

Utvikling av interaktive nettsteder Eksamen

I denne eksamen har vi valgt å sikte mot C-kravet. Vi har hatt et tett og godt samarbeid gjennom hele arbeidsprosessen, og har jevnlig møttes på skolen for å jobbe sammen. Arbeidsfordelingen har vært jevn gjennom hele prosessen, og vi har samarbeidet tett for å sikre at vi gjorde jobben på best mulig måte.

GitHub-alias på gruppemedlemmene:

CelinaPs – Celina Wedul Dahl-Jacobsen

Saraamel – Sara Amel Bejaoui

Aynazar1202222 – Aini Ayinazier

Martehalv – Marte Halvorsen

Link til GitHub repository:

https://github.com/Martehalv/UIN_Eksamen2025.git

Utfordringer:

En ting vi slet veldig med, og som ikke sto klart i oppgaven var hvordan vi skulle hente API-ene til festivalene. Dette var det vi brukte mest tid på, og skapte mye forvirring til resten av oppgaven. Når dette kom på plass, begynte ting å endelig å fungere. Siden vi har slitet med API, og har prøvd mange forskjellige løsninger, etter hvert har det ført til mye rot i koden, noe som førte til andre små feil. På grunn av det har vi prøvd i flere repositories slik at koden vår ble ryddig og funkete som vi ønsket.

Vi har hatt problemer med at data ikke alltid blir hentet inn på nettsiden, men kommer tilbake ved å «refresh» siden.

Usikkerhet i oppgaveteksten:

Vi var litt usikre på overgangen fra E til D krav angående EventPage og EventCard. Slik vi forsto det trengte vi ikke en ny fil som het EventCard, men at EventCard skulle bli en funksjon innenfor EventPage. Siden vi ikke så noen andre bruksområder for EventPage.

Kilder:

<https://lms.webtricks.blog/kurs/uin> - Webtricks er brukt i løpet av hele oppgaven.

https://developer.mozilla.org/en-US/docs/Web/API/HTMLFormElement/submit_event

<https://developer.mozilla.org/en-US/docs/Web/API/Event/preventDefault>

https://www.w3schools.com/jsref/event_onsubmit.asp

https://www.w3schools.com/react/react_state.asp

<https://stackoverflow.com/questions/55757761/handle-an-input-with-react-hooks>

<https://stackoverflow.com/questions/73402093/react-form-state-and-trimming-value>

[The Ticketmaster Developer Portal](#) – Hente event ID-er fra Ticketmaster til Festivalene

[ChatGPT](#) (Prompts ligger under)

[Handling Events – React](#)

[Array.prototype.some\(\) - JavaScript | MDN](#)

[Discovery API - The Ticketmaster Developer Portal](#)

ChatGPT:

Prompt: (Link til samtale: <https://chatgpt.com/share/682643e8-d5a4-8013-8377-dc8e182874a8>)

Jeg skal lage en nettside ved bruk av React. Jeg skal lage en hjemmeside, på hjemmesiden skal det ligge 4 festivaler. Disse 4 festivalene skal hentes med API fra Ticketmaster. Jeg har funnet "attraction" ID-en til festivalene, hver festival skal kunne klikkes på, og når den blir klikket på kommer vi til EventPage.jsx Jeg har startet litt på koden, men sliter litt. Jeg ønsker å bruke Fetch for å hente. Har lagt til bildet av API ID-ene, men vi skal kun bruke de jeg har lagt inn i koden under. Start Kode på Home.jsx

```
export default function Home() { const API_KEY = "5OYpmz3EGOfKvtU8vILa3cC6ASbAeGgE";  
const FESTIVAL_IDS = [ "K8vZ917_YJf", //NEON "K8vZ917oWOV", //Tons of Rock  
"K8vZ917K7fV", //Findings "K8vZ917bJC7", //Skeikampenfestivalen ]; const fetchFestival =  
async (id) => { const url =  
https://app.ticketmaster.com/discovery/v2/attractions/${id}.json?apikey=${API_KEY}; }; }
```

Bildet jeg la til:

The screenshot displays a web application interface for event discovery. It is organized into three main panels:

- Page Panel:** Shows a list of events. The first event is "NEON | Lørdagsspass" at "Dahls Arena" on "2025-06-07". Below it are three more events: "NEON | Lørdagsspass Premium", "NEON | Lørdagsspass Student", and "NEON | Lørdagsspass Premium Student", all at "Dahls Arena" on "2025-06-07". The last event is "Neon Prophets" on "2025-05-17" at "Warren's Backy...".
- Object Panel (Left):** Provides details for the selected event "NEON | Lørdagsspass". It includes the name, type (event), ID (Z698xZb_Z17q339), test status (false), URL (https://www.ticketmaster.no/event/neon-%7C-lordagsspass-billetter/752497), and locale (no-no).
- Object Panel (Right):** Provides details for the selected event "NEON Festival". It includes the name, type (attraction), ID (K8vZ917_YJf), test status (false), URL (https://www.ticketmaster.com/neon-festival-tickets/artist/2833327), and locale (en-us).

Both Object panels also display a list of images with their respective ratios and dimensions:

- Image 1: Ratio: 16_9 (639 x 1136)
- Image 2: Ratio: 3_2 (203 x 305)
- Image 3: Ratio: 16_9 (115 x 205)
- Image 4: Ratio: 3_2 (683 x 1024)

Hvorfor vi brukte Chat:

Vi brukte mye tid på å finne ut av henting av festivalene i Home.jsx. Vi sleit med å bruke API for å hente informasjon til vi skjønnte vi måtte hente ID-er fra en annen nettside. Alikavel klarte vi ikke å skjønne helt hvordan vi skulle få det til.

Svar fra ChatGPT:

Flott start! Du har allerede festival-ID-ene og en funksjon på plass for å hente data. For å gjøre Home.jsx funksjonell, må vi:

1. Bruke useEffect for å hente festivaldata ved lasting av komponenten.
2. Lagre festivaldata i en useState.
3. Mappe over dataene for å vise dem.
4. Bruke React Router for å navigere til EventPage.jsx.

Her er et eksempel på hvordan du kan fullføre Home.jsx:

jsx

KopierRediger

```
import { useEffect, useState } from "react";
```

```
import { useNavigate } from "react-router-dom";
```

```
export default function Home() {
```

```
  const API_KEY = "5OYpmz3EGOfKvtU8vILA3cC6ASbAeGgE";
```

```
  const FESTIVAL_IDS = [
```

```
    "K8vZ917_YJf", // NEON
```

```
    "K8vZ917oWOV", // Tons of Rock
```

```
    "K8vZ917K7fV", // Findings
```

```
    "K8vZ917bJC7", // Skeikampenfestivalen
```

```
];
```

```
const [festivals, setFestivals] = useState([]);
```

```
const navigate = useNavigate();
```

```
const fetchFestival = async (id) => {
```

```
  const url =
```

```
  `https://app.ticketmaster.com/discovery/v2/attractions/${id}.json?apikey=${API_KEY}`;
```

```
  const response = await fetch(url);
```

```
  const data = await response.json();
```

```
  return data;
```

```
};
```

```
useEffect(() => {
```

```
  const fetchAllFestivals = async () => {
```

```
    const results = await Promise.all(FESTIVAL_IDS.map(id => fetchFestival(id)));
```

```
    setFestivals(results);
```

```
  };
```

```
  fetchAllFestivals();
```

```
}, []);
```

```
return (
```

```
  <div>
```

```
    <h1>Festivaler 2025</h1>
```

```
<div style={{ display: "flex", gap: "20px", flexWrap: "wrap" }}>

  {festivals.map((festival) => (

    <div

      key={festival.id}

      style={{ border: "1px solid #ccc", padding: "10px", cursor: "pointer" }}

      onClick={() => navigate(`/event/${festival.id}`)}

    >

      <img

        src={festival.images?.[0]?.url}

        alt={festival.name}

        style={{ width: "200px", height: "auto" }}

      />

      <h3>{festival.name}</h3>

    </div>

  ))}

</div>

</div>

);

}
```

For å få dette til å fungere:

- Du må ha react-router-dom installert og satt opp med en rute som fanger /event/:id og peker til EventPage.jsx.
- I EventPage.jsx, bruk useParams() for å hente ID-en og vise detaljert informasjon om festivalen.

Vil du at jeg også hjælper med EventPage.jsx eller App.jsx for routing?

Follow up Prompt:

```
Kan du fjerne alle style? import { useEffect, useState } from "react"; import { useNavigate }
from "react-router-dom"; export default function Home() { const API_KEY =
"5OYpmz3EGOfKvtU8vILA3cC6ASbAeGgE"; const FESTIVAL_IDS = [ "K8vZ917_YJf", // NEON
"K8vZ917oWOV", // Tons of Rock "K8vZ917K7fV", // Findings "K8vZ917bJC7", //
Skeikampenfestivalen ]; const [festivals, setFestivals] = useState([]); const navigate =
useNavigate(); const fetchFestival = async (id) => { const url =
https://app.ticketmaster.com/discovery/v2/attractions/${id}.json?apikey=${API_KEY}; const
response = await fetch(url); const data = await response.json(); return data; }; useEffect(() =>
{ const fetchAllFestivals = async () => { const results = await Promise.all(FESTIVAL_IDS.map(id
=> fetchFestival(id))); setFestivals(results); }; fetchAllFestivals(); }, []); return ( <div>
<h1>Festivaler 2025</h1> {festivals.map((festival) => ( <div key={festival.id} style={{ border:
"1px solid #ccc", padding: "10px", cursor: "pointer" }} onClick={() =>
navigate(/event/${festival.id})} > <img src={festival.images?.[0]?.url} alt={festival.name}
style={{ width: "200px", height: "auto" }} /> <h3>{festival.name}</h3> </div> ))) </div>
); }
```

Svar fra ChatGPT:

```
import { useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";

export default function Home() {
  const API_KEY = "5OYpmz3EGOfKvtU8vILA3cC6ASbAeGgE";
  const FESTIVAL_IDS = [
    "K8vZ917_YJf", // NEON
    "K8vZ917oWOV", // Tons of Rock
    "K8vZ917K7fV", // Findings
    "K8vZ917bJC7", // Skeikampenfestivalen
  ];

  const [festivals, setFestivals] = useState([]);
  const navigate = useNavigate();
```

```
const fetchFestival = async (id) => {  
  const url =  
`https://app.ticketmaster.com/discovery/v2/attractions/${id}.json?apikey=${API_KEY}`;  
  const response = await fetch(url);  
  const data = await response.json();  
  return data;  
};
```

```
useEffect(() => {  
  const fetchAllFestivals = async () => {  
    const results = await Promise.all(FESTIVAL_IDS.map(id => fetchFestival(id)));  
    setFestivals(results);  
  };  
  fetchAllFestivals();  
}, []);
```

```
return (  
  <div className="home-container">  
    <h1>Festivaler 2025</h1>  
    <div className="festival-list">  
      {festivals.map((festival) => (  
        <div  
          key={festival.id}  
          className="festival-card"  
          onClick={() => navigate(`/event/${festival.id}`)}  
        >  
          <img  
            src={festival.images?.[0]?.url}  
            alt={festival.name}  
            className="festival-image"
```



```
        />
        <h3>{festival.name}</h3>
    </div>
    )})
</div>
</div>
);
}
```