Star Rating Component - Documentation

**Overview**

The **Star Rating Component** allows users to **rate items by selecting stars** dynamically. Users can hover over stars to preview their rating before selecting, and they can change their rating by clicking another star.

**Features**

✅ **Interactive Star Selection** – Users can select a rating by clicking stars.  
✅ **Live Rating Preview** – Hovering over stars shows a temporary preview before selection.  
✅ **Dynamic Updates** – Users can change their rating at any time.  
✅ **Accessible & Responsive** – Keyboard-friendly and visually clear UI.

**Technology Stack**

* **React (Vite)** – Fast and modern frontend framework.
* **React Hooks** (useState) – For managing rating state.
* **CSS** – For styling interactive UI elements.
* **React Icons** – For star icons (react-icons/ai).

**Installation & Setup**

1. **Clone the Repository**
2. **Install Dependencies**
3. **Run the Application**
4. **Open the App in the Browser**

**Project Structure**

/src

│── components

│ ├── StarRating.js # Star Rating Component

│── App.js # Main Application File

│── index.js # React Entry Point

│── styles.css # Styling for the Component

* **components/** – Contains the **StarRating.js** component.
* **App.js** – Main application file where StarRating is used.
* **styles.css** – Provides a visually appealing design.

**Functionality**

**User Interaction**

* **Clicking a Star**: Updates the rating and fills all previous stars.
* **Hovering Over Stars**: Temporarily highlights stars to preview selection.
* **Mouse Leaves**: Restores the last clicked rating.

**UI Behavior**

* Displays **5 interactive stars** (default).
* Shows the **selected rating** (e.g., 3 / 5 Rating).

**State Management**

* rating (stores the selected rating).
* hover (stores the hovered star for preview effect).

**Test Cases**

**✅ Test Case 1: User clicks on the 4th star.**

**Expected Output:**  
⭐️⭐️⭐️⭐️☆ → **4 / 5 Rating**

**✅ Test Case 2: User clicks on the 2nd star after selecting 4 stars.**

**Expected Output:**  
⭐️⭐️☆☆☆ → **2 / 5 Rating**

**✅ Test Case 3: User hovers over the 5th star but does not click.**

**Expected Behavior:**

* UI temporarily highlights **5 stars**.
* Once the mouse moves away, the rating **returns to the last clicked value**.

**Constraints & Best Practices**

⚡ **Use useState** to manage rating and hover effects.  
⚡ **Use .map()** to dynamically generate star elements.  
⚡ **Ensure Accessibility (aria-label)** for better usability.  
⚡ **Use onMouseEnter & onMouseLeave** to handle hover effects.  
⚡ **Use transition in CSS** for smooth animations.

**Future Enhancements**

✨ **Allow Half-Star Ratings** – Implement fractional ratings (e.g., 3.5 stars).  
✨ **Keyboard Accessibility** – Enable rating using arrow keys.  
✨ **Customizable Star Count** – Allow dynamic star numbers (e.g., 10-star rating).  
✨ **Persistent Ratings** – Save ratings in local storage or database.

**Conclusion**

The **Star Rating Component** is a reusable and interactive UI element that enhances user experience. It efficiently handles **state updates, animations, and real-time user feedback** while ensuring optimal performance.