

## Dokumentajson og kilder

Vanskligetsgrad:

Vi valde att gå för svårighetsgrad A

Github brukernavn:

MirnaBuntic = Mirna Buntic

JulieeRiise = Julie Riise

Danielladominey = Daniella Dominey

Vurdering av samarbete på github:

Vi vill gärna ha en vurdering av arbetsfördelningen.

[https://github.com/MirnaBuntic/UIN\\_eksamen.git](https://github.com/MirnaBuntic/UIN_eksamen.git)

Redogörelser av potentiella utfordringer:

- Vi upplevde att det var vanskelig å få till filtreringen på categorypage. Efter flera dagars försök valde vi tillslut att ge upp med att få det fullt fungerade. Vi har kun fått till filtrering av land överallt utom på attraktioner.
- Vi tyckte också att det var lite svårt när många essensiella grejer stod längst ner i oppgaven. Då var det lätt att glömma av grejer som man skulle gjort från första början. Därför glömde vi helt enkelt att börja med branches. Detta märkte vi dock efter ett tag då det blev problem när vi jobbade i samma filer i samma branch. Vi fixade dock branches efterhvert och då fungerade det mycket bättre eftersom vi kunde hantera konflikterna på ett lättare sätt.

Kilder:

Kilder till scss:

Underlinje ( Eventpage.scss linje nr: 35)

<https://oppgaver.kidsakoder.no/web/tekststil/tekststil>

Hindre scrolling (app.scss – linje nr: 4)

[https://www.w3schools.com/cssref/css3\\_pr\\_overflow-x.php](https://www.w3schools.com/cssref/css3_pr_overflow-x.php)

Skyggeboks brukt for å «ramme inn ulike elementer»

(categorypage.scss linje nr: 77) ( Eventpage.scss linje nr:97) (header.scss linje nr: 145)

[https://www.w3schools.com/css/css3\\_shadows\\_box.asp](https://www.w3schools.com/css/css3_shadows_box.asp)

Hover (citycard.scss linje nr: 23 )

<https://stackoverflow.com/questions/15983639/sass-nesting-for-hover-does-not-work>

Usynlig på knappen ( citycard.scss linje nr: 15)

[https://www.w3schools.com/cssref/pr\\_border-color.php](https://www.w3schools.com/cssref/pr_border-color.php)

<https://stackoverflow.com/questions/2506844/how-do-i-make-a-transparent-border-with-css-so-that-contents-dont-shift-position>

Footer for å få finere overgang ( footer.scss linje nr:2)

<https://developer.mozilla.org/en-US/docs/Web/CSS/border-top>

Tilpassede bilder (header.scss linje nr: 159)

[https://www.w3schools.com/css/css3\\_object-fit.asp](https://www.w3schools.com/css/css3_object-fit.asp)

Kilder till bilderna jag använde i sanity:

- Jonas Gahr Støre: <https://www.leirfjord.kommune.no/statsminister-jonas-gahr-stoere-kommer-til-leirfjord.6710050-176630.html>
- Caroline Nitter: <https://www.tv2.no/underholdning/onlyfans-smell-pa-330000-kroner/17240139/>
- Therese Johaug: <https://www.teamnor.no/profiler/langrenn/therese-johaug/>
- Leo Ajkic: <https://www.seher.no/kjendis/kan-glise-etter-million-salg/82846885>
- Erling Braut Haaland: <https://www.unisportstore.no/fotballdrakter/norge-hjemmedrakt-2024-haaland-9/355515/>
- Jo Nesbø:  
<https://www.facebook.com/photo/?fbid=277456443752026&set=a.277456423752028>

Länkar till hemsidor som används i koden:

- W3 schools – jag använde json.stringify eftersom jag vet att det bara är strings som kan sparas i localStorage. Jag sökte på google efter JSON.stringify då jag ville veta hur man konverterar tillbaka stringen till ett objekt. På w3 schools hittade jag JSON.parse:  
[https://www.w3schools.com/js/js\\_json\\_stringify.asp](https://www.w3schools.com/js/js_json_stringify.asp)
- <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-label>  
Brukt til å finne aria-labels.
- <https://fakeimg.pl/600x400?text=No+image+found/> /  
<https://betterplaceholder.com/>  
Brukt til å implementere en fake placeholder
- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/encodeURIComponent](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/encodeURIComponent)  
Brukt til å lese om encodeURIComponent. Fant ut av dette sammen med chatGPT, prompt og svar nedenfor.

Prompts och svar i chatgpt:

1.Jag frågade chatgpt om detta: Jag vill använda async och await i min kod. Hur kan jag göra så att den catchar error på smidigast sätt? Jag frågade också vad som var bäst av then...catch metoden eller try...catch och då sa den att try...catch var bättre i princip, lättare läst kod osv. Anledningen till att jag frågade om detta var för att jag vet att AC har föreläst om .then, men jag har hört på andra håll att try kan vara ett bättre alternativ. Därför frågade jag chat gpt.

Detta fick jag som svar:

För att använda async och await i din kod och hantera fel på ett smidigt sätt, kan du använda try...catch blocken. När du använder async och await, innebär det att du arbetar med asynkrona operationer som kan kasta fel, så att fånga dessa fel är viktigt.

Här är ett exempel på hur du kan använda async och await med en try...catch för att fånga och hantera fel:

**Exempel på kod:**

javascript

KopierRediger

```

async function fetchData() {
  try {
    // Försök att hämta data från en API
    const response = await fetch('https://api.exempel.com/data');

    if (!response.ok) {
      throw new Error('Något gick fel med att hämta data!');
    }

    const data = await response.json();
    console.log(data);

  } catch (error) {
    // Fångar fel och skriver ut dem på ett användarvänligt sätt
    console.error('Fel vid hämtning av data:', error.message);
    // Här kan du även lägga till en återhämtningslogik eller ett meddelande till användaren
  }
}

fetchData();

```

3.När jag hade fått fram genre så fick jag många av de samma eftersom värden låg under festivalpass och inte att själva festivalen var en genre. Så om flera festivalpass hade samma namn så fick jag alltså värdet flera gånger. Jag frågade därför chatgpt om detta efter att jag frågat den ovanstående frågan: hur tar jag bort dubletter av ett värde? Jag valde att använda set eftersom det verkade lättast. Det den gör är att lagra unika värden, därav försvinner dubletter. Jag valde att använda chatgpt till detta då det helt enkelt är det mest effektiva hjälpmidlet att använda.

#### Chats svar:

För att ta bort dubletter från en array i JavaScript finns det flera sätt att göra det. Här är några vanliga sätt:

##### **1. Använd Set för att ta bort dubbletter (enkelt sätt):**

En Set är en samling där alla värden är unika, så genom att konvertera en array till en Set och sedan tillbaka till en array, kan du enkelt ta bort dubletter.

javascript

KopierRediger

```
let data = [1, 2, 3, 3, 4, 5, 5, 6];
```

```
// Ta bort dubbletter genom att konvertera till en Set och tillbaka till en array
let uniqueData = [...new Set(data)];
```

```
console.log(uniqueData);
```

##### **Output:**

javascript

KopierRediger

[1, 2, 3, 4, 5, 6]

**2. Använd filter-metoden för att ta bort dubbleller manuellt:**

Du kan också använda filter tillsammans med indexOf för att ta bort dubbleller.

javascript

KopierRediger

```
let data = [1, 2, 3, 3, 4, 5, 5, 6];
```

```
// Ta bort dubbleller med filter och indexOf
```

```
let uniqueData = data.filter((value, index, self) => self.indexOf(value) === index);
```

```
console.log(uniqueData);
```

**Output:**

javascript

KopierRediger

[1, 2, 3, 4, 5, 6]

**3. Använd reduce för att skapa en array utan dubbleller:**

En annan metod är att använda reduce för att bygga upp en ny array och kontrollera om ett värde redan finns.

javascript

KopierRediger

```
let data = [1, 2, 3, 3, 4, 5, 5, 6];
```

```
// Ta bort dubbleller med reduce
```

```
let uniqueData = data.reduce((acc, value) => {
  if (!acc.includes(value)) {
    acc.push(value);
  }
  return acc;
}, []);
```

```
console.log(uniqueData);
```

**Output:**

javascript

KopierRediger

[1, 2, 3, 4, 5, 6]

1. Jag valde att fråga detta till chatgpt eftersom jag inte förstod felet i konsollen och därför inte hade något konkret att skriva in på google. Därörför var det lättast att bara skicka det som kom upp i konsollen till chat. Jag försökte få ut sociala medier vilket jag också fick i konsolloggen. Problemet var att det inte ville synas på hemsidan. Därför skickade jag denna koden som jag gjort till chatgpt + vad jag fick upp i konsollen:

```
const url =
https://app.ticketmaster.com/discovery/v2/events.json?apiKey=${apiKey}&locale=*&attractionId=
${attraction.id}; //Chat gpt, samma metod som i app.jsx nr1 try { const response = await
fetch(url); const data = await response.json(); const events = data._embedded?.events || []; const
genres = events .map(event => event.classifications?.[0]?.genre?.name) .filter(genre => genre &&
genre !== 'Undefined'); //Chatgpt visade hur man filtrerar bort undefined men eftersom det inte
var ett värde så var jag tvungen att sätta det som en string. nr 3 const singleGenres = [...new
Set(genres)]; //chatgpt nr 4 setGenres(singleGenres.length > 0 ? singleGenres : ["Ingen genre
tilgjengelig"]); const socialLinks = events[0]?._embedded?.attractions[0]?._externalLinks || [];
setSocialMedia(socialLinks); const passes = events.map(event => ({ id: event.id, name:
event.name, image: event.images?.[0]?._url, date: event.dates?.start?.localDate, venue:
event._embedded?.venues?.[0]?._name, })); setFestivalPasses(passes); const artistList =
events[0]?._embedded?.attractions || []; setArtists(artistList); } catch (error) {
console.error("Skjedde noe feil ved fetch av artister", error); } }; getFestivalData(); }, [attraction]); if
```

```
(!attraction) { return <p>Ingen festival funnet...</p>; } return ( <> <h2>{attraction.name}</h2>
<section> <article> <h3>Sjanger: </h3> <ul> {genres.map((genre, index) => ( <li
key={index}>{genre}</li> ))} </ul> </article> {socialMedia.length > 0 && ( <article> <h3>Følg oss på
sosiale medier:</h3> <ul> {socialMedia.map((link, index) => ( <li key={index}> <a
href={[link.url]}>{link.name}</a> </li> ))} </ul> </article> )} </section> {festivalPasses.length > 0 && (
<section> <h2>Festivalpass:</h2> {festivalPasses.map(pass => ( <EventCard key={pass.id}
event={pass} /> ))} )} {artists.length > 0 && ( <section> <h2>Artister:</h2> <ul>
{artists.map((artist) => ( <li key={artist.id}> <ArtistCard artist={artist} /> </li> ))} </ul> </section> )}
</> ); }
```

Detta fick jag som svar:

Nu när vi ser strukturen på socialLinks i loggen ({spotify: Array(1), instagram: Array(1), homepage: Array(1)}), vet vi att sociala medielänkar är grupperade under specifika nycklar som spotify, instagram, och homepage. Varje nyckel innehåller en array med objekt som har en url.

Så här ser datan ut för varje plattform:

- spotify: En array med objekt som innehåller en URL.
- instagram: En array med objekt som innehåller en URL.
- homepage: En array med objekt som innehåller en URL.

**Problemet:**

I din nuvarande kod försöker du iterera över hela socialMedia som om det var en lista med objekt, men i stället är det ett objekt med nycklar som innehåller arrayer av länkar.

För att rätta till detta måste vi justera hur vi iterarar över socialMedia och hämtar rätt länk.

**Förslag på lösning:**

1. **Iterera över nycklarna i objektet** (spotify, instagram, homepage).
2. **Hämta URL:en från varje array.**

Här är hur du kan justera din kod för att hantera strukturen korrekt:

**Uppdaterad kod:**

javascript

KopierRediger

```
import { useParams } from "react-router-dom";
import { useState, useEffect } from "react";
import ArtistCard from "./ArtistCard";
import EventCard from "./EventCard";
```

```
export default function EventPage({ attractions }) {
```

```
  const { slug } = useParams();
  const [genres, setGenres] = useState([]);
  const [socialMedia, setSocialMedia] = useState({});
```

```

const [festivalPasses, setFestivalPasses] = useState([]);
const [artists, setArtists] = useState([]);

const attraction = attractions.find(attraction =>
attraction.name.toLowerCase().replace(/\s+/g, '-') === slug
);

useEffect(() => {
  const getFestivalData = async () => {
    if (!attraction) return;

    const apiKey = //apikey;
    const url =
`https://app.ticketmaster.com/discovery/v2/events.json?apikey=${apiKey}&locale=*&attractionId=${attraction.id}`;

    try {
      const response = await fetch(url);
      const data = await response.json();

      const events = data._embedded?.events || [];
      const genres = events
        .map(event => event.classifications?[0]?genre?.name)
        .filter(genre => genre && genre !== 'Undefined');
      const singleGenres = [...new Set(genres)];
      setGenres(singleGenres.length > 0 ? singleGenres : ["Ingen genre tilgjengelig"]);

      // Extrahera externa länkar (social media links)
      const socialLinks = events[0]?._embedded?.attractions[0]?externalLinks || {};
      console.log(socialLinks); // Debugglogg för att se om länkarna finns
      setSocialMedia(socialLinks);

      const passes = events.map(event => ({
        id: event.id,
        name: event.name,
        image: event.images?[0]?url,
      }));
    }
  }
}

```

```
        date: event.dates?.start?.localDate,
        venue: event._embedded?.venues?.[0]?.name,
    });
    setFestivalPasses(passes);

    const artistList = events[0]?._embedded?.attractions || [];
    setArtists(artistList);
} catch (error) {
    console.error("Skjedde noe feil ved fetch av artister", error);
}
};

getFestivalData();
}, [attraction]);

if (!attraction) {
    return <p>Ingen festival funnet...</p>;
}

return (
<>
<h2>{attraction.name}</h2>

<section>
<article>
<h3>Sjanger: </h3>
<ul>
{genres.map((genre, index) => (
    <li key={index}>{genre}</li>
)))
</ul>
</article>
</section>
);

{Object.keys(socialMedia).length > 0 && (
<article>
```

```
<h3>Følg oss på sosiale medier:</h3>
<ul>
  {Object.keys(socialMedia).map((platform, index) => (
    <li key={index}>
      <a href={socialMedia[platform][0].url} target="_blank" rel="noopener noreferrer">
        {platform.charAt(0).toUpperCase() + platform.slice(1)}
      </a>
    </li>
  )))
</ul>
</article>
)}

</section>

{festivalPasses.length > 0 && (
  <section>
    <h2>Festivalpass:</h2>
    {festivalPasses.map(pass => (
      <EventCard key={pass.id} event={pass} />
    )))
  </section>
)}

{artists.length > 0 && (
  <section>
    <h2>Artister:</h2>
    <ul>
      {artists.map((artist) => (
        <li key={artist.id}>
          <ArtistCard artist={artist} />
        </li>
      )))
    </ul>
  </section>
)}
```

```
</>  
);  
}  
};
```

#### Förändringar:

1. **Uppdaterad hantering av sociala medier:** Vi använder Object.keys(socialMedia) för att iterera över de plattformar (som spotify, instagram, homepage) som finns i socialMedia. För varje plattform hämtar vi URL från den första posten i arrayen, exempelvis: socialMedia[platform][0].url.
2. **Visning av namn på plattformar:** Plattformens namn (som "Spotify", "Instagram") visas genom att ta den första bokstaven och göra den versal: platform.charAt(0).toUpperCase() + platform.slice(1).

#### Kontrollera:

- **Loggar:** Kolla i konsolen (högerklicka på sidan och välj "Inspect" > "Console") och se om du får några felmeddelanden. Om ingen data visas för sociala medier, kontrollera om socialMedia innehåller några data (det ska vara ett objekt med plattformar som spotify, instagram, etc.).
- **Länkar:** Nu kommer du att se länkar för varje social plattform, och varje länk kommer att öppnas i en ny flik (target="\_blank").

Prova att köra detta och se om sociala medielänkarna dyker upp på sidan.

2. Jag frågade chatgpt detta: hur kan jag lägga in ett nytt felt i schematypes för sanity som är obligatoriskt att fylla i. detta fick jag som svar:

För att lägga till ett **obligatoriskt fält** i en Sanity-schemafil (schemaTypes), gör du följande:

 **Steg för att lägga till ett fält som är obligatoriskt:**

Anta att du har ett schema för t.ex. en artist. Du vill lägga till ett nytt fält, till exempel "instagram", och kräva att det fylls i.

```
js  
KopierRediger  
// schemas/artist.js  
export default {  
  name: 'artist',  
  title: 'Artist',  
  type: 'document',  
  fields: [  
    {  
      name: 'name',  
      title: 'Navn',  
      type: 'string',  
      validation: Rule => Rule.required().error('Navn er påkrevd')  
    },  
    {  
      name: 'instagram',  
      title: 'Instagram-brukernavn',  
      type: 'string',  
      validation: Rule => Rule.required().error('Instagram-brukernavn må fylles ut')  
    },  
    // Lägg till fler fält här  
  ]  
}
```

 **Förklarat:**

- validation: Rule => Rule.required() gör fältet obligatoriskt.
- .error('...') låter dig visa ett eget felmeddelande i Studio.

3. Jeg spurte chat: det vises kun riktig utskrift av events, venues og attractions i sport slug?

Og limte inn koden min:

```
import React, { useEffect, useState } from "react"; import { useParams } from "react-router-dom"; import CategoryCard from "./CategoryCard"; import categories from "./DataCategory"; import CategoryCardTwo from "./CategoryTwoCard"; import "./styles/header.scss" import "./styles/categorypage.scss"  
//https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter  
//https://react.dev/learn/choosing-the-state-structure#forms //https://legacy.reactjs.org/docs/hooks-state.html export default function CategoryPage () { const { slug } = useParams (); const categoryName = categories.find(c => c.slug === slug)?.name || "Kategori"; const [attractions, setAttractions] =useState ([]); const [events, setEvents] =useState ([]); const [venues, setVenues] =useState ([]); const [wishlist, setWishlist] =useState ([]); const apiKey = "4P5afjX98PHm5yhdSLbee6G9PVKAQGB7"; useEffect(() => { const fetchData = async () => { const url ="https://app.ticketmaster.com/discovery/v2/suggest"; const params = apiKey=${apiKey}&locale=*&keyword=${slug} || ""; try { const attractionsRes = await fetch (${url}?${params}&resource=attractions); const attractionsData = await attractionsRes.json(); setAttractions(attractionsData._embedded?.attractions || []); const venuesRes = await fetch (${url}?${params}&resource=venues); const venueData = await venuesRes.json(); setVenues(venueData._embedded?.venues || []); const urlEvents =  
https://app.ticketmaster.com/discovery/v2/events.json?apikey=\${apiKey}&locale=\*&size=20&classificationName=\${slug} || ""; const eventRes = await fetch (urlEvents); const eventData = await eventRes.json(); setEvents(eventData._embedded?.events || []); } catch (error) { console.error("feil ved henting via suggest:", error); } fetchData(); }, [slug]); const toggleWishlist = (id) => { setWishlist((prev) => prev.includes(id)? prev.filter((x) => x !== id) : [...prev, id] ); }; return (<> <h1>{categoryName}</h1> <section className="Attraction"> <h2>Attraksjoner</h2> <div> {attractions.map((item) => ( <CategoryCardTwo key={item.id} item={item} isSaved={wishlist.includes(item.id)} onSave={() => toggleWishlist(item.id)} /))} </div> </section> <section className="Arrangement"> <h2>Arrangementer</h2> <div> {events.map((item) => ( <CategoryCard key={item.id} item={...item, date: item.dates?.start?.toLocaleDate, time: item.dates?.start?.localTime, venue: item._embedded?.venues?.[0]?.name, city: item._embedded?.venues?.[0]?.city?.name, country: item._embedded?.venues?.[0]?.country?.name } isSaved={wishlist.includes(item.id)} onSave={() => toggleWishlist(item.id)} /))} </div> </section> <section className="Venues"> <h2>Spillesteder</h2> <div> {venues.map((item) => ( <CategoryCardTwo key={item.id} item={item} isSaved={wishlist.includes(item.id)} onSave={() => toggleWishlist(item.id)} /))} </div> </section> </> ); } import "./styles/header.scss" import React from "react"; export default function CategoryCardTwo ({item, isSaved, onSave}) { console.log("CategoryCard mottar dette:", item); console.log(${item.name} er ${isSaved ? "lagret" : "ikke lagret"}); const imageUrl = item.images?.[0].url || ""; const name = item.name || "Uten navn"; return ( <div className="card"> {imageUrl && <img src={imageUrl} className="card-img"/>} <h3>{name}</h3> <button onClick={onSave}> <i className={isSaved ? "fa-solid fa-heart" : "fa-regular fa-heart"}></i> </button> </div> ); } import React from "react"; export default function CategoryCard ({item, isSaved, onSave}) { console.log("CategoryCard mottar dette:", item); console.log(${item.name} er ${isSaved ? "lagret" : "ikke lagret"}); const imageUrl = item.images?.[0].url || ""; const name = item.name || "Uten navn"; const venue = item._embedded?.venues?.[0]?.name || item.adress?.line1 || "sted ikke tilgjengelig"; const date = item.dates?.start?.toLocaleDate || "Ukjent dato"; return ( <div className="card"> {imageUrl && <img src={imageUrl} className="card-img"/>} <h3>{name}</h3> <p>{venue}</p> <p>{date}</p> <button onClick={onSave}> <i className={isSaved ? "fa-solid fa-heart" : "fa-regular fa-heart"}></i> </button> </div> ); } const categories = [ { id: 1, name: "Musikk", slug: "music" }, { id: 2, name: "Sport", slug: "sport" }, { id: 3, name: "Teater/Show", slug: "Arts & Theatre" } ]; export default categories; d
```

Dette fikk jeg som svar:

- Based on the provided code, it seems you have a React component CategoryPage that fetches data from the Ticketmaster API based on the slug parameter. You want to ensure that only data related to the "sport" category slug (categories.slug === 'sport') is displayed correctly for events, venues, and attractions.

- Here are a few observations and adjustments to ensure it displays correctly:
  - **Data Filtering:** Ensure that your `fetchData` function filters data correctly based on the `slug` parameter to fetch data relevant to the "sport" category.
  - **Conditional Rendering:** Make sure that sections for attractions, events, and venues are conditionally rendered based on the `slug` parameter matching "sport".
  - Here's how you can adjust your `useEffect` and conditional rendering to achieve this:
  - javascript
  - KopierRediger
- ```

export default function CategoryPage() {
  const { slug } = useParams();

  const categoryName =
    categories.find((c) => c.slug === slug)?.name || "Kategori";

  const [attractions, setAttractions] = useState([]);
  const [events, setEvents] = useState([]);
  const [venues, setVenues] = useState([]);
  const [wishlist, setWishlist] = useState([]);

  const apiKey = "4P5afjX98PHm5yhdSLbee6G9PVKAQGB7";

  useEffect(() => {
    const fetchData = async () => {
      const url = "https://app.ticketmaster.com/discovery/v2/suggest";
      const params = `apikey=${apiKey}&locale=*&keyword=${slug} || ""`;

      try {
        // Fetch attractions only if slug matches 'sport'
        if (slug === "sport") {
          const attractionsRes = await fetch(
            `${url}?${params}&resource=attractions`
          );
          const attractionsData = await attractionsRes.json();
          setAttractions(attractionsData._embedded?.attractions || []);
        }

        // Fetch venues only if slug matches 'sport'
        if (slug === "sport") {
          const venuesRes = await fetch(` ${url}?${params}&resource=venues`);
          const venueData = await venuesRes.json();
          setVenues(venueData._embedded?.venues || []);
        }

        // Fetch events for any slug
        const urlEvents =
          ` https://app.ticketmaster.com/discovery/v2/events.json?apikey=${apiKey}&local=e=&size=20&classificationName=${slug} || ""`;
        const eventRes = await fetch(urlEvents);
        const eventData = await eventRes.json();
        setEvents(eventData._embedded?.events || []);
      } catch (error) {
        console.error("Error fetching data:", error);
      }
    };
    fetchData();
  }, [slug]);

  const toggleWishlist = (id) => {
    setWishlist((prev) =>

```

```

        prev.includes(id) ? prev.filter((x) => x !== id) : [...prev, id]
    );
}

if (slug !== "sport") {
    return <h1>{categoryName}</h1>;
}

return (
    <>
        <h1>{categoryName}</h1>

        <section className="Attraction">
            <h2>Attraksjoner</h2>
            <div>
                {attractions.map((item) => (
                    <CategoryCardTwo
                        key={item.id}
                        item={item}
                        isSaved={wishlist.includes(item.id)}
                        onSave={() => toggleWishlist(item.id)}
                    />
                )))
            </div>
        </section>

        <section className="Arrangement">
            <h2>Arrangementer</h2>
            <div>
                {events.map((item) => (
                    <CategoryCard
                        key={item.id}
                        item={{
                            ...item,
                            date: item.dates?.start?.toLocaleDate,
                            time: item.dates?.start?.localTime,
                            venue: item._embedded?.venues?.[0]?.name,
                            city: item._embedded?.venues?.[0]?.city?.name,
                            country: item._embedded?.venues?.[0]?.country?.name,
                        }}
                        isSaved={wishlist.includes(item.id)}
                        onSave={() => toggleWishlist(item.id)}
                    />
                )))
            </div>
        </section>

        <section className="Venues">
            <h2>Spillesteder</h2>
            <div>
                {venues.map((item) => (
                    <CategoryCardTwo
                        key={item.id}
                        item={item}
                        isSaved={wishlist.includes(item.id)}
                        onSave={() => toggleWishlist(item.id)}
                    />
                )))
            </div>
        </section>
    </>
)

```

```
    );  
}
```

- In this adjusted code:
- The `fetchData` function is structured to fetch attractions and venues only if the `slug` matches "sport".
- Events are fetched regardless of the `slug`.
- Sections for attractions, events, and venues are conditionally rendered based on whether the `slug` is "sport". If the `slug` is not "sport", only the category name is rendered.
- Ensure your routing and URL parameters (`slug`) are correctly set up in your React Router configuration so that `useParams()` receives the correct value. Adjust the conditional rendering and data fetching logic further based on your specific requirements and API responses.

- **Du sa:**

- nei det er riktig for slug sport, men ikke for teater og show og musikk

- **ChatGPT sa:**

- Takk for avklaringen! Hvis kun "sport"-slugen viser riktige data (events, venues og attractions), men "music" og "Arts & Theatre" (teater/show) ikke viser resultater som forventet, er det trolig relatert til **hvordan slug brukes i API-kallene**, spesielt med Ticketmaster sin `suggest-` og `events.json`-endepunkt.
- La oss se på noen sannsynlige problemer og løsninger:
- 

- **1. slug-verdiens format i API-kallet**

- Du bruker `slug` direkte i API-spørringen som en **keyword** (`keyword=${slug}`).
- `slug` fungerer greit for "sport" (fordi det er et enkelt ord Ticketmaster gjenkjenner).
- Men "Arts & Theatre" inneholder **spesialtegn** og mellomrom.
- **Løsning:** Encode `slug`'en korrekt i URL.
- `js`
- `KopierRediger`
- `const params = `apikey=${apiKey}&locale=*&keyword=${encodeURIComponent(slug || "")}`;`

- 

- **2. slug brukes for classificationName i events.json**

- I denne linjen:
  - `js`
  - `KopierRediger`
  - `const urlEvents = `https://app.ticketmaster.com/discovery/v2/events.json?apikey=${apiKey}&local=e=*&size=20&classificationName=${slug || ""}`;`
- `classificationName` må være nøyaktig det Ticketmaster forventer – f.eks. "Arts & Theatre".
- Mellomrom og & må **URL-encodes** også her.
- **Løsning:**
- `js`
- `KopierRediger`
- `const urlEvents = `https://app.ticketmaster.com/discovery/v2/events.json?apikey=${apiKey}&local=e=*&size=20&classificationName=${encodeURIComponent(slug || "")}`;`

- - 🔎 **3. Valider at data faktisk finnes i attractionsData.\_embedded?.attractions osv.**
  - Noen ganger får du tomt svar hvis søkeordet ikke matcher internt i API-et. Legg inn konsoll-logging:
    - js
    - KopierRediger
    - ```
console.log("slug:", slug);
console.log("attractionsData:", attractionsData);
console.log("eventData:", eventData);
```
  - 
  - 📋 **Oppsummert tiltak (endre disse linjene):**
  - **Før:**
  - ```
const params = `apikey=${apiKey}&locale=*&keyword=${slug || ""}`;
const urlEvents =
`https://app.ticketmaster.com/discovery/v2/events.json?apikey=${apiKey}&local
e=*&size=20&classificationName=${slug || ""}`;
```
  - **Etter:**
  - ```
const params = `apikey=${apiKey}&locale=*&keyword=${encodeURIComponent(slug
|| "")}`;
const urlEvents =
`https://app.ticketmaster.com/discovery/v2/events.json?apikey=${apiKey}&local
e=*&size=20&classificationName=${encodeURIComponent(slug || "")}`;
```
  - 
  - **Bonus: Sjekk slug-verdier i categories.js**
  - Det må være akkurat det Ticketmaster API kjenner igjen. Eksempel:
  - js
  - KopierRediger
  - ```
const categories = [
{
  id: 1,
  name: "Musikk",
  slug: "music"
},
{
  id: 2,
  name: "Sport",
  slug: "sport"
},
```

```

        id: 3,
        name: "Teater>Show",
        slug: "Arts & Theatre" // denne MÅ matche det Ticketmaster forventer
    }
];

```

- Vil du at jeg skal hjelpe deg å se opp hvilke `classificationName`-verdier som er gyldige hos Ticketmaster?
- 4o

#### 4. Spørte chat: hva betyr encodeURIComponent

##### ChatGPT sa:

- `encodeURIComponent` er en JavaScript-funksjon som **"rømmer" (eller koder) spesialtegn i en tekststreng** slik at den trygt kan brukes som del av en URL, særlig i spørring (query string).
  - 🔑 **Hva gjør den?**
  - Den **konverterer spesialtegn** som mellomrom, &, /, ?, =, # osv. til deres URL-sikre representasjoner.
  - 
  - 🔎 **Eksempel**
    - js
    - KopierRediger
    - const slug = "Arts & Theatre";
 const safeSlug = encodeURIComponent(slug);
 console.log(safeSlug); // Output: Arts%20%26%20Theatre
  - 🔎 **Forklaring:**

| Tegn  | Hva det betyr | <code>encodeURIComponent</code> |
|-------|---------------|---------------------------------|
| Space | Mellomrom     | %20                             |
| &     | Og-tegn       | %26                             |
| /     | Skråstrek     | %2F                             |
| •     |               |                                 |

- 🔎 **Hvorfor er det viktig?**

- Når du setter en slik verdi i en API-URL:
  - js
  - KopierRediger
  - const url = `...?keyword=\${slug}`;
- Så vil en slug som "Arts & Theatre" ødelegge URL-en, fordi:
  - & blir tolket som nytt parameter
  - Resultatet blir **feil spørring** til API-et
  - Ved å bruke:
    - js
    - KopierRediger

- `keyword=${encodeURIComponent(slug)}`
- ...sørger du for at `slug` fungerer uansett hva det inneholder.
- 
- Vil du se en konkret forskjell i hvordan en riktig og feil-formatert URL påvirker resultatet?
-