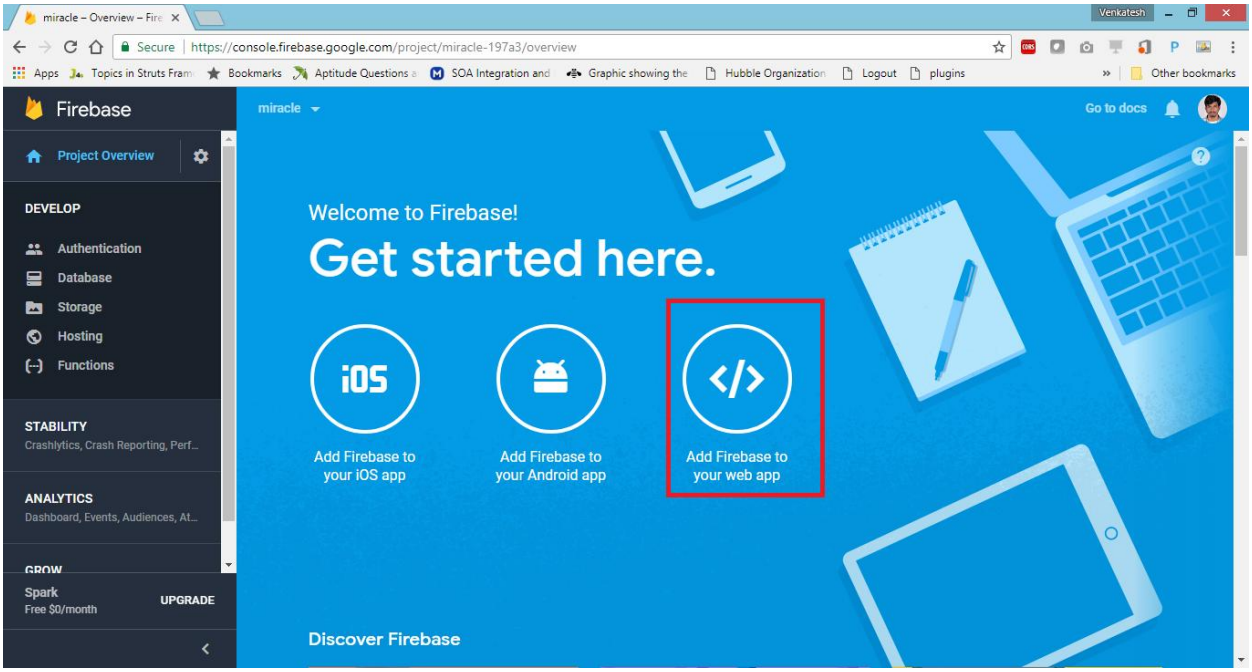
```
import { AngularFireAuthModule } from 'angularfire2/auth';
import { AngularFireDatabaseModule } from 'angularfire2/database';
import { AngularFireModule } from 'angularfire2';
```

Step #7 | Configure Firebase Authentication

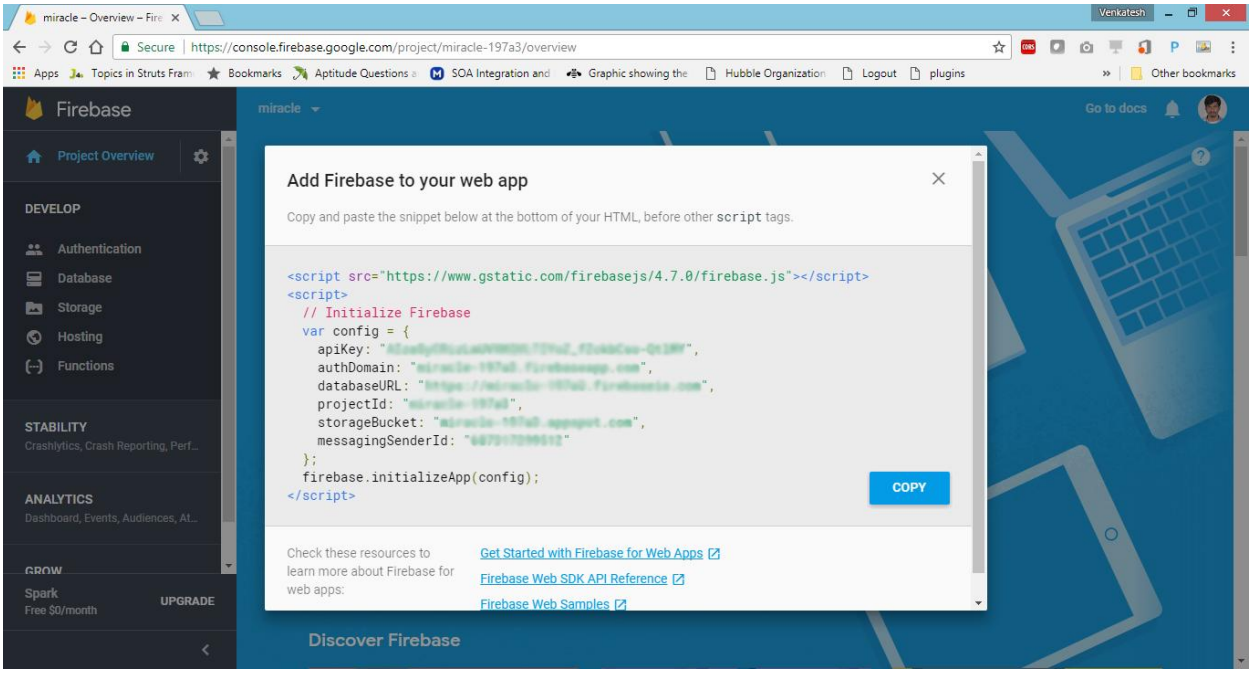
Go to home page (**Project Overview**) in the Firebase project console as shown in the below figure.



Once you click on the project overview, the following page will be displayed.



Click on **Add Firebase to your web app**, then a popup window will be opened with all the configuration details as shown below.



Click on the copy button and paste it in your Ionic application. Open the **app.module.ts** file and paste the below code with your configuration details.

```
export const firebaseConfig = {

apiKey: "<Your apikey >",

authDomain: "<Your authDomain>",

databaseURL: "<Your databaseURL>",

projectId: "<Your ProjectID>",

storageBucket: "<Your storageBucket>",

messagingSenderId: "<Your MessagingSenderID>"
```

Step #8 | Create an 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 below.

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

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

Provider Array -

```
providers: [

StatusBar,

SplashScreen,

Auth,
```

```
{provide: ErrorHandler, useClass: IonicErrorHandler}
]
```

Step #9 | Service Implementation for SignUp and SignIn

Here is the code snippet to create user with email and password. SignIn with same credentials

```
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);
}
}
```

Step #10 | Create SignIn and SignUp Pages

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

ionic g page <page name>

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 { Auth } from '../providers/auth';
import { AngularFireAuthModule } from 'angularfire2/auth';
```

```
import { AngularFireDatabaseModule } from 'angularfire2/database';

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

import { HomePage } from '../pages/home/home';
import { Auth } from '../providers/auth/auth';
import { RegisterPage } from '../pages/register/register';
import { SuccessPage } from '../pages/success/success';
import { AngularFireModule } from 'angularfire2';

        export const firebaseConfig = {

        apiKey: "<Your apikey >",

        authDomain: "<Your authDomain>",

        databaseURL: "<Your databaseURL>",

        projectId: "<Your ProjectID>",

        storageBucket: "<Your storageBucket>",

        messagingSenderId: "<Your MessagingSenderID>"

        };

@NgModule({

declarations: [

],

imports: [

  BrowserModule,

  IonicModule.forRoot(MyApp),

  AngularFireModule.initializeApp(firebaseConfig),
```

```
  AngularFireDatabaseModule,
  AngularFireAuthModule
],
bootstrap: [IonicApp],
entryComponents: [
],
providers: [
  StatusBar,
  SplashScreen,
  Auth,
  {provide: ErrorHandler, useClass: IonicErrorHandler}
]
})
export class AppModule {}
```

Step #11 | Designing the Screens

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 : [Please refer to the cheat sheet]

The outputs of the above screens are shown in the below screenshots. No, need to run the execution command multiple times, if you have made any changes then the application will automatically build them.

Screenshots

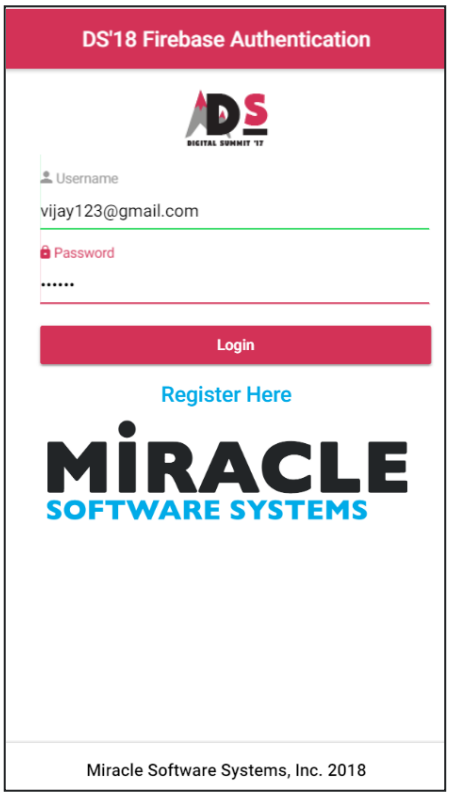
Step #12 | Accessing Service from Component

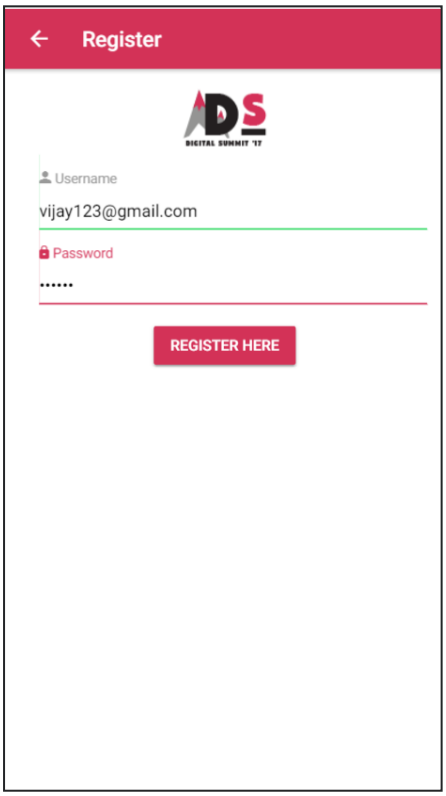
You are going to use home screen as a SignIn and Register screen as a SignUp. Below are the code snippets for accessing service from the component,

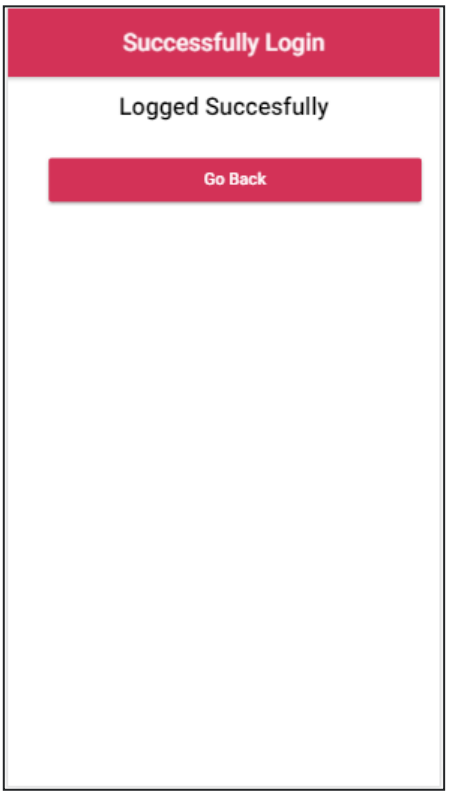
Code snippet : home.ts

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

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







import {

```
NavController } from 'ionic-angular';
import { Register } from '../register/register';
import { Auth } from '../../providers/auth/auth';
import { SuccessPage } from '../success-page/success-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(SuccessPage);
      });
    }
}
```

Code Snippet : register.ts

```
import { Component } from '@angular/core';
import { NavController, NavParams, Loading, AlertController } from 'ionic-angular';
import { Auth } from '../../providers/auth/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);
  });
}
}
```

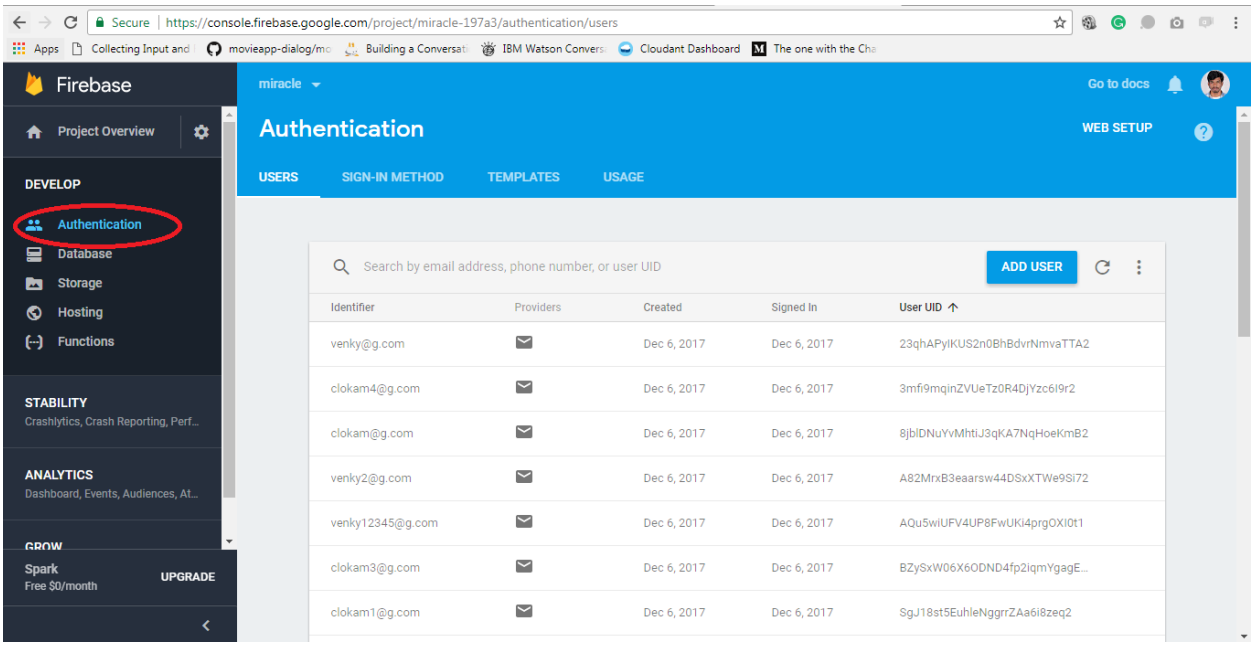
Code Snippet: success-page.ts

```
import { Component } from '@angular/core';
import { NavParams, NavController } from 'ionic-angular';
import { HomePage } from '../home/home';
@Component({
selector: 'page-success-page',
```

```
templateUrl: 'success-page.html',
})
export class SuccessPage
{
constructor(public navCtrl: NavController, public navParams: NavParams) {
}
ionViewDidLoad() {
  console.log('ionViewDidLoad SuccessPage');
}
```

Step #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 the left menu as shown below.



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.

For any questions regarding the lab please feel free to reach out to

innovation@miraclesoft.com. **We hope you enjoyed this lab!**