Documentation for the Star Rating Component

This documentation provides an overview of the **Star Rating Component** implemented in React. The component allows users to rate a movie by selecting stars, with dynamic updates based on user interactions.

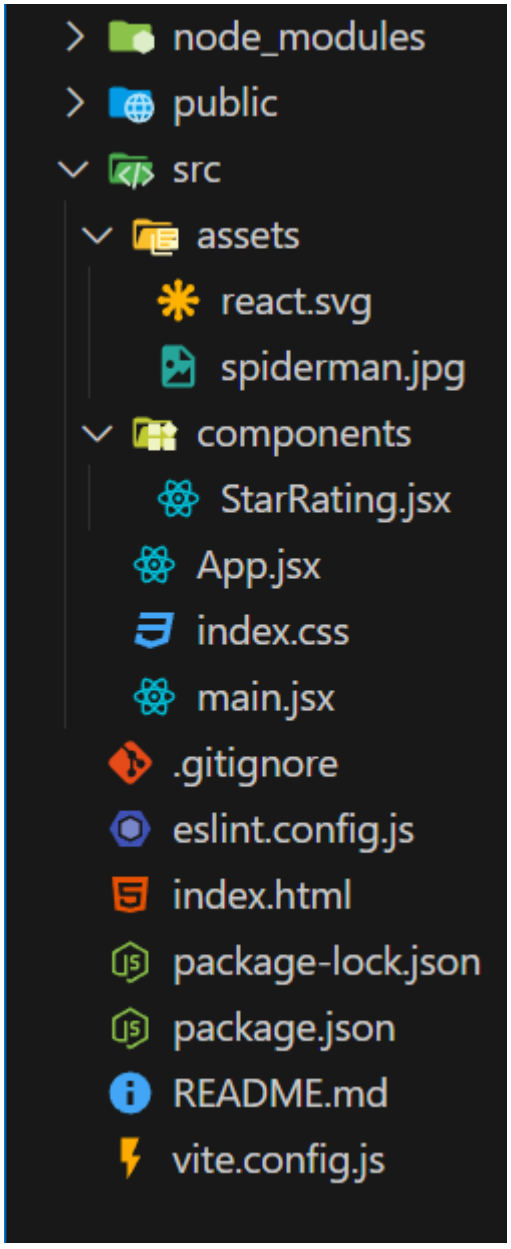
Component Overview

The StarRating component is a React functional component that renders a star rating interface. Users can click on stars to rate a movie, and the selected rating is displayed below the stars. The component also includes hover effects to preview the rating before selection.

Key Features

1. **Interactive Stars**: Users can click on stars to rate a movie.
2. **Hover Effect**: Stars temporarily highlight when hovered over, providing a preview of the rating.
3. **Dynamic Rating Display**: The selected rating is displayed below the stars in the format X/5 Rating.
4. **Reusable Component**: The component is designed to be reusable and can be easily integrated into other parts of the application.

Folder Structure



Code Breakdown

1. Imports

```
import { useState } from "react";

import spiderman from "../assets/spiderman.jpg";
```

- **useState**: A React hook used to manage the component's state (selected rating and hover state).
- **spiderman**: An image imported from the assets folder, used as a placeholder for the movie being rated.

2. State Management

```
const [rating, setRating] = useState(0);

const [hover, setHover] = useState(0);
```

- **rating**: Stores the current selected rating (0 by default).
- **hover**: Stores the hover state to temporarily highlight stars when the user hovers over them.

3. Event Handlers

```
const handleSetRating = (star) => {
  setRating(star);
};
```

- **handleSetRating**: This function updates the rating state when a user clicks on a star.

4. JSX Structure

```
return (
  <div className="card">
    <div className="container">
      <img src={spiderman} alt="" width={250} height={300} />
```

```
    <h1 className="heading">Rate Movie</h1>
    <div className="star-container">
      <div className="star-rating">
        {[1, 2, 3, 4, 5].map((star) => (
          <span
            key={star}
            className="star"
            onClick={() => handleSetRating(star)}
            onMouseEnter={() => setHover(star === 5 ? 5 : 0)}
            onMouseLeave={() => setHover(0)}
            style={{
              fontSize: "50px",
              cursor: "pointer",
                color: hover >= star || rating >= star ? "gold" : "grey"
            }}
          >
            ★
          </span>
        ))}
        <p className="rating-text">{rating}/5 Rating</p>
      </div>
    </div>
  </div>
);
```

- **card**: The main container for the component, centered on the page.

- **container**: A sub-container that holds the movie image, heading, and star rating interface.
- **star-rating**: A flex container that holds the 5 stars.
- **star**: Each star is rendered using the .map() method. The color of the star changes based on the hover and rating states.
- **rating-text**: Displays the selected rating below the stars.

5. CSS Styling

```
body {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  overflow: hidden;
}

.container {
  width: 450px;
  height: 550px;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color: black;
  border-radius: 10px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.star-rating {
```

```
  display: flex;
  flex-direction: row;
  justify-content: center;
  align-items: center;
  gap: 20px;
}

.rating-text {
  color: white;
  font: bold;
}

.heading {
  text-align: center;
  margin-top: 20px;
  color: blue;
}

.heading + h4 {
  color: green;
}

.card {
  width: 100%;
  display: flex;
  justify-content: center;
```

```
  align-items: center;
  min-height: 100vh;
  background-color: red;
}
```

- **body**: Resets default margin, padding, and box-sizing for the entire page.
- **container**: Styles the container that holds the movie image, heading, and star rating.
- **star-rating**: Styles the flex container for the stars.
- **rating-text**: Styles the text that displays the selected rating.
- **heading**: Styles the heading text.
- **card**: Styles the main card container, centering it on the page.

Functionality

1. **Clicking a Star**:
    - When a user clicks on a star, the handleSetRating function is called, updating the rating state.
    - The stars are dynamically rendered using .map(), and the color of the stars changes based on the rating state.
2. **Hover Effect**:
    - When a user hovers over a star, the hover state is updated, temporarily highlighting the stars.
    - The hover effect is removed when the mouse leaves the star area.
3. **Rating Display**:
    - The selected rating is displayed below the stars in the format X/5 Rating.

Usage

To use the StarRating component, simply import and include it in your React application:

```
import StarRating from "./components/StarRating";


function App() {
  return (
    <div>
      <StarRating />
    </div>
  );
}


export default App;
```

Dependencies

- **React**: The component uses React hooks (useState) for state management.
- **CSS**: Custom CSS is used for styling the component.

Limitations

- The component currently supports only 5 stars.
- The hover effect is limited to the 5th star due to the condition star === 5 ? 5 : 0.

Future Improvements

1. **Customizable Number of Stars**: Allow the component to accept a prop for the number of stars (e.g., 10 stars).

2. **Improved Hover Logic**: Enhance the hover effect to work for all stars, not just the 5th star.
3. **Accessibility**: Add ARIA labels and keyboard navigation for better accessibility.