1] index.html

<div id="app"></div>

2] Css file

html, body, input, button {

font-family: Sawasdee;

font-size: 20px;

}

.todo {

}

.todo\_\_list {

margin: 0;

padding: 0;

list-style-type: none;

}

.todo\_\_item {

padding: .5em .25em;

border-bottom: solid 1px #eee;

}

.todo\_\_item:hover {

background: #f7f7f7;

cursor: pointer;

}

.todo\_\_entry {

border: solid 1px #ccc;

padding: .25em .5em;

border-radius: .2em;

background: #f3f3f3;

width: 100%;

box-sizing: border-box;

}

.todo\_\_button {

border: 0;

border-radius: .2em;

background: #71B7FF;

color: #fff;

padding: .25em .5em;

margin: .5em 0;

margin-right: .25em;

cursor: pointer;

}

.todo\_\_button:hover {

background: #B2D8FF;

}

3] App.jsx

const { Map, List } = Immutable;

const { createStore } = Redux;

const { Provider, connect } = reactRedux;

const components = {

Todo({ todo }) {

if(todo.isDone) {

return <strike>{todo.text}</strike>;

} else {

return <span>{todo.text}</span>;

}

},

TodoList({ todos, toggleTodo, addTodo }) {

const onSubmit = (e) => {

const text = e.target.value;

if(e.which === 13 && text.length > 0) {

addTodo(text);

e.target.value = '';

}

};

const toggleClick = (id) => () => toggleTodo(id);

const { Todo } = components;

return (

<div className='todo'>

<input type='text'

className='todo\_\_entry'

placeholder='Add todo'

onKeyDown={onSubmit} />

<ul className='todo\_\_list'>

{todos.map(t => (

<li

key={t.get('id')}

className='todo\_\_item'

onClick={toggleClick(t.get('id'))}>

<Todo todo={t.toJS()} />

</li>

))}

</ul>

</div>

);

}

};

const actions = {

addTodo(text) {

return {

type: 'ADD\_TODO',

payload: {

id: Math.random().toString(34).slice(2),

isDone: false,

text

}

};

},

toggleTodo(id) {

return {

type: 'TOGGLE\_TODO',

payload: id

}

}

};

const init = List();

const reducer = function(state=init, action) {

switch(action.type) {

case 'ADD\_TODO':

return state.push(

Map(action.payload)

);

case 'TOGGLE\_TODO':

return state.map(t => {

if(t.get('id') == action.payload) {

return t.update('isDone', isDone => !isDone);

} else {

return t;

}

});

default:

return state;

}

};

const containers = {

TodoList: connect(

function mapStateToProps(state) {

return {

todos: state

};

},

function mapDispatchToProps(dispatch) {

return {

toggleTodo: (id) => dispatch(actions.toggleTodo(id)),

addTodo: (text) => dispatch(actions.addTodo(text))

};

}

)(components.TodoList)

};

const { TodoList } = containers;

const store = createStore(reducer);

ReactDOM.render(

<Provider store={store}>

<TodoList />

</Provider>,

document.getElementById('app')

);