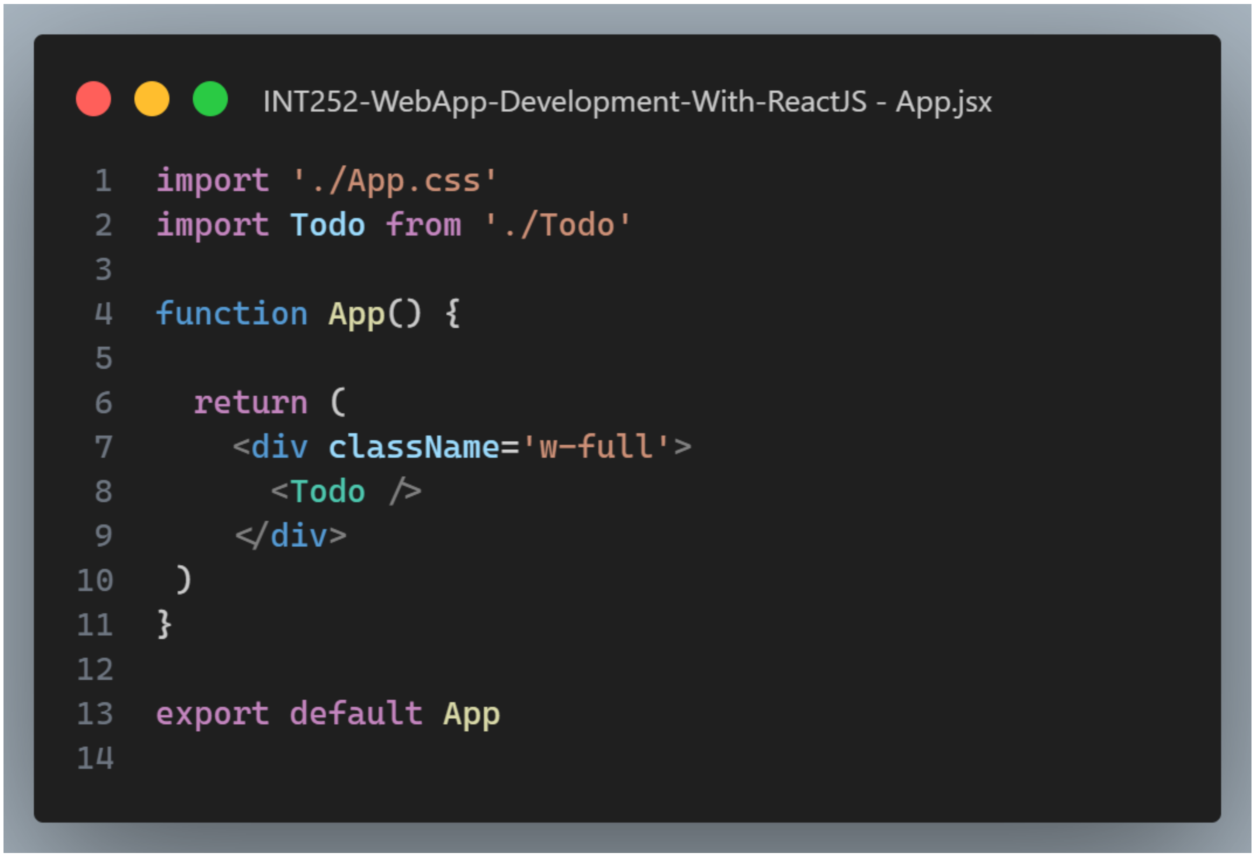
ACTIVITY – USEREDUCER

Name: Vansh Chaurasiya

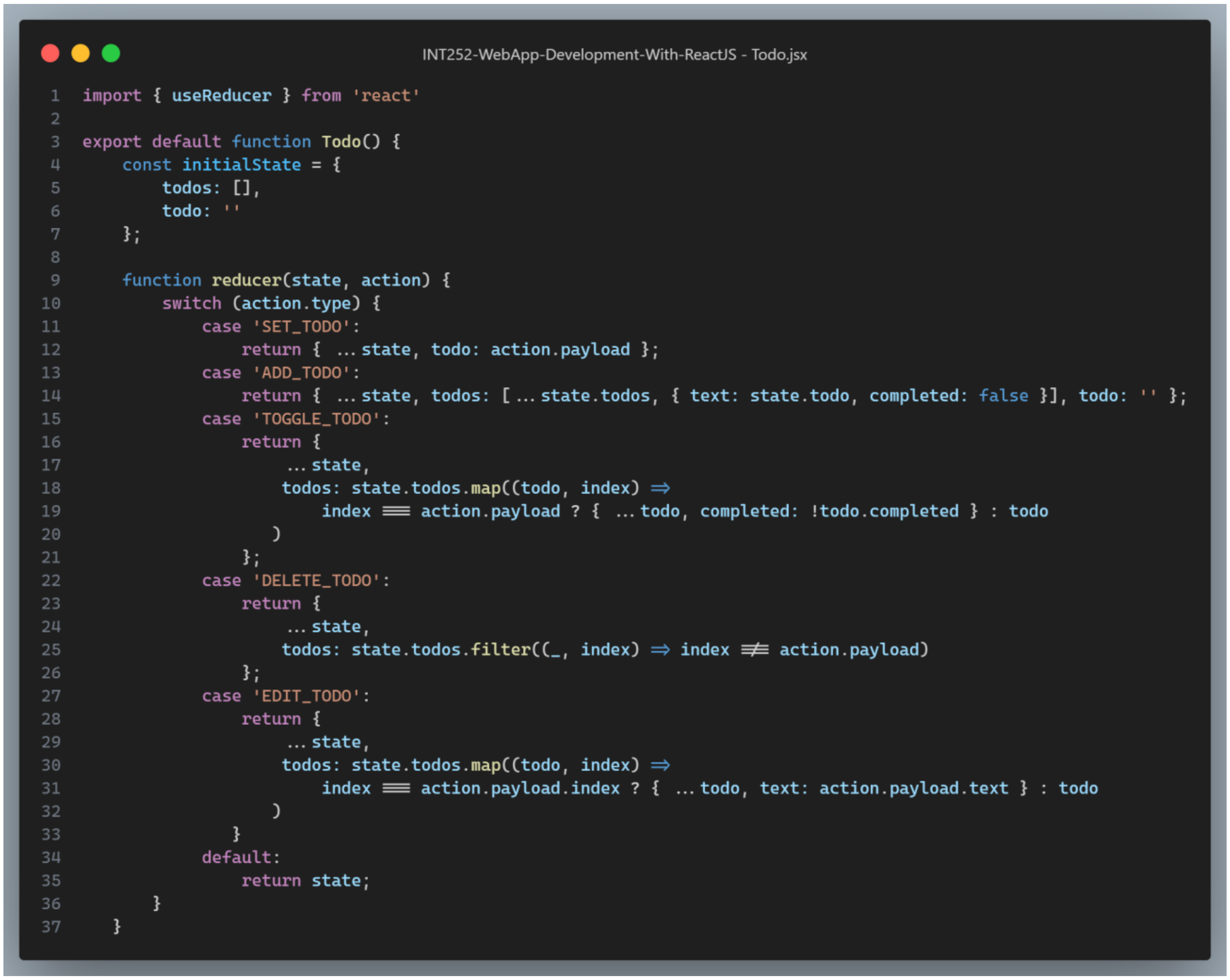
Roll.No: 53

Section: KM006

TODO using UseReducer

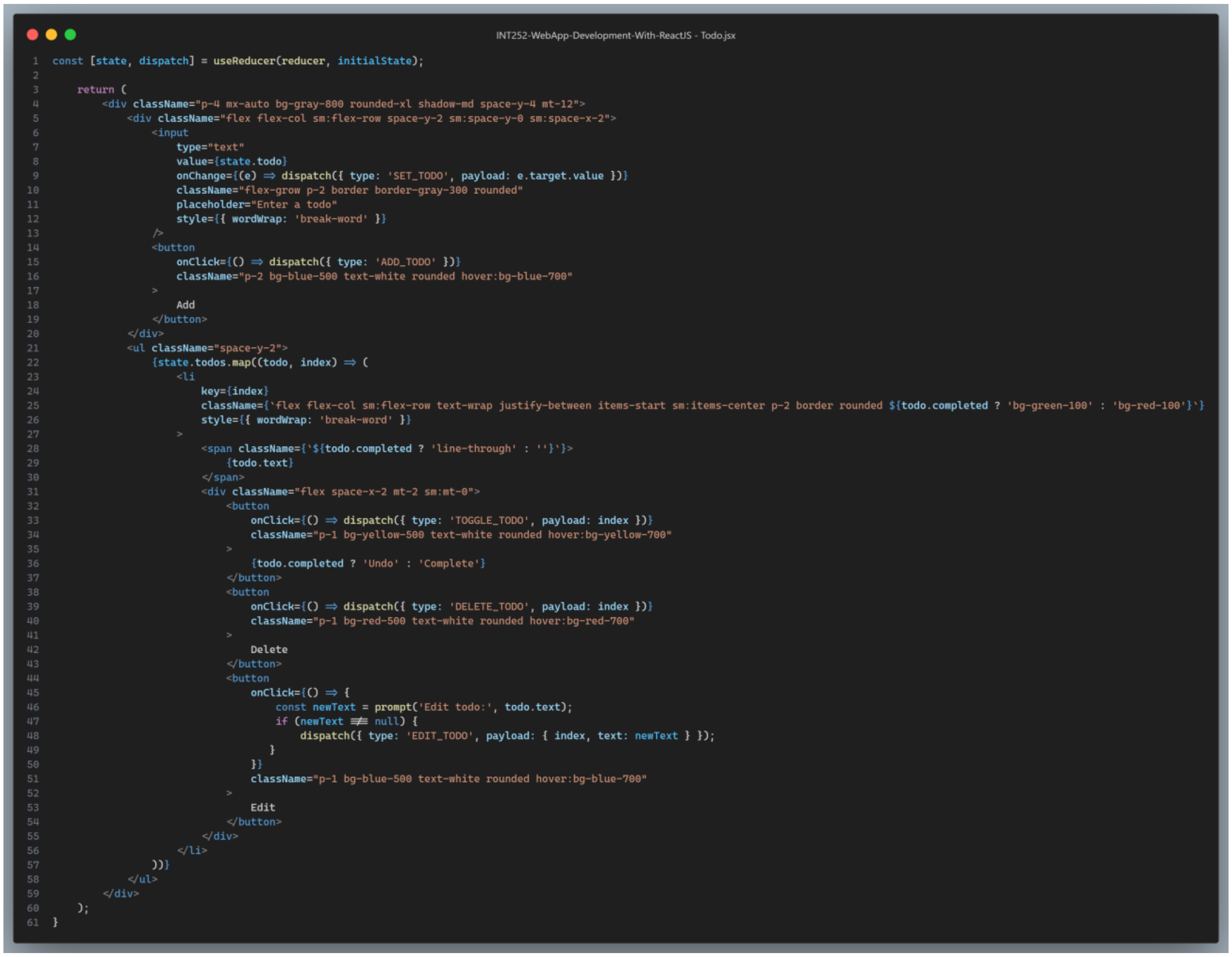


```jsx
import './App.css'
import Todo from './Todo'

function App() {

  return (
    <div className='w-full'>
      <Todo />
    </div>
  )
}

export default App

```



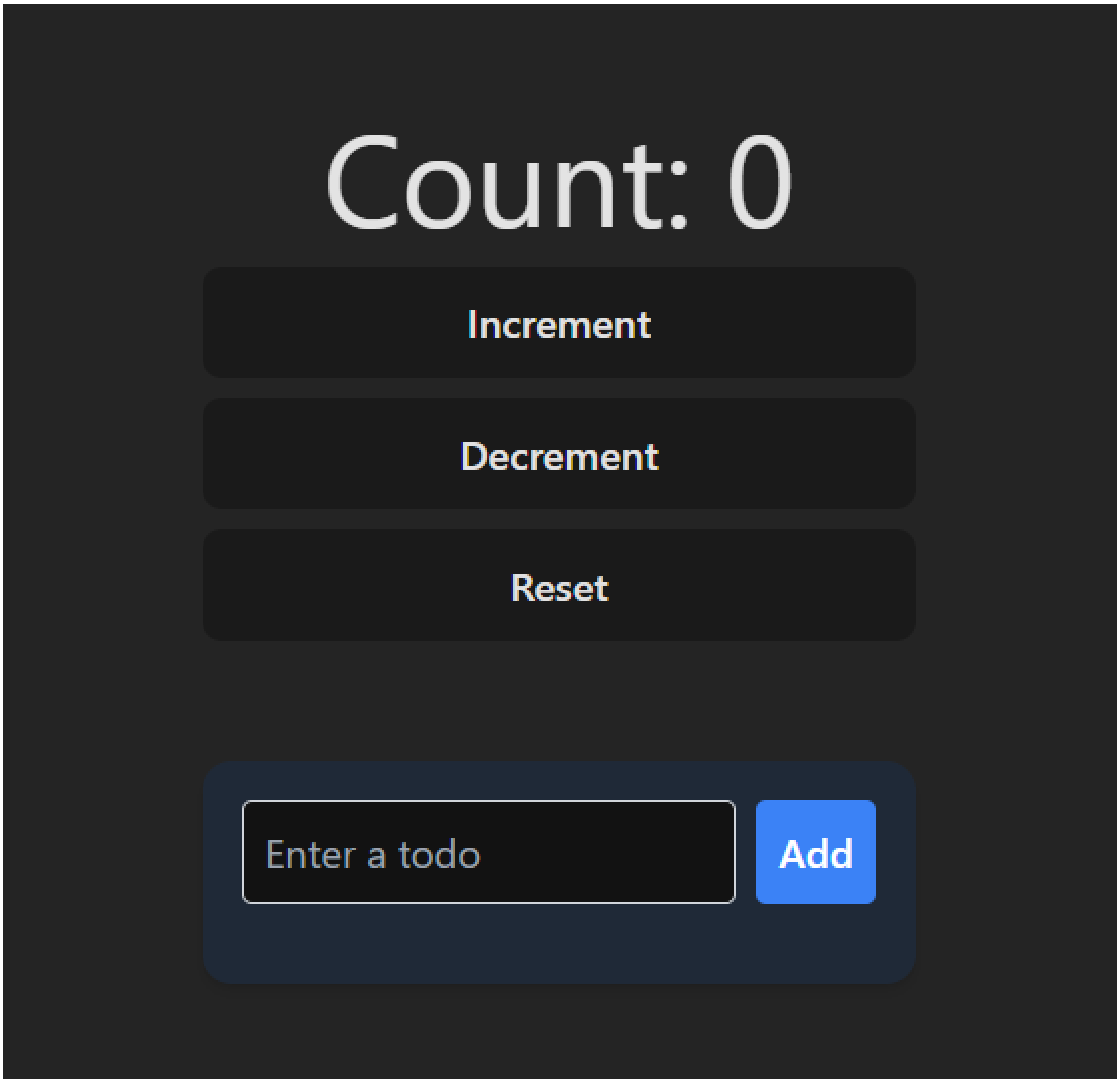
INT252-WebApp-Development-With-ReactJS - Todo.jsx

```jsx
import { useReducer } from 'react'

export default function Todo() {
    const initialState = {
        todos: [],
        todo: ''
    };

    function reducer(state, action) {
        switch (action.type) {
            case 'SET_TODO':
                return { ...state, todo: action.payload };
            case 'ADD_TODO':
                return { ...state, todos: [...state.todos, { text: state.todo, completed: false }], todo: '' };
            case 'TOGGLE_TODO':
                return {
                    ...state,
                    todos: state.todos.map((todo, index) =>
                        index === action.payload ? { ...todo, completed: !todo.completed } : todo
                    )
                };
            case 'DELETE_TODO':
                return {
                    ...state,
                    todos: state.todos.filter((_, index) => index !== action.payload)
                };
            case 'EDIT_TODO':
                return {
                    ...state,
                    todos: state.todos.map((todo, index) =>
                        index === action.payload.index ? { ...todo, text: action.payload.text } : todo
                    )
                }
            default:
                return state;
        }
    }
```

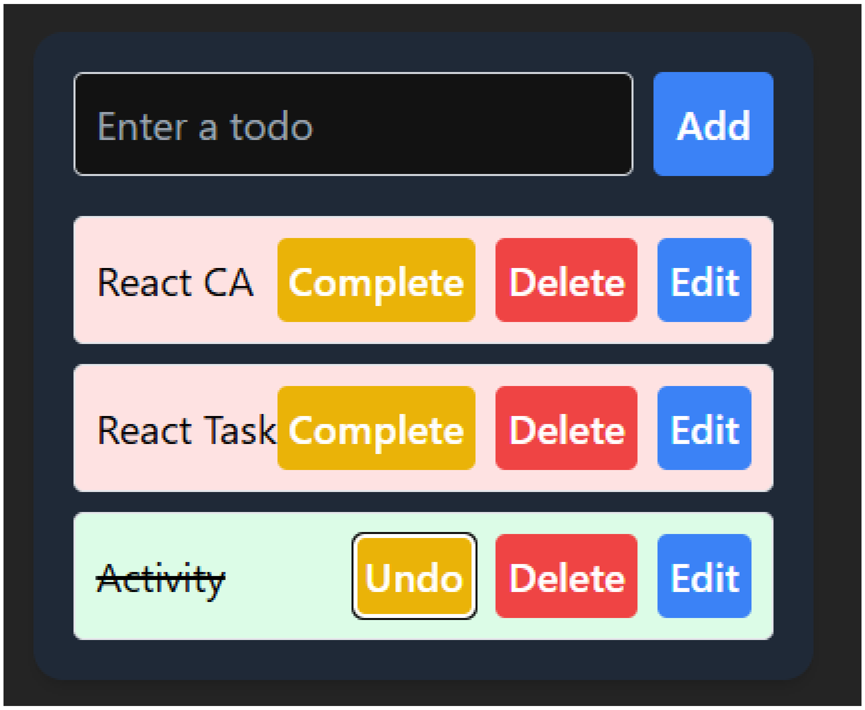


```jsx
const [state, dispatch] = useReducer(reducer, initialState);

    return (
        <div className="p-4 mx-auto bg-gray-800 rounded-xl shadow-md space-y-4 mt-12">
            <div className="flex flex-col sm:flex-row space-y-2 sm:space-y-0 sm:space-x-2">
                <input
                    type="text"
                    value={state.todo}
                    onChange={(e) => dispatch({ type: 'SET_TODO', payload: e.target.value })}
                    className="flex-grow p-2 border border-gray-300 rounded"
                    placeholder="Enter a todo"
                    style={{ wordWrap: 'break-word' }}
                />
                <button
                    onClick={() => dispatch({ type: 'ADD_TODO' })}
                    className="p-2 bg-blue-500 text-white rounded hover:bg-blue-700"
                >
                    Add
                </button>
            </div>
            <ul className="space-y-2">
                {state.todos.map((todo, index) => (
                    <li
                        key={index}
                        className={`flex flex-col sm:flex-row text-wrap justify-between items-start sm:items-center p-2 border rounded ${todo.completed ? 'bg-green-100' : 'bg-red-100'}`}
                        style={{ wordWrap: 'break-word' }}
                    >
                        <span className={`${todo.completed ? 'line-through' : ''}`}>
                            {todo.text}
                        </span>
                        <div className="flex space-x-2 mt-2 sm:mt-0">
                            <button
                                onClick={() => dispatch({ type: 'TOGGLE_TODO', payload: index })}
                                className="p-1 bg-yellow-500 text-white rounded hover:bg-yellow-700"
                            >
                                {todo.completed ? 'Undo' : 'Complete'}
                            </button>
                            <button
                                onClick={() => dispatch({ type: 'DELETE_TODO', payload: index })}
                                className="p-1 bg-red-500 text-white rounded hover:bg-red-700"
                            >
                                Delete
                            </button>
                            <button
                                onClick={() => {
                                    const newText = prompt('Edit todo:', todo.text);
                                    if (newText !== null) {
                                        dispatch({ type: 'EDIT_TODO', payload: { index, text: newText } });
                                    }
                                }}
                                className="p-1 bg-blue-500 text-white rounded hover:bg-blue-700"
                            >
                                Edit
                            </button>
                        </div>
                    </li>
                ))}
            </ul>
        </div>
    );
}
```

Output:



Count: 0


Enter a todo
Add
React CA
Complete
Delete
Edit
React Task
Complete
Delete
Edit
~~Activity~~
Undo
Delete
Edit

COMPLETED