Firebase

Firebase acts as a powerful backend as a service solution empowering developers to seamlessly integrate cloud service into their react applications.

What can you do with firebase?

1. Authentication – by helping manage who gets in and out, handling sign-ups and logins. Security made easy.
2. Storage - making it easy to show off our media in our React apps.
3. Real-time database – like a cloud storage

Steps of integrating firebase with react

* Go to firebase console
* Click on get started
* Then click on add project and follow the on-screen instructions – once project is created, you will be directed to project dashboard
* Add your App to the project
* You will be provided with a firebase Software Development Kit (SDK) snippet which you will need to add to your app
* To install firebase to your project run “npm install firebase”
* Then create a folder named config in your project root directory (optional)
* Inside the folder create a firebase.js file and paste your SDK snippet code from firebase to the file.

Firebase.jsx

// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import { getAuth } from "firebase/auth";

import { getFirestore } from "firebase/firestore";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

const firebaseConfig = {

apiKey: "AIzaSyAunjAfONSszGEYlZWc0OIjECJ7TcMkRqk",

authDomain: "my-app-3ac04.firebaseapp.com",

projectId: "my-app-3ac04",

storageBucket: "my-app-3ac04.firebasestorage.app",

messagingSenderId: "1047494719243",

appId: "1:1047494719243:web:13ed54802eeeaad41d940e"

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

export const auth = getAuth(app);

export const db = getFirestore(app);

Set Up Authentication methods

* Go to Authentication in the left hand menu of the firebase console
* Click on get started and Sign-in method tab
* Enable Email/Password under the Sign-in providers section and save
* Now you can create a sign-up and sign-in function in your app

Set Up firestore database

Navigate to firestore database and click on create database

Click on next with default location settings and use start in test mode and create your database

Login.jsx

// src/components/Login.js

import { useState } from "react";

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

import { createUserWithEmailAndPassword, signInWithEmailAndPassword } from "firebase/auth";

const Login = ({ setUser, setUserRole }) => {

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const [role, setRole] = useState("employee"); // default role

const [isNewUser, setIsNewUser] = useState(false);

const handleSubmit = async (e) => {

e.preventDefault();

try {

let userCredential;

if (isNewUser) {

// Create account

userCredential = await createUserWithEmailAndPassword(auth, email, password);

} else {

// Sign in

userCredential = await signInWithEmailAndPassword(auth, email, password);

}

setUser(userCredential.user);

setUserRole(role);

} catch (error) {

console.error("Error during authentication", error);

alert(error.message);

}

};

return (

<div>

<h2>{isNewUser ? "Sign Up" : "Login"}</h2>

<form onSubmit={handleSubmit}>

<div>

<label>Email: </label>

<input

type="email"

value={email}

onChange={(e) => setEmail(e.target.value)}

required />

</div>

<div>

<label>Password: </label>

<input

type="password"

value={password}

onChange={(e) => setPassword(e.target.value)}

required />

</div>

<div>

<label>Role: </label>

<select value={role} onChange={(e) => setRole(e.target.value)}>

<option value="employee">Employee</option>

<option value="employer">Employer</option>

</select>

</div>

<button type="submit">{isNewUser ? "Sign Up" : "Login"}</button>

</form>

<p>

{isNewUser ? "Already have an account?" : "New user?"}{" "}

<button onClick={() => setIsNewUser(!isNewUser)}>

{isNewUser ? "Login" : "Sign Up"}

</button>

</p>

</div>

);

};

export default Login;

EmployerDashboard.jsx

// src/components/EmployerDashboard.js

import { useState, useEffect } from "react";

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

import { collection, getDocs } from "firebase/firestore";

const EmployerDashboard = () => {

const [applications, setApplications] = useState([]);

useEffect(() => {

const fetchApplications = async () => {

try {

const querySnapshot = await getDocs(collection(db, "applications"));

const apps = querySnapshot.docs.map(doc => ({ id: doc.id, ...doc.data() }));

setApplications(apps);

} catch (error) {

console.error("Error fetching applications", error);

}

};

fetchApplications();

}, []);

return (

<div>

<h2>Employer Dashboard</h2>

<h3>Employee Applications</h3>

<ul>

{applications.map(app => (

<li key={app.id}>

{app.application} <br />

<small>{new Date(app.submittedAt.seconds \* 1000).toLocaleString()}</small>

</li>

))}

</ul>

</div>

);

};

export default EmployerDashboard;

``

App.jsx

// src/App.js

import { useState } from "react";

import Login from "./Components/Login";

import EmployeeDashboard from "./Components/EmployeeDashboard";

import EmployerDashboard from "./Components/EmployerDashboard";

function App() {

  const [user, setUser] = useState(null);

  const [userRole, setUserRole] = useState(null);

  // Simple logout function

  const logout = () => {

    setUser(null);

    setUserRole(null);

  };

  if (!user) {

    return <Login setUser={setUser} setUserRole={setUserRole} />;

  }

  return (

    <div>

      <header>

        <button onClick={logout}>Logout</button>

      </header>

      {userRole === "employee" && <EmployeeDashboard/>}

      {userRole === "employer" && <EmployerDashboard />}

    </div>

  );

}

export default App;