

Tổng quan

TODO App là ứng dụng quản lý công việc cá nhân cho phép người dùng:

- Thêm, đánh dấu hoàn thành, xóa, khôi phục, và sắp xếp nhiệm vụ.
- Lưu dữ liệu trực tiếp trên **trình duyệt** thông qua **LocalStorage**, không cần backend.

Cấu trúc chính

1. **index.html** – Giao diện người dùng

- Header (navigation: *Tasks* / *Trash*)
- Form thêm công việc: Task content + Deadline
- Danh sách nhiệm vụ (hiển thị động theo route)
- Footer

2. **main.js** – Logic ứng dụng

- **Model layer:**
 - `loadArray()` / `saveArray()` để lưu dữ liệu vào `localStorage`.
 - Dữ liệu gồm `activeTasks` và `trashTasks`.
- **Controller layer:**
 - `addTask()`, `toggleDone()`, `softDelete()`, `restoreTask()`, `hardDelete()`.
 - Dữ liệu luôn được lưu lại sau mỗi thao tác.
- **View layer:**
 - `renderActive()` và `renderTrash()` dựng UI theo route `#/active` hoặc `#/trash`.
 - Task được sắp xếp theo deadline, đổi màu khi quá hạn (Overdue) hoặc khi hoàn thành (Checked).

Chức năng chi tiết

Chức năng	Mô tả
Add Task	Nhập tiêu đề + deadline, nhấn Add để lưu task mới.
Toggle Done	Nhấn vào nút trạng thái để chuyển giữa <i>Pending</i> ↔ <i>Checked</i> ↔ <i>Overdue</i> .
Soft Delete	Gỡ khỏi danh sách Active, chuyển sang Trash.
Restore Task	Trong Trash, nhấn Restore để đưa về Active.
Hard Delete	Xóa vĩnh viễn khỏi Trash (có confirm).
Overdue Detection	Task bị quá hạn hiển thị nền đỏ và chữ trắng.
Persistent Storage	Dữ liệu lưu bằng LocalStorage, không mất khi reload.

Luồng hoạt động

1. Người dùng mở trang → app đọc dữ liệu từ localStorage.
2. Mặc định route `#/active` hiển thị danh sách công việc đang hoạt động.
3. Khi người dùng thêm hoặc xóa task → dữ liệu lưu lại → `render()` cập nhật lại UI.
4. Khi chuyển qua `#/trash`, app render lại danh sách rác.