

Eksamen UIN 2023

Gruppe 6

*

Gruppemedlemmer: Annette Liv Aune (annetteaune)

Jeg har gått for A-kravene, og mener jeg har løst alle oppgavene.

Forklaringer for det funksjonelle:

Lagrede brukere som kan logges inn med, er annettela@hiof.no og pk@epost.no. Har laget et registreringssystem som lagrer brukere på med e-post og brukernavn. Alle brukere kan lagre favoritter individuelt.

Velger å gå noe annerledes frem med «buy»-knapper i shop-kort. Ettersom jeg benytter APIet, får jeg tilgang til flere butikker spillet er tilgjengelig i, og syns derfor det er lite hensiktsmessig å låse en bruker til én butikk. Fra et brukerperspektiv ønsker jeg også å se litt informasjon om spillet uten å forlate appen jeg allerede er i, før jeg går til en potensiell kjøpside. «Buy»-knappen byttes derfor ut med en «View»-link, som fører til en side med informasjon om spillet, og som lister opp alle tilgjengelige kjøpsider med direkte linker til disse.

Syns det er sørgelig med et spillbibliotek hvor en ikke kan søke etter spill, så velger å implementere dette for moro skyld. Lar bevisst være å søke mens man skriver, for å unngå å bruke opp for mange api-requests.

Velger å returnere ulikt antall spill enn spesifisert i oppgaveteksten på shop-siden, da jeg heller vil returnere flere kategorier med færre spill, enn én kategori med mange. Returnerer likevel totalt flere spill enn minimumskravet.

Legger inn bilder manuelt i sanity for MyGames, disse er hentet fra APIet, og nevner dette i footer, samt attribute-felt i sanity.

Favorittikoner og logo er laget av meg og trenger derfor ingen henvisning på siden. Font i logo er fra: <https://www.1001fonts.com/vp-pixel-pro-font.html>

Ettersom token fra Sanity er hardkodet, får jeg varsel i konsollen om det. Men da dette ikke skal publiseres på noen måte, så velger jeg å ignorere varselet.

Forklaring for valg i kode:

Array.length istedenfor groq-spørring:

Jeg implementerte først groq-spørring etter antall spill og favoritter (legger ved skjermbilde av kjørende kode fra prosjektet mitt for å understreke at jeg har testet det), men valgte å bruke `array.length` i stedet, da dette gjør akkurat det samme, men med mye mindre kode, og uten flere requests til Sanity. Jeg oppfatter derfor `.length` som et langt mer fornuftig valg enn det som ble foreslått i oppgaveteksten.

```
//hente antall spill
//kilde: https://www.sanity.io/docs/query-cheat-sheet
export const fetchGamesCount = async () => {
  const data = await client.fetch(`count(*[_type == 'game'])`
);
  return data;
};
```

UseCallback:

Grunnet warnings i terminal om manglende dependancy i enkelte `useEffect`-hooks (for å unngå loops), har jeg (på forslag fra terminalen) prøvd å lære om `useCallback`, og har implementert dette flere steder i prosjektet.

API:

Velger å bruke `description_raw` fra api-kallet, da det virker risikabelt å hente inn den «ferdige» koden fra `description` uten å sanitere, ref:

- <https://stackoverflow.com/questions/69300076/how-can-i-use-html-tags-from-api-and-display-on-page>
- <https://blog.logrocket.com/using-dangerouslysetinnerhtml-in-a-react-application/>

Kilder:

Oppsett for react/sass/sanity:

- <https://webtricks.blog/start-et-prosjekt-med-react-sanity-og-sass-fra-scratch/>

Oppsett client:

- https://github.com/toremake/UIN2023_sanity_create/blob/main/frontend/src/utlis/sanity/client.js
- <https://www.sanity.io/docs/presenting-images>
- <https://webtricks.blog/oppdagere-et-array-felt-i-en-innholdstype-i-sanity-fra-et-react-grensesnitt/>
- https://github.com/toremake/UIN2023_sanity_create/blob/main/frontend/src/components/Sho.w.js
- <https://www.sanity.io/docs/query-cheat-sheet>

Schemas:

- <https://www.sanity.io/docs/schema-types>
- <https://www.sanity.io/docs/slug-type>
- https://github.com/toremake/UIN23_sanity_sandbox/blob/main/sanity/schemas/products.js

UseCallback:

- <https://react.dev/reference/react/useCallback>
- https://www.youtube.com/watch?v=_AyFP5s69N4
- https://www.w3schools.com/react/react_usecallback.asp

Breadcrumbs.js:

- <https://www.makeuseof.com/create-breadcrumbs-in-reactjs/>

WordCloud.js:

- <https://github.com/madox2/react-tagcloud#readme>
- <https://github.com/madox2/react-tagcloud/blob/master/examples/src/tag-props.js>

GamePage.js:

- <https://stackoverflow.com/questions/46849286/merge-two-array-of-objects-based-on-a-key>
- <https://stackoverflow.com/questions/21895233/how-to-split-string-with-newline-n-in-node>

MyGames.js:

- https://www.w3schools.com/jsref/jsref_slice_array.asp

GameCard.js:

- https://www.w3schools.com/jsref/jsref_isarray.asp
- https://www.w3schools.com/js/js_typeof.asp

LoginPage.js:

- <https://stackoverflow.com/questions/50644976/react-button-onclick-redirect-page>

Register.js:

- https://github.com/toremake/UIN2023_sanity_create/blob/main/frontend/src/components/AddShow.js
- <https://www.w3resource.com/javascript/form/email-validation.php>

SearchResults.js:

- <https://stackoverflow.com/questions/63359138/react-closing-a-dropdown-when-click-outside>

Search.js:

- https://github.com/annetteaune/uin23ak4_moviesearch_aune/blob/main/src/components/Search.js

CSS (profile.scss):

- <https://stackoverflow.com/questions/26421274/css-circular-cropping-of-rectangle-image>

Annet:

- <https://sentry.io/answers/defining-proper-key-in-props/>
- <https://webaim.org/resources/contrastchecker/>
- <https://addons.mozilla.org/en-US/firefox/addon/wave-accessibility-tool/>
- <https://www.color-hex.com/color-palette/61294>
- <https://www.1001fonts.com/vp-pixel-pro-font.html>

