**PRACTICAL 11**

**AIM :** Fullstack todo application

BACKEND CODE :

* Connection.js

const mongoose = require('mongoose');

try{

    mongoose.connect('mongodb+srv://21it008:wvPrTg3ykTB9glFK@cluster0.guwpnah.mongodb.net/Todo\_Database?retryWrites=true&w=majority').then(()=>console.log("connected"));

}catch(e){

    console.log("error connecting to databse !");

}

* Model.js

const { Schema, model } = require('mongoose');

const Todoschema = new Schema({

  title: {type:String,required:true,trim:true},

  description: {type:String,required:true,trim:true},

  status:{type:Boolean,required:true,default:false},

},{

  timestamps:true

});

module.exports=model('Todo-List',Todoschema);

* Routing.js

const express = require('express')

const app = express();

app.use(express.json());

const router=express.Router();

const Todo=require('../Model/Todo');

router.post('/todo',(req,res)=>{

    try{

        const newTodo=new Todo(req.body);

        newTodo.save();

        res.json(newTodo);

    }catch(err){

        res.json(err.message);

    }

})

router.get('/todo',async (req,res)=>{

    const newTodo=await Todo.find();

    res.json(newTodo);

})

// router.get('/todo/:id',async (req,res)=>{

//     console.log(req.params.id);

//     const newTodo=await Todo.findById(req.params.id);

//     if(newTodo){

//     res.json(newTodo);

//     }else{

//         res.json("no data found");

//     }

// })

// router.put('/todo/:id',async (req,res)=>{

//     const newTodo=await Todo.findByIdAndUpdate(req.params.id,req.body,{new:true});

//     if(newTodo){

//         res.json(newTodo)

//     }else{

//         res.json("no data found")

//     }

// })

router.delete('/todo/:id',async (req,res)=>{

    try{

        await Todo.findByIdAndDelete(req.params.id)

        res.json("deleted successfully")

    }catch(err){

        res.json("no data found")

    }

})

module.exports = router;

* App.js

const express= require('express');

const app= express();

const cors= require('cors');

const todorouter=require('./Routes/routing');

app.use(cors());

app.listen(8000,() => console.log("listening on 8000"));

app.use(express.json());

app.use('/',todorouter);

require('./connection/conn.js')

FRONTEND CODE:

* Home.jsx

import React,{useState,useEffect} from 'react';

import axios from 'axios';

import '../App.css'

export default function Home() {

    const [Alltask,setAlltask]=useState([]);

    const [Task,setTask]=useState({

        title:'',

        description:'',

        status:false

    });

    useEffect(() => {

        try{

            axios.get('http://localhost:8000/todo').then((response) =>{setAlltask(response.data)})

        }catch(e){

            console.error(e);

        }

    },Alltask)

    async function addTask(event){

        event.preventDefault();

        const res=await axios.post('http://localhost:8000/todo',Task)

        if(res){

            console.log("Task added")

        }

        else{

        console.log("Task not added")

        }

        window.location.reload();

    }

    function handlechange(e){

        const { name, value } = e.target;

        setTask(task => ({

          ...task,

          [name]: value

        }));

    }

function deleteTask(id){

    console.log("deleting")

    console.log(id)

axios.delete(`http://localhost:8000/todo/${id}`).then((res)=>{

    console.log("Task deleted")

    window.location.reload();

})

}

    return (

    <>

    <div className="main-todo">

   <div className="inner-todo">

        <h2>TODO LIST</h2>

        <div>

        <input type="text" placeholder='Add Task Title' name='title' value={Task.title} onChange={handlechange}/>

        <input type="text" placeholder='Add Task Description' name='description' value={Task.description} onChange={handlechange} />

        <button onClick={addTask}>submit</button>

        </div>

        All TASKS

        {Alltask.map(task => (

          <div className='box' key={task.\_id}><p className='title'>Title : {task.title}</p>

             <p className='dis'>Description : {task.description}</p>

                <button onClick={()=>deleteTask(task.\_id)}>delete</button>

          </div>

        ))}

         </div>

         </div>

    </> );

}

OUTPUT :

  