



Web 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:
 - a. MovieList – a container for all the Movie components and their data.
 - b. Movie – a component that represents movie data (i.e. image, synopsis, rating, etc...)
 - c. 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!)
 - d. ReviewList – a container inside of a Movie that houses Review components.
 - e. Review – A text review a user can leave on a movie.
 - f. 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.



PROMINEO TECH

Screenshots of Code:

CSS:

```
# index.css X JS index.js JS Navbar.js JS Stars.js JS StarShape.js JS ReviewList.js JS Reviews.js JS ReviewForm.js JS MovieList.js JS Movie.js
src > # index.css > #star-button
1 body {
2   margin: 0;
3   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen', 'Ubuntu', 'Cantarell',
4   'Fira Sans', 'Droid Sans', 'Helvetica Neue', sans-serif;
5   -webkit-font-smoothing: antialiased;
6   -moz-osx-font-smoothing: grayscale;
7   background-color: #rgb(130, 128, 245);
8   justify-content: center;
9 }
10
11 code {
12   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New', monospace;
13 }
14
15
16
17 /*Stars*/
18 #star-button {
19   background-color: transparent;
20   border: none;
21   outline: none;
22   cursor: pointer;
23 }
24 .on {
25   color: #rgb(212, 180, 0);
26 }
27 .off {
28   color: #rgb(223, 222, 222);
29 }
30
31
32
33 /*Navbar*/
34 nav {
35   background-color: #rgb(212, 180, 0);
36   height: 40px;
37   text-align: left;
38   font-size: 100px;
39   color: #blue;
40   margin: 2px;
41   padding: 10px;
42   border: 2px solid #blue;
43 }
44
45
46
47
48 /*Movie Cards*/
49 .mov-c {
50   background-color: #rgb(212, 180, 0);
51   text-align: left;
52   color: #blue;
53   margin: 2px;
54   padding: 10px;
55   border: 2px solid #blue;
56 }
57
58 .mov-h {
59   background-color: #rgb(33, 31, 146);
60   color: #white;
61   font-size: 50px;
62 }
63
64 .mov-b {
65   background-color: #white;
66   font-size: 20px;
67 }
68
69 .mov-f {
70   background-color: #rgb(33, 31, 146);
71 }
72
73 .mov-s {
74   text-align: center;
75   color: #rgb(33, 31, 146);
76   margin: 2px;
77   padding: 5px;
78 }
79
80 .mov-p {
81   text-align: center;
82   margin: 2px;
83   padding: 5px;
84 }
85
86 .poster {
87   height: 400px;
88   border: 4px solid #black;
89 }
90
91
```



PROMINEO TECH

```
01
02
03 /*Review Cards*/
04 .rev-c {
05   background-color: #white;
06   text-align: left;
07   color: #white;
08   margin: 2px;
09   padding: 5px;
100  border: 2px solid #rgb(212, 180, 0);
101 }
102
103 .rev-h {
104   background-color: #rgb(212, 180, 0);
105   color: #rgb(33, 31, 146);
106   font-size: 20px;
107 }
108
109 .rev-b {
110   background-color: #white;
111   color: #rgb(33, 31, 146);
112   font-size: 15px;
113 }
114
115 .rev-f {
116   background-color: #rgb(33, 31, 146);
117   font-size: 15px;
118 }
119
120
```

Index:

```
# index.css JS indexjs X JS Navbarjs JS Starsjs JS StarShapejs JS ReviewListjs JS Reviewjs JS ReviewFormjs JS MovieListjs JS Moviejs
src > JS indexjs
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './node_modules/bootstrap/dist/css/bootstrap.css'
4 import './index.css';
5 import App from './App';
6 import reportWebVitals from './reportWebVitals';
7 import MovieList from './components/MovieList';
8 import NavBar from './components/Navbar';
9
10 ReactDOM.render(
11   <React.StrictMode>
12     <div>
13       <NavBar />
14       <MovieList />
15     </div>
16   </React.StrictMode>,
17   document.getElementById('root')
18 );
19
20 reportWebVitals();
21
```

Navbar:

```
# index.css JS indexjs JS Navbarjs X JS Starsjs JS StarShapejs JS ReviewListjs JS Reviewjs JS ReviewFormjs JS MovieListjs JS Moviejs
src > components > JS Navbarjs > ...
1 import React from 'react';
2
3 export default class NavBar extends React.Component {
4   render() {
5     return(
6       <nav className="navbar">
7         <div className="container-fluid">
8           <h1>Movie-Rater</h1>
9         </div>
10      </nav>
11    );
12  }
13 }
14
15 }
16
```

Star-Rating:



PROMINEO TECH

```
# index.css JS index.js JS Navbars.js JS Stars.js X JS StarShape.js JS ReviewList.js JS Reviews.js JS ReviewForm.js JS MovieList.js JS Movies.js
src > components > JS Stars.js > ...
1  /*
2   * Stars - a one to five-star rating component that allows users to rate
3   * something (movies in this case, but remember that components are
4   * reusable, so you could use it elsewhere!) */
5
6  import React from "react";
7  import { useState } from "react";
8  import StarShape from "../StarShape";
9
10 const Stars = () => {
11   const [rating, setRating] = useState(0);
12   const [hover, setHover] = useState(0);
13   return (
14     <div className="stars">
15       {[...Array(5)].map((star, index) => {
16         index = index + 1;
17         return (
18           <button type="button" id="star-button" key={index} className={index <= rating ? "on" : "off"}
19             onClick={() => setRating(index)} onMouseEnter={() => setHover(index)} onMouseLeave={() => setHover(rating)}>
20             <StarShape />
21           </button>
22         );
23       })}
24     </div>
25   );
26 };
27
28 export default Stars;
29
```

Star-shape:

```
# index.css JS index.js JS Navbars.js JS Stars.js JS StarShape.js X JS ReviewList.js JS Reviews.js JS ReviewForm.js JS MovieList.js JS Movies.js
src > components > JS StarShape.js > ...
1  import React from 'react'
2
3  export default class StarShape extends React.Component {
4    render() {
5      return (
6        <span className="star">&#9733;</span>
7      )
8    }
9  }
10
```

Review-List:

```
# index.css JS index.js JS Navbars.js JS Stars.js JS StarShape.js JS ReviewList.js X JS Reviews.js JS ReviewForm.js JS MovieList.js JS Movies.js
src > components > JS ReviewList.js > ...
1  /* ReviewList - a container inside of a Movie that houses Review components. */
2
3  import React from "react";
4  import Review from "../Review";
5  import ReviewForm from "../ReviewForm";
6
7  export default class ReviewList extends React.Component {
8    constructor(props) {
9      super(props);
10     this.state = {
11       reviews: [
12         {
13           username: "Jake000",
14           movieReview: "I hated this"
15         },
16         {
17           username: "Hannah289",
18           movieReview: "This was awesome!!"
19         },
20         {
21           username: "moviecritic56698",
22           movieReview: "WORST MOVIE EVER!!!!!!!!!!!!!!!"
23         }
24       ];
25     };
26
27     addReview(review) {
28       this.setState({
29         reviews: [review, ...this.state.reviews]
30       });
31     }
32
33     render() {
34       let reviews;
35
36       if(this.state.reviews) {
37         reviews = this.state.reviews.map((review, index) => <Review key={index} {...review} />);
38       }
39       return (
40         <div>
41           <div><ReviewForm /></div>
42           <div className="d-flex flex-wrap justify-content-center">
43             {reviews}
44           </div>
45         </div>
46       )
47     }
48   }
49 }
```



PROMINEO TECH

Review:

```
# index.css JS index.js JS Navbars.js JS Stars.js JS StarShape.js JS ReviewList.js JS Review.js X JS ReviewForm.js JS MovieList.js JS Movie.js
src > components > JS Review.js > ...
1  /* Review - A text review a user can leave on a movie. */
2
3  import React from "react";
4  import Stars from "../Stars";
5
6  export default class Review extends React.Component {
7    constructor(props) {
8      super(props);
9      this.state = {
10        username: props.username,
11        movieReview: props.movieReview,
12      };
13    }
14
15    render() {
16      return (
17        <div className="card rev-c w-75">
18          <h5 className="card-header rev-h">
19            {this.state.username}
20          </h5>
21          <div className="card-body rev-b">
22            {this.state.movieReview}
23          </div>
24          <div className="card-footer rev-f">
25            <Stars />
26          </div>
27        </div>
28      );
29    }
30  }
```

Review-Form

```
# index.css JS index.js JS Navbars.js JS Stars.js JS StarShape.js JS ReviewList.js JS Review.js JS ReviewForm.js X JS MovieList.js JS Movie.js
src > components > JS ReviewForm.js > ReviewForm > render
1  /* ReviewForm - a form at the bottom of a Movie component that allows users to leave reviews.
2  When submitted, the review should be added to the movie. All this data can be stored in an
3  array, no networking or database needed for this assignment. */
4
5  import React from "react";
6  import Review from "../Review";
7
8  export default class ReviewForm extends React.Component {
9    constructor(props) {
10      super(props);
11      this.state = {
12        review: {
13          username: '',
14          movieReview: ''
15        },
16        reviews: []
17      };
18
19      this.handleChange = this.handleChange.bind(this);
20      this.handleSubmit = this.handleSubmit.bind(this);
21    }
22
23    handleChange(event) {
24      const {value, name} = event.target;
25      this.setState({
26        review: {
27          ...this.state.review,
28          [name]: value
29        }
30      });
31    };
32
33    handleSubmit(event) {
34      if(this.state.review !== ''){
35        this.state.reviews.push(new Review(this.state.username, this.state.movieReview))
36      }
37      event.preventDefault();
38    }
39
40    render() {
41      return (
42        <div className="card rev-c">
43          <form onSubmit={this.handleSubmit}>
44            <div className="card-header rev-h">
45              <h3>Leave a Review</h3>
46            </div>
47            <div className="card-body rev-b">
48              <div className="form-group">
```



PROMINEO TECH

```
48     <div className="card-body rev-b">
49       <div className="form-group">
50         <label>
51           Username:
52           <input
53             type="text" required
54             name="username"
55             placeholder="Username"
56             className="form-control"
57             value={this.state.review.username}
58             onChange={this.handleChange}/>
59         </label>
60       </div>
61
62       <br />
63
64       <div className="form-group">
65         <label>
66           Review:
67           <textarea
68             required rows="5" cols="30" minLength="5"
69             name="movieReview"
70             placeholder="Review"
71             className="form-control"
72             value={this.state.review.movieReview}
73             onChange={this.handleChange} />
74         </label>
75       </div>
76
77       <br />
78
79       <div className="card-footer rev-f">
80         <div className="form-group">
81           <button className="btn btn-primary">Submit</button>
82         </div>
83       </div>
84     </div>
85   </form>
86 </div>
87   );
88 }
89 }
90
91
92
93
```

Movie-List:

```
# index.css  JS index.js  JS Navbars.js  JS Stars.js  JS StarShape.js  JS ReviewList.js  JS Reviews.js  JS ReviewForm.js  JS MovieList.js  JS Movie.js
src > components > JS MovieLists > MovieList > constructor > movies
1  /* MovieList - a container for all the Movie components and their data. */
2
3  import Argo from '../assets/Argo.jpg';
4  import Gladiator from '../assets/Gladiator.jpg';
5  import Titanic from '../assets/Titanic.jpg';
6
7  import React from "react";
8  import Movie from "../Movie";
9
10 export default class MovieList extends React.Component {
11   constructor(props) {
12     super(props);
13     this.state = {
14       movies: [
15         {
16           title: "Argo",
17           image: Argo,
18           synopsis: "It's about Iran",
19           rating: "R",
20           reviews: []
21         },
22         {
23           title: "Titanic",
24           image: Titanic,
25           synopsis: "It's about a boat",
26           rating: "PG-13",
27           reviews: []
28         },
29         {
30           title: "Gladiator",
31           image: Gladiator,
32           synopsis: "It's about Rome",
33           rating: "R",
34           reviews: []
35         }
36       ]
37     };
38   }
39   render() {
40     let movies;
41     if(this.state.movies) {
42       movies = this.state.movies.map((movie, index) => <Movie key={index} {...movie} />);
43     }
44     return (
45       <div className="d-flex flex-wrap justify-content-center">
46         {movies}
47       </div>
48     );
49   }
50 }
```

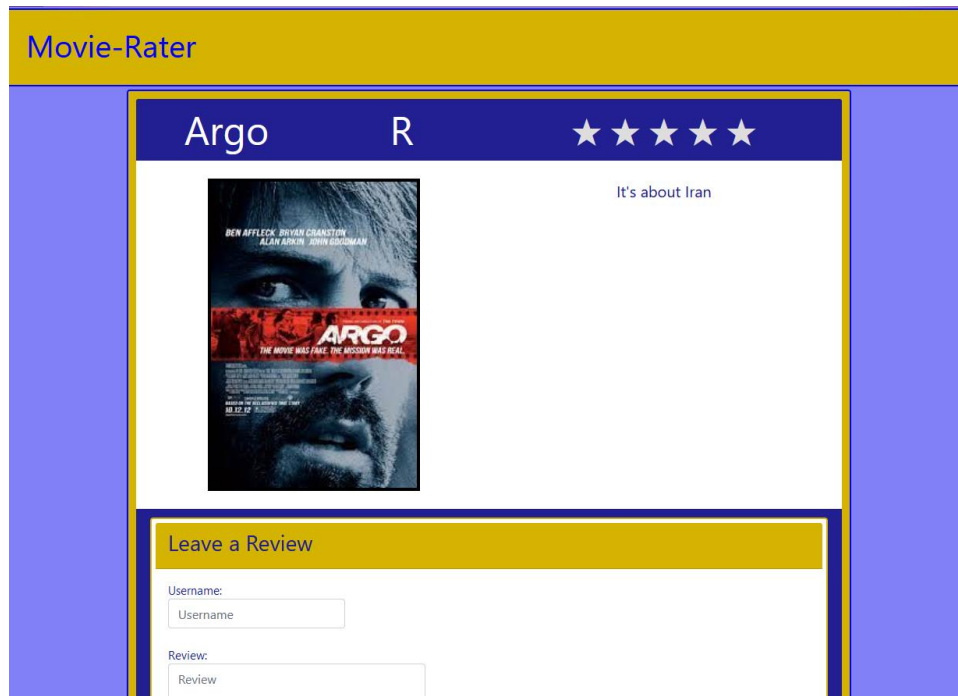
Movie:



PROMINEO TECH

```
# index.css JS index.js JS Navbar.js JS Stars.js JS StarShape.js JS ReviewList.js JS Reviews.js JS ReviewForm.js JS MovieList.js JS Movie.js x
src > components > JS Movie.js > Movie > constructor
1  /* Movie - a component that represents movie data (i.e. image, synopsis, rating, etc...) */
2
3  import React from "react";
4  import Stars from "../Stars";
5  import ReviewList from "../ReviewList";
6
7  export default class Movie extends React.Component {
8    constructor(props){
9      super(props);
10     this.state = {
11       title: props.title,
12       image: props.image,
13       synopsis: props.synopsis,
14       rating: props.rating,
15       reviews: []
16     };
17   }
18
19   render() {
20     return (
21       <div className="card mov-c w-75">
22         <h4 className="card-header text-center mov-h">
23           <div className="row">
24             <div className="col">{this.state.title}</div>
25             <div className="col">{this.state.rating}</div>
26             <div className="col-6"><Stars /></div>
27           </div>
28         </h4>
29         <div className="card-body mov-b">
30           <div className="row">
31             <div className="col mov-p"><img className="poster" src={this.state.image} alt={this.state.title}/></div>
32             <div className="col mov-s"><p className="card-text">{this.state.synopsis}</p></div>
33           </div>
34           <div className="card-footer mov-f">
35             <div><ReviewList /></div>
36           </div>
37         </div>
38       </div>
39     );
40   }
41 }
42
43 }
```

Screenshots of Running Application:





PROMINEO TECH

Leave a Review

Username:

Username

Review:

Review

Submit

JakeDDD

I hated this

Hannah289

This was awesome!!

moviecritic56698

WORST MOVIE EVER!!!!!!!!!!!!

Leave a review:

Leave a Review

Username:

moviecritic12345

Review:

Leonardo DiCaprio and Kate Winslet excel in this movie about...

Submit

JakeDDD

I hated this

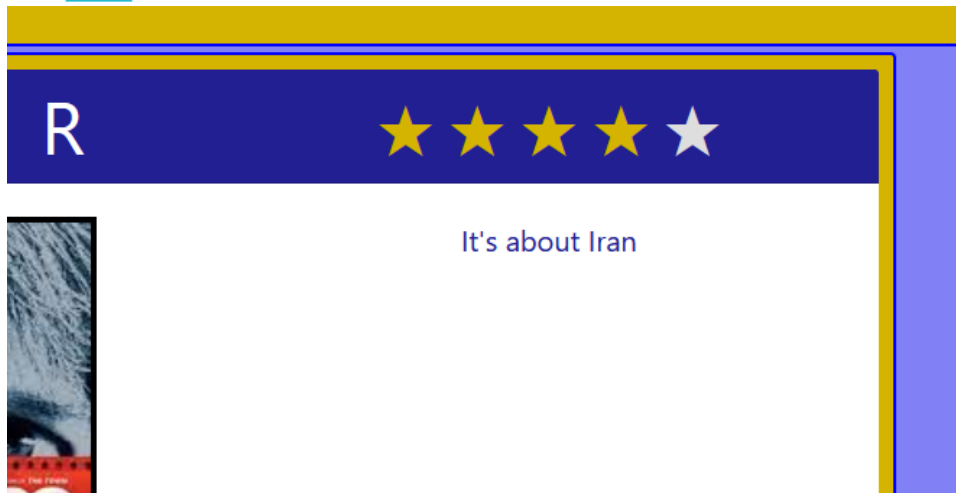
Hannah289

This was awesome!!

Leave a star rating:



PROMINEO TECH



URL to GitHub Repository:

https://github.com/aheiser2/Week-14-Assignment_Movie-Rater.git