

# ENGSE203 - Midterm

แก้ไขตัวแปรที่ App.js ในส่วน Frontend ให้ตรงตาม Database

```
86  const onChangeForm = (e) => {
87    if (e.target.name === "title") {
88      movie.title = e.target.value;
89    } else if (e.target.name === "genre") {
90      movie.genre = e.target.value;
91    } else if (e.target.name === "director") {
92      movie.director = e.target.value;
93    } else if (e.target.name === "release_year") {
94      movie.release_year = e.target.value;
95    }
96  }
```

	title	genre	director	release_year
▶	Joker	psychological thriller	Todd Phillips	2019
*	NULL	NULL	NULL	NULL

## MovieService.js

แก้ url จาก /api/movie เป็น url localhost port 4000 ตามด้วย  
หน้าต่างๆ แก้ไข บรรทัดที่ 13 ในวงเล็บจาก movie , data เป็น ...  
data เพราะต้องการส่งแค่ data ที่เราสร้างใหม่จาก frontend ไปยัง  
เซิร์ฟเวอร์ ผ่าน api

```
1  export async function getAllMovies() {
2    try {
3      const response = await fetch("http://localhost:4000/api/movie/all");
4      return await response.json();
5    } catch (error) {
6      return [];
7    }
8  }
9  export async function createMovie(data) {
10   const response = await fetch(`http://localhost:4000/api/movie/insert`, {
11     method: "POST",
12     headers: { "Content-Type": "application/json" },
13     body: JSON.stringify({ ...data }),
14   });
15   return await response.json();
16 }
```

## Movies.js

สร้างตัวแปร searchText เพื่อเก็บข้อความที่ใช้ค้นหาหนัง

สร้าง searchResults เพื่อเก็บผลลัพธ์ที่ได้จากค้นหาหนัง

สร้าง isSearching เพื่อเป็นสถานะบ่งชี้ว่า functions กำลังทำงานอยู่หรือไม่

Function handleSearch

- ถ้ากดปุ่ม search ฟังก์ชันนี้จะ fetch ข้อมูลจาก api โดยใช้ข้อมูลจากผู้ใช้ใส่เข้าไปใน input เช่น Joker , Avatar
- ถ้าหาเจอสถานะจะเป็น true ถ้าไม่ใช่จะเป็น false (ไม่มีหนังหรือหาไม่เจอ)

```
6 export const Movies = ({ movies }) => {
7   const [searchText, setSearchText] = useState('');
8   const [searchResults, setSearchResults] = useState([]);
9   const [isSearching, setIsSearching] = useState(false);
10
11   Codeium: Refactor | Explain | Generate JSDoc | X
12   const handleSearch = async () => {
13     try {
14       const response = await fetch(`http://localhost:3001/api/movie/search?search_text=${searchText}`);
15       const data = await response.json();
16
17       if (data.returnCode === 1) {
18         setSearchResults(data.data);
19         setIsSearching(true);
20       } else {
21         setSearchResults([]);
22         setIsSearching(false);
23       }
24     } catch (error) {
25       console.error('Error searching for movies:', error);
26     }
27   };
28 }
```

เพิ่ม input และ ปุ่ม search ไว้บนหัวตาราง

- ใน input ใส่ onChange เพื่อเรียกใช้ฟังก์ชัน setSearchText จะได้ค่าของ searchText ให้เป็นค่าที่เราใส่ไปใน input
- กำหนดให้เรียกใช้ฟังก์ชัน handleSearch เมื่อกด search

```
44   searchResults.map((movie, index) => MovieRow(movie, index)) .
45   movies.map((movie, index) => MovieRow(movie, index));
46
47   return (
48     <div className="container">
49       <div style={{ display: "flex", justifyContent: "flex-start", alignItems: "center" }}>
50         <div>
51           <h2>Movies</h2>
52           <div style={{ display: "flex", alignItems: "center" }}>
53             <input
54               type="text"
55               className="Search_input"
56               style={{
57                 padding: "10px",
58                 fontSize: "16px",
59                 border: "1px solid #ccc",
60                 borderRadius: "5px",
61                 width: "300px",
62                 marginRight: "10px",
63               }}
64               placeholder="Enter The Title Of Movie"
65               value={searchText}
66               onChange={(e) => setSearchText(e.target.value)}
67             />
68             <button
69               className="btn btn-warning"
70               style={{ padding: "10px", borderRadius: "5px" }}
71               onClick={handleSearch}
72             >
73               Search
74             </button>
75           </div>
76         </div>
77       </div>
78       <hr />
79
80       <table className="table table-dark table-striped">
81         <thead className="table-dark">
82           <tr>
```

## โค้ดทั้งหมดของ Movies.js

```
import React, { useState } from "react";
import "../App.css";

export const Movies = ({ movies }) => {
  const [searchText, setSearchText] = useState('');
  const [searchResults, setSearchResults] = useState([]);
  const [isSearching, setIsSearching] = useState(false);

  const handleSearch = async () => {
    try {
      const response = await
fetch(`http://localhost:3001/api/movie/search?search_text=${searchText}`);
      const data = await response.json();
```



```

        placeholder="Enter The Title Of Movie"
        value={searchText}
        onChange={(e) => setSearchText(e.target.value)}
    />
    <button
        className="btn btn-warning"
        style={{ padding: "10px", borderRadius: "5px" }}
        onClick={handleSearch}
    >
        Search
    </button>
</div>
</div>
</div>
<hr />

<table className="table table-dark table-striped">
    <thead className="table-dark">
        <tr>
            <th>Movie Id</th>
            <th>Title</th>
            <th>Genre</th>
            <th>Director</th>
            <th>Release</th>
        </tr>
    </thead>
    <tbody>
        {movieTable}
        {isSearching && searchResults.length === 0 && <tr><td colspan="5">No
movies found</td></tr>}
    </tbody>
</table>
</div>
);
};

```

## Backend หน้า index.js

ในส่วนของ backend หน้า index.js ในส่วนของ search

ใช้ url ตรงกับหน้า Movies.js กำหนด cors มี origin เป็น port 3001 เพราะตอนทำไม่สามารถดึงข้อมูลหนัง จาก port 4000 มาได้ ติด cors error ถ้าเปิดแล้ว ทำให้ frontend ที่ทำงานอยู่ใน port 3001 สามารถเรียก api ที่อยู่บน port 4000 ได้

```
74  server.route({
75    method: 'GET',
76    path: '/api/movie/search',
77    config: {
78      cors: {
79        origin: ['http://localhost:3001'],
80        additionalHeaders: ['cache-control', 'x-requested-width']
81      }
82    },
83    handler: async function (request, reply) {
84      var param = request.query;
85      const search_text = param.search_text;
86      //const title = param.title;
87
88      try {
89
90        const responsedata = await Movies.MovieRepo.getMovieSearch(search_text)
91        if (responsedata.error) {
92          return responsedata.errorMessage;
93        } else {
94          return responsedata;
95        }
96      } catch (err) {
97        server.log(["error", "home"], err);
98        return err;
99      }
100    }
101  });
102
```

# ทดสอบการทำงาน



Create Movies

Title Movie

Genre

title

genre

Director

director

Year

release\_year

Create

Movies

Enter The Title Of Movie

Search

Movie Id	Title	Genre	Director	Release
1	Avatar	Fantasy	James Camelon	2007
2	Joker	psychological thriller	Todd Phillips	2019

Movies Created

2

Get all Movies

## Created Movies



Create Movies

Title Movie

Genre

Eternal

hero

Director

John snow

Year

2024

Create

Movies

Enter The Title Of Movie

Search

Movie Id	Title	Genre	Director	Release
1	Avatar	Fantasy	James Camelon	2007
2	Eternal	hero	John snow	2024
3	Joker	psychological thriller	Todd Phillips	2019

Movies Created

3

Get all Movies

## Search Movie



Create Movies

Title Movie

Genre

Eternal

hero

Director

John snow

Year

2024

Create

Movies

Eternal

Search

Movie Id	Title	Genre	Director	Release
1	Eternal	hero	John snow	2024

Movies Created

3

Get all Movies

Before :

NUNTHI PLUS+

Create Movies

Title Movie

title

genre

Director

director

Year

release\_year

Create

Movies Created

2

Get all Movies

Movies

Enter The Title Of Movie

Search

Movie Id	Title	Genre	Director	Release
1	Avatar	Fantasy	James Camelon	2007
2	Joker	psychological thriller	Todd Phillips	2019

After :

NUNTHI PLUS+

Create Movies

Title

Title

Genre

Genre

Genre

Director

Director

Release

Release

Create

Movies Created

2

Get all Movies

Movies

Enter The Title Of Movie

Search

Movie Id	Title	Genre	Director	Release
1	Avatar	Fantasy	James Camelon	2019
2	Joker	psychological thriller	Todd Phillips	2019

นาย นันทิพัฒน์ สมบัติวงศ์ 66543210018-8