



COLLEGE CODE: 9111

COLLEGE :SRM Madurai College for Engineering and Technology

NAME : R.Karishma

DEPARTMENT : Information technology

STUDENT NM-ID :EEC121412BECD8580134A2ADB6F76377

ROLL NO : 23it18

DATE : 22/09/2025

Completed the project named as Phase_3_

TECHNOLOGY PROJECT NAME : Todo list app

SUBMITTED BY,

NAME :Karishma.R

MOBILE NO :7904410709

1. Project Setup

- Initialized the **MERN stack environment** (MongoDB, Express.js, React.js, Node.js).
 - Installed necessary dependencies using **npm/yarn** (e.g., express, mongoose, cors, react-router, axios).
 - Configured **backend server** with Express.js to handle REST API requests.
 - Set up **React.js frontend** with components for task listing and task creation.
 - Established **connection to MongoDB** for data storage using Mongoose.
-

2. Core Features Implementation

The following **CRUD operations** were implemented:

- **Create Task:** Add a new task with task name, status, and deadline.
 - **Read Tasks:** Display all tasks in a tabular format.
 - **Update Task:** Edit task details (status or deadline).
 - **Delete Task:** Remove unwanted or completed tasks.
 - **Deadline Management:** Store and display deadlines for each task.
-

3. Data Storage (Local State / Database)

- **Frontend (React.js):** Managed temporary data using **React useState & useEffect hooks**.
 - **Backend (Node.js + Express.js):** Handled API endpoints for CRUD operations.
 - **Database (MongoDB):** Persisted task data to ensure that tasks remain available even after refreshing or closing the app.
-

4. Testing Core Features

- Verified **task addition** by checking if new tasks appear in the task list.
 - Tested **edit functionality** to confirm that task status and deadlines update correctly.
 - Checked **delete functionality** to ensure tasks are removed from both UI and database.
 - Used **Postman** to test backend API endpoints independently.
 - Performed **UI testing** to ensure smooth interaction between frontend and backend.
-

5. Version Control (GitHub)

- Initialized a **Git repository** for the project.
- Pushed the codebase to **GitHub** for version control.
- Used **branches** for developing features separately before merging into the main branch.
- Maintained commit history for tracking progress and debugging issues

CODINGS and output

```
<!-- index.html -->

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Todo List</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>

  </body>
</html>
```

```
//App.js

import React from 'react';
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import 'bootstrap/dist/css/bootstrap.min.css';
import Todo from './components/Todo';

function App() {
  const headStyle = {
    textAlign: "center",
  }
  return (
    <div>
      <h1 style={headStyle}>Todo List</h1>
      <BrowserRouter>
        <Routes>
          <Route path="/" element={<Todo/>}></Route>
        </Routes>
      </BrowserRouter>
    </div>
```

```
);  
}
```

```
export default App;
```

```
import axios from "axios";  
import React from "react";  
import { useEffect, useState } from "react";  
  
function Todo() {  
  const [todoList, setTodoList] = useState([]);  
  const [editableId, setEditableId] = useState(null);  
  const [editedTask, setEditedTask] = useState("");  
  const [editedStatus, setEditedStatus] = useState("");  
  const [newTask, setNewTask] = useState("");  
  const [newStatus, setNewStatus] = useState("");  
  const [newDeadline, setNewDeadline] = useState("");  
  const [editedDeadline, setEditedDeadline] = useState("");  
  
  // Fetch tasks from database  
  useEffect(() => {  
    axios.get('http://127.0.0.1:3001/getTodoList')  
      .then(result => {  
        setTodoList(result.data)  
      })  
      .catch(err => console.log(err))  
  }, [])  
  
  // Function to toggle the editable state for a specific row  
  const toggleEditable = (id) => {  
    const rowData = todoList.find((data) => data._id === id);  
    if (rowData) {  
      setEditableId(id);  
      setEditedTask(rowData.task);  
      setEditedStatus(rowData.status);  
      setEditedDeadline(rowData.deadline || "");  
    } else {  
      setEditableId(null);  
      setEditedTask("");  
      setEditedStatus("");  
      setEditedDeadline("");  
    }  
  }  
};
```

```

// Function to add task to the database
const addTask = (e) => {
  e.preventDefault();
  if (!newTask || !newStatus || !newDeadline) {
    alert("All fields must be filled out.");
    return;
  }

  axios.post('http://127.0.0.1:3001/addTodoList', { task: newTask, status:
newStatus, deadline: newDeadline })
    .then(res => {
      console.log(res);
      window.location.reload();
    })
    .catch(err => console.log(err));
}

// Function to save edited data to the database
const saveEditedTask = (id) => {
  const editedData = {
    task: editedTask,
    status: editedStatus,
    deadline: editedDeadline,
  };

  // If the fields are empty
  if (!editedTask || !editedStatus || !editedDeadline) {
    alert("All fields must be filled out.");
    return;
  }

  // Updating edited data to the database through updateById API
  axios.post('http://127.0.0.1:3001/updateTodoList' + id, editedData)
    .then(result => {
      console.log(result);
      setEditableId(null);
      setEditedTask("");
      setEditedStatus("");
      setEditedDeadline(""); // Clear the edited deadline
      window.location.reload();
    })
    .catch(err => console.log(err));
}

```

```

// Delete task from database
const deleteTask = (id) => {
  axios.delete('http://127.0.0.1:3001/deleteTodoList' + id)
    .then(result => {
      console.log(result);
      window.location.reload();
    })
    .catch(err =>
      console.log(err)
    )
}

return (
  <div className="container mt-5">
    <div className="row">
      <div className="col-md-7">
        <h2 className="text-center">Todo List</h2>
        <div className="table-responsive">
          <table className="table table-bordered">
            <thead className="table-primary">
              <tr>
                <th>Task</th>
                <th>Status</th>
                <th>Deadline</th>
                <th>Actions</th>
              </tr>
            </thead>
            {Array.isArray(todoList) ? (
              <tbody>
                {todoList.map((data) => (
                  <tr key={data._id}>
                    <td>
                      {editableId === data._id ? (
                        <input
                          type="text"
                          className="form-control"
                          value={editedTask}
                          onChange={(e) =>
setEditedTask(e.target.value)}
                        </td>
                      ) : (
                        data.task
                      )}
                    </td>
                    <td>

```

```

                {editableId === data._id ? (
                    <input
                        type="text"
                        className="form-control"
                        value={editedStatus}
                        onChange={(e) =>
setEditedStatus(e.target.value)}

                    />
                ) : (
                    data.status
                )}
            </td>
            <td>
                {editableId === data._id ? (
                    <input
                        type="datetime-local"
                        className="form-control"
                        value={editedDeadline}
                        onChange={(e) =>
setEditedDeadline(e.target.value)}

                    />
                ) : (
                    data.deadline ? new
Date(data.deadline).toLocaleString() : ''
                )}
            </td>
            <td>
                {editableId === data._id ? (
                    <button className="btn btn-
success btn-sm" onClick={() => saveEditedTask(data._id)}>
                        Save
                    </button>
                ) : (
                    <button className="btn btn-
primary btn-sm" onClick={() => toggleEditable(data._id)}>
                        Edit
                    </button>
                )}
                <button className="btn btn-danger
btn-sm ml-1" onClick={() => deleteTask(data._id)}>
                    Delete
                </button>
            </td>
        </tr>

```



```

        ))}
      </tbody>
    ) : (
      <tbody>
        <tr>
          <td colspan="4">Loading products...</td>
        </tr>
      </tbody>
    )}
  </table>
</div>
</div>
<div className="col-md-5">
  <h2 className="text-center">Add Task</h2>
  <form className="bg-light p-4">
    <div className="mb-3">
      <label>Task</label>
      <input
        className="form-control"
        type="text"
        placeholder="Enter Task"
        onChange={(e) => setNewTask(e.target.value)}
      />
    </div>
    <div className="mb-3">
      <label>Status</label>
      <input
        className="form-control"
        type="text"
        placeholder="Enter Status"
        onChange={(e) => setNewStatus(e.target.value)}
      />
    </div>
    <div className="mb-3">
      <label>Deadline</label>
      <input
        className="form-control"
        type="datetime-local"
        onChange={(e) => setNewDeadline(e.target.value)}
      />
    </div>
    <button onClick={addTask} className="btn btn-success btn-sm">

```

```

        Add Task
      </button>
    </form>
  </div>

</div>
</div>
)
}
export default Todo;

```

//index.js - Serves as the entry point for your React application

```

import React from 'react';
import ReactDOM from 'react-dom';
import App from './App'; // Your main application component

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
)

```

Add Task

Task

Enter Task

Status

Enter Status

Deadline

dd-----yyyy --:-- --

Add Task

Todo List

Task	Status	Deadline	Actions
Complete Assignment	Completed	10/11/2023, 12:00:00 PM	<button>Edit</button> <button>Delete</button>
Submit Project	Completed	10/11/2023, 5:22:00 PM	<button>Edit</button> <button>Delete</button>
Practice DSA	Ongoing	10/11/2023, 7:00:00 PM	<button>Edit</button> <button>Delete</button>
Fix the bug	Pending	10/11/2023, 8:30:00 PM	<button>Edit</button> <button>Delete</button>

Todo List

Todo List

Task	Status	Deadline	Actions
Practice DSA	Completed	10/11/2023, 7:10:00 AM	<button>Edit</button> <button>Delete</button>
Complete Assignment	Pending	10/11/2023, 12:00:00 PM	<button>Edit</button> <button>Delete</button>

Add Task

Task

Status

Deadline

Add Task

Activate Windows
Go to Settings to activate Windows

Task	Status	Deadline	Actions
Practice DSA	Completed	10/11/2023, 7:10:00 AM	<button>Edit</button> <button>Delete</button>
Complete Assignment	Pending	10/11/2023, 12:00:00 PM	<button>Edit</button> <button>Delete</button>

October 2023
↑
↓

12
22
AM

01 23 PM
02 24
03 25
04 26
05 27
06 28

Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 1 2 3 4
5 6 7 8 9 10 11

Clear Today

dd-----yyyy --:-- --

Add Task