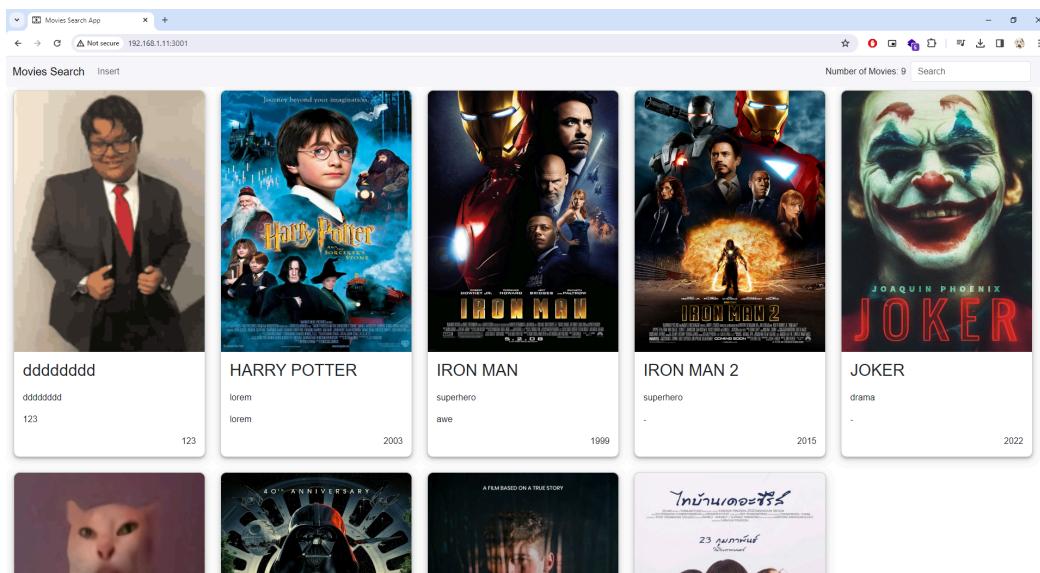


## ENGSE203 - Mid-term

### แสดงหน้าเว็บและทดสอบ

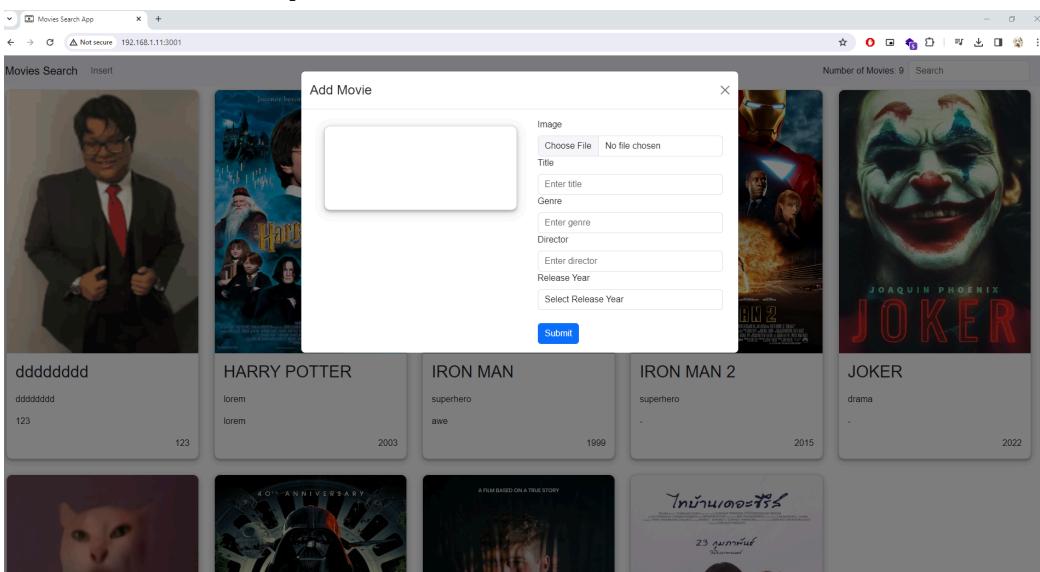
#### หน้าเว็บ

##### แสดงหนังทุกเรื่อง



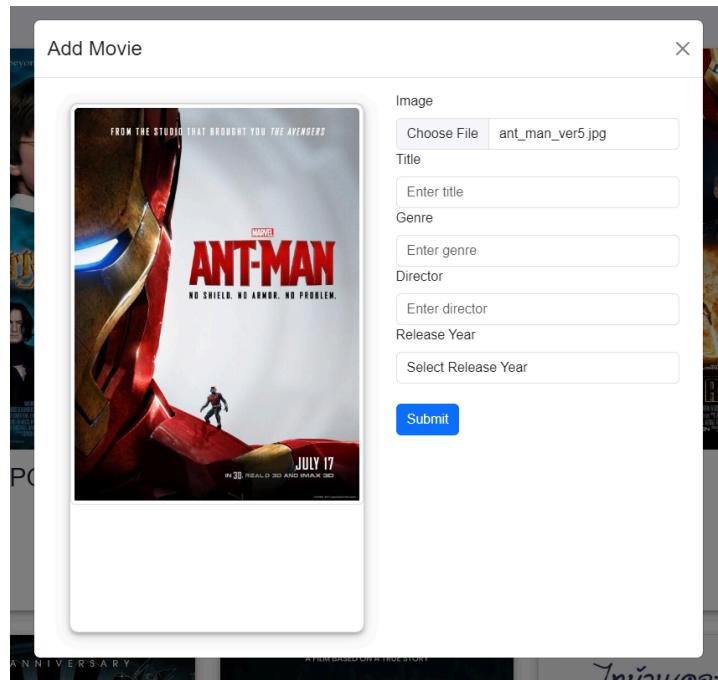
#### เพิ่มหนัง

ปุ่ม insert เพื่อทำการเพิ่มหนัง  
หน้าต่างเพิ่มหนังแสดงขึ้น ใส่ข้อมูลด้านขวา จะแสดงตัวอย่างด้านซ้าย

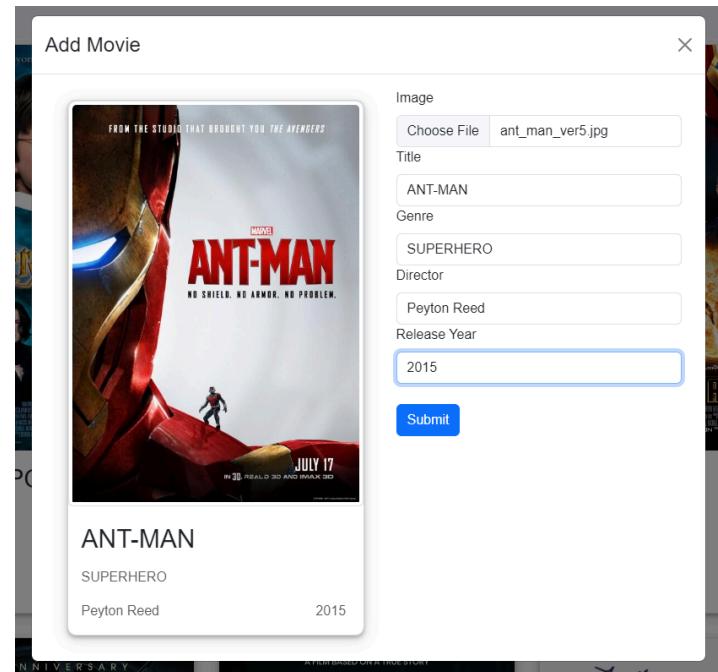


นาย จักรภัทร อินผูก 66543210005-5

### ทดลองใส่รูป

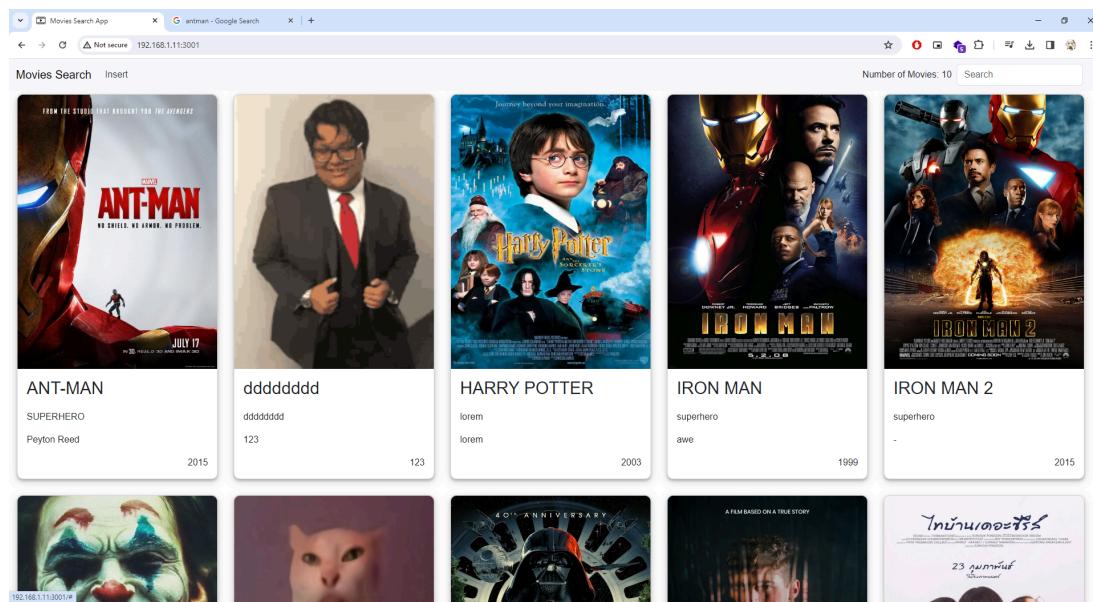


### ทดลองใส่ข้อมูลหนัง



นาย จักรภัทร อินผูก 66543210005-5

หลังจากนั้นกด Submit เพื่อเพิ่มนั้ง



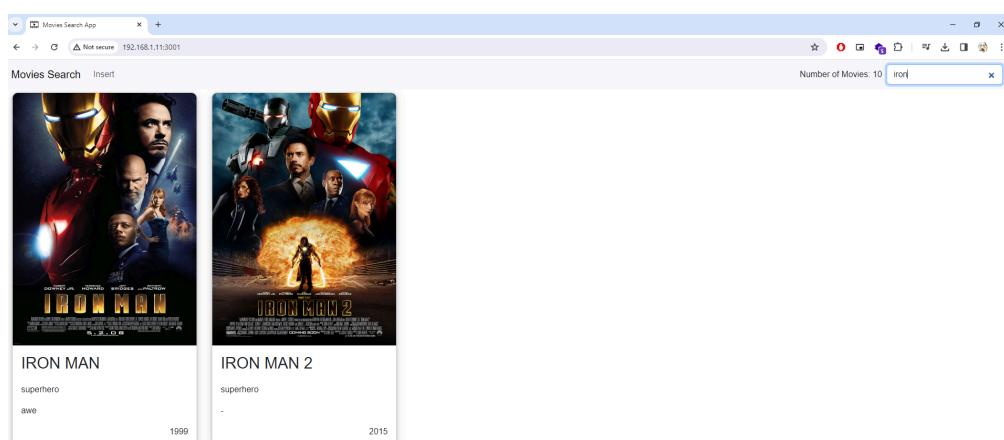
หนังก็จะทำการเพิ่มลงไปในเว็บเรียบร้อย

## ค้นหาหนัง

ใส่ชื่อที่การลงในช่องค้นหา Search

Number of Movies: 10

ทดลองค้นหา iron ผลลัพธ์



## Client side app

### สิ่งที่ทำการปรับปรุง

```

import React, { useState, useEffect } from "react";
import "bootstrap/dist/css/bootstrap.min.css";
import "./App.css";
import { Header } from "./components/Header";
import { users } from "./components/Users";
import { displayBoard } from "./Components/DisplayBoard";
import { createUser } from "./components/CreateUser";
import { getAllUsers, createUser } from "./services/UserService";
//-----
import { Movies } from "./components/Movies";
import { getAllMovies, createMovie } from "./services/MovieService";

function App() {
  const [user, setUser] = useState();
  const [users, setUsers] = useState();
  const [numberOfUsers, setNumberOfUsers] = useState(0);
  //-----
  const [movies, setMovies] = useState();
  const [numberOfMovies, setNumberOfMovies] = useState(1);
  const userCreate = (e) => {
    createUser(user)
      .then(response => {
        console.log(response);
        setNumberOfUsers(numberOfUsers + 1);
      });
  };
  const fetchAllUsers = () => {
    getAllUsers()
      .then((movies) => {
        console.log(movies);
        setMovies(movies);
        setNumberOfMovies(movies.length);
      });
  };
  //-----
  useEffect(() => {
    getAllMovies().then((movies) => {
      console.log(movies);
      setMovies(movies);
      setNumberOfMovies(movies.length);
    });
  }, []);
}

function App() {
  const [showModal, setShowModal] = useState(false);
  const handleToggleModal = () => {
    setShowModal(!showModal);
  };
  const handleInsertClick = () => {
    handleToggleModal();
  };
  /* showMovie */
  const [movie, setMovie] = useState();
  const [movies, setMovies] = useState();
  const [numberOfMovies, setNumberOfMovies] = useState(0);
  //-----
  useEffect(() => {
    getAllMovies().then((movies) => {
      console.log(movies);
      setMovies(movies);
      setNumberOfMovies(movies.length);
    });
  }, []);
}

function App() {
  const [showModal, setShowModal] = useState(false);
  const handleToggleModal = () => {
    setShowModal(!showModal);
  };
  const handleInsertClick = () => {
    handleToggleModal();
  };
  /* showMovie */
  const [movie, setMovie] = useState();
  const [movies, setMovies] = useState();
  const [numberOfMovies, setNumberOfMovies] = useState(0);
  //-----
  useEffect(() => {
    getAllMovies().then((movies) => {
      console.log(movies);
      setMovies(movies);
      setNumberOfMovies(movies.length);
    });
  }, []);
}

```

### Header

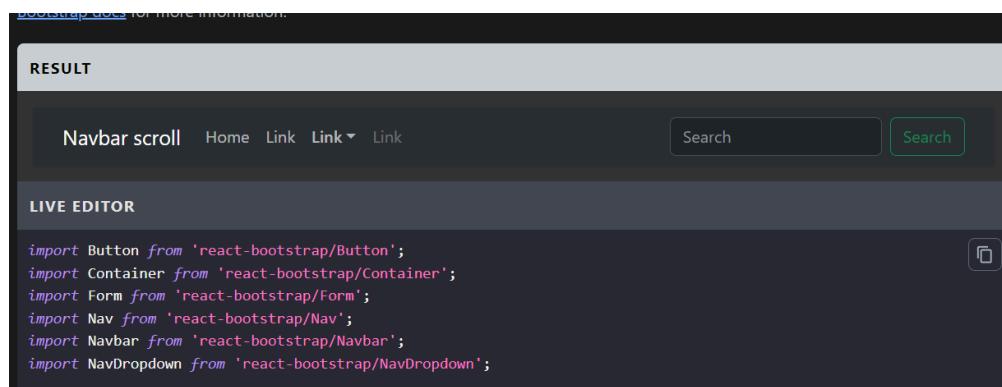
#### Header แบบเก่า



#### Header ที่ทำการปรับปรุง



ทำการเพิ่ม css framework bootstrap เพื่อให้ง่ายต่อการใช้งาน ทำการเอา [template navbar](#) ของ bootstrap มาใช้



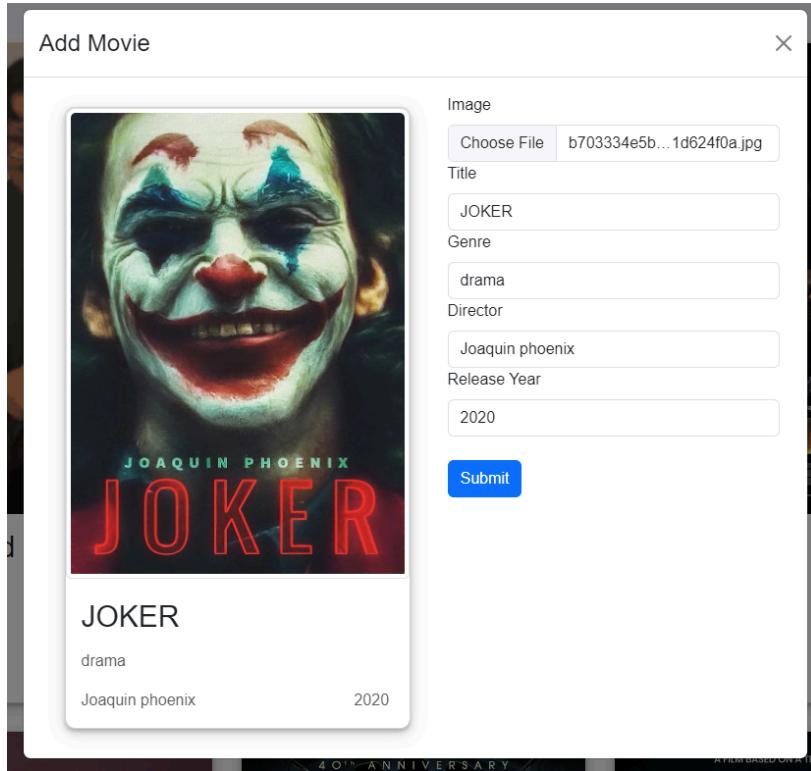
และทำการปรับแต่งเล็กน้อย โดยการนำปุ่มเปิดปิดฟอร์มในการเพิ่มหนังลงในเว็บลง จำนวนของหนังทั้งหมด และค้นหาหนัง ลงไปในส่วนนี้เช่นกัน

```
import React from "react";
import Container from "react-bootstrap/Container";
import Form from "react-bootstrap/Form";
import Nav from "react-bootstrap/Nav";
import Navbar from "react-bootstrap/Navbar";
export function getNumberOfMovies() [
  return;
]
export const Header = ({ onInsertClick, onSearch, numberOfMovies }) => {
  return (
    <Navbar expand="lg" className="bg-body-tertiary fixed-top">
      <Container fluid>
        <Navbar.Brand href="">Movies Search </Navbar.Brand>
        <Navbar.Toggle aria-controls="navbarScroll" />
        <Navbar.Collapse id="navbarScroll">
          <Nav
            className="me-auto my-2 my-lg-0"
            style={{ maxHeight: "100px" }}
            navbarScroll
          >
            <Nav.Link onClick={onInsertClick}>Insert</Nav.Link>
          </Nav>
          <Form className="d-flex">
            <Nav.Link disabled>
              Number of Movies: {numberOfMovies} &ampnbsp
            </Nav.Link>
          </Form>
          <Form className="d-flex">
            <Form.Control
              type="search"
              placeholder="Search"
              className="me-2"
              aria-label="Search"
              onChange={(e) => onSearch(e)}
            />
            </Form>
          </Navbar.Collapse>
        </Container>
    </Navbar>
  );
}
```

ใช้ค่าสั่ง onclick เมื่อกดแล้วจะทำให้ app.js ใช้ฟังก์ชัน onInsertclick และ onchang หลังจากผู้ใช้พิมพ์ชื่อหนังจะส่งไปให้ที่ app.js เพื่อทำการค้นหาหนังตามที่ผู้ใช้กำหนด

นาย จักรภัทร อินผูก 66543210005-5

## Add movie



ทำการปรับปรุงให้สามารถใส่รูปได้และแสดงตัวอย่างก่อนเพิ่มข้อมูลเข้าไปบนเว็บ

```
import React, { useState } from "react";
import { Modal, Button, Form, Image, Container, Row, Col } from "react-bootstrap";

const MovieFormModal = ({
  onChangeForm,
  showModal,
  handleClose,
  createMovie,
}) => {
  const [title, setTitle] = useState("");
  const [genre, setGenre] = useState("");
  const [director, setDirector] = useState("");
  const [releaseYear, setReleaseYear] = useState("");
  const [image, setImage] = useState(null);

  const currentYear = new Date().getFullYear();
  const numberOfYearsInPast = 50; // Change this value as needed

  const years = Array.from({ length: numberOfYearsInPast + 1 }, (_, index) =>
    (currentYear - index).toString()
  );
}
```

ส่วนแรกจะเป็นการนำเข้าส่วนเสริมที่ต้องการใช้งานและตัวแปลลต่างๆเพื่อใช้เก็บข้อมูลในฟอร์ม

```
const handleLocalFormSubmit = async (e) => {
  e.preventDefault();
  try {
    // Reset form fields and selected file after submission
    setTitle("");
    setGenre("");
    setDirector("");
    setReleaseYear("");
    setImage(null);

    // Close the modal
    handleClose();
  } catch (error) {}
};

const handleImageChange = (e) => {
  const file = e.target.files[0];
  if (file && (file.type === "image/jpeg" || file.type === "image/png")) {
    setImage(file);
  } else {
    window.alert("Invalid file type. Please select a JPG or PNG file.");
    setImage(null);
  }
};

const handleModalHide = () => {
  // Reset form fields when the modal is closed
  setTitle("");
  setGenre("");
  setDirector("");
  setReleaseYear("");
  setImage(null);
};
```

ฟังก์ชัน

handleLocalFormSubmit ใช้สำหรับเคลียร์ฟอร์มหลังกด submit และปิด form

handleImageChange ใช้สำหรับคัดหรองรูปภาพให้เป็นนามสกุล png และ jpeg เท่านั้นหลังจากนั้นก็นำรูปไปแสดงที่หน้าเว็บเพื่อแสดงตัวอย่างรูป

handleModalHide ใช้สำหรับเวลากดปิดฟอร์มจะเคลียร์ค่าทั้งหมดในฟอร์ม

```
<Modal.Header closeButton>
| <Modal.Title>Add Movie</Modal.Title>
</Modal.Header>
<Modal.Body>
<Container>
  <Row>
    <Col>
      <div className="d-flex justify-content-center">
        <div className="card movie_card">
          {image && (
            <Image src={URL.createObjectURL(image)} thumbnail />
          )}
          <div className="card-body">
            <p className="card-title">{title}&nbsp;</p>
            <p className="movie_info genre">{genre}&nbsp;</p>
            <span className="movie_info director">
              {director}&nbsp;
            </span>
            <span className="movie_info float-end">
              {releaseYear}&nbsp;
            </span>
          </div>
        </div>
      </Col>
      <Col>
```

ส่วนแรกของ form จะเป็นการส่วนของการแสดงตัวอย่างของฟอร์มแล้วจะสามารถเห็นตัวอย่างก่อนที่จะนำลงเข้าสู่เว็บ

โดยการใช้ onChange เป็นการตั้งข้อมูลที่ผู้ใช้ใส่namaและแสดงตัวอย่าง

```
<Col>
  <Form
    onSubmit={handleLocalFormSubmit}
    onChange={(e) => onChangeForm(e)}
  >
    <Form.Group controlId="formImage">
      <Form.Label>Image</Form.Label>
      <Form.Control type="file" onChange={handleImageChange} />
    </Form.Group>
    <Form.Group controlId="formTitle">
      <Form.Label>Title</Form.Label>
      <Form.Control
        type="text"
        placeholder="Enter title"
        value={title}
        onChange={(e) => setTitle(e.target.value)}
      />
    </Form.Group>

    <Form.Group controlId="formGenre">
      <Form.Label>Genre</Form.Label>
      <Form.Control
        type="text"
        placeholder="Enter genre"
        value={genre}
        onChange={(e) => setGenre(e.target.value)}
      />
    </Form.Group>
```

นาย จักรภัทร อินผูก 66543210005-5

onChange บน form เมื่อมีการเปลี่ยนแปลง form จะส่งค่าของผู้ใช้ไปเก็บไว้ในตัวแปรใน App.js เพื่อใช้งานต่อไปในการเพิ่มข้อมูลลงใน server

```
<Form.Group controlId="formReleaseYear">
  <Form.Label>Release Year</Form.Label>
  <Form.Control
    as="select"
    value={releaseYear}
    onChange={(e) => setReleaseYear(e.target.value)}
  >
    <option value="">Select Release Year</option>
    {years.map((year) => (
      <option key={year} value={year}>
        {year}
      </option>
    )))
  </Form.Control>
</Form.Group>

<br><br>
<Button
  variant="primary"
  type="submit"
  onClick={(e) => createMovie()}
>
  Submit
</Button>
</Form>
</Col>
</Row>
</Container>
</Modal.Body>
</Modal>
);
};

export default MovieFormModal;
```

หลังจากกดปุ่ม submit ปุ่มจะไปเรียกใช้ฟังก์ชัน createmovie ใน app.js

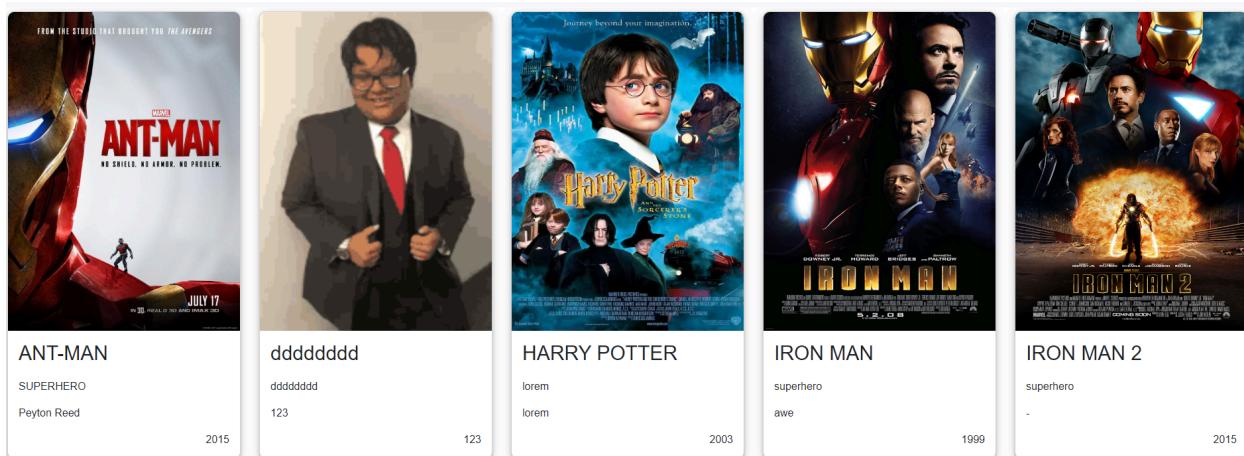
## MovieCard

ทำการปรับปรุงจากตารางเดิม

## Users

User Id	Firstname	Lastname	Email
1	first1	last1	abc@gmail.com
2	first2	last2	abc@gmail.com
3	first3	last3	abc@gmail.com

ให้เป็น card สวยๆ



```
import React from "react";
import "../css/moviecard.css";

const MovieCard = (props) => {
  const API_IMG = "http://192.168.1.21:4000/api/images/";

  return (
    <div className="card movie_card">
      <img
        src={API_IMG + props.Image_name}
        alt={`Movie poster for ${props.title}`}
        className="card-img-top"
      />
      <div className="card-body">
        <h5 className="card-title">{props.title}</h5>
        <p className="card-text">{props.genre}</p>
        <p className="card-text">{props.director}</p>
        <p className="card-text float-end">{props.release_year}</p>
      </div>
    </div>
  );
}

export default MovieCard;
```

โดยการนำข้อมูลที่ได้มาจากการฟังก์ชันใน app.js ที่ดึงมาจากdatabase แต่ละข้อมูลเพื่อมาแสดงบนเว็บ

## App

```
function App() {
  /* Modal */
  const [showModal, setShowModal] = useState(false);

  const handleToggleModal = () => {
    setShowModal(!showModal);
  };
  const handleInsertClick = () => {
    handleToggleModal();
  };
  /* showMovie */
  const [movie, setMovie] = useState([]);
  const [movies, setMovies] = useState([]);
  const [numberOfMovies, setNumberOfMovies] = useState(0);
```

ทำการเพิ่มตัวแปรnidหน่อยเพื่อปรับปรุงการทำงานต่างๆในส่วนของการเปิดปิด ฟอร์ม

## ฟังก์ชัน

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

const MovieInsert = (e) => {
  console.log(movie);
  createMovie(movie).then((response) => {
    console.log(response);
    getAllMovies().then((movies) => {
      console.log(movies);
      setMovies(movies);
      setNumberOfMovies(movies.length);
    });
    setNumberOfMovies(numberOfMovies + 1);
  });
};
```

useEffect คือฟังก์ชันในตอนเปิดเว็บให้ไปเรียกขอข้อมูลจาก database เพื่อนำมาแสดงบนหน้าเว็บและจำนวนหนังตามจำนวนข้อมูล

MovieInsert เป็นการเพิ่มหนังโดยจะนำตัวแปรที่ได้จากการใส่ form ส่งไปยัง server และไปเรียกข้อมูลหลังการเพิ่มข้อมูลเพื่อนำมาเปลี่ยนแปลงบนหน้าเว็บ และเพิ่มจำนวนหนังไป 1 ทุกการเพิ่ม 1 ครั้ง

```
const handleSearch = (e) => {
  console.log(e.target.value);
  searchMovie(e.target.value).then((response) => {
    console.log(response.data);
    setMovies(response.data);
  });
};

const onChangeForm = (e) => {
  if (e.target.id === "formTitle") {
    movie.title = e.target.value;
  } else if (e.target.id === "formGenre") {
    movie.genre = e.target.value;
  } else if (e.target.id === "formDirector") {
    movie.director = e.target.value;
  } else if (e.target.id === "formReleaseYear") {
    movie.releaseYear = e.target.value;
  } else if (e.target.id === "formImage") {
    movie.image = e.target.files[0];
  }
  setMovie(movie);
};
```

handleSearch คือฟังก์ชันในการส่งข้อมูลที่ได้จากผู้ใช้การค้นหาข้อมูลใน database และนำมาแสดงบนเว็บ

onChangeForm หลังจากผู้ใช้ใส่ข้อมูลลงใน form จะนำข้อมูลมาเก็บในตัวแปร movie และนำไปใช้ต่อในฟังก์ชัน MovieInsert เพื่อส่งข้อมูลไปยัง server

```
return (
  <div className="App">
    <Header
      onInsertClick={handleInsertClick}
      onSearch={handleSearch}
      numberOfMovies={numberOfMovies}
    />
    <MovieFormModal
      showModal={showModal}
      handleClose={handleToggleModal}
      onChangeForm={onChangeForm}
      createMovie={MovieInsert}
    />
    <div className="movies">
      {movies && Array.isArray(movies) ? (
        movies.map((moviee) => <MovieCard key={moviee.id} {...moviee} />)
      ) : (
        <></>
      )}
    </div>
  </div>
);
```

แสดงส่วนต่างๆที่นำเข้ามาจาก component อื่นๆ และนำข้อมูลที่ได้จาก server มา map เป็น card แสดงบนเว็บ

## MovieService

```
export async function getAllMovies() {
  try {
    const response = await fetch("http://192.168.1.21:4000/api/movie/all");
    return await response.json();
  } catch (error) {
    return [];
  }
}

export async function createMovie(data) {
  const formData = new FormData();
  formData.append("title", data.title);
  formData.append("genre", data.genre);
  formData.append("director", data.director);
  formData.append("release_year", data.releaseYear);
  formData.append("Image_name", data.image);
  console.log(formData);
  const response = await fetch(`http://192.168.1.21:4000/api/movie/insert`, {
    method: "POST",
    body: formData,
  });

  return await response.json();
}

export async function searchMovie(search_text) {
  const response = await fetch(
    `http://192.168.1.21:4000/api/movie/search?search_text=${search_text}`,
    {
      method: "GET",
      headers: { "Content-Type": "application/json" },
    }
  );
  return await response.json();
}
```

ปรับปรุงโดยการเพิ่มฟังก์ชันส่งคำสั่งค้นมาไปยัง server

# Server API

## Index

สิงที่ที่ปรับปรุง

```
const hapi = require("@hapi/hapi");
const Inert = require("@hapi/inert");
const env = require("./env.js");
const Movies = require("./repository/movie.js");
// const express = require('express');
// const app = express();
const fs = require("fs");
```

ทำการเพิ่มใช้สำหรับจัดการไฟล์และไดเรกทอรีบนระบบปฏิบัติการใช้ในการเขียนไฟล์รูปภาพที่รับมากจากผู้ใช้งาน

```
server.route({
  method: "GET",
  path: "/api/images/{filename}",
  config: {
    cors: {
      origin: ["*"],
      additionalHeaders: ["cache-control", "x-requested-width"],
    },
  },
  handler: (request, h) => {
    const { filename } = request.params;
    return h.file(`./resources/static/assets/uploads/${filename}`);
  },
});
```

เพิ่ม api ในการนำรูปที่นำเข้ามาแสดงเว็บ ผ่าน /api/image/ชื่อรูป

ในส่วนของการดึงข้อมูลกับหาข้อมูลจะเหมือนเดิมไม่ได้แก่ไขอะไร คือการส่งคำสั่งไปดึงข้อมูลจาก database มาและส่งไปให้ผู้เรียกข้อมูล

```
server.route({
  method: "GET",
  path: "/api/movie/all",
  config: {
    cors: {
      origin: ["*"],
      additionalHeaders: ["cache-control", "x-requested-width"],
    },
  },
  handler: async function (request, reply) {
    //var param = request.query;
    //const category_code = param.category_code;

    try {
      const responddata = await Movies.MovieRepo.getMovieList();

      if (responddata.error) {
        return responddata.errorMessage;
      } else {
        console.log("/api/movie/all called!");
        console.log(responddata);
        return responddata;
      }
    } catch (err) {
      server.log(["error", "home"], err);
      return err;
    }
  },
});
```

```
server.route({
  method: "GET",
  path: "/api/movie/search",
  config: {
    cors: {
      origin: ["*"],
      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 responddata = await Movies.MovieRepo.getMovieSearch(search_text);
      if (responddata.error) {
        return responddata.errorMessage;
      } else {
        return responddata;
      }
    } catch (err) {
      server.log(["error", "home"], err);
      return err;
    }
  },
});
```

```
server.route({
  method: "POST",
  path: "/api/movie/insert",
  config: {
    payload: {
      output: "stream",
      multipart: true,
      uploads: "file",
    },
    cors: {
      origin: ["*"],
      additionalHeaders: ["cache-control", "x-requested-width"],
    },
  },
  handler: async function (request, h) {
    console.log("request : " + request);
    try {
      console.log(request.payload.Image_name.hapi.filename);

      const { title, genre, director, release_year } = request.payload;
      handleFileUpload(request.payload.Image_name);
      const imageName = MoviesName; // Extract image name
      // console.log(MoviesName);
      const responsedata = await Movies.MovieRepo.postMovie(
        title,
        genre,
        director,
        release_year,
        imageName
      );
    }
  }
});
```

ในส่วนของการเพิ่มข้อมูลจากผู้ใช้หลังจากที่ผู้ใช้ส่งข้อมูลมา�ัง server จะนำข้อมูลจาก payload ที่ผู้ใช้ส่งมาไปเก็บยังตัวแปรต่างๆที่กำหนดเหมือนที่ผู้ใช้ส่งมา

ต่อไปจะนำรpayload ไฟล์ที่ส่งมาลงไปยังฟังก์ชัน handleFileUploadเพื่อทำการเปลี่ยนชื่อไฟล์และเก็บไฟล์ลงในเครื่องจากนั้นจะนำชื่อไฟล์ใหม่มายัง imagename และนำตัวแปรทั้งหมดส่งไปยังฟังก์ชันpostMovieในmovie.js เพื่อนำไปเก็บข้อมูลต่างๆไว้ใน database

## Movie database

### Database

title	genre	director	release_year	Image_name
ANT-MAN	SUPERHERO	Peyton Reed	2015	1706431393888-ant_man_ver5.jpg
HARRY POTTER	lorem	lorem	2003	1706379014466-poster-for-movie.png
IRON MAN	superhero	awe	1999	1706375571077-ironman1.jpg
IRON MAN 2	superhero	-	2015	1706377521811-ironman2.jpg

### ฟังก์ชัน

```
async function getMovieList() {
  var Query;
  var pool = mysql.createPool(config);

  return new Promise((resolve, reject) => {
    //Query = `SELECT * FROM movies WHERE warehouse_status = 1 ORDER BY release_date`;
    Query = `SELECT * FROM movies`;

    pool.query(Query, function (error, results, fields) {
      if (error) throw error;

      if (results.length > 0) {
        pool.end();
        return resolve(results);
      } else {
        pool.end();
        return resolve({
          statusCode: 404,
          returnCode: 11,
          message: 'No movie found',
        });
      }
    });
  });
}
```

getMovieList ทำการรับค่าจากผู้ใช้และส่งข้อมูลนั้นทั้งหมดไปยังผู้ใช้

นาย จักรภัทร อินผูก 66543210005-5

```
async function getMovieSearch(search_text) {  
  var Query;  
  var pool = mysql.createPool(config);  
  
  return new Promise((resolve, reject) => {  
    Query = `SELECT * FROM movies WHERE title LIKE '%${search_text}%'`;  
  
    pool.query(Query, function (error, results, fields) {  
      if (error) throw error;  
  
      if (results.length > 0) {  
        pool.end();  
        return resolve({  
          statusCode: 200,  
          returnCode: 1,  
          data: results,  
        });  
      } else {  
        pool.end();  
        return resolve({  
          statusCode: 404,  
          returnCode: 11,  
          message: 'No movie found',  
        });  
      }  
    });  
  });  
}
```

getMovieSearch ทำการรับชื่อหนังที่ผู้ใช้ต้องการค้นหาและส่งคำสั่งค้นหาไปยัง database หลังจากนั้น ส่งข้อมูลหนังที่ค้นหามาได้ส่งคืนไปยังผู้ใช้งาน

```
async function postMovie(p_title,p_genre,p_director,p_release_year,p_imageName) {  
  var Query;  
  var pool = mysql.createPool(config);  
  
  return new Promise((resolve, reject) => {  
    //Query = `SELECT * FROM movies WHERE title LIKE '%${search_text}%'`;  
  
    var post = {  
      title: p_title,  
      genre: p_genre,  
      director: p_director,  
      release_year: p_release_year,  
      Image_name: p_imageName  
    };  
  
    console.log('post is: ', post);  
  
    Query = 'INSERT INTO movies SET ?';  
    pool.query(Query, post, function (error, results, fields) {  
      //pool.query(Query, function (error, results, fields){  
      console.log(results);  
      if (error) {  
        console.log('error_code_msg: '+error.code+ ' : '+error.sqlMessage);  
        pool.end();  
        return resolve({  
          statusCode: 405,  
          returnCode: 9,  
          error_code: error.code+ ' : '+error.sqlMessage,  
        });  
      } else {  
        if (results.affectedRows > 0) {  
          pool.end();  
          return resolve({  
            statusCode: 200,  
            returnCode: 1,  
            message: 'Movie list was inserted',  
          });  
        }  
      }  
    });  
  });  
}
```

postMovie ทำการรับข้อมูลต่างๆที่ผู้ใช้ต้องการจะเพิ่มเข้าไปยังเว็บ นำมาเข้ามาเก็บยัง database เช่น ชื่อ หนัง แนวหนัง ชื่อผู้กำกับ ปีที่ออก ชื่อไฟล์รูปภาพ

นาย จักรภัทร อินผูก 66543210005-5

## อ้างอิง

<https://react-bootstrap.netlify.app/docs/components/navbar/>

<https://medium.com/today-i-learned-chai/how-to-upload-files-with-node-js-hapi-js-c788b5c8586a>

<https://getbootstrap.com/docs/5.0/components/modal/>

<https://codepen.io/sukhsingh2729/pen/rRxEGW>

<https://www.youtube.com/watch?v=IJYrrQW775q>