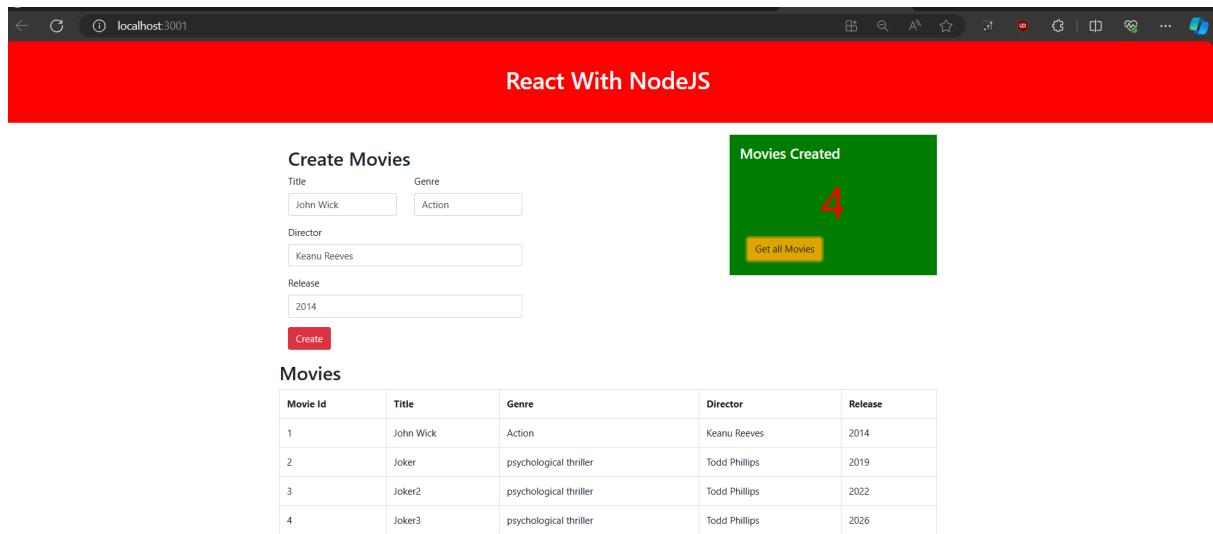
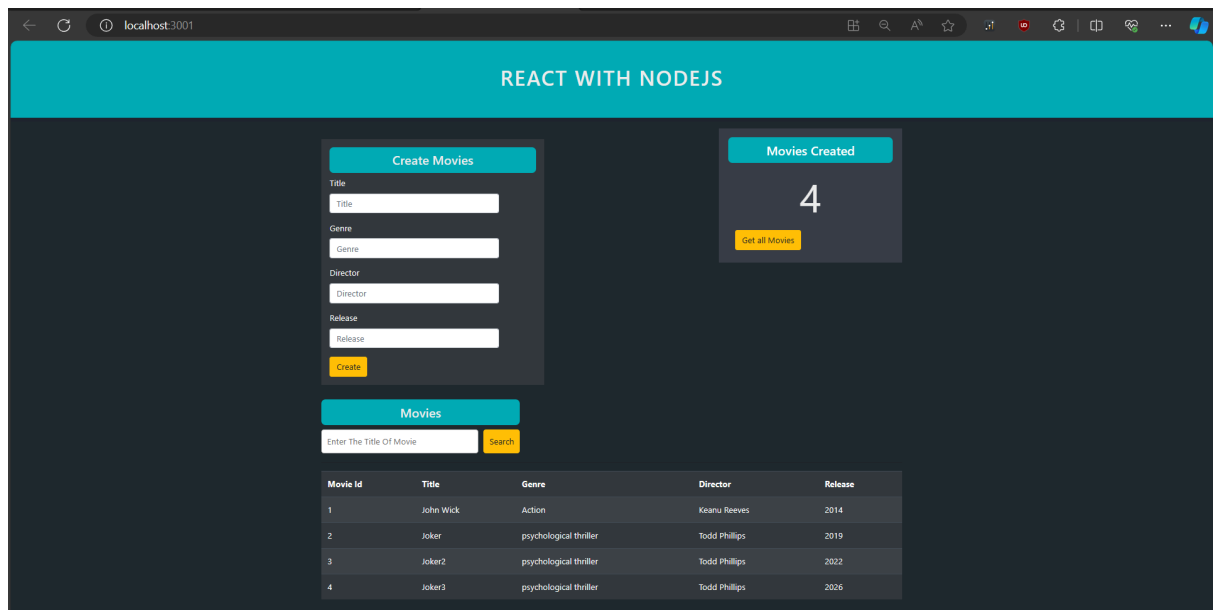


ENGSE203 - Mid-term

หน้า WEB ก่อนปรับปรุง



หน้า WEB หลังปรับปรุง



การทำงานของ Web

กรอกข้อมูลแล้วกด Create

The screenshot shows a web application running on localhost:3001. The header is teal with the text "REACT WITH NODEJS". The main content area has a dark background. On the left, there is a "Create Movies" form with fields for Title (John Wick 2), Genre (Action), Director (Keanu Reeves), and Release (2018), and a yellow "Create" button. To the right, a "Movies Created" box shows the number "5" and a yellow "Get all Movies" button. Below the form is a "Movies" section with a search bar and a yellow "Search" button. At the bottom, a table lists movies with columns: Movie Id, Title, Genre, Director, and Release.

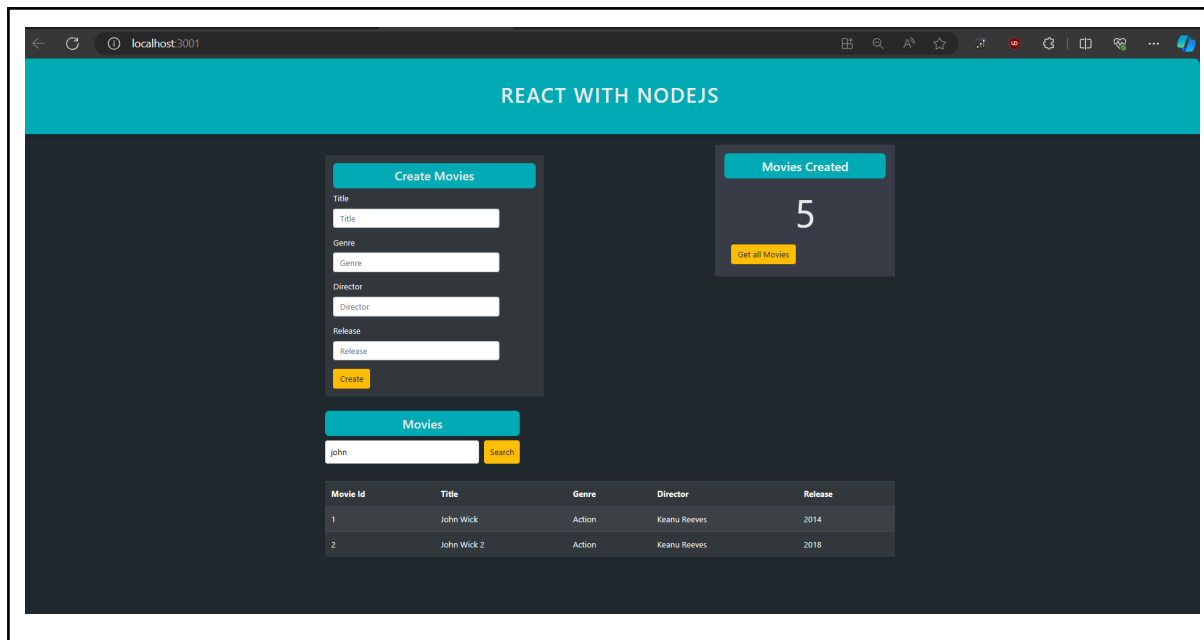
Movie Id	Title	Genre	Director	Release
1	John Wick	Action	Keanu Reeves	2014
2	Joker	psychological thriller	Todd Phillips	2019
3	Joker2	psychological thriller	Todd Phillips	2022
4	Joker3	psychological thriller	Todd Phillips	2026

ตัวเลขจะเพิ่มขึ้นจากนั้นให้กด GET ALL MOVIE เพื่ออัปเดตข้อมูล

This screenshot shows the same web application after clicking the "Get all Movies" button. The "Movies Created" box now displays the number "5". The table below the search bar has been updated with five rows of movie data.

Movie Id	Title	Genre	Director	Release
1	John Wick	Action	Keanu Reeves	2014
2	John Wick 2	Action	Keanu Reeves	2018
3	Joker	psychological thriller	Todd Phillips	2019
4	Joker2	psychological thriller	Todd Phillips	2022
5	Joker3	psychological thriller	Todd Phillips	2026

Search เพื่อค้นหาชื่อหนัง



Code ที่ปรับแต่ง

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 { getAllMovies, createMovie } from "../services/MovieService";
```

นำเข้าคอมโพเนนต์ Header, DisplayBoard, Movies, และ CreateMovie
นำเข้าฟังก์ชันบริการ (getAllMovies และ createMovie) จาก MovieService

```
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 userCreate = (e) => {
  createUser(user)
    .then(response => {
      console.log(response);
      setNumberOfUsers(numberOfUsers+1)
    });
}
*/

```

ประกาศตัวแปรสถานะหลายตัวด้วย useState hook เพื่อเก็บข้อมูลให้กับ movie, movies, และ numberOfMovies

```

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

/*
const fetchAllUsers = () => {
  getAllUsers()
    .then(users => {
      console.log(users)
      setUsers(users);
      setNumberOfUsers(users.length)
    });
}
*/

```

สร้างฟังก์ชัน movieCreate ที่ใช้ในการสร้างหนังใหม่ โดยเรียกใช้จาก createMovie และ อัปเดตสถานะด้วยหนังใหม่

```

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

```

```

        setNumberOfMovies(movies.length);
    });
};

/*
useEffect(() => {
    getAllUsers()
    .then(users => {
        console.log(users)
        setUsers(users);
        setNumberOfUsers(users.length)
    });
});
*/

```

สร้างฟังก์ชัน `fetchAllMovies` เพื่อใช้ในการดึงข้อมูลหนังทั้งหมด โดยใช้จาก `getAllMovies`

```

useEffect(() => {
    getAllMovies().then((movies) => {
        console.log(movies);
        setMovies(movies);
        setNumberOfMovies(movies.length);
    });
}, []);

/*
const onChangeForm = (e) => {
    if (e.target.name === 'firstname') {
        user.firstName = e.target.value;
    } else if (e.target.name === 'lastname') {
        user.lastName = e.target.value;
    } else if (e.target.name === 'email') {
        user.email = e.target.value;
    }
    setUser(user)
}
*/

```

สร้าง Hook `useEffect` ใช้ในการดึงข้อมูลหนังทั้งหมด

```

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);
  };

```

แก้ไขตัวแปร title , genre , director , release_year เพื่อให้ตรงกับข้อมูลใน database

```

return (
  <div className="App">
    <Header></Header>
    <div className="container mrgnbtm">
      <div className="row">
        /* <div className="col-md-8">
          <CreateUser
            user={user}
            onChangeForm={onChangeForm}
            createUser={userCreate}
          >
          </CreateUser>
        </div> */
        <div className="col-md-8">
          <CreateMovie
            movie={movie}
            onChangeForm={onChangeForm}
            createMovie={movieCreate}
          ></CreateMovie>
        </div>
      <div className="col-md-4">
        <DisplayBoard
          numberOfMovies={numberOfMovies}
          getAllMovies={fetchAllMovies}
        ></DisplayBoard>
      </div>
    </div>{" "}
    /* <div className="row mrgnbtm">
      <Users users={users}></Users>
    </div> */

```

```

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

export default App;

```

กำหนดโครงสร้าง UI แสดง UI ของแอปพลิเคชัน React โดยใช้ Bootstrap classes

CreateMovies.js

```

import React from "react";

const CreateMovies = ({ onChangeForm, createMovie }) => {
  return (
    <div className="container">
      <div className="row">
        <div className="p-3 mb-2 bg-dark text-white col-md-7 mrgnbtm">
          <h2>Create Movies</h2>
          <form>
            <div className="row">
              <div className="form-group col-md-10">
                <label htmlFor="exampleInputEmail1">Title</label>
                <input
                  type="text"
                  onChange={ (e) => onChangeForm(e) }
                  className="form-control"
                  name="title"
                  id="title"
                  aria-describedby="emailHelp"
                  placeholder="Title"
                />
              </div>
            </div>
            <div className="row">
              <div className="form-group col-md-10">
                <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-10">
        <label htmlFor="exampleInputEmail1">Director</label>
        <input
          type="text"
          onChange={ (e) => onChangeForm(e) }
          className="form-control"
          name="director"
          id="director"
          aria-describedby="emailHelp"
          placeholder="Director"
        />
      </div>
    </div>
    <div className="row">
      <div className="form-group col-md-10">
        <label htmlFor="exampleInputEmail1">Release</label>
        <input
          type="text"
          onChange={ (e) => onChangeForm(e) }
          className="form-control"
          name="release_year"
          id="release_year"
          aria-describedby="emailHelp"
          placeholder="Release"
        />
      </div>
    </div>
    <button
      type="button"
      onClick={ (e) => createMovie() }
      className="btn btn-warning"
```



```

        >
        Create
      </button>
    </form>
  </div>
</div>
</div>
);
};

export default CreateMovies;

```

แก้ไขชื่อของ title , genre , director และเพิ่ม column release_year

DisplayBoard.js

```

import React from "react";

export const DisplayBoard = ({ numberOfMovies, getAllMovies }) => {
  return (
    <div style={{ backgroundColor: "#393E46" }}
      className="display-board">
      <h2 style={{ color: "white" }}>Movies Created</h2>
      <div className="number">{numberOfMovies}</div>
      <div className="btn">
        <button
          type="button"
          onClick={ (e) => getAllMovies() }
          className="btn btn-warning"
        >
          Get all Movies
        </button>
      </div>
    </div>
  );
};

```

ใช้แสดงจำนวนหนังและมีปุ่มเพื่อเรียกใช้ getAllMovies อัปเดตข้อมูลหนัง

MovieService.js

```
export async function getAllMovies() {
  try {
    const response = await
fetch("http://localhost:4000/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();
}
```

แก้ url ของ getAllMovies และ createMovie ให้ตรงกับ api ที่จะเรียกใช้

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);
```

ประกาศตัวแปรสถานะหลายตัวด้วย useState hook เพื่อเก็บข้อมูลให้กับ searchText, searchResults, และ issearching

```

const handleSearch = async () => {
  try {
    const response = await
fetch(`http://localhost:4000/api/movie/search?search_text=${searchTex
t}`);
    const data = await response.json();

    if (data.returnCode === 1) {
      setSearchResults(data.data);
      setIsSearching(true);
    } else {
      setSearchResults([]);
      setIsSearching(false);
    }
  } catch (error) {
    console.error('Error searching for movies:', error);
  }
};

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 = isSearching ?
  searchResults.map((movie, index) => MovieRow(movie, index)) :
  movies.map((movie, index) => MovieRow(movie, index));

```

เมื่อ handlesearch ทำงาน(กดปุ่มsearch)จะเรียกใช้ fetch ข้อมูลจาก api

```

return (
  <div className="container">
    <div style={{ display: "flex", justifyContent: "flex-start",
alignItems: "center" }}>
      <div>
        <h2>Movies</h2>
        <div style={{ display: "flex", alignItems: "center" }}>
          <input
            type="text"
            className="Search_input"
            style={{
              padding: "10px",
              fontSize: "16px",
              border: "1px solid #ccc",
              borderRadius: "5px",
              width: "300px",
              marginRight: "10px",
            }}
            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>
    </table>
  </div>
)

```

```

        </tr>
    </thead>
    <tbody>
        {movieTable}
        {isSearching && searchResults.length === 0 && <tr><td
colSpan="5">No movies found</td></tr>}
    </tbody>
</table>
</div>
);
};

```

เพิ่ม searchinput และ ปุ่ม search ใช้ onChange เพื่อเก็บข้อความที่ต้องการค้นหา

app.css

```

.header {
    width: 100%;
    padding: 2%;
    background-color: burlywood;
    color: white;
    text-align: center;
}

.display-board {
    width: 100%;
    background-color: rgb(555, 200, 789);
    padding: 5%;
}

.number {
    color: #eeeeee;
    font-size: 75px;
    text-align: center;
}

.mrgnbtm {
    margin-top: 20px;
}

```

```

}
body {
  background-color: #222831;
}
h2 {
  background-color: #00adb5;
  color: #eeeeee;
  padding: 10px;
  text-align: center;
  font-size: 24px;
  border-radius: 8px;
}

```

ตกแต่ง web

ฝั่ง api

index.js

```

server.route({
  method: 'GET',
  path: '/api/movie/search',
  config: {
    cors: {
      origin: ['http://localhost:3001'],
      additionalHeaders: ['cache-control',
'x-requested-width']
    }
  },
  handler: async function (request, reply) {
    var param = request.query;
    const search_text = param.search_text;
    //const title = param.title;

    try {

      const responsedata = await
Movies.MovieRepo.getMovieSearch(search_text);
      if (responsedata.error) {

```

```
        return responsedata.errMessage;
    } else {
        return responsedata;
    }
} catch (err) {
    server.log(["error", "home"], err);
    return err;
}

}

});
```

เปลี่ยน origin เป็น localhost port 3001 เพื่อให้ web สามารถเรียกใช้ api ได้