File include :

App.js

Components

Header.js

Todoinput.js

Todoitem.js

“app.js”

// import React, { Component } from 'react';

import React from 'react';

import './App.css';

import Header from './components/header';

import TodoInput from './components/todoinput';

import TodoItems from './components/todoitems';

class App extends React.Component {

// declare a constructor

constructor(props){

super(props);

this.state = {

todos:[

{id:0, text:"wishvanath"},

{id:1, text:"Aakash"},

{id:2, text:"khagesh"},

{id:3, text:"sankar"}

],

nextId:4

}

// bind the function

this.addTodo = this.addTodo.bind(this);

this.removeTodo = this.removeTodo.bind(this);

}

// define the addTodo function

addTodo(todoText){

// check in console

//console.log("todo list is added:", todoText);

let todos = this.state.todos.slice();

todos.push({id: this.state.nextId, text: todoText});

this.setState({

todos: todos,

nextId: ++ this.state.nextId

});

}

// define the removeTodo function

removeTodo(id){

// check in console

//console.log("item is removing:",id);

// implement the logic to remove the items

this.setState({

todos: this.state.todos.filter((todo, index) => todo.id !== id)

});

}

render() {

return (

<div className="App">

<div className="todo-wrapper">

<Header/>

<TodoInput todoText = "" addTodo = {this.addTodo}/>

<ul>

{

this.state.todos.map((todo) => {

return<TodoItems todo={todo} key={todo.id} id={todo.id} removeTodo = {this.removeTodo} />

})

}

</ul>

</div>

</div>

);

}

}

export default App;

“header.js”

import React from 'react';

import '../App.css';

class Header extends React.Component{

render(){

return(

<div className ="todo-header">

<h2>React TODO Application</h2>

<hr className = "header\_line"/>

</div>

);

}

}

export default Header;

“todoinput.jsx”

import React from 'react';

//import the css file of todoinput components

import './todoinput.css';

class TodoInput extends React.Component {

constructor(props) {

super(props);

this.handleChange = this.handleChange.bind(this);

this.addTodo = this.addTodo.bind(this);

this.state = {

value: "",

};

}

// to handle the input change function

handleChange(e){

// to check in console

//console.log("input function change here");

this.setState({value: e.target.value});

}

// to handle the submit button function

addTodo(todo){

// check in console

// console.log("TODO:",todo);

// call the addTodo function from parent class

// ensure the todotext field is not empty

if(todo.length > 0){

this.props.addTodo(todo);

this.setState(

{value:""}

);

}

}

render() {

return (

<div>

<input className ="user\_input" type = "text" value={this.state.value} onChange = {this.handleChange} />

<button className = "btn btn-primary" onClick = {() => this.addTodo(this.state.value)}>Submit</button>

</div>

);

}

}

export default TodoInput;

“todoitem.jsx”

import React from 'react';

import './todoitems.css';

class TodoItems extends React.Component{

constructor(props){

super(props);

}

removeTodo(id){

this.props.removeTodo(id);

}

render(){

return(

<div className = "todoWrapper">

<button className = "remove\_btn" onClick = {(e) => this.removeTodo(this.props.id)}>Remove</button>{this.props.todo.text}

</div>

);

}

}

export default TodoItems;

stateful Example-(using props)

import React, { Component } from 'react';

class App extends Component {

// define the constructor for state ful example

constructor(){

super();

this.state= {

data:[

{id:0,name:"wishvanath",age:25},

{id:1, name:"Monika", age:23},

{id:2, name:"Rakesh", age:26}

]

}

}// end of consructor

render() {

return (

<div className="App">

{/\* exmaple of multiple components \*/}

<Header />

<table border="2px">

<tbody>

{this.state.data.map((data => <TableRow key ={data.id} data={data}/>))}

</tbody>

</table>

</div>

);

}

}

// define the tablerow componet

class TableRow extends Component{

render(){

return(

<tr>

<td>{this.props.data.id}</td>

<td>{this.props.data.name}</td>

<td>{this.props.data.age}</td>

</tr>

);

}

}

// this is my header component

class Header extends Component{

render(){

return(

<div>

<h1>This is my Header</h1>

</div>

);// end of returnn

}// end of render

}// end of header components

export default App;

More Concept on State and props:

// example of default props

import React, { Component } from 'react';

class App extends Component {

constructor(){

super();

this.state={

data:[

{name:"wishvanath",title:"student"},

{name:"wishv",title:"softwareDev"},

],

myname:"wishvanath",

some\_text:"nothing want to say"

}

}

render() {

let data = this.state.data;

return (

<div>

{data.map(( data=> <ListFormat key={data.name} data={data} some\_text={this.state.some\_text} />))}

</div>

);

}

}

class ListFormat extends Component {

render() {

return (

<div>

<ul>

<li>{this.props.data.name}</li>

<li>{this.props.data.title}</li>

</ul>

<h1>{this.props.some\_text}</h1>

</div>

);

}

}

// define the header componet

// class Header extends Component{

// render(){

// return(

// <div>

// <h1>{this.props.data}</h1>

// </div>

// );

// }

// }

export default App;

Set State Example –

// example of set state in form

import React from 'react';

class App extends React.Component {

constructor(props) {

super(props);

this.state = {

data:[]

};

// bind the click me function

this.clickMe = this.clickMe.bind(this);

}

// define the function

clickMe(event){

let item ="apple"

event.preventDefault();

console.log("you have clicked me");

let data = this.state.data;

data.push(item);

this.setState({

data:data

});

console.log(data);

}

render() {

let data = this.state.data;

return (

<div>

<form>

<button type="submit" name="clickMe" onClick={this.clickMe}>Click me</button>

</form>

<h2>Your data is :</h2>

<p>

{JSON.stringify(data)}

</p>

<h3>{data}</h3>

</div>

);

}

}

export default App;

One way data binding-

// bind a input box and show the input data on the form

// one way data binding

import React, { Component } from 'react';

class App extends Component {

// define a constructor

constructor(){

super();

this.state={

value:"wishvanath"

};

// bind the function

this.updateText=this.updateText.bind(this);

}

// define the function

updateText(e){

var input\_val = e.target.value;

this.setState({

value:input\_val

})

}

render() {

return (

<div>

<input type="text" name="txt\_name" ref="txt\_name" value={this.state.value} onChange={this.updateText}/>

<h3>{this.state.value}</h3>

</div>

);

}

}

export default App;

One way databinding using the concept of parent components

// another example of one way databinding using the concept of nexted component

import React from 'react';

class ComponentName extends React.Component {

constructor(props) {

super(props);

this.state = {

data:"somthing..."

};

// bind the update function

this.updateState = this.updateState.bind(this);

}

// define the function

updateState(e){

// set the state value of the parent components

this.setState({

data:e.target.value

})

}

render() {

return (

<div>

<ChildCompo showdata={this.state.data} updateFunction = {this.updateState}/>

</div>

);

}

}

// create the childCompo

class ChildCompo extends React.Component{

render(){

return(

<div>

<input type="text" value={this.props.showdata} onChange={this.props.updateFunction}/>

<h3>{this.props.showdata}</h3>

</div>

);

}

}

export default ComponentName;

To update a single item

// event example

import React from 'react';

class App extends React.Component {

constructor(props) {

super(props);

this.state = {

data:"wishvanath"

};

// bind the function

this.updateState = this.updateState.bind(this);

}

// define the function

updateState(){

this.setState({

data:'anand'

})

}

render() {

return (

<div>

<button onClick={this.updateState}>Click me</button>

<h1>{this.state.data}</h1>

</div>

);

}

}

export default App;

Update data from child component-

// event example

import React from 'react';

class App extends React.Component {

constructor(props) {

super(props);

this.state = {

data:"wishvanath"

};

// bind the function

this.updateState = this.updateState.bind(this);

}

// define the function

updateState(){

this.setState({

data:'anand'

})

}

render() {

return (

<div>

<ChildComp showdata={this.state.data} showFunction = {this.updateState}/>

</div>

);

}

}

// create a child component

class ChildComp extends React.Component{

render(){

return(

<div>

<button onClick={this.props.showFunction}>Click me</button>

<h2>{this.props.showdata}</h2>

</div>

);

}

}

export default App;