**Software Requirements Document**

**Project:** Custom To-Do List App (Frontend JavaScript)  
**Version:** 1.0  
**Prepared by:** [Your Name]

**1. Introduction**

A lightweight, user-friendly to-do list app that helps users manage tasks with local storage persistence. Built with vanilla JavaScript, HTML, and CSS.

**Key Features:**

✔ Add, edit, delete, and mark tasks as complete  
✔ Data saved in localStorage (no server needed)  
✔ Clean UI with visual feedback for completed tasks  
✔ Optional: Search, filter, and priority tagging

**2. Functional Requirements**

**Core Features (Must-Have)**

| **Feature** | **Description** |
| --- | --- |
| **Task Management** | Users can add, edit, delete, and toggle tasks |
| **Data Persistence** | Tasks saved in localStorage (retain after refresh) |
| **Dynamic UI** | Tasks displayed in a list with interactive controls |
| **Completion Styling** | Visual distinction (strikethrough/color) for completed tasks |

**Advanced Features (Nice-to-Have)**

| **Feature** | **Description** |
| --- | --- |
| **Search & Filter** | Real-time task filtering by keyword |
| **Priority Levels** | Tag tasks as High/Medium/Low priority |

**3. Technical Requirements**

**Frontend**

* **Languages:** HTML, CSS, JavaScript (ES6+)
* **Storage:** localStorage for persistence
* **Events:** Custom DOM events for component communication
* **Responsive Design:** Works on mobile, tablet, and desktop

**File Structure**

Copy

Download

/to-do-app/

├── index.html # Main HTML structure

├── styles.css # Styling

├── activityService.js # CRUD + localStorage logic

├── activityList.js # Renders & manages task list

└── (Optional) searchFilter.js # Search functionality

**4. User Interface (UI) Requirements**

**Components**

1. **Task Input Form**
   * Text field + "Add" button
2. **Task List**
   * Each task shows:
     + Title (editable)
     + Checkbox (complete/incomplete)
     + Delete button
   * Strikethrough + color change for completed tasks
3. **(Optional) Search Bar**
   * Filters tasks as user types
4. **(Optional) Priority Tags**
   * Color-coded labels (e.g., red = High priority)

**5. Non-Functional Requirements**

| **Category** | **Requirement** |
| --- | --- |
| **Performance** | Instant UI updates, no lag in task operations |
| **Usability** | Intuitive controls, clear visual feedback |
| **Compatibility** | Works on Chrome, Firefox, Edge, and mobile browsers |
| **Maintainability** | Modular code for easy updates |

**6. Testing Requirements**

**Test Cases**

✅ **CRUD Operations**

* Add, edit, delete, and toggle tasks
* Verify localStorage updates correctly

✅ **UI/UX**

* Check styling for completed tasks
* Ensure responsive design on different screens

✅ **(Optional) Search & Priority**

* Filtering works in real-time
* Priority tags display correctly

**7. Timeline & Deliverables**

| **Phase** | **Tasks** | **Estimated Time** |
| --- | --- | --- |
| **Core Setup** | Data model, CRUD, localStorage | 1-2 days |
| **UI Implementation** | Task list, input form, styling | 1 day |
| **Enhancements** | Search, priority tags (optional) | 1 day |
| **Testing & Polish** | Bug fixes, responsiveness checks | 1 day |

**Next Steps**

* Start with **Phase 1 (Core Features)**
* Build incrementally (UI → Logic → Extras)
* Test frequently!

Need any refinements or additional details? Let me know! 🚀

**Approved by:** [Your Name]  
**Date:** [Today's Date]