

Mappedokument

Oppgave 3 – Register og Admin HTML sider

Linking til JavaScript fra oppgave 1

Å linke til JavaScript filene i Oppgave 1 uten å flytte dem gikk fint, da det har vært innøvd av referering til filstier innen Webutvikling. I Webutvikling var hovedfokuset på stilark, men samme prinsippet kan gjelde for script også, med et par småendringer.

Jeg måtte søke opp hvordan jeg skulle få referere til begge script filene, og kom fram til å bruke modulær tilkobling . Ved bruk av ES Modules kunne jeg linke register.js til actions.js ved bruk av «export» før variabler, og da bruke actions.js som main filen til javascriptet (Mozilla.org (u.å)).

Register.html - Utforming

FILTRERING AV AVDELINGER OG STILLINGER:

Mesteparten av koden fra Oppgave 1 var brukbar direkte fra «jukset» i den oppgaven, men filtreringsknappene ville ikke funke, så har måtte det til med en endret løsning. Ved å slå sammen funksjonene 3 og 5, som skulle hver for seg filtrere etter stillinger og avdelinger, slo jeg disse sammen til en funksjon.

Med en rekke if-else tester, kunne jeg filtrere direkte etter angitt parameter, ettersom at jeg har laget to variabler for avdelingene, som slår sammen rollene i role-arrayet til sine gitte verdier (OpenAI, 2025, 09. desember versjon 1). Ved å angi [data-role] til knappene i HTML dokumentet, fant jeg at det gjorde koden mye lettere ved å bruke target.matches, istedenfor en separat eventListener for hver enkelt knapp.

Ved å gi staff-list, id-en hvor alle ansattkort printes ut, forskjellig klasse i register.html og admin.html, kunne jeg bruke disse klassene til å deklarere hvor jeg skulle ha delete-btn synlig og tilgjengelig i dokumentet. Dette gjorde slik at jeg kunne fortsatt fylle den inn slik som createStaffCard() funksjonen gjør, men knappen er kun synlig innen admin.html, hvor den skal være.

Kurslisten var forholdsvis problemfri, utenom å måtte finne rekkefølgen den funksjonen kjørte, siden den skapte problemer for brukerinput først i admin.html. løsningen var heller ikke relatert til selve funksjonen, men html feil ved admin.html.

Admin.html - Utforming

BRUKERINPUT OG SLETTE ANSATTE:

For å få brukerinput til å funke, måtte jeg endre en del på funksjonen. Første jeg ville gjøre, var å lage en dynamisk dropdown meny, som inneholdt alle tilgjengelige stillinger. Disse valgene i denne menyen kommer fra role-arrayet, og hvis dette arrayet blir større, med flere stillinger, er koden satt opp slik at disse rollene vil automatisk fylles inn i dropdown menyen (Alex Turpin, 2012). For at dette skulle funke, måtte jeg ta dette ut av funksjonen som lager en ansatt av brukerinput, og gjøre en separat funksjon av det først (OpenAI, 2025 09. desember versjon 2). Jeg endte opp med å få HTMLSelectElement istedenfor valgte verdi, så med litt undersøkelse, fant jeg problemet (N. Campolungo, 2013).

Nå som dropdown menyen funker, tok jeg å hentet alle input values å puttet dem direkte inn i et nytt objekt som skal bli pushet inn i staff-arrayet, altså skippet over å lage variabler for alle input values. Unntaket var kursansvar og stilling. Stilling måtte ha en variabel for å kunne motta string value av valgte stilling utifra dropdown menyen, og kursansvar har en egen if else test med ternary operators som sier ifra om stillingen er Professor eller Lektor. Grunnen til dette er at du ikke skal kunne gi en ansatt med stilling innenfor administrasjon et kursansvar med et uhell. Nå, med denne testen, selv om det testes inn et kurs, hvis den valgte stillingen ikke er Professor eller Lektor, vil det fortsatt printes ut «Ingen kursansvar», siden administrasjon ikke har det (OpenAI, 2025, 08. desember versjon).

Av koden jeg fikk laget i Oppgave 1, inneholdt det en del overfladisk kode, som nå er kommentert vekk og erstattet med et enklere oppsett. Knappen «Legg til ansatt», med eventListener på, kjører så en funksjon som kjører funksjonen som skaper ansattkort en gang til, hvor da den nye ansatte er pushet inn i staff-arrayet og vil nå følge på den etablerte listen.

Funksjonen for å slette en ansatt, som funnet i Oppgave 1, etter litt tilpassing av klasse- og id-navn, funket så fort som brukerinput funksjonen var ferdig. Delete-btn knyttet til hvert ansattkort finner index-nummeret til det gitte ansattkortet, og fjerner så dette «objektet» fra staff-arrayet, som gjør det så at det nå er ute av systemet. Med samme prinsipp som knappen for å legge til en ansatt, er en lignende funksjon på slett ansatt knappen, men funksjonen som kjøres er eneste forskjellen.

CSS – Design

Jeg har puttet stilarket som skal kjøre på begge html-filene i en egen mappe kalt «assets», for en renere mappestruktur. Både register- og admin.html bruker samme stilark, hvor forskjellene er kun gitt av forskjeller på navn i klasser og id-er.

For å få brukerinput og slett ansatt til å funke, måtte funksjonen som kjører ut alle kursene også eksistere på admin.html, men som vist på eksempelvideo av Oppgave 3, var det ingen synlig kursliste i admin.html. Ved å gi outputen for denne funksjonen, altså sectionen hvor kursliste printes ut, et ekstra klassenavn i admin.html, kunne jeg ta i bruk css for å skjule denne listen ved å gi den gitte klassen, som ble kalt «hidden», `display: none`.

Ved å gjøre som nevnt over, fikk jeg koden til å kjøre optimalt, uten noe problem, og så stilrent som jeg fikk det til.

Jeg tok også å endret plassering av slett ansatt knappen, slik at den er lettere synlig og vanskeligere å klikke på ved uhell, skulle dette vært et reelt register (L. Anonymouse, 2020).

Referanseliste

- OpenAI. (2025). *ChatGPT (08. desember versjon)*. [Stor språkmodell]. #Oppd1
 - <https://chatgpt.com/share/6936ccc9-03c4-8009-afe5-4d03d70f0c3b>
- OpenAI. (2025). *ChatGPT (09. desember versjon 1)*. [Stor språkmodell]. #Oppd3
 - <https://chatgpt.com/share/6937f8cb-c388-8009-92df-80a82082dab5>
- OpenAI. (2025). *ChatGPT (09. desember versjon 2)*. [Stor språkmodell]. #Oppd3
 - <https://chatgpt.com/share/69383c3f-c030-8009-8b8b-e88a321ddd3d>
- Google. (2025). *Gemini (08. desember versjon)* [Stor språkmodell]. #Oppd1
 - <https://gemini.google.com/share/e9c11f4d6ff0>
- Google. (2025). *Gemini (09. desember versjon)* [Stor språkmodell]. #Oppd3
 - <https://share.google/aimode/p8BeeClwxRlkEoSLx>
- W3 Schools. (u.å.). *CSS Text Indentation and Spacing*. (Text Spacing). Hentet 09. desember 2025 fra
 - https://www.w3schools.com/css/css_text_spacing.asp
- W3 Schools. (u.å.). *CSS Forms*. Hentet 09. desember 2025 fra
 - https://www.w3schools.com/css/css_form.asp
- Mozilla.org (u.å.) *JavaScript Modules*. Hentet. 03. desember 2025 fra
 - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules>
- Turpin, Alex (2012, 27. mars) JavaScript – populate dropdown list with array. *StackOverflow*.
 - <https://stackoverflow.com/questions/9895082/javascript-populate-drop-down-list-with-array>
- Lovecat, Anonymouse (2020, 23. august) How to position a button on website using html/css. *StackOverflow*.
 - <https://stackoverflow.com/questions/63541670/how-to-position-a-button-on-website-using-html-css>
- Campolungo, Niccolo (2013, 21. mai). JavaScript HTML object HTMLSelectElement. *StackOverflow*.
 - <https://stackoverflow.com/questions/16676679/javascript-html-object-htmlselectelement>
- Bro Code (2023, 15. oktober). *How to accept JavaScript USER INPUT in 5 minutes*. [Video]. Youtube.
 - https://www.youtube.com/watch?v=JeXqaKeJSRI&list=PLZPZq0r_RZOO1zkgO4bldfuLpizCeHYKv&index=4