

Midterm

Frontend

เริ่มตั้งแต่ปรับปรุง App.js (frontend)

```
41 const onChangeForm = (e) => {  
42   if (e.target.name === "movies_title") {  
43     movie.title = e.target.value;  
44   } else if (e.target.name === "movies_genre") {  
45     movie.genre = e.target.value;  
46   } else if (e.target.name === "movies_director") {  
47     movie.director = e.target.value;  
48   } else if (e.target.name === "movies_release") {  
49     movie.release_year = e.target.value;  
50   }
```

แก้ไขตัวแปร title , genre , director , release_year ให้ตรงกับ column ใน db ดังรูป

	title	genre	director	release_year	
	Batman	Action , Detective	Nolan	2010	
	Dunkirk	Based on story	Nolan	2019	
	Forrest Gump	Comedy	Todd Phillips	2019	
	Joker	psychological thriller	Todd Phillips	2019	
	NULL	NULL	NULL	NULL	

MovieService.js

```
34 export async function getAllMovies() {
35
36   try{
37     //const response = await fetch('/api/users');
38     const response = await fetch('http://localhost:4000/api/movie/all');
39     //const response = await fetch('/api/movie/all');
40     return await response.json();
41   }catch(error) {
42     return [];
43   }
44
45 }
46
47 //---- ยังไม่เสร็จ ----
48 export async function createMovie(data) {
49   const response = await fetch('http://localhost:4000/api/movie/insert', {
50     method: 'POST',
51     headers: {'Content-Type': 'application/json'},
52     body: JSON.stringify({...data})
53   })
54   return await response.json();
55 }
56
```

แก้ url จาก /api/movie เป็น url localhost port 4000 ตามด้วยหน้าต่างๆ
แก้ไข บรรทัดที่ 52 ในวงเล็บจาก movie , data เป็น ... data เพราะผมต้องการ
ส่งแค่ data ที่เราสร้างใหม่จาก frontend ไปยังเซิร์ฟเวอร์ ผ่าน api

Movies.js

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

สร้างตัวแปร searchText มาเก็บข้อความที่ใช้ค้นหาหนัง
สร้าง searchResults เก็บผลลัพธ์ที่ได้จากการค้นหา

สร้าง isSearching เพื่อเป็นสถานะบูลีน ว่า functions กำลังทำงานอยู่หรือไม่

Function handleSearch

- ถ้ากดปุ่ม search ฟังก์ชันนี้จะ fetch ข้อมูลจาก api โดยใช้ข้อมูลจากผู้ใช้ ใส่เข้าไปใน input เช่น batman
- ถ้าหาเจอ สถานะจะเป็น true ถ้าไม่ใช่จะเป็น false (ไม่มีหนังหรือหาไม่เจอ)

npm install cors

```
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-bordered table-hover">
81         <thead className="table-dark">
82           <tr>
83             <th>Movie Id</th>
84             <th>Title</th>
85             <th>Genre</th>
86             <th>Director</th>
87             <th>Release</th>
88           </tr>
89         </thead>
90         <tbody>
91           {movieTable}
92           {isSearching && searchResults.length === 0 && <tr><td colSpan="5">No movies found</td></tr>}
93         </tbody>
94       </table>
95     </div>
96   );
97 );
```

เพิ่ม input และ ปุ่ม search ไว้บนหัวตาราง

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

ทดสอบการทำงาน

localhost:3001/api/movie/all

React With NodeJS

Create Movie

Title

Genre

Director

Release_year

Create

Movies

Search

Movie Id	Title	Genre	Director	Release
1	Batman	Action , Detective	Nolan	2010
2	Dunkirk	Based on story	Nolan	2019
3	Forrest Gump	Comedy	Todd Phillips	2019
4	Joker	psychological thriller	Todd Phillips	2019

Movies Created test

4

Get all Movies

localhost:3001/api/movie/all

React With NodeJS

Create Movie

Title

Genre

Director

Release_year

Create

Movies

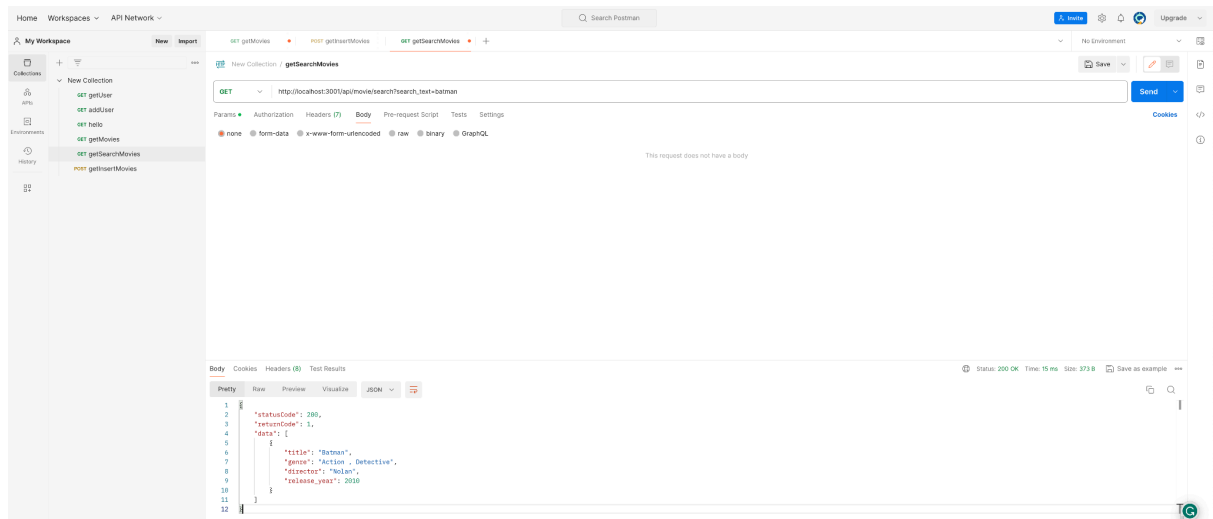
Search

Movie Id	Title	Genre	Director	Release
1	Batman	Action , Detective	Nolan	2010

Movies Created test

4

Get all Movies



ทดสอบใน api
url เามาจากหน้า Movies.js ในฟังก์ชัน handleSwitch

```
11  < const handleSearch = async () => {  
12  <   try {  
13  <     const response = await fetch(`http://localhost:3001/api/movie/search?search_text=${searchText}`);  
14  <     const data = await response.json();  
15  <  
16  <     if (data.returnCode === 1) {  
17  <       setSearchResults(data.data);  
18  <       setIsSearching(true);  
19  <     } else {  
20  <       setSearchResults([]);  
21  <       setIsSearching(false);  
22  <     }  
23  <   } catch (error) {  
24  <     console.error('Error searching for movies:', error);  
25  <   }  
26  < };
```

ในส่วนของ backend หน้า index.js ในส่วนของ search

```
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
87         try {
88           const responsedata = await Movies.MovieRepo.getMovieSearch(search_text);
89           if (responsedata.error) {
90             return responsedata.errMessage;
91           } else {
92             return responsedata;
93           }
94         } catch (err) {
95           server.log(["error", "home"], err);
96           return err;
97         }
98       }
99     });
```

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 ที่รันอยู่บน port 4000 ได้ครับ

npm install cors

Before :

React With NodeJS

Create Movie

Title

Movies Title

Genre

Genre

Director

Director

Release_year

Release

Create

Movies Created test

4

Get all Movies

Movies

batman

Search

Movie Id	Title	Genre	Director	Release
1	Batman	Action , Detective	Nolan	2010

After :

REACT WITH NODEJS

Create Movies

Title

Movie Title

Genre

Movie Genre

Director

Director Name

Release

Release Year

Create

Movies Created

2

Get all Movies

Movies List

Enter The Title Of Movie

Search

Movie Id	Title	Genre	Director	Release
1	Batman	Detective , Action	Chirstopher Nolan	2010
2	Joker	psychological thriller	Todd Phillips	2019

นายพลพัต ทรวงหิรัญ 66543210020-4