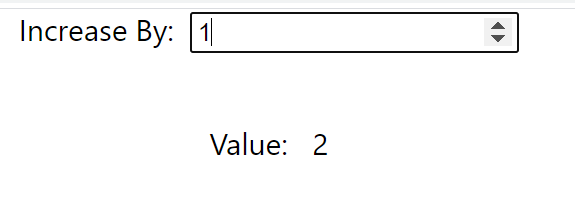
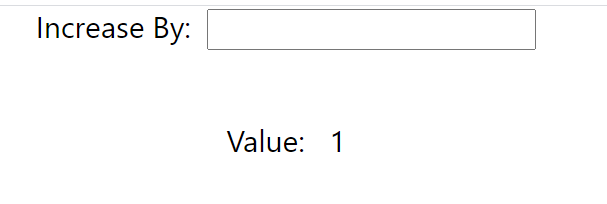
1. Create a web application using React JS

**Objective**: How to create React JS application using tool, understand folder structure, React components, props and state of component, class and functional component

**Capability Code**: WEBR0019 [Create first React app]



Description:

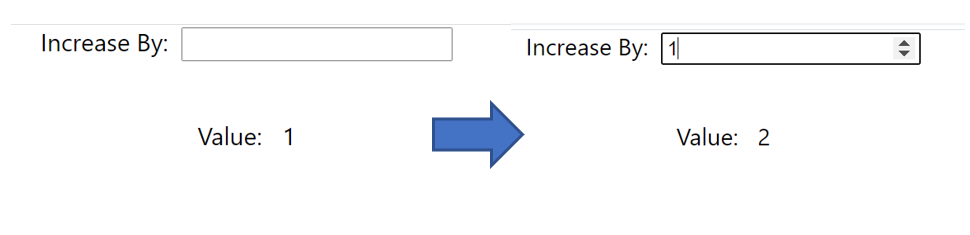
1. The application must use TypeScript as language
2. Convert the App component from functional to class component
3. Use ‘state’ property of the component to store a number type value through a key (property)
4. Provide a method in the App component class to increment the state key value. The method must accept a value by which the state key will be increased
5. Create a functional component which will be nested in App component design.

You have to pass the state key value and the function to increase the state key value to this component

1. Define a ‘type’ for the ‘props’ type of the functional component for property validation
2. The nested component in its design must have a text box element through which user will enter a numerical value
3. The nested component must pass the value entered into the text box to the parent component’s function (which increases the value stored in the state), immediately a numerical value is entered into the text box
4. The nested component should display the incremented value through a ‘p’ element in its design
5. Create a web application using React JS

**Objective**: How to create React JS application using tool, understand folder structure, React components, props and state of component, class and functional component

**Capability Code**: WEBR0019 [Create first React app]



Description:

Complete the same application once again. The difference is that the nested component must be a class component this time.

1. Create a React JS application without help of any CLI, manually from scratch to display the message ‘Welcome to React JS’ (use JS)