

RAPPORT INFO 134 (253)&(241)

DEL 1

Der vår kode referer til javascript-filen object.js, referer MongoDB-utgaven tilfilen profile.js. Der kjører profile.js den innebygde javascript-metoden XMLHttpRequest(), som er en konstruktør. Der bruker den open()-funksjonen for å sende en request til en nettside eller server, i dette tilfeller wildboy.uib/mongodb , med et brukernavn som variabel og bruker deretter responseType til å få informasjonen fra server til å bli en JSON-object i den notasjonen/formateringen. Deretter bruker koden dot-notasjon for å hente elementet "rows", som består av en array bestående av elementer, der noen er andre arrayer. Ut fra rows.response[0], opprettes et object av alt som er inne i rows, og kalles 'profil'. Så brukes igjen dot-notasjon for å hente verdiene brukeren har lånt ved hjelp av løkken i form av

```
" for (movie_id_index in profile.loans) { movie_id = profile.loans[movie_id_index] "
```

Denne henter id på alle filmer brukeren har lånt.

Når brukeren søker i søkefeltet så legges søkeparameterne med i url'en til search_results.html fks: filmbibliotek.no/search_results.html?film_title=søkestreng. Når search_results har lastet inn sin html body så kjører funksjonen show_search_results i search_results.js, her kjøres så funksjonen get_query_string_parameters fra query_params.js denne retunerer parameterene i url'en. Parameterene lagres så i en variabel kallt query_params. Show_search_results sammenlikner så alle filmtitlene i movies_object i object.js mot query_params, dersom en filmtittel inneholder søkeparameterne så skapes det en a tag med filmens tittel som tekst og som refererer til show_movie.html?id=movie.id der movie.id er id'en til filmen som tittelen inneholdt søkeparameterene. Dette a elementet føyes så til som et barn av et li element som igjen føyes til som et barn av den uordnete listen med id res_list i search_results.html. Ønsker å påpeke at det kan bli laget fler slike li elementer da search_results .js ittererer mellom alle filttittelene i movies_object og skaper elementene i en for løkke.

I index.html så ville jeg laget en dropp down meny som poppet opp på samme måte som når man trykker på søkeknappen, denne popper så opp om man trykker på min liste eller mine lån i søkefeltet. og fyller inn på samme måte som eksempel index som vi fikk med i siste zip, der det i profile.js henter bukernavn og lån fra databasen ved en XMLHttpRequest. Hovedforskjellen er at jeg ville brukt id'en til lånet så for å hente navnet på filmene brukeren har lånt og vist dette til brukeren med en link til filmsiden, ved å bruke en XMLHttpRequest noe al av

```
var xhr = new XMLHttpRequest();
xhr.open("GET", "http://wildboy.uib.no/mongodb/objects/?filter_id=" + filmid + "&limit=1",
true); xhr.responseType = "json";
xhr.onload = function() {
  film = xhr.response.rows[0];
  var ntitle = film.ntitle;
```

og så brukt ntitle til å fylle in li elementet på samme måte som i eksemplet.

Når det gjelder navigasjonsmiljøet midt på siden ville jeg ha skrevet om slik at det velges ut faste filmer som hentes fra nye objecter som er mindre en objects fra serveren, disse vil da oppretes ut fra fks 30 forskjellige id'er fra objects på wildboy. I og med at serveren ikke har satt opp egne objecter for sist tilgjengelige, sist lånt eller anbefalinger. I all_movies.js ville da linje 12 blitt endret slik at function addFilm(filmobjekter, where, start) og linje 17 til: for (movie_id in filmobjekter) {
der filmobjekter er en av de nye objekter.

I show_movie.html ville HTML-koden blitt identisk, men javascripten ville vært forskjellig.

Istedenfor å hente objektet movie_object fra filen object.js, ville koden ha laget en ny

XMLHttpRequest();, brukt var xhr = new XMLHttpRequest(); så

```
xhr.open("GET", "http://wildboy.uib.no/mongodb/objects/?filter_id=" + query_params.id, true);
og deretter.
```

```
movie_object = xhr.response.rows[0];
```

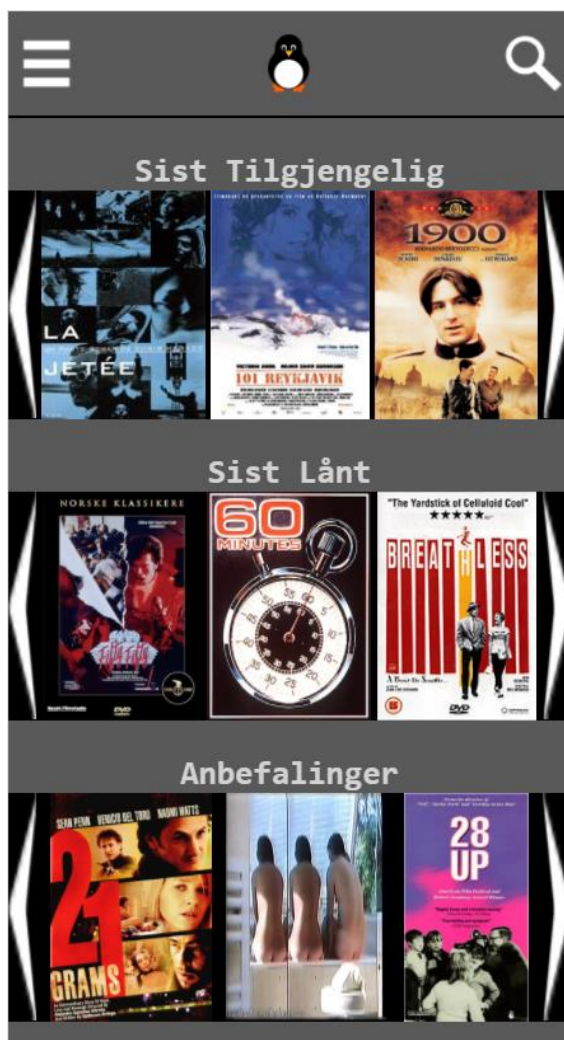
, og fått objektet fra rett index via query_params og parset/splittet det i mindre deler. Disse ville bli brukt på samme måte som om man hadde hentet det fra object.js.

I search_results.js ville vi ikke brukt movies_object men brukt filter funksjonen i

XMLHttpRequest noe al av http://wildboy.uib.no/mongodb/objects/?filter_otitle=filmtittel

der filmtittel erstattes med søkestrengen dette gjør at en ikke trenger å laste ned alle objectene i objects, desverre så må man da skrive inn hele filmtittelen helt riktig. Dette kan gjøres på samme måte for actor director og country.

DEL 3



Forsiden består av 3 ruller med bilder av filmer der hvis man klikker på høyre pil så lastes det inn ett nytt bilde på høyre side, og et bilde på høyre side fjernes. Klikker du venstre pil så lastes det inn det inn et bilde på venstre side og fjerner ett fra høyre. Dersom man klikker på et bilde i disse rullene kommer man videre til filmens side med dens informasjon

I toppen av siden er det en felt med en knapp på hver side.

Søk

Klikker man på høyre knapp så vises en søkebar som bruker søkefunksjonen quicksearch, her kan man søke på filmtittel samt skuespiller og regisør. Oppgave teksten nevnte kun film tittel, men vi mente at det var mer praktisk å kunne søke på disse tre fra forsiden. Dessuten om det blir for mye kan linje 127 og 128 i search_results.js fjernes for at kun filmtittel skal kunne søkes på.

Hvis du klikker venstre kanppen i toppen så åpnes menyen som viser mine reservasjoner, min bruker, min liste, mine lån og logg ut.



I bunnen av index.html så ser man rutiner for lån av filmer.

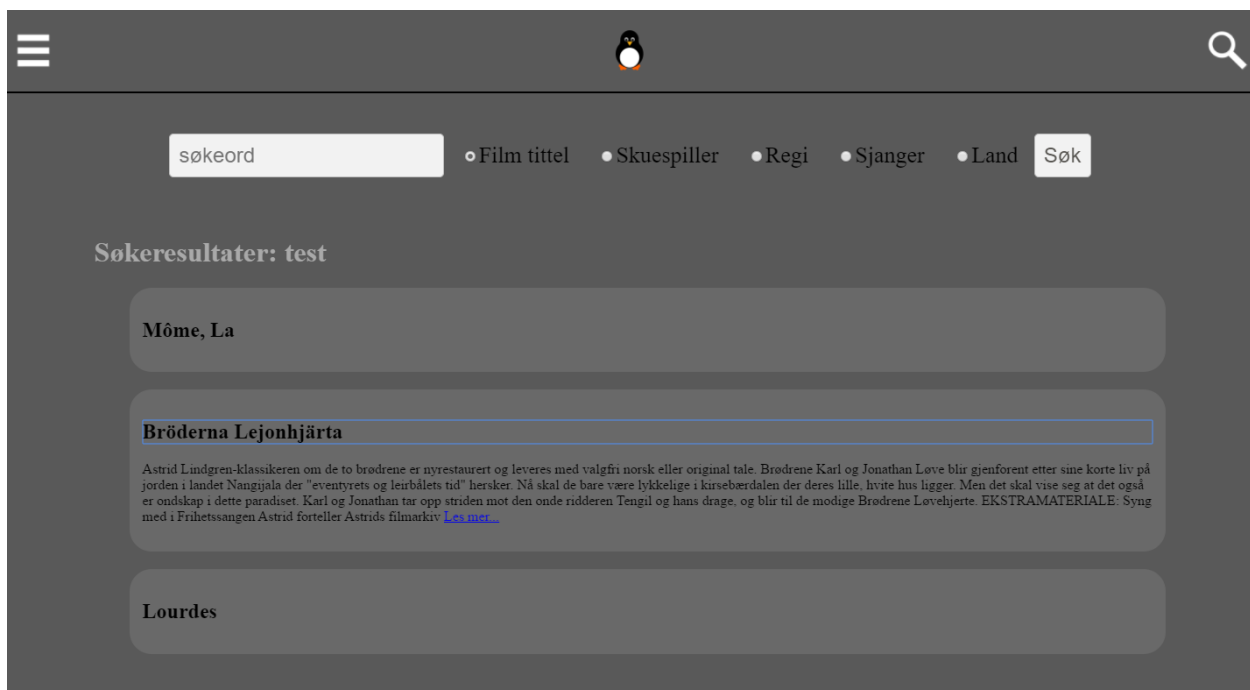
Søkesiden inneholder en avansert søkebar i toppen av siden, rett under top banneren. Her kan man så søke vidre på enten filmtittel, skuespiller, regisør, sjanger, eller land.

Under vises de forskjellige filmene som matcher ditt tidligere søk. Dersom man vil spesifikt søke etter filmtittel, skuespiller, regisør, sjanger eller land etter å ha kommet hit fra søket på forsiden må man søke på nytt.

Rutiner for lån

- Følgende kan låne: staben ved Institutt for informasjons- og medievitenskap, studenter ved instituttet, på grunnlag av faglig begrunnelse, andre ved UiB etter særlig avtale for et avgrenset tidsrom eller for et avgrenset prosjekt.
- Bestillinger vil bli effektivert hver arbeidsdag mellom klokken 14:00 og 15:00.
- Ansatte ved instituttet får filmen lagt i posthyllen, studenter og andre får tilbakemelding om hvor og når filmen kan hentes.
- For alt utlån gjelder en generell regel om lånetid på 1 uke, som kan forlenges med ytterligere en uke. Ønskes en film lånt over en lengre periode må dette begrunnes, og hver enkelt forespørsel vil bli vurdert.
- Film skal leveres tilbake i posthyllen merket retur av film.

Dersom man klikker på en av filmtittelen så åpnes en tekst om filmen med en link til filmens side.



La inn søkebar og menybaren fra index.html i show_movie.html for å lett kunne søke seg til andre filmer fra filmsiden en allerede er på . La inn et enkelt kommentarfelt, der brukere som er logget inn lett kan skrive en kort vurdering av filmen. Håper også på at det kan implementeres en lagringsfunksjon som tar med dato, brukernavn og vurdering og lagrer det i review-objektet.



The screenshot shows a dark-themed web interface for movie reviews. At the top, under the heading "Brukervurderinger:", there is a list of previous reviews, including one from "2008-11-17, Bruker: xyz001" with a rating of "VELDIG BRA". Below this, the section "Din vurdering:" contains a large, empty white text area for the user's review. At the bottom left of the form is a "send inn" button.

show_movie.html henter alt den trenger fra objektene, og renderer det i siden. Bilder, hentes som i index.html fra en nettside.