# To-Do List Manager Documentation

## Overview

The To-Do List Manager is a web-based task management app that helps users organize their daily activities. It allows users to add, edit, and remove tasks, as well as set priorities and filter tasks. The app uses local storage to persist data across sessions, making it easy for users to manage tasks anytime.

## Features and Functionality

- \*\*Add Tasks:\*\* Users can input tasks into the app and set their priority level (High, Medium, Low).

- \*\*Remove Tasks:\*\* Users can delete tasks that are no longer needed.

- \*\*Edit Tasks:\*\* Tasks can be updated, allowing users to change the task description or priority.

- \*\*Filter Tasks:\*\* Users can filter tasks based on their priority or status (Completed/Pending).

- \*\*Task Persistence:\*\* Tasks are saved locally in the browser's LocalStorage, ensuring data persists across sessions.

- \*\*Responsive UI:\*\* The app is designed to work well across devices, with an interactive interface to add, edit, and manage tasks.

## Key Technologies and Libraries Used

- \*\*HTML:\*\* For creating the basic structure of the app.

- \*\*CSS:\*\* For styling and creating a responsive layout.

- \*\*JavaScript:\*\* For handling the core functionality of the app, including DOM manipulation and local storage management.

- \*\*FontAwesome:\*\* For adding icons to the app interface.

- \*\*LocalStorage:\*\* For storing tasks and ensuring they persist across browser sessions.

## Code Explanation

### Functions and Flow:

1. \*\*Initialization:\*\* The app initializes by loading tasks from the LocalStorage and displaying them in the task list.

2. \*\*Add Task:\*\* When a task is added, the app creates a new task object and stores it in the LocalStorage.

3. \*\*Edit Task:\*\* Users can edit the details of an existing task using a modal that updates both the task details and LocalStorage.

4. \*\*Delete Task:\*\* Tasks can be removed by clicking a delete button next to each task.

5. \*\*Filter Tasks:\*\* The app includes filters for task priority and status, updating the displayed task list based on user selection.

### Code Structure:

- The main functionality is in the `script.js` file, which includes the logic for adding, editing, deleting, and filtering tasks.

- Tasks are stored as objects in an array, and this array is saved in the LocalStorage.

- The HTML file includes sections for displaying the task list, task input fields, and filter options.

- The CSS file is used to style the task list, buttons, and filters.

## How to Use the Application

1. \*\*Clone the Repository:\*\* Download the project files or clone the repository from GitHub.

2. \*\*Open in Browser:\*\* Open the `index.html` file in any modern web browser to run the app.

3. \*\*Start Managing Tasks:\*\* Use the input field to add tasks with a selected priority level. You can edit, delete, or filter tasks as needed.

## Future Enhancements

- \*\*User Authentication:\*\* Implement a login system to allow users to save tasks in the cloud.

- \*\*Task Deadlines:\*\* Add a feature to set deadlines for tasks and send reminders.

- \*\*Task Categories:\*\* Organize tasks into categories for better management.

- \*\*Priority Notifications:\*\* Add notifications when tasks with a high priority are due.

## Conclusion

This To-Do List Manager provides a simple yet effective way to manage daily tasks. With features like task filtering, priority management, and local data storage, it offers an intuitive experience to keep users on top of their tasks.