* ***FireBase:***

*A tool made by google that provides services like Authentication and Database storage.*

* *Creating a new Project in Firebase:* ***Go to console -> create a project***
* *Installing Firebase:* ***yarn add firebase***

***CRUD: Create Read Update and Delete.***

***Q. How Sigin with Google works with Firebase ?***

* *Firstly the user/client from our website does signin with google, if their credentials are correct google provides them a Authentication Code. Now, this Authentication code is sent to firebase from our website. Now, firebase checks that whether the given Authentication code is correct or not with Google.*
* *Now after this Google verifies this stuff by sending a verification token to Firebase.*
* *If everything is fine and user is verified , then Firebase send a access token to our defining what operations it can perform. Now, our websites uses this access token to perform crud operation.*

***Firebase Setup in Firebase Files.(Firebase.js)***

*import {initializeApp} from 'firebase/app';* ***// creates app function based on some config.***

*import {getAuth,signInWithRedirect,signInWithPopup,GoogleAuthProvider} from 'firebase/auth';*

***// consists of secret keys to be used to access our project and initialize our firebaseAPP.***

*const firebaseConfig = {*

*apiKey: "AIzaSyD6NWxVEsG9hUlV5syqHCG\_RUK2wGPsQ7s",*

*authDomain: "bucketio.firebaseapp.com",*

*projectId: "bucketio",*

*storageBucket: "bucketio.appspot.com",*

*messagingSenderId: "462189121092",*

*appId: "1:462189121092:web:19d665e13c67c9ef53f611",*

*measurementId: "G-LCEL10V40N"*

*};*

***// Initialize FirebaseApp***

*const firebaseApp = initializeApp(firebaseConfig)*

***//Setting up google Auth provider. Here we are always forcing to select an account while authenticating by use of custom parameters.***

*const provider = new GoogleAuthProvider();*

*provider.setCustomParameters({*

*prompt:'select\_account'*

*})*

***// creating Auth instances***

*export const auth = getAuth();*

***//Signin with Google Popup***

*export const signInWithGooglePopup = ()=>signInWithPopup(auth, provider);*

***NOTE:***

* ***After this Go to Build->Authentication->SignIn method -> Enable Google in Firbase in order to do signin with google.***

***Signin Page Setup:***

***SNIPPET:***

*const logGoogleuser = async ()=>{* ***//As it takes time so this function needs to be async***

*const response = await signInWithGooglePopup();*

*console.log(response);*

*}*

***FireStore Database***

*(current/latest database, realtime database is the old one.)*

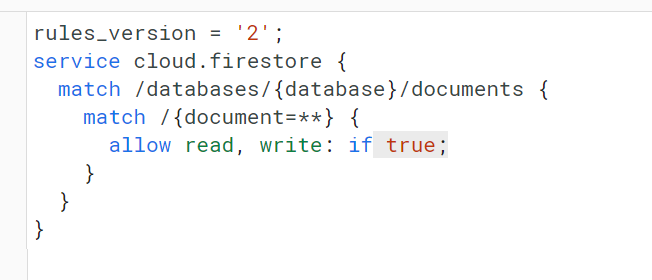
* ***Inside Firestore DB how data resides:***

*Inside Collection => inside document => data we use for our CRUD operations.*

***Q. How to create Database and start it:***

*Go to Firestore database -> Do create database => once database is created go*

*to rules and update the if statement to True:*

**

***Setting up Firestore Database in Firebase.js File:***

***SNIPPET:***

*export const db = getFirestore();* ***// Initializing our firestore database***

*export const createUserDocumentFromAuth = async(userAuth)=>{*

***//Call this func inside signiwithgoogle func and users object from the response.***

*const userDocRef=  doc(db,'users',userAuth.uid****);***

***// (db is database instance,users is collection, userAuth.id is a***

***unique identifier to identify our document.)***

*const userSnapshot = await getDoc(userDocRef);*

*console.log(userSnapshot.exists()); // Checking this particular document exists or not*

*}*

***NOTE:***

* ***Docs:*** *helps to get a document instance*
* ***Getdocs:*** *Getdoc helps in getting data from document*
* ***Setdocs:*** *Setdoc helps in setting data from document*

***Creating user if user not exists after checking it with getDoc() on signing in:***

***SNIPPET:***

*if(!userSnapshot.exists()){*

*const {displayName,email} = userAuth;*

*const dates = new Date();*

*try{*

***setDoc(userDocRef,{displayName,email,dates});***

*}*

*catch(error){*

*console.log("Error Message: ",error )*

*}*

*}*

***If user not exists we will take userDocumentReference along***

***with name, email and today’s date and create a new user using setDoc().***

***SigninWithRedirect(using GoogleAuthProvider):***

***SNIPPET:***

***Firebase.js:***

*export const signInWithGoogleRedirect = ()=>signInWithRedirect(auth, provider);*

***Sigin.js:***

*const logGoogleuserRedirect = async ()=>{*

*const {user} = await signInWithGoogleRedirect();*

*createUserDocumentFromAuth(user);*

*}*

***Note:*** *As in signin with redirect we move to a different site due to which our* ***authentication***

***states gets lost here****. So, inorder to prevent this we use* ***auth instance which is like a memory bank holding the states of authentication.***

***SNIPPET:***

***Signin.js:***

*const Signin = () => {*

*useEffect(() => {*

*const getUsers = async () => {*

***const response = await getRedirectResult(auth);***

***//auth stores the authentication state here***

*}*

*getUsers();*

*}, []);*

*return (*

*<div>*

*<button onClick={signInWithGoogleRedirect}>SignIn</button>*

*</div>*

*)*

*}*