Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_11,12\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **01** | Create a todo with MERN technology. |
|  |  |
|  |  |
|  |  |

Submitted On

28-12-2023

(Date: DD/MM/YY)

**Task 01:** Create a todo with MERN technology.

**Solution:**

**Server.js**

const express=require('express');

const mongoose=require('mongoose');

const bodyParser=require('body-parser');

const cors=require('cors');

const {dbConnect}=require('./dbConnect')

const {addTodo}=require('./route/todoRoute')

const app=express();

const PORT=process.env.PORT || 5000;

app.use(cors())

app.use(bodyParser.json());

dbConnect();

app.post('/add-todo',addTodo)

app.listen(PORT, () => {

console.log(`Server is running athttp://localhost:${PORT}`);})

**dbConnect.js:**

const mongoose=require('mongoose');

const dbConnect=async ()=>{

try {

    const response=await mongoose.connect('mongodb+srv://shoaibakhter181422:todo@cluster0.x3d8fsb.mongodb.net/',{useNewUrlParser: true,

    useUnifiedTopology: true,})

    console.log("database is connected");

} catch (error) {

    console.log(error);}}

module.exports={dbConnect};

**todoModel.js**

const mongoose=require('mongoose');

const todoSchema=mongoose.Schema({task:String})

const Todo=mongoose.model("TodoModel",todoSchema)

module.exports=Todo;

**todoRoute:**

const Todo = require("../model/todoModel");

const addTodo=async (req,res)=>{

    try {

        const {task}=req.body;

    const newTodo=new Todo({task});

    await newTodo.save();

    res.json({'message':"data is uploaded"});

    } catch (error) {

        console.log(error);}}

module.exports={addTodo}

**App.jsx:**

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

import axios from "axios";

function App() {

  const [task, setTask] = useState("");

  const [todos, setTodo] = useState([]);

  useEffect(() => { const fetchTodos = async () => {try {

        const response = await axios.get("http://localhost:5000/get-todo");

        console.log(response.data);

        setTodo(response.data);

      } catch (error) {    fetchTodos();

  }, [todos]);

  const addTodo = async () => {

    try {

  await axios.post("http://localhost:5000/add-todo", { task});

      setTask("");

    } catch (error) {

  return (

    <>

      <h1>Todo App</h1>

      {editMode ? (

        <><inputtype="text" value={task}

            onChange={(e) => setTask(e.target.value)}/>

          <button onClick={() => updateTodo(editId)}>Update</button>

    <button onClick={cancelEditMode}>Cancel</button></>) : (<>

<input type="text" value={task} onChange={(e) => setTask(e.target.value)}/>

          <button onClick={addTodo}>Add Task</button></>)}

      {todos.map((todo) => (

        <div key={todo.\_id}>

          <ul>{todo.task}</ul>

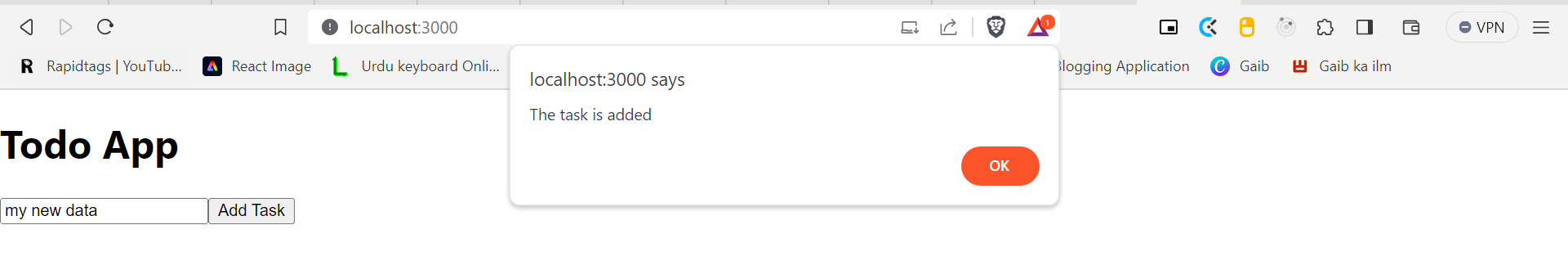
   <button onClick={() => enterEditMode(todo.\_id, todo.task)}>

            Edit</button>

          <button onClick={() => deleteTodo(todo.\_id)}>Delete</button></div>))}</>);}

export default App;

**Output:**

****

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**