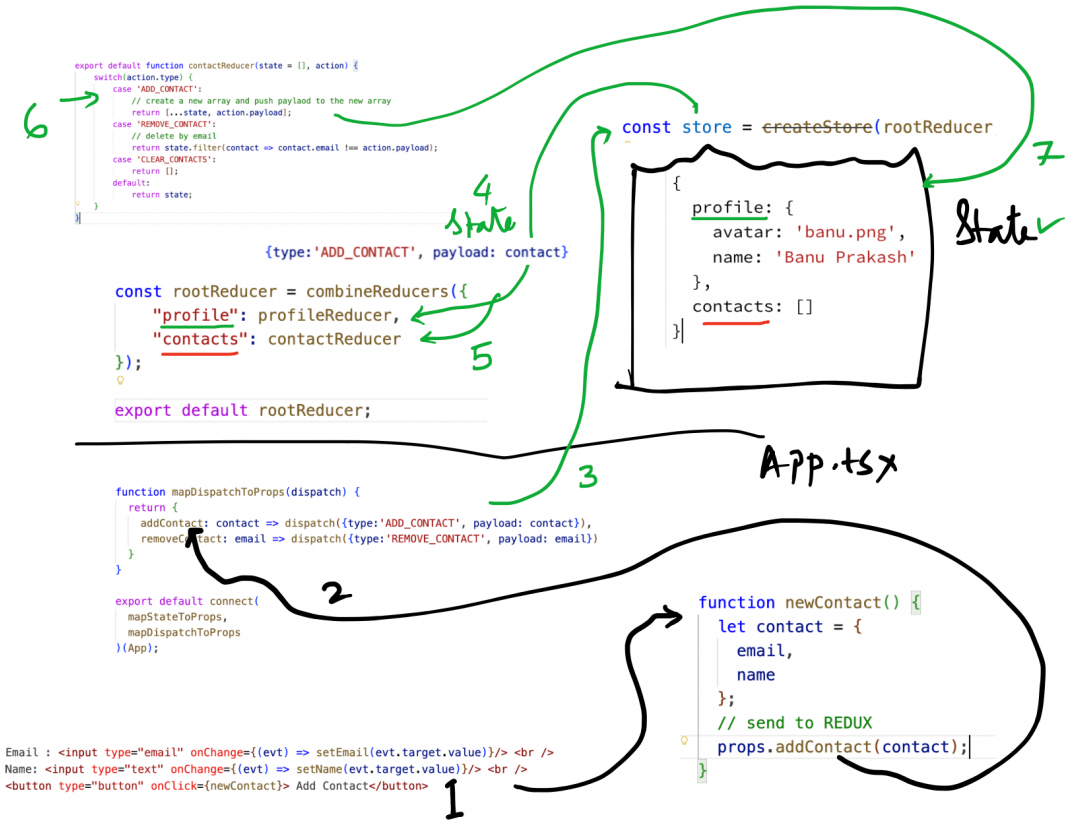
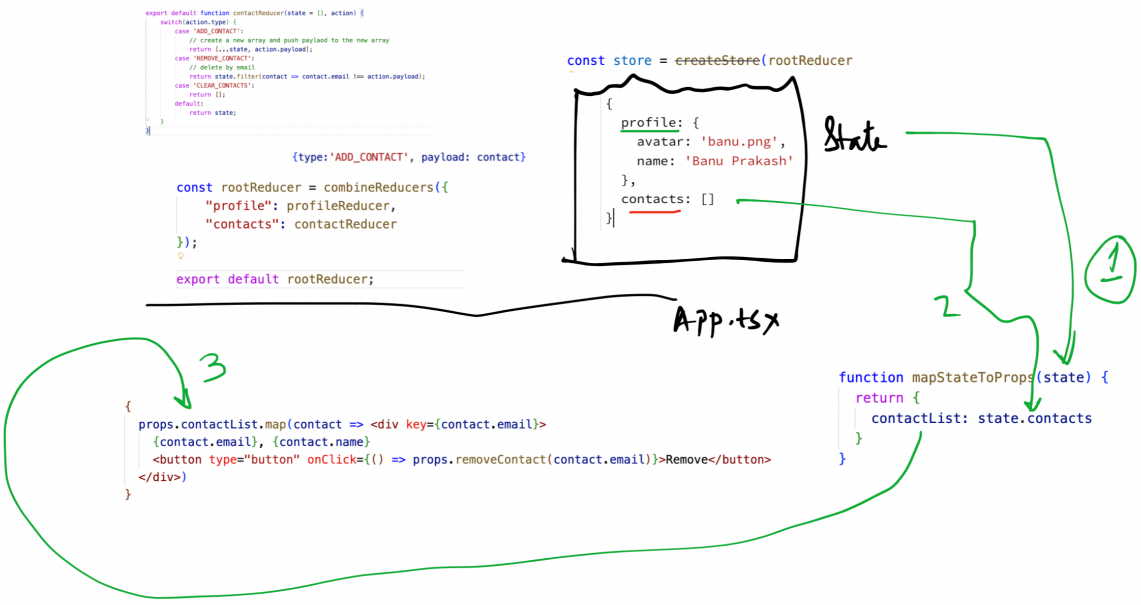
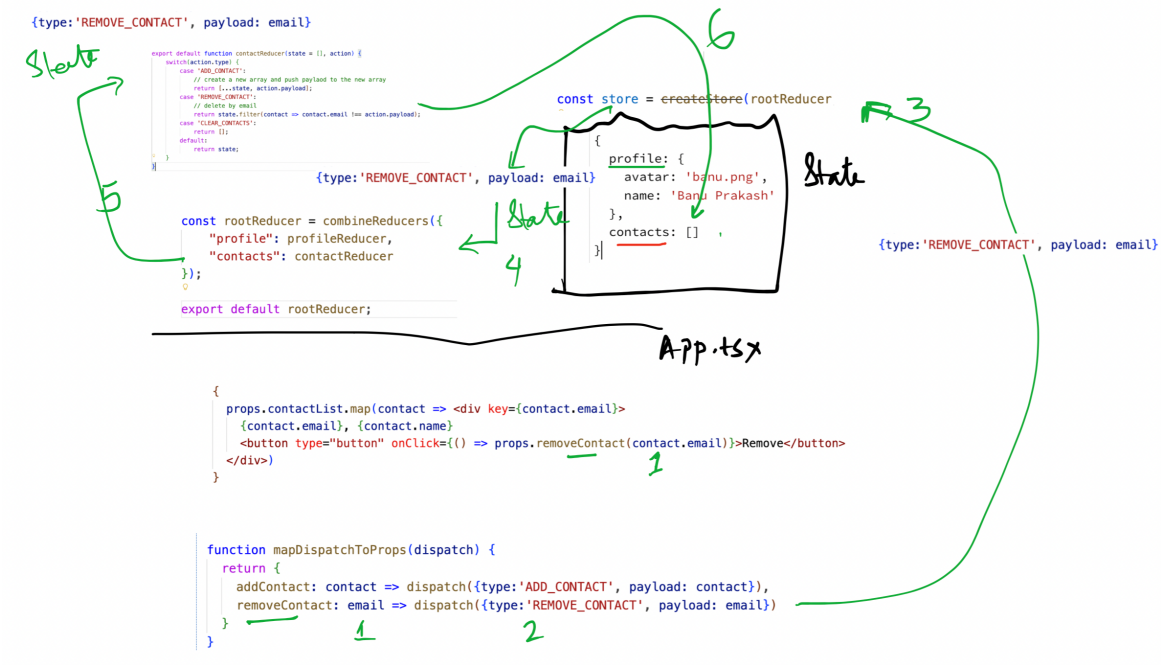

export default function contactReducer(state = [], action) {
switch(action.type) {
case 'ADD_CONTACT':
// create a new array and push paylaod to the new array
return [...state, action.payload];
case 'REMOVE_CONTACT':
// delete by email
return state.filter(contact => contact.email !== action.payload);
case 'CLEAR_CONTACTS':
return [];
default:
return state;
}
}
6
4
State
{type:'ADD_CONTACT', payload: contact}
const rootReducer = combineReducers({
"profile": profileReducer,
"contacts": contactReducer
});
export default rootReducer;
5
const store = createStore(rootReducer
{
profile: {
avatar: 'banu.png',
name: 'Banu Prakash'
},
contacts: []
}
7
State
App.tsx
3
function mapDispatchToProps(dispatch) {
return {
addContact: contact => dispatch({type:'ADD_CONTACT', payload: contact}),
removeContact: email => dispatch({type:'REMOVE_CONTACT', payload: email})
}
}
export default connect(
mapStateToProps,
mapDispatchToProps
)(App);
2
function newContact() {
let contact = {
email,
name
};
// send to REDUX
props.addContact(contact);
}
Email : <input type="email" onChange={(evt) => setEmail(evt.target.value)}/>
Name: <input type="text" onChange={(evt) => setName(evt.target.value)}/>
<button type="button" onClick={newContact}> Add Contact</button>
1


export default function contactReducer(state = [], action) {
switch(action.type) {
case 'ADD_CONTACT':
// create a new array and push paylaod to the new array
return [...state, action.payload];
case 'REMOVE_CONTACT':
// delete by email
return state.filter(contact => contact.email !== action.payload);
case 'CLEAR_CONTACTS':
return [];
default:
return state;
}
}
{type:'ADD_CONTACT', payload: contact}
const rootReducer = combineReducers({
"profile": profileReducer,
"contacts": contactReducer
});
export default rootReducer;
App.tsx
const store = createStore(rootReducer
{
profile: {
avatar: 'banu.png',
name: 'Banu Prakash'
},
contacts: []
}
State
1
2
3
{
props.contactList.map(contact =>
{contact.email}, {contact.name}
<button type="button" onClick={() => props.removeContact(contact.email)}>Remove</button>
)
}
function mapStateToProps(state) {
return {
contactList: state.contacts
}
}




```
{
  props.contactList.map(contact => <div key={contact.email}>
    {contact.email}, {contact.name}
    <button type="button" onClick={() => props.removeContact(contact.email)}>Remove</button>
  </div>)
}
```

1

```
function mapDispatchToProps(dispatch) {
  return {
    addContact: contact => dispatch({type:'ADD_CONTACT', payload: contact}),
    removeContact: email => dispatch({type:'REMOVE_CONTACT', payload: email})
  }
}
```

1 2