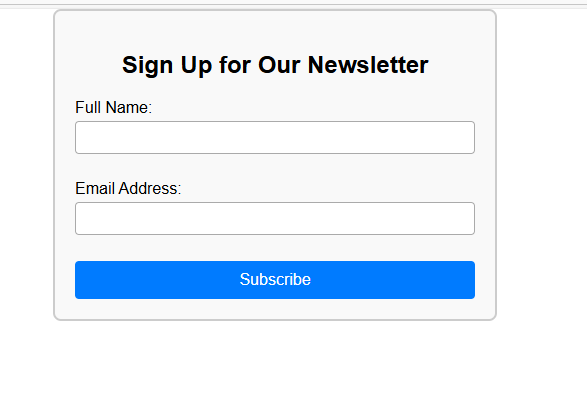
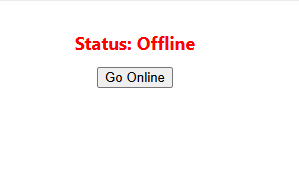
**Exercise 3 – React JS  
(Forms, Events, Conditional Rendering, CSS)**

1. Design and develop a functional component called NewsletterSignup. The form should include fields for full name and email address. Use useState to control the form inputs. On form submission, display a thank-you message using conditional rendering. Style the form using an external CSS file with padding, border, and hover effects.

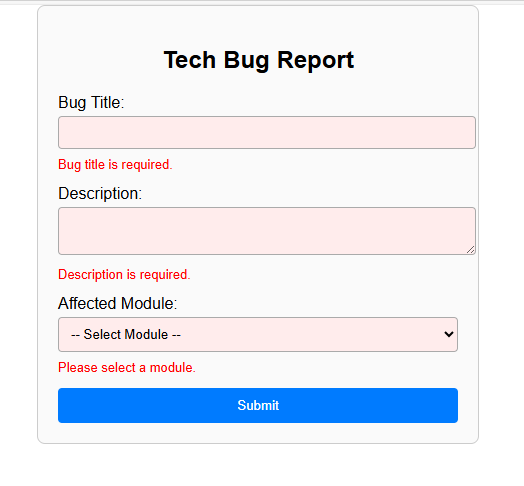
****

1. Design and develop a functional component named UserStatusSwitcher that toggles a user's status between “Online” and “Offline” using a button. Use useState to manage status and onClick to update it. Display the current status with conditional rendering. Style the status text with inline CSS (e.g., green for online, red for offline).



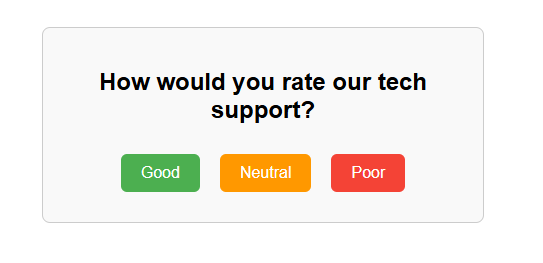
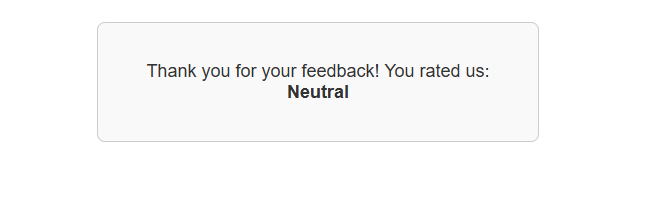
1. Design and develop a functional component called TechBugReportForm. The form should include the following fields:
   * Bug Title (text)
   * Description (textarea)
   * Affected Module (dropdown: e.g., UI, API, Database, Network)

Validate that all fields are filled before submission. Use conditional rendering to show inline error messages if any field is empty and display a submission success message otherwise. Apply external CSS to organize the form and highlight input errors.

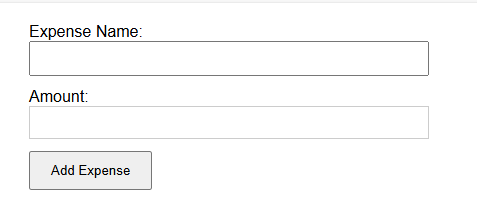


**Exercise 4 – React JS  
(Forms, Events, Conditional Rendering, CSS)**

1. Design and develop a functional component named FeedbackPoll. Display a question such as “How would you rate our tech support?” with three buttons: Good, Neutral, Poor. When a user clicks one, use conditional rendering to show a thank-you message including their selected choice. Use external CSS to style the poll area, buttons, and feedback message.

1. Design and develop a functional component called ExpenseTrackerInput. Include inputs for expense name and amount. Use useState for form control. Validate that the amount is a positive number. On form submission, show a success message or an error message using conditional rendering. Use inline CSS to highlight the amount field in red if validation fails.



1. Design and develop a functional component named ThemeSelector. Provide two radio buttons: Light Mode and Dark Mode. Use useState to track the selected theme. Conditionally render a preview box styled with the appropriate theme (dark or light). Use external CSS classes to apply theme-based background and text styling.

