Exam in “utvikling av interaktive nettsider"

Github repo :
https://github.com/CakeDsx/Gruppe16_Eksamen_UIN

Group members(group 16) :

Martine Müller - Mulleirno1 - studentId: 223045
Martine Resberg Kihl - mkihl98 - studentId: 172454
Philip Viet Ngyuen - CakeDsx - studentId: 223049

Grade goal : C

We didn't have too high expectations since we found this class to be pretty difficult, however as we were working we realized we understood more, and we could set our goal higher, we weren't able to complete the genre page or make the genres work in the way we wanted to, but we were able to pull out each users favorite movies and compare them to each other, we were also able to make the API’s work and understand the use of sanity better. We used a lot of base code from class with changes adjusted based on searching and new code lines.
Seeing that we can add new users in our sanity project and see them appearing has been pretty fun and finally making the API work felt great! Overall we are pretty happy with the work we put into this and we all worked at school together so the commits might be somewhat off from how much work everyone put in. We all put in just about equally as much effort and work into this project.

Overall we were pretty happy with the work we got done and got further than expected. Going to Veiledninger helped clear our minds when we were pretty stuck with a few things and we got our questions answered well which helped us along the way.

Sanity :

For sanity to work within our website, you would need to add a new user in users then add your name, image, favorite movies, genres and wishlist.
For the movie images you would need to write the title of the movie exactly as they've written in imdb movies as well as when wanting a movie to appear in the genres page, you would need to go into the movies themselves and give them the category you see fit. This way they will also appear when clicking on the genre they belong to since we have used referencing in our schemas.

Ps: do add an image as we´ve made it the clickable part and you would also be needing the images URL.

Problems :

We are not great at using branches, so some code was simply tested until it worked and then pushed. We do see this as a weakness but we did sometimes attempt using the branches, but overall we did prefer to just use the main branch.

We do also see that we missed the mark with the writeClient part, since that would be the thing containing the token and so on and that we could have simply used import writeclient ,or understood the use of client a bit more, where we now have used the entire setup of JSON and so on, however, we are simply happy our code got to where it did.

The genre page was fine to set up, but we were not able to figure out how to make the button react and send info to sanity to update a user's favorite genre.

We succeeded in getting the titles of all movies when clicking on a specific genre, but we were not able to update our sanity when clicking favorite

We also attempted to use the way that ratings were made in the LEGODUDES sanity project to make the sanity be updated with the button click, but we did not make that work.

Another thing worth mentioning is that we were unsure about just how much was wanted in the moviecard, so most of the code you will find in the movecard component since that is something we continuously forgot to ask about in veiledning.

Main sources :

We used sources from, mozilla, stackoverflow, the class projects, previous projects along with w3schools as our main sources, using chatGPT as a tool to see where mistakes had been made in the code to correct them if we were stuck on something, this would allow us to add the code into our code and see if it suddenly would work or not, if it did we would put it into KDiff to see where the mistakes would appear, an example of this is here :

```
function MovieImage({ userId }) {
  const [favoriteMovies, setFavoriteMovies] = useState([])
  const [favoriteGenres, setFavoriteGenres] = useState([])
  const [wishList, setWishList] = useState([])
```

```
function MovieImage({ userId }) {
  const [favoriteMovies, setFavoriteMovies] = useState([]);
  const [favoriteGenres, setFavoriteGenres] = useState([]);
  const [wishList, setWishList] = useState([]);
```

References

API :
https://rapidapi.com/SAdrian/api/moviesdatabase/

Sanity set up sources :
https://www.youtube.com/watch?v=OcTPaUfay5I

Class -
https://github.com/toremake/LEGODUDES_sanity/blob/main/frontend/sanity/services/productServices.js

Javascript, schemas and so on sources -
https://github.com/toremake/LEGODUDES_sanity/blob/main/frontend/sanity/services/productServices.js

https://www.youtube.com/watch?v=OcTPaUfay5I

JSON.stringify(). (n.d.). W3Schools. Retrieved May, 2024, from

https://www.w3schools.com/js/js_json_stringify.asp

(n.d.). dhiwise. Retrieved May, 2024, from

https://www.dhiwise.com/post/how-to-use-json-server-in-frontend-development

Using some and filter together was inspired from the link provided in the exam paper. *Array.prototype.some() - JavaScript | MDN*. (2023, November 27). MDN Web Docs. Retrieved May, 2024, from https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/some

Communicating with a JSON Server with GET, POST, DELETE, and PATCH Requests...It's Easier Than You Think. (2022, July 10). DEV Community. Retrieved May, 2024, from https://dev.to/nickwarren47/communicating-with-a-json-server-with-get-post-delete-and-patch-requestsits-easier-than-you-think-3ca2

Crockford, D. (2024, March 17). *JSON.stringify() - JavaScript | MDN*. MDN Web Docs. Retrieved May, 2024, from https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify

How to get image from url in ReactJS. (2023, November 11). Altcademy.com. Retrieved May, 2024, from https://www.altcademy.com/blog/how-to-get-image-from-url-in-reactjs/#fetching-images-dynamically

How to set up Font Awesome in React. (2023, January 7). DEV Community. Retrieved May, 2024, from https://dev.to/davidemaye/how-to-set-up-font-awesome-in-react-5a8d

JavaScript Array map() Method. (n.d.). W3Schools. Retrieved May, 2024, from https://www.w3schools.com/jsref/jsref_map.asp

Making a POST request using json-server. (2022, December 11). DEV Community. Retrieved May, 2024, from https://dev.to/ldakanno/making-a-post-request-using-json-server-h7c

(we used a lot of the different documents from the LEGODUDES but i'm linking the ones that helped the most in this code, here it benign the const used filter and map in one along with the use of making {...item} an individual array and into a new one})
Marius, T. (n.d.). *LEGODUDES*. github. Retrieved May, 2024, from https://github.com/toremake/LEGODUDES_sanity/blob/main/frontend/sanity/services/productServices.js

Queries | HTTP API. (2024, March 28). Sanity. Retrieved May, 2024, from https://www.sanity.io/docs/http-query

uin23ak4_booksearch. (n.d.). github. Retrieved May, 2024, from https://github.com/Mullerino1/uin23ak4_booksearch_MULLERN/blob/main/start/src/Components/Layout.jsx

javascript - split path name to get routing parameter. (2011, July 31). Stack Overflow. Retrieved May, 2024, from https://stackoverflow.com/questions/6888783/split-path-name-to-get-routing-parameter