**1. State Management - Counter Component**

**Description:**

Create a React component called Counter which has a button to increment and decrement the count. The count value should be managed using React's useState hook.

**Requirements:**

* The component should display the current count value.
* Create two buttons: one to increment and one to decrement the count.
* The initial count should be set to 0.
* The count value should update when the buttons are clicked.

**Bonus:**

* Add a reset button to set the count back to 0.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**2. State Management - Toggle Visibility**

**Description:**

Create a React component called ToggleVisibility that manages the visibility of a text paragraph. Initially, the paragraph should be hidden.

**Requirements:**

* Use the useState hook to manage the visibility state.
* Add a button that toggles the visibility of the paragraph.
* The text should say "Hello, world!".
* The button should display "Show" when the paragraph is hidden and "Hide" when the paragraph is visible.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**3. Passing Props from Parent to Child**

**Description:**

Create two components: Parent and Child. The Parent component should pass a name prop to the Child component, and the Child component should display that name.

**Requirements:**

* The Parent component should have a name variable (e.g., "Alice") in its state.
* The Parent component should pass this name to the Child component as a prop.
* The Child component should display "Hello, [name]" using the name prop.

A screenshot of a computer

AI-generated content may be incorrect.

**4. Passing Props and Handling Events**

**Description:**

Create a parent-child component interaction where the Parent component has a button, and clicking it updates a count. The count should be passed as a prop to the Child component, which displays the count.

**Requirements:**

* The Parent component should manage a count state (initial value 0).
* The Parent component should pass the count as a prop to the Child component.
* Add a button in the Parent component to increment the count by 1 each time it's clicked.
* The Child component should display the current count.

A screenshot of a computer

AI-generated content may be incorrect.

**5. State Management with Multiple Components**

**Description:**

Create a simple shopping cart with two components: Cart and Item.

**Requirements:**

* The Cart component should manage the state of the total items in the cart.
* The Item component should receive a name and price prop and have an "Add to Cart" button.
* When the button is clicked, the Cart component's state should be updated by adding 1 to the total number of items.
* The Cart component should display the total number of items and the total cost.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**6. Passing Props for Dynamic Content Rendering**

**Description:**

Create a UserList component that receives a list of user objects as a prop. Each user object should have properties: id, name, and email.

**Requirements:**

* The UserList component should map over the array of user objects and pass each user’s details to a User component.
* The User component should display the name and email of each user.
* Each user should have a unique key (use the id property).

A screenshot of a computer

AI-generated content may be incorrect.

**7. Conditional Rendering with Props**

**Description:**

Create a Greeting component that takes a name prop and conditionally displays one of two messages based on whether name is passed or not.

**Requirements:**

* If name is provided, display a greeting message like "Hello, [name]!".
* If name is not provided, display "Hello, Guest!".
* The parent component should pass the name prop to the Greeting component.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**8. Passing Functions as Props**

**Description:**

Create two components: Parent and Child. The Parent component should pass a function to the Child component as a prop.

**Requirements:**

* The Parent component should manage a message state (initial value "Hello").
* The Parent component should pass a function as a prop to the Child component. The function should update the message state.
* The Child component should call the function when a button is clicked, updating the message to "Goodbye".
* The Parent component should display the message.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

**9. Handling Form Input State**

**Description:**

Create a form component that allows the user to input their name and age.

**Requirements:**

* The form should have two input fields: one for the name and one for the age.
* Use the useState hook to manage the name and age input values.
* Display the name and age in a paragraph when the user submits the form.
* The form should display an error message if the user tries to submit the form with an empty name or age.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

**10. State Management and Prop Drilling**

**Description:**

Create three components: Grandparent, Parent, and Child. The Grandparent component should pass data to the Parent, and the Parent should pass that data down to the Child.

**Requirements:**

* The Grandparent component should manage a theme state (with values "light" or "dark").
* The Grandparent should pass the theme state down to the Parent component.
* The Parent component should pass the theme prop to the Child component.
* The Child component should display the theme (e.g., "The current theme is light").

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.