⭐ Star Rating Component

💻 **Difficulty:** Easy
🔍 **Category:** React, UI Interaction

Problem Statement:

You are required to implement a **Star Rating Component** where users can rate by selecting stars. The rating should be dynamically updated based on user interactions.

Your component should meet the following functionality:

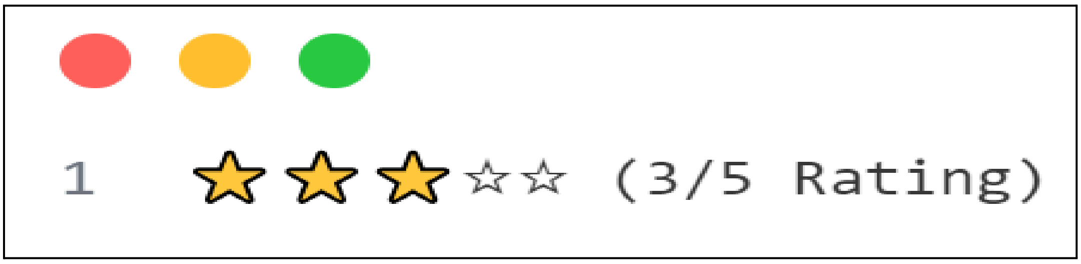
- Display **5 clickable stars** for rating.
- Clicking a star should highlight it and all previous stars.
- Allow users to **change their rating** by clicking another star.
- Show the **selected rating** (e.g., `3/5 Rating`) below the stars.
- Ensure stars change **on hover** before selection.

Input Format:

- **User Interaction:** The user clicks on a star (1 to 5).
- **Hover Effect:** The stars should visually change on hover before selection.

Output Format:

- A UI displaying **5 interactive stars**.
- Below the stars, the **selected rating** should be displayed in the format

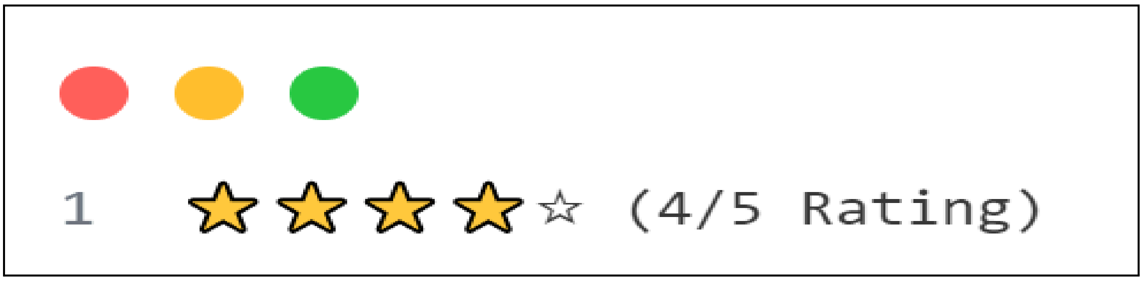


```
⭐⭐⭐☆☆ (3/5 Rating)
```

Example Test Cases:

Test Case 1:

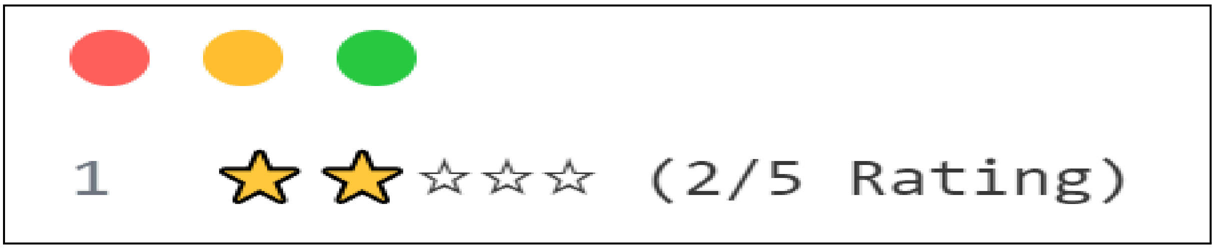
✅ **User clicks on the 4th star.**
🔹 **Expected Output:**



```
1   ⭐⭐⭐⭐☆ (4/5 Rating)
```

Test Case 2:

✅ User clicks on the 2nd star after selecting 4 stars previously.
🔹 Expected Output:



```
1   ⭐⭐☆☆☆ (2/5 Rating)
```

Test Case 3:

✅ User hovers over the 5th star but does not click.
🔹 Expected Behavior:

- The UI should temporarily highlight 5 stars.
- Once the mouse moves away, the rating should return to the last clicked value

Constraints:

- Use `useState` to manage the selected rating.
  Use `onMouseEnter` for hover effects.
  Use `.map()` to dynamically generate stars.
  Use CSS or icons (e.g., FontAwesome, Unicode) for styling.
  Ensure a well-structured and reusable component.

Hints:

💡 Store the selected rating in state using `useState`.
💡 Implement a hover state to preview ratings before clicking.
💡 Use an array with `.map()` to render star elements dynamically.
💡 Handle click and hover separately for smooth UX.