

ENGSE203 - Mid-term (Take-home)

1. ส่วนนี้เป็น explorer ของ ตัว front-end และ back-end api

The image displays two file explorers side-by-side, comparing the directory structures of a front-end application and a back-end API.

Left Explorer (Front-end Project Structure):

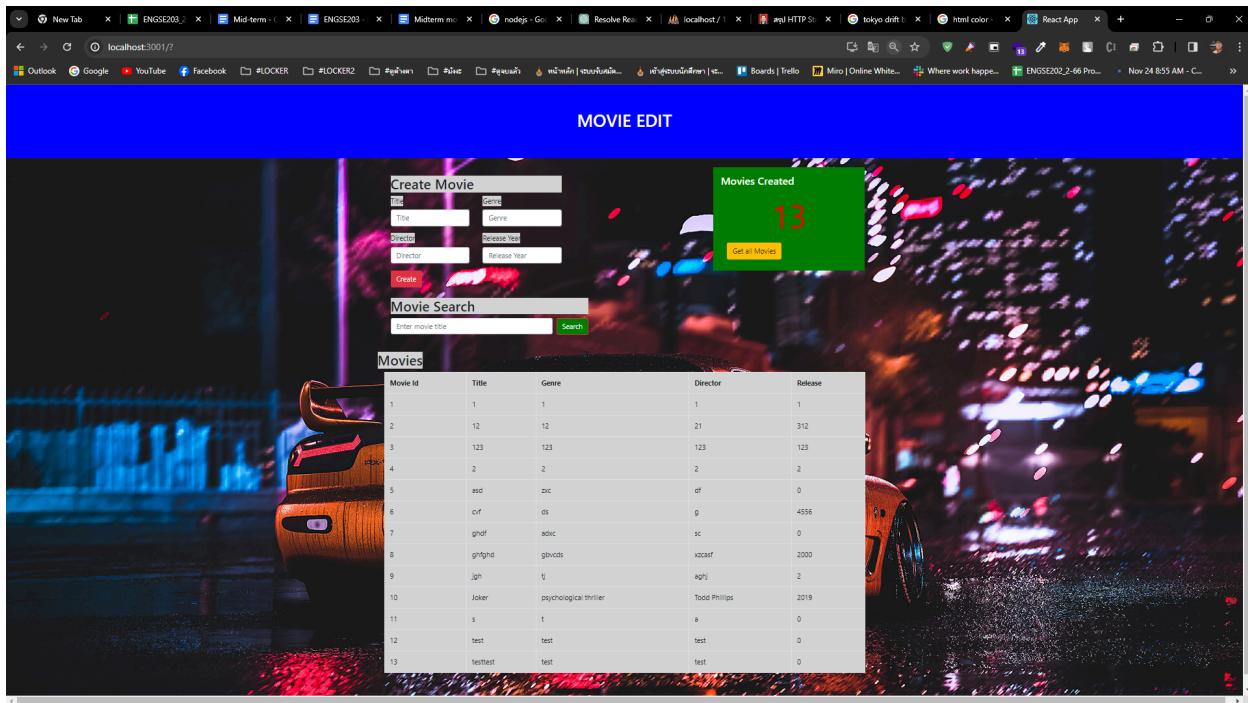
- MY-APP
 - build
 - node_modules
 - public
 - images
 - car.jpg
 - favicon.ico
 - index.html
 - logo192.png
 - logo512.png
 - manifest.json
 - robots.txt
 - src
 - components
 - AppError.js
 - CreateMovie.js
 - CreateUser.js
 - DisplayBoard.js
 - Header.js
 - Movies.js
 - SearchPage.js
 - Users.js
 - services
 - App backup.js
 - # App.css
 - App.js
 - App.test.js
 - # index.css
 - index.js
 - logo.svg
 - serviceWorker.js
 - setupTests.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md

Right Explorer (Back-end Project Structure):

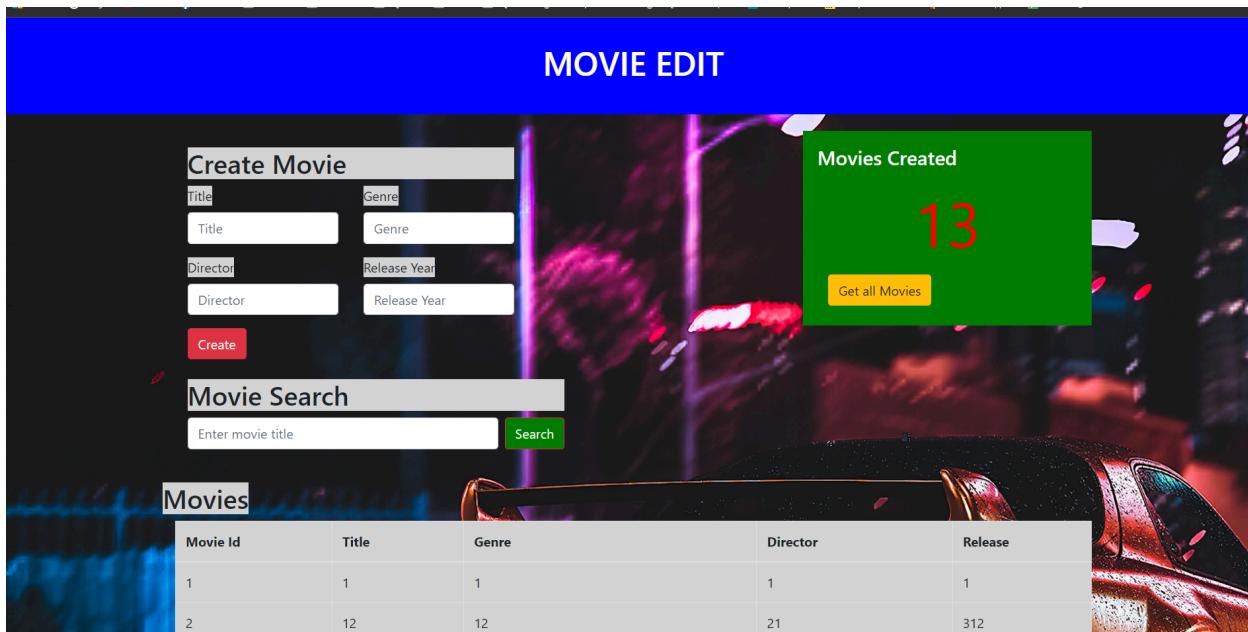
- LAB4
 - backend
 - myreact
 - node_modules
 - public
 - src
 - _.package.json
 - .gitignore
 - package-lock.json
 - package.json
 - README.md
 - react-nodejs-exam...
 - api
 - node_modules
 - repository
 - movie.js
 - dbconfig.js
 - env.js
 - gulpfile.js
 - index.js
 - package-lock.json
 - package.json
 - README.md
 - server.bundle.js
 - server.js
 - webpack.config.js
 - my-app
 - .dockerignore
 - .gitignore
 - Dockerfile
 - _.DS_Store

นาย ศุภวิชญ์ แก่นนาค 66543210031-1

2.ภาพหน้าเว็บไซต์(zoom out ภาพอาจจะเล็ก)



(zoom in)



3. ผลลัพธ์การทำงานของเว็บไซต์

Create Movie

Title: spiderman

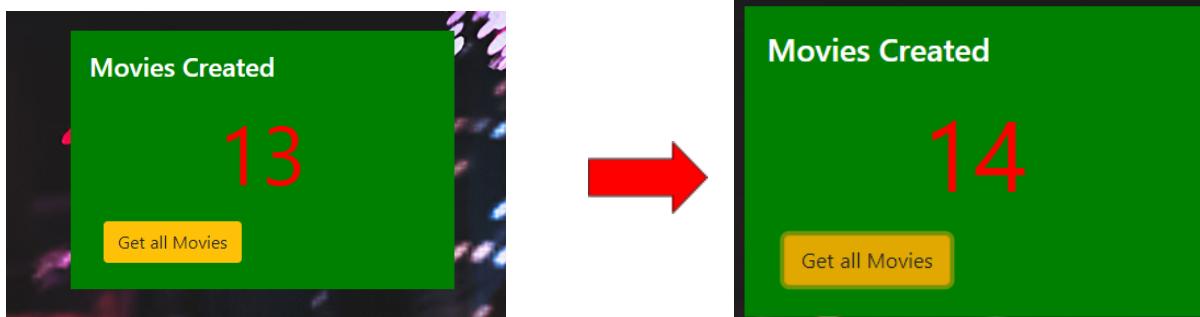
Genre: sci-fi

Director: Sam Raimi

Release Year: 2010

Create

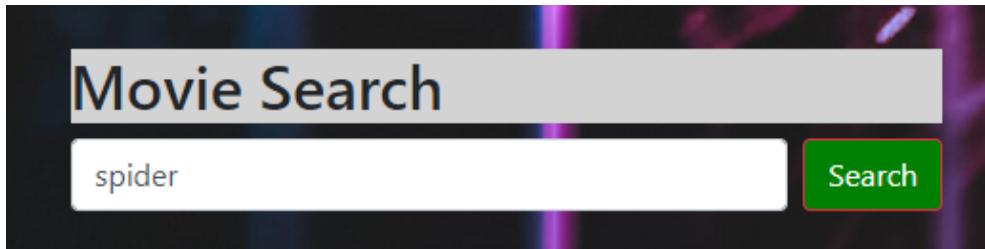
กรอกข้อมูลที่ต้องการเพิ่มและกดปุ่ม create เพื่อสร้าง



กด get all movies เพื่อทำการเรียกข้อมูลหนังทั้งหมด

Movie Id	Title	Genre	Director	Release
1	1	1	1	1
2	12	12	21	312
3	123	123	123	123
4	2	2	2	2
5	asd	zxc	df	0
6	cvf	ds	9	4556
7	ghdf	adxc	sc	0
8	ghfgdh	gbvcds	xzcasf	2000
9	jgh	tj	aghj	2
10	Joker	psychological thriller	Todd Phillips	2019
11	s	t	a	0
12	spiderman	sci-fi	Sam Raimi	2010
13	test	test	test	0
14	testtest	test	test	0

รายชื่อหนังที่เพิ่มมาอยู่ในวงลีดง



ใช้ keyword เป็น title ใน การค้นหา พิมพ์ title และ กด search

This image displays a movie creation form and a search interface. On the left, a "Create Movie" form is shown with fields for Title (spiderman), Genre (sci-fi), Director (Sam Raimi), and Release Year (2010). A red "Create" button is at the bottom. To the right, a "Movies Created" box shows the number "1" and a "Get all Movies" button. Below these, a "Movie Search" form has "spider" entered in the search bar, with a green "Search" button. The results are displayed in a table titled "Movies" with columns: Movie Id, Title, Genre, Director, and Release. The first row shows Movie Id 1, Title spiderman, Genre sci-fi, Director Sam Raimi, and Release 2010. The entire interface is set against a background of a shiny car.

ผลลัพธ์ที่ได้อบูตรงท่วงสีแดง

A large red arrow points from a yellow "Get all Movies" button on the left towards a table of movies on the right. The table is titled "Movies" and has columns: Movie Id, Title, Genre, Director, and Release. It lists 14 rows of movie data. The "Get all Movies" button is highlighted with a red border.

Movie Id	Title	Genre	Director	Release
1	1	1	1	1
2	12	12	21	312
3	123	123	123	123
4	2	2	2	2
5	asd	zxc	df	0
6	cvf	ds	g	4556
7	ghfdf	adxc	sc	0
8	ghfghd	gbvcds	xzcasf	2000
9	jgh	tj	aghj	2
10	Joker	psychological thriller	Todd Phillips	2019
11	s	t	a	0
12	spiderman	sci-fi	Sam Raimi	2010
13	test	test	test	0
14	testtest	test	test	0

กด get all movies อีกครั้ง
เพื่อเรียกข้อมูลทั้งหมดกลับมา

4. ส่วนของโค๊ดที่ได้แก้ไข และไฟล์ที่ได้สร้างขึ้นมาใหม่ (Front-end)

สร้างไฟล์ใหม่ CreateMovie.js ข้างใน component โดยมี base มาจาก CreateUser.js

```
import React from 'react';

const CreateMovie = ({ onChangeForm, createMovie }) => {
  return (
    <div className="container">
      <div className="row">
        <div className="col-md-7 mrgnbtm">
          <h2 style={{ backgroundColor: 'lightgray' }}>Create Movie</h2>
          <form>
            <div className="row">
              <div className="form-group col-md-6">
                <label htmlFor="exampleInputEmail1" style={{
backgroundColor: 'lightgray' }}>Title</label>
                <input
                  type="text"
                  onChange={(e) => onChangeForm(e)}
                  className="form-control"
                  name="title"
                  id="title"
                  aria-describedby="emailHelp"
                  placeholder="Title"
                />
              </div>
              <div className="form-group col-md-6">
                <label htmlFor="exampleInputPassword1" style={{

backgroundColor: 'lightgray' }}>Genre</label>
                <input
                  type="text"
                  onChange={(e) => onChangeForm(e)}
                  className="form-control"
                  name="genre"
                  id="genre"
                  placeholder="Genre"
                />
              </div>
            </div>
            <div className="row">
```

นาย គុរីមួយៗ កែនណាគ 66543210031-1

```
<div className="form-group col-md-6">
    <label htmlFor="exampleInputEmail1" style={{ 
backgroundColor: 'lightgray' }}>Director</label>
    <input
        type="text"
        onChange={(e) => onChangeForm(e)}
        className="form-control"
        name="director"
        id="director"
        aria-describedby="emailHelp"
        placeholder="Director"
    />
</div>
<div className="form-group col-md-6">
    <label htmlFor="exampleInputPassword1" style={{ 
backgroundColor: 'lightgray' }}>Release Year</label>
    <input
        type="text"
        onChange={(e) => onChangeForm(e)}
        className="form-control"
        name="release_year"
        id="release_year"
        placeholder="Release Year"
    />
</div>
</div>
<button
    type="button"
    onClick={(e) => createMovie()}
    className="btn btn-danger">
    Create
</button>
</form>
<br><br>
</div>
</div>
</div>
);
};

export default CreateMovie;
```

การทำงานของโค้ดนี้คือ สร้าง UI สำหรับกรอกข้อมูลเพื่อนำไป insert เข้าใน database โดยจะมี header คือ create movie มีช่องกรอกข้อมูลหนึ่งอยู่ 4 ช่อง คือ title, genre, director, release year และ มีปุ่ม create ในการเพิ่มข้อมูล

ต่อไป

สร้างไฟล์ใหม่ชื่อ SearchPage.js ใน component

```
import React, { useState } from 'react';
import { getMovieSearch } from '../services/MovieService';

const SearchPage = ({ setMovies, setNumberOfMovies }) => {
  const [searchText, setSearchText] = useState('');
  const [errorMessage, setErrorMessage] = useState('');

  const handleSearch = async (e) => {
    e.preventDefault(); // ป้องกัน default behavior ของ form submission

    // Check if the search text is empty
    if (!searchText.trim()) {
      setErrorMessage('Please enter a movie title for search.');
      return;
    }

    try {
      const response = await getMovieSearch(searchText);
      if (response.statusCode === 200) {
        setMovies(response.data);
        setNumberOfMovies(response.data.length);
        setErrorMessage('');
      } else {
        setMovies([]);
        setNumberOfMovies(0);
        setErrorMessage(response.message || 'No movie found');
      }
    } catch (error) {
      console.error('Error searching movie:', error);
      setMovies([]);
      setNumberOfMovies(0);
      setErrorMessage('Error searching movie');
    }
  }
}
```

นาย ศุภวิชญ์ แก่นนาค 66543210031-1

```
        }

    };

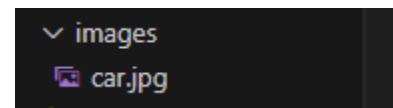
    return (
      <div>
        <h2 style={{ backgroundColor: 'lightgray' }}>Movie Search</h2>
        <form className="d-flex">
          <input
            type="text"
            value={searchText}
            onChange={(e) => setSearchText(e.target.value)}
            placeholder="Enter movie title"
            className="form-control mr-2" // Bootstrap class for
margin-right
          />
          <button
            onClick={handleSearch}
            className="btn btn-danger"
            style={{ backgroundColor: 'green' }}
          >
            Search
          </button>
        </form>
        {errorMessage && <p style={{ color: 'red' }}>{errorMessage}</p>}
      </div>
    );
  };

export default SearchPage;
```

การทำงานของโค้ดส่วนนี้คือ สร้างหน้า UI ใช้ค้นหาชื่อหนังโดยมี title เป็น keyword ในการใช้ค้นหาซึ่งจะเชื่อมกับตาราง movies และเอาเฉพาะ keyword ที่พิมพ์เข้าไปในช่องแสดงข้อมูลมาถ้ามี หนังอื่นมี keyword เหมือนกัน เช่นมี ตัว a เมื่อันกันก็จะแสดงมาทั้งสองอัน หน้านี้มีส่วนประกอบหลักคือ h2 เป็นหัวข้อ Movie search มีช่องค้นหาที่มี placeholder บอกว่าต้องใส่ title เพื่อค้นหา มีปุ่ม Search เพื่อใช้กดค้นหา

ຕ່ອໄປ

ສ້າງ folder images ຂັ້ນໃນ folder public ເພື່ອໃຫ້ໄສຮູບ



ຕ່ອໄປ

ແກ່ໃນ MovieService.js

```
export async function getAllMovies() {  
  
    try{  
        //const response = await fetch('/api/users');  
        const response = await  
fetch('http://localhost:3001/api/movie/all');  
        //const response = await fetch('/api/movie/all');  
        return await response.json();  
    }catch(error) {  
        return [];  
    }  
}  
  
export async function createMovie(data) {  
    const response = await fetch('http://localhost:4000/api/movie/insert'  
{  
    method: 'POST',  
    headers: {'Content-Type': 'application/json'},  
    body: JSON.stringify({...data})  
})  
    return await response.json();  
  
export async function getMovieSearch(search_text) {  
    try {  
        const response = await  
fetch(`http://localhost:4000/api/movie/search?search_text=${search_text}`)  
  
        return await response.json();  
    } catch (error) {  
        return [];  
    }  
}
```

นาย ศุภวิชญ์ แก่นนาค 66543210031-1

โค้ดหน้านี้ที่ได้ทำการแก้ไขมี

- 1.ส่วนของ export async function createMovie "ได้ทำการแก้ไข response ให้เป็น await fetch("http://localhost:4000/api/movie/insert",
และ แก้ไข body ให้เป็น JSON.stringify({...data}) ใช้ในการส่งข้อมูลไปยัง database
- 2.ส่วนของ export async function getMovieSearch(search_text)
ทำการเพิ่มมาใหม่ทั้งหมดเพื่อให้ function การค้นหาหนังใช้งานได้

ต่อไป

แก้ไขไฟล์ Header.js

```
import React from 'react'

export const Header = () => {

    const headerStyle = {

        width: '100%',
        padding: '2%',
        backgroundColor: "blue",
        color: 'white',
        textAlign: 'center'
    }

    return (
        <div style={headerStyle}>
            <h1>MOVIE EDIT</h1>
        </div>
    )
}
```

ส่วนที่แก้ไขคือในส่วน backgroundColor เป็นสีน้ำเงิน และส่วนของ header เป็น movie edit

ต่อไป

แก้ไขไฟล์ App.js

```
import React, { useState, useEffect } from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';
import './App.css';
import { Header } from './components/Header'
import { DisplayBoard } from './components/DisplayBoard'
//-----
import { Movies } from './components/Movies'
import CreateMovie from './components/CreateMovie'
```

นาย គុរីធម្មោ កែនណាគ 66543210031-1

```
import { getAllMovies, createMovie } from './services/MovieService'

//เพิ่มใหม่
import SearchPage from './components/SearchPage';

function App() {

  const [user, setUser] = useState({})
  const [users, setUsers] = useState([])
  const [numberOfUsers, setNumberOfUsers] = useState(0)

  //-----
  const [movie, setMovie] = useState({})
  const [movies, setMovies] = useState([])
  const [numberOfMovies, setNumberOfMovies] = useState(0)
  const [searchResults, setSearchResults] = useState([])

  const movieCreate = (e) => {
    createMovie(movie)
      .then(response => {
        console.log(response);
        setNumberOfMovies(numberOfMovies+1)
      });
  }

  const fetchAllMovies = () => {
    getAllMovies()
      .then(movies => {
        console.log(movies)
        setMovies(movies);
        setNumberOfMovies(movies.length)
        setSearchResults([]); //เพิ่ม
      });
  }

  useEffect(() => {
    getAllMovies()
      .then(movies => {
```

```
        console.log(movies)
        setMovies(movies);
        setNumberOfMovies(movies.length)
    });
}, [])

const onChangeForm = (e) => {
    if (e.target.name === 'title') {
        movie.title = e.target.value;
    } else if (e.target.name === 'genre') {
        movie.genre = e.target.value;
    } else if (e.target.name === 'director') {
        movie.director = e.target.value;
    } else if (e.target.name === 'release_year') {
        movie.release_year = e.target.value;
    }
    setMovie(movie)
}

return (
    <div className="App" style={{backgroundImage:
        'url("/images/car.jpg")', backgroundSize: 'cover', minHeight: '100vh'}}>
        <Header></Header>
        <div className="container mrgnbtm">
            <div className="row">
                <div className="col-md-8">
                    <CreateMovie
                        Movies={Movies}
                        onChangeForm={onChangeForm}
                        createMovie={movieCreate}>
                    </CreateMovie>
                    <div className="col-md-8">
                        <SearchPage
                            setMovies={setMovies}
                            setNumberOfMovies={setNumberOfMovies}
                            setSearchResults={setSearchResults}>
                        </SearchPage>
                    </div>
                </div>
            </div>
        </div>
    </div>
)
```

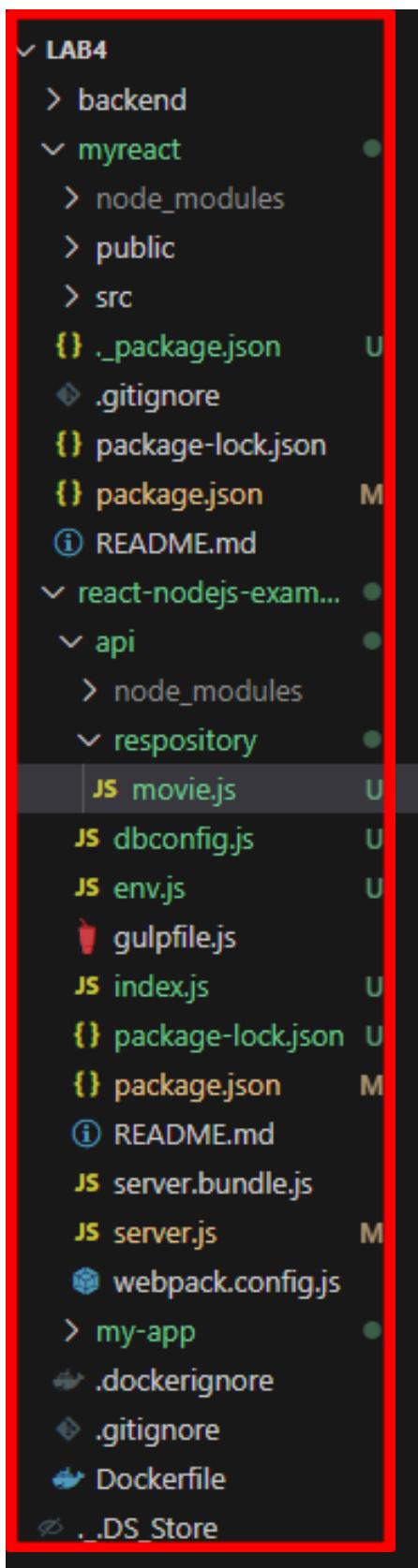
```
<div className="col-md-4">
  <DisplayBoard
    numberOfMovies={numberOfMovies}
    getAllMovies={fetchAllMovies}
  >
  </DisplayBoard>
</div>
</div>
<div className="row mrgnbtm">
  <Movies movies={searchResults.length > 0 ? searchResults
movies}></Movies>
</div>
</div>
) ;
}

export default App;
```

การแก้ไขของโค้ดหน้านี้

- 1.เพิ่ม `import SearchPage from './components/SearchPage';` เพื่อที่จะสามารถเรียกใช้ component SearchPage ได้
- 2.เพิ่ม `const [searchResults, setSearchResults] = useState([]);` เข้าไปในฟังก์ชัน App
- 3.เพิ่ม `setSearchResults([]);` เข้าไป const fetchAllMovie
- 4.เพิ่ม `style={{backgroundImage: 'url("/images/car.jpg")', backgroundSize: 'cover', minHeight: '100vh'}}>` เข้าไปใน div ของ classname App(ข้างใน bracket div)เพื่อใส่ รูปใน background ของหน้าเว็บ
- 5.เพิ่ม div และ เพิ่ม function เรียกใช้ `<SearchPage setMovies={setMovies}
setNumberOfMovies={setNumberOfMovies} setSearchResults={setSearchResults}/>`
- 6.เพิ่ม `<div className="row mrgnbtm"> <Movies movies={searchResults.length > 0 ? searchResults : movies}></Movies> </div>`

5. ส่วนโคดที่ไม่ต้องแก้ไข (API Back-end)



```
LAB4
  backend
  myreact
    node_modules
    public
    src
    .package.json
    .gitignore
    package-lock.json
    package.json
    README.md
  react-nodejs-exam...
    api
      node_modules
      repository
        movie.js
        dbconfig.js
        env.js
        gulpfile.js
        index.js
        package-lock.json
        package.json
        README.md
        server.bundle.js
        server.js
        webpack.config.js
    my-app
      .dockerignore
      .gitignore
      Dockerfile
      .DS_Store
```



ไม่ต้องแก้ไข

นาย ศุภวิชญ์ แก่นนาค 66543210031-1

Phpmyadmin ใช้ตรวจสอบการ insert ข้อมูล(xampp)

The screenshot shows the phpMyAdmin interface for the 'moviedb' database. The 'movies' table is selected. The table structure includes columns: title, genre, director, and release_year. The data grid displays 14 rows of movie information. At the bottom, there are navigation and search/filter options.

	title	genre	director	release_year
1	1	1	1	1
12	12	21		312
123	123	123		123
2	2	2		2
asd	zxc	df		0
cvf	ds	g		4556
ghdf	adx	sc		0
ghfghd	gbvcds	xzcasf		2000
jgh	tj	aghj		2
Joker	psychological thriller	Todd Phillips		2019
s	t	a		0
spiderman	sci-fi	Sam Raimi		2010
test	test	test		0
testtest	test	test		0