# App Design with React Week 2 Coding Assignment

**Points possible:** 70

|  |  |  |
| --- | --- | --- |
| Category | Criteria | % of Grade |
| Functionality | Does the code work? | 25 |
| Organization | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| Creativity | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| Completeness | All requirements of the assignment are complete. | 25 |

**Instructions:** In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week’s assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week’s repository to this document where instructed and submit this document to your instructor when complete.

**Coding Steps:**

1. Using what you’ve learned this week, create a page of an application that enables users to vote and leave reviews on movies.
2. You should include at least the following components:
   1. MovieList – a container for all the Movie components and their data.
   2. Movie – a component that represents movie data (i.e. image, synopsis, rating, etc…)
   3. Stars – a one to five-star rating component that allows users to rate something (movies in this case, but remember that components are reusable, so you could use it elsewhere!)
   4. ReviewList – a container inside of a Movie that houses Review components.
   5. Review – A text review a user can leave on a movie.
   6. ReviewForm – a form at the bottom of a Movie component that allows users to leave reviews. When submitted, the review should be added to the movie. All this data can be stored in an array, no networking or database needed for this assignment.

**Screenshots of Code:**

**File: src\index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<meta

name="description"

content="Web site created using create-react-app"

/>

<link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

<!--

manifest.json provides metadata used when your web app is installed on a

user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<!--

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

-->

<link

rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"

integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+l"

crossorigin="anonymous"

/>

<title>My Movie App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.

To create a production bundle, use `npm run build` or `yarn build`.

-->

<script src='https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js'></script>

</body>

</html>

**File: src\App.js**

import React from 'react';

import MovieList from './Components/MovieList';

import './App.css';

function App() {

return (

<div className="MovieApp">

<h1>Student: Jorge Cruz de Leon</h1>

<h1>Movie Reviews: Marvel Universe: SpiderMan</h1>

<div className="container">

<div><MovieList />

</div>

</div>

</div>

)

}

export default App;

**File: src\index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import '../node\_modules/bootstrap/dist/css/bootstrap.css'

import './index.css';

import App from './App';

ReactDOM.render(

<App />,

document.getElementById('root')

);

**File: \src\App.css**

body {

background: rgb(167, 8, 8);

color: white

}

.MovieList {

display: flex;

justify-content: center;

margin: 0;

}

h1,

p {

text-align: center

}

h1 {

margin-top: 25px

}

p {

font-size: 1rem

}

img {

width: 200px;

padding: 5px;

}

.aveRating {

padding: 20px;

}

/\*synopsis styling\*/

.summary{

text-align: start;

padding: 20px;

}

.movie-container {

display: flex;

flex-wrap: wrap;

}

.movie-card{

border: 5px solid rgb(227,154,51);

width: 500px;

padding: 5px;

margin: 1rem;

/\* position: relative; \*/

overflow: hidden;

}

form label {

display: inline-block;

text-align: left;

padding: 5px;

}

.reviewForm {

border: 2px solid rgb(182,116,59);

display: flex;

align-self: flex-end;

justify-content: center;

}

button {

display: flex;

justify-content: center;

background-color: rgb(182,116,59);

}

/\*reviewer name field\*/

input {

display: flex;

box-sizing: border-box;

width: 400px;

height: 25px;

padding-left: 5px;

}

/\*review text field\*/

textarea {

display: flex;

box-sizing: border-box;

width: 400px;

height: 100px;

padding-left: 5px;

vertical-align: top;

}

/\*star rating formatting\*/

.rating {

display: flex;

flex-direction: row-reverse;

justify-content: center

}

.rating>input {

display: none

}

.rating>label {

position: relative;

width: 1em;

font-size: 3vw;

color: #FFD600;

cursor: pointer

}

.rating>label::before {

content: "\2605";

position: absolute;

opacity: 0

}

.rating>label:hover:before,

.rating>label:hover~label:before {

opacity: 1 !important

}

.rating>input:checked~label:before {

opacity: 1

}

.rating:hover>input:checked~label:before {

opacity: 0.4

}

.reviewList {

background-color: white;

color: black;

position: absolute;

bottom: 0;

left: 0;

right: 0;

max-height: 100%;

transform: translateY(100%);

transition: transform 0.5s ease-in-out;

}

.movie-card:hover .reviewList {

transform: translateY(0%);

}

**File: src\index.css**

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

code {

font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

monospace;

}

**File : Components\Movie.js**

import React from 'react';

import ReviewForm from './ReviewForm';

import ReviewList from './ReviewList';

import 'bootstrap/dist/css/bootstrap.css';

import Row from 'react-bootstrap/esm/Row'

import Col from 'react-bootstrap/esm/Col'

import Container from 'react-bootstrap/esm/Container';

function Movie(props) {

return (

<div className="movie-card">

<h3>Movie Title: {props.title}</h3>

<Container>

<Row>

<Col sm={5}><img src={props.imageUrl} /></Col>

<Col sm={7}>

<Row>

<span className="aveRating">{props.rating}</span>

</Row>

<Row>

<p className="summary">Synopsis: {props.synopsis}</p>

</Row>

</Col>

</Row>

</Container>

<div className="reviewForm">

<ReviewForm />

</div>

<div className="reviewList">

<ReviewList />

</div>

</div>

)

}

export default Movie;

**File : Components\Moviedata.js**

const movieData = [

{

id: 1,

title: "Spider-Man (2002)",

rating: "Placeholder: ratingAve",

synopsis: "The very first big-screen adaptation of Spider-Man. There had been made-for-TV movies and television shows based on the Marvel comic series, but Spidey did not make his theatrical debut until 2002. Directed by Sam Raimi, this version starred Tobey Maguire as awkward high school student Peter Parker and Kirsten Dunst as Mary Jane Watson, his love interest. The film documents Peter’s evolution into becoming Spider-Man, from that initial bite in the laboratory to his battle against the Green Goblin.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/spiderman-1642713480.jpeg",

altDescribe: "Spider-Man of 2002"

},

{

id: 2,

title: "Spider-Man 2",

rating: "Placeholder: ratingAve",

synopsis: "The sequel in Maguire’s tenure finds Peter Parker struggling to balance his personal life and his Spider-Man responsibilities. He is estranged from Mary Jane, who is now engaged to John Jameson, the son of the Daily Bugle editor, J. Jonah Jameson. Peter abandons his superhero identity, but as Harry Osborn grows more vengeful over his father’s death (the Green Goblin) and Dr. Otto Octavius emerges as a supervillain, Peter realizes that he has no choice but to get back in the game.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/spiderman2-1642713538.jpeg",

altDescribe: "Spider-Man 2 of 2004"

},

{

id: 3,

title: "Spider-Man 3",

rating: "Placeholder: ratingAve",

synopsis: "After a supernatural symbiote absorbs into his suit and turns it ink-black, Spider-Man realizes that his powers have strengthened, and in turn, have corrupted his motives. He’s confronted by his own personal demons and many villains — including a new Green Goblin, Venom, and the real killer of his Uncle Ben, Sandman.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/spiderman3-1642713578.jpeg",

altDescribe: "Spider-Man 3 of 2007"

},

{

id: 4,

title: "The Amazing Spider-Man",

rating: "Placeholder: ratingAve",

synopsis: "Andrew Garfield assumes the role of Spider-Man in this retelling of the webslinger’s origin story, directed by Marc Webb. Emma Stone plays Peter Parker’s crush, Gwen Stacy, daughter of police captain George Stacy. After discovering his superhero abilities and the secrets of his past, Spider-Man sets off on his quest to defeat the Lizard.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/amazingspiderman-1642713646.jpeg",

altDescribe: "he Amazing Spider-Man"

},

{

id: 5,

title: "The Amazing Spider-Man 2",

rating: "Placeholder: ratingAve",

synopsis: "Similar to the storyline of Raimi’s third installment, The Amazing Spider-Man 2 depicts the superhero battling a score of supervillains, including Electro, the Green Goblin, and Rhino. Gwen Stacy returns, as Peter Parker is torn between his strong feelings for her and her father’s dying wish that he stay away and protect Gwen from harm.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/amazingspiderman2-1642713696.jpeg",

altDescribe: "Spider-Man: Homecoming 2017"

},

{

id: 6,

title: "Spider-Man: Homecoming",

rating: "Placeholder: ratingAve",

synopsis: "Tom Holland’s trilogy officially begins. Directed by Jon Watts, Homecoming follows Peter Parker as he settles into his dual identity of high school student and superhero. Meanwhile, Tony Stark watches over the novice as he hones his skills. But when the villainous Vulture arrives on the scene, Spider-Man faces the ultimate challenge. This film introduces Zendaya as Michelle Jones-Watson, a.k.a MJ, and Jacob Batalon as Ned, Peter’s best friend.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/spider-man-homecoming-1642713844.jpeg",

altDescribe: "Spider-Man: Homecoming 2017"

},

{

id: 7,

title: "Avengers: Infinity War",

rating: "Placeholder: ratingAve",

synopsis: "Spider-Man returns for this major Avengers and Guardians of the Galaxy crossover event. When Thanos sets off on a mission to collect the six Infinity Stones for his wicked ways, Iron Man, Thor, Captain America, the Hulk, the Black Widow, and the rest of the Avengers unite in an effort to defeat and destroy the villain.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/avengersinfinitywar-1642713911.jpeg",

altDescribe: "Avengers: Infinity War 2018"

},

{

id: 8,

title: "Avengers EndGame",

rating: "Placeholder: ratingAve",

synopsis: "The sequel to Infinity War arrived just one year later, chronicling the aftermath of the Avengers’ clash with Thanos. The universe is in ruins and half of all life was destroyed at the hands of the warlord. The Avengers that remain band together to resurrect their fallen allies and battle Thanos one last time.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/avengersendgame-1642714188.jpeg",

altDescribe: "Avengers Endgame 2019"

},

{

id: 9,

title: "Spider-Man: Far From Home",

rating: "Placeholder: ratingAve",

synopsis: "The events of Endgame set up those in Far From Home, the second chapter in Tom’s Spider-Man journey. Peter Parker embarks on a summer European adventure with his classmates, where his attraction to MJ continues to strengthen. But the fun and games are brought to an abrupt end when Spider-Man is recruited by Nick Fury, who needs help resolving the mysterious Elemental attacks that are threatening society.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/spider-man-far-from-home-1642714214.jpeg",

altDescribe: "Spider-Man: Far From Home 2019"

},

{

id: 10,

title: "Spider-Man: No Way Home",

rating: "Placeholder: ratingAve",

synopsis: "In the final installment of Watts’ iteration, Spider-Man’s true identity is revealed and Peter Parker has a tough time dealing with the widespread attention. He visits Dr. Strange, in hopes that he could help make the world forget that he is Spider-Man; but when the spell goes terribly wrong, the multiverse is split open and the most powerful villains of Spider-Man’s past re-emerge once more.",

imageUrl: "https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/spider-man-no-way-home-1642714239.jpeg",

altDescribe: "Spider-Man: No Way Home 2021"

},

];

export default movieData

**File : Components\MovieList.js**

import React from 'react';

import Movie from './Movie';

import movieData from './movieData';

function MovieList() {

return (

<div className="movie-container">

{movieData.map(movie =>

<Movie key={movie.id} {...movie}/>)}

</div>

);

}

export default MovieList;

**File : Components\Rating.js**

import React from 'react'

function Rating() {

return (

<div className="rating">

<input type="radio" name="rating" value="5" id="5"/><label for="5">☆</label>

<input type="radio" name="rating" value="4" id="4"/><label for="4">☆</label>

<input type="radio" name="rating" value="3" id="3"/><label for="3">☆</label>

<input type="radio" name="rating" value="2" id="2"/><label for="2">☆</label>

<input type="radio" name="rating" value="1" id="1"/><label for="1">☆</label>

</div>

)

}

export default Rating;

**File : Components\ReviewComponent.js**

import React from 'react';

import Rating from './Rating'

function ReviewComponent(props) {

return (

<div className="flex-container">

<form onSubmit={props.handleSubmit}>

<label>Name:

<input

type="text"

value={props.name}

onChange={props.handleChange}

name="name"

placeholder="Please enter your name"

/>

</label>

<br />

<label>Review:

<textarea

type="textarea"

value={props.review}

onChange={props.handleChange}

name="review"

placeholder="Please enter your comments here"

/>

</label>

<br />

<div className="rating">

<Rating

value={props.rating}

onChange={props.handleChange}

name="rating"

/>

</div>

<br />

<button>Submit</button>

</form>

</div>

)

}

export default ReviewComponent

**File : Components\ReviewData.js**

import React from 'react'

class ReviewData extends React.Component {

constructor(props) {

super(props);

this.state = {

value: '',

reviews: [],

};

}

onAddItem = () => {

this.setState(state => {

const reviews = state.reviews.concat(state.value);

return {

reviews,

value: "",

};

});

};

render() {

return(

<div>

<ul>

{this.state.reviews.map(item => (

<li key={item}>{item}</li>

))}

</ul>

</div>

);

}

}

export default ReviewData;

**File : Components\ReviewForm.js**

import React from 'react';

import ReviewComponent from './ReviewComponent'

class ReviewForm extends React.Component {

constructor(props) {

super(props);

this.state = {

name: '',

review: '',

rating: '',

}

this.handleChange = this.handleChange.bind(this);

this.handleSubmit = this.handleSubmit.bind(this);

}

handleChange(event) {

const {name, value, type, checked} = event.target

type === "radio" ? this.setState({

[name]: checked }) : this.setState({[name]: value

});

}

handleSubmit(event) {

event.preventDefault();

const reviewData = {

name: this.name.value,

review: this.review.value,

rating: this.rating.value,

}

return reviewData;

}

render() {

return (

<ReviewComponent

handleChange={this.handleChange}

handleSubmit={this.handleSubmit}

data={this.state}

/>

)

}

}

export default ReviewForm;

**File : Components\ReviewList.js**

import React, { useState, useEffect } from 'react';

import ReviewData from './ReviewData';

//list of reviews per movie to display after review form

function ReviewList() {

const [ reviews, setReviews ] = useState([]);

useEffect(() => {

//get reviews from review data file

fetch(ReviewData)

// .then(res => res.json())

.then((reviews) => {

setReviews(reviews);

});

}, []);

return <div>

{reviews.length > 0 && reviews.map((review) =>

<ReviewData key={review.id} data={review}/>

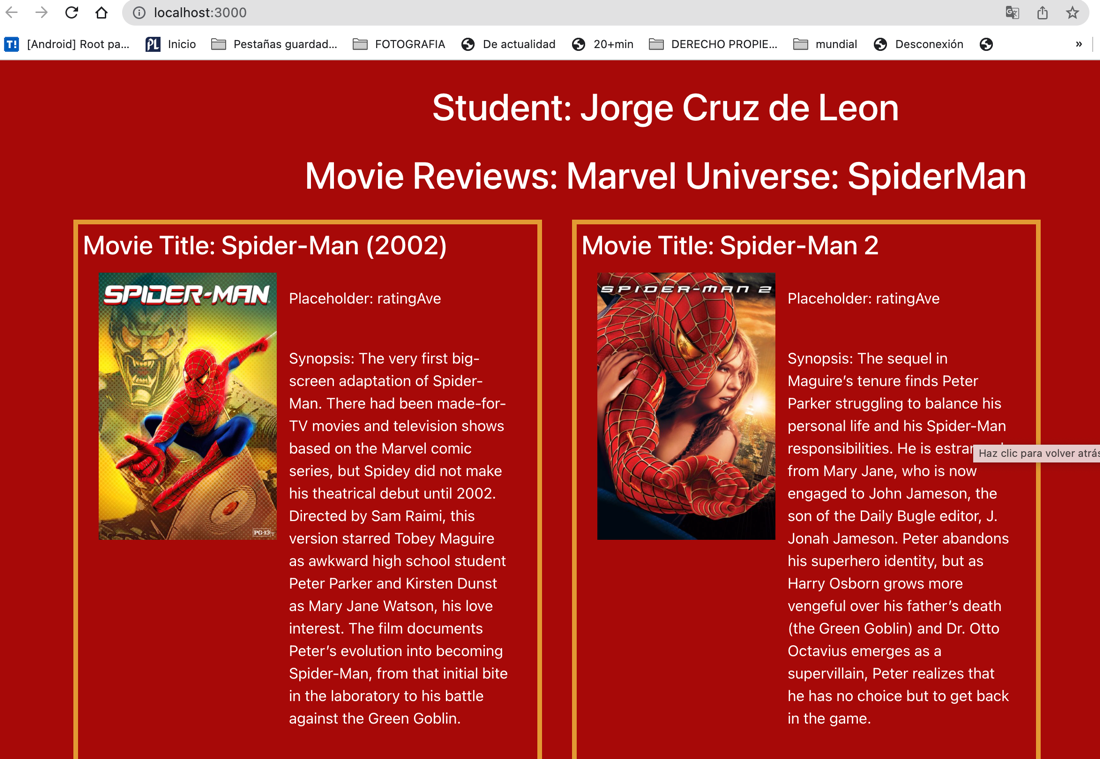
)}

</div>

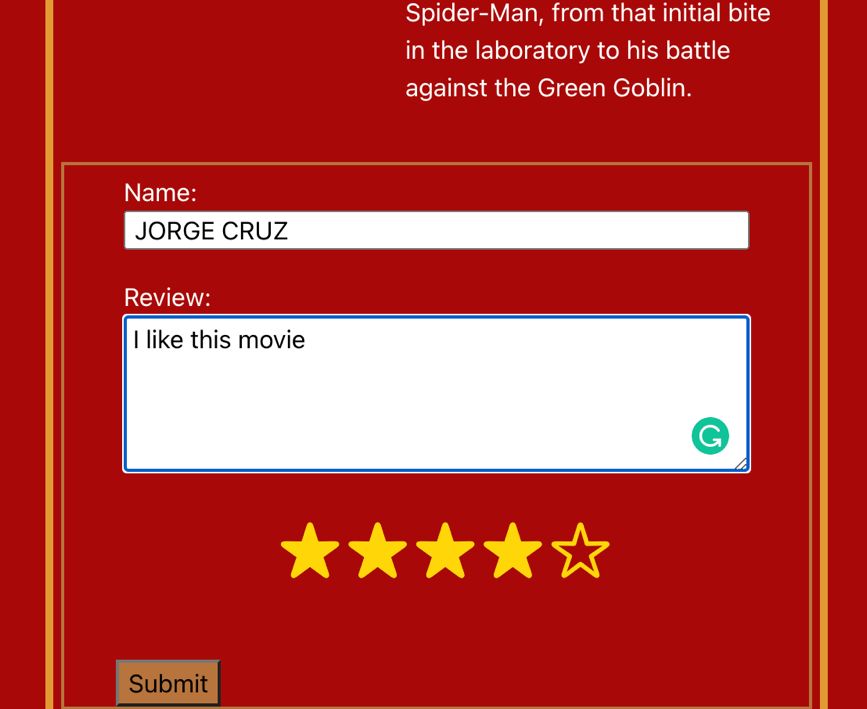
}

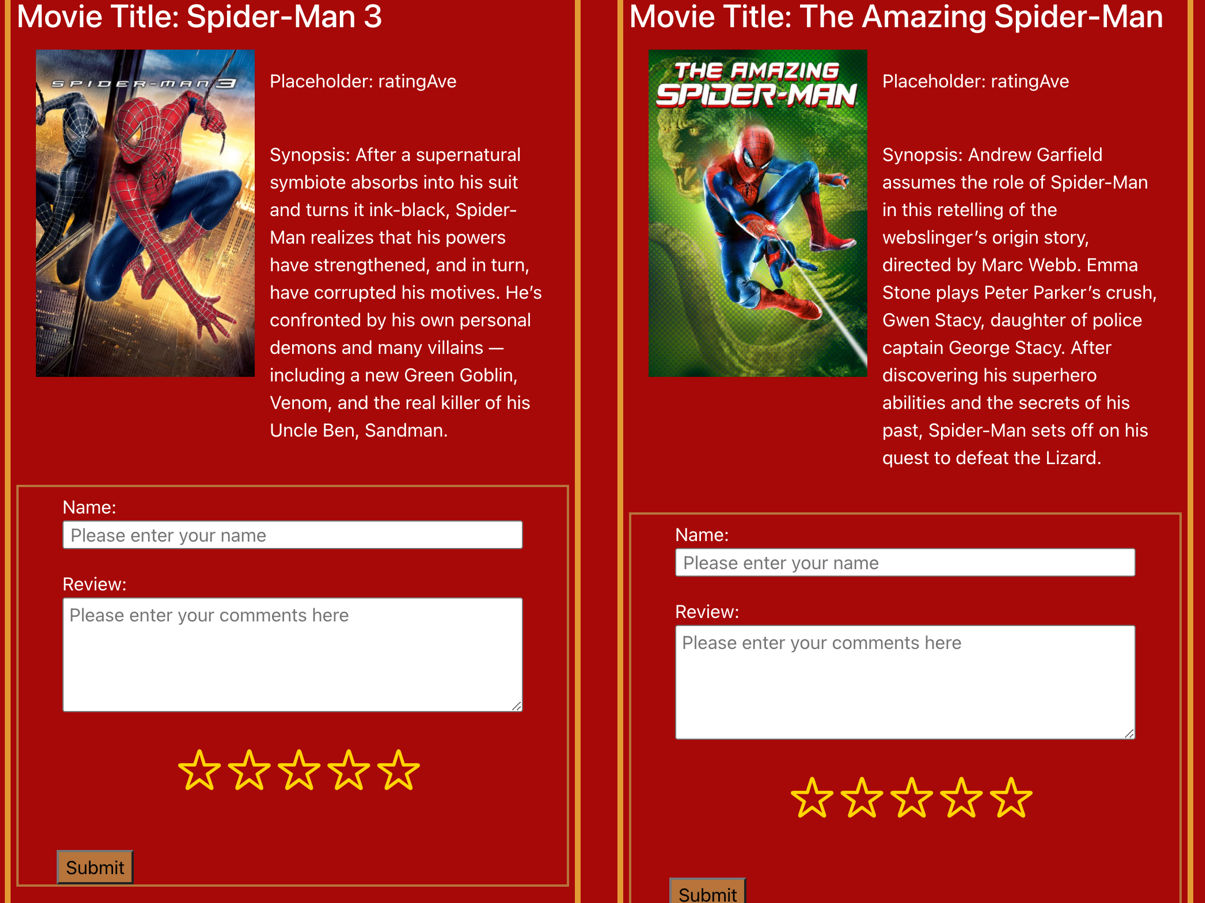
export default ReviewList;

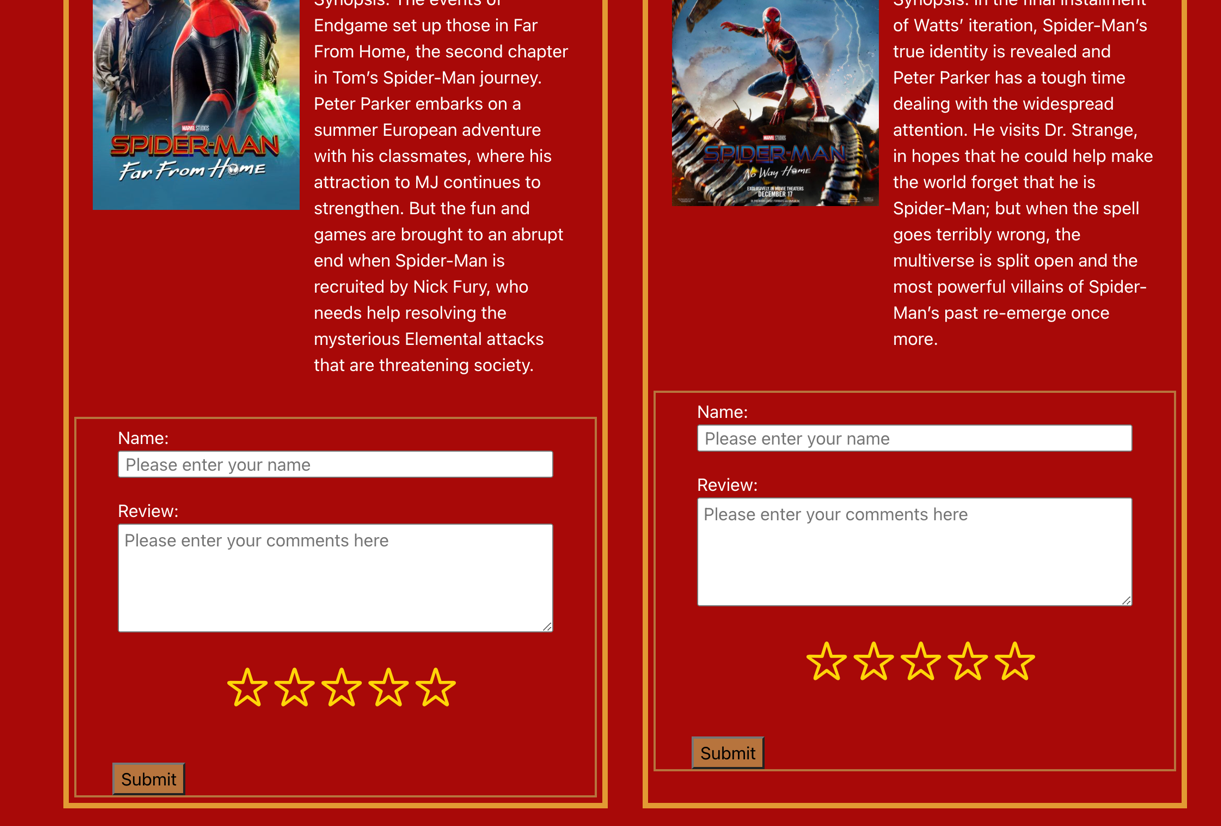
**Screenshots of Running Application:**

****

**\**

****

****

****

**URL to GitHub Repository:**

[**https://github.com/rafacruz78gt/react\_week2\_project\_movie**](https://github.com/rafacruz78gt/react_week2_project_movie)