Firebase Authentication

Firstly go to [www.console.firebase.google.com](http://www.console.firebase.google.com) and create a project.

Go to the authentication section on the left.

Here go to the Sign-in method tab and choose any login method that you wish to use..

# Email/Password:

After enabling email and password option, click on the web setup (in the case of a web project) and copy the code. Paste it in your html file or better yet make an anonymous function in your JS file.

(function () {

// Initialize Firebase

var config = {

apiKey: "AIzaSyCTgCahhu3Ge7y5isGf3JxyoH3AlHK8f6g",

authDomain: "doneapp2.firebaseapp.com",

databaseURL: "https://doneapp2.firebaseio.com",

projectId: "doneapp2",

storageBucket: "doneapp2.appspot.com",

messagingSenderId: "786827425800"

};

firebase.initializeApp(config);

//Getting the elements

const userName = document.getElementById("userName");

const userPass = document.getElementById("userPass");

const btnLogin = document.getElementById("btnLogin");

const btnSignup = document.getElementById("btnSignup");

Here’s a file link that can help you with the rest:

<https://firebase.google.com/docs/auth/web/password-auth?utm_campaign=Firebase_featureoverview_education_auth_en_07-26-16&utm_source=Firebase&utm_medium=yt-desc>

And here’s a video link:

<https://www.youtube.com/watch?v=-OKrloDzGpU&index=6&list=PLl-K7zZEsYLnq6DvMtUJVDLe4w5wcTwUl>

# Sign In:

use this for sign in on the event listener of the sign in button

firebase.auth().signInWithEmailAndPassword(name, pass)

and to catch any errors use:

const promise = firebase.auth().signInWithEmailAndPassword(name, pass);

promise.catch(e => console.log(e.message));

The complete event listener for the sign in button might look like this:

btnLogin.addEventListener("click", e => {

const name = userName.value;

const pass = userPass.value;

const auth = firebase.auth(); //Using the firebase's auth method

//Signing in with firebase auth

const promise = auth.signInWithEmailAndPassword(name, pass);

promise.catch(e => console.log(e.message));

})

# Sign Up:

Similarly for Signing up use this code after applying an event listener on the signup button: (createUser)

const promise = auth.createUserWithEmailAndPassword(name, pass);

promise.catch(e => console.log(e.message));

The complete event listener for signup button might look like this:

btnSignup.addEventListener("click", e => {

const name = userName.value;

const pass = userPass.value;

const auth = firebase.auth(); //Using the firebase's auth method

//Signing in with firebase auth

const promise = auth.createUserWithEmailAndPassword(name, pass);

promise.catch(e => console.log(e.message));

})

# Logging in after sign up:

Now you have signed up but how to login after the signup.

One way is to use promise.then method but there is a drawback to that, which is that it will run only once.

Hence we have to create a realtime listener which checks for changes in the authentication database.

//Adding a realtime listener

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

if (user) {

console.log(user);

}

else {

console.log('not logged in')

}

})

Here user is a parameter, variable which contains all the user info if they are logged in or “not logged in”.

Here we are consoling the user info only for development purposes, it is not something that we should do in practice.

Now what about logout?

Well create an HTML logout button with Id “btnLogout” (for simplification) and then we add an eventlistener to it in JS file.

const btnLogout = document.getElementById("btnLogout");

btnLogout.addEventListener('click', e => {

firebase.auth().signOut();

})

Logging in Anonymously