

Mappedokument

Forutsetninger

Oppgave 1: Dette gikk ut på at vi skulle jukse så mye vi ville, derfor tolket jeg dette som at vi kan kopiere all kode fra AI, og få AI til å gjøre oppgaven «for deg». Prompts og resultat kommer senere i dokumentet. Jeg syntes det var litt vanskelig å forstå hva som faktisk skulle inkluderes i videoen. Jeg tolket dette som at vi skal vise resultatene våres og hvordan vi fikk resultatene. Jeg kunne også forklart all kode som jeg fikk av AI i videoen, men 5 minutter hadde ikke vært nok å forklart alt, så jeg tolket som at det ikke var det som var ment å ha med i videoen.

Oppgave 2: Hvor vi skal refaktorere javascript-kode at man kan bruke prinsipper fra webtricks og undervisningen. Jeg tolket det som at ting fra undervisning er også fra arbeidskrav. Mer om dette på hjelpemidler seksjonen.

Oppgave 3: Det jeg syntes ble litt utydelig i denne oppgaven var hvor viktig CSS var. Vi har ikke brukt mye tid på CSS i dette faget, så jeg syntes i utgangspunktet det var rart at vi skulle lage våres egen CSS her. Her tok jeg også i bruk av AI, men redigerte noe i CSS-filen slik at det ble litt mer som jeg ønsket. Det ble også fortalt når vi fikk oppgaven at selve CSSen IKKE skulle bli vurdert, men i oppgave teksten så står det at det vil bli vurdert. Dette syntes jeg ble litt forvirrende og motsigende. Derfor har jeg gjort litt endringer selv også i CSSen. Jeg har ikke kommentert i selve CSS koden hva jeg har endret på fordi funksjonaliteten var prioriteringen.

Hjelpemidler

Oppgave 1: Som nevnt tidligere, så brukte jeg AI for å løse dette. Jeg nevner det en gang til, men jeg tolket dette som at her var «alt» lov, inkludert å kopiere alt AI har laget til deg. Her brukte jeg Microsoft CoPilot. De to første bildene er prompts som jeg sendte til CoPilot.

Første bilde er for register.js og andre bilde er for actions.js. De andre bildene er resultatene jeg fikk av CoPilot. Bilder av prompt og resultat:

Hei! Jeg skal lage to js filer. Den ene skal være register.js, som skal bearbejdes senere. Register.js skal inneholde en datastruktur som består av ansatte på en høyskole. En ansatt skal bestå av etternavn, fornavn, hvilket kontor de har, hvilken stilling (professor, dekan, rektor, vaktmester og lektor), epost adresse og kursansvar. Stilling skal være en egen liste, og du skal bruke indexen for å fortale hvilken stilling en ansatt har i oversikten over de ansatte. Kursansvar er en liste over hva slags kurs noen personer har. Det er kun undervisere (professor og lektor) som har kursansvar. Alle undervisere skal ha kursansvar og minst tre undervisere skal ha ansvar for minst to kurs. Personene kan være fiktive. Det skal også være en rektor, to dekaner, tre vaktmestere, fem professorer og ti lektorer. Si fra hvis noe er uklart

Flott, tusen takk :) Det er en annen fil som heter actions.js. Denne filen skal inneholde ulike funksjoner. Den første funksjonen skal ta imot indexen til en ansatt og returnere html strukturen om informasjon om den ansatte. Så det vil si at ett "ansattkort" er det htmlen som skal returneres. Den neste funksjonen skal liste ut alle ansatte og bruke den første funksjonen for å strukturere utskriften av htmlen. At når man skriver ut en ansatt så brukes den første funksjonen for å generere html strukturen med riktig informasjon og riktig ansatt. Den tredje funksjonen skal ta imot et parameter som angir stilling og dermed filtrerer alle ansatte på den innsendte parameteren. Som for eksempel hvis man trykker på lektor, så skal det kun være ansatte med stilling som lektor som dukker opp. Man trenger også en knapp for å "vise alle ansatte". Når man trykker på denne, vil alle ansatte dukke opp. Den fjerde funksjonen skal returnere alle kurs som tilbys på skolen. Det vil si å finne alle kurs i kursansvar oversikten og lage en liste over alle disse kursene som kan skrives ut. Den femte funksjonen skal filtrere ut alle undervisere (professorer og lektorer). Det skal også være en funksjon som filtrerer ut alle i administrasjonen (rektor, dekan og vaktmestere). Den syvende funksjonen skal ta imot parametere for all informasjon om en underviser og legge denne til i registeret. Denne funksjonen kommer bare til å fungere senere, og ikke nå. Den siste funksjonen skal ta imot en parameter som identifiserer en ansatt og sletter denne fra registeret. Si fra om noe er uklart

```
JavaScript
// register.js
// Liste over stillinger
const stillinger = [
  "Rektor", // index 0
  "Dekan", // index 1
  "Vaktmester", // index 2
  "Professor", // index 3
  "Lektor" // index 4
];

// Liste over ansatte
const ansatte = [
  // Rektor
  {
    etternavn: "Hansen",
    fornavn: "Kari",
    kontor: "001",
    stilling: 0,
    epost: "kari.hansen@hoysskole.no",
    kursansvar: []
  },
  // Dekaner
  {
    etternavn: "Olsen",
    fornavn: "Per",
    kontor: "B201",
    stilling: 1,
    epost: "per.olsen@hoysskole.no",
    kursansvar: []
  },
  {
    etternavn: "Johansen",
    fornavn: "Anne",
    kontor: "B202",
    stilling: 1,
    epost: "anne.johansen@hoysskole.no",
    kursansvar: []
  },
  // Vaktmestere
  {
    etternavn: "Nilsen",
    fornavn: "Tom",
    kontor: "Vaktmesterbod",
    stilling: 2,
    epost: "tom.nilsen@hoysskole.no",
    kursansvar: []
  },
  {
    etternavn: "Berg",
    fornavn: "Lise",
    kontor: "Vaktmesterbod",
    stilling: 2,
    epost: "lise.berg@hoysskole.no",
    kursansvar: []
  },
  {
    etternavn: "Andreasen",
    fornavn: "Morten",
    kontor: "Vaktmesterbod",
    stilling: 2,
    epost: "morten.andreasen@hoysskole.no",
    kursansvar: []
  }
];
```

```
JavaScript
// Professorer
{
  etternavn: "Sundsen",
  fornavn: "Esp",
  kontor: "001",
  stilling: 3,
  epost: "esp.sundsen@hoysskole.no",
  kursansvar: ["Matematikk", "Statistikk"] // minst to kurs
},
{
  etternavn: "Nagelsen",
  fornavn: "Jonas",
  kontor: "002",
  stilling: 3,
  epost: "jonas.nagelsen@hoysskole.no",
  kursansvar: ["Fysikk"]
},
{
  etternavn: "Pettersen",
  fornavn: "Bjørn",
  kontor: "003",
  stilling: 3,
  epost: "bjorn.pettersen@hoysskole.no",
  kursansvar: ["Kjemi", "Biologi"] // minst to kurs
},
{
  etternavn: "Solberg",
  fornavn: "Henrik",
  kontor: "C304",
  stilling: 3,
  epost: "henrik.solberg@hoysskole.no",
  kursansvar: ["Informatikk"]
},
{
  etternavn: "Dahl",
  fornavn: "Camilla",
  kontor: "C305",
  stilling: 3,
  epost: "camilla.dahl@hoysskole.no",
  kursansvar: ["Økonomi"]
},
// Lektorer
{
  etternavn: "Rasmussen",
  fornavn: "Sara",
  kontor: "001",
  stilling: 4,
  epost: "sara.rasmussen@hoysskole.no",
  kursansvar: ["Engelsk", "Fransk"] // minst to kurs
},
{
  etternavn: "Lars",
  fornavn: "Ole",
  kontor: "002",
  stilling: 4,
  epost: "ole.lars@hoysskole.no",
  kursansvar: ["Historie"]
},
{
  etternavn: "Halvorsen",
  fornavn: "Malin",
  kontor: "003",
  stilling: 4,
  epost: "malin.halvorsen@hoysskole.no",
  kursansvar: ["Litteratur"]
},
{
  etternavn: "Fjellhaugen",
  fornavn: "Ola",
  kontor: "004",
  stilling: 4,
  epost: "ola.fjellhaugen@hoysskole.no",
  kursansvar: ["Geografi"]
},
{
  etternavn: "Lund",
  fornavn: "Emma",
  kontor: "005",
  stilling: 4,
  epost: "emma.lund@hoysskole.no",
  kursansvar: ["Pedagogikk"]
},
{
  etternavn: "Kjennsen",
  fornavn: "Sam",
  kontor: "006",
  stilling: 4,
  epost: "sam.kjennsen@hoysskole.no",
  kursansvar: ["IT-Systemer"]
}
```

```
JavaScript
// 1. Funksjon som lager HTML for én ansatt
// Tar imot index (plassering i ansatte-listen)
function lagAnsattKort(index) {
  const person = ansatte[index]; // Henter personen fra listen
  if (!person) return "<p>Ansatt ikke funnet</p>"; // Hvis index er feil

  const stilling = stillinger[person.stilling]; // Henter stilling fra sti

  // Lager kursliste (hvis personen har kursansvar)
  let kursTekst = "";
  if (person.kursansvar.length > 0) {
    kursTekst = "<ul>";
    for (let kurs of person.kursansvar) {
      kursTekst += "<li>${kurs}</li>";
    }
    kursTekst += "</ul>";
  } else {
    kursTekst = "<p>Ingen kursansvar</p>";
  }

  // Returnerer HTML-struktur for ansattkortet
  return `
    <h3>${person.fornavn} ${person.etternavn}</h3>
    <p>Stilling: ${stilling}</p>
    <p>Kontor: ${person.kontor}</p>
    <p>E-post: ${person.epost}</p>
    <p>Kursansvar: </p>
    ${kursTekst}
  `;
}
```

```
JavaScript
// 2. Funksjon som viser alle ansatte
// Bruker lagAnsattKort for hver ansatt
function visAlleAnsatte() {
  let resultat = "";
  for (let i = 0; i < ansatte.length; i++) {
    resultat += lagAnsattKort(i);
  }
  return resultat;
}

// 3. Funksjon som filtrerer ansatte basert på stilling
// Tar imot stillingsnavn (f.eks. "Lektor")
function filterAnsattePaaStilling(stillingNavn) {
  const stillingIndex = stillinger.indexOf(stillingNavn); // Finnes
  if (stillingIndex === -1) return "<p>Stilling ikke funnet</p>";

  let resultat = "";
  for (let i = 0; i < ansatte.length; i++) {
    if (ansatte[i].stilling === stillingIndex) {
      resultat += lagAnsattKort(i);
    }
  }
  return resultat;
}

// 4. Funksjon som henter alle kurs som tilbys
function hentAlleKurs() {
  let kursListe = [];
  for (let person of ansatte) {
    for (let kurs of person.kursansvar) {
      if (!kursListe.includes(kurs)) { // Sjekker at kurset
        kursListe.push(kurs);
      }
    }
  }

  // Lager HTML-liste av kursene
  let resultat = "<ul>";
  for (let kurs of kursListe) {
    resultat += "<li>${kurs}</li>";
  }
  resultat += "</ul>";
  return resultat;
}

// 5. Funksjon som henter alle undervisere (professorer og lektorer)
function hentUndervisere() {
  let resultat = "";
  for (let i = 0; i < ansatte.length; i++) {
    if (ansatte[i].stilling === 3 || ansatte[i].stilling === 4) {
      resultat += lagAnsattKort(i);
    }
  }
  return resultat;
}
```

```

/**
 * 6. Funksjon som henter administrasjonen (rektor, dekan, vaktmester)
 */
function hentAdministrasjon() {
  let resultat = "";
  for (let i = 0; i < ansatte.length; i++) {
    if (ansatte[i].stilling === 0 || ansatte[i].stilling === 1 || ansatte[i].stilling === 2) {
      resultat += lagAnsattKort(i);
    }
  }
  return resultat;
}

/**
 * 7. Funksjon som legger til en underviser (skal brukes senere)
 */
function leggTilUnderviser(fornavn, etternavn, kontor, stilling, epost, kursansvar) {
  ansatte.push({
    fornavn: fornavn,
    etternavn: etternavn,
    kontor: kontor,
    stilling: stilling,
    epost: epost,
    kursansvar: kursansvar
  });
}

/**
 * 8. Funksjon som sletter en ansatt basert på index
 */
function slettAnsatt(index) {
  if (index >= 0 && index < ansatte.length) {
    ansatte.splice(index, 1); // Fjerner ansatt fra listen
    return true;
  }
  return false;
}

```

Oppgave 2: I tolking av python koden brukte jeg W3 Schools fordi Python er ikke noe jeg er godt kjent med. Selv om koden vi fikk var lett å tolke, var det noe jeg var usikker på som append, split og len. Se litteraturliste for kilde. I refaktorerer av javascript-kode så brukte jeg arbeidskrav 2 som hjelp for å finne ut av gjennomsnittet siden problemet var helt likt, bare at på denne oppgaven var det navn inkludert. Se litteratur liste for referanse til dette.

Oppgave 3: I denne oppgaven har jeg brukt en liten blanding av hjelpemidler. Jeg har brukt internett og CoPilot. Jeg brukte CoPilot og internett til å finne ut av hvordan man kan få «slett» knapp på den ene html-filen (admin) og ikke den andre (register).

Howdan kan jeg få slett knappen til å dukke opp på admin.html, og ikke register.html?

```

// sjekk hvilken side vi er på
const erAdmin = location.pathname.includes("admin.html");

```

Det denne koden gjør er at den sjekker om den er på admin.html eller ikke. Dette gjør man ved å bruke location.pathname. Det denne gjør er at den finner URL som den blir gitt.

Derfor legger man til includes for at den skal vite hvilken URL (eller i dette tilfelle, filen) som den skal finne. Og deretter filnavnet i parentes.

Jeg trengte også hjelp til hvordan man skulle legge til undervisere. Jeg visste at man må hente verdiene fra skjemaet. Jeg la inn disse ved å bruke document.getElementById, navnet og med value. Men når fylte inn feltene og fikk laget et ansattkort, la kursansvar seg nedover hverandre. Jeg lærte senere at jeg måtte bruke value.split(';'). Dette gjør (sopm i likhet med python) lukker tomrom og gjør at bokstavene kommer ved siden av hverandre. Jeg ville også at skjemaet skulle tømme seg etter jeg hadde lagt til et ansattkort. Jeg fant ut at man kan bruke .reset() for at dette skal fungere.

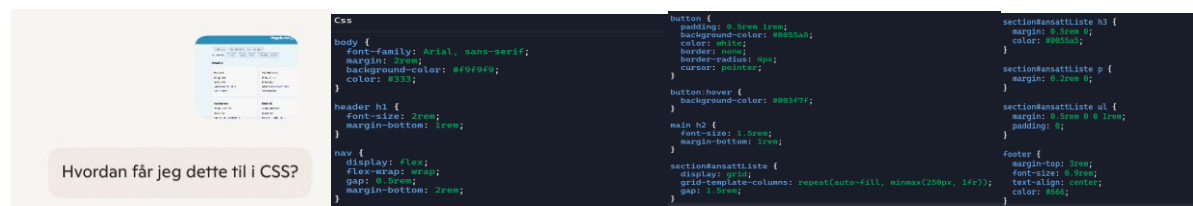
Det siste jeg trengte hjelp med var å få til å slette ansattkortene. Her brukte jeg igjen CoPilot for å spørre hva jeg måtte gjøre. Funksjonen var allerede laget, så det var bare en linje med kode som jeg trengte. Det jeg trengte er det som står etter innerHTML.

Hvordan kan jeg få en knapp til å slette en artikkel/objekt?

```
document.getElementById("ansattListe").innerHTML =  
ansatte.map((_, i) => lagAnsattKort(i)).join("");
```

Først og fremst bruker denne linjen `.map`. Det den gjør er at den kan «transformere» et objekt i en array. I dette tilfelle er det å fjerne/slette dette objektet. I parantesen er det en understrek. Dette er for å lage et element som ikke betyr noe i indeksen, som «`i`» står for. Deretter returnerer den funksjonen `lagAnsattKort`, slik at ikke alle ansattkortene forsvinner. Deretter `.join` som setter objektene sammen.

Som nevnt tidligere, så brukte jeg AI når det kom til CSS. Jeg måtte endre på noen små detaljer og noen av de ulike navnene jeg fikk av AI fordi ellers hadde ikke disse funket fordi det ikke hadde vært riktig navn på id/class.



Referanseliste

HTMLFormElement: Reset() method - Web APIs | MDN. (2025, april 10). MDN Web Docs.
<https://developer.mozilla.org/en-US/docs/Web/API/HTMLFormElement/reset>

JavaScript String split() Method. (u.å.). Hentet 11. desember 2025, fra
https://www.w3schools.com/jsref/jsref_split.asp

Jesper Østli Simensen, "Arbeidskrav 2", Oktober, 2025

Location: Pathname property - Web APIs | MDN. (2023, november 26). MDN Web Docs.
<https://developer.mozilla.org/en-US/docs/Web/API/Location/pathname>

Microsoft. (2025). Copilot [AI-assistent]. <https://copilot.microsoft.com/>

Python Dictionary keys() Method. (u.å.). Hentet 11. desember 2025, fra
https://www.w3schools.com/python/ref_dictionary_keys.asp

Python len() Function. (u.å.). Hentet 11. desember 2025, fra
https://www.w3schools.com/python/ref_func_len.asp

Python List append() Method. (u.å.). Hentet 11. desember 2025, fra
https://www.w3schools.com/python/ref_list_append.asp

Python String split() Method. (u.å.). Hentet 11. desember 2025, fra
https://www.w3schools.com/python/ref_string_split.asp

Link til repositoriet: https://github.com/StormyDoom/Eksamen2025_Programmering