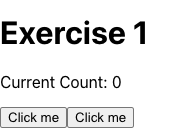
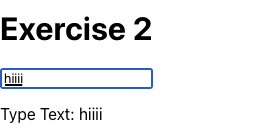
Ex



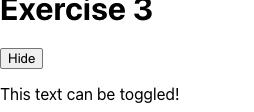
import *React*, {*useState*} from 'react';  
import {*Button*} from "react-bootstrap";  
  
const *Counter* = () => {  
 const [count , setCount] =*useState*(0);  
 return (  
 <div>  
 <h1>Exercise 1</h1>  
 <p>Current Count: {count}</p>  
 <Button onMouseUp={() => setCount(count + 1)}>  
 Click me  
 </Button>  
  
 <Button onMouseDown={() => {  
 if (count > 0) {  
 setCount(count - 1);  
 }  
 }}>  
 Click me  
 </Button>  
 </div>  
 );  
};  
  
export default *Counter*;

Ex2:



import *React*, {*useState*} from 'react';  
  
const *InputField* = () => {  
 const [inputValue , setInputValue] = *useState*('');  
  
 return (  
 <div>  
 <h1>Exercise 2</h1>  
 <input  
 type="text"  
 value={inputValue}  
 onChange={(e) => setInputValue(e.target.value)}  
 placeholder="Type something..."  
 />  
 <p>Type Text: {inputValue}</p>  
 </div>  
 );  
};  
  
export default *InputField*;

Ex3:

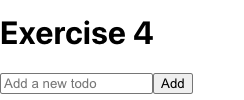




import *React*, {*useState*} from 'react';  
import {*Button*} from "react-bootstrap";  
  
const *ToggleVisibility* = () => {  
 const [isVisible , setIsVisible] = *useState*(false);  
  
 return (  
 <div>  
 <h1>Exercise 3</h1>  
 <Button onClick={() => setIsVisible(!isVisible)}>  
 {isVisible ? 'Hide' : 'Show'}  
 </Button>  
 {isVisible && <p>This text can be toggled!</p>}  
 </div>  
 );  
};  
  
export default *ToggleVisibility*;  
//Giải thích: Ban đầu isVisible là false, nên văn bản bị ẩn. Khi nhấn nút, trạng thái được đảo ngược, và văn bản sẽ hiển thị hoặc ẩn tương ứng .

Ex4:





import *React*, {*useState*} from 'react';  
import {*Button*} from "react-bootstrap";  
  
const *ToggleVisibility* = () => {  
 const [isVisible , setIsVisible] = *useState*(false);  
  
 return (  
 <div>  
 <h1>Exercise 3</h1>  
 <Button onClick={() => setIsVisible(!isVisible)}>  
 {isVisible ? 'Hide' : 'Show'}  
 </Button>  
 {isVisible && <p>This text can be toggled!</p>}  
 </div>  
 );  
};  
  
export default *ToggleVisibility*;  
//Giải thích: Ban đầu isVisible là false, nên văn bản bị ẩn. Khi nhấn nút, trạng thái được đảo ngược, và văn bản sẽ hiển thị hoặc ẩn tương ứng .

Ex5:

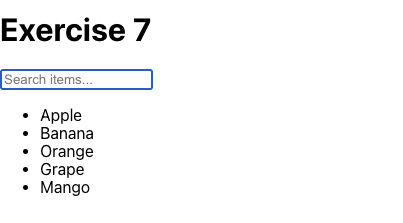


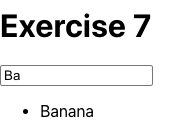
import *React*, { *useState* } from 'react';  
  
function *ColorSwitcher*() {  
 const [selectedColor, setSelectedColor] = *useState*('white');  
  
 return (  
 <div>  
 <h1>Exercise 6</h1>  
 <select  
 value={selectedColor}  
 onChange={(e) => setSelectedColor(e.target.value)}  
 >  
 <option value="red">Red</option>  
 <option value="blue">Blue</option>  
 <option value="green">Green</option>  
 <option value="yellow">Yellow</option>  
 </select>  
 <div  
 style={{  
 width: '100px',  
 height: '100px',  
 backgroundColor: selectedColor,  
 marginTop: '10px',  
 }}  
 ></div>  
 </div>  
 );  
}  
  
export default *ColorSwitcher*;

Ex6:

import *React*, { *useState* } from 'react';  
  
function *SearchFilter*() {  
 const [searchQuery, setSearchQuery] = *useState*('');  
 const items = ['Apple', 'Banana', 'Orange', 'Grape', 'Mango'];  
  
 const filteredItems = items.filter(item =>  
 item.toLowerCase().includes(searchQuery.toLowerCase())  
 );  
  
 return (  
 <div>  
 <h1>Exercise 7</h1>  
 <input  
 type="text"  
 value={searchQuery}  
 onChange={(e) => setSearchQuery(e.target.value)}  
 placeholder="Search items..."  
 />  
 <ul>  
 {filteredItems.map((item, index) => (  
 <li key={index}>{item}</li>  
 ))}  
 </ul>  
 </div>  
 );  
}  
  
export default *SearchFilter*;

Ex7:





import *React*, { *useState* } from 'react';  
  
function *DragDropList*() {  
 const [items, setItems] = *useState*(['Item 1', 'Item 2', 'Item 3', 'Item 4']);  
 const [draggingItem, setDraggingItem] = *useState*(null);  
  
 const handleDragStart = (e, index) => {  
 setDraggingItem(index);  
 e.dataTransfer.effectAllowed = 'move';  
 };  
  
 const handleDragOver = (e, index) => {  
 e.preventDefault();  
 e.dataTransfer.dropEffect = 'move';  
 };  
  
 const handleDrop = (e, index) => {  
 e.preventDefault();  
 if (draggingItem === null) return;  
 const copyListItems = [...items];  
 const dragItemContent = copyListItems[draggingItem];  
 copyListItems.splice(draggingItem, 1);  
 copyListItems.splice(index, 0, dragItemContent);  
 setDraggingItem(null);  
 setItems(copyListItems);  
 };  
  
 return (  
  
 <ul>  
 <h1>Exercise 8</h1>  
 {items.map((item, index) => (  
 <li  
 key={index}  
 draggable  
 onDragStart={(e) => handleDragStart(e, index)}  
 onDragOver={(e) => handleDragOver(e, index)}  
 onDrop={(e) => handleDrop(e, index)}  
 style={{padding: '10px', border: '1px solid #ccc', margin: '5px'}}  
 >  
 {item}  
 </li>  
 ))}  
 </ul>  
 );  
}  
  
export default *DragDropList*;