Create a new folder.

Open it in visual studio code.

Open terminal and write: npx create-react-app sso\_demo

In terminal write: cd sso\_demo

Now we want to install some dependencies, so in terminal write: npm install firebase

And also: npm install react-firebaseui

Go to App.js and write this to be able to access all the firebase controls:

import firebase from "firebase"

And import StyledFirebaseAuth to give us the buttons directly:

import StyledFirebaseAuth from "react-firebaseui/StyledFirebaseAuth"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Go to firebase and add a new project.

Then go to authentication and click ‘Get started’.

Choose Facebook and enable it.

Go to: <https://developers.facebook.com/apps/>

And register as a developer by clicking on the sidebar and click ‘Get Started’.

Go back to : <https://developers.facebook.com/apps/>

And click ‘Create app’, choose ‘Build Connected Experiences’ and give it a name.

NOTE: if the security check does not show up, do this in an incognito window.

Copy the App ID. Go back to firebase, click on facebook, and paste the App ID.

Then for the app secret, go back to the facebook for developers. Click set up facebook login.

Click Web. In the site URL write: <http://localhost:3000/>

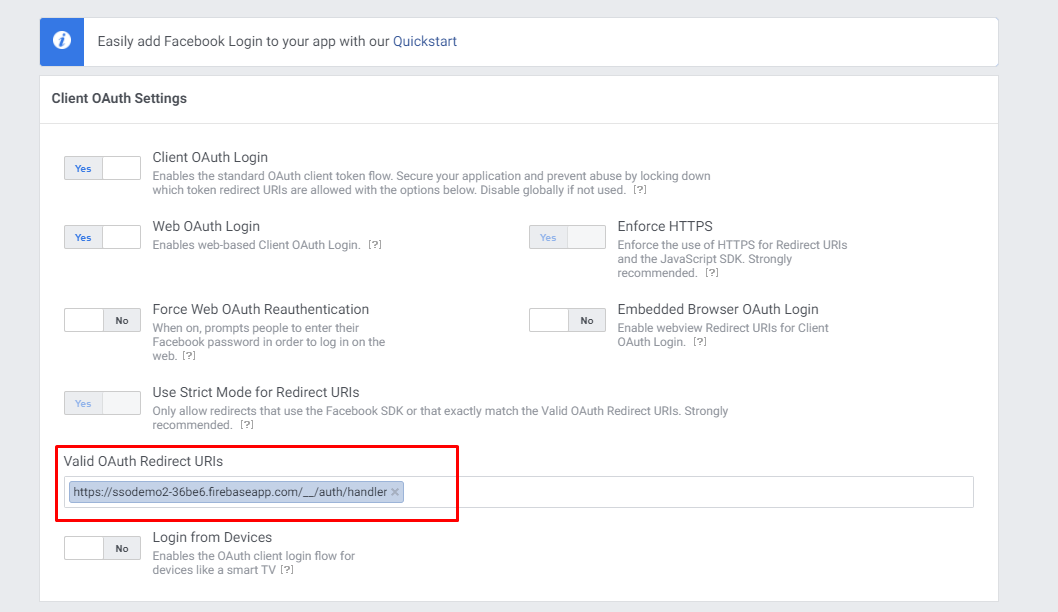
Then click next for all of them.

In the sidebar click settings, then click basic.

Copy the app secret and paste it in firebase. Before clicking save, copy the OAuth redirect URI.

Go back to facebook for developers. Go to dashboard and set up facebooklogin. From the sidebar click facebook login, and then click settings.

Paste the OAuth URI in ‘Valid OAuth Redirect URIs’



Then click save changes.

Go back to firebase, and click save.

Now we will enable Google. From firebase, click on Google and click enable.

Give an email and click save.

In App.js write:

class App extends Component {

  state = {isSignedIn: false}

  render() {

    return (

    <div className="App">

        {this.state.isSignedIn ? (

          <div>Signed in!</div>

        ) : (

          <div>Not signed in!</div>

        )}

      </div>

    )

  }

}

What this does is that we will be using this to check if a user is signed in or not using the state ‘isSignedIn’. This will be useful so that the user is not allowed to click on the signed in buttons again once he is already logged in.

Under the state write:

uiConfig = {

    signInFlow: "popup",

    signInOptions: [

      firebase.auth.GoogleAuthProvider.PROVIDER\_ID,

      firebase.auth.FacebookAuthProvider.PROVIDER\_ID

    ],

    callbacks: {

      signInSuccess: () => false

    }

  }

What the popup does is so that it doesn’t redirect the user to a different page.

We are also providing the authentications that we have set up previously.

Once the user signs in successfully, we do nothing, as we can see in the callback we have set it to false.

After this write:

  componentDidMount = () => {

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

      this.setState({isSignedIn: !!user})

    })

  }

When the component is mounted this function will be called. If the user is not signed in yet, it will be changed to true and if the user is signed in it will be changed to false. That is why we use !!user (not not user).

Now we will get the api key and the authDomain from the firebase.

Above the class App write:

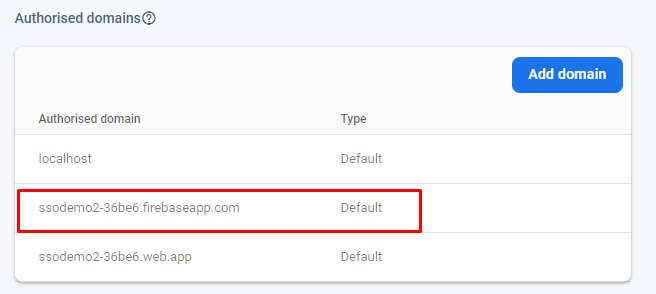
firebase.initializeApp({

  apiKey: "",

  authDomain: ""

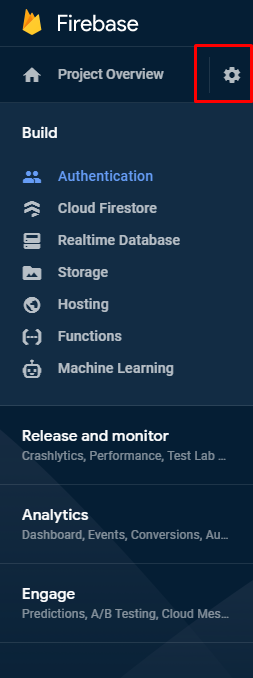
})

Go to firebase, authentication, sign-in method. Scroll down to the auth domain and copy it.

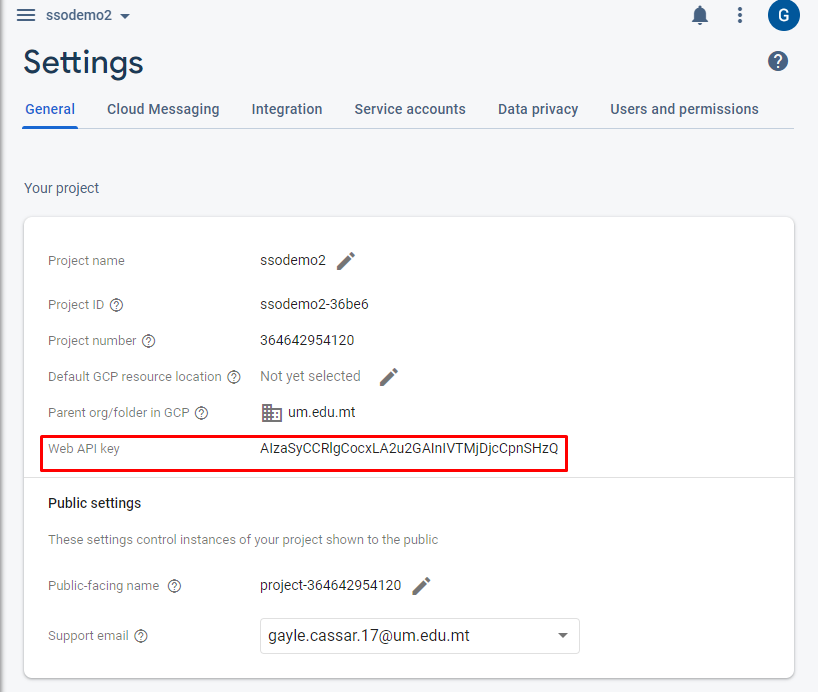


Paste it in the code in quotation marks.

For the API key go to the side bar and click on the cog:



Then click on “users and permission”. Then click on ‘General” and copy the Web Api Key:



Paste it in the apiKey of the code with quotation marks:

firebase.initializeApp({

  apiKey: "AIzaSyCCRlgCocxLA2u2GAInIVTMjDjcCpnSHzQ",

  authDomain: "ssodemo2-36be6.firebaseapp.com"

})

Now we will create a sign out button.

In the return statement, wrap the signed in div into a span. Below the div signed in add the button:

 <span>

      <div>Signed in!</div>

      <button onClick={() => firebase.auth().signOut()}>Sign Out</button>

 </span>

Change the Not Signed In to:

 <StyledFirebaseAuth

          uiConfig={this.uiConfig}

          firebaseAuth={firebase.auth()}

          />