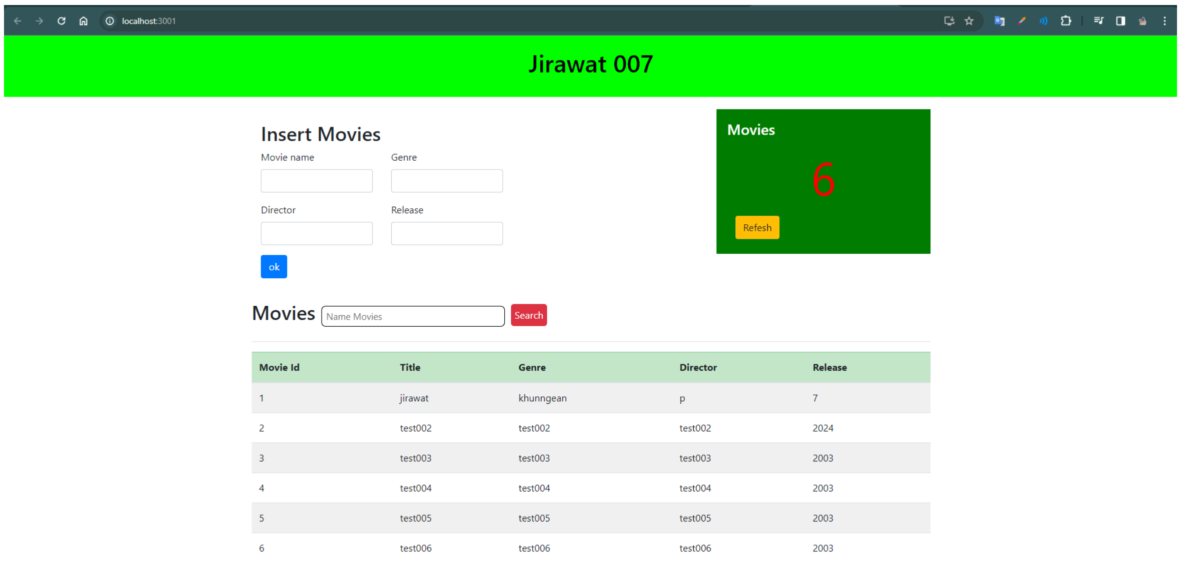
ENGSE203-2 Mid-term

Home page



localhost:3001

Jirawat 007

Insert Movies

Movie name | Genre

Director | Release

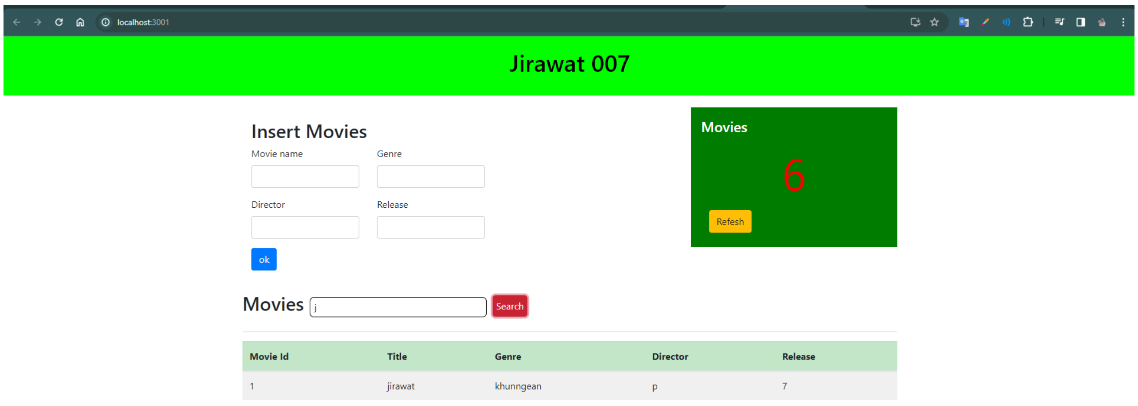
ok

Movies 6 Refesh

Movies [Name Movies] Search

Movie Id	Title	Genre	Director	Release
1	jirawat	khunngean	p	7
2	test002	test002	test002	2024
3	test003	test003	test003	2003
4	test004	test004	test004	2003
5	test005	test005	test005	2003
6	test006	test006	test006	2003

Home page show some name



localhost:3001

Jirawat 007

Insert Movies

Movie name | Genre

Director | Release

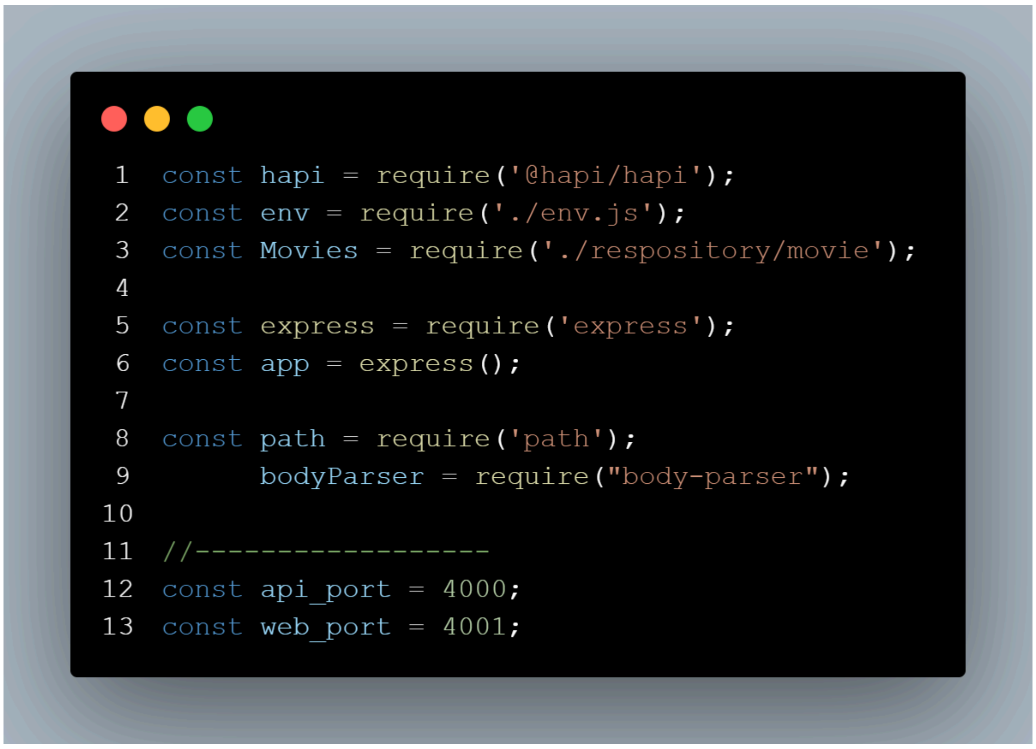
ok

Movies 6 Refesh

Movies [j] Search

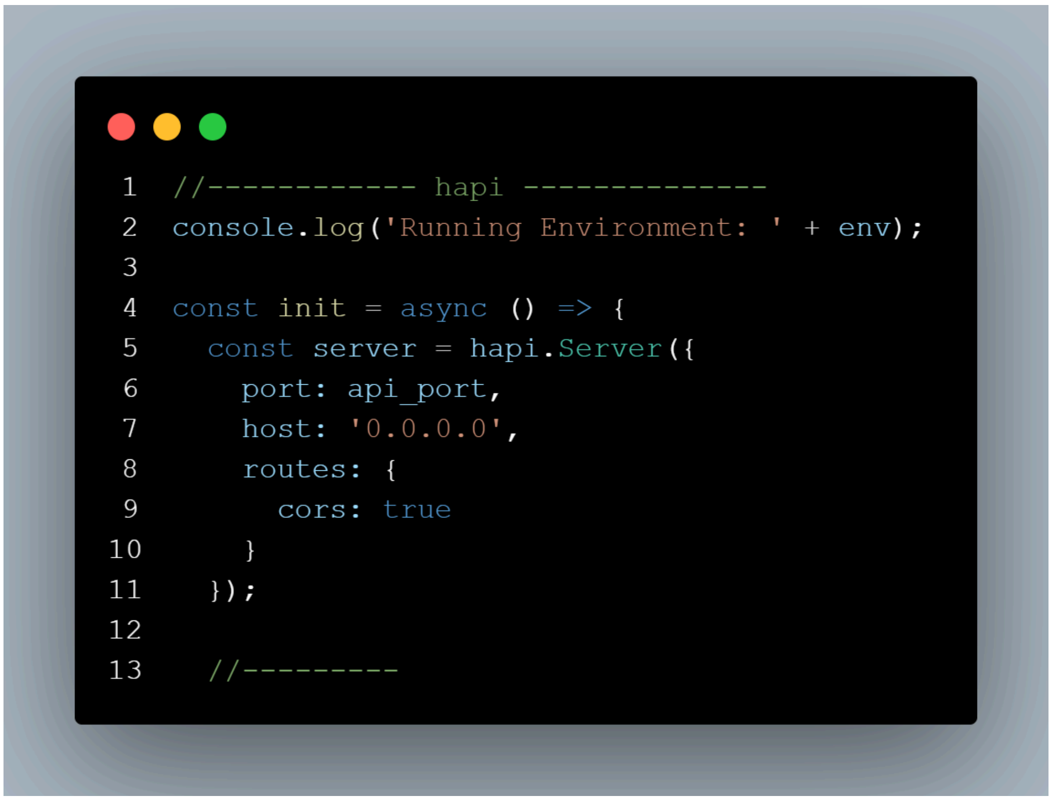
Movie Id	Title	Genre	Director	Release
1	jirawat	khunngean	p	7

ap/index.js

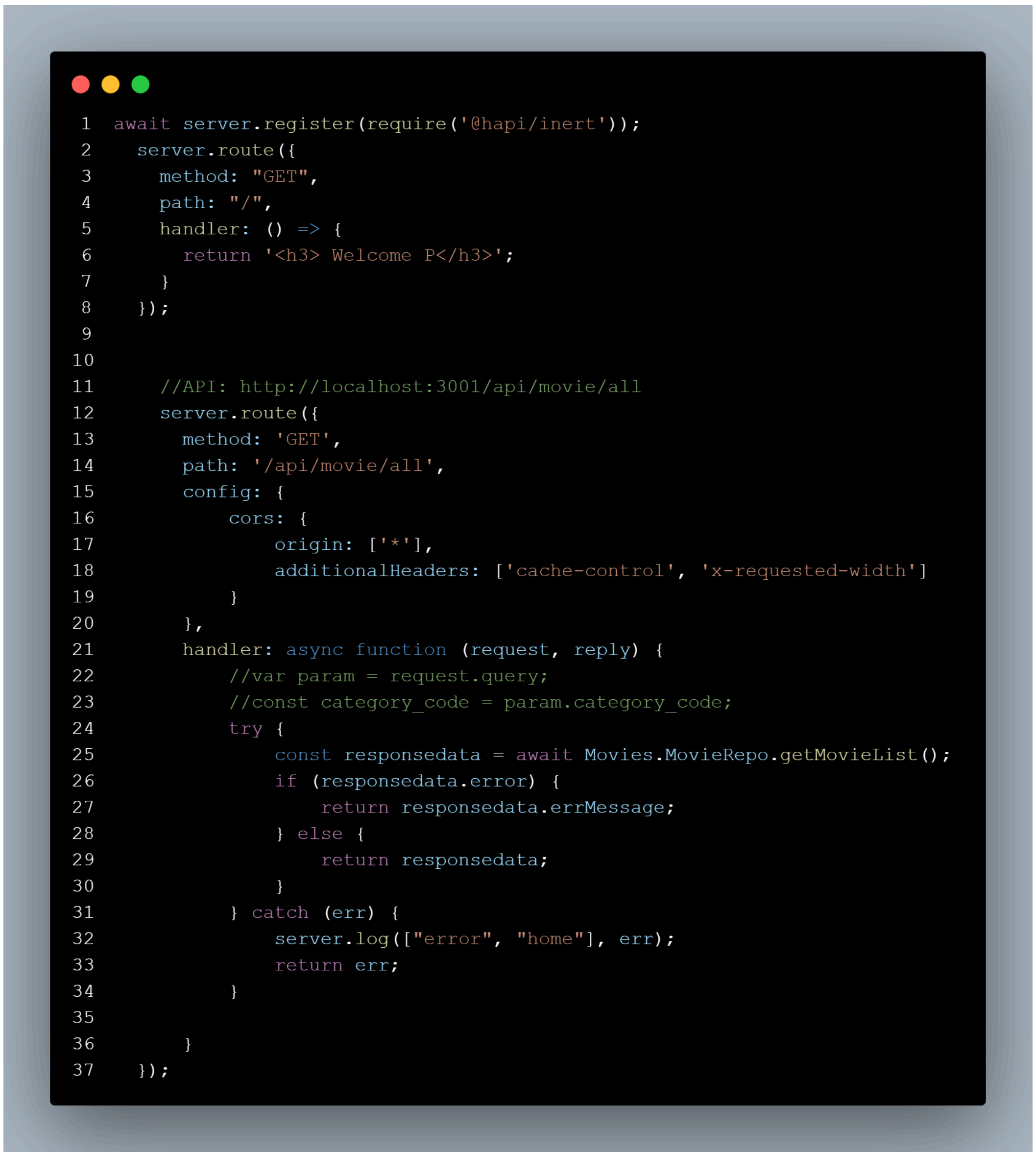


```
const hapi = require('@hapi/hapi');
const env = require('./env.js');
const Movies = require('./respository/movie');

const express = require('express');
const app = express();

const path = require('path');
      bodyParser = require("body-parser");

//------------------
const api_port = 4000;
const web_port = 4001;
```

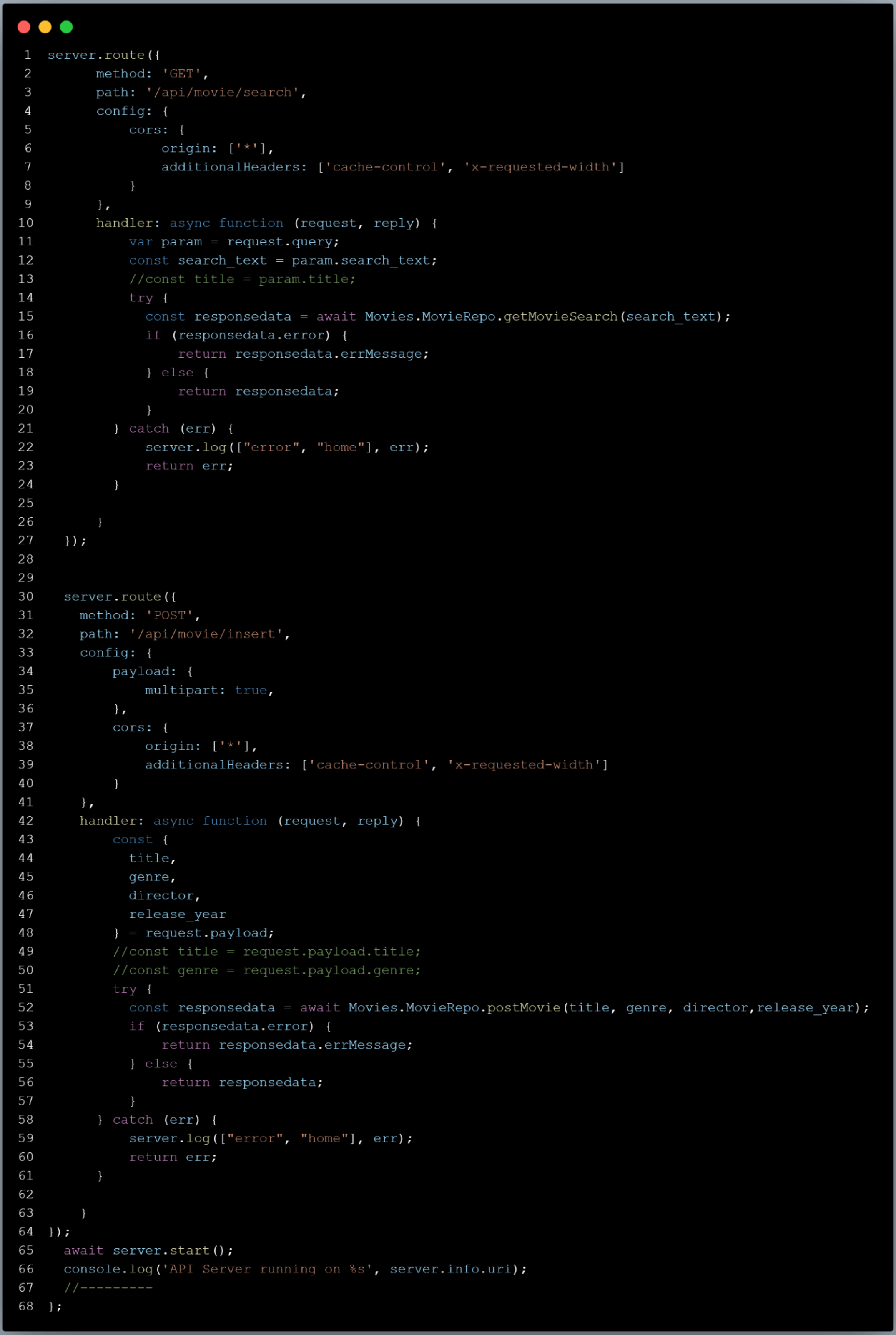
เฟรมเวิร์กและพอร์ต



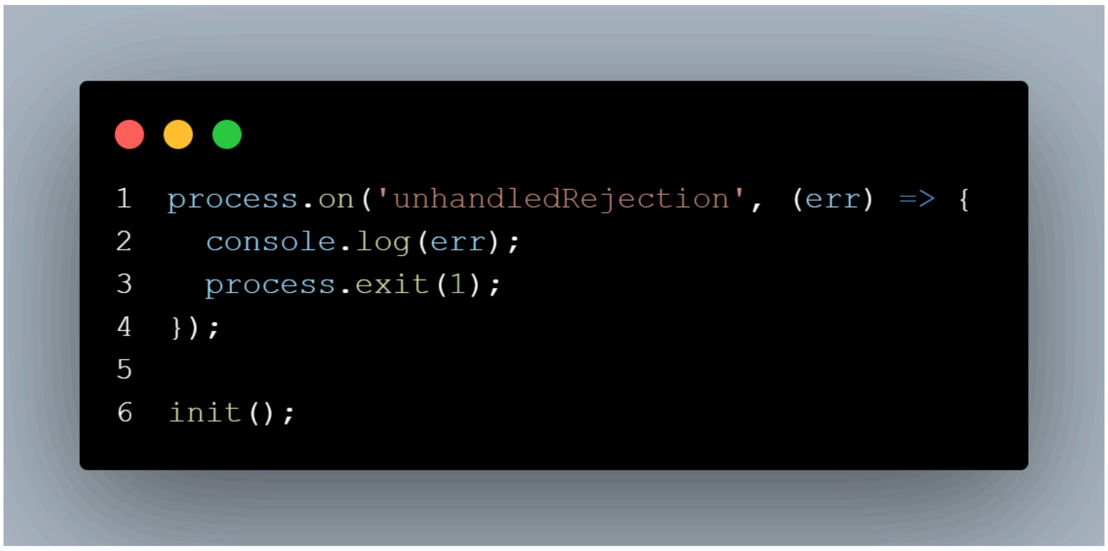
```
//------------ hapi --------------
console.log('Running Environment: ' + env);

const init = async () => {
  const server = hapi.Server({
    port: api_port,
    host: '0.0.0.0',
    routes: {
      cors: true
    }
  });

  //---------
```



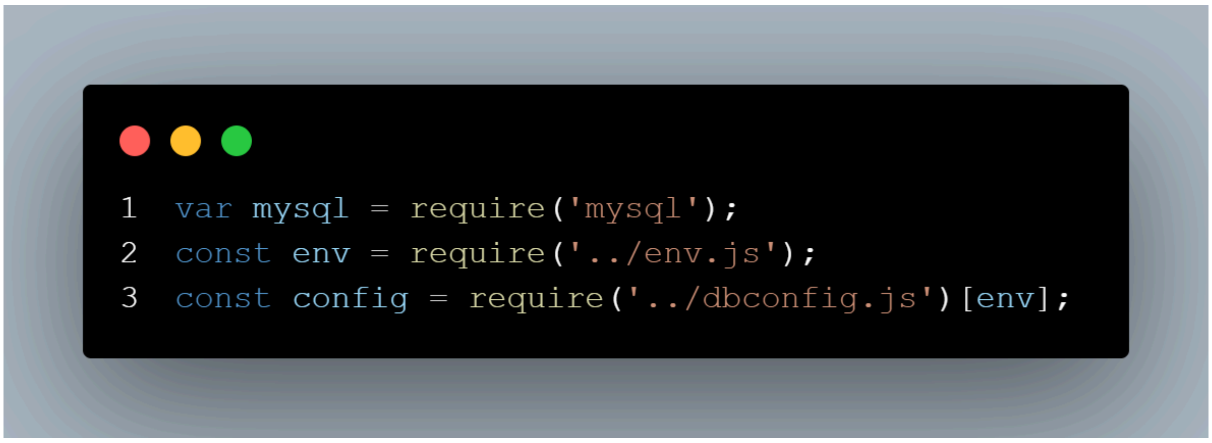
```
await server.register(require('@hapi/inert'));
  server.route({
    method: "GET",
    path: "/",
    handler: () => {
      return '<h3> Welcome P</h3>';
    }
  });


    //API: http://localhost:3001/api/movie/all
    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 responsedata = await Movies.MovieRepo.getMovieList();
              if (responsedata.error) {
                  return responsedata.errMessage;
              } else {
                  return responsedata;
              }
          } catch (err) {
              server.log(["error", "home"], err);
              return err;
          }

      }
  });
```

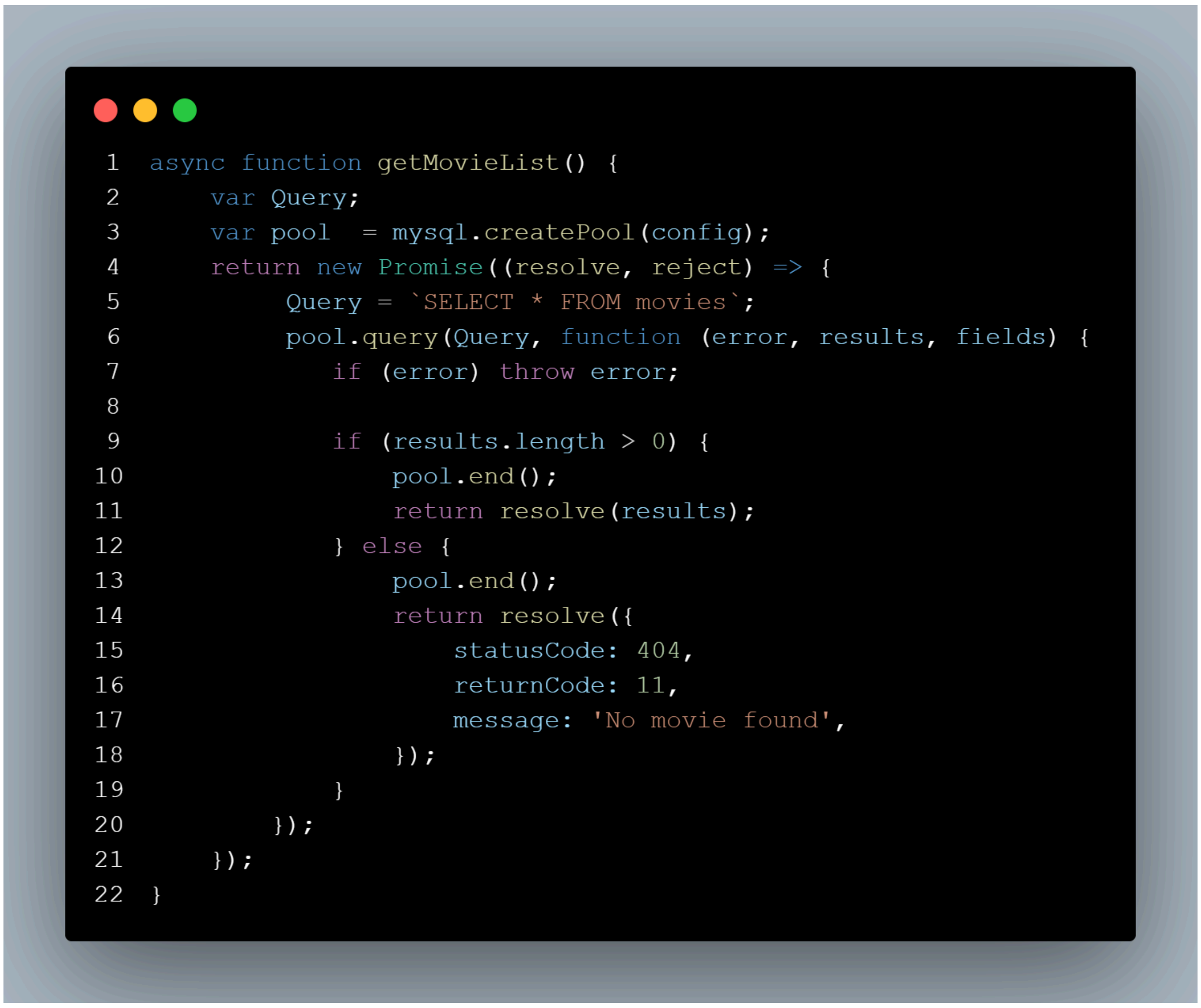


```
server.route({
      method: 'GET',
      path: '/api/movie/search',
      config: {
          cors: {
              origin: ['*'],
              additionalHeaders: ['cache-control', 'x-requested-width']
          }
      },
      handler: async function (request, reply) {
          var param = request.query;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

      }
  });


  server.route({
    method: 'POST',
    path: '/api/movie/insert',
    config: {
        payload: {
            multipart: true,
        },
        cors: {
            origin: ['*'],
            additionalHeaders: ['cache-control', 'x-requested-width']
        }
    },
    handler: async function (request, reply) {
        const {
          title,
          genre,
          director,
          release_year
        } = request.payload;
        //const title = request.payload.title;
        //const genre = request.payload.genre;
        try {
          const responsedata = await Movies.MovieRepo.postMovie(title, genre, director,release_year);
          if (responsedata.error) {
              return responsedata.errMessage;
          } else {
              return responsedata;
          }
      } catch (err) {
          server.log(["error", "home"], err);
          return err;
      }

    }
});
  await server.start();
  console.log('API Server running on %s', server.info.uri);
  //---------
};
```

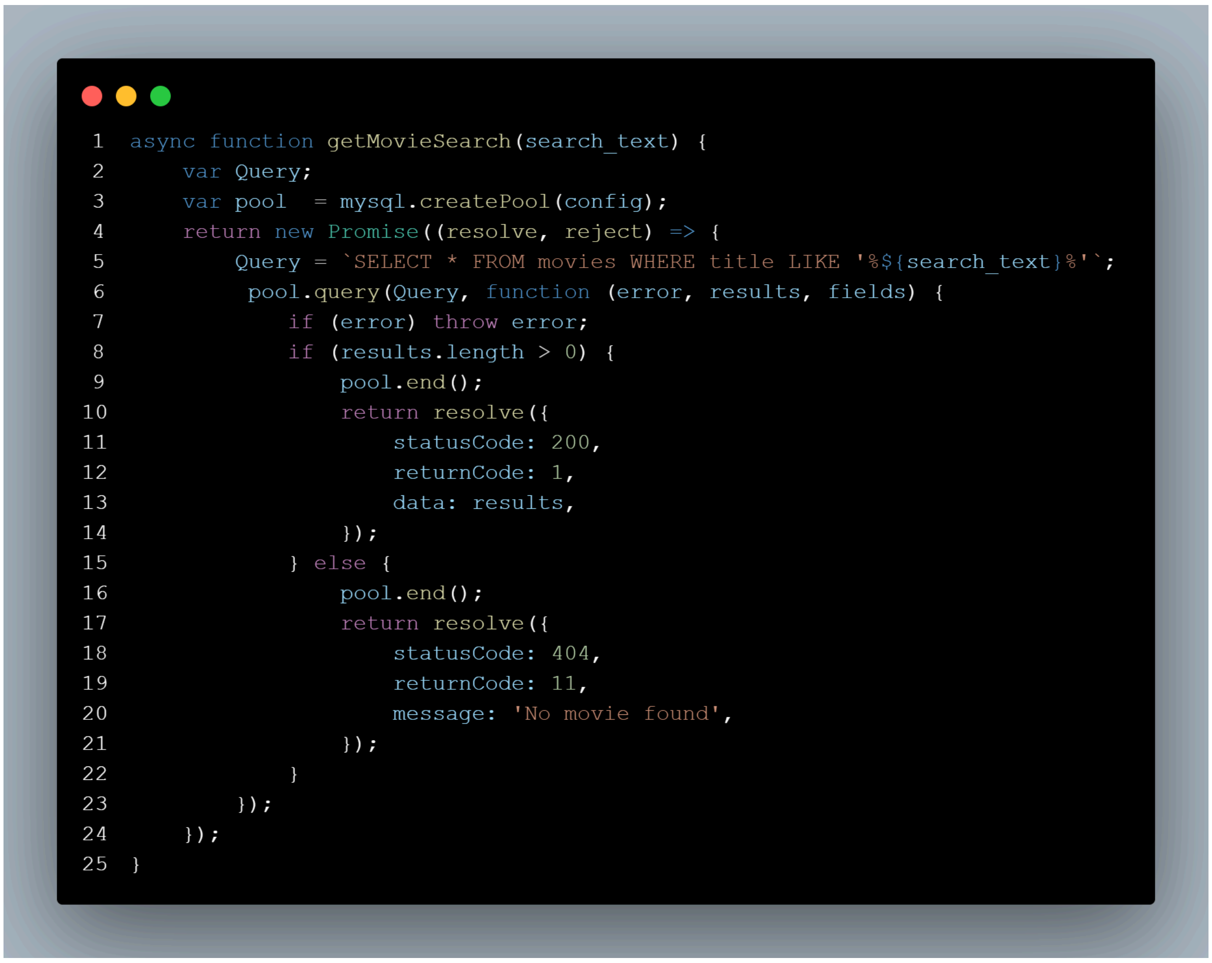

```
process.on('unhandledRejection', (err) => {
  console.log(err);
  process.exit(1);
});

init();
```

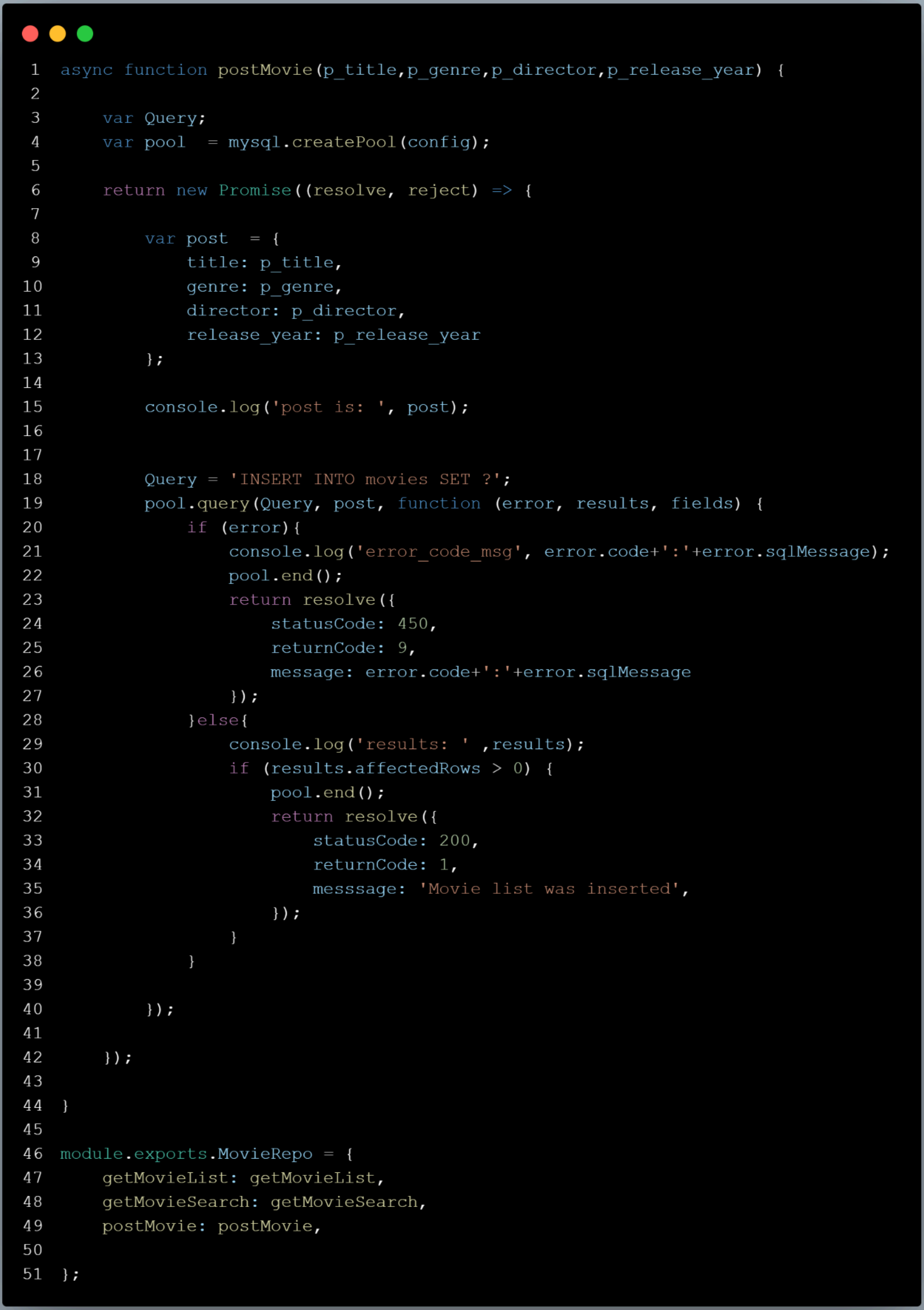
api/respository/movie.js


```
var mysql = require('mysql');
const env = require('../env.js');
const config = require('../dbconfig.js')[env];
```



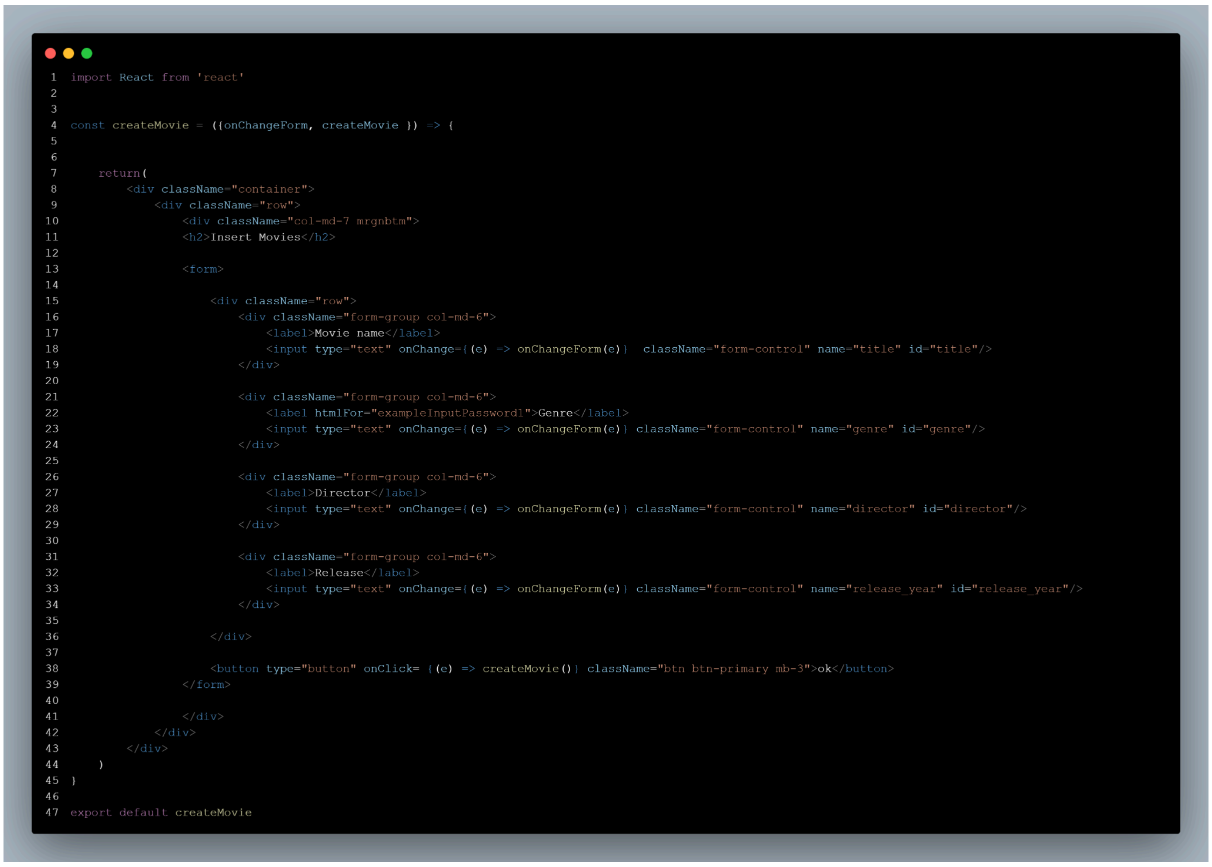
```
async function getMovieList() {
    var Query;
    var pool  = mysql.createPool(config);
    return new Promise((resolve, reject) => {
         Query = `SELECT * FROM movies`;
         pool.query(Query, function (error, results, fields) {
            if (error) throw error;

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


```
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',
                });
            }
        });
    });
}
```



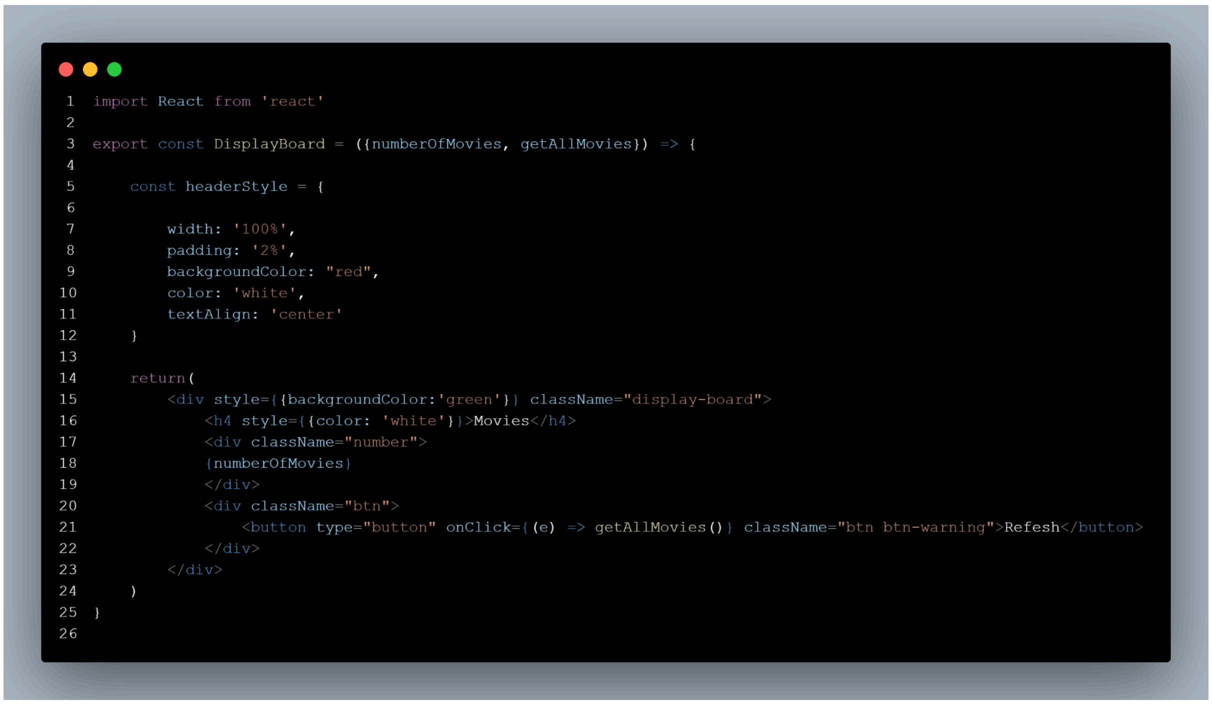
```javascript
async function postMovie(p_title,p_genre,p_director,p_release_year) {

    var Query;
    var pool  = mysql.createPool(config);

    return new Promise((resolve, reject) => {

        var post  = {
            title: p_title,
            genre: p_genre,
            director: p_director,
            release_year: p_release_year
        };

        console.log('post is: ', post);


        Query = 'INSERT INTO movies SET ?';
        pool.query(Query, post, function (error, results, fields) {
            if (error){
                console.log('error_code_msg', error.code+':'+error.sqlMessage);
                pool.end();
                return resolve({
                    statusCode: 450,
                    returnCode: 9,
                    message: error.code+':'+error.sqlMessage
                });
            }else{
                console.log('results: ' ,results);
                if (results.affectedRows > 0) {
                    pool.end();
                    return resolve({
                        statusCode: 200,
                        returnCode: 1,
                        messsage: 'Movie list was inserted',
                    });
                }
            }

        });

    });

}

module.exports.MovieRepo = {
    getMovieList: getMovieList,
    getMovieSearch: getMovieSearch,
    postMovie: postMovie,

};
```

my-app/src/components/CreateMovie.js
เป็นหน้าการเพิ่มข้อมูลว่าเพิ่มอะไรไปบ้าง



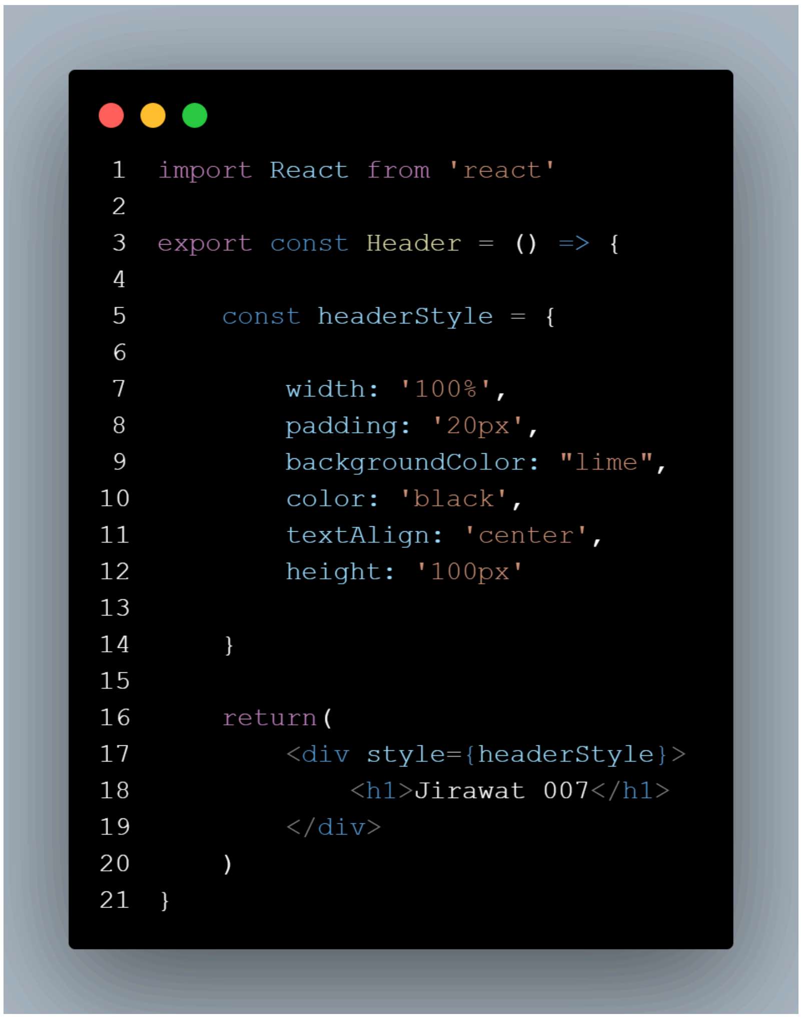
```
import React from 'react'


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


    return(
        <div className="container">
            <div className="row">
                <div className="col-md-7 mrgnbtm">
                <h2>Insert Movies</h2>

                <form>

                    <div className="row">
                        <div className="form-group col-md-6">
                            <label>Movie name</label>
                            <input type="text" onChange={(e) => onChangeForm(e)}  className="form-control" name="title" id="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"/>
                        </div>

                        <div className="form-group col-md-6">
                            <label>Director</label>
                            <input type="text" onChange={(e) => onChangeForm(e)} className="form-control" name="director" id="director"/>
                        </div>

                        <div className="form-group col-md-6">
                            <label>Release</label>
                            <input type="text" onChange={(e) => onChangeForm(e)} className="form-control" name="release_year" id="release_year"/>
                        </div>

                    </div>

                    <button type="button" onClick= {(e) => createMovie()} className="btn btn-primary mb-3">ok</button>
                </form>

                </div>
            </div>
        </div>
    )
}

export default createMovie
```

my-app/src/components/DisplayBoard.js



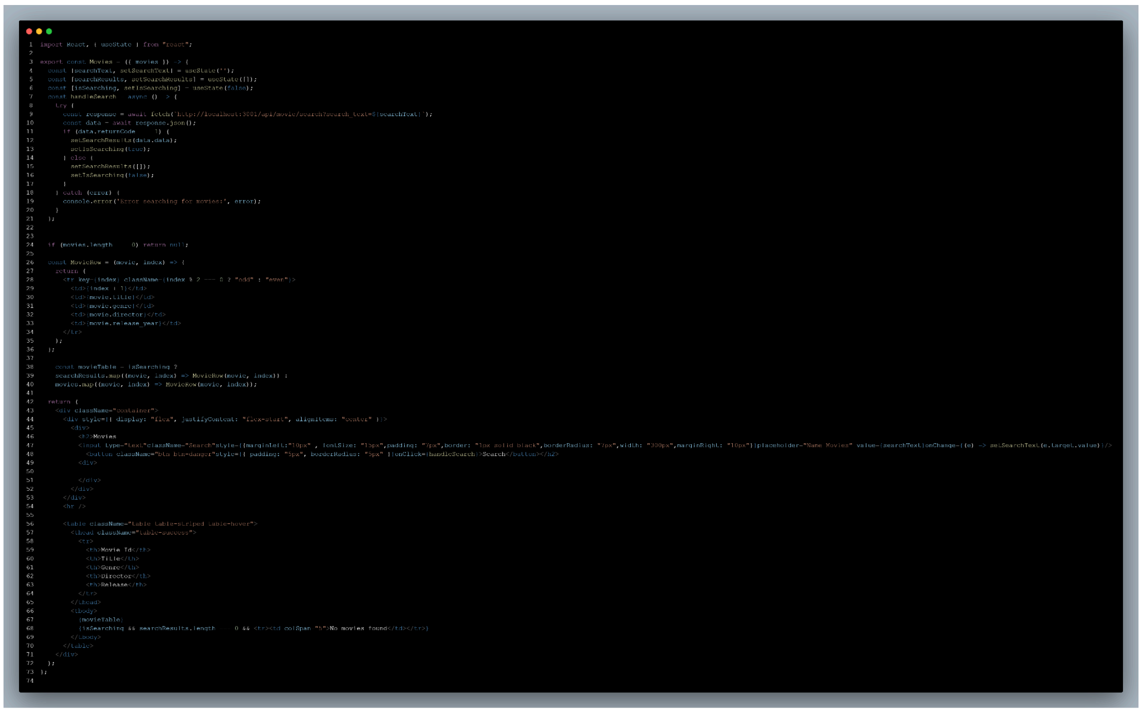
```
import React from 'react'

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</h4>
            <div className="number">
            {numberOfMovies}
            </div>
            <div className="btn">
                <button type="button" onClick={(e) => getAllMovies()} className="btn btn-warning">Refesh</button>
            </div>
        </div>
    )
}

```

my-app/src/components/Header.js



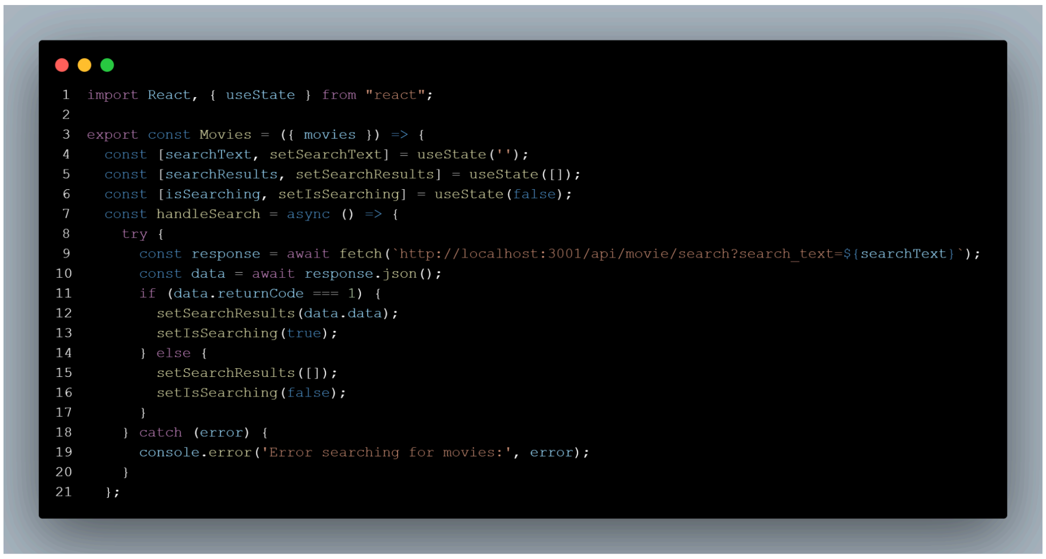
```javascript
import React from 'react'

export const Header = () => {

    const headerStyle = {

        width: '100%',
        padding: '20px',
        backgroundColor: "lime",
        color: 'black',
        textAlign: 'center',
        height: '100px'

    }

    return(
        <div style={headerStyle}>
            <h1>Jirawat 007</h1>
        </div>
    )
}
```

my-app/src/components/Movie.js

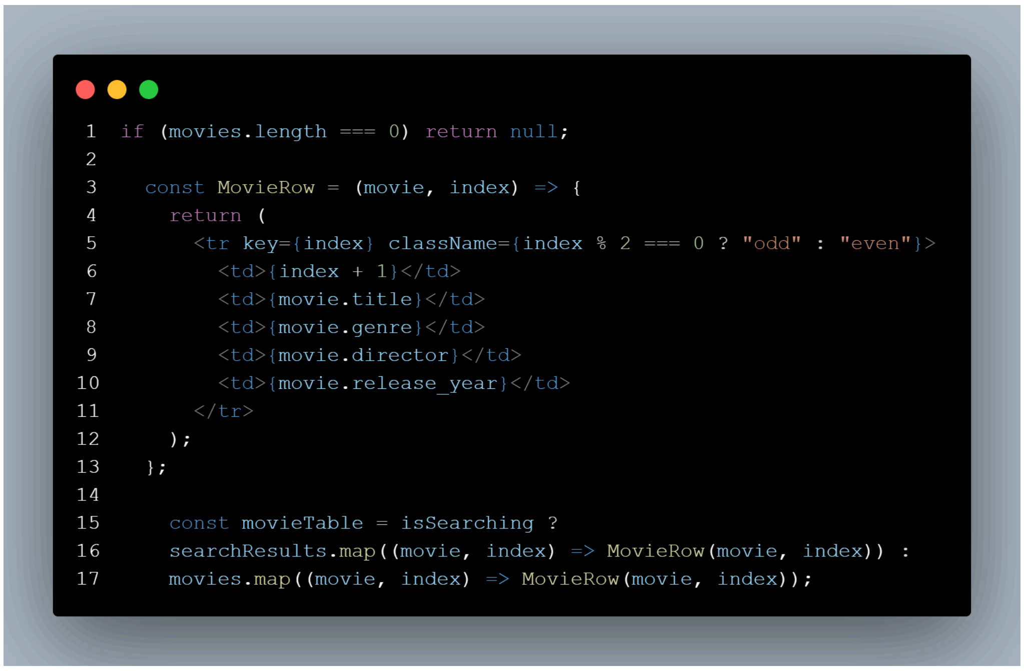


```javascript
import React, { useState } from "react";

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();
      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));

  return (
    <div className="container">
      <div style={{ display: "flex", justifyContent: "flex-start", alignItems: "center" }}>
        <div>
          <h2>Movies
          <input type="text"className="Search"style={{marginLeft:"10px" , fontSize: "15px",padding: "7px",border: "1px solid black",borderRadius: "7px",width: "300px",marginRight: "10px"}}placeholder="Name Movies" value={searchText}onChange={(e) => setSearchText(e.target.value)}/>
            <button className="btn btn-danger"style={{ padding: "5px", borderRadius: "5px" }}onClick={handleSearch}>Search</button></h2>
          <div>

          </div>
        </div>
      </div>
      <hr />

      <table className="table table-striped table-hover">
        <thead className="table-success">
          <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>
  );
};

```

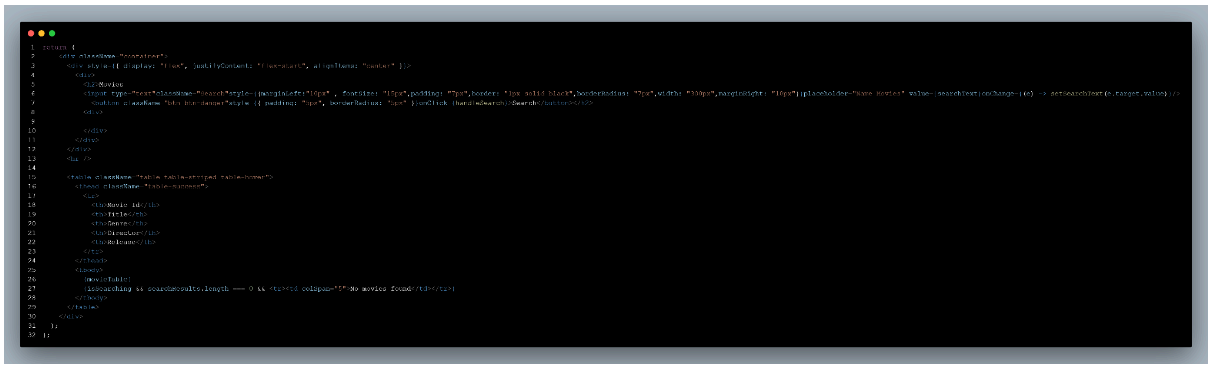
my-app/src/components/Movies.js



```
import React, { useState } from "react";

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



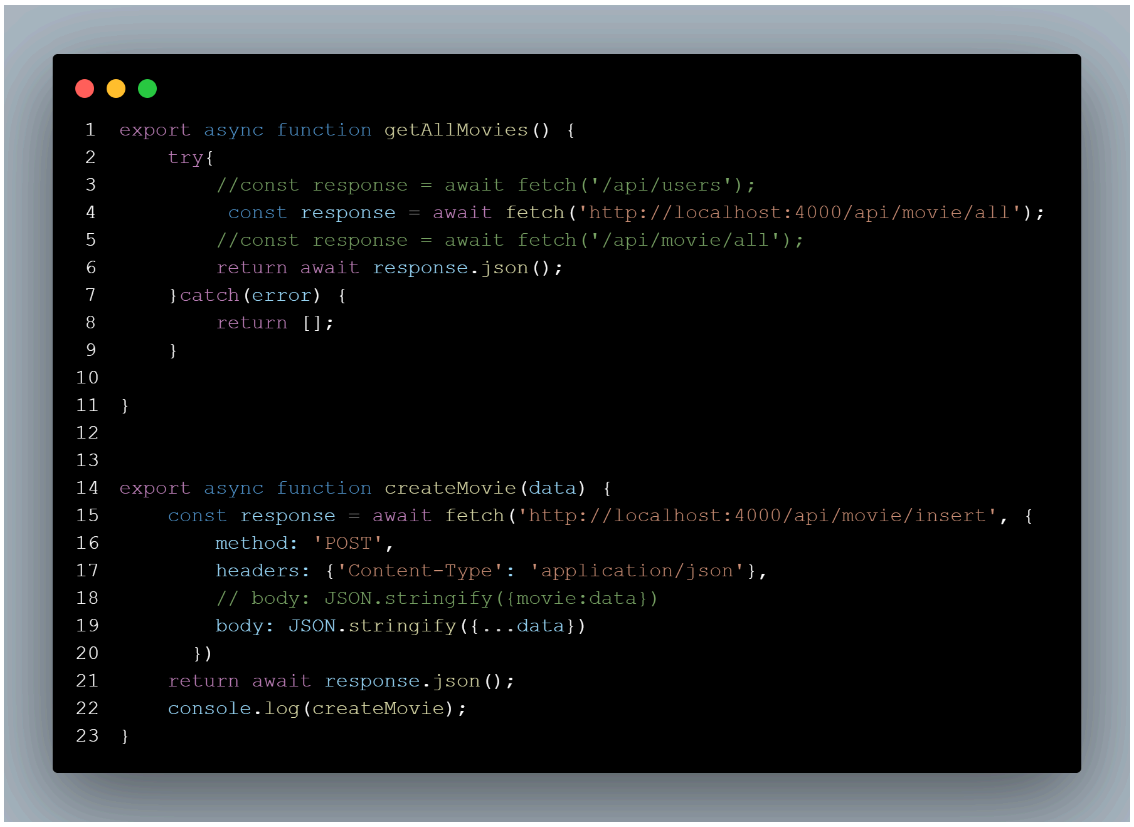
```
if (movies.length === 0) return null;

 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));
```



```
return (
  <div className="container">
   <div style={{ display: "flex", justifyContent: "flex-start", alignItems: "center" }}>
    <div>
     <h2>Movies
     <input type="text"className="Search"style={{marginLeft:"10px" , fontSize: "15px",padding: "7px",border: "1px solid black",borderRadius: "7px",width: "300px",marginRight: "10px"}}placeholder="Name Movies" value={searchText}onChange={(e) => setSearchText(e.target.value)}/>
      <button className="btn btn-danger"style={{ padding: "5px", borderRadius: "5px" }}onClick={handleSearch}>Search</button></h2>
     <div>

     </div>
    </div>
   </div>
   <hr />

   <table className="table table-striped table-hover">
    <thead className="table-success">
     <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>
 );
};
```

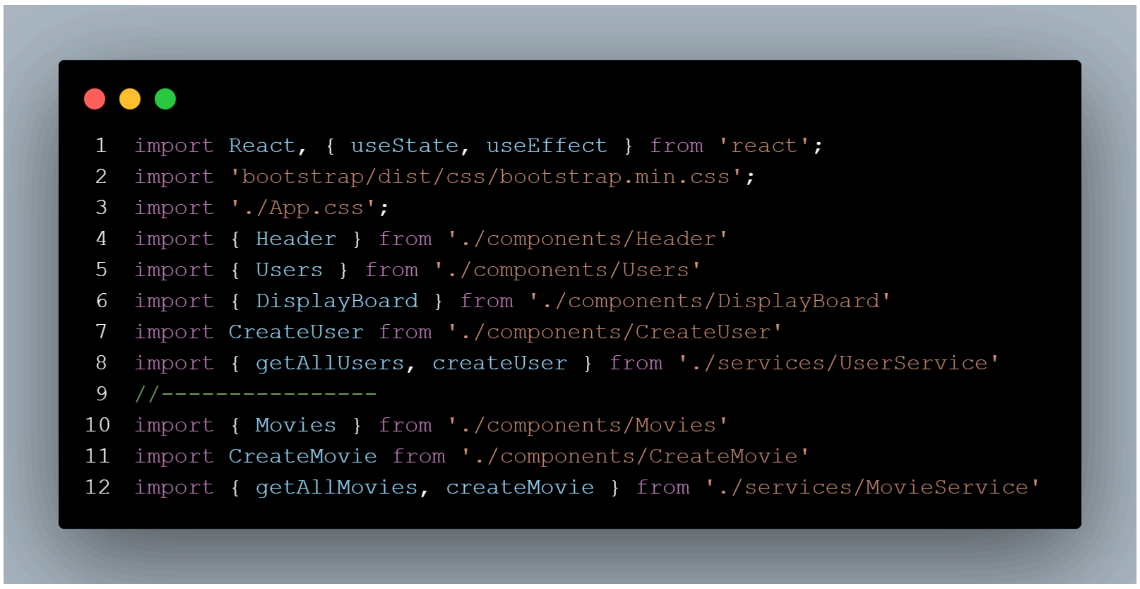
my-app/src/services/MovieService.js

Change this ({movie:data}) -> body: JSON.stringify({...data}) || PORT 3001 -> 4000 -> 4000 is PORT backend

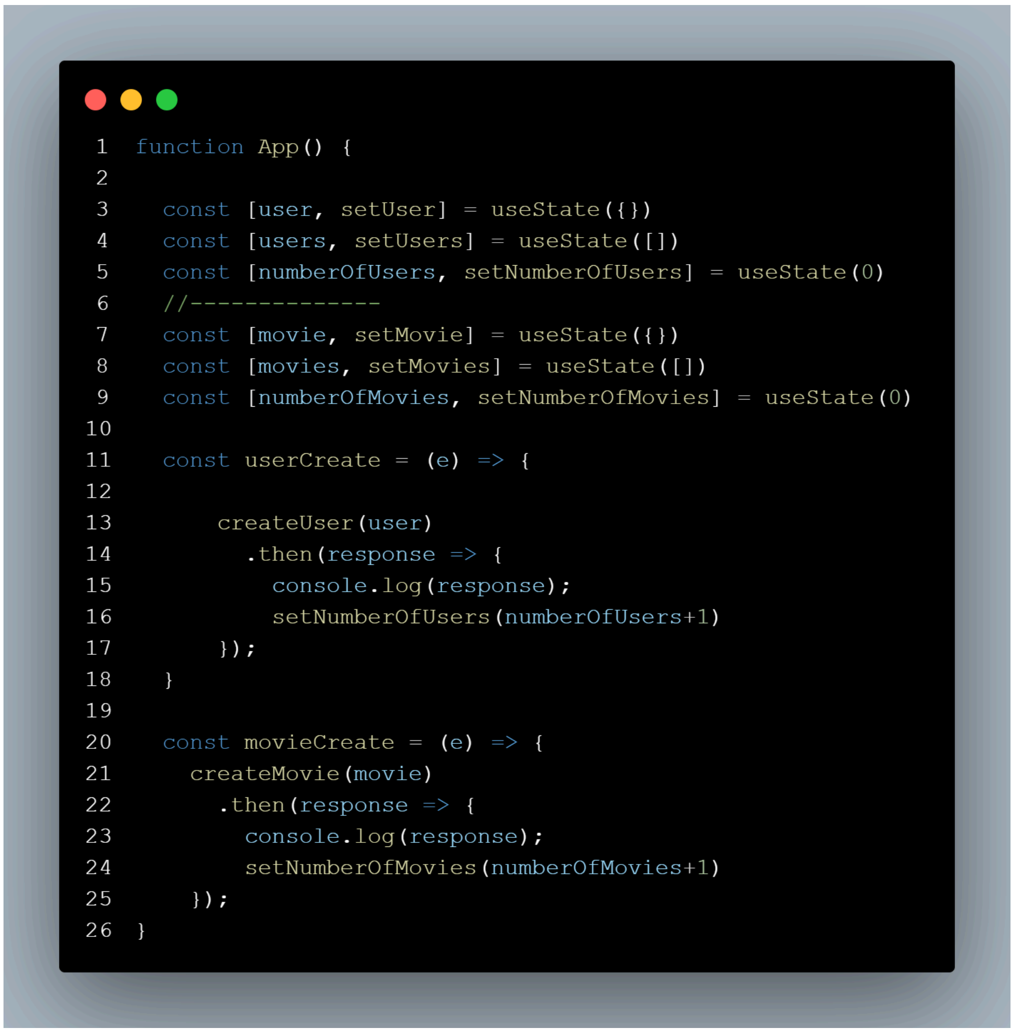


```
export async function getAllMovies() {
    try{
        //const response = await fetch('/api/users');
         const response = await fetch('http://localhost:4000/api/movie/all');
        //const response = await fetch('/api/movie/all');
        return await response.json();
    }catch(error) {
        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({movie:data})
        body: JSON.stringify({...data})
      })
    return await response.json();
    console.log(createMovie);
}
```

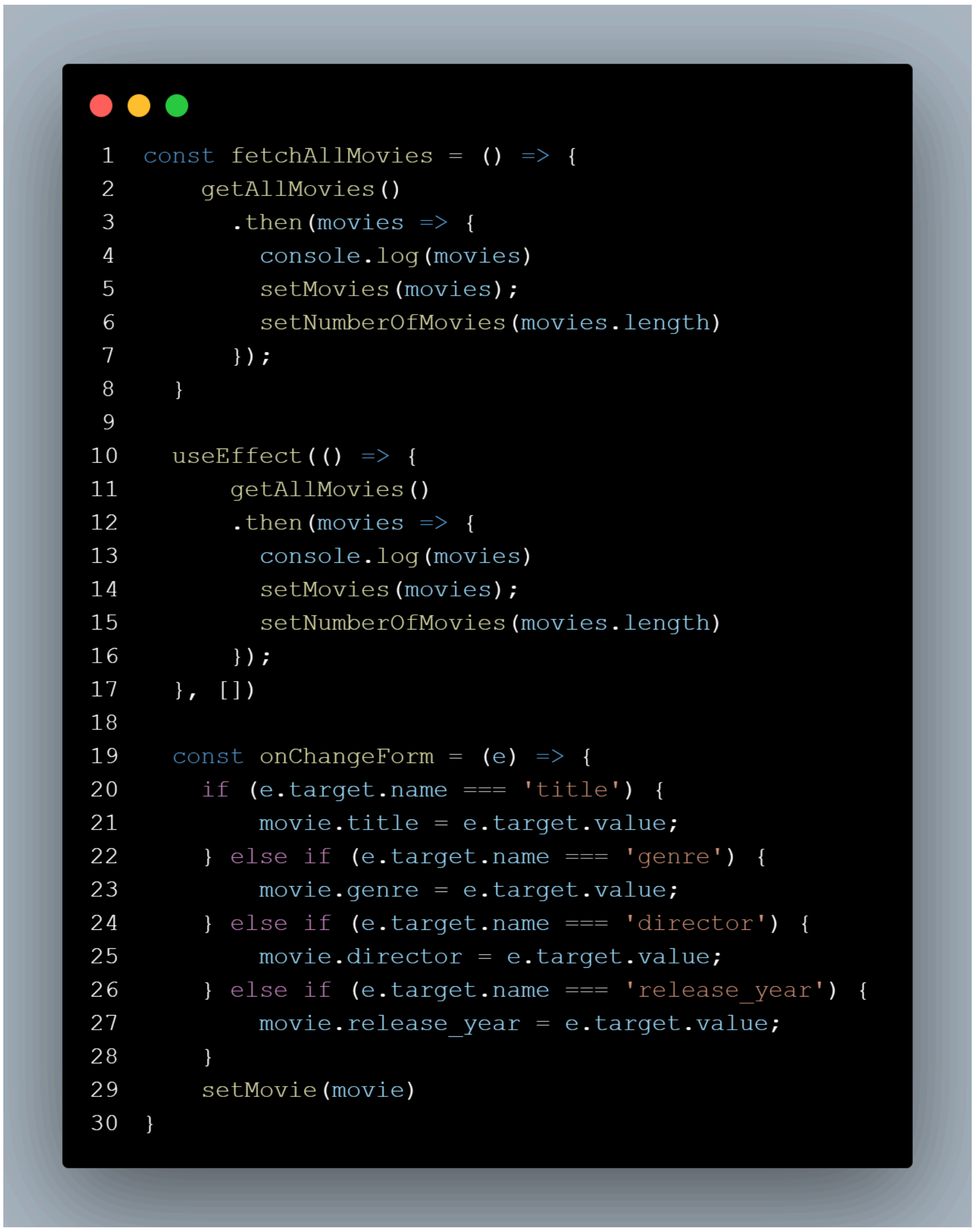
my-app/App.js



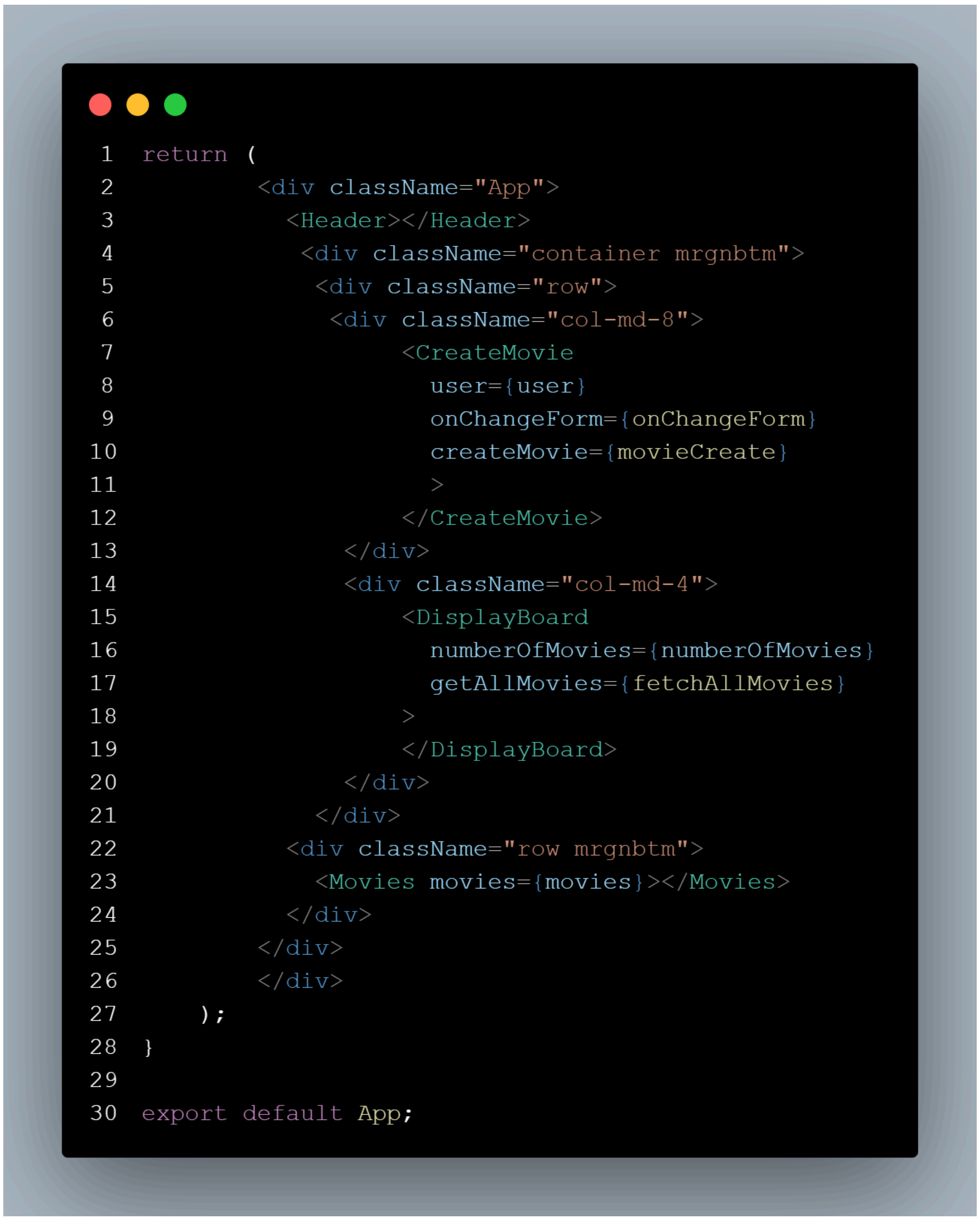
```
import React, { useState, useEffect } from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';
import './App.css';
import { Header } from './components/Header'
import { Users } from './components/Users'
import { DisplayBoard } from './components/DisplayBoard'
import CreateUser from './components/CreateUser'
import { getAllUsers, createUser } from './services/UserService'
//----------------
import { Movies } from './components/Movies'
import CreateMovie from './components/CreateMovie'
import { getAllMovies, createMovie } from './services/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)
      });
  }

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



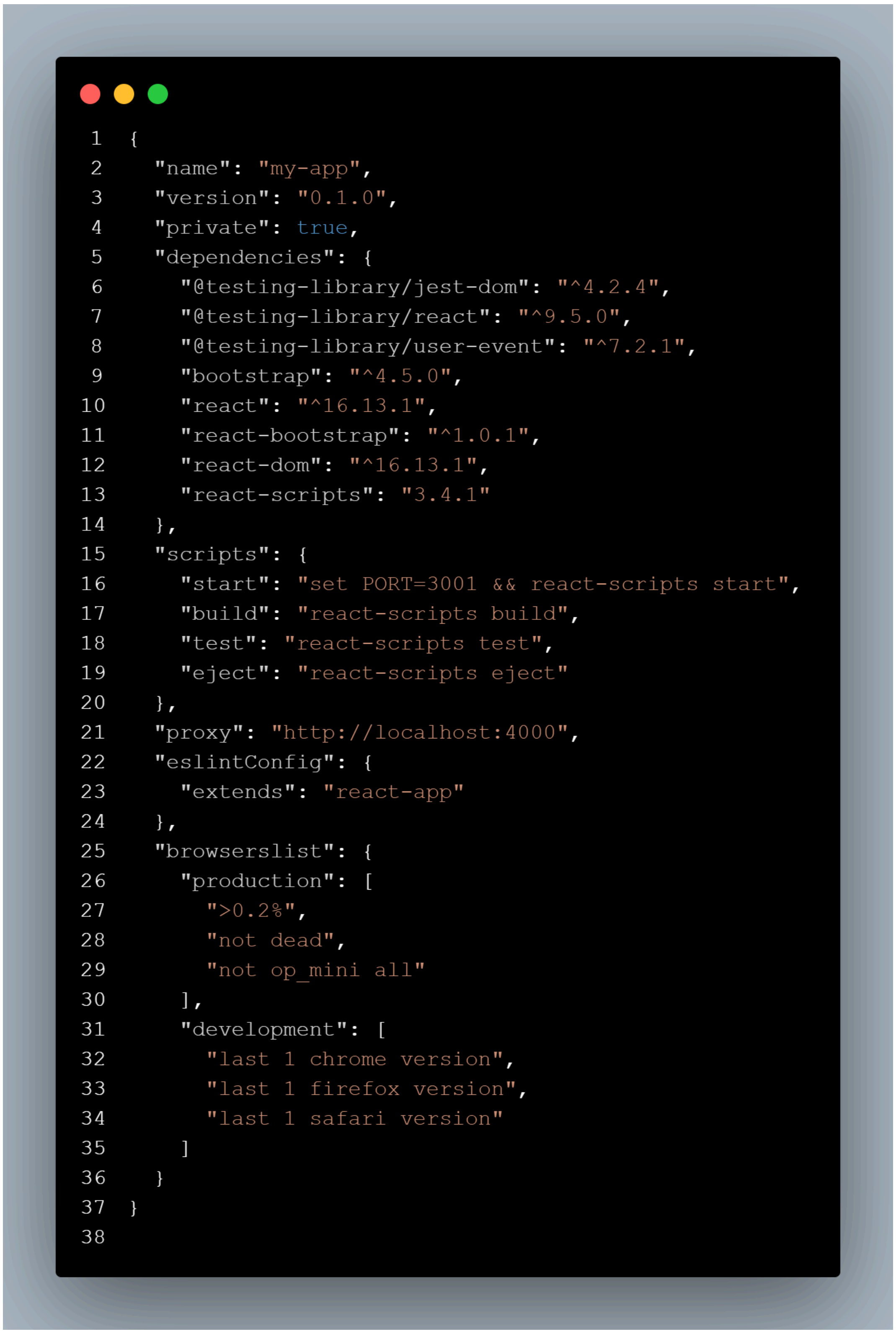
```javascript
const fetchAllMovies = () => {
    getAllMovies()
      .then(movies => {
        console.log(movies)
        setMovies(movies);
        setNumberOfMovies(movies.length)
      });
  }

  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)
}
```



```
return (
        <div className="App">
          <Header></Header>
           <div className="container mrgnbtm">
            <div className="row">
             <div className="col-md-8">
                  <CreateMovie
                    user={user}
                    onChangeForm={onChangeForm}
                    createMovie={movieCreate}
                    >
                  </CreateMovie>
              </div>
              <div className="col-md-4">
                  <DisplayBoard
                    numberOfMovies={numberOfMovies}
                    getAllMovies={fetchAllMovies}
                  >
                  </DisplayBoard>
              </div>
            </div>
          <div className="row mrgnbtm">
            <Movies movies={movies}></Movies>
          </div>
        </div>
        </div>
    );
}

export default App;
```

my-app ก่อน npm start
$env:NODE_OPTIONS = "--openssl-legacy-provider" (windows)

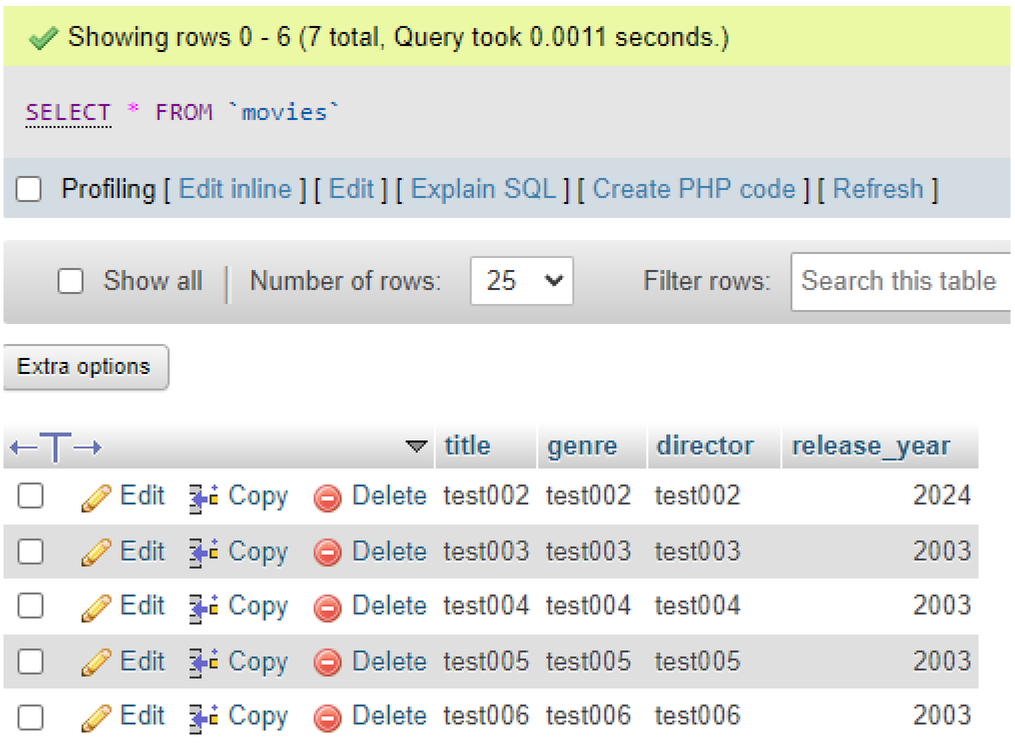
my-app/package.json



```json
{
  "name": "my-app",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^4.2.4",
    "@testing-library/react": "^9.5.0",
    "@testing-library/user-event": "^7.2.1",
    "bootstrap": "^4.5.0",
    "react": "^16.13.1",
    "react-bootstrap": "^1.0.1",
    "react-dom": "^16.13.1",
    "react-scripts": "3.4.1"
  },
  "scripts": {
    "start": "set PORT=3001 && react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "proxy": "http://localhost:4000",
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}

```

"start": "set PORT=3001 && react-scripts start" (windows)

DATABASES
movies
title | genre | director | releases_year



Showing rows 0 - 6 (7 total, Query took 0.0011 seconds.)

```
SELECT * FROM `movies`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
Extra options

	title	genre	director	release_year
Edit Copy Delete	test002	test002	test002	2024
Edit Copy Delete	test003	test003	test003	2003
Edit Copy Delete	test004	test004	test004	2003
Edit Copy Delete	test005	test005	test005	2003
Edit Copy Delete	test006	test006	test006	2003

จิรวัฒน์ ขุนเงิน 6654321007-1
ศุภวิชญ์ กะตะศิลา 66543210032-9