

# Rapport – Frederik Skrubbeltrang – H1WE080122

<https://github.com/fredeskrubba/stringsonline>

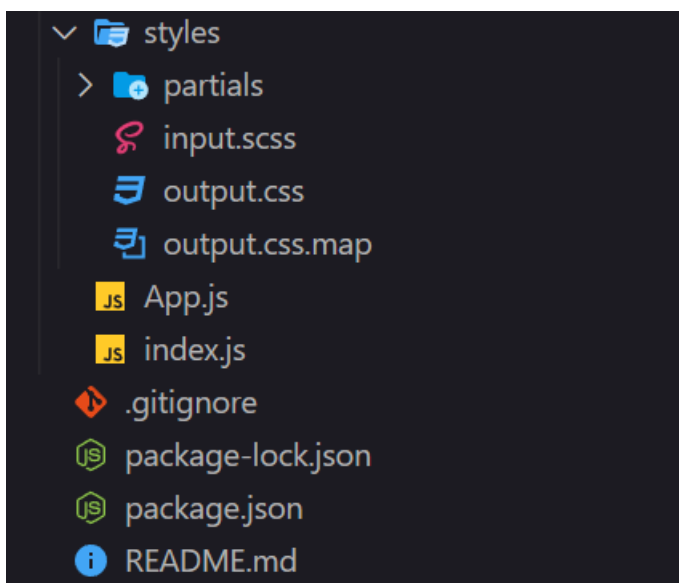
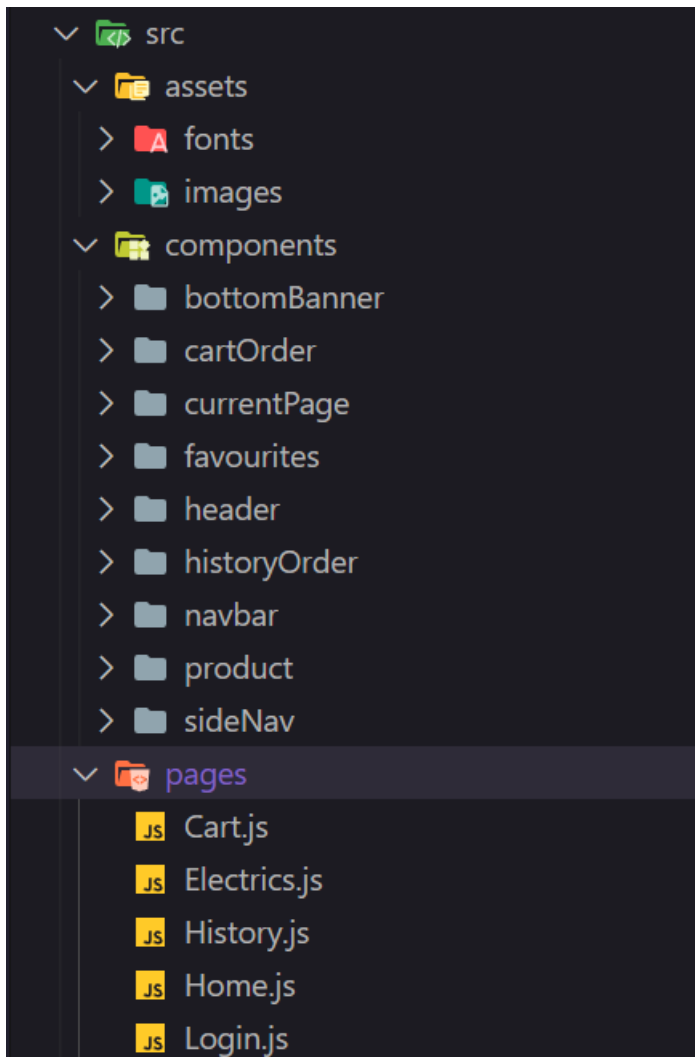
## Vurdering af egen indsats:

I løbet af opgaven har jeg hver dag siddet og arbejdet med opgaven, og gjort mit bedste for at skrive nemt læseligt og vel organiseret kode, hvilket jeg selv synes jeg har løst fyldestgørende. Jeg har inddraget API'et for at få data på forsiden, men pga størrelsen på opgaven, er jeg ikke noget i mål med at lave hver underside færdig, eller fuld funktionel. Jeg har dog bevist at jeg har en fyldestgørende forståelse for HTML, CSS/SCSS og Javascript/React, da alt koden er skrevet i før nævnte. Mit hovedproblem i forhold til at løse opgaven har været størrelsen på opgaven, da jeg synes der er for meget i den til at kunne løse fyldestgørende på en uge.

## Argumentation for valg:

Under opgaven har jeg valgt at benytte SCSS til styling, da det er den teknologi jeg er mest fortrolig med, når det kommer til styling. Derudover har jeg brugt React for at kunne opdele min kode i komponenter, og gøre filstrukturing og koden mere overskuelig. I forhold til State management har jeg benyttet zustand, da jeg godt kan lide hvordan boiler plate er minimalt, og det derfor bidrager til gennemskelig kode. På samme måde har jeg brugt react-wouter til routing, hvilket er en npm pakke inspireret af react-router, men igen har det kun minimalt boilerplate, og kun de funktioner som jeg skal bruge. På den måde er jeg kommet frem til en løsning med ren kode, som gør hvad den skal.

## Særlige punkter til bedømmelse:



Min filstruktur er bygget op således, at hvert component ligger i en mappe, hvor andre components som er relevante ligger sammen med. På samme måde er min styling delt op i

partials, til hvert component. Pages mappen er udelukkende mine route components, og alt medie ligger i assets. Dette har gjort det nemt for mig under arbejdet at finde rundt i projektet.



```
JS favouriteStore.js X
src > components > favourites > JS favouriteStore.js > ...
1  import { create } from 'zustand'
2
3  const usefavouriteStore = create((set) => ({
4    favourites: undefined,
5    fetchFavourites: async(endpoint) => {
6      const response = await fetch(endpoint)
7      const data = await response.json()
8      set({favourites: data})
9    }
10
11  }))
12
13  export default usefavouriteStore
```

I koden her benytter jeg zustand til at fetche API'et, og opdatere min store, som holder styr på state jeg bruger på forsiden's favoritter sektion. Det viser at jeg både er i stand til at hente data fra api, og implementerer det på frontend ved brug af state management og react.