

# Task Manager App

Code:

## 1. Project Folder Structure:

The screenshot shows a file explorer window with the following folder structure:

- task-manager
  - node\_modules
  - index.js
  - package-lock.json
  - package.json
  - task-manager-IP.code-workspace
- task-manager-frontend
  - node\_modules
  - public
  - src
    - Pages
      - AddTask.js
      - DeleteTask.js
      - FormStyles.css
      - task-manager-IP.code-workspace
      - UpdateTask.js
    - # App.css
    - App.js
    - # index.css
    - index.js
    - task-bg.png
    - task-bg.png.png
  - .gitignore
  - package-lock.json
  - package.json
  - README.md

At the bottom of the file explorer, there are two buttons: > OUTLINE and > TIMELINE.

## 2. Backend Code Snippets

<pre>JS index.js ● {} task-manager-IPcode-workspace task-manager &gt; JS index.js &gt; ... 1  const express = require("express"); 2  const mongoose = require("mongoose"); 3  const cors = require("cors"); 4 5  const app = express(); 6  app.use(cors()); 7  app.use(express.json()); 8 9  // Mongodb Connection 10 mongoose.connect("mongodb://127.0.0.1:27017/tasks", { 11   useNewUrlParser: true, 12   useUnifiedTopology: true, 13 }) 14 .then(() =&gt; console.log("MongoDB Connected")) 15 .catch((err) =&gt; console.log("MongoDB Error:", err)); 16 17 // Mongoose Model 18 const Task = mongoose.model("Task", { 19   title: String, 20   completed: Boolean, 21   deadline: String, 22 }); 23 24 // Get All Tasks 25 app.get("/tasks", async (req, res) =&gt; { 26   const tasks = await Task.find(); 27   res.json(tasks); 28 }); 29 30 // Add New Task 31 app.post("/tasks", async (req, res) =&gt; { 32   const task = new Task(req.body); 33   await task.save(); 34   res.json(task); 35 }); 36 37 // Update Task by ID 38 app.put("/tasks/:id", async (req, res) =&gt; { 39   await Task.findByIdAndUpdate(req.params.id, req.body); 40   res.json({ success: true }); 41 }); 42 43 // Delete Task by ID 44 app.delete("/tasks/:id", async (req, res) =&gt; { 45   await Task.findByIdAndDelete(req.params.id); 46   res.json({ success: true }); 47 }); 48</pre>	<pre>JS index.js ● {} task-manager-IPcode-workspace task-manager &gt; JS index.js &gt; ... 48 49 app.listen(5000, () =&gt; { 50   console.log("Server running on http://localhost:5000"); 51 }); 52 // Seed demo tasks 53 app.get("/seed", async (req, res) =&gt; { 54   const demoTasks = [ 55     { 56       title: "Complete Assignment", 57       completed: false, 58       deadline: "2025-06-25", 59     }, 60     { 61       title: "Project PPT", 62       completed: true, 63       deadline: "2025-06-20", 64     }, 65     { 66       title: "Submit Report", 67       completed: false, 68       deadline: "2025-06-30", 69     }, 70   ]; 71 72 try { 73   await Task.insertMany(demoTasks); 74   res.send("Demo tasks inserted!"); 75 } catch (err) { 76   res.status(500).send("Error inserting tasks: " + err); 77 } 78 }); 79</pre>
--	---

### 3. Frontend Code Snippets :

#### App.js:

```
JS App.js ×
task-manager-frontend > src > JS App.js > ...
1 import React, { useEffect, useState } from "react";
2 import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
3 import axios from "axios";
4 import "./App.css";
5 import AddTask from "./Pages/AddTask";
6 import UpdateTask from "./Pages/UpdateTask";
7 import DeleteTask from "./Pages/DeleteTask";
8
9 function App() {
10   const [tasks, setTasks] = useState([]);
11
12   useEffect(() => {
13     axios.get("http://localhost:5000/tasks")
14       .then((res) => setTasks(res.data))
15       .catch((err) => console.error("Error fetching tasks:", err));
16   }, []);
17
18   return (
19     <Router>
20       <div className="app-container">
21         <h1 className="title">Task Manager</h1>
22
23         <div className="nav-buttons">
24           <Link to="/">Home</Link>
25           <Link to="/add">Add Task</Link>
26           <Link to="/update">Update Task</Link>
27           <Link to="/delete">Delete Task</Link>
28         </div>
29
30         <Routes>
31           <Route path="/" element={(
32             <ul style={{ listStyle: "none", padding: 0 }}>
33               {tasks.map((task) => (
34                 <li key={task._id} style={({
35                   background: "#rgba(255,255,255,0.85)",
36                   margin: "10px auto",
37                   padding: "10px 20px",
38                   borderRadius: "10px",
39                   width: "300px",
40                   boxShadow: "0 4px 10px #rgba(0,0,0,0.1)"
41                 })}
42                 <strong>{task.title}</strong><br />
43                 Status: {task.completed ? "✓ Completed" : "✗ Pending"}<br />
44                 Deadline: {task.deadline}<br />
45                 <span style={{ fontSize: "12px", color: "#555" }}>
46                   | Task ID: <code>{task._id}</code>
47                 </span>
48               </li>
49             ))}
50           </ul>
51         </Route>
52         <Route path="/add" element={<AddTask />} />
53         <Route path="/update" element={<UpdateTask />} />
54         <Route path="/delete" element={<DeleteTask />} />
55       </Routes>
56     </div>
57   </Router>
58 }
59 
```

#### AddTask:

```
JS AddTask.js ×
task-manager-frontend > src > Pages > JS AddTask.js > ...
1 import React, { useState } from "react";
2 import axios from "axios";
3 import "./FormStyles.css";
4 import { useNavigate } from "react-router-dom";
5
6 const AddTask = () => {
7   const [title, setTitle] = useState("");
8   const [deadline, setDeadline] = useState("");
9   const navigate = useNavigate();
10
11   const handleAdd = async () => {
12     if (!title || !deadline) {
13       alert("⚠ Please fill all fields");
14       return;
15     }
16
17     try {
18       await axios.post("http://localhost:5000/tasks", {
19         title,
20         completed: false,
21         deadline,
22       });
23       alert("✓ Task added successfully!");
24       navigate("/");
25     } catch (err) {
26       alert("✗ Error adding task");
27       console.log(err);
28     }
29   };
30
31   return (
32     <div className="form-container">
33       <h2 className="form-title">Add New Task</h2>
34       <input
35         type="text"
36         placeholder="Task Title"
37         value={title}
38         onChange={(e) => setTitle(e.target.value)}
39         className="form-input"
40       />
41       <input
42         type="date"
43         value={deadline}
44         onChange={(e) => setDeadline(e.target.value)}
45         className="form-input"
46       />
47       <button onClick={handleAdd} className="form-button">Add Task</button>
48     </div>
49   );
50 }
51
52 export default AddTask;
53 
```

## Update Task:

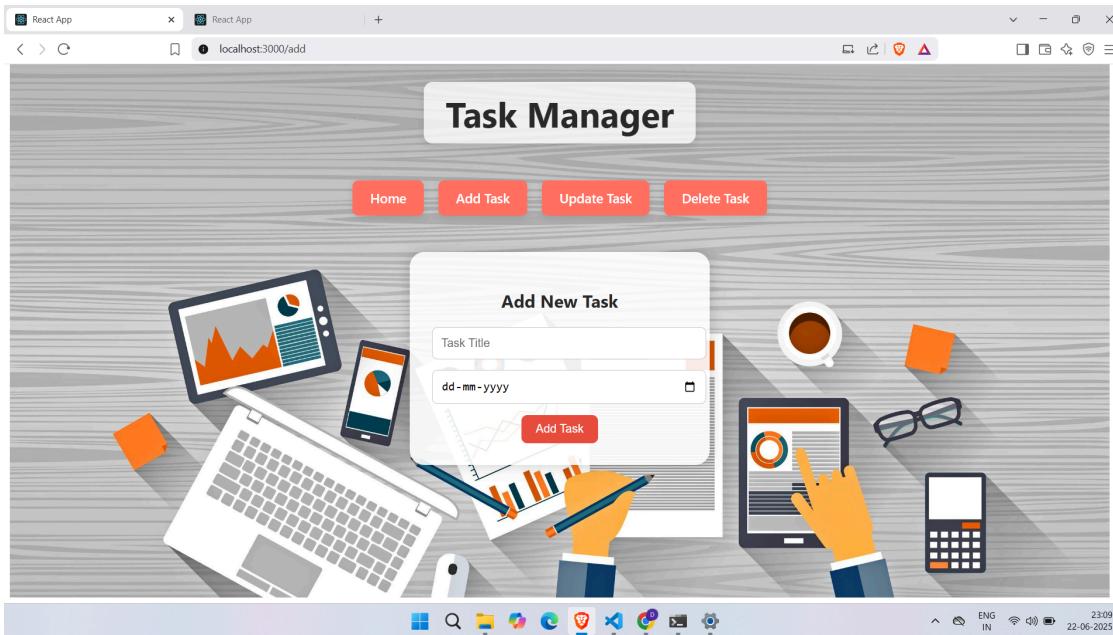
```
JS UpdateTask.js ×
task-manager-frontend > src > Pages > JS UpdateTask.js > ...
1 import React, { useState } from "react";
2 import axios from "axios";
3 import "./FormStyles.css";
4
5 function UpdateTask() {
6   const [taskId, set taskId] = useState("");
7   const [updatedData, set updatedData] = useState({
8     title: "",
9     completed: false,
10    deadline: ""
11  });
12
13  const handleUpdate = async (e) => {
14    e.preventDefault();
15    try {
16      await axios.put(`http://localhost:5000/tasks/${taskId}`, updatedData);
17      alert("✓ Task updated successfully!");
18      set taskId("");
19      set updatedData({ title: "", completed: false, deadline: "" });
20    } catch (error) {
21      alert("✗ Failed to update task");
22      console.error(error);
23    }
24  };
25
26  return (
27    <div className="form-container">
28      <h2 className="form-title">Update Task</h2>
29      <form onSubmit={handleUpdate}>
30        <input
31          className="form-input"
32          type="text"
33          placeholder="Enter Task ID"
34          value={taskId}
35          onChange={(e) => set taskId(e.target.value)}
36          required
37        />
38        <input
39          className="form-input"
40          type="text"
41          placeholder="New Title"
42          value={updatedData.title}
43          onChange={(e) => set updatedData({ ...updatedData, title: e.target.value })}
44        />
45        <input
46          className="form-input"
47          type="date"
48          value={updatedData.deadline}
49          onChange={(e) => set updatedData({ ...updatedData, deadline: e.target.value })}
50        />
51        <label className="form-checkbox">
52          <input
53            type="checkbox"
54            checked={updatedData.completed}
55            onChange={(e) => set updatedData({ ...updatedData, completed: e.target.checked })}
56          />
57          Completed
58        </label>
59        <button className="form-button" type="submit">Update Task</button>
60      </form>
61    </div>
62  );
63
64  export default UpdateTask;
65
66
```

## Delete Task:

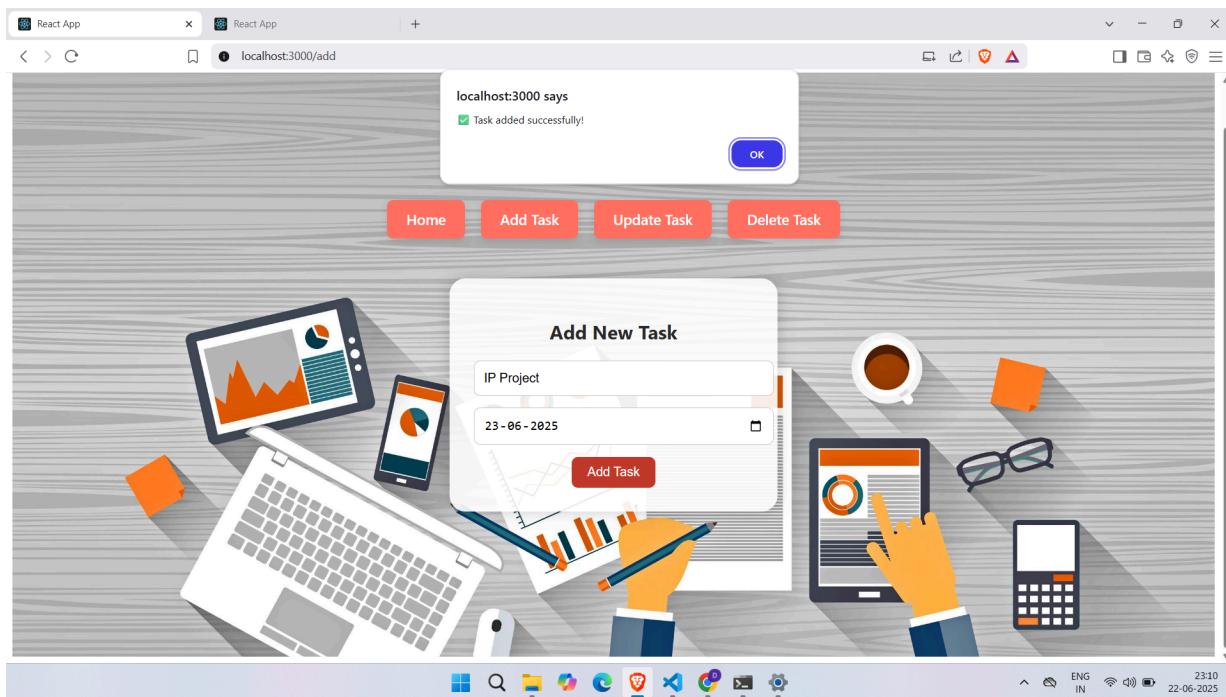
```
JS DeleteTask.js ×
task-manager-frontend > src > Pages > JS DeleteTask.js > ...
1 import React, { useState } from "react";
2 import axios from "axios";
3 import "./FormStyles.css";
4
5 function DeleteTask() {
6   const [taskId, set taskId] = useState("");
7   const [message, set message] = useState("");
8
9   const handleDelete = async () => {
10     if (!taskId || taskId.length !== 24) {
11       setMessage("✗ Please enter a valid 24-character Task ID.");
12       return;
13     }
14
15     try {
16       const res = await axios.delete(`http://localhost:5000/tasks/${taskId}`);
17       if (res.data.success) {
18         const set taskId: (value: React.SetStateAction<string>) => void
19         set taskId("");
20       } else {
21         setMessage("✗ Task not found or couldn't be deleted.");
22       }
23     } catch (err) {
24       setMessage("✗ Error deleting task. Please try again.");
25       console.error(err);
26     }
27   };
28
29  return (
30    <div className="form-container">
31      <h2 className="form-title">Delete Task</h2>
32      <input
33        type="text"
34        placeholder="Enter Task ID (24 chars)"
35        value={taskId}
36        onChange={(e) => set taskId(e.target.value)}
37        className="form-input"
38      />
39      <button onClick={handleDelete} className="form-button delete">
40        Delete Task
41      </button>
42      {message && <p className="form-message">{message}</p>}
43    </div>
44  );
45
46  export default DeleteTask;
47
48
```

## Form style:

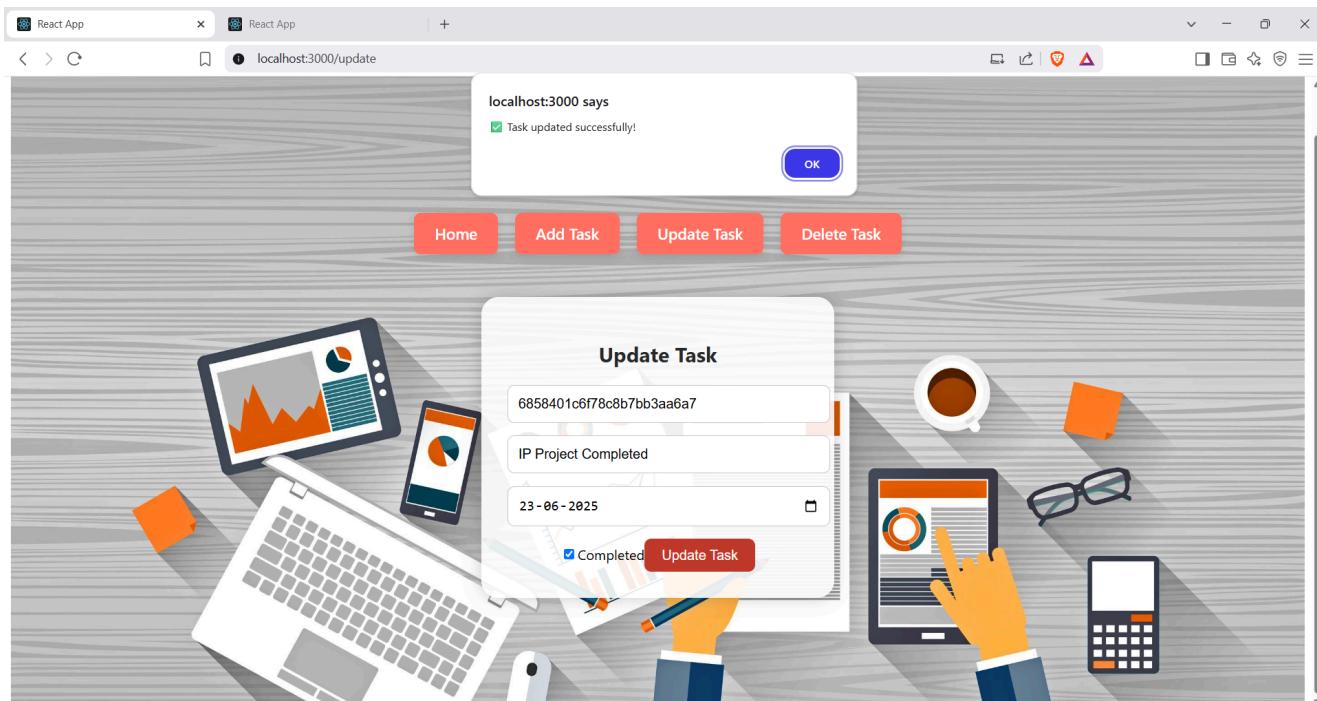
```
# FormStyles.css X  
task-manager-frontend > src > Pages > # FormStyles.css > ...  
1  .form-container {  
2    background: ■rgba(255, 255, 255, 0.9);  
3    padding: 30px;  
4    margin: 50px auto;  
5    border-radius: 20px;  
6    width: 350px;  
7    text-align: center;  
8    box-shadow: 0 0 20px □rgba(0,0,0,0.2);  
9  }  
10  
11  .form-title {  
12    font-size: 24px;  
13    margin-bottom: 20px;  
14    font-weight: bold;  
15    color: □#2b2b2b;  
16  }  
17  
18  .form-input {  
19    width: 100%;  
20    padding: 12px;  
21    margin-bottom: 15px;  
22    border: 1px solid ■#ccc;  
23    border-radius: 8px;  
24    font-size: 16px;  
25  }  
26  
27  .form-button {  
28    padding: 10px 20px;  
29    border: none;  
30    font-size: 16px;  
31    border-radius: 8px;  
32    cursor: pointer;  
33    background-color: ■#e74c3c;  
34    color: ■white;  
35    transition: background-color 0.3s ease;  
36  }  
37  
38  .form-button:hover {  
39    background-color: ■#c0392b;  
40  }  
41  
42  .form-message {  
43    margin-top: 10px;  
44    color: □#333;  
45  }  
46
```

**Home Page:****Add Task and Deadline:**

## Task Added to Home page Successfully:



## Update Task And Added to home page successfully:



**Delete Task and Added to home page :**