PROMINEO TECH

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.

Screenshots of Code:

```javascript
import Armageddon from '../assets/Armageddon.jpg';
import Avatar from '../assets/Avatar.jpg';
import Titanic from '../assets/Titanic.png';

import React from "react";
import Movie from "./Movie";

export default class MovieList extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            movies: [ {
                title: "Armageddon",
                image: Armageddon,
                synopsis: "After discovering that an asteroid the size of Texas is going to impact Earth in less than a month, NASA recruits a misfit team of deep-core drillers to save the planet.",
                reviews: []
            },
            {
                title: "Avatar",
                image: Avatar,
                synopsis: "A paraplegic Marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.",
                reviews: []
            },
            {
                title: "Titanic",
                image: Titanic,
                synopsis: "A seventeen-year-old aristocrat falls in love with a kind but poor artist aboard the luxurious, ill-fated R.M.S. Titanic.",
                reviews: []
            }]
        };
    };
    render() {
        let movies;

        if(this.state.movies) {
            movies = this.state.movies.map((movie, index) => <Movie key={index} {...movie} />);
        }
        return (
            <div className="d-flex flex-wrap justify-content-center">
                {movies}
            </div>
        )
    }
}
```

```javascript
import React from "react";
import Stars from "./Stars";
import ReviewList from "./ReviewList";

export default class Movie extends React.Component {
    constructor(props){
        super(props);
        this.state ={
            title: props.title,
            image: props.image,
            synopsis: props.synopsis,
            rating: props.rating,
            reviews: []
        };
    }

    render() {
        return (
            <div className="card mov-c w-75">
                <h4 className="card-header text-center mov-h">
                    <div className="row">
                        <div className="col">{this.state.title}</div>
                        <div className="col">{this.state.rating}</div>
                        <div className="col-6"><Stars /></div>
                    </div>
                </h4>
                <div className="card-body mov-b">
                    <div className="row">
                        <div className="col mov-p"><img className="poster" src={this.state.image} alt={this.state.title}/></div>
                        <div className="col mov-s"><p className="card-text">{this.state.synopsis}</p></div>
                    </div>
                </div>
                <div className="card-footer mov-f">
                    <div><ReviewList /></div>
                </div>
            </div>
        )
    }
}
```

Tabs: Movie.js | MovieList.js | index.css | index.js | Review.js | ReviewForm.js | ReviewList.js | Stars.js ✕

movie_rater > src > components > JS Stars.js > [∅] Stars > ⊙ map() callback

```javascript
1
2  import React from "react";
3  import { useState } from "react";
4
5
6  const Stars = () => {
7    const [rating, setRating] = useState(0);
8    const [hover, setHover] = useState(0);
9    return (
10     <div className="stars">
11       {[...Array(5)].map((star, index) => {
12         index = index + 1;
13         return (
14           <button type="button" id="star-button" key={index} className={index <= rating ? "on" : "off"}
15             onClick={() => setRating(index)} onMouseEnter={() => setHover(index)} onMouseLeave={() => setHover(rating)}>
16             <span className="star">&#9733;</span>
17           </button>
18         );
19       })}
20     </div>
21   );
22 };
23
24 export default Stars;
25
26
```

Tabs: Movie.js | MovieList.js | index.css | index.js | Review.js | ReviewForm.js | ReviewList.js ✕ | Stars.js

movie_rater > src > components > JS ReviewList.js > ❖ ReviewList > ⊙ constructor > ⚿ reviews > ⚿ movieReview

```javascript
1  import React from "react";
2  import Review from "./Review";
3  import ReviewForm from "./ReviewForm";
4
5  export default class ReviewList extends React.Component {
6    constructor(props) {
7      super(props);
8      this.state = {
9        reviews: [ {
10         username: "John",
11         movieReview: "It was amazing!"
12       },
13       {
14         username: "Mary",
15         movieReview: "Boring!"
16       },
17       {
18         username: "Alex",
19         movieReview: "Not too bad!!"
20       }
21       ]
22     };
23   };
24
25   addReview(review) {
26     this.setState({
27       reviews: [review, ...this.state.reviews]
28     });
29   }
30
31   render() {
32     let reviews;
33
34     if(this.state.reviews) {
35       reviews = this.state.reviews.map((review, index) => <Review key={index} {...review} />);
36     }
37     return (
38       <div>
39         <div><ReviewForm /></div>
40         <div className="d-flex flex-wrap justify-content-center">
41           {reviews}
42         </div>
43       </div>
44     )
45   }
46
```

Tabs: JS Movie.js | JS MovieList.js | # index.css | JS index.js | JS Review.js | JS ReviewForm.js ✕ | JS ReviewList.js | JS Stars.js

movie_rater > src > components > JS ReviewForm.js > ⅁ ReviewForm > ⬡ render

```javascript
import React from "react";
import Review from "./Review";

export default class ReviewForm extends React.Component {
    constructor(props) {
        super(props);
        this.state ={
          review: {
            username: '',
            movieReview: ''
          },
          reviews: []
        };

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

    handleChange(event) {
      const {value, name} = event.target;
      this.setState({
        review: {
          ...this.state.review,
          [name]: value
        }
      });
    };

    handleSubmit(event) {
        if(this.state.review !== ''){
          this.state.reviews.push(new Review(this.state.username, this.state.movieReview))
        }
        event.preventDefault();
    }
```

Tabs: JS Movie.js | JS MovieList.js | # index.css | JS index.js | JS Review.js ✕ | JS ReviewForm.js | JS ReviewList.js | JS Stars.js

movie_rater > src > components > JS Review.js > ...

```javascript
import React from "react";
import Stars from "./Stars";

export default class Review extends React.Component {
    constructor(props){
        super(props);
        this.state ={
            username: props.username,
            movieReview: props.movieReview,
        };
    }

    render() {
        return (
            <div className="card rev-c w-75">
                <h5 className="card-header rev-h">
                    {this.state.username}
                </h5>
                <div className="card-body rev-b">
                    {this.state.movieReview}
                </div>
                <div className="card-footer rev-f">
                    <Stars />
                </div>
            </div>
        )
    }
}
```

```
JS Movie.js      JS MovieList.js    # index.css    JS index.js    JS Review.js    JS ReviewForm.js ×    JS ReviewList.js    JS Stars.js

movie_rater > src > components > JS ReviewForm.js > ⅍ ReviewForm > ⊙ render
29      handleSubmit(event) {
30          if(this.state.review !== ''){
31              this.state.reviews.push(new Review(this.state.username, this.state.movieReview))
32          }
33          event.preventDefault();
34      }
35
36      render() {
37        return (
38          <div className="card rev-c">
39              <form onSubmit={this.handleSubmit}>
40                  <div className="card-header rev-h">
41                      <h3>Review this movie</h3>
42                  </div>
43
44                  <div className="card-body rev-b">
45                      <div className="form-group">
46                      <label>
47                          Username:
48                          <input
49                          type="text" required
50                          name="username"
51                          placeholder="Username"
52                          className="form-control"
53                          value={this.state.review.username}
54                          onChange={this.handleChange}/>
55                      </label>
56                      </div>
57
58                      <br />
59
60                      <div className="form-group">
61                      <label>
62                          Review:
63                          <textarea
64                          required rows="5" cols="30" minLength="5"
65                          name="movieReview"
66                          placeholder="Review"
67                          className="form-control"
68                          value={this.state.review.movieReview}
69                          onChange={this.handleChange} />
70                      </label>
71                      </div>
72                  </div>
```

```
73
74                      <br />
75
76                  <div className="card-footer rev-f">
77                      <div className="form-group">
78                      <button className="btn btn-primary">Submit</button>
79                      </div>
80                  </div>
81              </form>
82          </div>
83        );
84      }
85  }
86
87
88
89
```

Screenshots of Running Application:

Armageddon ★ ★ ★ ★ ★

After discovering that an asteroid the size of Texas is going to impact Earth in less than a month, NASA recruits a misfit team of deep-core drillers to save the planet.

Review this movie

Username:
Username

Review:
Review

Submit

John
It was amazing!
★ ★ ★ ★ ★

Mary
Boring!
★ ★ ★ ★ ★

Alex
Not too bad!
★ ★ ★ ★ ★

Avatar ★ ★ ★ ★ ★

A paraplegic Marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.

Review this movie

Username:
Username

Review:
Review

Submit

John
It was amazing!
★ ★ ★ ★ ★

Mary
Boring!
★ ★ ★ ★ ★

Alex
Not too bad!
★ ★ ★ ★ ★

Titanic ★ ★ ★ ★ ★

A seventeen-year-old aristocrat falls in love with a kind but poor artist aboard the luxurious, ill-fated R.M.S. Titanic.

Review this movie

Username:
Username

Review:
Review

Submit

John
It was amazing!
★ ★ ★ ★ ★

Mary
Boring!
★ ★ ★ ★ ★

Alex
Not too bad!
★ ★ ★ ★ ★

URL to GitHub Repository:

Note: This project was inspired on Abby Heiser's project.