

## Firebase authentication in React.

BR → console.firebase.google.com/u/4/

→ login with email

→ create a project

→ Enter your project name (Book list with Firebase)

☒

☒

→ Continue

→ ☒

↑ Disable Google Analytics

→ Create project

→ Continue

→ </> → Register app

Book-list-with-firebase

☒ Also setup Firebase Hosting for this app.

Register app → Copy all code →

→ T → npm install firebase

FL → src → firebase (folder) → config.js → paste it here

Next

↓

npm install -g firebase-tools

Next

→ Continue to console

↓

Build

→ Authentication

:

click → Get started → Email/Password → Enable

☒ Enable

Save →

FL → src → firebase → config.js →  
 import {getAuth} from "firebase/auth";  
 export const auth = getAuth(app);

FL → src → views → LoginPage.jsx →  
 import {auth} from '../firebase/config.js';  
 f  
 c  
 c  
 → console.log(auth);  
 return

- Now to store email & password we want state, we will create only 1 state & then store both of them in it & change them dynamically.

FL → src → views → LoginPage.jsx →  
 :

function L

c

c

→ const [userCredentials, setUserCredentials] = useState({});

→ function handleCredentials(e) {  
 setUserCredentials({ ...userCredentials, [e.target.name]: e.target.value });  
 console.log(userCredentials);  
 }  
 return

<label> Email

→ <input onChange={e => {handleCredentials(e)}}

<label> Password

→ <input onChange={e => {handleCredentials(e)}}



BR.

Email

Password

Console

① {}

② {email: 'anil'}

③ {email: 'anil', password: 'yadav'}

- Now we want when we click on ~~login~~<sup>signup</sup> we can send credentials to firebase

FL →

LoginPage.jsx →  
func handleC

}

Function handleSignup(e) {

e.preventDefault();

console.log('signup');

}

<button onClick={e => {handleSignup<sup>(e)</sup>}} > Sign Up </button>

BR →

← Click

Email

Password

← Click

Console

signup

firebase docs → Build → Authentication → Web → Password Authentication

Create a password-based account

1.

2.

Web module API

FL → LoginPage.jsx → copy import part but not getAuth as we have it already as auth.  
↓ paste here

- import { createUserWithEmailAndPassword } from "firebase/auth";
- copy ~~remaining~~ code other than getAuth() part & paste to it is handleSignup(e)

FL → LoginPage.jsx →

paste  
here →

```
function handleSignup(e) {
  e.preventDefault();
  createUserWithEmailAndPassword(auth, userCredentials.email,
    userCredentials.password) ← 1 edited
    .then((userCredential) => {
      // Signed in
      const user = userCredential.user;
      console.log(user);
      //...
    })
    .catch((error) => {
      const errorCode = error.code;
      const errorMessage = error.message;
      console.log(errorCode);
      " (errorMessage);
      //...
    });
}
```

BR →

Email  
(a a a)

Password

(1 2 3)

Signup

Signup ← click → then

Console →

auth/invalid-email  
Firebase: Error  
(auth/invalid-email).



Email  
react.firebase.course@gmail.com  
 Password  
123456  
SignUp ← click

Console  
 User Impl {  
 .  
 advarInfo: {  
 .  
 email: "react.firebase.course@gmail.com"  
 .  
 uid: "WTk, . . ."  
 }

BR → Firebase → Authentication → Users →

Identifier	Providers	Created	Signed In	User UID
react.f...@gmail.com		Aug 31, 2023	Aug 31, 2023	WTkSm...

- Handling Signup Errors received back by firebase when we put invalid email.

FL → LoginPage.jsx →

```

→ const [error, setError] = useState("");
  {
    handleSignup(e) {
      setError(""); // to create error empty once we try to Signup again
    }
  }

  .catch((error) => {
    //
    //
    setError(error.message);
    //
    //
  })
  }
  
```

↓

error &lt;div className="error">

error <div>

<p ... >Forgot Password </p>

BR →

Signup ← Click

Email

(aaa)

Password

(123)

Signup

← Click

Firebase: Error (auth/invalid-email).

- Now use a correct email & password

Email

(aaa@aaa.com)

Password:

(12345678)

Signup

← Click

This name now

BR → Firebase → Authentication → Users →

aaa@aaa.com	<input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/>

- Signing in existing users (Login)

Firebase docs →

Password Authentication →

Create

Sign in a user with an email address &amp; password

- Copy important part of signInWithEmailAndPassword



FL → LoginPage.jsx →

→ import { create  
"firebase/auth";

, signInWithEmailAndPassword } from

• Now copy signInWithEmailAndPassword from docs &

FL → LoginPage.jsx →

function handleLogin(e) {

e.preventDefault();

setError("");

signInWithEmailAndPassword(auth, userCredentials.email,  
userCredentials.password)

• then((userCredential)) => {

console.log(userCredential.user); // similar in handleSignup did too  
}

• catch((error)) => {

setError(error.message);

});

}

<button onClick={e => {handleLogin(e)}} > login

• BR → Login ← click

Email

(aaaa)

Password

(1234)

Login

Click

Firebase: Error (auth/invalid-email)

Email

(react at the base ...)

Password

(correct Pass)

Login

Click

{ disappear  
err }

Console  
UserImpl {

# • The RESET PASSWORD FUNCTION (FORGOT PASSWORD)

FL → LoginPage.jsx →

import { ... , sendPasswordResetEmail } from 'firebase/auth';

function handlePasswordReset() {

const email = prompt('Please enter your email');

sendPasswordResetEmail(auth, email);

alert('Email sent! Check your inbox for password reset instructions.');

}

<button onClick={e => {handlePasswordReset}} >Forgot Password

BR →

localhost:5174 says  
Please enter your email  
react.

Cancel Ok ← Click

Forgot Password ← Click

localhost:5174 says  
Email sent! Check your inbox for password reset instructions  
Ok ← click

Gmail → Firebase email → click on link →

Reset your password for react-firebase.

New password

New password

SAVE ← Click

Password changed  
You can now sign in with your new password



BR →

Login ← Click

Email

Contact

Password

(New Password)

Login ← Click

Console

User Temp 1

2. Reset password msg template in email is editable in  
 Firebase → Authentication → Templates → Password reset →  
 → edit → Save

- Showing the homepage after login
- To store user we will use Redux.

FL → src → store → usersSlice.js →

{copy from notesSlice.js &amp; paste here then edit}

const usersSlice = createSlice({

name: 'users'

initialState: {

currentUser: null

},

reducers: {

setUser: (users, action) =&gt; {

users.currentUser = action.payload;

}

}

})

export const {setUser} = usersSlice.actions;

export const selectUsers = state =&gt; state.users;

export default usersSlice.reducer;

Want to connect to firebase s/w after 5

Page No.

10

Date

FL → src → store → store.js →

→ import usersReducer from './usersSlice.js';

ex

red

b

n

→ users: usersReducer

}

})

FL → LoginPage.jsx →

- import { useDispatch } from 'react-redux';

- import { setUser } from '../store/usersSlice.js';

func L

- const dispatch = useDispatch();

;

function handleSignup(e) {

createUserWithEmailAndPassword(auth,

• then((userCredential) => {

console

- dispatch(setUser({id: userCredential.user.uid, email: userCredential.

- user.email}));

})

;

function handleLogin(e) {

;

• then

console.log

{param1}

console.log



FL → usersSlice.js →

setUser: (

- console.log('setUser', action.payload);  
 users.  
 }

BD →

Login ← Click

Email

(react + ...)

Password

confirm Pass

Login ← click

console.

setUser usersSlice.js?  
 {  
 email: "react + ...",  
 id: "WISmX...",  
 }

UserImpl {  
 }  
 }

know remove console here?

FL → - import {selectUsers} from './store/usersSlice.js';

End App

- import {useSelector} from 'react-redux';

function App() {

- const user = useSelector(selectUsers);

return (

< >

{user: currentUser ?

<BrowserRouter>

<Routes>

<Route index element={}<BookPage/>}/>

< path="add-book" element={}<AddBookPage/>}/>

< path="book/:id" element={}<SingleBookPage/>}/>

</Routes>

</BrowserRouter>:

<LoginPage/>}

</> ) }

# NOTES

BR -> localhost:5174/add-book

=>

Login

Email

Search...

Password

-----

Login

← click => BR =>

/add-book

Add Book

Title

Book Cover



- Persisting the login state {when we switch page the login still be there} For if we login in 1 tab & open same url in another tab, we are not login in another tab}

Firebase docs → search sit Persistence → first → Authentication State Persistence →

Firebase docs → Authentication → GET Started →

Set an authentication state observer and get user data

For

Web modular API

FL → LoginPage.jsx →

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

const [error

onAuthStateChanged(auth, (user) => {

if (user) {

dispatch(SetUser({id: user.uid, email: user.email}));

//const uid =

//

} else {

dispatch(SetUser(null));

}

});

function handleCredentiaIs(e) {

• Now when we refresh we do not get logout once we login  
 FL → LoginPage.jsx →

```
f. handleSignup(e) {
```

```
  e
```

```
  s
```

```
  c
```

```
  // .then((
```

```
  //   dispat
```

```
  //   })
```

```
  :
```

```
f. handleLogin(e) {
```

```
  e
```

```
  s
```

```
  s
```

```
  // .then((
```

```
  //   dispa
```

```
  //   con
```

```
  //   })
```

```
  signIn
```

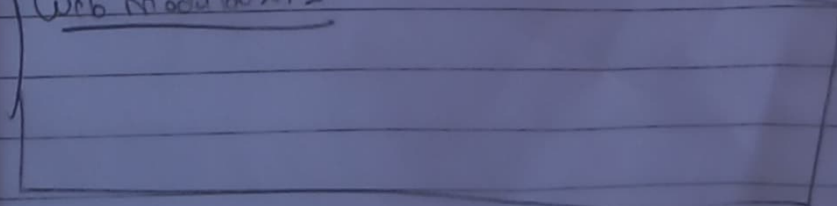
```
  :
```

• Logout function

firebase docs → Authentication → web → Password Authentication →  
 at the bottom →

To sign out a user, call signOut:

Web module API





FL → src → components → Header.js →

- import {signOut} from "firebase/auth";
- import {auth} from '../firebase/config.js';
- import {useDispatch} from 'react-redux';
- import {setUser} from '../store/userSlice.js';

function Header()

- const dispatch = useDispatch();
- function handleSignOut() {
  - if (confirm('Are you sure you want to log out?')) {
    - signOut(auth).then(() => {
      - dispatch(setUser(null));
    - }).catch((error) => {
      - console.log(error);

return (

- <button onClick={handleSignOut}
  - Logout

BR →

Are you sure you want to log out? ← Click → Logout

Cancel Take  
Click

localhost:5174

Login / Signup

Email

Password

In ~~another~~ tab as well we are logout

- Now when we login & then go to another url in another tab then for a moment we see login page but we want to see loader there in place of that.
- Adding a loader to the login page

FL → LoginPage.jsx

```
- const [isLoading, setIsLoading] = useState(true);
```

```
onAuthStateChanged(
```

```
  () {
```

```
    // ...
```

```
  }
```

```
  if(!isLoading) {setIsLoading(true)};
```

```
});
```

- Now we see refresh page we see loader for a moment