

Web App Design with React Final Project

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 an online API of your choice, or multiple APIs (or if no API can be found, using an array for in-memory storage is okay as well), create a React project of your choice. You will be working on this for the next three weeks.
- 2. Project must meet the following criteria:
 - a. Use React Router and have at least 3 pages
 - **b.** Use React Bootstrap or an alternative styling library
 - **c.** Contain at least 10 components
 - **d.** Allow for all CRUD operations

Screenshots of Code:



```
## Apply | X

## St. | S. Apply | X

## St. |
```



```
ACCOMENS X

WRY DE COMMENS NOT COMMENSATE AND THE SECRET OF THE SECRET O
```



```
const CookiesPage = () => {
       const match = useRouteMatch();
       const findCookieById = (id) =>
         cookiesMenu.filter((cookie) => cookie.id == id)[0];
       return (
             {cookiesMenu.map((cookie, index) => {
                 <Alert key={index} variant="primary">
                   <Link to={`${match.url}/${cookie.id}`}>
                    {cookie.name}
             <Route
               path={`${match.path}/:cookieMenuId`}
               render={(props) => (
                   {...props}
                   data={findCookieById(props.match.params.cookieMenuId)}
             <Route path={match.path}>
              <h3>Please Select an Item.</h3>
function Cookie(props) {
  const {data} = props;
   return data == undefined ? <h1>404 Not Found</h1> :(
       <Card.Header>{data.name}</Card.Header>
        <Card.Subtitle>{data.name}</Card.Subtitle>
         <Card.Text>{data.description}</Card.Text>
       </Card.Body>
export default CookiesPage
```



```
### Support React from "react-)

### Support React from "react-)

### import React from "react-bootstrap/Card";

### import Card from "react-bootstrap/Card";

### import Card from "react-bootstrap/Alert";

### import Card from "react-bootstrap/Alert";

### Switch,

### Switch,

### Switch,

### Const cupcakesMenu = [

### id: 1,

### name: "Cookies N' Cream",

### description: "Double creme filled chocolate cake and OREO" Cookie Frosting!",

### id: 2,

### name: "Cookies N' Cream",

### description: "Sweet lemon cake filled with tangy whipped cream, topped in smooth cream cheese rimmed in crumbled lemon shortbread."

### id: 2,

### description: "Sweet lemon cake filled with tangy whipped cream, topped in smooth cream cheese rimmed in crumbled lemon shortbread."

### id: 3,

### description: "Sweet lemon cake filled with tangy whipped cream, topped in smooth cream cheese rimmed in crumbled lemon shortbread."

### id: 4,

### name: "Mainight Magic",

### description: "Sweet peach cake frosted with simply peach buttercream and a handful of tasty chocolate chips."

### id: 4,

### name: "Peach Blossom",

### description: "Sweet peach cake frosted with simply peach buttercream."

### id: 5,

### name: "Peach Blossom",

### description: "Sweet peach cake frosted with simply peach buttercream."

### id: 5,

### name: "Scarlett's Red Velvet",

### description: "Rich scarlet red chocolate cake filled with creamy vanilla topped with our classic cream cheese frosting."

### id: 6,

### id: 5,

### name: "Scarlett's Red Velvet",

### description: "Rich scarlet red chocolate cake filled with creamy vanilla topped with our classic cream cheese frosting."

### id: 6,

### id: 6,

### id: 7,

##
```



```
JS Cupcakes.js X
src > JS Cupcakes.js > ♡ Cupcake
       const CupcakesPage = () => {
              const match = useRouteMatch();
              const findCupcakeById = (id) =>
                cupcakesMenu.filter((cupcake) => cupcake.id == id)[0];
                <div className="cupcakebody">
                  <h2>Cupcakes</h2>
                     {cupcakesMenu.map((cupcake, index) => {
                      return (
                        <Alert key={index} variant="primary">
                          <Link to={`${match.url}/${cupcake.id}`}>
                            {cupcake.name}
                        </Alert>
 60
                      path={`${match.path}/:cupcakesMenuId`}
                      render={(props) => (
                        KCupcake
                           data={findCupcakeById(props.match.params.cupcakesMenuId)}
                     <Route path={match.path}>
                      <h3>Please Select an Item.</h3>
       function Cupcake(props) {
          const {data} = props;
           return data == undefined ? <h1>404 Not Found</h1> :(
 80
              <Card.Header>{data.name}</Card.Header>
            <Card.Subtitle>{data.name}</Card.Subtitle>
 85
                <Card.Text>{data.description}</Card.Text>
              </Card.Body>
            </Card>
      export default CupcakesPage;
```



```
## SP Double | ## Double | ##
```



```
JS Donuts.js X
src > JS Donuts.js > [@] DonutsPage
       const DonutsPage = () => {
              const match = useRouteMatch();
              const findDonutById = (id) =>
               donutMenu.filter((donut) => donut.id == id)[0];
                    {donutMenu.map((donut, index) => {
                      return (
                        <Alert key={index} variant="primary">
                          <Link to={`${match.url}/${donut.id}`}>
                            {donut.name}
                        </Alert>
                    <Route
                      path={`${match.path}/:donutMenuId`}
                      render={(props) => (
                          {...props}
                          data={findDonutById(props.match.params.donutMenuId)}
                    <Route path={match.path}>
                     <h3>Please Select an Item.</h3>
      function Donut(props) {
         const {data} = props;
          return data == undefined ? <h1>404 Not Found</h1> :(
              <Card.Header>{data.name}</Card.Header>
              <Card.Body>
                <Card.Subtitle>{data.name}</Card.Subtitle>
                <Card.Text>{data.description}</Card.Text>
                <Review />
      export default DonutsPage
```



```
import { uuid } from "uuidv4";
export const Form = ({
  }) => {
  const handleChange = (e) => {
       const { name, value } = e.target;
setForm({ ...form, [name]: value });
    const handleSubmit = (e) => {
     e.preventDefault();
      setReviews([...reviews, form]);
setForm({ item: "", review: "", id: uuid() });
    const handleUpdate = (e) => {
       e.preventDefault();
       setEditing(false);
       const updatedReviews = reviews.map((review) =>
       setReviews(updatedReviews);
setForm({ item: "", review: "", id: uuid() });
     return (
        <form className="form" onSubmit={editing ? handleUpdate : handleSubmit}>
               <input type="text" placeholder="Menu Item" id="item" name="item" value={form.item} onChange={handleChange} />
       <label htmlFor="Review">Review</label>
ctextarea placeholder="Leave a review." id="review" name="review" onChange={handleChange} /

<button type="submit">{editing ? "Update" : "Submit"}</button>
export default Form:
```



```
JS review-list.js X
src > components > review components > Js review-list.js > ...
      import React from 'react'
       import ReviewItem from "./ReviewItem"
       const ReviewList = ({ reviews, setEditing, setForm, setReviews }) => {
           const handleDelete = (id) => {
             const updatedReviews = reviews.filter((review) => review.id !== id);
             setReviews(updatedReviews);
           const handleEdit = (id) => {
            const editedPost = reviews.filter((review) => review.id === id);
             setForm(editedPost[0]);
            setEditing(true);
          return (
            <div className="reviewList">
              {reviews.map((review) => (
                   key={review.id}
                   review={review}
                   handleDelete={handleDelete}
                   handleEdit={handleEdit}
         export default ReviewList;
```





```
border: 1px solid | #0d0d0d;
box-shadow: 1px 1px | #0d0d0d, 2px 2px | #0d0d0d;
padding: 1rcm;
width: 20rcm;
reviewItem {
border: 1px solid □ meddedd;
bor-shadow: 1px 1px □ meddeddd, 2px 2px □ meddedd;
padding: 1rem;
wigth: 20-rem;
margin-top: 1rem;
```

Screenshots of Running Application:

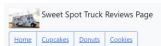




Sweet Spot Truck Reviews Page

Welcome to the Sweet Spot Truck reviews website! This is a website for the Sweet Spot Truck customers that would like to leave a review of their products!





Cupcakes

Cookies N' Cream

Lemon Crumble

Midnight Magic

Peach Blossom

Scarlett's Red Velvet

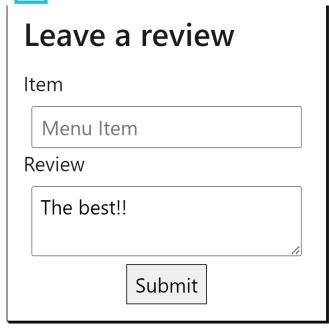
Please Select an Item.



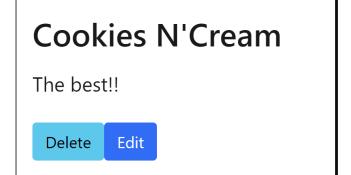


Cookies N' Cream Double creme filled chocolate cake and OREO® Cookie Frosting! Item Review Leave a review Item Menu Item Review Leave a review. Submit







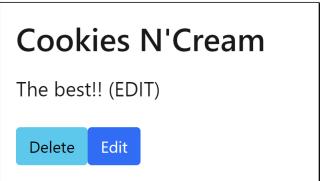




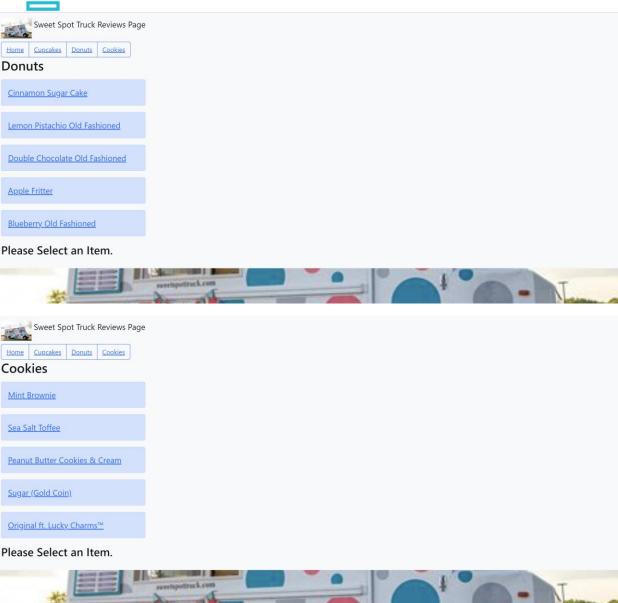
Item Review











URL to GitHub Repository:

https://github.com/JoanaBarao7/React_Week4_6