

# Din Mægler

Helena Agerbeck Warming WU11

**GitHub:** <https://github.com/rts-cmk-wu12/din-m-gler-th3mxtrixz>

**API:** [Din Mægler](#)

**Brugernavn/Email:** [helena@mail.dk](mailto:helena@mail.dk)

**Password:** 123456

## Brugte libraries:

- [Cookies Next](#)
- [Zod](#)

## Brugte frameworks:

- [Next.js](#)

## Redegørelse for teknologi-stack:

Jeg har valgt at bruge Next.js, da det blandt andet er det, vi har lært, men også fordi jeg føler, det er "lettere" end fx. React. Next.js tilbyder en lettere måde at implementere server-side rendering og routing, hvilket gør det nemt at skabe hurtigere og SEO-venlige applikationer. Desuden tilbyder Next.js en nemmere måde at integrere API-router i samme projekt. Personligt synes jeg også, at Next.js er lettere at læse og lettere at forstå.

## Redegørelse for tredjeparts kode, og personlige ændringer i forhold til det tildelte layout:

### ● Billede Ændring:

- Jeg har valgt at tilføje et andet billede, end det jeg har fået tildelt på Figmaen. Da jeg synes billedet på Figmaen er sløret, hvilket gør at siden ser uprofessionel ud.

Billedekilde : <https://www.nordbohus.no/klassisk-hus>

### ● Custom On Hover:

- Jeg har valgt at tilføje en orange "on hover" ved alle Anchor (<a></a>) tags, da jeg synes det giver en god kontrast til den mørkeblå farve som går igen på siden. Farven giver også en tydelig indikation på, at du hover/markere din mus over noget.

Kilde: [Tailwind CSS](#)

- **Ændring af tekst:**
  - “Søg blandt 158 boliger til salg i 74 butikker” ifølge API'en giver den kun adgang til et bestemt antal boliger, så ved brug af en variabel viser den kun de antal (14) boliger som er tilgængelige, og ikke 158.
- **Input type="range" slider thumb:**
  - Farveændring på "Input type="range"" cirklen (så den matcher Figmaen)  
Kilde: [Input Thumb](#) + Sten
- **Log in funktion:**
  - Ved brug af API'ens token og cookies, kan jeg logge ind med det tildelte brugernavn og password. Ved brug af conditional rendering, kan jeg også visualisere for brugeren om de er logget ind eller ej (log ind bliver til log ud).  
Kilde: [Din Mægler](#) + Patricks hjælp
- **Error handling:**
  - I min "Subscription"-funktion anvender jeg en metode kaldet "Error Handling," som giver brugeren en visuel indikation af, om tilmeldingen til nyhedsbrevet er gennemført.  
Kilder: [Error Handling](#) + [Using promises](#)

Jeg valgte blandt andet ikke at tilføje en "like" knap på mæglernes personlige profil, da API ikke giver noget end-point.

## Fremhævelse af eventuelle særlige punkter til bedømmelse:

```
export default function WishlistButton({homes, token, savedFavorite = []}) {
  const [isWishlist, setIsWishlist] = useState(false)

  function toggleWishlist(e) {
    e.preventDefault()
    setIsWishlist(prevState => !prevState)
    put(homes.id)
  }

  useEffect(() => {
    setIsWishlist(savedFavorite?.includes(homes.id))
  }, [savedFavorite, homes.id])
}
```

```
"use client"
import put from "@actions/put"
import { useRef } from "react"

export default function FavoriteCard({items}){
  const myRef = useRef()

  function handleRemoveFavorite (){
    put(items.id)
    myRef.current.remove()
  }

  return(
    <section ref={myRef} className="flex justify-center w-[75rem] shadow-lg my-[1rem] bg-white relative rounded border-[0.1rem] border-[#D5E8EA] p-6 items-center">
      <div className="flex w-full justify-between">
        <div className="flex">
          <img className="h-[8rem] w-[15rem] mr-10" src={items.images[0].url} />
          <address className="flex flex-col gap-4">
            <span className="font-semibold text-lg">{items.adress1}{items.adress2 ? " " + " " : ""}{items.adress2 ? items.adress2 : null}</span>
            <span className="text-lg">{items.postalcode} {items.city}</span>
            <span className="text-lg">
              <span className="font-semibold text-xl">{items.type}</span> • Ejerdgift: {items.cost} kr.</span>
          </address>
        </div>
        <div className="flex flex-col items-end justify-between">
          <div className="flex justify-between items-start">
            <div className="flex mr-10">
              <span
                className={`mr-6 ${items.energylabel} font-semibold text-[1.4rem] text-white h-[1.8rem] w-[1.8rem] flex items-center justify-center`}
              >{items.energylabel}</span>
              <span className="text-lg">{items.rooms} værelser • {items.livingspace} m²</span>
            </div>
            <span className="font-semibold text-2xl">Kr. {items.price.toLocaleString()}</span>
          </div>
          <button onClick={handleRemoveFavorite} className="bg-[#162A41] h-[3.5rem] w-[12rem] text-white">Ejern fra favoritter</button>
        </div>
      </div>
    </section>
  )
}
```

## Forklaring af kode:

- **Wishlist Button:**
  - Ved brug af “useState” og parametre giver denne kode brugeren mulighed for at tilføje “homes/items” til en side ved brug af “.map()” hvor de derved vil blive vist (det kræver dog du er logget ind).
- **FavoriteCard:**
  - FavoriteCard generer de kort/houses som brugeren har tilføjet til sin ønskeliste. På denne side er der også en funktion som gør at du kan slette dem igen, ved brug af “useRef” som automatisk trigger “handleRemoveFavorite” funktionen.