**Data Binding  
- State in Function Component  
- One Way Binding  
- Fetch data from API  
- Fakestore API**[**http://fakestoreapi.com/products**](http://fakestoreapi.com/products)**: returns all products**[**http://fakestoreapi.com/products/categories**](http://fakestoreapi.com/products/categories)**: returns all categories  
  
Ex:  
ShoppingComponent.js  
  
import { useState, useEffect } from "react";  
  
export default function ShoppingComponent()  
{  
    const [categories, setCategories] = useState([]);  
    const [products, setProducts] = useState([]);  
  
    function LoadCategories(){  
        fetch('**[**http://fakestoreapi.com/products/categories&#39**](http://fakestoreapi.com/products/categories&#39)**;)  
        .then(response=> response.json())  
        .then(data=> {  
            data.unshift('All');  
            setCategories(data);  
        })  
    }  
  
    function LoadProducts(){  
        fetch('**[**http://fakestoreapi.com/products&#39**](http://fakestoreapi.com/products&#39)**;)  
        .then(response=> response.json())  
        .then(data=>{  
            setProducts(data);  
        })  
    }  
  
    useEffect(()=>{  
        LoadCategories();  
        LoadProducts();  
    },[])  
  
  
    return(  
        <div className="container-fluid">  
          <header className="bg-danger text-white text-center p-2">  
             <h1> <span className="bi bi-cart"></span> Shopping Home</h1>  
          </header>  
          <section className="row">  
            <nav className="col-3">  
               <div>  
                <label>Select a Category</label>  
                <div>  
                    <select className="form-select">  
                        {  
                            categories.map(category=>  
                                <option key={category}>{category.toUpperCase()}</option>  
                                )  
                        }  
                    </select>  
                </div>  
               </div>  
            </nav>  
            <main className="col-9 d-flex flex-wrap overflow-auto" style={{height:'600px'}} >  
                {  
                    products.map(product=>  
                        <div key={product.id} className="card m-2 p-2 w-25">  
                            <img src={product.image} className="card-img-top" height="150" />  
                            <div className="card-header">  
                                <p>{product.title}</p>  
                            </div>  
                            <div className="card-body">  
                                <dl>  
                                    <dt>Price</dt>  
                                    <dd>{product.price}</dd>  
                                    <dt>Rating</dt>  
                                    <dd>  
                                        <span className="bi bi-star-fill text-success"></span>  
                                        {product.rating.rate} <span>[{product.rating.count}]</span>  
                                    </dd>  
                                </dl>  
                            </div>  
                        </div>  
                        )  
                }  
            </main>  
          </section>  
        </div>  
    )  
}  
  
Two Way Binding  
------------------------  
- React will not support two-way-binding.  
- You have to explicitly implement by using "EventBinding"  
- It is the process of identifying the changes in UI and updating the changes into data source.  
  
  
What is Event?  
- Event is a message sent by sender to its subscriber in order to notify the change.  
- Event follows a software design pattern called "Observer".  
- It is a communication pattern.  
- Event uses "Delegate" mechanism [ function pointer ].  
- JavaScript events are managed by "browser" window object.  
- JavaScript Browser Objects  
        - window  
        - location  
        - navigator  
        - history  
        - document  
- React can't directly use the browser events [Actual DOM]  
- React uses "SyntheticEvents" object for Virtual DOM.  
- Synthetic Events will map to browser events.  
- React Synthetic Events will support only one "event" argument.  
  [ 'this' is not allowed ]  
  
            event.EventProperties  
            event.target.objectProperty  
  
Syntax:  
            function Insert(e)  
            {  
                e.clientX, e.keyCode;  
                e.target.id, e.target.name, e.target.value;  
            }  
  
            <button  onClick={Insert}>  
  
- React uses all JavaScript Events  
  
1. Mouse Events  
        onMouseOver,  
        onMouseOut  
        onMouseMove  
        onMouseDown  
        onMouseUp  
2. Keyboard Events  
        onKeyUp      
        onKeyDown  
        onKeyPress  
3. Element State Events  
        onChange  
        onSelect  
        onFocus  
        onBlur  
        onChecked  
4. Button Events  
        onClick  
        onDblClick  
        onContextMenu  
5. Timer Events  
        setInterval()  
        clearInterval()  
        setTimeout()  
        clearTimeout()  
6. Clipboard Events  
        onCut()  
        onCopy()  
        onPaste()  
7. Touch Events  
        onTouchStart()  
        onTouchEnd()  
        onTouchMove()  
8. Form Events  
        onSubmit()  
        onReset()  
  
Note: If you are binding any value to HTML form element using "value" property, then it is configured as "read-only".  
        It allow read/write you have to bind "onChange" event for HTML form element.  
        If value binding is not defined, then you can configure without "onChange".  
  
Syntax:  
        const [ userName ] = useState('John');  
  
        <input type="text" value={userName}>  // invalid  
        <input type="text" value={userName} onChange="">  // valid  
  
Ex:  
EventBindingComponent.js  
  
import { useState, useEffect } from "react";  
  
export default function EventBindingComponent()  
{  
    const [userName, setUserName] = useState('John');  
  
    function handleUserName(e){  
        setUserName(e.target.value);  
    }  
  
    return(  
        <div className="container-fluid">  
            <dl>  
                <dt>User Name</dt>  
                <dd><input type="text" value={} onBlur={handleUserName}  /></dd>  
            </dl>  
            <h3>Hello ! {userName}</h3>  
        </div>  
    )  
}**