

Dokumentation

Stacks

I dette afsnit har jeg valgt at redegøre og perspektivere to af de stacks jeg bl.a. har arbejdet med.

Jeg har valgt at arbejde med Reactjs, da det egner sig bedst til frontend udvikling, da det gør det simpelt at lave interaktive user interfaces. Reactjs er et opensource Javascript bibliotek.

Modsat til f.eks. Vue, skriver man i Reactjs, i JSX (JavaScript Expressions) og i Vue er det mere normalt at bruge HTML templates, dog er det en mulighed også at skrive i JSX. Reactjs er den mest brugte JS teknologi, men Vue har en bedre dokumentation.

Fuse.js, som jeg har brugt til at lave Search, er en fuzzy search, mere formelt en approximate string matching, det er en teknik der finder strings der er næsten lig med det givende mønster. Med Fuse.js er det ikke nødvendigt at lave backend.

I stedet for Fuse.js kunne jeg have brugt Lunr.js. som er en lokal search, som virker både med og uden network connection. Det bruger JSON dokumenter.

Arbejdsproces

Jeg startede med at lave en Trello, med de punkter jeg skulle nå inden deadline. Jeg har brugt Trello gennem hele min proces og taget et screenshot hverdag, for at kunne vise hvad jeg fik lavet i løbet af dagen. Jeg har valgt at bruge trello, fordi det er nemt og lige til, det giver også en god ide om hvor man er i sin arbejdsproces.

Valgfrie opgave

Jeg har valgt at arbejde med den valgfrie opgave C - automatiseret deployment. Jeg har valgt at deploye via Vercel.

Torsdag:

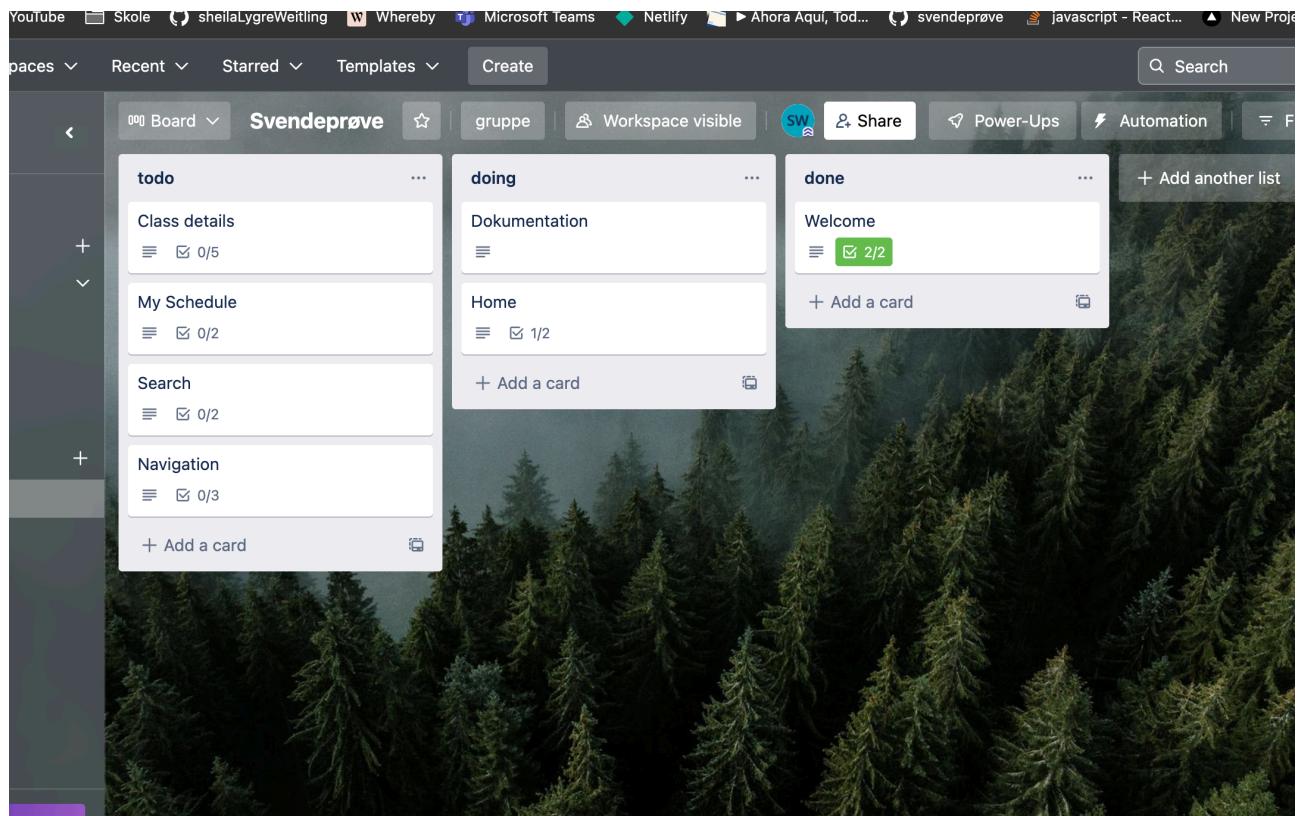
The screenshot shows a Trello board titled "Svendeprøve". The board has three lists: "todo", "doing", and "done". The "todo" list contains cards for "Welcome", "Home", "Class details", "My Schedule", "Search", "Navigation", and "Dokumentation". The "done" list is empty. The background of the board is a photo of a forest.

Jeg lavede min trello, tilføjede cards og skrev kort om dem indeni, samt lavede checklister.

Eks:

The screenshot shows a Trello card for "Welcome" in the "done" list. The card has a description: "Åbne hvergang appen åbnes I bunden skal der være en knap der leder videre til home". Below the description is a checklist with items: "velkommen" (checked), "billeder" (checked), and "knap "home"" (checked). There are buttons for "Hide checked items" and "Delete".

Fredag:

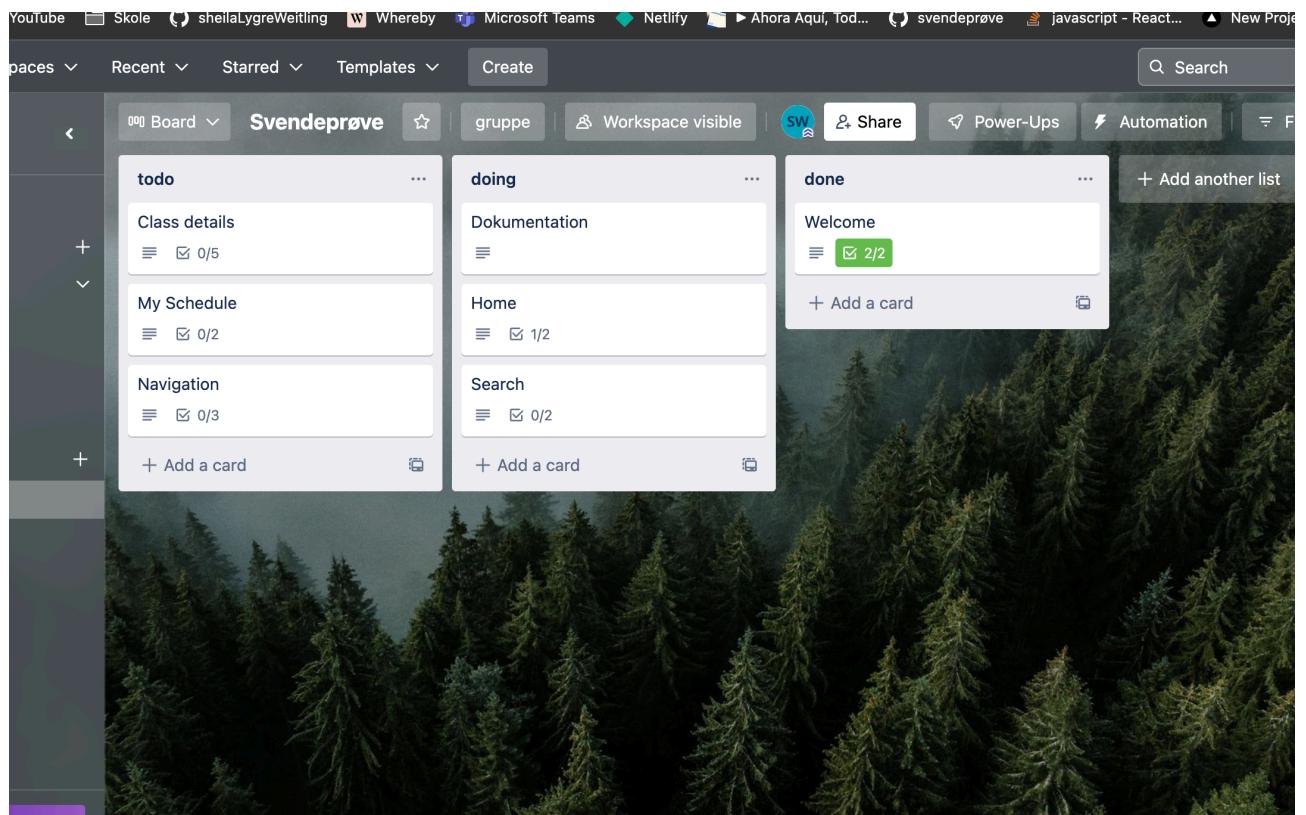


Jeg fik lavet Welcome siden og gjorde klar til at gå i gang med Home siden.

```
return (
  <div>
    <div className='relative h-screen'
      style={{
        backgroundImage: `url(${welcomebackground})`,
        backgroundSize: 'cover',
        backgroundPosition: 'center',
      }>
      <div className='relative'>
        <h1 className='font-bold text-[56px] text-[#f1c40e] p-8'>Believe Yourself</h1>
        <h2 className='font-semibold text-[36px] text-[#fffffe]'>- Train like a pro</h2>
      </div>
    <div className='absolute h-1/2 w-full left-0 bottom-0 z-10'
      style={{
        backgroundImage: `url(${welcomecenter})`,
        backgroundSize: 'cover',
        backgroundPosition: 'center',
      }}>
    </div>
  </div>
)
```

På Welcome siden er de to billeder baggrundsbilleder med en fast højde, for at få dem til at passe til designet.

Mandag:



Efter en veloverstået weekend, gik jeg også i gang med Search delen, da jeg sad lidt fast i Home...

```
const results = fuse.search(searchResults);

const classSearchResults = searchResults ? results.map((result) => result.item) : searchTerm;

const [isVisible, setIsVisible] = useState(false);

function handleSearch({ currentTarget = {} }) {

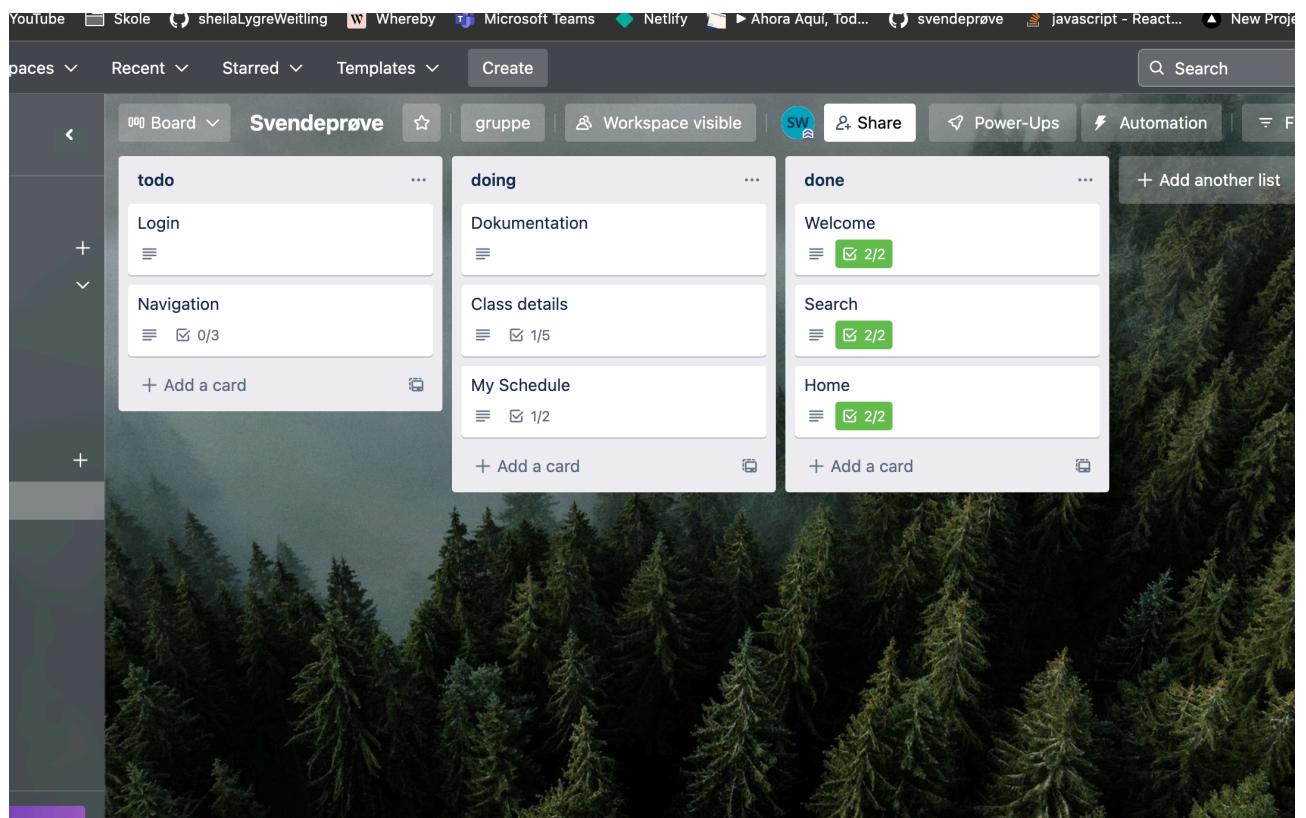
  const { value } = currentTarget;

  setSearchResults(value);

  if (value === "") {
    setIsVisible(false);
  } else {
    setIsVisible(true);
  }
}
```

Det var til denne del af opgaven, jeg brugte Fuse.js.

Tirsdag:



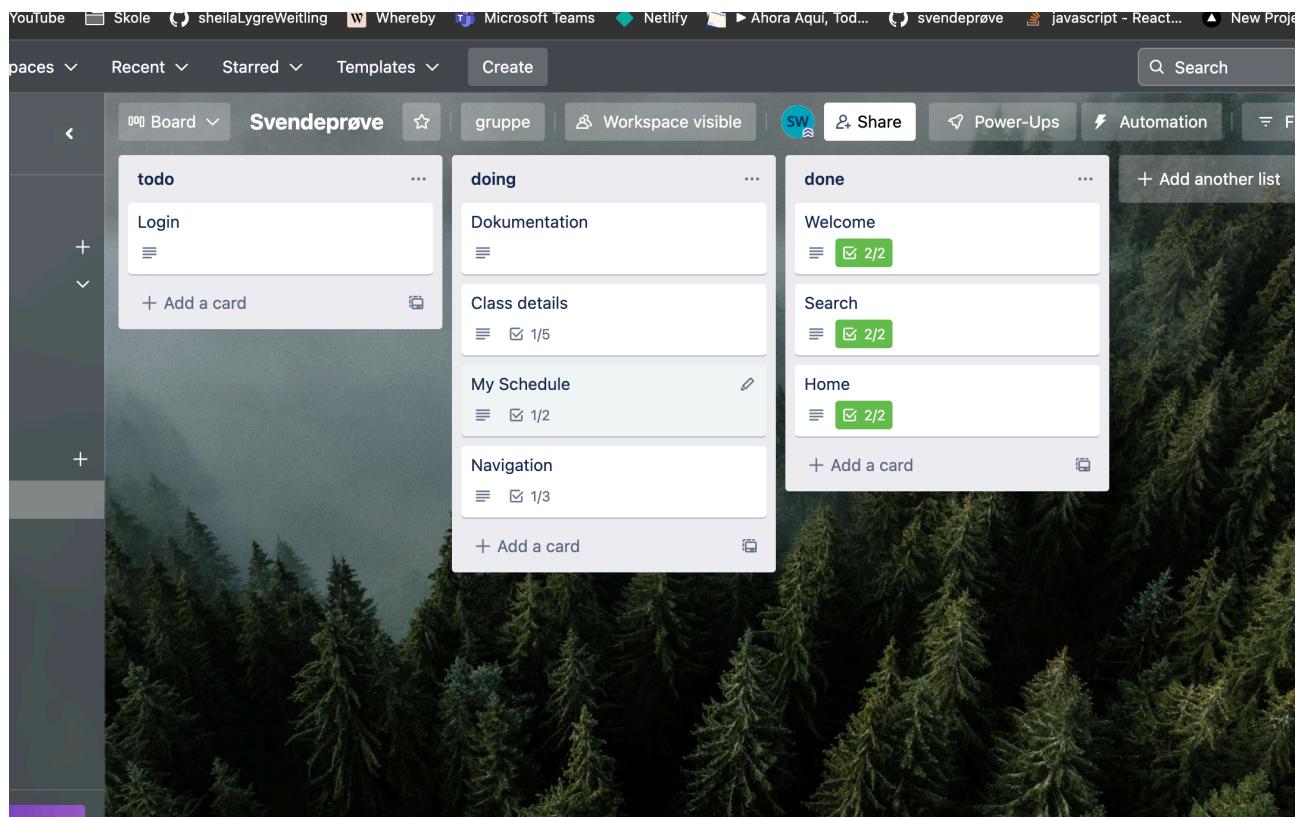
Så kom der fart over feltet og jeg fik lavet både Search og Home færdig, så jeg gik straks i gang med at lave Class Details og My Schedule.

```
fetch('http://localhost:4000/api/v1/classes/4',
  {
    method: 'GET',
  }
)
.then(res => res.json())
.then(data => {
  setSecondSubscribedClass(data)
})
, []
)

return (
  <div>
    <div className='p-4 border m-4 rounded-lg'>
      <h2 className='p-4'>{firstSubscribedClass.className}</h2>
      <p className='p-4'>{firstSubscribedClass.classDay} - {firstSubscribedClass.classTime}</p>
    </div>
    <div className='p-4 border m-4 rounded-lg'>
      <h2 className='p-4'>{secondSubscribedClass.className}</h2>
      <p className='p-4'>{secondSubscribedClass.classDay} - {secondSubscribedClass.classTime}</p>
    </div>
  </div>
)
```

Da jeg ikke nåede at blive færdig med log in delen af opgaven, så er My Schedule tilgængelig og hentet fra API'et.

Onsdag:



Da jeg, som sagt, ikke nåede at lave login, så er de tre: Class Details, My Schedule og Navigation, ikke blevet helt færdige. Dog fik jeg lavet funktionaliteten til bruger menu delen af Navigation.

```
10 const [isNavOpen, setIsNavOpen] = useState(false);
11
12 return (
13   <div>
14     <nav>
15       <section className=' text-[24px]'>
16         <div onClick={() => setIsNavOpen((prev) => !prev)} className=' text-[#9e9e9e] text-[24px]'>
17           <HiMenuAlt3 />
18         </div>
19         <div className={isNavOpen ? "showMenuNav" : "hideMenuNav"}>
20           <div
21             className="absolute top-0 right-0 px-8 py-8 text-[#9e9e9e] text-xl"
22             onClick={() => setIsNavOpen(false)}>
23             <>CgClose />
24           </div>
25           <ul className="flex flex-col items-center justify-between min-h-[250px]">
26             <Link to="/home">
27               <li className=" my-8 ">
28                 <button>Home</button>
29               </li>
```