

# Mappedokument

## Forutsetninger

Jeg valget å begynne med oppgave 3, så jeg lagde admin.html og register.html. Deretter så lager jeg javascripten til register.js og action.js meg hjelp av KI. Også kobler jeg dem til funksjonen inne i html sidene mine.

## HjelpeMidler

Oppgave 1 og 3 Register.js og action.js, har KI løst . Jeg laget først register.html etter det så lagde jeg admin.html. register.js brukte jeg Lag variabel med array (2025) som gir meg to funksjoner som heter postions og ansatte. Postions er array med stilling som har professor, lektor, vaktmester, dekan og rektor.

```
const positions = ["Professor", "Dekan", "Rektor", "Vaktmester", "Lektor"];
```

En array med informasjon til ansatte hva de heter, hvilket kontor de har, stilling og hvis de professor eller lektor skal de være kursansvarlig. Vis du ikke kursansvarlig så skal de ingenting.

```
{ lastname: "Moen",
  firstname: "Randi",
  office: "v003",
  position: 3,
  email: "randi.moen@hiøf.no",
  KursListe: []
},

{ lastname: "Berg",
  firstname: "Ola",
  office: "P301",
  position: 0,
  email: "ola.berg@hiøf.no",
  KursListe: ["Programmering 1", "Webutvikling"]
},
```

Action.js brukte jeg Javascript register.js løsning (2025), W3Schools.com preventDefault() Event Method (u.å) og JavaScript String indexOf() Method (u.å.). I Denne javascripten laget jeg 8 funksjoner som skal kobles til både register Og admin.html.

Den første funksjonen heter Lagansattkort som jeg ba Javascript register.js løsning. (2025) Om å lage en ansatt der den genererer html fra data i ansattobjekt som bruker en betinget metode som sjekker om listen finnes og de elementene. Ki brukte map for å gå gjennom om kursene ligger inne der og vises som en standard tekst.

```
// 1) Ansattkort
function lagAnsattkort(i) {
  const e = ansatte[i];
  if (!e) return "Ansatt ikke funnet";
// #OPPD3 denne la jeg til for at begge html har med kurs i nettsiden
  const KursListe = e.KursListe || [];
  const kurs = KursListe.length
    ? "<ul>" + KursListe.map(k => `<li>${k}</li>`).join("") + "</ul>"
    : "Ingen kursansvar";

  return `
<article>
  <h3>${e.firstname} ${e.lastname}</h3>
  <p>Stilling: ${positions[e.position]}</p>
  <p>Kontor: ${e.office}</p>
  <p>E-post: ${e.email}</p>
  ${[kurs]}
</article>
`;
```

Javascript register.js løsning (2025) jeg ba den lagde en funksjon som heter listAllEmployees som skal ha med map for å gjennom lista, og bruker join til slå sammen til en ekle html streng

```
// 2) Vis alle ansatte
function listAllEmployees() {
  return ansatte.map(_, i) => lagAnsattkort(i)).join("")
}
```

Javascript register.js løsning (2025) ba jeg den lage en funksjon som filterer stilling og den en ga meg en kode. Men jeg vil at den skulle bruke indexOf det brukes til å finne index i et array eller element. Jeg la til en kode so jeg skrev selv som heter om jeg kunne bruke den og Ki skal ja. Det kan du se i prompt. Jeg fant denne lenken JavaScript String indexOf() Method (u.å.) for å forstå hva den gjør.

```
// 3) Filtrer etter stilling
function filterByPosition(stillingsNavn) [
  const idx = positions.indexOf(stillingsNavn);
  return ansatte.map((e, i) =>
    e.position === idx ? lagAnsattkort(i) : ""
  ).join("");
]
```

Javascript register.js løsning (2025) den løste funksjonen som henter alle kursene. Det brukte en metode som heter flatmap og den henter ut alle kurslistene fra ansatte og slår dem sammen til en samlet liste. Og bruker new set for å filterer bort duplikater.

```
function getAllCourses() {
  return [...new Set(ansatte.flatMap(e => e.KursListe || []))];
```

Jeg har lag denne funksjonen der den viser kursene til de ansatte som har kurs eller ikke. jeg brukte JavaScript Array flatMap() Method (u.å.) for å samle inn å slå sammen alle kurslistene i samlet liste. Bruker (Mozilla Developer Network, 2025) for å fjerne dupikater. Når listen over unike kurs er klar, bygges HTML-innholdet ved hjelp av template literals, hvor hvert kurs gjøres om til et listeelement ved hjelp av map og deretter settes sammen til én sammenhengende tekst med join. Til slutt settes den genererte HTML-en inn i dokumentet ved å bruke innerHTML på elementet med id "kurs".

```
function showAllCourses() {
  const kurs = ansatte.flatMap(a => a.KursListe || []);
  const unike = [...new Set(kurs)];

  const html = `
    <h2>Kursoversikt</h2>
    <ul>
      ${unike.map(k => `<li>${k}</li>`).join("")}
    </ul>
  `;

  document.getElementById("kurs").innerHTML = html;
}
```

*Javascript register.js løsning.* (2025) jeg ba om løse den for meg. Jeg klarte å fikse denne koden. Det er funksjonen som skal vise ansatte. Brukte map til å gjennom å gå hver ansatt og lage lage en HTML-tekst som inneholder navn, stilling, kontor, e-post og kurslisten. For kurs brukes (a.KursListe || []).join(", "), som betyr at hvis kurslisten finnes, blir den gjort om til en kommaseparert tekst, og hvis ikke vises "Ingen". Til slutt brukes join for å samle inn alle de generete html og bruker output.innerHTML sånn at det kan vises i html siden.

```

function VisAnsatte() {
    output.innerHTML = ansatte
        .map((a, i) => `
            <article>
                <h3>${a.firstname} ${a.lastname}</h3>
                <p>Stilling: ${positions[a.position]}</p>
                <p>Kontor: ${a.office}</p>
                <p>E-post: ${a.email}</p>

                <p>Kurs: ${({a.KursListe || []}).join(", ")} || "Ingen"</p>
                <button class="del" data-i="${i}">Slett ansatt</button>
            </article>
        `)
        .join("");
}

```

*Javascript register.js løsning.* (2025) ville at KI skulle løse den. Men det endte opp meg jeg skrev koden selv jeg brukte W3Schools.com. (u.å.). preventDefault() Event Method slik at nettsiden ikke lastes seg på nytt når funksjonen kjøres. Etter det hentet jeg teksten fra inputfelt for kurs, og den deles opp til en liste ved hjelp av spilt. Det denne metoden gjør at lager en array basert på et komma som skillevegg. Brukte push som pusher nye objekter inn på slutten av listen. Metoden resetform for å tømme skjemaet.

```

function leggTilAnsatt(e) {
    e.preventDefault();

    const kursTekst = document.getElementById("kurs").value.trim();
    const KursListe = kursTekst
        ? kursTekst.split(",").map(k => k.trim()).filter(Boolean)
        : [];

    ansatte.push({
        firstname: Fornavn.value,
        lastname: Etternavn.value,
        email: email.value,
        office: kontor.value,
        position: Number(stilling.value),

        KursListe: KursListe
    });

    addForm.reset();
    VisAnsatte();
}

```

*Javascript register.js løsning.* (2025) KI løste denne funksjonen som sletter ansatte. Jeg ba den bruke splice som sletter et element på angitte index. Når den ansatte bli fjernet brukes visansatte på nytt for å oppdatere nettleseren så at den ikke finnes lengre.

```
...function slettAnsatt(i) {
  ansatte.splice(i, 1);
  VisAnsatte();
}
```

*Javascript register.js løsning.* (2025) jeg ga den en kode jeg skrev selv med det visste seg at jeg har mange feil i den og KI sa til hvor jeg hadde mine feil. Når nettsiden er ferdig lastet inn, kjører koden inne i DOMContentLoaded. Først fylles nedtrekksmenyen for stilling ved å bruke map til å lage option-elementer basert på listen positions. Deretter settes leggtilAnsatt som funksjonen som skal kjøre når skjemaet sendes inn. Videre lytter koden etter klikk inne i output-området; hvis elementet som ble klikket på har klassen "del", betyr det at det er en slett-knapp, og da hentes riktig indeks fra dataset og slettansatt kjøres. Til slutt kalles visansatte for å vise eksisterende ansatte på siden med en gang siden lastes.

```
document.addEventListener("DOMContentLoaded", () => {
  stilling.innerHTML = positions.map((p, i) => `<option value="${i}">${p}</option>`).join("");
  addForm.onsubmit = leggTilAnsatt;

  output.addEventListener("click", e => {
    if (e.target.classList.contains("del")) {
      slettAnsatt(Number(e.target.dataset.i));
    }
  });
  VisAnsatte();
});
```

KI ga meg forslag:

- Testet alt inne nettleseren for å se om det var riktig
- Jeg kobla sammen html lenkene
- Kalte funksjonen mine navn
- Om det var feil så fiksa det KI det

# Referanseliste

## KI-Samtaler:

*Javascript register.js løsning.* (2025). <https://chatgpt.com/share/693b052d-4a5c-8001-b1a6-dd567edad2ea>

*Lag variabel med array.* (2025). <https://chatgpt.com/share/693b0567-2484-8001-84f3-56e3d9efa7d5>

## Nettsider med info

*W3Schools.com.* (u.å.). *preventDefault() Event Method.*

W3Schools. [https://www.w3schools.com/jsref/event\\_preventdefault.asp](https://www.w3schools.com/jsref/event_preventdefault.asp)

*JavaScript String indexOf() Method* (u.å.).

W3Schools. [https://www.w3schools.com/jsref/jsref\\_indexof.asp](https://www.w3schools.com/jsref/jsref_indexof.asp)

*JavaScript Array flatMap() Method* (u.å.).

W3Schools. [https://www.w3schools.com/jsref/jsref\\_array\\_flatmap.asp](https://www.w3schools.com/jsref/jsref_array_flatmap.asp)

*Mozilla Developer Network.* (2025, December 9). *Set – JavaScript.* MDN Web Docs.

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Set](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set)

*Mozilla Developer Network.* (2025, November 12). *Template literals (Template strings) –*

*JavaScript.* MDN Web Docs. [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template\\_literals](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals)

*Mozilla Developer Network.* (2025, September 25). *Document: DOMContentLoaded event.*

MDN Web Docs. [https://developer.mozilla.org/en-US/docs/Web/API/Document/DOMContentLoaded\\_event](https://developer.mozilla.org/en-US/docs/Web/API/Document/DOMContentLoaded_event)