To do App documentation -

First create a React Application through the “create-react-app ‘appname’” command

Now start the code in “app.js” file

To show the data on button event –

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

list:[],

title:"Simple React TODO Application"

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

}

render() {

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

<form action="">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

</div>

);

}

}

export default App;

To set and push the data into parent state- TO PUSH MULTIPLE DATA INTO THE STATE

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application"

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

let store = {

txt\_item,

txt\_another\_item

}

let lists = this.state.lists;

lists.push(store);

// to send the data into the parent state

this.setState({

lists:lists

})

}

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

<form action="">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{JSON.stringify(lists)}

</section>

</div>

);

}

}

export default App;

TO PUSH THE SINGLE DATA INTO THE STATE-

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application"

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

// let store = {

// txt\_item,

// txt\_another\_item

// }

let lists = this.state.lists;

lists.push(txt\_item);

// to send the data into the parent state

this.setState({

lists:lists

})

}

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

<form action="">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{JSON.stringify(lists)}

</section>

</div>

);

}

}

export default App;

To Reset the form value using the concept of REF-

// to reset the value of form

this.refs.txt\_item.value = "";

this.refs.txt\_another\_item.value = "";

To Show the input data on the form (list)

The syntax is –

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application"

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

let store = {

txt\_item,

txt\_another\_item

}

let lists = this.state.lists;

// lists.push(txt\_item); SINGLE ITEM

lists.push(store);

// to send the data into the parent state

this.setState({

lists:lists

});

// to reset the value of form

this.refs.txt\_item.value = "";

this.refs.txt\_another\_item.value = "";

}

//end of additem event

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

{/\* to reset the form value define a ref properties \*/}

<form ref = "todoForm" name = "todoForm">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{/\* {JSON.stringify(lists)} \*/}

<ul>

{lists.map( (lists => <li key ={lists.txt\_item}>{lists.txt\_item} & {lists.txt\_another\_item} <button>Remove</button> </li>))}

</ul>

</section>

</div>

);

}

}

export default App;

to add the key value in map()

here we did not define the id of each item so we are going to ini one counter variable and pass it on the all the method.

The syatax is –

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application",

counter:0

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to initilize the counter variable to define the key attributes in list

let counter = this.state.counter;

counter++;

console.log(counter);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

let store = {

txt\_item,

txt\_another\_item,

counter

}

let lists = this.state.lists;

// lists.push(txt\_item); SINGLE ITEM

lists.push(store);

// to send the data into the parent state

this.setState({

lists:lists,

counter:counter

});

// to reset the value of form

this.refs.txt\_item.value = "";

this.refs.txt\_another\_item.value = "";

}

//end of additem event

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

{/\* to reset the form value define a ref properties \*/}

<form ref = "todoForm" name = "todoForm">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{/\* {JSON.stringify(lists)} \*/}

<ul>

{lists.map( (lists => <li key ={lists.counter}>{lists.txt\_item} & {lists.txt\_another\_item} <button>Remove</button> </li>))}

</ul>

</section>

</div>

);

}

}

export default App;

The better way to initialize the counter variable from the 0 index

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application",

counter:0

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to initilize the counter variable to define the key attributes in list

let counter = this.state.counter;

console.log(counter);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

let store = {

txt\_item,

txt\_another\_item,

counter

}

counter+=1;

let lists = this.state.lists;

// lists.push(txt\_item); SINGLE ITEM

lists.push(store);

// to send the data into the parent state

this.setState({

lists:lists,

counter:counter

});

// to reset the value of form

this.refs.txt\_item.value = "";

this.refs.txt\_another\_item.value = "";

}

//end of additem event

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

{/\* to reset the form value define a ref properties \*/}

<form ref = "todoForm" name = "todoForm">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{/\* {JSON.stringify(lists)} \*/}

<ul>

{lists.map( (lists => <li key ={lists.counter}>{lists.txt\_item} & {lists.txt\_another\_item} <button>Remove</button> </li>))}

</ul>

</section>

</div>

);

}

}

export default App;

Bind and Define the Remove button on each list item

The syntax is –

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application",

counter:0

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

// bind the remove btn

this.btn\_remove = this.btn\_remove.bind(this);

}

// define the remove function

btn\_remove(){

console.log("you have clicked the remove btn");

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to initilize the counter variable to define the key attributes in list

let counter = this.state.counter;

console.log(counter);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

let store = {

txt\_item,

txt\_another\_item,

counter

}

counter+=1;

let lists = this.state.lists;

// lists.push(txt\_item); SINGLE ITEM

lists.push(store);

// to send the data into the parent state

this.setState({

lists:lists,

counter:counter

});

// to reset the value of form

this.refs.txt\_item.value = "";

this.refs.txt\_another\_item.value = "";

}

//end of additem event

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

{/\* to reset the form value define a ref properties \*/}

<form ref = "todoForm" name = "todoForm">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{/\* {JSON.stringify(lists)} \*/}

<ul>

{lists.map( (lists => <li key ={lists.counter}>{lists.txt\_item} & {lists.txt\_another\_item} <button onClick={this.btn\_remove}>Remove</button> </li>))}

</ul>

</section>

</div>

);

}

}

export default App;

Define and check the remove button

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application",

counter:0

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

// bind the remove btn

this.btn\_remove = this.btn\_remove.bind(this);

}

// define the remove function

btn\_remove(index){

console.log(`You have clicked the ${index} button`);

// now pass the counter variable value in index on the button event

// use the concept of bind method

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to initilize the counter variable to define the key attributes in list

let counter = this.state.counter;

console.log(counter);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

let store = {

txt\_item,

txt\_another\_item,

counter

}

counter+=1;

let lists = this.state.lists;

// lists.push(txt\_item); SINGLE ITEM

lists.push(store);

// to send the data into the parent state

this.setState({

lists:lists,

counter:counter

});

// to reset the value of form

this.refs.txt\_item.value = "";

this.refs.txt\_another\_item.value = "";

}

//end of additem event

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

{/\* to reset the form value define a ref properties \*/}

<form ref = "todoForm" name = "todoForm">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{/\* {JSON.stringify(lists)} \*/}

<ul>

{lists.map( (lists => <li key ={lists.counter}>{lists.txt\_item} & {lists.txt\_another\_item} <button onClick={this.btn\_remove.bind(null,lists.counter)}>Remove</button> </li>))}

</ul>

</section>

</div>

);

}

}

export default App;

The complete syntax of the Remove Button-

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application",

counter:0

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

// bind the remove btn

this.btn\_remove = this.btn\_remove.bind(this);

}

// define the remove function

btn\_remove(index){

console.log(`You have clicked the ${index} button`);

// now pass the counter variable value in index on the button event

// use the concept of bind method

let lists = this.state.lists;

// find the exact item name which you want to remove

// let exact\_item = lists.find(function(exact\_item){

// return exact\_item.counter === index

// } )

// to remove the item use splice method

lists.splice(lists,1);

// set the state

this.setState({

lists:lists

});

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to initilize the counter variable to define the key attributes in list

let counter = this.state.counter;

console.log(counter);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

let store = {

txt\_item,

txt\_another\_item,

counter

}

counter+=1;

let lists = this.state.lists;

// lists.push(txt\_item); SINGLE ITEM

lists.push(store);

// to send the data into the parent state

this.setState({

lists:lists,

counter:counter

});

// to reset the value of form

this.refs.txt\_item.value = "";

this.refs.txt\_another\_item.value = "";

}

//end of additem event

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

{/\* to reset the form value define a ref properties \*/}

<form ref = "todoForm" name = "todoForm">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{/\* {JSON.stringify(lists)} \*/}

<ul>

{lists.map( (lists => <li key ={lists.counter}>{lists.txt\_item} & {lists.txt\_another\_item} <button onClick={this.btn\_remove.bind(null,lists.counter)}>Remove</button> </li>))}

</ul>

</section>

</div>

);

}

}

export default App;

Another Syntax –

import React, { Component } from 'react';

import './App.css';

class App extends Component {

//define the constructor

constructor(props){

super(props);

// define the state

this.state ={

lists:[],

title:"Simple React TODO Application",

counter:0

}

// bind the method or btn function

this.addItem = this.addItem.bind(this);

// bind the remove btn

this.btn\_remove = this.btn\_remove.bind(this);

}

// define the remove function

btn\_remove(index){

console.log(`You have clicked the ${index} button`);

// now pass the counter variable value in index on the button event

// use the concept of bind method

let lists = this.state.lists;

// find the exact item name which you want to remove

let exact\_item = lists.find(function(exact\_item){

return exact\_item.counter === index

} )

// to remove the item use splice method

lists.splice(exact\_item,1);

// set the state

this.setState({

lists:lists

});

}

// define the function

addItem(event){

// to prevent the auto refresh call the previous state

event.preventDefault();

// to check the refrence value of the input fields

// console.log(this.refs);

// to fetch the input txt\_item field value

let txt\_item = this.refs.txt\_item.value;

console.log(txt\_item);

// to initilize the counter variable to define the key attributes in list

let counter = this.state.counter;

console.log(counter);

// to fetch the input txt\_another\_item field value

let txt\_another\_item = this.refs.txt\_another\_item.value;

console.log(txt\_another\_item);

// create a variable to stor the data

let store = {

txt\_item,

txt\_another\_item,

counter

}

counter+=1;

let lists = this.state.lists;

// lists.push(txt\_item); SINGLE ITEM

lists.push(store);

// to send the data into the parent state

this.setState({

lists:lists,

counter:counter

});

// to reset the value of form

this.refs.txt\_item.value = "";

this.refs.txt\_another\_item.value = "";

}

//end of additem event

render() {

let lists = this.state.lists;

return (

<div className="App">

<h2>{this.state.title}</h2>

{/\* define the form to take the user input \*/}

{/\* to reset the form value define a ref properties \*/}

<form ref = "todoForm" name = "todoForm">

<input type="text" name="txt\_item" ref="txt\_item" placeholder="Enter Item:"/>

<input type="text" name="txt\_another\_item" ref="txt\_another\_item" placeholder="Enter another Item"/>

<button onClick = {this.addItem}>Add</button>

</form>

<section>

<h3>Enter data</h3>

{/\* {JSON.stringify(lists)} \*/}

<ul>

{lists.map( (lists => <li key ={lists.counter}>{lists.txt\_item} & {lists.txt\_another\_item} <button onClick={this.btn\_remove.bind(null,lists.counter)}>Remove</button> </li>))}

</ul>

</section>

</div>

);

}

}

export default App;

**Recape to set the value of state in any event-**

import React from 'react';

class App extends React.Component{

// define the btn add functionality

btn\_add(event){

// to avoid the default refresh of the browser

event.preventDefault();

console.log("YOU HAVE PRESSED THE ADD ITEM BUTTONS");

// first fetch the input items details in any variable

let text\_item\_name = this.refs.text\_item\_name.value;

console.log(text\_item\_name);

// to push the item in state

let list = this.state.list;

list.push(text\_item\_name);

// set the state now

this.setState({

list:list

});

// to clear the value of input text box

this.refs.text\_item\_name.value="";

}

// define the constructor

constructor(){

super();

this.state = {

list:[],

title:"Make Your Own Shoping list"

}

// bind the btn add function

this.btn\_add = this.btn\_add.bind(this);

}// end of constructor

render(){

// to receive the title from state in a variable

let title = this.state.title;

// to receive the list data from the state

var list = this.state.list;

return(

<div>

<h2>{title}</h2>

<h3>Shoping List</h3>

<form action="" name="shop\_Form" ref="shop\_Form">

<input type="text" name="text\_item\_name" ref="text\_item\_name" placeholder="Enter Items Name:"/>

<button onClick = {this.btn\_add}>Add Items</button>

</form>

<section>

{/\* to show the data of list \*/}

{JSON.stringify(list)}

</section>

</div>

);// end of return method

}//end of render

}// end of class

//export this component to use in anothe module

export default App;

To show the data on the form –

import React from 'react';

class App extends React.Component{

// define the btn add functionality

btn\_add(event){

// to avoid the default refresh of the browser

event.preventDefault();

console.log("YOU HAVE PRESSED THE ADD ITEM BUTTONS");

// first fetch the input items details in any variable

let text\_item\_name = this.refs.text\_item\_name.value;

console.log(text\_item\_name);

// to push the item in state

let list = this.state.list;

list.push(text\_item\_name);

// set the state now

this.setState({

list:list

});

// to clear the value of input text box

this.refs.text\_item\_name.value="";

}

// define the constructor

constructor(){

super();

this.state = {

list:[],

title:"Make Your Own Shoping list"

}

// bind the btn add function

this.btn\_add = this.btn\_add.bind(this);

}// end of constructor

render(){

// to receive the title from state in a variable

let title = this.state.title;

// to receive the list data from the state

let list = this.state.list;

return(

<div>

<h2>{title}</h2>

<h3>Shoping List</h3>

<form action="" name="shop\_Form" ref="shop\_Form">

<input type="text" name="text\_item\_name" ref="text\_item\_name" placeholder="Enter Items Name:"/>

<button onClick = {this.btn\_add}>Add Items</button>

</form>

<section>

{/\* to show the data of list in json formt \*/}

{/\* {JSON.stringify(list)} \*/}

{/\* to show the data in list for that you need to define the key attributes \*/}

<ul>

{list.map( (list => <li>{list} </li>))}

</ul>

</section>

</div>

);// end of return method

}//end of render

}// end of class

//export this component to use in anothe module

export default App;