**Firebase Database Setup and Features**

Login/Register Setup

In our code, we first must initialize and setup a connection to the firebase database. To do this will have to import a few libraries from the firebase API.

import { initializeApp } from "https://www.gstatic.com/firebasejs/9.8.1/firebase-app.js";

import { getDatabase } from "https://www.gstatic.com/firebasejs/9.8.1/firebase-database.js";

import { getAuth,

createUserWithEmailAndPassword,

signInWithEmailAndPassword}

from "https://www.gstatic.com/firebasejs/9.8.1/firebase-auth.js";

Once we have imported all the libraries needed, we will initialize the app and connect the website to the database.

*// Your web app's Firebase configuration*

*// For Firebase JS SDK v7.20.0 and later, measurementId is optional*

*const firebaseConfig = {*

*apiKey: "AIzaSyAizKiBWMmyxDAF9BkJFTfsk8VnEXzCnnQ",*

*authDomain: "trippr-62b9f.firebaseapp.com",*

*databaseURL: "https://trippr-62b9f-default-rtdb.firebaseio.com",*

*projectId: "trippr-62b9f",*

*storageBucket: "trippr-62b9f.appspot.com",*

*messagingSenderId: "836402456269",*

*appId: "1:836402456269:web:3fd354f16e2d5f84d8ab84",*

*measurementId: "G-MN1VTEJYGM"*

*};*

*// Initialize Firebase*

*const app = initializeApp(firebaseConfig);*

*const database = getDatabase(app);*

*const auth = getAuth();*

Once we have configured the firebase. We will call a function which will take three arguments: **auth**, ***email, password.***

*createUserWithEmailAndPassword(auth, email, password)*

*.then((userCredential) => {*

*// Signed in*

*const user = userCredential.user;*

*window.location.href = '../userinformation/userinformation.html';*

*})*

*.catch((error) => {*

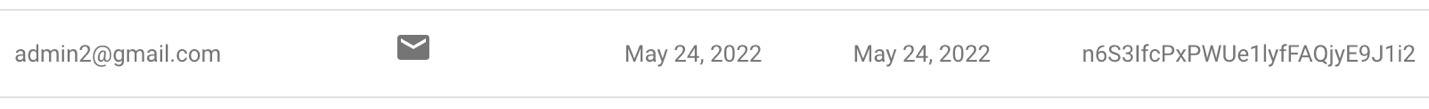
*const errorCode = error.code;*

*const errorMessage = error.message;*

*console.log(errorCode);*

*console.log(errorMessage);*

*});*

Above, it will call the function, and pass it three values. The email and password is retrieved from user inputs. If this is successful, it will create a user in the firebase, and generate a UID. Here is a sample user below:

The user is also asked basic information. They will be asked to provide their address, first and last name along with other things. We will have to create another table to hold this information. This table will be linked with user id generated by firebase. In order to do this we will have to add some extra libraries to our import statements. Check below:

import { initializeApp } from "https://www.gstatic.com/firebasejs/9.8.1/firebase-app.js";

import { getDatabase,

ref,

set } from "https://www.gstatic.com/firebasejs/9.8.1/firebase-database.js";

import { getAuth,

createUserWithEmailAndPassword,

signInWithEmailAndPassword,

onAuthStateChanged}

from "https://www.gstatic.com/firebasejs/9.8.1/firebase-auth.js";

Next we will check to see if a user is logged in. If the user is logged in, it will get the user id and create a new entry in a table with that as the ID.

The code below, will create a table called USERS and set its nodes as the user id, the nodes will contain the user information.

const user = auth.currentUser;

if (user) {

// User is signed in

e.preventDefault();

const uid = user.uid;

var fname = document.getElementById('fname').value;

var lname = document.getElementById('lname').value;

var city = document.getElementById('city').value;

set(ref(database, 'users/' + uid), {

firstname: fname,

lastname: lname,

city: city

});

var delayInMilliseconds = 1000; //1 second

setTimeout(function() {

window.location.href = '../questionaire/questionaire.html'

}, delayInMilliseconds);

}else {

alert("wrong");

}

Here is a sample of the dataset:

Graphical user interface, application

Description automatically generated

We are also saving the user preferences in a separate table, which is also linked to the user id.

The code below, will create a collection with the name as the user id. This is one table. Inside the user id, we are able to create multiple tables that will be linked to that user id only. We call these **documents.** Inside each document we will store information. For now, we are only creating one document, which hold the user preferences.

const user = auth.currentUser;

if(user){

var id = user.uid;

const docRef = setDoc(doc(db, id, "users-prefs"),radiodict);

var delayInMilliseconds = 1000; //1 second

setTimeout(function() {

window.location.href = '../navigation/navigation.html';

}, delayInMilliseconds);

}else{

console.log("no login");

}

As you can see, we are getting the user Id of the current logged in user, creating a collection of it, and then creating a document called “user-prefs” and pass it the values as a **dictionary**.

Here is a sample output from the database:

Graphical user interface, application, Word

Description automatically generated