

Lab4 Reactjs (Front-end) - งานเพิ่ม Movies

นำ Lab 3 มาใช้งานเป็น back-end

The screenshot shows the VS Code interface with the following details:

- Project Structure:** The left sidebar shows a tree view of the project. It includes a 'backend' folder containing 'index.js', 'package.json', and 'path'. A 'LAB4' folder contains 'backend' (with 'node_modules', '.gitignore', 'dbconfig.js', 'env.js'), 'frontend' (with 'build', 'node_modules', 'public', 'src' containing 'package-lock.json', 'package.json', and 'users.js'), and 'index.js'. The 'index.js' file under 'frontend' is currently selected.
- Code Editor:** The main editor area displays the content of 'index.js'. The code is as follows:

```
1 const hapi = require('@hapi/hapi');
2 var users = require('./users');
3 const env = require('./env.js');
4 const Movies = require('./repository/movie');

5
6 const express = require('express');
7 const app = express();

8
9 //-----
10 const web_port = 3010;
11 const api_port = 3001;

12
13
14 //----- express -----
15 app.get('/', (req, res) => {
16   res.send('<h1> Hello World main_server </h1>');
17 }
18 )
19
20 app.get('/users', function (req, res) {
21   res.json(users.findAll());
22 });
23
24 app.get('/user/:id', function (req, res) {
25   var id = req.params.id;
26   res.json(users.findById(id));
27 });
28
29 app.listen(web_port, () => {
```

The code uses Hapi for the backend API and Express for the front-end server. It defines routes for '/users' and '/user/:id'.

ทดสอบการทำงานของ API

The screenshot shows the Postman interface. On the left, the 'My Workspace' sidebar lists collections, environments, and history. In the center, a request is being made to 'SystemD / test show all'. The method is 'GET' and the URL is 'http://localhost:3001/api/movie/all'. The 'Params' tab shows a query parameter 'search_text' with the value 'Fight'. The 'Body' tab shows the response as a JSON array of movies:

```

1 [
2   {
3     "title": "Avengers: Endgame",
4     "genre": "superhero",
5     "director": "Anthony Russo, Joe Russo",
6     "release_year": 2019
7   },
8   {
9     "title": "Fight Club",
10    "genre": "drama",
11    "director": "David Fincher",
12    "release_year": 1999
13  },
14  {
15    "title": "Forrest Gump",
16    "genre": "drama",
17    "director": "Robert Zemeckis",
18    "release_year": 1994
19  }

```

The status bar at the bottom indicates a 200 OK response with a time of 23 ms and a size of 2.28 KB.

ແລະນຳ UI ຂອງ lab 1 ມາໃຊ້ຈຳນັກເປົ້າ front-end

The screenshot shows the VS Code interface. The left sidebar shows a project structure with 'OPEN EDITORS' containing 'index.js backend' and 'index.js frontend/src'. The main area shows the 'index.js' file from the 'frontend/src' editor:

```

1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
6 import { AppError } from './components/AppError';

7
8 ReactDOM.render(
9   <React.StrictMode>
10    <AppError>
11      <App />
12    </AppError>
13  </React.StrictMode>,
14  document.getElementById('root')
15 );
16
17 // If you want your app to work offline and load faster, you can change
18 // unregister() to register() below. Note this comes with some pitfalls.
19 // Learn more about service workers: https://bit.ly/CRA-PWA
20 serviceWorker.unregister();
21

```

The status bar at the bottom shows various developer tools like Live Share, Git Graph, and Prettier.

ແກ້ໄຂ package.json ແກ້ປ່ຽນທາງການທຳມານເຮືອງ version ຂອງ nodejs ທີ່ການ start ແລະ build

```
"scripts": {  
  "start": "export PORT=4000 && react-scripts --openssl-legacy-provider start",  
  "build": "react-scripts --openssl-legacy-provider build",  
  ....  
},
```

เพิ่มไฟล์ frontend/src/components/Movies.js

```
import React from 'react'  
  
export const Movies = ({movies}) => {  
  
  console.log('movies length:::', movies.length)  
  if (movies.length === 0) return null  
  
  const MovieRow = (movie, index) => {  
  
    return(  
      <tr key = {index} className={index%2 === 0?'odd':'even'}>  
        <td>{index + 1}</td>  
        <td>{movie.title}</td>  
        <td>{movie.genre}</td>  
        <td>{movie.director}</td>  
        <td>{movie.release_year}</td>  
      </tr>  
    )  
  }  
  
  const movieTable = movies.map((movie, index) => MovieRow(movie, index))  
  
  return(  
    <div className="container">  
      <h2>Movies</h2>  
      <table className="table table-bordered">  
        <thead>  
          <tr>  
            <th>Movie Id</th>  
            <th>Title</th>  
            <th>Genre</th>  
            <th>Director</th>  
            <th>Release</th>  
          </tr>  
        </thead>  
        <tbody>  
          {movieTable}  
        </tbody>  
      </table>  
    </div>  
  )  
}
```

เพิ่มไฟล์ frontend/src/components/CreateMovie.js

```

import React from 'react'

const CreateMovie = ({onChangeForm, createMovie }) => {

    return(
        <div className="container">
            <div className="row">
                <div className="col-md-7 mrgnbtm">
                    <h2>Create Movie</h2>
                    <form>
                        <div className="row">
                            <div className="form-group col-md-6">
                                <label htmlFor="exampleInputEmail1">Title</label>
                                <input type="text" onChange={(e) => onChangeForm(e)} 
className="form-control" name="title" id="title" placeholder="Title" />
                            </div>
                            <div className="form-group col-md-6">
                                <label htmlFor="exampleInputPassword1">Genre</label>
                                <input type="text" onChange={(e) => onChangeForm(e)} 
className="form-control" name="genre" id="genre" placeholder="Genre" />
                            </div>
                        </div>
                        <div className="row">
                            <div className="form-group col-md-6">
                                <label htmlFor="exampleInputEmail1">Director</label>
                                <input type="text" onChange={(e) => onChangeForm(e)} 
className="form-control" name="director" id="director" placeholder="Director" />
                            </div>
                            <div className="form-group col-md-6">
                                <label htmlFor="exampleInputEmail1">Release</label>
                                <input type="text" onChange={(e) => onChangeForm(e)} 
className="form-control" name="release_year" id="release_year" placeholder="Release" />
                            </div>
                        </div>
                        <button type="button" onClick= {(e) => createMovie()} 
className="btn btn-danger">Create</button>
                    </form>
                </div>
            </div>
        )
    }

    export default CreateMovie
}

```

เพิ่มไฟล์ frontend/src/components/SearchMovie.js

```

import React from 'react'

const SearchMovie = ({onChangeForm, searchMovie }) => {

```

```

return(
  <div className="container">
    <div className="row">
      <div className="col-md-7 mrgnbtm">
        <h2>Search Movie</h2>
        <form>
          <div className="row">
            <div className="form-group col-md-6">
              <input type="text" onChange={(e) => onChangeForm(e)}>
            </div>
            <button type="button" onClick= {(e) => searchMovie()}>
              Search
            </button>
          </div>
        </form>
      </div>
    </div>
  )
}

export default SearchMovie

```

แก้ไขไฟล์ frontend/src/components/DisplayBoard.js

```

import React from 'react'

export const DisplayBoard = ({numberOfMovies, getAllMovies}) => {

  const headerStyle = {

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

  return(
    <div style={{backgroundColor: 'green'}} className="display-board">
      <h4 style={{color: 'white'}}>Movies Created</h4>
      <div className="number">
        {numberOfMovies}
      </div>
      <div className="btn">
        <button type="button" onClick={(e) => getAllMovies()} className="btn btn-warning">Get all Movies</button>
      </div>
    </div>
  )
}

```

เพิ่มไฟล์ frontend/src/services/MovieService.js

```
export async function searchMovie(search_data) {  
  try {  
    console.log('search_data: ' + search_data);  
  
    const response = await fetch('http://localhost:3001/api/movie/search?  
search_text=' + search_data);  
  
    return await response.json(); //***  
  
    // console.log('Search Movie response: ' + await response);  
  } catch (error) {  
    return [];  
  }  
}  
  
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 [];  
  }  
}  
  
//---- ยังไม่เสร็จ ----  
export async function createMovie(data) {  
  const response = await fetch(`/api/user`, {  
    method: 'POST',  
    headers: {'Content-Type': 'application/json'},  
    body: JSON.stringify({user: data})  
  })  
  return await response.json();  
}
```

ปรับปรุง 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'
import SearchMovie from './components/SearchMovie'
import { getAllMovies, createMovie, searchMovie } from './services/MovieService'

function App() {

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

  const movieCreate = (e) => {

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

  fetchAllMovies(); /* */
}

const movieSearch = (e) => {

  console.log(search_data.search_text);

  searchMovie("jok")
    .then(movies => {
      console.log(movies.data);
      setMovies(movies.data);
      setNumberOfMovies(movies.data.length);
    });
}

const fetchAllMovies = () => {
  getAllMovies()
    .then(movies => {
      console.log(movies);
      setMovies(movies);
      setNumberOfMovies(movies.length);
    });
}

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)
}

const onChangeForm2 = (e) => {
  if (e.target.name === 'search_text') {
    search_data.search_text = e.target.value;
  }
  setSearch_data(search_data)
}

return (
  <div className="App">
    <Header></Header>
    <div className="container mrgnbm">
      <div className="row">
        <div className="col-md-8">
          <CreateMovie
            // user={user}
            movie={movie}
            onChangeForm={onChangeForm}
            createMovie={movieCreate}
          >
          </CreateMovie>
        </div>
        <div className="col-md-4">
          <DisplayBoard
            numberOfMovies={numberOfMovies}
            getAllMovies={fetchAllMovies}
          >
          </DisplayBoard>
        </div>
      </div>
    <div className="row mrgnbm">
      <SearchMovie
        // user={user}
        movie={movie}
      >
      </SearchMovie>
    </div>
  </div>
)

```

```

        onChangeForm={onChangeForm}
        searchMovie={movieSearch}
      >
    </SearchMovie>

    {/* <SearchBoard
      // user={user}
      search_data={search_data}
      onChangeForm2={onChangeForm2}
      searchMovies={searchMovie}
    >
    </SearchBoard> */}

  </div>
  <div className="row mrgnbtm">
    <Movies movies={movies}></Movies>
  </div>

</div>
);
}

export default App;

```

รันคำสั่ง

```
npm start
```

React With NodeJS

Create Movie

Title Director Genre Release

Search Movie

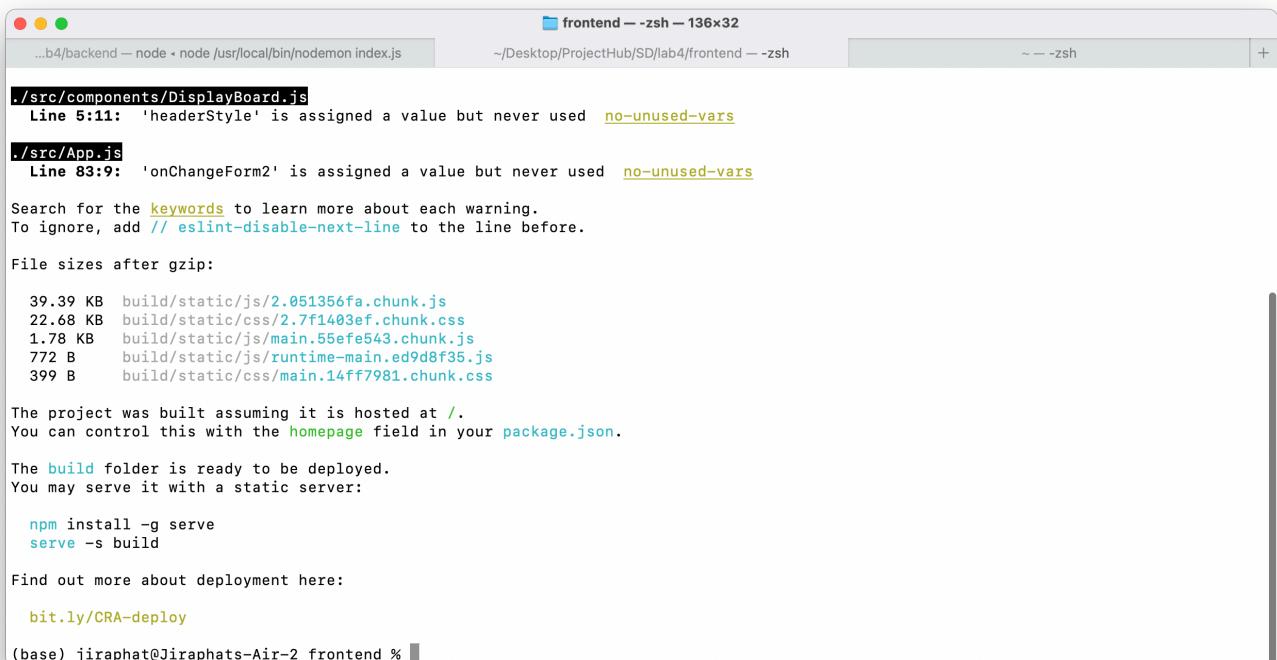
Director

Movies

Movie Id	Title	Genre	Director	Release
1	Avengers: Endgame	superhero	Anthony Russo, Joe Russo	2019
2	Fight Club	drama	David Fincher	1999
3	Forrest Gump	drama	Robert Zemeckis	1994

รันคำสั่ง

```
npm run build
```



```
./src/components/DisplayBoard.js
Line 5:11:  'headerStyle' is assigned a value but never used  no-unused-vars
./src/App.js
Line 83:9:  'onChangeForm2' is assigned a value but never used  no-unused-vars

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

File sizes after gzip:

39.39 KB  build/static/js/2.051356fa.chunk.js
22.68 KB  build/static/css/2.7f1403ef.chunk.css
1.78 KB   build/static/js/main.55efe543.chunk.js
772 B     build/static/js/runtime-main.ed9d8f35.js
399 B    build/static/css/main.14ff7981.chunk.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:
  bit.ly/CRA-deploy

(base) jiraphat@Jiraphats-Air-2 frontend %
```

เพิ่มเติม

สร้าง function ให้ครบ search data, insert data, delete data

แก้ไขการค้นหา **search data**

แก้ไขให้ใช้ `search_data.search_text []`

```
const movieSearch = (e) => {

  console.log(search_data.search_text);

  // searchMovie("jok")
  searchMovie(search_data.search_text)
    .then(movies => {
      console.log(movies.data);
      setMovies(movies.data);
      setNumberOfMovies(movies.data.length);
    });
}
```

แก้ไขให้ใช้ onChangeForm2 และ onChangeForm

```
...
function App() {
  //-----
```

```

    ...
    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)
    }

    const onChangeForm2 = (e) => {
      // console.log("TEST ", e.target.name)
      if (e.target.name === 'search_text') {
        search_data.search_text = e.target.value;
      }
      setSearch_data(search_data)
    }

    return (
      ...
      <SearchMovie
        // user={user}
        movie={movie}
        onChangeForm={onChangeForm2}
        searchMovie={movieSearch}
      >
      </SearchMovie>
      ...
    )
  }

  export default App;

```

แก้ไขการตั้งค่าชื่อ name จาก director เป็น search_text

```

import React from 'react'
const SearchMovie = ({onChangeForm, searchMovie }) => {
  return(
    <div className="container">
      <div className="row">
        <div className="col-md-7 mrgnbtm">
          <h2>Search Movie</h2>
          <form>
            <div className="row">
              <div className="form-group col-md-6">

```

```

        <input type="text" onChange={(e) => onChangeForm(e)}
      className="form-control" name="search_text" id="search_text" placeholder="movie name"
    />
      </div>

      </div>
      <button type="button" onClick= {(e) => searchMovie("test")}
    className="btn btn-danger">Search</button>
      </form>
      </div>
    </div>

  </div>
)
}

export default SearchMovie

```

ผลการทดสอบ

The screenshot shows a web browser window with several tabs open at the top. The active tab is titled "Create Movie". The page contains two forms: one for creating a new movie and one for searching existing movies.

Create Movie Form:

- Fields: Title (input), Director (input), Genre (input), Release (input).
- Buttons: Create (red button), Get all Movies (yellow button).

Search Movie Form:

- Input field: La
- Button: Search (red button).

Movies Table:

Movie Id	Title	Genre	Director	Release
1	Gladiator	historical drama	Ridley Scott	2000
2	Interstellar	science fiction	Christopher Nolan	2014
3	La La Land	musical drama	Damien Chazelle	2016
4	The Silence of the Lambs	psychological horror	Jonathan Demme	1991

แก้ไขการแทรกข้อมูล insert data

เขียน function ในการสร้างข้อมูลเพื่อไป insert ใน DB ให้ถูกต้อง [frontend/src/services/MovieService.js]

```
// insert
export async function createMovie(data) {
  console.log("vreate test", data)
  const response = await fetch('http://localhost:3001/api/movie/insert', {
    method: 'POST',
    headers: {'Content-Type': 'application/json'},
    body: JSON.stringify({
      title: data.title,
      genre: data.genre,
      director: data.director,
      release_year: data.release_year
    })
  })
  return await response.json();
}
```

ผลการทดสอบ

Create Movie

Title	Genre
test3	test3
Director	Release
test3	3333

Create

Movies Created

3

Get all Movies

Search Movie

Search

Movies

Movie Id	Title	Genre	Director	Release
1	test	test	test	1111
2	test2	test2	test2	2222
3	test3	test3	test3	3333

เพิ่มการลบข้อมูล **delete data**

สร้างไฟล์ frontend/src/components/DeleteMovie.js

```
import React from 'react'

const DeleteMovie = ({onChangeForm, deleteMovie }) => {

  return(
    <div className="container">
      <div className="row">
        <div className="col-md-7 mrgnbtm">
          <h2>Delete Movie</h2>
          <form>
            <div className="row">
              <div className="form-group col-md-6">
                <input type="text" onChange={(e) => onChangeForm(e)} className="form-control" name="delete_text" id="delete_text" placeholder="movie name" />
              </div>
            </div>
            <button type="button" onClick= {(e) => deleteMovie("test") } className="btn btn-danger">Delete</button>
          </form>
        </div>
      </div>
    </div>
  )
}
```

```

        </div>
    )
}

export default DeleteMovie

```

เพิ่ม function movieDelete และ onChangeForm3 ใน App.js

```

...
//-----
...
import DeleteMovie from './components/DeleteMovie';
import { getAllMovies, createMovie, searchMovie, deleteMovie} from
'./services/MovieService'

function App() {

//-----
...
const [delete_data, setDelete_data] = useState({})

...
const movieDelete = (e) => {
    console.log("delete_data: ", delete_data.delete_text);

    deleteMovie(delete_data.delete_text)
        .then(movies => {
            console.log("delete data : ", movies?.data);

            if (movies && movies.data) {
                setMovies(movies.data);
                setNumberOfMovies(movies.data.length);
            } else {
                setMovies([]);
                setNumberOfMovies(0);
            }
        })
        .catch(error => {
            console.error("Error deleting movie: ", error);
            setMovies([]);
            setNumberOfMovies(0);
        });
}

...
const onChangeForm3 = (e) => {
    // console.log("TEST", e.target.name);
    if (e.target.name === 'delete_text') {
        delete_data.delete_text = e.target.value;
    }
    setDelete_data(delete_data)
}

```

```

return (
  <div className="App">
    <Header></Header>
    <div className="container mrgnbm">
      <div className="row">
        <div className="col-md-8">
          <CreateMovie
            // user={user}
            movie={movie}
            onChangeForm={onChangeForm}
            createMovie={movieCreate}
          >
          </CreateMovie>
        </div>
        <div className="col-md-4">
          <DisplayBoard
            numberOfMovies={numberOfMovies}
            getAllMovies={fetchAllMovies}
          >
          </DisplayBoard>
        </div>
      </div>
    </div>
    <div className="row mrgnbm">

      <SearchMovie
        // user={user}
        movie={movie}
        onChangeForm={onChangeForm2}
        searchMovie={movieSearch}
      >
      </SearchMovie>

      <DeleteMovie
        movie={movie}
        onChangeForm={onChangeForm3}
        deleteMovie={movieDelete}>
      </DeleteMovie>

    {/* <SearchBoard
      // user={user}
      search_data={search_data}
      onChangeForm2={onChangeForm2}
      searchMovies={searchMovie}
    >
    </SearchBoard> */}

    </div>
    <div className="row mrgnbm">
      <Movies movies={movies}></Movies>
    </div>

  </div>
);

```

```
}

export default App;
```

เพิ่ม function เชื่อมต่อกับ back-end ในการ delete ในไฟล์ [frontend/src/services/MovieService.js]

```
...

export async function deleteMovie(delete_data) {
    try {
        console.log('delete_data: ' + delete_data);
        const response = await fetch('http://localhost:3001/api/movie/delete?
search_text=' + delete_data);
        // console.log('Delete Movie response: ' + await response);
        return await response.json(); //***
    } catch (error) {
        return [];
    }
}
...
```

ผลการทดลอง

The screenshot shows a web application interface with several components:

- Create Movie Form:** A form with fields for Title (test), Director (test), Genre (test), and Release (1111). A red "Create" button is at the bottom.
- Search Movie Form:** A search bar containing "test". A red "Search" button is below it.
- Delete Movie Form:** A form with a single input field containing "test". A red "Delete" button is below it.
- Movies Table:** A table titled "Movies" with columns: Movie Id, Title, Genre, Director, and Release. It contains one row with values: 1, test, test, test, 1111.
- Success Message:** A green box on the right side displays the number "1" and a yellow "Get all Movies" button.

หลังกดปุ่ม delete

React With NodeJS

Create Movie

Title	Genre
<input type="text" value="test"/>	<input type="text" value="test"/>
Director	Release
<input type="text" value="test"/>	<input type="text" value="1111"/>

[Create](#)

Search Movie

[Search](#)

Delete Movie

[Delete](#)

Movies Created

0

[Get all Movies](#)

code เดี๋มอยู่ใน GitHub