**AIM: Create modern, scalable and high-speed Web Applications with React and Node.js +Express.**

**CODE:**

const mongoose = require('mongoose');

import React, { useState } from 'react';

import { useNavigate } from 'react-router-dom';

import axios from 'axios';

const AddTask = () => {

  const [title, setTitle] = useState('');

  const [description, setDescription] = useState('');

  const navigate = useNavigate();

  const handleSubmit = async (e) => {

    e.preventDefault();

       if (!title) {

      alert('Please add a title');

      return; }

       try {

      await axios.post('http://localhost:5000/api/tasks', {

        title,

        description,

        completed: false});

      navigate('/');

    } catch (err) {

      console.error(err);}};

export default AddTask;

<div className="task-actions">

        {!task.completed && (

          <button

            className="btn btn-complete"

            onClick={() => toggleComplete(task.\_id, task.completed)}  >

            Mark Complete

          </button>)}

        <button className="btn btn-delete" onClick={() => deleteTask(task.\_id)}>

          Delete

        </button>

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

import axios from 'axios';

import TaskItem from './TaskItem';

const TaskList = () => {

  const [tasks, setTasks] = useState([]);

  const [loading, setLoading] = useState(true);

  useEffect(() => {

    const fetchTasks = async () => {

      try {

        const res = await axios.get('http://localhost:5000/api/tasks');

        setTasks(res.data);

        setLoading(false);

      } catch (err) {

        console.error(err);

        setLoading(false);}};

    fetchTasks();

  }, []);

  const deleteTask = async (id) => {

    try {

      await axios.delete(`http://localhost:5000/api/tasks/${id}`);

      setTasks(tasks.filter(task => task.\_id !== id));

    } catch (err) {

      console.error(err); }};

  const toggleComplete = async (id, completed) => {

    try {

      const res = await axios.patch(`http://localhost:5000/api/tasks/${id}`, {

        completed: !completed  });

      setTasks(tasks.map(task =>

        task.\_id === id ? { ...task, completed: res.data.completed } : task ));

    } catch (err) {

      console.error(err); }};

  if (loading) {

    return <h2>Loading...</h2>;}

  return (

    <div className="task-list">

      <h2>Your Tasks</h2>

      {tasks.length === 0 ? (

        <p>No tasks found. Add some tasks!</p>

      ) : (

        tasks.map(task => (

          <TaskItem

            key={task.\_id}

            task={task}

            deleteTask={deleteTask}

            toggleComplete={toggleComplete}

          />)))}

    </div> );};

export default TaskList;

const express = require('express');

const router = express.Router();

const Task = require('../models/Task');

// Get all tasks

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

  try {

    const tasks = await Task.find().sort({ createdAt: -1 });

    res.json(tasks);

  } catch (err) {

    res.status(500).json({ message: err.message });

  }});

// Create a task

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

  const task = new Task({

    title: req.body.title,

    description: req.body.description,

    completed: req.body.completed });

  try {

    const newTask = await task.save();

    res.status(201).json(newTask);

  } catch (err) {

    res.status(400).json({ message: err.message });

  }});

// Update a task

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

  try {

    const task = await Task.findById(req.params.id);

    if (!task) {

      return res.status(404).json({ message: 'Task not found' }); }

    if (req.body.title) task.title = req.body.title;

    if (req.body.description) task.description = req.body.description;

    if (req.body.completed !== undefined) task.completed = req.body.completed;

    const updatedTask = await task.save();

    res.json(updatedTask);

  } catch (err) {

    res.status(500).json({ message: err.message });

  }});

// Delete a task

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

  try {

    const task = await Task.findById(req.params.id);

       if (!task) {

      return res.status(404).json({ message: 'Task not found' });}

       // Replace task.remove() with findByIdAndDelete

    await Task.findByIdAndDelete(req.params.id);

    res.json({ message: 'Task deleted' });

  } catch (err) {

    res.status(500).json({ message: err.message });}});

module.exports = router;

**OUTPUT:**



