**React tasks covering all topics**

Components & Props

1. Create a Welcome component that accepts a name prop and displays: “Hello, [name]!”

2. Build a ProfileCard component to display user details like name, age, and location using props.

3. Create a component that takes a list of hobbies as props and displays them as a list.

4. Pass a default prop value for a component that displays a user’s favourite color.

5. Create a parent component that passes data (e.g., a movie list) to a child component via props.

useState Hook

6. Build a counter with increment, decrement, and reset buttons using useState.

7. Create a toggle button that switches between "Light Mode" and "Dark Mode."

8. Build a form that updates a state variable for each input field in real time.

9. Create a component that adds a new task to a to-do list and displays the updated list.

10. Build a simple shopping cart where clicking “Add to Cart” increases a product’s count.

Event Handling

11. Add a button that displays an alert when clicked.

12. Build a form where clicking “Submit” displays the entered values in a console.

13. Create a component with an input field where typing updates a paragraph below it.

14. Implement a button that toggles the visibility of a paragraph.

15. Create a dropdown menu that displays the selected value when changed.

useEffect Hook

16. Fetch a random joke from an API and display it when the component loads.

17. Create a clock that updates every second using useEffect.

18. Track mouse coordinates and display them in real time on the screen.

19. Display a “Welcome Back” message only once when a user visits the page.

20. Use useEffect to log a message every time a counter is incremented.

Conditional Rendering

21. Create a login form that displays “Welcome, [username]!” after submission.

22. Build a weather app that shows different messages for sunny, rainy, or cloudy weather.

23. Display a “Loading…” message until data is fetched from an API.

24. Show a custom message when no tasks are available in a to-do list.

25. Build a component that displays different themes based on a selected value.

Lists & Keys

26. Create an array of colors and display them as a list.

27. Map through a list of students and display their names and grades.

28. Add a button to delete an item from a list using its index.

29. Display an array of objects (e.g., products) with their names and prices.

30. Create a searchable list where typing filters the displayed items.

Forms

31. Build a login form with username and password fields.

32. Create a form to accept user feedback and display it below after submission.

33. Build a registration form that validates the email format.

34. Create a form where users can add items to a shopping list.

35. Build a simple contact form with input fields and a submit button.

Custom Hooks

36. Create a custom hook for fetching data from an API.

37. Build a custom hook that tracks window width and height.

38. Create a hook that toggles a boolean value (e.g., useToggle).

39. Write a hook that fetches and caches user data to avoid unnecessary API calls.

40. Implement a hook that tracks whether a user is idle or active.

React Router

41. Set up routes for Home, About, and Contact pages in a React app.

42. Use useParams to fetch and display details of a selected product.

43. Create a navigation menu that highlights the active page.

44. Redirect users to a “404 Not Found” page for invalid routes.

45. Create a protected route that requires login to access certain pages.

State Management

46. Build a shopping cart using React Context for global state.

47. Implement a theme switcher using React Context and useReducer.

48. Create a basic Redux setup to manage a counter state.

49. Use Redux to fetch and store a list of users from an API.

50. Build a React app that manages tasks using Redux (add, edit, delete tasks).

Bonus (for fun)

Create a component that simulates typing a message one letter at a time.

Build a random quote generator that fetches quotes from an API.

Create a small quiz app with multiple-choice questions.

Build a Pomodoro timer with start, pause, and reset buttons.

Create a memory card game where cards flip when clicked.