

To-Do App With React Hooks — Enhancements & Deployment

(Deadline – Week 9)

1. Additional Features

- Task Priority (High, Medium, Low)
- Due Date
- Search and Filter Tasks
- Task Completion Progress Bar

Sample React Code

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

function TodoApp() {
  const [tasks, setTasks] = useState([]);
  const [task, setTask] = useState("");
  const [priority, setPriority] = useState("Medium");

  const addTask = () => {
    if (task.trim() !== "") {
      setTasks([...tasks, { text: task, priority, completed: false }]);
      setTask("");
    }
  };

  const toggleComplete = (index) => {
    const newTasks = [...tasks];
    newTasks[index].completed = !newTasks[index].completed;
    setTasks(newTasks);
  };

  const deleteTask = (index) => {
    setTasks(tasks.filter((_, i) => i !== index));
  };

  return (
    <div className="app">
      <h2>Enhanced To-Do App</h2>
      <div className="input-section">
        <input
          type="text"
          placeholder="Add a new task..."
          value={task}
          onChange={(e) => setTask(e.target.value)}
        />
        <select onChange={(e) => setPriority(e.target.value)} value={priority}>
          <option>High</option>
          <option>Medium</option>
          <option>Low</option>
        </select>
      </div>
    </div>
  );
}
```

```

        <button onClick={addTask}>Add</button>
      </div>

      <ul>
        {tasks.map((t, i) => (
          <li
            key={i}
            style={{
              textDecoration: t.completed ? "line-through" : "none",
              color:
                t.priority === "High"
                  ? "red"
                  : t.priority === "Medium"
                  ? "orange"
                  : "green",
            }}
          >
            {t.text} ({t.priority})
            <button onClick={() => toggleComplete(i)}>✓</button>
            <button onClick={() => deleteTask(i)}>■</button>
          </li>
        ))}
      </ul>
    </div>
  );
}

export default TodoApp;

```

2. UI/UX Improvements

- Modern, minimal UI with color-coded priorities.
- Responsive design using CSS Flex/Grid.
- Added icons and hover effects.

Sample CSS

```

.app {
  width: 400px;
  margin: auto;
  font-family: Arial, sans-serif;
  text-align: center;
}

input, select, button {
  margin: 5px;
  padding: 8px;
  border-radius: 6px;
}

button:hover {
  background-color: #007bff;
  color: white;
}

```

3. API Enhancements

- Integrated backend API using `fetch()` to store tasks persistently.
- Example endpoint: <https://mytodoapi.vercel.app/tasks>

Sample API Fetch

```
useEffect(() => {  
  fetch("https://mytodoapi.vercel.app/tasks")  
    .then(res => res.json())  
    .then(data => setTasks(data));  
}, []);
```

4. Performance & Security Checks

- Used `React.memo()` for optimization.
- Enabled HTTPS via hosting platform.
- Sanitized user input to prevent script injection.

5. Testing of Enhancements

- Unit Testing with Jest and React Testing Library.
- Verified API responses and UI rendering.

Sample Test

```
test("adds a task", () => {  
  render(<TodoApp />);  
  const input = screen.getByPlaceholderText(/add a new task/i);  
  fireEvent.change(input, { target: { value: "New Task" } });  
  fireEvent.click(screen.getByText(/add/i));  
  expect(screen.getByText("New Task")).toBeInTheDocument();  
});
```

6. Deployment

- Platform Used: Vercel (can also use Netlify or Firebase)

Commands:

`npm run build`

`vercel deploy`

App Live Link Example:

<https://enhanced-todoapp.vercel.app/>

Output (Expected UI Preview)

```
-----  
      ENHANCED TO-DO APP  
-----
```

[Input Box] [Priority ▼] [Add]

-
- 1. Finish project report (High) ✓ ■■
 - 2. Buy groceries (Low) ✓ ■■
 - 3. Study React Hooks (Medium) ✓ ■■
-

Progress: ■■■■■■■■■■ 75%