**Using useState:**

**index.js**

**import React, { useState, useReducer } from "react";**

**import Modal from "./Modal";**

**import { reducer } from "../setup/reducer";**

**import { data } from "../../../data";**

**import { act } from "react-dom/test-utils";**

**const Index = () => {**

**const [name, setName] = useState("");**

**const [people, setPeople] = useState(data);**

**const [showModal, setShowModal] = useState(false);**

**const handleSubmit = (e) => {**

**e.preventDefault();**

**if (name) {**

**setPeople([**

**...people, \*\*\*\*\* *age j data gulo ache shegulo nite hobe age***

**{ id: new Date().getTime().toString(), name: name }, \*\*\* *array er vitor r ekti element add hoye gelo***

**]);**

**setShowModal(true);**

**setName(""); \*\*\*\*\* *input field empty kore dite hobe***

**} else {**

**setShowModal(true);**

**}**

**};**

**return (**

**<>**

**{showModal && <Modal />}**

**<form className="form" onSubmit={handleSubmit}>**

**<div>**

**<input**

**type="text"**

**value={name}**

**onChange={(e) => setName(e.target.value)}**

**/>**

**</div>**

**<button type="btn" className="btn">**

**Submit**

**</button>**

**</form>**

**{people.map((person) => { \*\*\*\**submit korar por shob gulo data dekhabe one by one***

**const { id, name } = person;**

**return (**

**<div key={id}> \*\*\*\**id use korte hobe to prevent warning***

**<h4>{name}</h4>**

**</div>**

**);**

**})}**

**</>**

**);**

**};**

**export default Index;**

**Modals.js**

**import React, { useEffect } from "react";**

**const Modal = () => {**

**return <div>I am a Modal</div>;**

**};**

**export default Modal;**

**data.js**

**export const data = [**

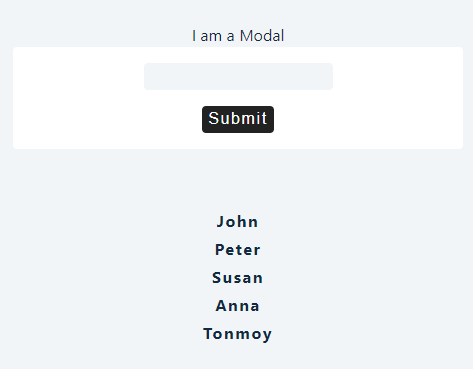
**{ id: 1, name: 'john' },**

**{ id: 2, name: 'peter' },**

**{ id: 3, name: 'susan' },**

**{ id: 4, name: 'anna' },**

**];**

****

**useReducer Refactor:**

**const reducer = (state, action) => {}; \*\*\*\**reducer function. Action means what we want to do***

**const defaultState = { \*\*\*\**default states will be written here***

**people: data,**

**isModalOpen: true,**

**modalContent: "Hello World",**

**};**

**const Index = () => {**

**const [name, setName] = useState("");**

**const [state, dispatch] = useReducer(reducer, defaultState); \*\*\*\* *state er moddhe reducer and defaultState ache***

**const handleSubmit = (e) => {**

**e.preventDefault();**

**if (name) {**

**} else {**

**}**

**};**

**return (**

**<>**

**{state.isModalOpen && <Modal modalContent={state.modalContent} />}**

**<form className="form" onSubmit={handleSubmit}>**

**<div>**

**<input**

**type="text"**

**value={name}**

**onChange={(e) => setName(e.target.value)}**

**/>**

**</div>**

**<button type="btn" className="btn">**

**Submit**

**</button>**

**</form>**

**{state.people.map((person) => {**

**const { id, name } = person;**

**return (**

**<div key={id}>**

**<h4>{name}</h4>**

**</div>**

**);**

**})}**

**</>**

**);**

**};**

**export default Index;**

**useReducer Add Item:**

**const reducer = (state, action) => { \*\*\*\* *reducer theke must state return korte hobei***

**if (action.type === "ADD\_ITEM") { \*\*\* *dispatch pass koreche type ta***

**const newPeople = [...state.people, action.payload]; \*\*\* *previous people nilam from state.people***

**return { \*\*\*\* *r payload er newpeople ta nilam***

***// ...state,***

**people: newPeople, \*\*\*\* *ekhane state update kore dilam***

**isModalOpen: true,**

**modalContent: "Item Added",**

**};**

**}**

**if (action.type === "NO\_VALUE") {**

**return {**

**...state,**

**isModalOpen: true,**

**modalContent: "Please,Enter a value",**

**};**

**}**

**throw new Error("No Matching action type");**

**};**

**const defaultState = {**

**people: data,**

**isModalOpen: true,**

**modalContent: "Hello World",**

**};**

**const Index = () => {**

**const [name, setName] = useState("");**

**const [state, dispatch] = useReducer(reducer, defaultState);**

**const handleSubmit = (e) => {**

**e.preventDefault();**

**if (name) {**

**const newPerson = { id: new Date().getTime().toString(), name: name };**

**dispatch({ type: "ADD\_ITEM", payload: newPerson }); \*\*\**dispatch action type hishebe pass korbe to reducer***

**setName(""); \*\*\*\* *payLoad use kora hoi to pass data to reducer-> action***

**} else {**

**dispatch({ type: "NO\_VALUE" });**

**}**

**};**

**return (**

**<>**

**{state.isModalOpen && <Modal modalContent={state.modalContent} />}**

**<form className="form" onSubmit={handleSubmit}>**

**<div>**

**<input**

**type="text"**

**value={name}**

**onChange={(e) => setName(e.target.value)}**

**/>**

**</div>**

**<button type="btn" className="btn">**

**Submit**

**</button>**

**</form>**

**{state.people.map((person) => {**

**const { id, name } = person;**

**return (**

**<div key={id}>**

**<h4>{name}</h4>**

**</div>**

**);**

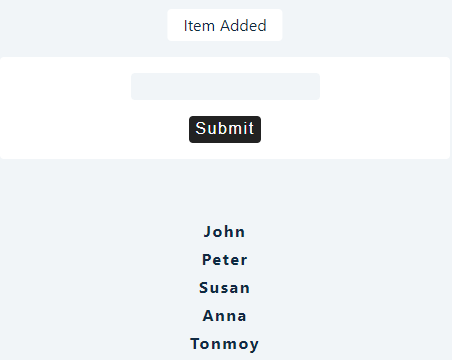
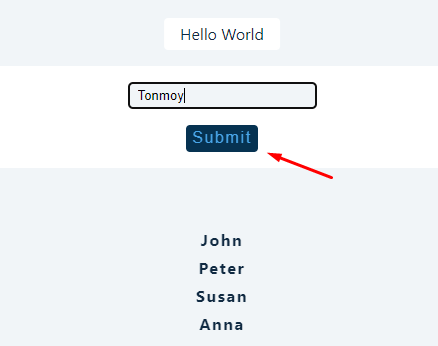
**})}**

**</>**

**);**

**};**

**export default Index;**

****

**Remove Item:**

**index.js**

**const reducer = (state, action) => {**

**if (action.type === "ADD\_ITEM") {**

**const newPeople = [...state.people, action.payload];**

**return {**

***// ...state,***

**people: newPeople,**

**isModalOpen: true,**

**modalContent: "Item Added",**

**};**

**}**

**if (action.type === "NO\_VALUE") {**

**return {**

**...state,**

**isModalOpen: true,**

**modalContent: "Please,Enter a value",**

**};**

**}**

**if (action.type === "CLOSE\_MODAL") {**

**return {**

**...state,**

**isModalOpen: false,**

**};**

**}**

**if (action.type === "REMOVE\_ITEM") {**

**const newPeople = state.people.filter((person) => {**

**return person.id !== action.payload;**

**});**

**return {**

**...state,**

**people: newPeople,**

**isModalOpen: true,**

**modalContent: "Item Removed",**

**};**

**}**

**throw new Error("No Matching action type");**

**};**

**const defaultState = {**

**people: data,**

**isModalOpen: true,**

**modalContent: "Hello World",**

**};**

**const Index = () => {**

**const [name, setName] = useState("");**

**const [state, dispatch] = useReducer(reducer, defaultState);**

**const handleSubmit = (e) => {**

**e.preventDefault();**

**if (name) {**

**const newPerson = { id: new Date().getTime().toString(), name: name };**

**dispatch({ type: "ADD\_ITEM", payload: newPerson });**

**setName("");**

**} else {**

**dispatch({ type: "NO\_VALUE" });**

**}**

**};**

**const closeModal = () => {**

**dispatch({ type: "CLOSE\_MODAL" });**

**};**

**return (**

**<>**

**{state.isModalOpen && (**

**<Modal modalContent={state.modalContent} closeModal={closeModal} />**

**)}**

**<form className="form" onSubmit={handleSubmit}>**

**<div>**

**<input**

**type="text"**

**value={name}**

**onChange={(e) => setName(e.target.value)}**

**/>**

**</div>**

**<button type="btn" className="btn">**

**Submit**

**</button>**

**</form>**

**{state.people.map((person) => {**

**const { id, name } = person;**

**return (**

**<div key={id} className="item">**

**<h4>{name}</h4>**

**<button**

**type="btn"**

**className="btn"**

**onClick={() => {**

**dispatch({ type: "REMOVE\_ITEM", payload: id });**

**}}**

**>**

**Remove**

**</button>**

**</div>**

**);**

**})}**

**</>**

**);**

**};**

**export default Index;**

**Modal.js**

**import React, { useEffect } from "react";**

**const Modal = ({ modalContent, closeModal }) => {**

**useEffect(() => {**

**setTimeout(() => {**

**closeModal();**

**}, 3000);**

**});**

**return <div className="modal">{modalContent}</div>;**

**};**

**export default Modal;**