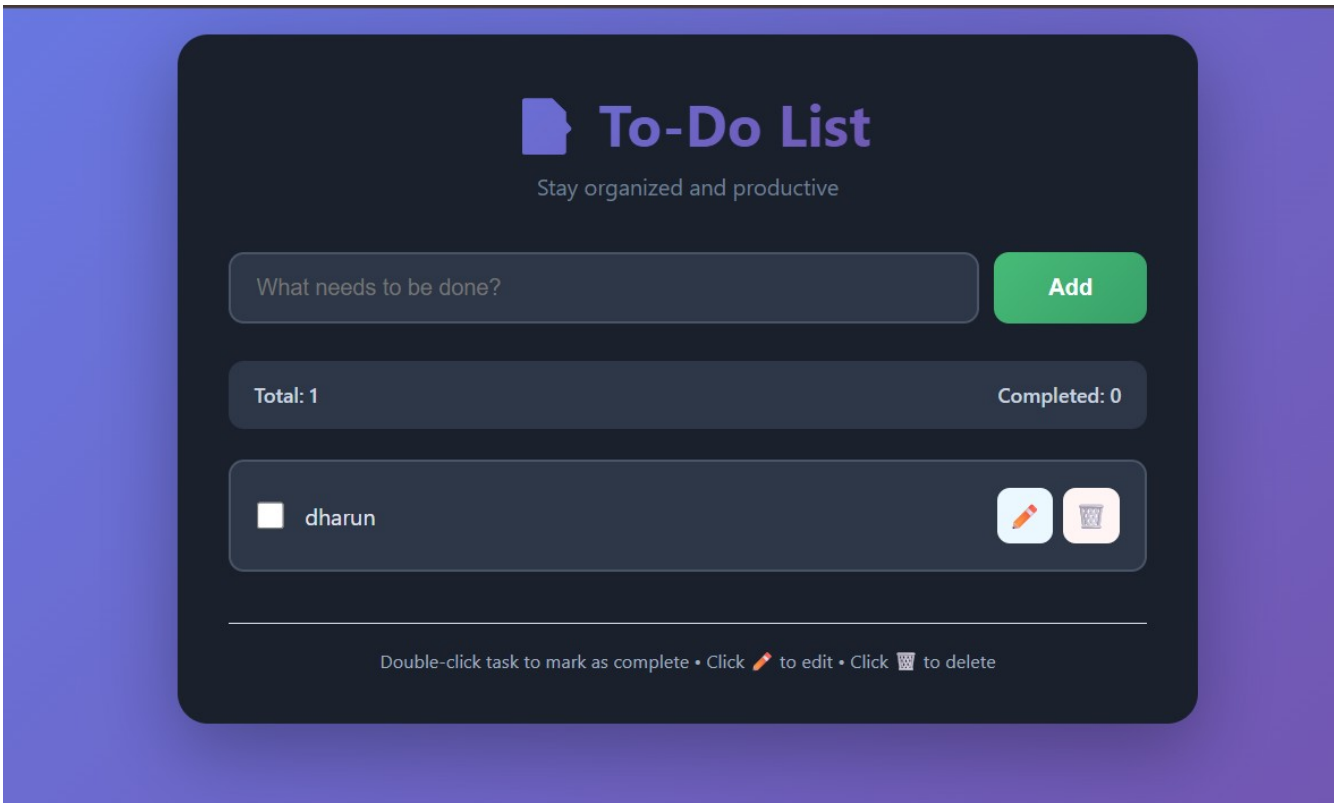
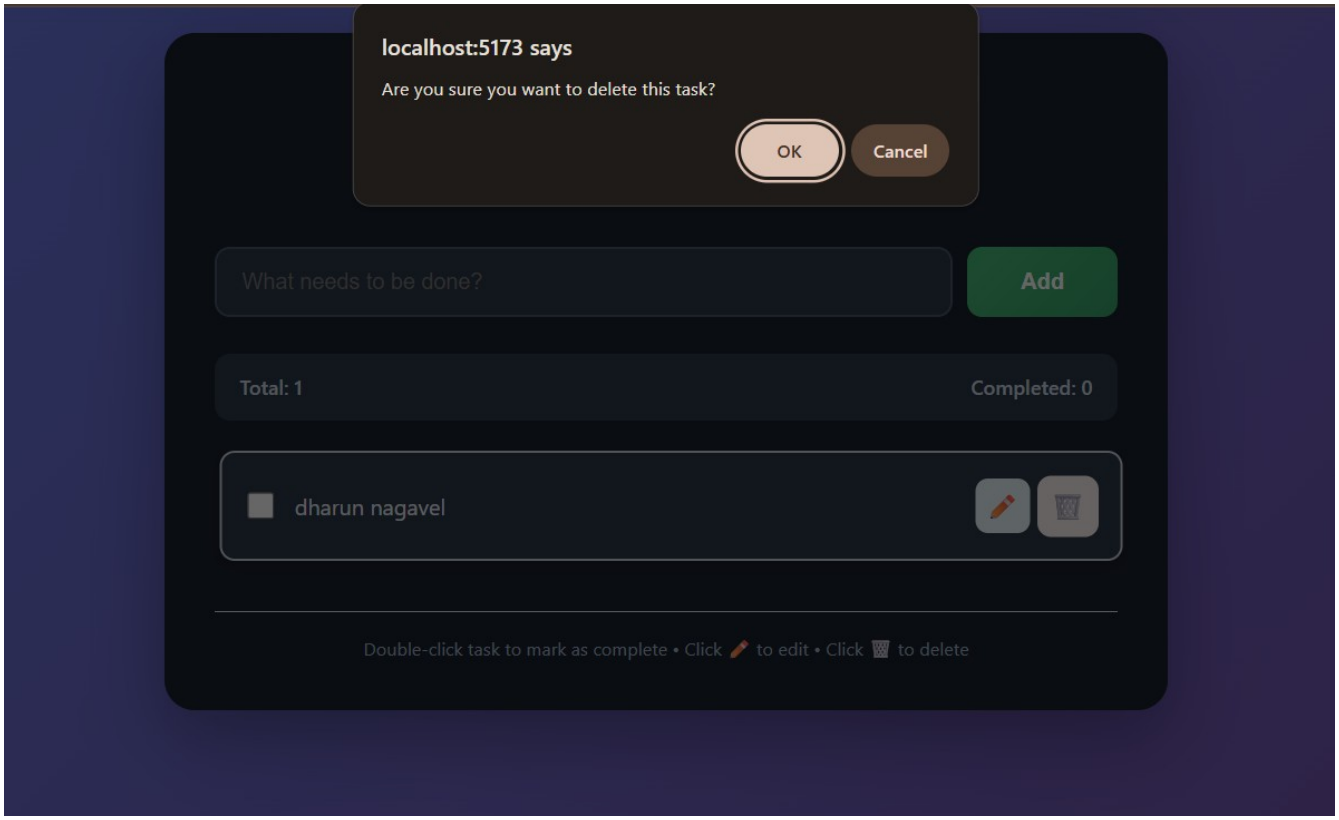
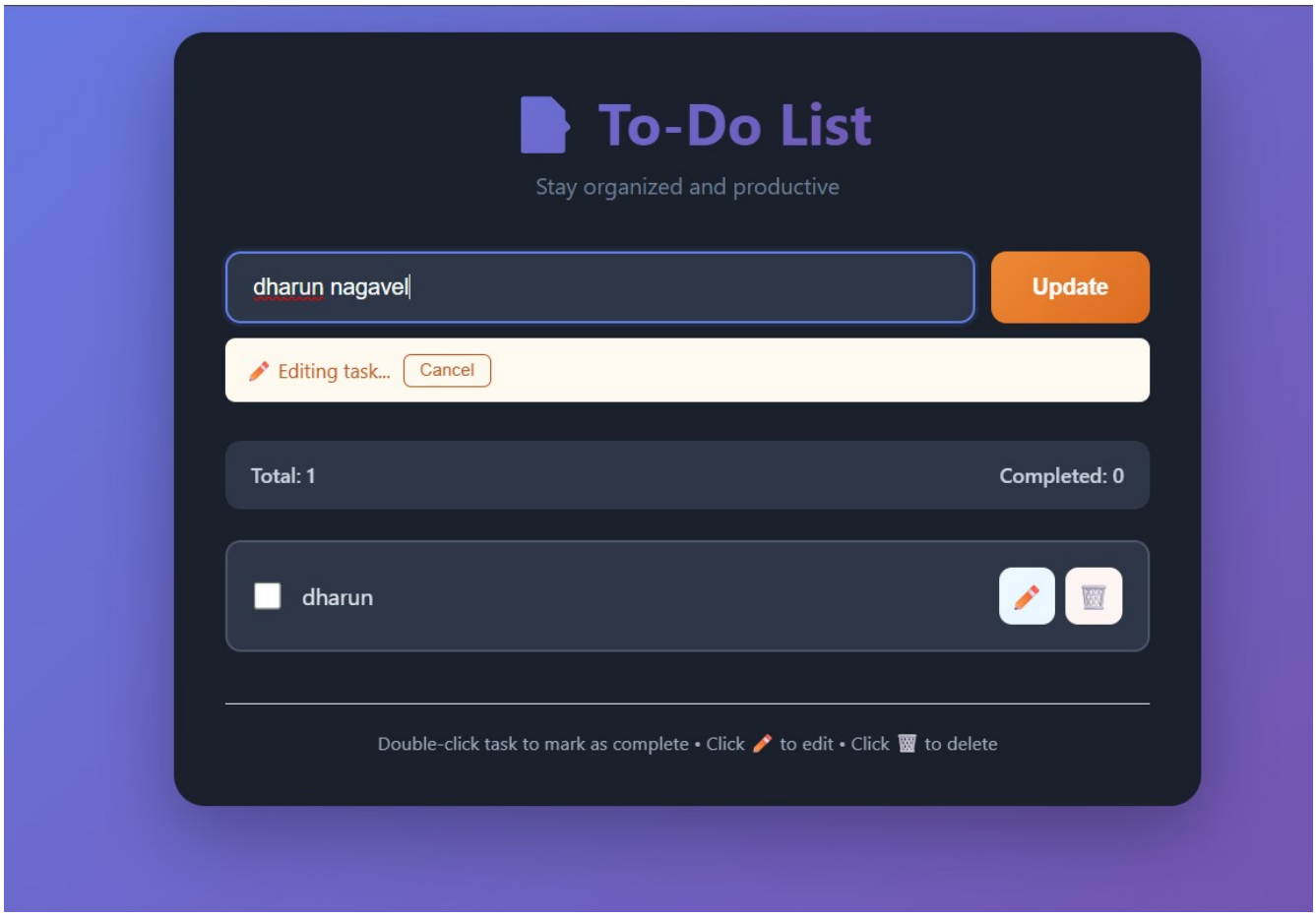

To-Do List
Stay organized and productive
What needs to be done?
Add
Total: 1
Completed: 0
dharun
Double-click task to mark as complete • Click to edit • Click to delete


localhost:5173 says
Are you sure you want to delete this task?
OK
Cancel
What needs to be done?
Add
Total: 1
Completed: 0
dharun nagavel
Double-click task to mark as complete • Click to edit • Click to delete




```
import { useEffect, useState } from "react";
import "./App.css";

function App() {
  const [todos, setTodos] = useState([]);
  const [text, setText] = useState("");
  const [editId, setEditId] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  const API = "http://localhost:5000/todos";

  useEffect(() => {
    setLoading(true);
    fetch(API)
      .then(res => {
        if (!res.ok) throw new Error('Failed to fetch todos');
        return res.json();
      })
      .then(data => {
        setTodos(data);
        setError(null);
      })
      .catch(err => {
        console.error("Error fetching todos:", err);
        setError("Failed to load todos. Please try again.");
      })
      .finally(() => setLoading(false));
  }, []);

  const saveTodo = () => {
    if (!text.trim()) {
```

```
    alert("Please enter a task");
    return;
  }

  if (editId) {

    fetch(`${API}/${editId}`, {
      method: "PUT",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ text, completed: todos.find(t => t.id === editId)?.completed || false })
    })
      .then(res => res.json())
      .then(updatedTodo => {
        setTodos(todos.map(t => t.id === editId ? updatedTodo : t));
        setEditId(null);
        setText("");
      })
      .catch(err => console.error("Error updating todo:", err));
  } else {
    // ADD
    fetch(API, {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ text, completed: false })
    })
      .then(res => res.json())
      .then(todo => {
        setTodos([...todos, todo]);
        setText("");
      })
      .catch(err => console.error("Error adding todo:", err));
  }
};

// Toggle todo completion
const toggleTodo = (id) => {
  const todo = todos.find(t => t.id === id);
  if (!todo) return;

  fetch(`${API}/${id}`, {
    method: "PUT",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ ...todo, completed: !todo.completed })
  })
    .then(res => res.json())
    .then(updatedTodo => {
      setTodos(todos.map(t => t.id === id ? updatedTodo : t));
    })
    .catch(err => console.error("Error toggling todo:", err));
};

// Edit todo
const editTodo = (todo) => {
  setText(todo.text);
  setEditId(todo.id);
  document.querySelector('input').focus();
};

// Delete todo
const deleteTodo = (id) => {
  if (!window.confirm("Are you sure you want to delete this task?")) return;

  fetch(`${API}/${id}`, { method: "DELETE" })
    .then(() => setTodos(todos.filter(t => t.id !== id)))
    .catch(err => console.error("Error deleting todo:", err));
```

```
};

// Handle Enter key press
const handleKeyPress = (e) => {
  if (e.key === 'Enter') saveTodo();
};

// Clear all completed todos
const clearCompleted = () => {
  const completedIds = todos.filter(t => t.completed).map(t => t.id);
  if (completedIds.length === 0) return;

  if (!window.confirm(`Delete ${completedIds.length} completed task(s)?`)) return;

  // Delete all completed todos
  Promise.all(completedIds.map(id =>
    fetch(`${API}/${id}`, { method: "DELETE" })
  ))
    .then(() => setTodos(todos.filter(t => !t.completed)))
    .catch(err => console.error("Error clearing completed:", err));
};

return (
  <div className="app">
    <div className="container">
      <header className="header">
        <h1 className="title">📝 To-Do List</h1>
        <p className="subtitle">Stay organized and productive</p>
      </header>

      <div className="input-section">
        <div className="input-group">
          <input
            value={text}
            onChange={e => setText(e.target.value)}
            onKeyPress={handleKeyPress}
            placeholder="What needs to be done?"
            className="todo-input"
          />
          <button
            onClick={saveTodo}
            className={`save-btn ${editId ? 'update' : 'add'}`}
          >
            {editId ? "Update" : "Add"}
          </button>
        </div>
        {editId && (
          <div className="edit-notice">
            ✏ Editing task... <button onClick={() => { setEditId(null); setText(""); }} className="cancel-btn">Cancel</button>
          </div>
        )}
      </div>

      <div className="stats-bar">
        <span className="total-tasks">Total: {todos.length}</span>
        <span className="completed-tasks">
          Completed: {todos.filter(t => t.completed).length}
        </span>
        {todos.some(t => t.completed) && (
          <button onClick={clearCompleted} className="clear-btn">
            Clear Completed
          </button>
        )}
      </div>
```

```
        {loading ? (
          <div className="loading">Loading todos...</div>
        ) : error ? (
          <div className="error-message">{error}</div>
        ) : todos.length === 0 ? (
          <div className="empty-state">
            <div className="empty-icon">📪</div>
            <h3>No tasks yet</h3>
            <p>Add a task above to get started!</p>
          </div>
        ) : (
          <ul className="todo-list">
            {todos.map(todo => (
              <li
                key={todo.id}
                className={`todo-item ${todo.completed ? 'completed' : ''}`}
              >
                <div className="todo-content">
                  <input
                    type="checkbox"
                    checked={todo.completed}
                    onChange={() => toggleTodo(todo.id)}
                    className="todo-checkbox"
                  />
                  <span
                    className="todo-text"
                    onClick={() => toggleTodo(todo.id)}
                  >
                    {todo.text}
                  </span>
                </div>
                <div className="todo-actions">
                  <button
                    onClick={() => editTodo(todo)}
                    className="action-btn edit-btn"
                    aria-label="Edit"
                  >
                    ✏
                  </button>
                  <button
                    onClick={() => deleteTodo(todo.id)}
                    className="action-btn delete-btn"
                    aria-label="Delete"
                  >
                    🗑
                  </button>
                </div>
              </li>
            ))}
          </ul>
        )}

        <footer className="footer">
          <p>Double-click task to mark as complete • Click ✏ to edit • Click  to delete</p>
        </footer>
      </div>
    </div>
  );
}

export default App;
```