

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:





```
··background-color: □rgba(7, 71, 168, 0.3);
··padding: 10px;
  restanding: center;
--align:items: center;
--text-emphasis-color: ■gray;
   font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
align-content: left;
align-items: flex-start;
align-content: left;
align-items: flex-start;
#menu {
| border: 10px dotted;
| border-color: □rgb(12, 12, 99);
| align-content: center;
| text-align: center;
#movielitie {
    font-size: 50px;
    padding-bottom: 10px;
    border-bottom: 5px □rgb(12, 12, 99) solid;
    background-color: □rgba(7, 9, 46, 0.35);
```



```
review.reboo > src > # App.css > ..
     #movieGenre {
      border: 3px □rgb(12, 12, 99) solid;
      background-color: □rgba(7, 9, 46, 0.35);
     #movieDate {
      border: 3px □rgb(12, 12, 99) solid;
       background-color: ☐rgba(7, 9, 46, 0.35);
       align-items: left;
       width: 250px;
       padding: 5px;
      #movieSynopsis {
       float: right;
      align-content: right;
       width: 750px;
       font-size: 20px;
       border: 10px dotted □rgb(12, 12, 99);
       margin: 1px;
      background-color: \square rgba(7, 9, 46, 0.5);
      #movieImage {
```



```
import React from 'react';
export default class ReviewForm extends React.Component {
    constructor(props) {
       super(props);
       this.state = {
           reviewInfo: '',
       this.submitForm = this.submitForm.bind(this);
       this.handleInputChange = this.handleInputChange.bind(this);
       this.setState({
           reviewInfo: '',
    submitForm() {
       this.props.onSubmit(this.state);
       this.resetForm();
   handleInputChange(event) {
          const target = event.target;
           const name = target.name;
           const value = target.value;
              ··[name]: value
    render() {
```



```
review.reboo > src > components > JS navbarjs > ...

import React from 'react';

export default class NavBar extends React. Component {

''render() {

''render() {

''render() {

''render() {

''return (

''ret
```

```
Troinex.reboo > src > components > # ratings > ** Ratings
```







```
average(ating: 0,

average(ating: 0,

average(ating: 0,

busitars: 0,

two stars: 0,

fourstars: 0,

fourstars: 0,

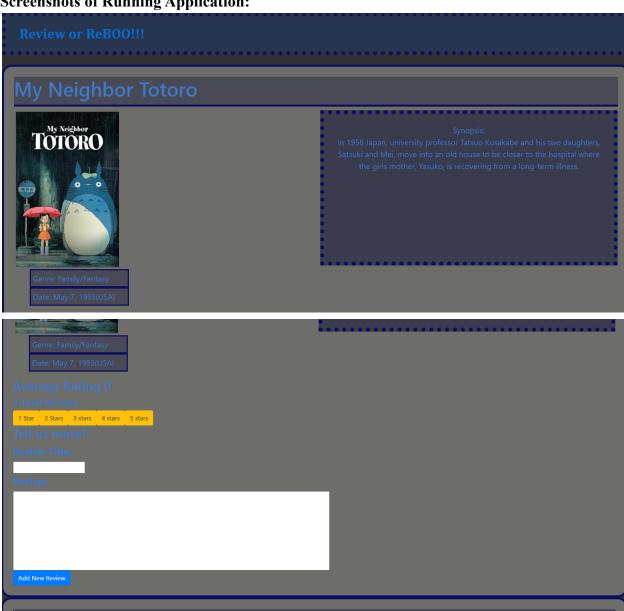
fourstars: 0,

fivestars: 0,

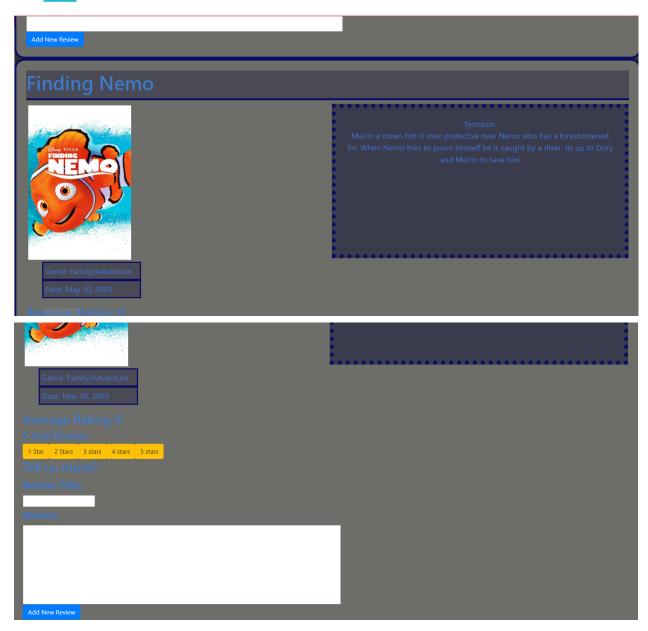
fivestars:
```



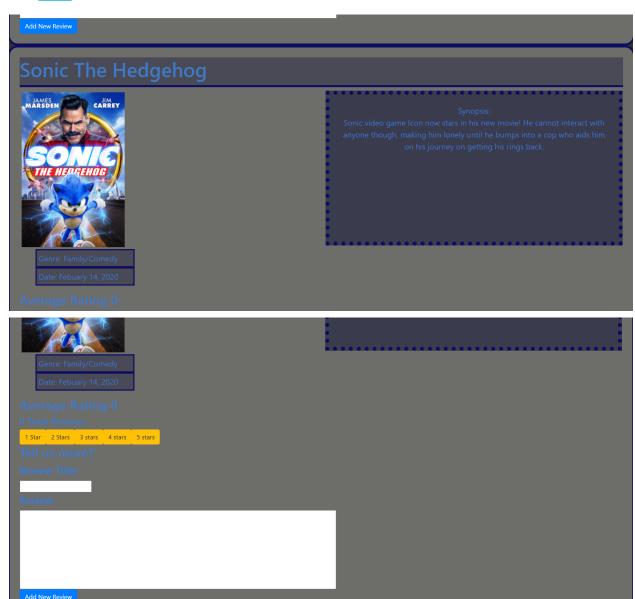
Screenshots of Running Application:



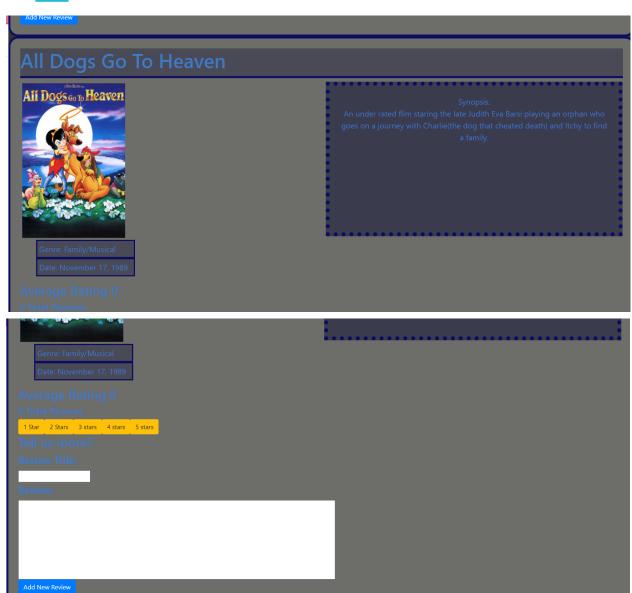




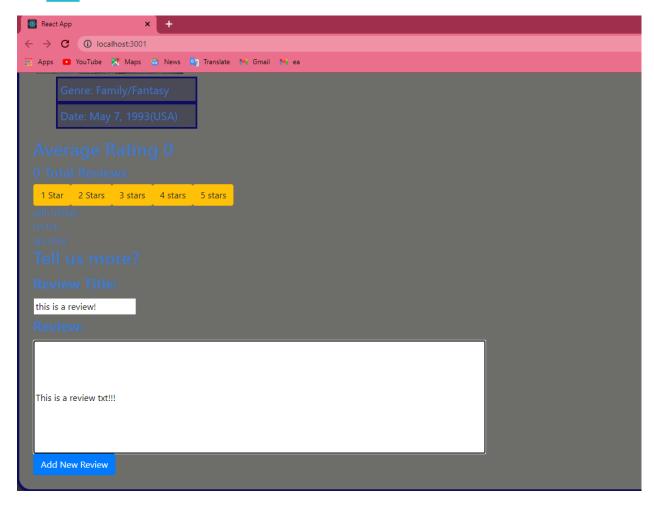






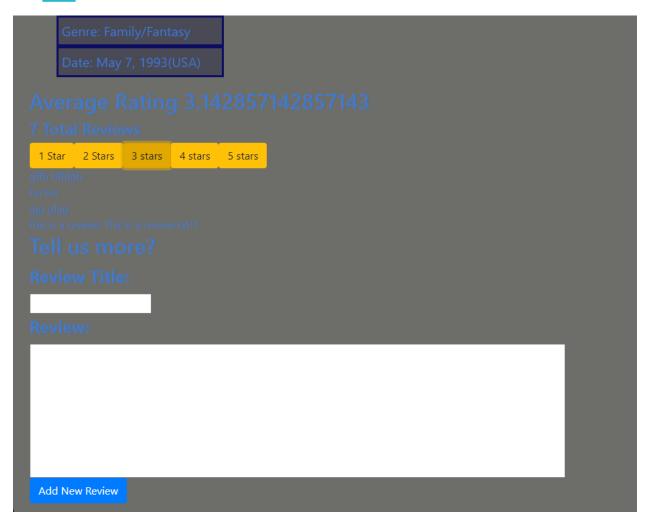








| Avera | age F | Rating | g 0 | |
|---------|----------|---------|---------|---------|
| 0 Total | Revie | ws | | |
| 1 Star | 2 Stars | 3 stars | 4 stars | 5 stars |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Add Nev | w Review | | | |



URL to GitHub Repository:

https://github.com/JoleneMel/Review.or.ReBOO/tree/onward