Firebase

Package:

npm install -g firebase-tools

Initializing Firebase - <https://www.loom.com/share/ee9dfcb00dbc42bd8971bcb916fb8418>

Populating Data in Firebase - <https://www.loom.com/share/8c9de52911f54dbe8c6eb31220f33d1b>

<https://console.firebase.google.com/u/0/project/m-city-4c8ad/settings/general/web:ODAzMjcxOTctMTAxNS00ZmM5LTk5YTAtOTdkZWQ5NmE2NGMx>

<https://firebase.google.com/docs/hosting/reserved-urls?authuser=0>

<https://firebase.google.com/docs/web/setup?authuser=0>

src/firebase.js – basic config file

import firebase from "firebase/app";

import "firebase/app";

import "firebase/database";

import "firebase/auth";

import "firebase/storage";

// Your web app's Firebase configuration

var firebaseConfig = {

  apiKey: "AIzaSyCROOna6NC7F5H22k1mnZ7VkarHdRHmOcc",

  authDomain: "m-city-4c8ad.firebaseapp.com",

  databaseURL: "https://m-city-4c8ad.firebaseio.com",

  projectId: "m-city-4c8ad",

  storageBucket: "m-city-4c8ad.appspot.com",

  messagingSenderId: "1017221765198",

  appId: "1:1017221765198:web:6da7e6ce9a4c15ea1b8b1f",

  measurementId: "G-RVPCYPH668"

};

// Initialize Firebase

firebase.initializeApp(firebaseConfig);

// Initialize DB Alias

const firebaseDB = firebase.database();

const firebaseMatches = firebaseDB.ref("matches");

const firebasePromotions = firebaseDB.ref("promotions");

const firebaseTeams = firebaseDB.ref("teams");

const firebasePlayers = firebaseDB.ref("players");

export {

  firebase,

  firebaseMatches,

  firebasePromotions,

  firebaseTeams,

  firebasePlayers,

  firebaseDB

};

// fetching data

// firebaseDB

//   .ref("matches")

//   .once("value")

//   .then(snapshot => {

//     console.log(snapshot.val());

//   });

* Get last 6 data from firebase

Blocks.js (component using firebase)

import { firebaseMatches } from "../../../firebase";

componentDidMount() {

    firebaseMatches

      .limitToLast(6)

      .once("value")

      .then(snapshot => {

        console.log(snapshot.val());

      });

  }

* Check if data is there

A

Enroll.js

import { firebasePromotions } from "../../../firebase";

// ======= Insert into firebase DB: Start ======

      firebasePromotions

        .orderByChild("email")

        .equalTo(dataToSubmit.email)

        .once("value")

        .then(snapshot => {

          if (snapshot.val() === null) {

            firebasePromotions.push(dataToSubmit); // insert to firebase db

            this.resetFormSuccess(true);

          } else {

            this.resetFormSuccess(false); // already exist case

          }

        });

      // ======= Insert into firebase DB: End ======

Firebase Auth Setup –

Video - <https://www.loom.com/share/85ccc187c800497cba7626889e3a38eb>

Src/firebase.js

import firebase from "firebase/app";

import "firebase/app";

import "firebase/database";

import "firebase/auth";

… contd.

Src/signin/index.js

import React, { Component } from "react";

import { firebase } from "../../firebase";

submitForm(event) {

    event.preventDefault();

    let dataToSubmit = {};

    let formIsValid = true;

    // looping through above state to get input values

    for (let key in this.state.formdata) {

      dataToSubmit[key] = this.state.formdata[key].value;

      formIsValid = this.state.formdata[key].value;

    }

    if (formIsValid) {

      // ===== Checking in firebase for valid email and password for Auth: Start       firebase

        .auth()

        .signInWithEmailAndPassword(dataToSubmit.email, dataToSubmit.password)

        .then(() => {

          this.props.history.push("/dashboard");

          // console.log("User is authenticated...");

        })

        .catch(error => {

          this.setState({

            formError: true

          });

        });

      // ====== Checking in firebase for valid email and password for Auth: Start

    } else {

      this.setState({

        formError: true

      });

    }

  }

* Logout / Signout with firebase

import { firebase } from "../../../firebase";

firebase

      .auth()

      .signOut()

      .then(

        () => {

          console.log("Logged out successful.");

        },

        error => {

          console.log("Error logging out");

        }

      );

🡪 Firebase get data based on ID

// fetching matches collection data by match id

      firebaseDB

        .ref(`matches/${matchId}`)

        .once("value")

        .then(snapshot => {

          const match = snapshot.val();

          getTeams(match, "Edit Match");

        });

* Insert / Update in Firebase

if(this.state.formType === 'Edit Match'){

                firebaseDB.ref(`matches/${this.state.matchId}`)

                .update(dataToSubmit).then(()=>{

                    this.successForm('Updated correctly');

                }).catch((e)=>{

                    this.setState({ formError: true })

                })

            } else {

                firebaseMatches.push(dataToSubmit).then(()=>{

                    this.props.history.push('/admin\_matches');

                }).catch((e)=>{

                    this.setState({ formError: true })

                })

            }

React firebase file uploader

* For this – npm install react-firebase-file-uploader
* Check src/component/ui/fileuploader.js (reusable component)

Will be used in addEditPlayer.js

(src\Components\admin\players\addEditPlayers.js)

// This method is auto called before render - 2

  static getDerivedStateFromProps(props, state) {

    console.log("App - Get Derived state from Props");

    console.log(props, state);

    return null; // you must return state or props or null

  }

Fileuploader documentation (react-firebase-file-uploader = package)

<https://github.com/fris-fruitig/react-firebase-file-uploader>

Firebase Storage

firebasePlayers.storage().ref(this.props.dir)}

For uploading

 <FileUploader

    accept="image/\*"

    name="image"

    randomizeFilename

    storageRef={firebase.storage().ref(this.props.dir)}

    onUploadStart={this.handleUploadStart}

    onUploadError={this.handleUploadError}

    onUploadSuccess={this.handleUploadSuccess}

            />

handleUploadStart = () => {

    this.setState({

      isUploading: true

    });

  };

  handleUploadError = () => {

    this.setState({

      isUploading: false

    });

  };

  handleUploadSuccess = filename => {

    console.log(filename); // image name

    this.setState({

      name: filename,

      isUploading: false

    });

// GETTING image download URL

    // firebase provides only name, we need to get download image URL

    firebase

      .storage()

      .ref(this.props.dir)

      .child(filename)

      .getDownloadURL()

      .then(url => {

        console.log(url);

        this.setState({ fileURL: url });

      });

  };

Firebase hosting - <https://www.loom.com/share/4273fd4b6fa34f3d9484d3bd8bbb824e>

Project Overview - <https://www.loom.com/share/dc4e7bdafa9d4d28ab9fdefbce28bc03>