**React StreamList App Setup**

Eric F. Vara

The University of Arizona Global Campus

INT 499: Capstone for Information Technology

Professor Miguel Kanto

July 29, 2024

**React StreamList App Setup**

Creating the StreamList app was an interesting project that combined several parts of web development using React. This paper will explain my experience building the navigation bar and the form component. I’ll share the steps I took, how I used React Router, and any challenges or insights I had along the way.

**Navigation Bar and React Router**

**Steps to Create the Navigation Bar**

To make it easy for users to move between different pages (like the homepage, Movies, Cart, and About pages), I created a navigation bar. Here’s how I did it:

1. Set Up React Router: I installed react-router-dom to manage the navigation.
   1. npm install react-router-dom
2. Create the Navigation Component: I made a Navigation component with links to different pages.
3. Add Routes in the App Component: I set up routes in the main App component to show different components based on the URL path.

**Experience with React Router**

Using React Router made it easy to manage the different pages of the app. It dynamically loads the right component based on the URL, which makes the app feel smooth and responsive.

**Form Component**

**Steps to Build the Form Component:**

The form component is where users can add movies to their list. Here’s what I did:

1. State Management: I used React's useState hook to keep track of the movie input and the list of movies.
2. Handle Input and Submit: I created functions to handle changes to the input field and to add the movie to the list when the form is submitted.

**Props Used**

In this component, I didn’t use any props because everything was managed within the component itself. However, in the future, we could use props to pass data from a parent component or to interact with a backend service.

**Challenges and Insights**

**Security Challenges**

CSS Styling: Making sure the CSS matched the provided theme was a bit challenging. I used CSS variables and modern layout techniques to get it right.

Responsive Design: Ensuring the app looked good on all devices required careful use of media queries.

**Insights**

Component Reusability: Breaking the app into smaller, reusable components made the code easier to manage and understand.

React Router: I learned a lot about how powerful React Router is for managing navigation in single-page applications.

**Conclusion**

Building the StreamList app involved setting up a navigation system and a functional form component. React Router helped manage the navigation, and state management in React made the form component work smoothly. This project gave me valuable experience with modern web development practices.

**GitHub Link**

<https://github.com/EricIVara/EZ_TechMovie>

**References**

Ayoola, React Documentation. (n.d.). Retrieved from <https://reactjs.org/docs/getting-started.html>

W3Schools. (n.d.). CSS Media Queries. Retrieved from [https://www.w3schools.com](http://www.w3schools.com/css/css3_mediaqueries.asp)