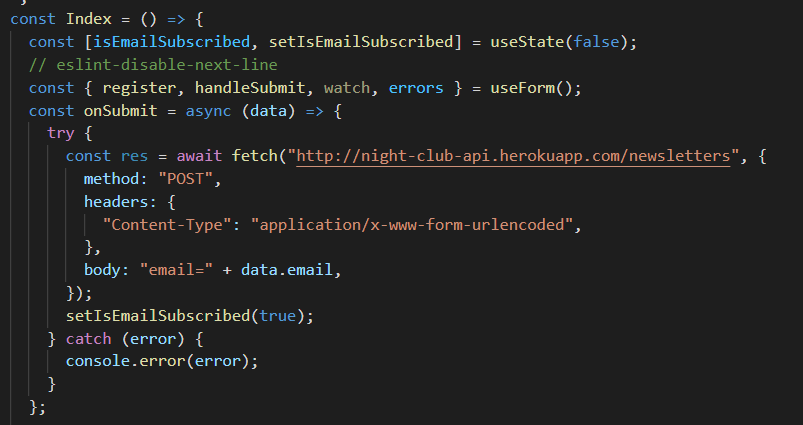
**Henrik Mundt Milo - Terminsprøve 2020**

**Teknologier og begrundelse**

* React: Jeg har brugt React fordi det er et komponent baseret Javascript Framework der tilbyder fleksibilitet. Der er mange dependencies i forbindelse med React som forbedrer min arbejdsgang.
* Recoil: Jeg har brugt Recoil som Global State Management library. Det tilbyder lidt mere funktionalitet i forhold til React useContext API, og det er en relative lille pakke.
* Styled-components: Jeg har brugt Styled-components fordi pakken tilbyder Scss styling med Javascript funktionalitet som betingelser, og matematiske udregninger i styling, genbrugning af Komponenters styles og mere.
* Javascript: Jeg arbejder med Javascript i forbindelse med React som er bygget på Javascript, og det er det kodesprog som vi har arbejde mest med i undervisningen.

**Teknisk dokumentation**



Billedet ovenfor er en lille del at min Form validering på Newsletter sektionen på forsiden. Jeg har 2 hooks som re-render komponent når deres værdi opdater. Vi har min isEmailSubscribed som er en Boolean som jeg bruger i forbindelse med Form interaktivitet, efter min post request bliver den true og så kan du ikke længerer bruge formen, dens styling bliver halv gennemsigtig og du får en besked tilbage at Email er Subscribed.

useForm hook bruger jeg i forbindelse med validering af form, fejlbesked håndtering og funktionaliteten på Submit.



Længerer nede kan i se min Form, og den har en attribut onSubmit som kalder på handleSubmit fra useForm, og den har et argument som er funktionen den vil kalde og sende formData til hvis formen er valid.

På input kan man se at jeg har brugt register funktionen i ref som referer til dom elementet, og i denne funktion kan jeg håndter input validering. Udfor en masse objekter kan man lave validering med 2 properties, value som er dens værdi, og message som er den fejlbesked som kommer fra errors objektet i useForm.

Og til sidst har jeg nogle betingelser som tilføjer noget styling, og disable input hvis isEmailSubscribed er true.

Med alle disse funktioner kan du lave simple til komplekse former.

**Soft Dokumentation**

Når du først kommer ind på forsiden ser du et stort Hero biled som velkommer dig til hjemmesiden. Nedeunder har vi en navigationsbar der som du kan bruge til at navigere rundt på siden.

Welcome sektionen giver dig 3 kort, hvis du hover kortene vil du se oplysninger omkring kortene med en blødeffekt.

Sektionen nedeunder er Events, det er de senete Events, hvor du kan skifte imellem dem. Der dato, og område oplysninger, og hvis du hover over Event vil du se en fed animation, og vil slide ind flere oplysninger omkring Eventet og en knap som sender dig viderer til at book elementet.

Galleriet nedeunder kan du klikke på, for at få flere oplysninger omkring det biled. Hvor du så kan skiftet imellem billed.

Testemonials kan du også skifte imellem anbefaling og se deres kommentar, billed og sociale links.

Recent blogs har de seneste opslag. De sender dig videre til en side med alle oplysninger og et kommentarfelt hvor du kan kommenter og svare på kommentar.

Newsletters i bunden er en Form som giver dig muligheden til at subscribe til newsletter.

Footer indeholder virksomhed informationer.

**Arbejdsprocessen**

Jeg har brugt create-react-app, og brugt et andet design mønstre end jeg er vant til. Det har ikke været den bedste oplevelse, jeg har brugt 3 timer på at sætte mit projekt op, og design mønstre gjorde mappe strukturen forvirrende som i sidste ende forhindre mig i at være så effektivt som muligt.

Carousel dependencien var begrænset, og gjorde at krævede funktioner i opgaven var ikke muligt uden en dårlig hack metode som jeg helst vil ungå, jeg har fundet ud af at flickity er den bedste carousel løsning.