



Week 16: Final Project

URL to GitHub Repository:

<https://github.com/Rcruzn33/movieSearchReactRouter.git>

URL to Your Coding Assignment Video:

<https://youtu.be/9TRA1mlAz4c>

Instructions:

- In Visual Studio Code, write the code that accomplishes the objectives listed below and ensures that the code compiles and runs as directed.
- Create a new repository on GitHub for this week's assignments and push this document, with your project code, to the repository.
- Include the URLs for this week's repository and video where instructed. Submit this document as a .PDF file in the LMS.

Coding Steps:

- Using an online API of your choice (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.
- Project must meet the following criteria:
 - Use React Router and have at least 3 pages using React Bootstrap or an alternative styling library
 - Contain at least 10 components
 - Allow for all CRUD operations



Week 16: Final Project

Video Steps:

- Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed.
- This video should be done using screen share and voice over.
- This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend.
 - You can create a new meeting, start screen sharing, and start recording.
 - This will create a video recording on your computer.
- This should then be uploaded to a publicly accessible site, such as YouTube.
 - Ensure the link you share is **PUBLIC** or **UNLISTED**!
 - If it is not accessible by your grader, your project will be graded based on what they can access.



PROMINEO TECH

Week 16: Final Project

```
//HTTP Requests to communicate with API
class MoviesApi {
  get = async () => {
    try {
      const resp = await fetch(MOVIE_ENDPOINT);
      const data = await resp.json();
      return data;
    } catch(e) {
      console.log('Oops, looks like fetchMovies had an issue.', e);
    }
  }

  put = async (movie) => {
    try {
      const resp = await fetch(`${MOVIE_ENDPOINT}/${movie.id}`, {
        method: 'PUT',
        headers: {
          'Content-Type': 'application/json'
        },
        body: JSON.stringify(movie)
      });
      return await resp.json();
    } catch(e) {
      console.log('Oops, looks like updating movies had an issue.', e);
    }
  }

  post = async (movie) => {
    try {
      const resp = await fetch(`${MOVIE_ENDPOINT}`);
    }
  }
}
```

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully
Terminate batch job (Y/N)? y

PS C:\Users\frsen\OneDrive\Desktop\routerbootstrap-react> fwd-i-search: _

```
import React, { useState, useEffect } from 'react';
import { BrowserRouter as Router, Link, Switch, Route } from 'react-router-dom';
import { moviesApi } from './MoviesApi';
import SearchResults from './search-Results';
import Container from 'react-bootstrap/container';
import Button from 'react-bootstrap/button';
import MoviePage from './Movie';
import styles from './Movies.module.css';
import AddMovies from './addMovies';

//fetch async function that make API calls
const MainPage = () => {
  const [movies, setMovies] = useState([]);

  const fetchMovies = async () => {
    const moviesData = await moviesApi.get();
    setMovies(moviesData);
  };

  const updateMovie = async (updatedMovie) => {
    await moviesApi.put(updatedMovie);
    fetchMovies();
  };

  const addMovie = async (newMovie) => {
    await moviesApi.post(newMovie);
    fetchMovies();
  };
}
```

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully
Terminate batch job (Y/N)? y

PS C:\Users\frsen\OneDrive\Desktop\routerbootstrap-react> fwd-i-search: _



PROMINEO TECH

Week 16: Final Project

```
src > Components > JS Main-Page.js > MainPage
useEffect(() => {
  fetchMovies();
}, []);

//react router to navigate to different paths (web pages)
return (
  <div className={styles.movies}>
    <Container>
      <Router>
        <Button variant="outline-secondary">
          <Link to="/search/:query">Search</Link>
        </Button>
        <Button variant="outline-secondary">
          <Link to="/mainPage">Movie List</Link>
        </Button>
        <Switch>
          <Route path="/search/:query">
            <SearchPage movies={movies}/>
          </Route>
          <Route path="/movies/:movieId">
            <MoviePage movies={movies} updateMovie={updateMovie} deleteMovie={deleteMovie} addMovie={addMovie}/>
          </Route>
          <Route path="/mainPage">
            <AddMovies movies={movies} addMovie={addMovie} deleteMovie={deleteMovie}/>
          </Route>
        </Switch>
      </Router>
    </Container>
  </div>
);
```

webpack compiled successfully
Terminate batch job (Y/N)? y

```
src > Components > JS addMovies.js > AddMovies
<div>
  <h2 className={styles.white}>Year:</h2>
  <input
    type="text"
    value={newMovieYear}
    onChange={(e) => setNewMovieYear(e.target.value)}
  />
</div>
<button type="submit" className="btn btn-primary">Add Movie</button>
<h2 className={styles.red}> Movies List</h2>
<div>
  <div>
    <Card key={movie.id}>
      <Card.Body>
        <Card.Title>{movie.Title}</Card.Title>
        <Card.Title>{movie.Year}</Card.Title>
        <Button
          variant="danger"
          onClick={() => handleDeleteMovie(movie.id)} //deleteMovie
        >Delete
        </Button>
      </Card.Body>
    </Card>
  </div>
</div>
</form>
</div>
```

webpack compiled successfully
Terminate batch job (Y/N)? y



PROMINEO TECH

Week 16: Final Project

```
src > Components > JS search-Results.js > [0] filteredMovies > [0] movies.filter() callback
1 import React, { useState } from 'react';
2 import { useHistory, useParams, Link } from 'react-router-dom';
3 import styles from './Movies.module.css';
4
5 //initiating states
6 function SearchPage(props) {
7   const [movies] = props;
8   const [searchQuery, setSearchQuery] = useState('');
9   const history = useHistory();
10   const { query } = useParams();
11
12   //searchQuery being handled and pushed to end of url
13   const handleSearch = (e) => {
14     e.preventDefault();
15     history.push(`/search/${searchQuery}`);
16   };
17
18   //searchQuery being set to input value
19   const handleInputChange = (e) => {
20     setSearchQuery(e.target.value);
21   };
22
23   // Filter movies based on search query
24   const filteredMovies = movies.filter((movie) => {
25     movie.Title.toLowerCase().includes(query.toLowerCase());
26   });
27
28   return (
29     <div>
```

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**
Terminate batch job (Y/N)? y

PS C:\Users\frsen\OneDrive\Desktop\routerbootstrap-react> fwd-i-search: _

```
src > Components > JS Movie.js > MoviePage
1 //this class is to show a new review
2 const addNewReview = (reviews) => updateMovie({...movie, reviews: [...movie.reviews, reviews]});
3
4 //jsx that uses useRouteMatch to link and route to ReviewForm. Also renders movie page and object
5 return (
6   <Card style={{backgroundColor: "black", backgroundSize: "cover", backgroundPosition: "center", backgroundRepeat: "no-repeat", position: "fixed",
7     <Card.Header>
8     <Card.Title>{movie.Title}</Card.Title>
9     <Card.Subtitle>{movie.Year}</Card.Subtitle>
10    </Card.Header>
11    <Card.Body>
12    <div className="row">
13      <div className="col">
14        <img src={movie.Poster} alt={movie.Title} />
15      </div>
16      <div className="col">
17        <Link to={`/${match.url}/reviews`} >
18          <NewReviewForm
19            movie={movie}
20            addNewReview={addNewReview}
21            deleteMovie={deleteMovie}
22            updateMovie={updateMovie}
23            addMovie={addMovie}
24          />
25        </Link>
26        <Switch>
27          <Route path={`/${match.path}/reviews/:reviewId`} >
28            <NewReviewForm
29              movie={movie}
30            />
31          </Route>
32        </Switch>
33      </div>
34    </div>
35  </Card.Body>
36 </Card>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
```

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**
Terminate batch job (Y/N)? y

PS C:\Users\frsen\OneDrive\Desktop\routerbootstrap-react> fwd-i-search: _



PROMINEO TECH

Week 16: Final Project

```
src > Components > ReviewForm.js > [0] NewReviewForm
1 import React, { useState } from "react";
2 import StarRating from "../Star-Review";
3 import Card from "react-bootstrap/Card";
4 import styles from "../Movies.module.css";
5
6
7 //sets the state values
8 const NewReviewForm = ({ movie, addNewReview, updateMovie }) => {
9   const [allReviews, setAllReviews] = useState(movie.reviews || []);
10  const [reviews, setReviews] = useState("");
11  const [rating, setRating] = useState(0);
12  const [allRatings, setAllRatings] = useState([]);
13
14
15  //uses PUT to update removal of review
16  const deleteReview = (reviewId) => {
17    console.log("delete")
18    const updatedReviews = allReviews.filter(
19      (reviews) => reviews.id !== reviewId
20    );
21    updateMovie({ ...movie, reviews: updatedReviews });
22    setAllReviews(updatedReviews);
23  };
24  console.log(allReviews);
25
26  //handles submission of the review, uses PUT to update reviews property in API
27
28  const onSubmit = (e) => {
29    console.log("submit")
30  };
31
32  Note that the development build is not optimized.
33  To create a production build, use npm run build.
34
35  webpack compiled successfully
36  Terminate batch job (Y/N)? y
37  PS C:\Users\frsn\OneDrive\Desktop\routerbootstrap-react>
38  fwd-i-search: _
```

```
src > Components > ReviewForm.js > [0] NewReviewForm
81
82 Add Review
83 </button>
84 </div>
85 {/*renders the review object and review list*/}
86 allReviews.map((reviews) => {
87   <Card key={reviews.id}>
88     <Card.Header>User</Card.Header>
89     <Card.Body>
90       <StarRating rating={reviews.rating}/>
91       <Card.Text>{reviews.reviews}</Card.Text> {/*review component*/}
92     <button
93       className="btn btn-danger"
94       onClick={() => deleteReview(reviews.id)}
95     >
96       Delete
97     </button>
98   </Card.Body>
99 </Card>
100   )}
101 </div>
102 </form>
103 </div>
104
105 );
106
107 export default NewReviewForm;
108
109
110 Note that the development build is not optimized.
111 To create a production build, use npm run build.
112
113 webpack compiled successfully
114 Terminate batch job (Y/N)? y
115 PS C:\Users\frsn\OneDrive\Desktop\routerbootstrap-react>
116 fwd-i-search: _
```



PROMINEO TECH

Week 16: Final Project