



# Product Requirements & Specification Document

---

## Project Name

**TaskFlow - To-Do List Manager**

## Description

TaskFlow is an open-source to-do list manager enabling users to create, edit, complete, and filter tasks. It features a clean, modern UI and is built with React, TypeScript, HTML, CSS, and Tailwind.

---

## 1. Goals & Objectives

Goal	Description
Task Management	Allow users to manage tasks efficiently
Modern UI	Provide a clean, intuitive, and responsive interface
Open Source	Codebase is public and encourages contributions
Entrepreneurial Utility	Suitable for personal and small business productivity

---

## 2. Core Features

Feature	Description
Task Creation	Add new tasks with title and optional description
Task Editing	Edit existing tasks
Task Completion	Mark tasks as complete/incomplete
Task Deletion	Remove tasks
Task Filtering	Filter tasks by status: All, Active, Completed
Persistent State	Store tasks in browser local storage
Responsive Design	UI adapts to mobile and desktop screens

---

## 3. User Stories

As a...	I want to...	So that...
User	Add a new task	I can remember things to do
User	Edit a task	I can update task details
User	Mark a task as complete/incomplete	I can track progress



User	Delete a task	I can remove tasks I no longer need
User	Filter tasks	I can focus on relevant tasks

## 4. Functional Requirements

ID	Requirement
FR1	Users can add tasks with a title (required) and description (optional)
FR2	Users can edit task title and description
FR3	Users can mark tasks as complete/incomplete
FR4	Users can delete tasks
FR5	Users can filter tasks by All, Active, Completed
FR6	Tasks persist in local storage across sessions
FR7	UI is responsive and accessible

## 5. Non-Functional Requirements

ID	Requirement
NFR1	Built with React, TypeScript, Tailwind CSS
NFR2	Codebase is open-source (MIT License)
NFR3	Follows accessibility best practices (WCAG)
NFR4	Loads in under 2 seconds on modern browsers

## 6. Technical Specifications

### 6.1 Architecture

- **Frontend:** React (functional components, hooks), TypeScript
- **Styling:** Tailwind CSS
- **State Management:** React useState/useReducer
- **Persistence:** Browser localStorage

### 6.2 Component Structure

Component	Purpose
App	Root component, manages global state
TaskList	Displays list of tasks
TaskItem	Represents a single task
TaskForm	Handles task creation and editing



FilterBar	Allows filtering by status
Header/Footer	Branding and navigation (optional)

6.3 Data Model

```
type Task = {
  id: string;
  title: string;
  description?: string;
  completed: boolean;
  createdAt: string;
  updatedAt: string;
};
```

7. UI/UX Requirements

Requirement	Description
Clean, modern design	Minimalist, distraction-free interface
Responsive layout	Works on mobile, tablet, and desktop
Accessible interactions	Keyboard navigation, ARIA labels
Visual feedback	Animations for add/edit/delete/complete actions

8. Open Source & Community

Aspect	Specification
License	MIT
Repository	Public on GitHub
Contribution	Clear guidelines in CONTRIBUTING.md

9. Out of Scope

- User authentication
- Task prioritization or deadlines
- Collaboration/multi-user features
- Backend/server-side storage

10. Milestones & Deliverables

Milestone	Deliverable	Timeline
-----------	-------------	----------



MVP UI	Basic UI with add/edit/delete/complete	Week 1
Filtering & Persistence	Filtering and localStorage integration	Week 2
Responsive & Accessible	Mobile support, accessibility improvements	Week 3
Open Source Launch	Documentation, repo setup, public release	Week 4

## 11. Acceptance Criteria

- All core features implemented and tested
- UI is responsive and accessible
- Tasks persist across sessions
- Code is open-source and documented

## 12. Appendix

### Example Task JSON

```
{
  "id": "1",
  "title": "Design TaskFlow UI",
  "description": "Create wireframes and UI components",
  "completed": false,
  "createdAt": "2024-06-01T10:00:00Z",
  "updatedAt": "2024-06-01T10:00:00Z"
}
```

End of Document