Props:

1.Question: Create a React component called PersonCard that takes the following props:

.name: The name of the person (a string).

.age: The age of the person (a number).

.hobbies: An array of hobbies of the person.

The PersonCard component should render a card-like structure that displays the person's name, age, and a list of their hobbies.

2. Question: You are building a React application for a movie review website. You have a Movie component that displays information about a movie, including its title and rating. You also have a MovieList component that displays a list of movies. You want to pass movie data as props to the Movie component and render the list of movies in the MovieList component. Here's a code skeleton for the Movie and MovieList.

3. **Question:** You are building a React application for a bookstore. You have a **Book** component that displays information about a book, such as its title and author. The book data is passed to the **Book** component as props. Your task is to create a **Book** component and render it within a **BookList** component. The **Book** component should take the following props:

**title**: The title of the book (a string).

**author**: The author of the book (a string).

**isbn**: The ISBN (International Standard Book Number) of the book (a string).

**price**: The price of the book (a number).

You should display this information in a user-friendly way within the **Book** component. Additionally, you need to create a **BookList** component that renders multiple **Book** components with different book details.

4. **Question:** You are tasked with creating a simple React component for a weather app. The component, called **WeatherDisplay**, should receive weather data as props and display it. The weather data consists of the following properties: **city**, **temperature**, and **conditions**. Your goal is to create the **WeatherDisplay** component that can render this data in a user-friendly way.

Your task is to:

Create the **WeatherDisplay** component.

Accept the weather data (city, temperature, and conditions) as props.

Display the weather information in a nicely formatted manner, for example: "The weather in [city] is [temperature]°C with [conditions]."

Write the code for the **WeatherDisplay** component, and an example of how you would use it in a React application.

5.Question**:** Build a React application that simulates a simple online store. You have two components: Product and Cart. The Product component should display a list of products, and the Cart component should display the items added to the shopping cart.

You need to implement the following:

Create a Product component that receives an array of product objects as props. Each product object should have a name, price, and an "Add to Cart" button.

When a user clicks the "Add to Cart" button for a product, it should be added to the shopping cart. The Cart component should display the items in the shopping cart, along with the total price of all the items.

The Cart component should also receive a prop for the list of items in the cart.

You can assume that the basic structure of the React application and CSS styling is already set up. Your task is to implement the Product and Cart components and ensure that they communicate using props.

6.Question**:** Create a React application for a simple to-do list. You should have two components: **TodoList** and TodoItem. The TodoList component should display a list of to-do items, and the **TodoItem** component should represent each individual item.

Your tasks are as follows:

Create a TodoList component that receives an array of to-do items as props. Each to-do item should have a title and a boolean value indicating whether it's completed.

Display each to-do item in the TodoList component as a list. Use the title to display the item's text and apply a strikethrough style if the item is marked as completed.

Create a **TodoItem** component that represents an individual to-do item. It should receive a single to-do item as a prop.

Implement a way to toggle the "completed" status of a to-do item when it is clicked. This should update the component's state in the **TodoList**.

Map method:

**1.Question:** Build a React application that displays a list of tasks. You have a **TaskList** component that receives an array of task objects as props. Each task object should have an **id**, a **title**, and a **completed** status.

You need to implement the following:

Create the **TaskList** component that receives an array of task objects as props.

Inside the **TaskList** component, use the **map** method to render each task as a separate **Task** component.

Create a **Task** component that receives a single task object as props and displays the task's title and a checkbox indicating whether it's completed or not

When a user clicks on the checkbox in the **Task** component, it should toggle the **completed** status for that specific task in the list.

Bonus: Display the total number of completed tasks and the total number of tasks remaining in the list.

2. **question:** You are building a React application to display a list of tasks. You have an array of task objects, and you need to use the **map** method to render each task as a list item in a component. Your task is to implement this using React.

Create a React component called **TaskList** that receives an array of task objects as a prop. Each task object should have a **title** and a **completed** property.

Inside the **TaskList** component, use the **map** method to iterate through the array of tasks and render each task as a list item. If a task is marked as completed (**completed: true**), it should be displayed with a strikethrough style.

Add a button to each task that allows the user to mark a task as completed or uncompleted. When the button is clicked, it should toggle the **completed** property of the task.

Ensure that the component re-renders when a task is marked as completed or uncompleted, reflecting the updated state.

3. Question: You are building a React application that displays a list of books. You have an array of book objects, and you want to render them using the map function. Create a component called BookList that receives an array of book objects as a prop. The BookList component should render each book with its title and author.

4. Question: Create a React application that displays a list of items and allows the user to perform certain operations on the items using the map method. You should have two components: ItemList and Item. The ItemList component should receive an array of items as props and display them, and the Item component should display an individual item.

5. Question: Build a React application that displays a list of tasks. You have a TaskList component and a Task component. The TaskList component should receive an array of task objects as props. Each task object should have a name and a status (e.g., "completed" or "incomplete").

useState

1.Question: You are tasked with creating a simple counter application using React. Create a functional component called Counter that displays a counter value. Implement the component using the useState hook.

You need to do the following:

Create a Counter component using the functional component syntax.

Use the useState hook to manage the counter value.

Display the current counter value on the screen.

Implement buttons for "Increment" and "Decrement" that allow the user to increase or decrease the counter value.

Ensure that the counter value cannot go below 0 when decrementing.

Display the current counter value and update it when the "Increment" and "Decrement" buttons are clicked.

1. ****Question:**** You are developing a React application for managing a to-do list. Create a **Todo** component that allows users to add and delete tasks. Implement this component using the **useState** hook.

You need to do the following:Create a functional component called **Todo**.Implement a state variable, **tasks**, using the **useState** hook. Initially, it should be an empty array.Display a list of tasks from the **tasks** array.

Provide an input field and a "Add Task" button that allows users to add new tasks to the list.

Implement a "Delete" button next to each task that removes that specific task from the list.Your task is to create the **Todo** component using the **useState** hook to manage the task list, adding new tasks, and deleting existing ones.

This question assesses your ability to work with state in functional components, handle user interactions, and update component state using the **useState** hook in React.

3.Question: You are developing a React application for a to-do list. Create a Todo component that allows users to add and remove tasks. Use the useState hook to manage the list of tasks.

You need to do the following:

Create a functional component called Todo that maintains a list of tasks in its state using the useState hook.

Display the list of tasks in an unordered list (<ul>) within the component.

Implement an input field and a button that allows users to add new tasks to the list. When a new task is added, it should appear in the list.

Each task in the list should have a "Remove" button next to it. Clicking the "Remove" button should remove the respective task from the list.

Ensure that you use the useState hook correctly to manage the list of tasks and re-render the component when the list changes.

Your task is to create the Todo component using the useState hook for state management. You should demonstrate your understanding of how to use useState to manage and update state in a React component.

4.Question: You are working on a React application that allows users to toggle the visibility of a certain element. Create a ToggleButton component that uses the useState hook to manage the visibility state.

You need to do the following:

Create a functional component called ToggleButton that initially displays a button with the label "Toggle Element."

Use the useState hook to manage the visibility state of a <div> element within the ToggleButton component.

When the "Toggle Element" button is clicked, the <div> element should toggle its visibility. You can use CSS to hide and show the element.

Make sure that the component's behavior is controlled by the useState hook, and the text on the button should change to "Hide Element" when the <div> is visible and "Show Element" when it's hidden.Your task is to implement the ToggleButton component using the useState hook to manage the visibility state and handle user interactions.

5.Question: You are developing a React application that allows users to toggle the visibility of a text element. Create a ToggleText component that utilizes the useState hook to toggle the visibility of a text.

You need to do the following:

Create a functional component called ToggleText.

Inside the component, use the useState hook to maintain the state for the text visibility. Initially, the text should be hidden.

Display a button within the component that, when clicked, toggles the visibility of the text between hidden and visible.

Render the text element with a message like "This text is visible!" when the visibility state is true and hide it when the state is false.

Your task is to implement the ToggleText component using the useState hook to manage the visibility state and toggle the text's visibility when the button is clicked.

6.Question: You are developing a React application that allows users to like or dislike posts. Create a Post component that displays a post with a like button. Implement this component using the useState hook to manage the number of likes for each post.

You need to do the following:

Create a functional component called Post that receives a post object as a prop. The post object should have a title and an initial number of likes.

Use the useState hook to manage the number of likes for the post.

Display the post's title and the current number of likes within the component.

Implement a button that allows users to increase the number of likes when clicked. Update the state using the useState hook.

Ensure that the component can handle any valid post object passed as a prop and displays the post's information correctly.

7.Question: You are working on a React application that allows users to toggle a dark mode. Create a DarkModeToggle component that uses the useState hook to manage the dark mode state.

You need to do the following:

Create a functional component called DarkModeToggle.

Use the useState hook to manage a darkMode boolean state variable with an initial value of false.

Implement a button in the DarkModeToggle component that allows users to toggle between dark mode and light mode. Clicking the button should update the darkMode state.

Apply a CSS class to the DarkModeToggle component or the body of the document based on the darkMode state. For example, if darkMode is true, apply a dark theme; if darkMode is false, apply a light theme.

Ensure that the dark mode state is correctly managed and that the UI updates when the user toggles it.

8.Question: You are working on a React application that involves managing a user's profile information. Create a UserProfile component that allows a user to update their name and email. Implement this component using the useState hook.

You need to do the following:

Create a functional component called UserProfile.

Use the useState hook to manage the user's name and email as state variables.

Display input fields for the user's name and email within the component. Initially, these fields should be populated with the user's current name and email.

When the user changes the name or email in the input fields, update the corresponding state variables using useState.

Implement a "Save" button that, when clicked, should update the user's profile with the new name and email values. You can simulate this update by displaying an alert with the updated values.

9.Question: You are working on a React application for a to-do list. Create a Todo component that allows users to add and remove tasks. Implement the component using the useState hook to manage the task list.

You need to do the following:

Create a functional component called Todo.

Inside the Todo component, use the useState hook to manage the state of the task list. Initialize it as an empty array.

Display the list of tasks, which should be part of the component's state.

Provide an input field and a button to add new tasks. When a user enters a task and clicks the button, the task should be added to the list of tasks.

Each task in the list should have a "Delete" button next to it. When the "Delete" button is clicked, the respective task should be removed from the list

10.Question: You are tasked with creating a simple React application that allows users to like and dislike an item. Implement a LikeButton component using the useState hook to manage the like count.

You need to do the following:

Create a LikeButton component that displays a button and a count.

Initialize a state variable, likeCount, using the useState hook with an initial value of 0.

When the user clicks the button, the likeCount should increment by 1. If the user clicks again, the count should decrement by 1 (dislike).

Display the current likeCount value next to the button.

11.Question: You are tasked with creating a simple React application that allows users to toggle the visibility of a message. Implement a Message component that displays a message, and a button to show/hide the message using the useState hook.

You need to do the following:

Create a functional component called Message that initially displays a message, such as "Hello, World!"

Use the useState hook to manage the visibility of the message. The initial state should be visible.Display a button within the Message component that, when clicked, toggles the visibility of the message.

When the message is visible, it should be displayed. When it is hidden, it should not be visible on the screen.

Make sure to provide a clear and concise user interface that indicates whether the message is currently visible or hidden.

React Forms

1. create employee form in react.

add filed name= emp\_name, emp\_deparment, emp\_salary,emp\_email,emp\_position.

1. create Doctor form in react.

add any doctor releted filed.

3.create Registrion form in react.

add filed name=email, passowr, userame,phone\_number,confirm password

1.Question: You are developing a React application to manage a list of tasks. Create a TaskList component that receives an array of tasks as props. Each task is represented as an object with properties like id, title, and completed. Implement the TaskList component using the spread operator to filter and display tasks based on their completion status.

You need to do the following:

Create a functional component called TaskList that receives an array of task objects as props.

Use the spread operator to filter the tasks into two separate arrays: one for completed tasks and one for incomplete tasks.

Display the list of completed tasks and incomplete tasks in separate sections within the component.

Each task should have a title and a checkbox indicating its completion status.

The checkbox should toggle the completion status of a task when clicked.

2.Question: You are building a React application for managing user profiles. Create a UserProfile component that receives a user object as a prop. The user object contains information such as name, email, and additional details.

You need to do the following:

Create a functional component called UserProfile that receives the user object as a prop.

Use the spread operator (...) to display the user's basic information (name and email) and any additional details from the user object.

Ensure that the component can handle any valid user object passed as a prop and correctly displays the user's basic information and any additional details.

Implement a user interface that effectively displays the user's profile information.

1. Question: You are building a React application that involves managing a list of items. Implement a List component that receives an array of items as props. You need to use the spread operator to add a new item to the list and then display the updated list.You need to do the following:Create a functional component called List that receives an array of items as a prop.Use the spread operator to create a new array that includes the existing items from the prop.Add a new item to the array using the spread operator.Display the updated list of items in the component

4 .Question: You are developing a React application that displays a list of products. Implement a feature to allow users to add multiple items to their shopping cart. Create a Cart component that receives an array of selected items and displays their details. Use the spread operator to achieve this.

You need to do the following:

Create a Cart component that receives an array of selected products as a prop. Each product object should have properties like name, price, and quantity.

Use the spread operator to display the product details in the Cart component.

Calculate and display the total price for all the items in the cart.

Ensure that the Cart component can handle an array of products as a prop and display the details of each selected item using the spread operator.

5.Question: You are working on a React application that displays a list of items. Create a ItemList component that receives an array of items as a prop. Implement the ItemList component using the spread operator to render each item individually.

You need to do the following:

Create a functional component called ItemList that receives an items array as a prop.

Inside the ItemList component, use the spread operator to render each item in the items array individually.

Display the items in a list or any suitable format. You can assume that each item is a string.

Ensure that the component can handle any valid array of items passed as a prop and display them using the spread operator