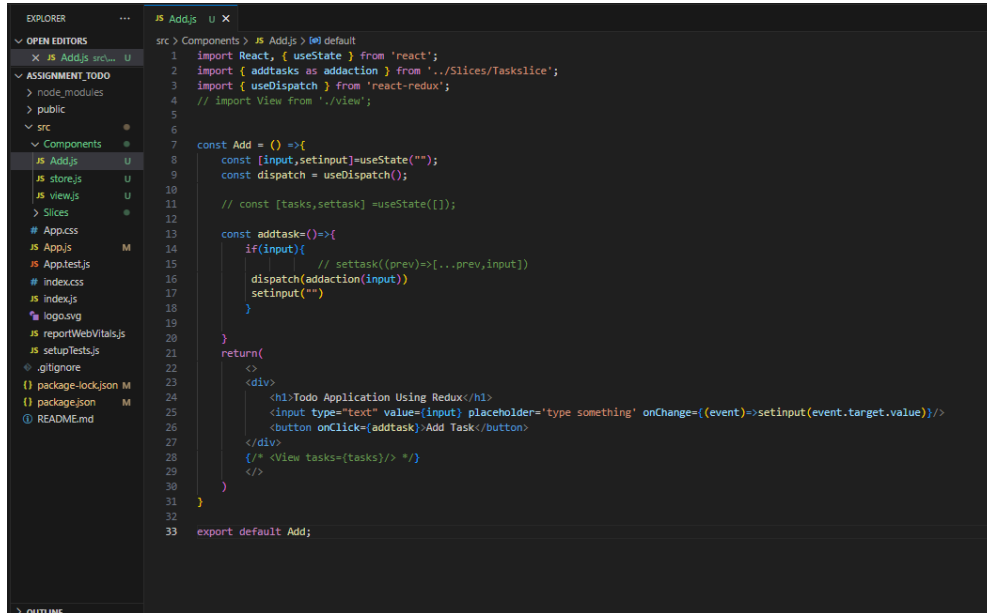


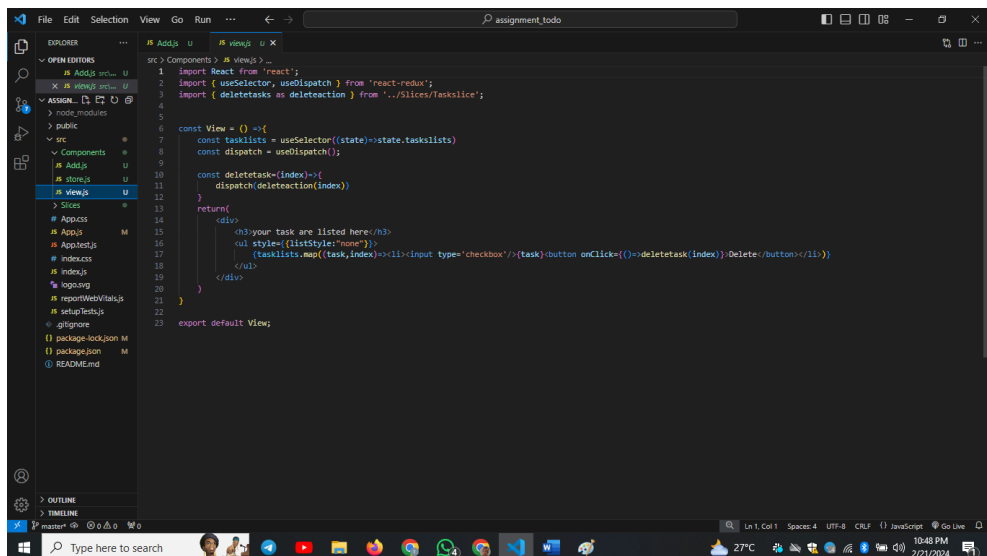
REACT MODULE – 2

Step-1 (Add.js)



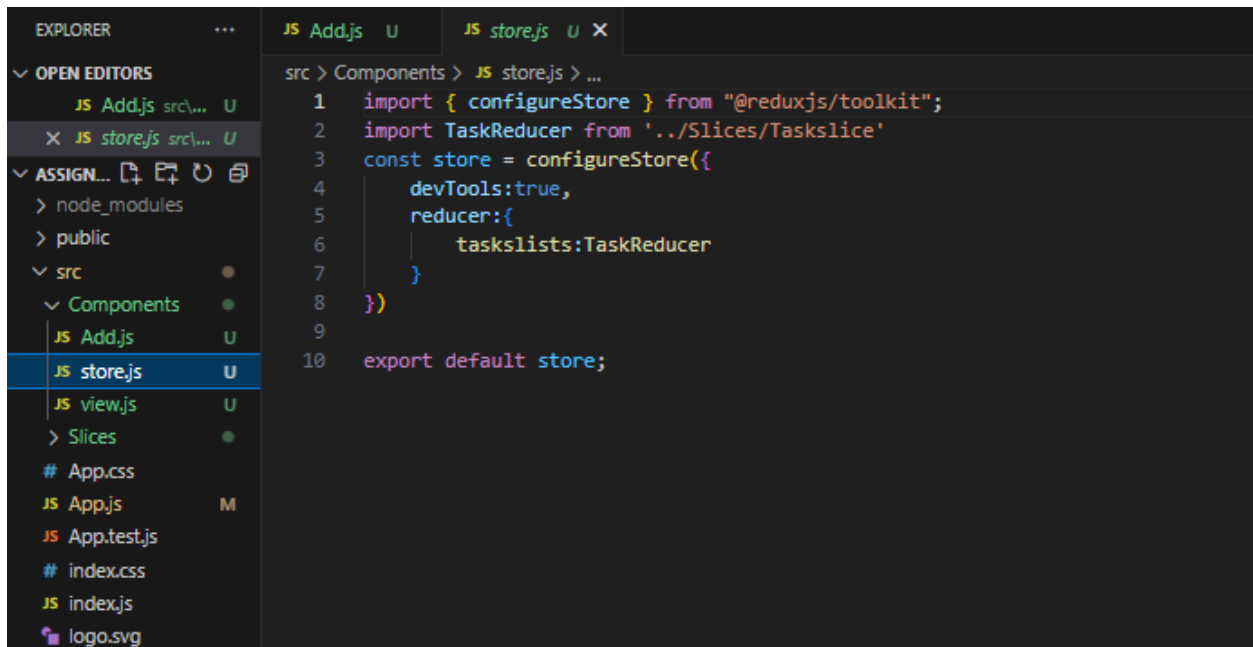
```
1 import React, { useState } from 'react';
2 import { addtasks as addaction } from '../Slices/Taskslice';
3 import { useDispatch } from 'react-redux';
4 // import View from './View';
5
6
7 const Add = () =>{
8   const [input,setinput]=useState("");
9   const dispatch = useDispatch();
10
11   // const [tasks,settask] =useState([]);
12
13   const addtask=()=>{
14     if(input){
15       // settask((prev)=>[...prev,input])
16       dispatch(addaction(input))
17       setinput("")
18     }
19   }
20
21   return(
22     <>
23     <div>
24       <h1>Todo Application Using Redux</h1>
25       <input type="text" value={input} placeholder="type something" onChange={(event)=>setinput(event.target.value)}/>
26       <button onClick={addtask}>Add Task</button>
27     </div>
28     < /* <View tasks={tasks}/> */
29     </>
30   )
31 }
32
33 export default Add;
```

Step-2 (view.js)



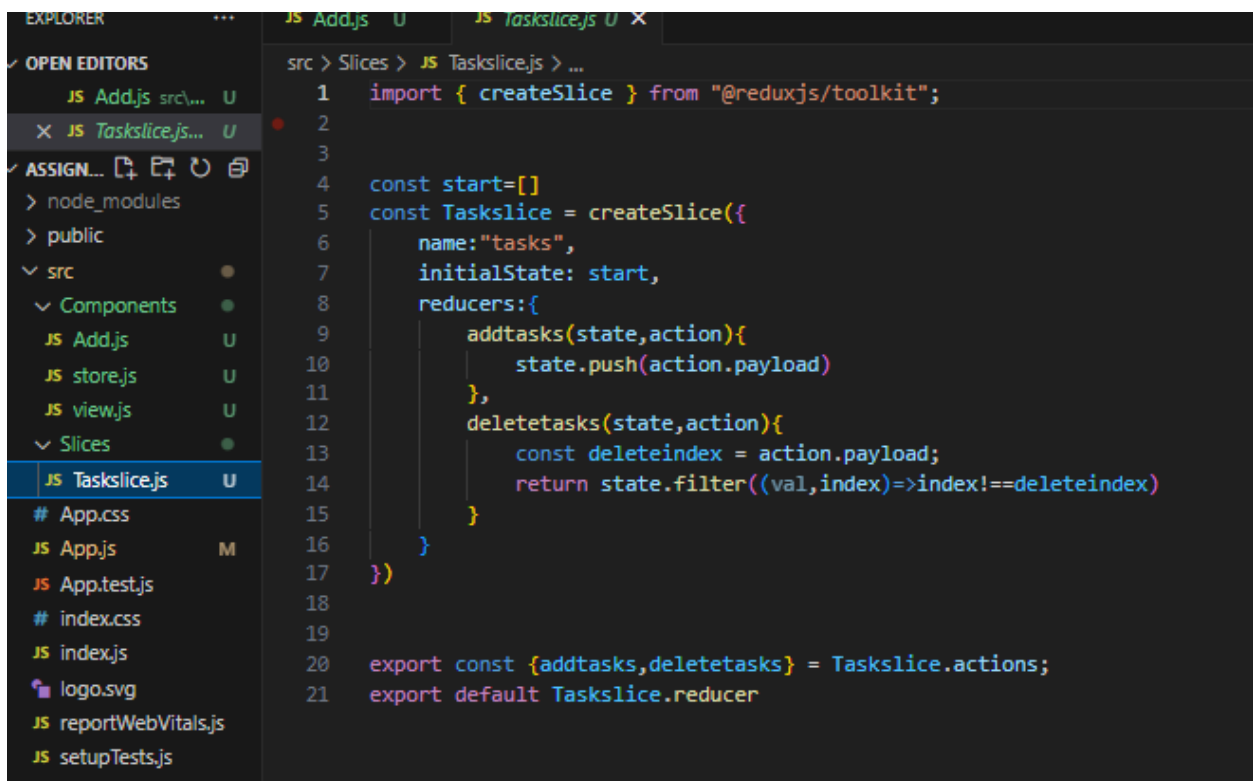
```
1 import React from 'react';
2 import { useSelector, useDispatch } from 'react-redux';
3 import { deletetasks as deletion } from '../Slices/Taskslice';
4
5
6 const View = () =>{
7   const tasklists = useSelector((state)=>state.tasklists)
8   const dispatch = useDispatch();
9
10   const deletetask=(index)=>{
11     dispatch(deletion(index))
12   }
13
14   return(
15     <div>
16       <p>your task are listed here:</p>
17       <ul style={listStyle:"none"}>
18         {tasklists.map((task,index)=><li><input type="checkbox"/>{task}<button onClick={()=>deletetask(index)}>Delete</button></li>)}
19       </ul>
20     </div>
21   )
22 }
23 export default View;
```

Step-3 (store.js)



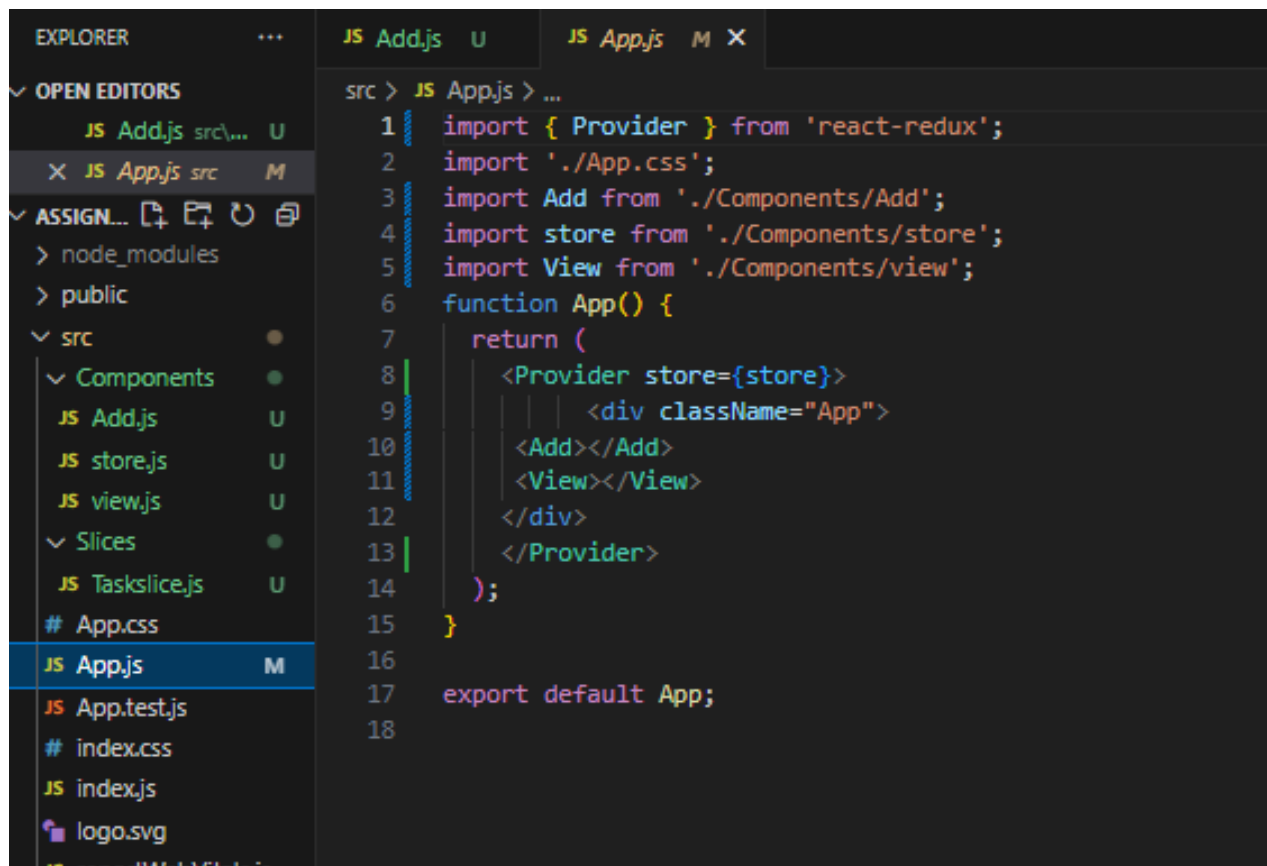
```
1 import { configureStore } from "@reduxjs/toolkit";
2 import TaskReducer from '../Slices/Taskslice'
3 const store = configureStore({
4   devTools:true,
5   reducer:{
6     taskslists:TaskReducer
7   }
8 })
9
10 export default store;
```

Step-4 (slice.js)



```
1 import { createSlice } from "@reduxjs/toolkit";
2
3
4 const start=[]
5 const Taskslice = createSlice({
6   name:"tasks",
7   initialState: start,
8   reducers:{
9     addtasks(state,action){
10       state.push(action.payload)
11     },
12     deletetasks(state,action){
13       const deleteindex = action.payload;
14       return state.filter((val,index)=>index!==deleteindex)
15     }
16   }
17 })
18
19
20 export const {addtasks,deletetasks} = Taskslice.actions;
21 export default Taskslice.reducer
```

Step-5 (app.js)



The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the project structure: 'src' contains 'Components' (with 'Add.js', 'store.js', and 'view.js') and 'Slices' (with 'Taskslice.js'). The 'App.js' file is selected and highlighted in blue. The main editor area shows the code for 'App.js' with line numbers 1 through 18. The code imports 'Provider' from 'react-redux', 'App.css', 'Add' from './Components/Add', 'store' from './Components/store', and 'View' from './Components/view'. It defines a function 'App()' that returns a JSX element wrapped in a 'Provider' with the 'store' prop. The JSX element has a 'div' with 'className="App"' containing 'Add' and 'View' components. Finally, 'App' is exported as the default export.

```
1 import { Provider } from 'react-redux';
2 import './App.css';
3 import Add from './Components/Add';
4 import store from './Components/store';
5 import View from './Components/view';
6 function App() {
7   return (
8     <Provider store={store}>
9       <div className="App">
10        <Add></Add>
11        <View></View>
12      </div>
13    </Provider>
14  );
15 }
16
17 export default App;
18
```

Output:

Todo Application Using Redux

your task are listed here

Todo Application Using Redux

your task are listed here

☐ task-1