

# Eksamensdokument\_Gruppe16

Gruppemedlemmer

Joachim André Arnesen Fjell

Nojoud Ayman Banishamsa

Emnekode:

ITM30617

Emnetittel:

Utvikling av Interaktive nettsteder



## Contents

Utfordringer:.....	3
Kilder.....	4
Kildebrukt i css.....	4
Dashboard .....	5
GameCard og GameTile.....	6
GamePage .....	6
GameShop, MyFavourites og MyGames .....	7
Layout .....	7
App.js .....	8

## Utfordringer:

Vår gruppe har gått etter Karakter:

D

Vi fikk ikke til å bruke slug når det skal brukes til flere components, fikk det samme resultatet på alle sidene, derfor har vi valgt å pathe på en annen måte i App.js. Hvor vi bare har brukt navnene på de ulike componentene. Linkingen mellom spillene og spill komponentet GamePage var noe vi ikke greide å løse i tide.

Nojoudb – Nojoud Ayman Banishamsa

Joafjell – Joachim André Arnesen Fjell

## Kilder

Kildebrukt i css

[https://github.com/joafjell/uin23ak4\\_moviesearch\\_fjell/blob/main/src/components/MoviesMainPage.js](https://github.com/joafjell/uin23ak4_moviesearch_fjell/blob/main/src/components/MoviesMainPage.js)

```
main {
  width: 100%;
  max-width: var(--max-width);
  display: flex;
  flex-wrap: wrap;
  flex-direction: column;
  margin: 0 auto;
}

.movie-card {
  display: flex;
  flex-direction: column;
  padding: 0 0 2rem 5rem;
  max-width: 500px;
  width: 100%;
}
```

<https://stackoverflow.com/questions/9067892/how-to-align-two-elements-on-the-same-line-without-changing-html>

```
body { /*body may pose problem depend on you context, there is no better then have a specif
  white-space: nowrap;
  overflow-x: auto;
}

#element1, #element2{ /*you can like put each one separately, if the margin for the first e
  display: inline-block;
  margin-left: 10px;
}
```

<https://cssreference.io/property/margin-left/#:~:text=margin%2Dleft%3A%20auto%3B,a%20fixed%20width%20is%20defined.>

```
margin-left: auto;
```

The **auto** keyword will give the left side a share of the **remaining space**.

When combined with `margin-right: auto`, it will **center** the element, if a fixed width is defined.

For breakpointsene til media queries

<https://testsigma.com/blog/css-breakpoints/>

### Which CSS breakpoints to use?

Although every device demands separate CSS breakpoints depending upon their screen resolution, it is not necessary to use CSS breakpoints media queries for individual device width. It is impossible to add CSS breakpoints to every new device as more and more are released. Instead, make an effort to select breakpoints based on web design and audience usage patterns for target devices in the target markets.

For devices with varying screen sizes, there are a few common breakpoints –

- **For Mobile Devices** – 320px—480px
- **iPads and Tablets** – 481px—768px
- **Laptops and small screen** – 769px—1024px
- **Large screens and Desktops** – 1025px—1200px
- **TV and Extra Large Screens** – 1201px and more

Brukte fonten fra denne siden, i de ulike breakpointsene

<https://matthewjamestaylor.com/responsive-font-size>

### Fluid font sizes per breakpoint

Screen width	Font size
320px (eg: iPhone 4 & 5)	16px
768px (eg: iPad portrait)	18px
1024px (eg: iPad landscape)	19px
1280px	20px
1536px	21px
1920px	23px
2560px	25px

### Dashboard

For å hente ut tilfeldige objekter fra Arrayen brukte vi litt av denne kodesnutten som inspirasjon

```
const array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
const shuffledArray = array.sort((a, b) => 0.5 - Math.random());
```

<https://dev.to/codebubb/how-to-shuffle-an-array-in-javascript-2ikj>

og

```
const slicedArray = array.slice(0, n);
```

<https://stackoverflow.com/questions/34883068/how-to-get-first-n-number-of-elements-from-an-array>

For å finne spill som var fav: true, så brukte vi w3school,  
[https://www.w3schools.com/jsref/jsref\\_filter.asp](https://www.w3schools.com/jsref/jsref_filter.asp)

#### Example 1

Return an array of all values in ages[] that are 18 or over:

```
const ages = [32, 33, 16, 40];
const result = ages.filter(checkAdult);

function checkAdult(age) {
  return age >= 18;
}
```

Try it Yourself »

### Definition and Usage

The `filter()` method creates a new array filled with elements that pass a test provided by a function.

The `filter()` method does not execute the function for empty elements.

The `filter()` method does not change the original array.

Og `.slice` koden ovenfor, den ble også brukt igjen innenfor samme component.

## GameCard og GameTile

Brukte denne koden for å sette opp componentene GameCard og GameTile.

[https://github.com/joafjell/uin23ak4\\_moviesearch\\_fjell/blob/main/src/components/MovieCards.js](https://github.com/joafjell/uin23ak4_moviesearch_fjell/blob/main/src/components/MovieCards.js)

```
16 lines (13 sloc) | 560 Bytes
Raw Blame

1  import { Link } from "react-router-dom";
2
3  export default function MovieCards({title, img, type, year, imdbID}){
4    return (
5      <article className="movie-card">
6        <Link to={imdbID}><h2 className="film-title">{title}</h2>
7        <img src={img} alt={title} className="images"/>
8        </Link>
9        <p>Type: <span>{type}</span></p>
10       <p>Utgivelses år: <span>{year}</span></p>
11       <Link to={imdbID}><button className="more-info-button">Click for more info</button></Link>
12     </article>
13   )
14 }
15
16 }
```

## GamePage

Her brukte vi et github repository fra en forelesning,

<https://github.com/ackarlse/gitcollab/blob/main/src/components/RecipePage.js//>

```
20 lines (17 sloc) 646 Bytes
1 import { useParams } from "react-router-dom"
2 export default function RecipePage({recipes}){
3   const {slug} = useParams()
4   const recipe = recipes.find((rec) => rec?.recipe?.label.replace(/\s/g, "-").toLowerCase() === slug)
5   console.log(recipe?.recipe?.totalTime)
6
7
8   return (
9     <section>
10      <h1>{recipe?.recipe?.label}</h1>
11      <img src={recipe?.recipe?.image} alt={recipe?.recipe?.label} />
12
13      <ul>
14        {recipe?.recipe?.ingredients.map((ing, index) => (
15          <li key={index}>{ing.text} vekt: {ing.weight}</li>
16        ))}
17      </ul>
18    </section>
19  )
20 }
```

## GameShop, MyFavourites og MyGames

Brukte denne koden fra GitHub.

[https://github.com/joafjell/uin23ak4\\_moviesearch\\_fjell/blob/main/src/components/MoviesMainPage.js](https://github.com/joafjell/uin23ak4_moviesearch_fjell/blob/main/src/components/MoviesMainPage.js)

```
17 lines (16 sloc) 730 Bytes
1 import MovieCards from "../MovieCards";
2 import SearchResults from "../SearchResults";
3
4 export default function MoviesMainPage({movies, setSearch, getMovies}){
5   return (
6     <>
7       <SearchResults setSearch={setSearch} getMovies={getMovies}/>
8       <section className="mainpage-movie">
9         {movies?.map((movie, index) => {
10           const {Title, Poster, Year, Type, imdbID} = movie;
11           //console.log(movie);
12           return <MovieCards key={index} title={Title} img={Poster} year={Year} type={Type} imdbID={imdbID} slug={movie?.movie?.label.replace(/\s/g, "-").toLowerCase()} />
13         })}
14       </section>
15     </>
16   );
17 }
```

## Layout

Brukte dette githubbet so inspirasjon til Layouten vår,

[https://github.com/joafjell/uin23ak4\\_moviesearch\\_fjell/blob/main/src/components/Layout.js](https://github.com/joafjell/uin23ak4_moviesearch_fjell/blob/main/src/components/Layout.js)

```
Code Blame 19 lines (17 loc) · 438 Bytes
1 import { Link, Outlet } from "react-router-dom";
2
3 export default function Layout(){
4   return(
5     <div className="container">
6       <header>
7         <Link to="/">
8           <h1 className="pageTitle">Moviepage</h1>
9         </Link>
10      </header>
11      <main>
12        <Outlet />
13      </main>
14      <footer>
15
16      </footer>
17    </div>
18  )
19 }
```

## App.js

Vi har brukt koden fra dette github repositoriet,

[https://github.com/joafjell/uin23ak4\\_moviesearch\\_fjell/blob/main/src/App.js](https://github.com/joafjell/uin23ak4_moviesearch_fjell/blob/main/src/App.js)

```
37 lines (29 sloc) 994 Bytes
1 import { useEffect, useState } from 'react';
2 import { Route, Routes } from 'react-router-dom';
3 import './App.css';
4 import './css/main.css';
5 import Layout from './components/Layout';
6 import MovieSite from './components/MovieSite';
7 import MoviesMainPage from './components/MoviesMainPage';
8
9
10 function App() {
11
12   const [movies, setMovies] = useState([])
13   const [search, setSearch] = useState('James Bond')
14
15   const getMovies = async() =>{
16     const response = await fetch('http://www.omdbapi.com/?s=${search}&type=movie&apikey=e3f58a23')
17     const movieInformation = await response.json()
18     setMovies(movieInformation.Search)
19   }
20
21   useEffect(() => {
22     getMovies()
23   }, [])
24
25   console.log(movies)
26
27   return (
28     <Routes>
29       <Route element={<Layout/>}>
30         <Route index element={<MoviesMainPage movies={movies} setSearch={setSearch} getMovies={getMovies}/>}>
31         <Route path=':slug' element={<MovieSite movies={movies}/>}>
32       </Route>
33     </Routes>
34   );
35 }
36
37 export default App;
```