

Eksamensdokument Gruppe 8

Bestående av:

Jan Luis Aragon Ferrer – Kandidatnummer: **233613** – Github-alias: **flaveur**

Joakim Skjær Svendsen – Kandidatnummer: **222750** – Github-alias: **jokke660**

Johannes Olin – Kandidatnummer: **232295** - Github-alias: **johannesolin**

Victor Wilhelmsen – Kandidatnummer: **233163** – Github-alias: **Victorwil**

Vanskelighetsgrad:

Gruppen vår har valgt å strekke oss etter **vanskelighetsgrad A.**

Githubprosjekt

https://github.com/flaveur/Gruppe_8_UiN_Eksamen

Fordeling av oppgave:

Felles – Bidratt med oppsett av Sanity, mappestruktur, routing og package installasjon.

Jan - Dashboard.jsx, EventCard.jsx, Layout.jsx, Footer.jsx, lib-mappe og tilsvarende styling.

Johannes - ArtistCard.jsx, Category.jsx, CategoryPage.jsx og tilsvarende styling.

Joakim - Eventdetails.jsx, Header.jsx, Home.jsx og tilsvarende styling.

Victor – NotFound.jsx, SanityEventDetails.jsx, Search.jsx, SearchResults.jsx og tilsvarende styling.

Redegjørelser og forutsetninger

Vi erfarte en liten del av oppgaveteksten som var uklar:

Bruk av Ticketmaster Discovery API

Vi opplevde at oppgaveteksten var uklar når det gjaldt bruk av Ticketmaster Discovery API. Det ble kun opplyst at vi skulle benytte API-et, uten veiledning om hvordan vi skulle finne riktige ID-er eller søkeparametere.

For å løse dette benyttet vi API Explorer v2 (<https://developer.ticketmaster.com/api-explorer/v2/>), som viste seg å være et langt mer brukervennlig verktøy enn å se på JSON-kode. Der fant vi hvordan vi kan filtrere på nøkkelord, kategorislugs og andre parametere, slik at vi kan hente ut nødvendig arrangementsdata som segment-ID-er og event-ID-er. Bortsett fra denne avklaringen, var det ingen ytterligere forutsetninger, og oppgaven var gjennomgående utfordrende.

Kilder

Mozilla. (u.å.-a). *Nullish coalescing operator (??)*. MDN Web Docs.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Nullish_coalescing

Mozilla. (u.å.-b). *Optional chaining (?.)*. MDN Web Docs.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Optional_chaining

Mozilla. (u.å.-c). *Using Fetch - Checking that the fetch was successful*. MDN Web Docs.

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch#checking_that_the_fetch_was_successful

Mozilla. (u.å.-d). *Window.localStorage*. MDN Web Docs.

<https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

React. (u.å.). *Fetching data with useEffect*.

<https://react.dev/reference/react/useEffect#fetching-data-with-effects>

Stack Overflow. (2021, desember 22). *How do you add toggle functionality to a wishlist button to add and remove items?* <https://stackoverflow.com/questions/70447055/how-do-you-add-toggle-functionality-to-a-wishlist-button-to-add-and-remove-items>

TanStack. (u.å.). *React Query: Background Fetching Indicators*.

<https://tanstack.com/query/latest/docs/framework/react/guides/background-fetching-indicators>

Ticketmaster. (u.å.-a). *Discovery API v2 overview*.

<https://developer.ticketmaster.com/products-and-docs/apis/discovery-api/v2/>

Ticketmaster. (u.å.-b). *Search Events v2*. <https://developer.ticketmaster.com/products-and-docs/apis/discovery-api/v2/#search-events-v2>

Ticketmaster. (u.å.-c). *Segment List v2*. <https://developer.ticketmaster.com/products-and-docs/apis/discovery-api/v2/#segment-list-v2>

AI

Claude Sonnet 3.7

Prompt: "(Limte inn koden over for kontekst). Hvordan implementerer jeg fjerning av passordfeltet før det returneres?"

Bruk: linje 58-64 sanity.js, grunnen til bruk var fordi vi var usikre på om vi forstod konseptet, så vi spurte Claude om implementeringen.

Microsoft Copilot

Microsoft Copilot anbefalte oss å sette opp en felles variabel-fil, slik at det blir lettere for alle i gruppa å følge en slags standard for stilsetting av sidene våre.

Prompt: "Jeg skal implementere bruken av scss i prosjektet mitt. Hvordan gjør jeg det og hvordan bør jeg settet det opp med tanke på at jeg har egen mappe for styling og egen mappe for components?"

Bruk: Variables.scss i styles. grunn: usikre på hvordan vi skulle sette opp mappestruktur, så på Zelda-API mappe fra kursets Github-Repo, men dobbeltsjekke med CoPilot for implementering.

Hjertesymbol SVG. Dette ble foreslått av Microsoft Copilot, egentlig med uhell da vi prøve å lage et symbol med generativ AI.

Prompt: "Lag et hjerte som SVG, som jeg kan bruke til en favorittfunksjon i mitt react-prosjekt" (Joakim)

Bruk: SanityEventDetails.jsx linje 149-153.

