

Prosjektdokumentasjon

1. Vanskelighetsgrad

Valgt nivå: C-B

Fullført funksjonalitet inkluderer:

Responsivt design som fungerer på desktop, tablet og mobil.

Dynamisk bildevelger på produksiden.

Skjemaer med grunnleggende validering.

Tilpassede stiler for produksiden, handlekurv og leveringsalternativer.

2. Forutsetninger

Alle funksjoner og design er basert på oppgaveteksten og inspirasjon fra eksterne kilder. Jeg har fullført alle oppgaver på nivå E og D i filen D i henhold til kravene. I filen C har jeg gjort nødvendige endringer i stil og beriket innholdet. Samtidig som jeg oppfylte kravene for nivå C, har jeg også forsøkt å gjennomføre oppgaver på nivå B. Kan vises fil B.

B-nivå funksjonaliteter og forklaringer

1. Dynamisk oppdatering av handlekurvens totalsum

Handlekurvens totalsum oppdateres automatisk når brukeren endrer antallet på et produkt. Dette gjøres ved hjelp av JavaScript, som beregner summen basert på produktpris og antall.

Funksjonaliteten sikrer at brukeren får en umiddelbar oversikt over kostnadene i handlekurven uten å måtte oppdatere siden.

2. Dynamisk oppdatering av totalsum basert på leveringsmetode

Når brukeren velger en leveringsmetode, oppdateres totalsummen automatisk for å inkludere leveringskostnaden.

Dette gjøres ved å koble JavaScript til radioknappene for leveringsalternativene, som legger til riktig beløp basert på brukerens valg.

En tekstlinje viser også valgte leveringsmetode og kostnad for ekstra brukervennlighet.

3. Optimalisering av hovedbildet på produktsiden

Produktsiden inkluderer nå en interaktiv funksjon der hovedbildet oppdateres når brukeren klikker på et miniatyrbilde.

Denne funksjonen forbedrer brukeropplevelsen ved å la brukeren få en bedre visuell oversikt over produktet.

Funksjonen implementeres ved hjelp av JavaScript som endrer hovedbildets kilde (src) til den valgte miniatyrens.

4. Forbedret responsiv layout

Layouten til nettstedet er optimalisert for ulike skjermstørrelser, inkludert desktop, nettbrett og mobil.

På mobil vises produktbilder som en horisontal rullbar liste for enklere navigering, mens nettbrett og desktop viser flere kolonner for bedre oversikt.

Dette er implementert ved hjelp av CSS Grid og media queries for å sikre en sømløs opplevelse på tvers av enheter

3. utfordringer og løsninger

- Utfordring: Implementering av dynamisk oppdatering av handlekurven.
 - Løsning: Bruk av JavaScript for å oppdatere totalpris i handlekurven basert på brukerens input.
- Utfordring: Tilpassing av layout for ulike skjermstørrelser.
 - Løsning: Brukte media queries for å sikre at innholdet ser bra ut på både små og store skjermer.

4. Kilder

- Dokumentasjon:
 - MDN Web Docs: <https://developer.mozilla.org>
 - W3Schools: <https://www.w3schools.com>
- Ikoner og bilder:
 - Bootstrap Icons: <https://icons.getbootstrap.com>
 - Lorem Picsum: <https://picsum.photos>

Forklaring på bruk av plassholderbilder

For dette prosjektet er plassholderbilder fra Lorem Picsum (<https://picsum.photos>) brukt for å demonstrere design, layout og funksjonalitet. Disse bildene er midlertidige og kan enkelt erstattes med faktiske bilder ved behov.

- **AI bruk:**

- ChatGPT ble brukt for å forbedre koden, spesifikt for:
 - Dynamisk oppdatering av handlekurven.
 - Optimalisering av responsivt design.
 - Forbedring av kodestruktur.

Java:

- **Filterknapp dynamiske effekter**

```
document.querySelectorAll('.filter-buttons button').forEach((button) => {  
    button.addEventListener('click', () => {  
        alert(`Filter applied: ${button.textContent}`);  
    });  
});
```

- **Dynamisk navigasjon til produktside**

```
document.querySelectorAll('.view-details').forEach((link) => {  
    link.addEventListener('click', (e) => {  
        e.preventDefault(); // Hindrer standard oppførsel  
        alert('Redirecting to product details page...');  
        window.location.href = link.href; // Navigerer til lenken  
    });  
});
```

- **Velg produktstørrelse**

```
document.querySelector('#size').addEventListener('change', (e) => {  
    alert(`Valgt størrelse: ${e.target.value}`);  
});
```

- **Legg produkt i handlekurven**

```
document.querySelector('.add-to-cart').addEventListener('click', () => {  
    alert('Produktet er lagt til i handlekurven!');  
});
```

- **Oppdater hovedbildet med miniatyrbilde**

```
document.querySelectorAll('.thumbnail-grid img').forEach((thumbnail) => {  
    thumbnail.addEventListener('click', (e) => {  
        const mainImage = document.querySelector('.main-image img');  
  
        // Oppdater hovedbildet  
        mainImage.src = e.target.src;  
  
        // Fjern valget fra andre miniatyrbilder og marker valgt bilde  
        document.querySelectorAll('.thumbnail-grid img').forEach((img) =>  
img.classList.remove('selected'));  
        e.target.classList.add('selected');  
    });  
});
```

```

// Oppdater dynamisk antall varer i handlekurven
document.querySelectorAll('.antall-input').forEach((input) => {
  input.addEventListener('input', () => {
    const quantity = input.value;
    console.log(`Updated quantity to: ${quantity}`);
  });
});

// Bytt dynamisk leveringsmetoder og vis priser
document.querySelectorAll('input[name="delivery"]').forEach((radio) => {
  radio.addEventListener('change', (e) => {
    const price = e.target.closest('label').querySelector('.price').textContent;
    alert(`Selected delivery option with price: ${price}`);
  });
});

document.querySelectorAll('.antall-input').forEach((input) => {
  input.addEventListener('input', () => {
    const totalPriceElement = document.querySelector('.total-price strong');
    let total = 0;
    document.querySelectorAll('.shopping-item').forEach((item) => {
      const price = parseInt(item.querySelector('.item-info p').textContent.replace('Kr ', ''), 10);
      const quantity = parseInt(item.querySelector('.antall-input').value, 10);
      total += price * quantity;
    });
    totalPriceElement.textContent = `${total} kr`;
  });
});

document.querySelectorAll('input[name="delivery"]').forEach((radio) => {
  radio.addEventListener('change', (e) => {
    const selectedOption = e.target.dataset.name;
    const price = parseInt(e.target.value, 10);
    const basePrice = 1199;
    const totalPriceElement = document.querySelector('.total-price strong');
    const deliveryInfo = document.querySelector('.delivery-info');

    // Oppdater fraktmetode og totalpris
    totalPriceElement.textContent = `${basePrice + price} kr`;
    deliveryInfo.textContent = `Valgt levering: ${selectedOption}, Pris: ${price} kr`;
  });
});

```

- Dynamisk oppdatering av antall varer i handlekurven
- Dynamisk bytte av leveringsmetoder og visning av priser
- Dynamisk oppdatering av totalpris i handlekurven
- Dynamisk oppdatering av totalpris med leveringsmetode

Github: <https://github.com/mxxdccc/Webutvikling.git>