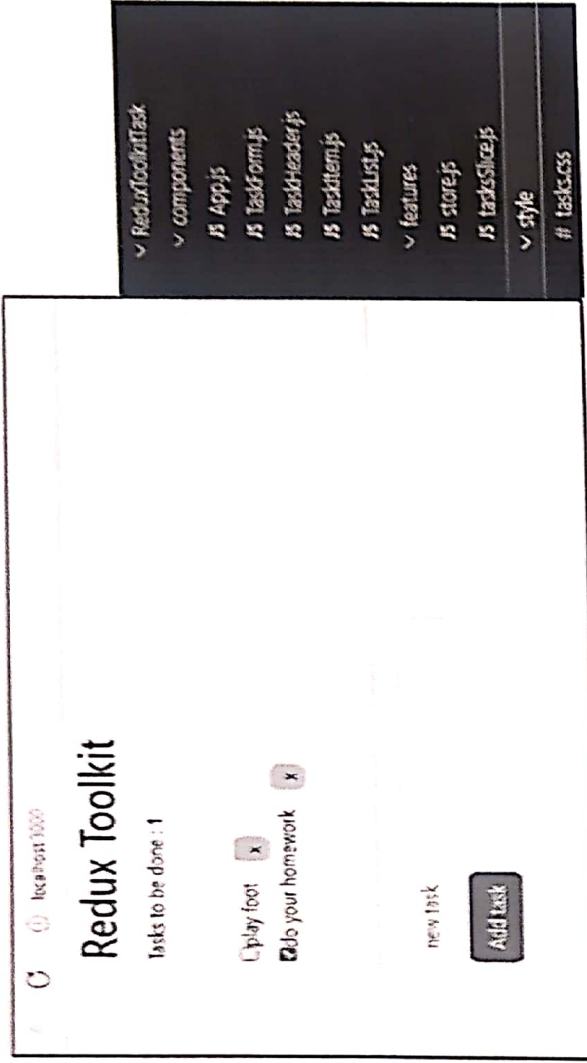


Redux Toolkit Tasks



App.js

```
import React from "react";
import { Provider } from "react-redux";
import TaskForm from "./TaskForm";
import TaskHeader from "./TaskHeader";
import TaskList from "./TaskList";
import {store} from '../features/store'
import '../style/tasks.css'
```

```
export default function App(){
  return (
    <Provider store={store}>
      <div className="container">
        <TaskHeader/>
```

```
      <hr/>
      <TaskList/>
    </div>
    <TaskForm/>
```

```
  </div>
</Provider>
)
}
```

TaskHeader.js

```
import React, { useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";

export default function TaskHeader(){
  const tasks=useSelector((state)=>state.tasks)
  const undoneTasks=tasks.filter((item)=>item.done===false)

  return (<div>
    <h1>Redux Toolkit </h1>
    <p>Tasks to be done : <b>{undoneTasks.length}</b></p>
    </div>
  )
}
```

TaskList.js

```
import React from "react";
import { useSelector } from "react-redux";

import TaskItem from './TaskItem'

export default function TaskList(props){
  const tasks=useSelector((state)=> state.tasks)
  return (<div>
    {tasks.map((item)=>(
      <TaskItem key={item.id} task={item}/>
    ))}
    </div>
  )
}
```

TaskItem.js

```
import React from "react";
import { useDispatch } from "react-redux";
import { deleteTask, toggleTask } from "../features/tasksSlice";

export default function TaskItem(props){
  const {task}=props;
  const dispatch=useDispatch();
  const handleClick=()=>{
    dispatch(deleteTask(task.id))
  }
  return (<div>
    <input type="checkbox" checked={task.done} onChange={()=>
      dispatch(toggleTask(task.id))}/>{task.text}
    <button onClick={handleClick} className="btn btn-warning"
      style={{padding: "0px", marginLeft: "20px", height: "25px", width: "25px"}}>x</button>
    </div>
  )
}
```

TaskForm.js

```
import React, { useState } from "react";
import { useDispatch, useSelector } from "react-redux";
import { addTask } from "../features/tasksSlice";
export default function TaskForm(){
  const [text, setText]=useState('')

  const dispatch=useDispatch();
  const handleClick=(e)=>{
    e.preventDefault() ;
    dispatch(addTask(text))
    setText('')
  }

  return (<form>

<input className="form-control mb-2" onChange={(e)=>setText(e.target.value)}
placeholder="new task" type="text" value={text} name="task"/>
<button className="btn btn-success" onClick={handleClick}>Add task</button>

    </form>
  )

  store.js

import { configureStore } from "@reduxjs/toolkit";

import tasksReducer from './tasksSlice.js'
export const store=configureStore({
  reducer:{
    tasks:tasksReducer
  }
})
```

tasksSlice.js

```
import { createSlice, current } from "@reduxjs/toolkit";

const todoSlice=createSlice( {
  name:"todo",
  initialState:[
    {id:1,text:"play foot",done:false},
    {id:2,text:"do your homework",done:true}
  ],
  reducers: {
    //todo/addTask
    addTask:(state,action)=>{
      const newTask={id:Date.now(),
        done:false,
        text:action.payload}
      state.push(newTask)
    },
    toggleTask:(state,action)=>{
      const task=state.find(t=>t.id==action.payload)
      task.done=!task.done
    },
    deleteTask:(state,action)=>{
      return state.filter(item=>item.id!=action.payload)
    }
    // const index=state.findIndex(item=>item.id==action.payload)
    // state.splice(index,1)
  }
})
export const {addTask,deleteTask,toggleTask}=todoSlice.actions
export default todoSlice.reducer
```