

Video : https://video.deakin.edu.au/media/t/1_7jw0x314

Git: <https://github.com/CameronHart1/SIT313-FullStack>

Screenshots:

Create a DEV@DEAKIN Account

First Name:

Last Name:

Email:

Password:

Confirm Password:

Login

Sign Up

Your email

Me

Your password

Login

Login with Google

```
import { useParams } from "react-router-dom";
import "../CSS/p_login.css";
import { Link, useNavigate } from "react-router-dom";
import {
  signInWithGooglePopup,
  createUserDocFromAuth,
  createAuthUserWithEmailAndPassword,
  signIn
} from "../utils/firebase";

// const for email,
// Check validity boxes for password /
// optional name box

const LoginPage = () => {
  const { SignType } = useParams();
  const [contact, setContact] = useState({
    firstName: "",
    lastName: "",
    email: "",
    password: "",
    confirmPassword: "",
  });

  const navigate = useNavigate();
  const { firstName, lastName, email, password, confirmPassword } = contact;
  // updating these vals
  const handleChange = (e) => {
    const { name, value } = e.target;
    setContact((preValue) => {
      return {
        ...preValue,
        [name]: value,
      };
    });
  };

  // logging in with google
  const logGoogleUser = async (e) => {
    // otherwise button reloads page
    e.preventDefault();
    const { user } = await signInWithGooglePopup();
    const userDocRef = await createUserDocFromAuth(user);
  };
  // logging in with email
  const handleSignIn = async (e) => {
    e.preventDefault();
    // Check validity boxes for password /
    // optional name box
  };
};
```

```

});

// logging in with google
const logGoogleUser = async (e) => {
  // otherwise buton reloads pagee
  e.preventDefault();
  const { user } = await signInWithGooglePopup();
  const userDocRef = await createUserDocFromAuth(user);
};

// logging in with email
const handleSignIn = async(e) => {
  e.preventDefault();
  const { user } = await signIn(email,password);
  const userDocRef = await createUserDocFromAuth(user);
  console.log(user);
};

// creating a new user via email
const handleSignUp = async(e) => {
  e.preventDefault();
  if (password !== confirmPassword) {
    alert("password mismatch");
    return;
  }
  try {
    const {user} = await createAuthUserWithEmailAndPassword(
      email,
      password
    );
    await createUserDocFromAuth(user,{displayName:`${firstName},${lastName}`});
    navigate("/login/sign-in");
  } catch (error) {
    console.log("error creating user" + error.message);
  }
};

```

```

// Rendering
return (
  <div className="LoginContainer">
    /* logic for choosing which page to show*/
    {SignInType == "sign-in" ? (
      <SignIn
        InEmail={email}
        emailHandler={handleChange}
        logGoogleUser={logGoogleUser}
        handleLogin={handleSignIn}
      />
    ) : SignInType == "sign-up" ? (
      <SignUp InEmail={email} handleChange={handleChange} handleSubmit={handleSignUp}/>
    ) : null}
  </div>
);
};
// -----
const SignIn = (props) => {
  return (
    <div>
      <Link id="LoginButton" to="/login/sign-up">
        <button className="InvertButton RightButton">Sign Up</button>
      </Link>
      <form className="ColumnStretch">
        <label>Your email</label>
        <TextBar
          name="email"
          placeholder="example@email.com"
          default={props.InEmail}
          autoC="username"
          type="email"
          Handler={props.emailHandler}
        />
        <label>Your password</label>
        <TextBar
          placeholder="Password"
          type="password"
          autoC="current-password"
        />
        <button className="HollowButton LoginButtonPadding" onClick={props.handleLogin}> Login </button>
        <button
          className="HollowButton LoginButtonPadding"
          onClick={props.logGoogleUser}
        />
      </form>
    </div>
  );
};

```

```

};
// -----
const SignUp = (props) => {
  return (
    <div>
      <h1>Create a DEV@DEAKIN Account</h1>
      <form className="LargerSignUp">
        <div className="SignUpField">
          <label>First Name:</label>
          <TextBar
            name="firstName"
            type="textBox"
            Handler={props.handleChange}
          />
        </div>
        <div className="SignUpField">
          <label>Last Name:</label>
          <TextBar
            name="lastName"
            type="textBox"
            Handler={props.handleChange}
          />
        </div>
        <div className="SignUpField">
          <label>Email:</label>
          <TextBar
            placeholder="example@email.com"
            name="email"
            default={props.InEmail}
            type="email"
            Handler={props.handleChange}
          />
        </div>
        <div className="SignUpField">
          <label>Password:</label>
          <TextBar
            name="password"
            placeholder="Password"
            type="password"
          />
        </div>
      </form>
    </div>
  );
};

```

```

67     <label>Confirm Password:</label>
68     <TextBar
69       name="confirmPassword"
70       placeholder="Password"
71       type="password"
72       Handler={props.handleChange}
73     />
74   </div>
75   <button className="HollowButton LoginButtonPadding" onClick={props.handleSubmit}> Login </button>
76 </form>
77 </div>
78 );
79 };
80 // -----
81 // to stop bloat
82 const TextBar = (props) => {
83   return (
84     <input
85       className="StretchText"
86       name={props.name}
87       placeholder={props.placeholder ? props.placeholder : null}
88       defaultValue={props.default ? props.default : null}
89       type={props.type ? props.type : "textbox"}
90       autoComplete={props.autoC}
91       onInput={props.Handler ? props.Handler : null}
92     ></input>
93   );
94 };
95
96 export default LoginPage;
97

```

Add user
↻
⋮

Identifier	Providers	Created ↓	Signed In	User UID
graphix17@bigpond.com	✉	Sep 16, 2022	Sep 16, 2022	D31c9fk0izfYfonSyZ7PQpRSPo32
dddd@mm.com	✉	Sep 16, 2022	Sep 16, 2022	SDCcwHicKeWdfRqYmvDXnoOEb...
ddd@mm.com	✉	Sep 16, 2022	Sep 16, 2022	IOTWwwq2lDctwYEs8be4DNQD6T...
dd@mm.com	✉	Sep 16, 2022	Sep 16, 2022	7fCG2PI5mdWwWwOy7lLEi4wn9i2

Rows per page: 50
1 - 4 of 4

Daily Active Users

Current billing period
Sep 1 - Oct 1 (PDT)

Daily active users

4 peak



Monthly Active Users

Current billing period
Sep 1 - Oct 1 (PDT)

Monthly active users

4

