00.14

JAVASCRIPT.JS
Opslag

javascript.js

PASSWORD

password must match the following: "/^(?!.*\s)(?=.{8,256}$)(?=.*[a-z])(?=.*[A-Z])((?=.*[0-9])|(?=.*[~!@#$%^&*()-=_+[\]{}|;:,./<>?])).*$/"

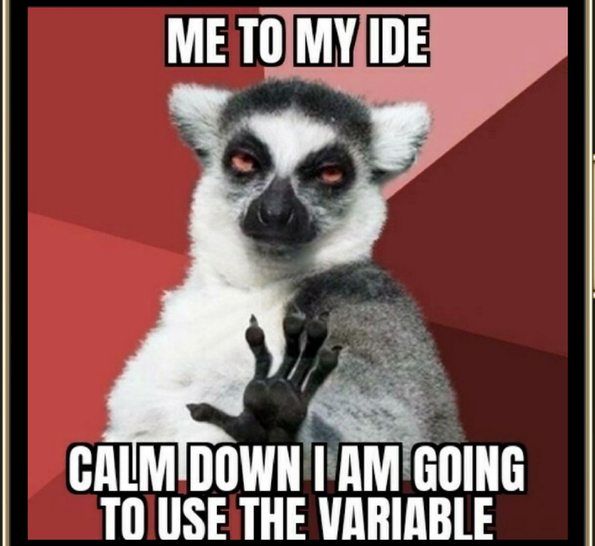
Synes godt om fra **kasialaniecka** og andre
javascript.js 😅... mere
Vis alle 51 kommentarer
for 5 dage siden · **Se oversættelse**

javascript.js

00.15

JAVASCRIPT.JS
Opslag

javascript.js



Synes godt om fra **wolflorena** og andre
javascript.js 🤓... mere
Vis alle 41 kommentarer
31. marts · **Se oversættelse**

javascript.js

00.18

JAVASCRIPT.JS
Opslag

javascript.js

Synes godt om fra **gabriela.rodickova** og andre
javascript.js Good one - JS 🔥🙌... mere
Vis alle 185 kommentarer
7. januar 2023 · **Se oversættelse**

javascript.js

Adam Bertram
@adbertram

Authentication
&
Authorization

localhost:3000/signin

Sign In

Mail: Type your mail...

Password: Type your password...

Sign In

No account? Sign up here.

localhost:3000/signup

Sign Up

Mail: Type your mail...

Password: Type your password...

Sign Up

Already have an account? Sign in here.

Remix Post App

localhost:3000/posts

POSTS | ADD POST | PROFILE

Posts

Rasmus Cederdorff
Senior Lecturer

Exploring the city center of Aarhus

Dan Okkels Brendstrup
Lecturer

A cozy morning with coffee

Rasmus Cederdorff
Senior Lecturer

Serenity of the forest

Maria Louise Bendixen

Anne Kirketerp
Head of Department

Rasmus Cederdorff
Senior Lecturer

localhost:3000/posts/65d1f502f5b3f142ef417bb8

localhost:3000/profile

POSTS | ADD POST | PROFILE

Profile

Name: Rasmus Cederdorff

Title: Senior Lecturer

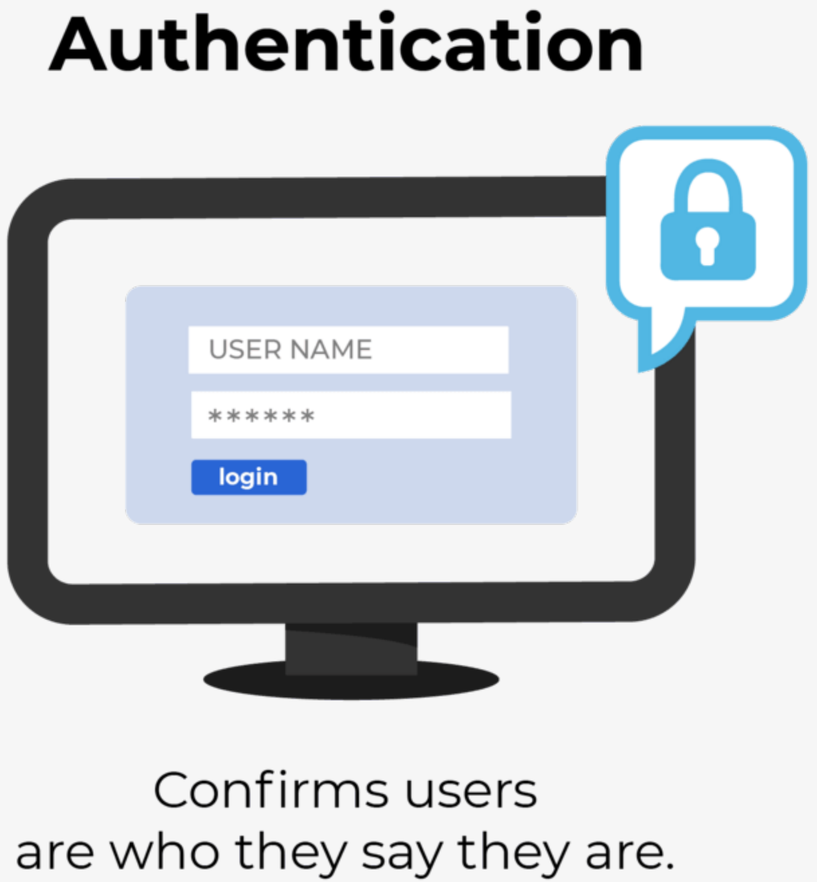
Mail: race@eaaa.dk

Logout

Authentication (godkendelse)

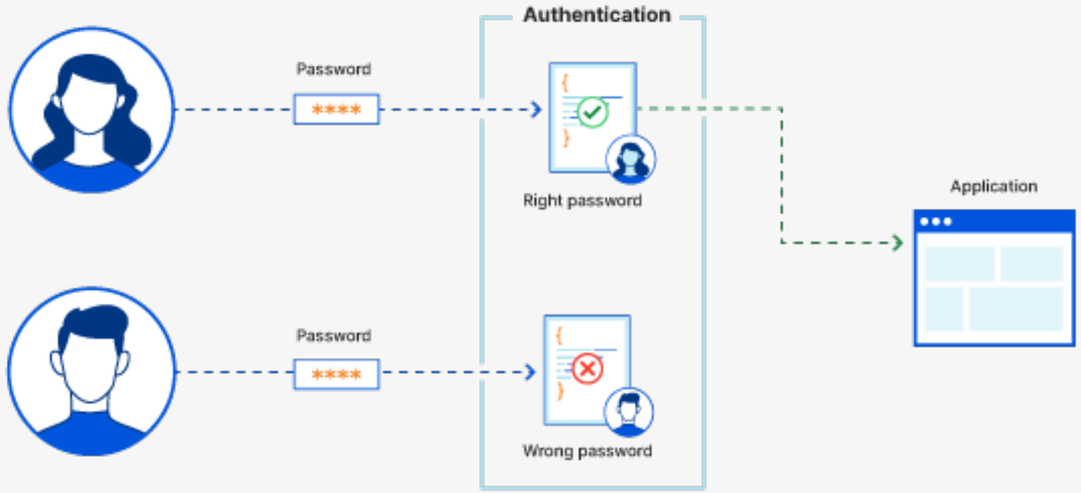
Bekræfter hvem en bruger er.

- Eks.: Logge ind med e-mail og adgangskode, Google-login, Facebook, GitHub osv.
- Hvis godkendelsen lykkes, får brugeren adgang til systemet.



Authentication?

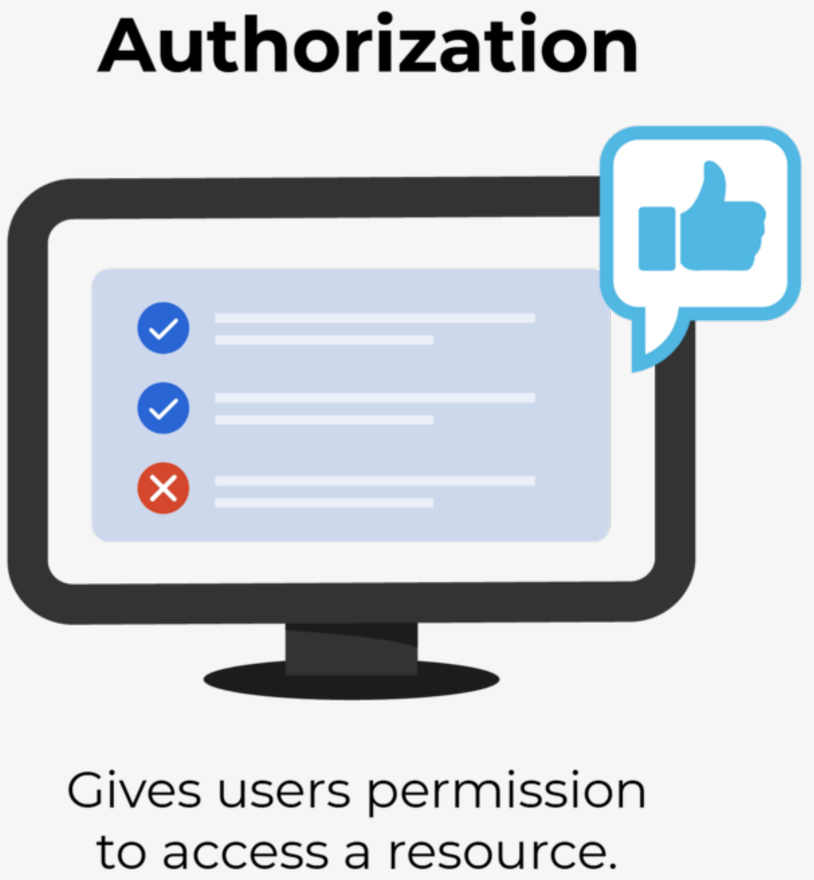
Authentication is the process of **verifying the identity of a user or system**, typically through credentials like **usernames and passwords**, **biometric data**, or security **tokens**, to ensure that individuals are who they claim to be before **granting access** to secure systems or information.



Authorization (autorisering)

Bestemmer hvad en bruger har adgang til.

- Eks.: En admin-bruger kan se og ændre indstillinger, mens en almindelig bruger kun kan se sin egen profil.
- Finder sted **efter** authentication.

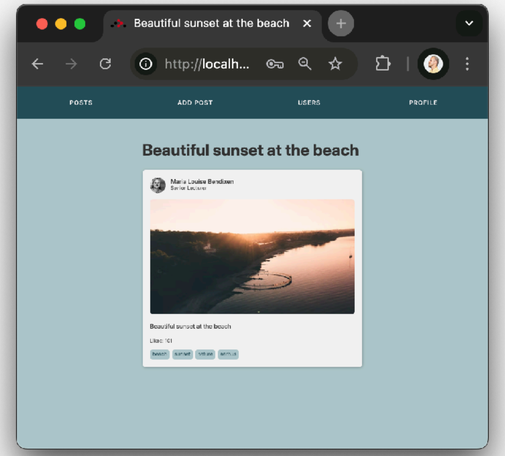


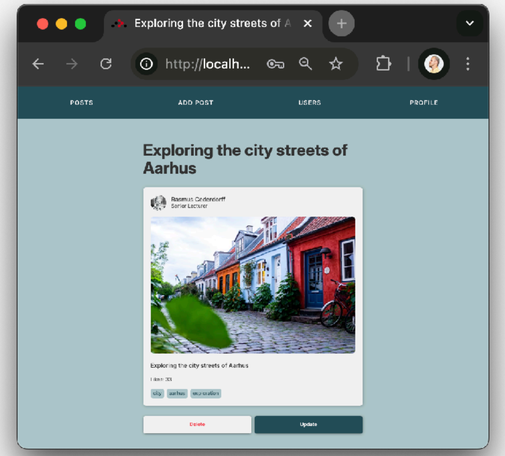
Authentication = "Er du den, du siger, du er?”;

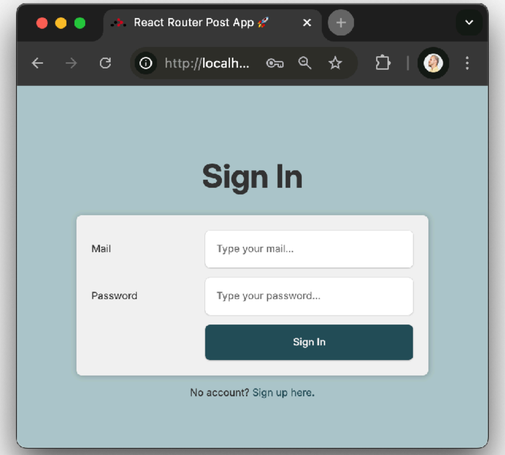
Authorization = "Hvad har du lov til at gøre?”;

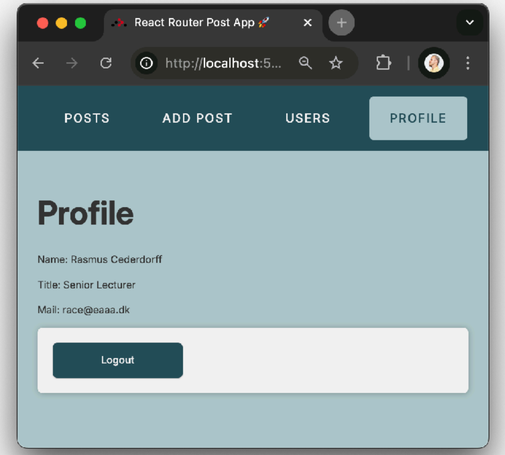
Eksempel 1: Webapp med login

- **Authentication**:
  - Brugeren logger ind med e-mail og adgangskode eller Google/Facebook-login.
- **Authorization**:
  - En almindelig bruger kan kun se sin egen konto. Og kan kun oprette, redigere og slette egne posts.
  - En admin-bruger kan redigere alle brugere og ændre systemindstillinger.









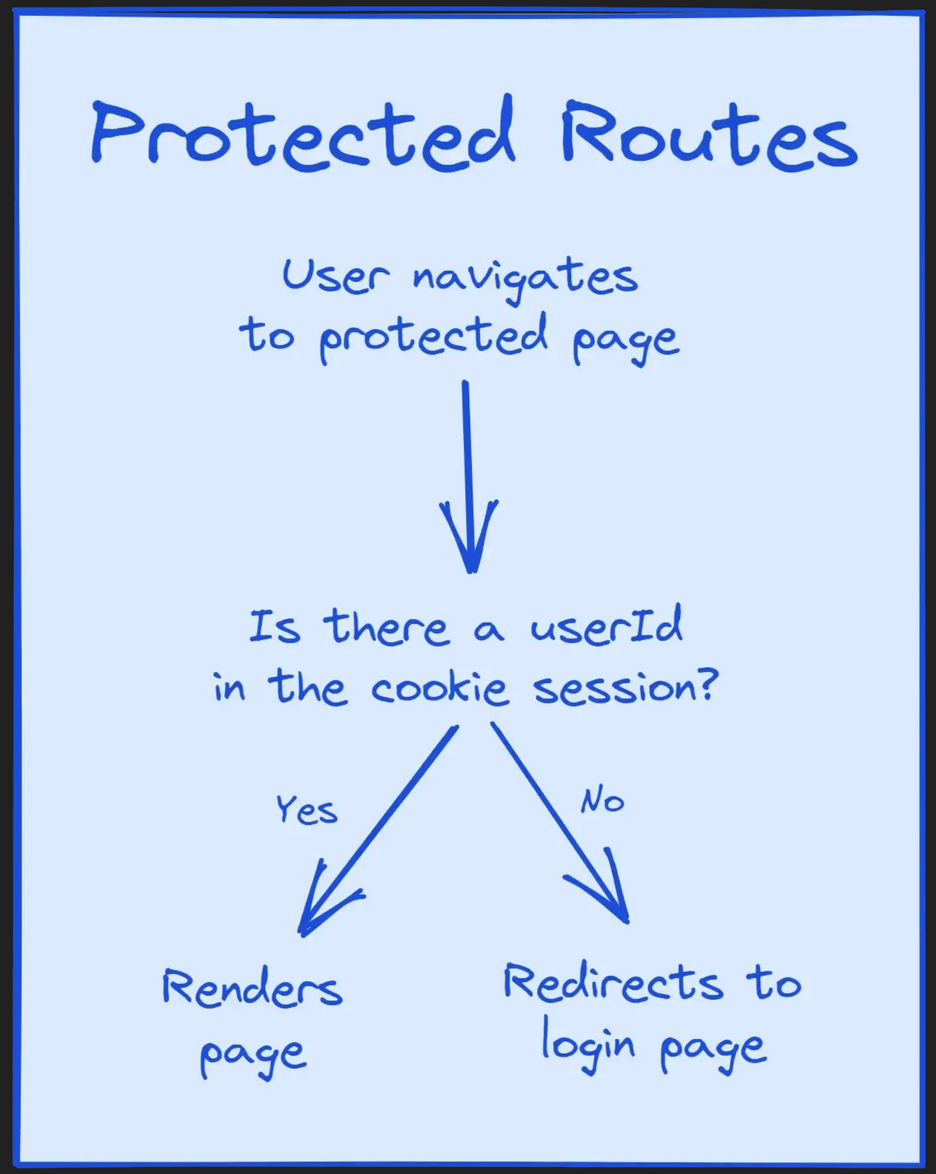
Eksempel 2: GitHub

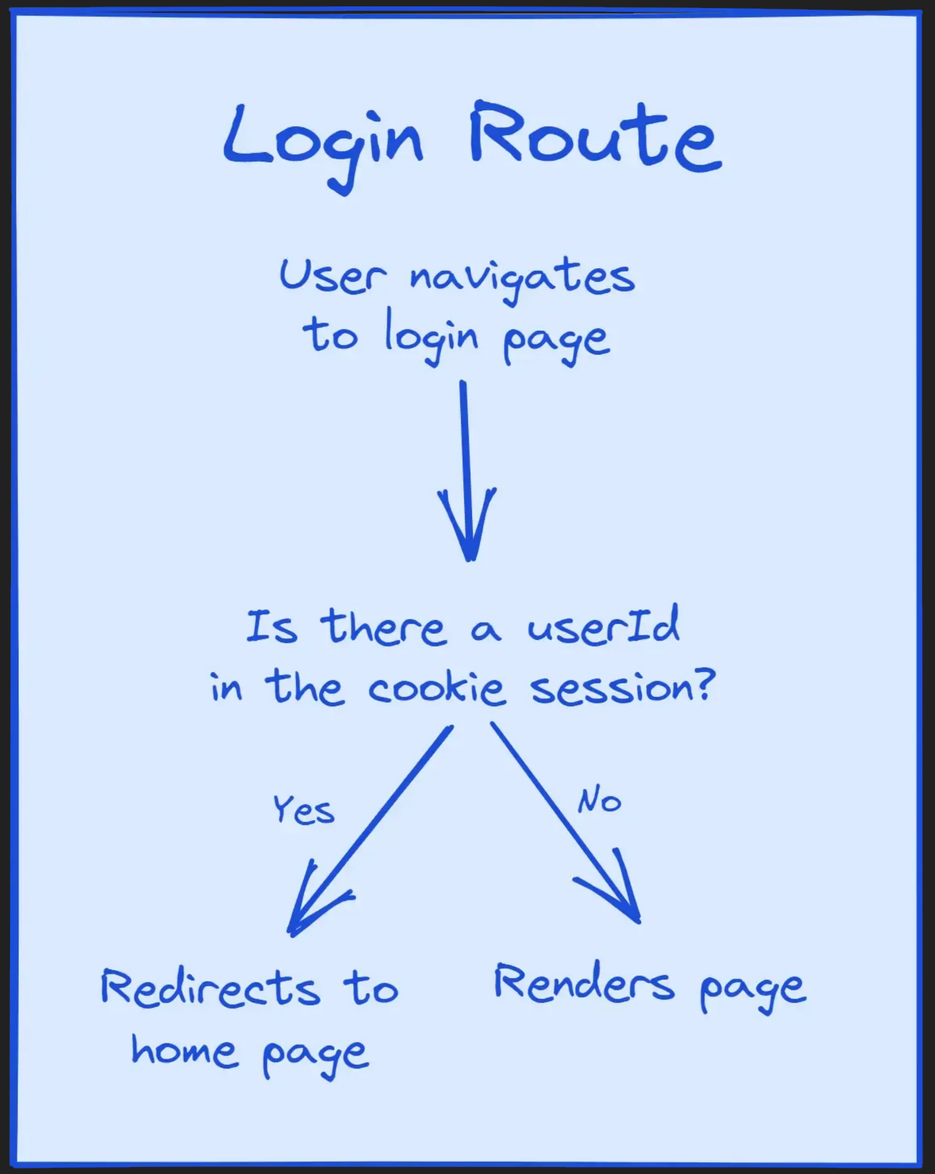
- **Authentication**:
  - Du logger ind på GitHub med dit brugernavn og adgangskode.
- **Authorization**:
  - Du kan læse og ændre dine egne repositories.
  - Du kan kun bidrage til andres repositories, hvis du har fået de rette tilladelser (fx read, write eller admin access).

Eksempel 3:
Firebase Authentication + Firestore Rules

- **Authentication**:
  - Firebase håndterer login med e-mail, Google, Facebook osv.
- **Authorization:**
  - Firestore Security Rules sikrer, at:
    - Brugere kun kan læse/skrive deres egne data.
    - Admins har fuld adgang til databasen.


```
{
  "rules": {
    "users": {
      ".read": true,
      "$user":{
        ".write": "auth != null && auth.uid == $user",  // Only authenticated users
        ".validate": "newData.hasChildren(['name', 'mail', 'image'])",
        "name": { ".validate": "newData.val().length > 0" },
        "mail": { ".validate": "newData.val().length > 0" },
        "image": { ".validate": "newData.val().length > 0"}
      },
      ".indexOn": ["name", "mail", ".value"]
    },
    "posts": {
      ".read": true,
      "$post":{
        ".write": "auth !== null && (auth.uid == newData.child('uid').val() || auth
        ".validate": "newData.hasChildren(['caption', 'image', 'uid'])",
        "caption": { ".validate": "newData.val().length > 0" },
        "image": { ".validate": "newData.val().length > 0"},
        "uid": { ".validate": "newData.val().length > 0"}
      },
      ".indexOn": ["uid"]
    }
  }
}
```

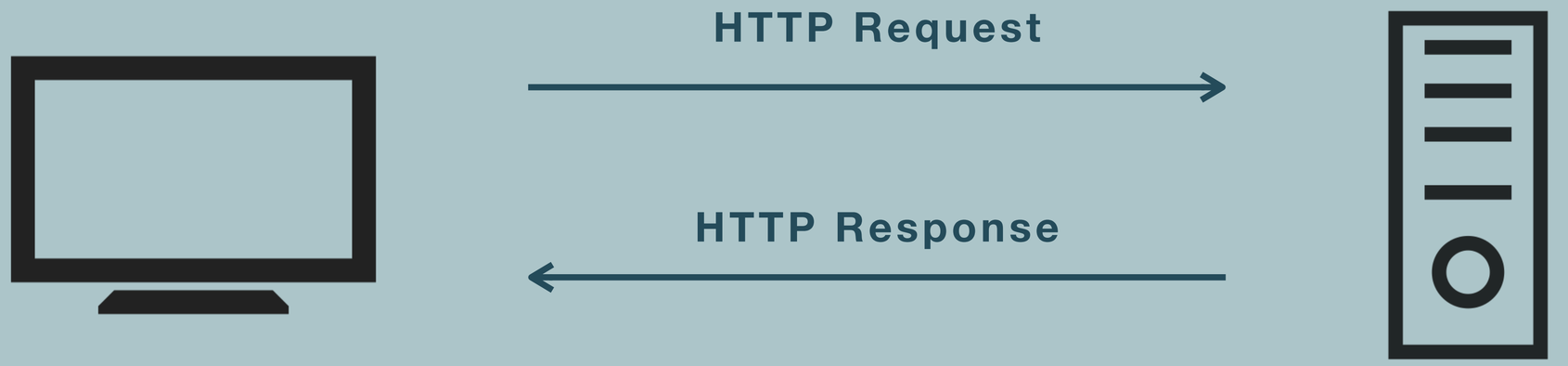




https://www.mattstobbs.com/remix-authentication/

Client-Server Model

Communication between web **clients** and web **servers**.



Client-Server Model

Communication between web **clients** and web **servers**.

Session vs Token Authentication in 100 Seconds

https://www.youtube.com/watch?v=UBUNrFtufWo

Auth.js

Authentication for the Web.

- Et moderne JavaScript authentication-library
- Bruges til at håndtere login, sessioner og sikkerhed
- Understøtter OAuth-providers (Google, GitHub), email og credentials
- Fungerer med Next.js, Remix, SvelteKit m.fl.
- Modulariseret, fleksibelt og nemt at integrere i projekter