TASK
MANAGEMENT

Stay organized and get things done

Presented By

YASELA DISSANAYAKE

MAIN
FEATURES

Task CRUD   Create, Retrieve, Update and Delete tasks

Status   Statuses: To Do(Default) , In Progress, Done.
management

Filtering   Filter by "Status" or "Dead line"

List & detail   List view with description, status pill, due badge
UX   (overdue highlighted)

USER'S NEED

- The user wants a simple tool to keep track of their tasks.
- They should be able to add new tasks, edit them if details change, and delete them when they're no longer needed.
- Each task needs a status — like "To Do', "In Progress', or "Done' — so the user can see their progress.
- Tasks can also have a due date, and the app should clearly show if a task is overdue.
- The user should be able to filter the task list to only see certain statuses or tasks with certain deadlines, so they can focus on what's urgent or important.
- They should also be able to sort tasks, for example, by due date or by status.
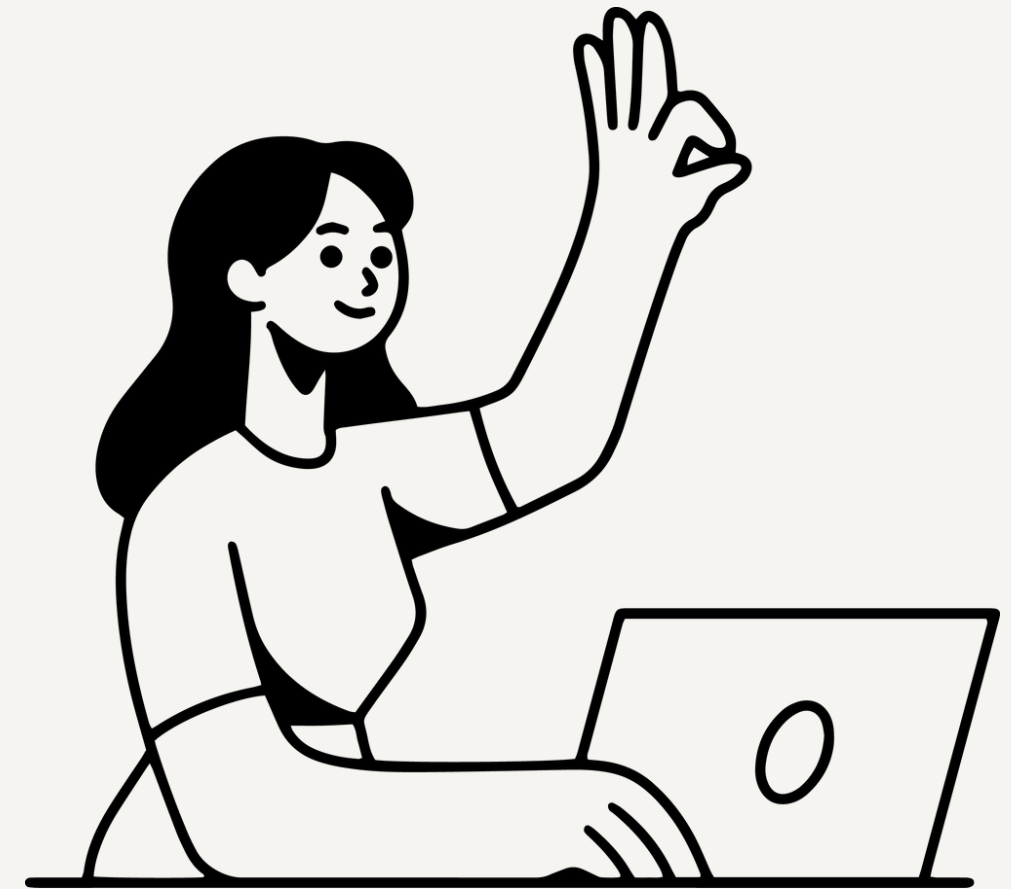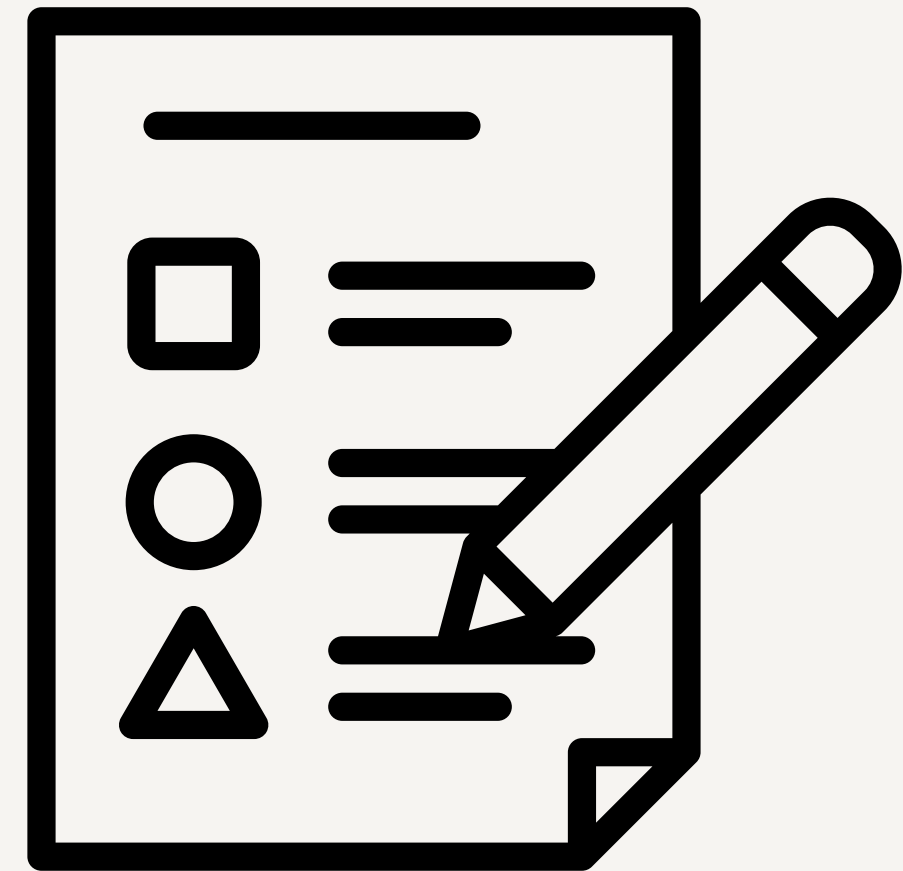-

To do → **In-Progress** → **Done**

VALIDATION &
EDGE CASES

- Empty/whitespace descriptions → disallow.

- Very long descriptions → enforce max length (e.g., 160 chars for the short field).

- Duplicate tasks → allowed, but consider a subtle hint; do not block.

- Invalid dates (e.g., Feb 30) → block with clear error.

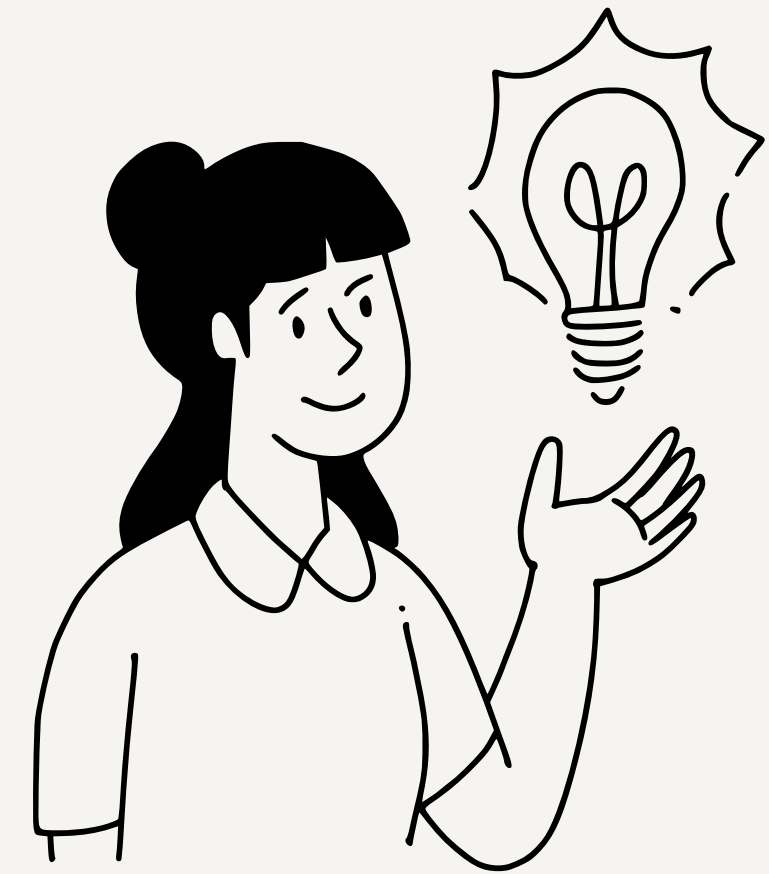- Past due date on create → allowed, but show "Overdue'.

ASSUMPTIONS

- Single user; no login/auth.

- Date-only due dates (no time-of-day).

- Three fixed statuses only; no custom status names.

- No subtasks, attachments, comments, or tags.

- No notifications (email/push).

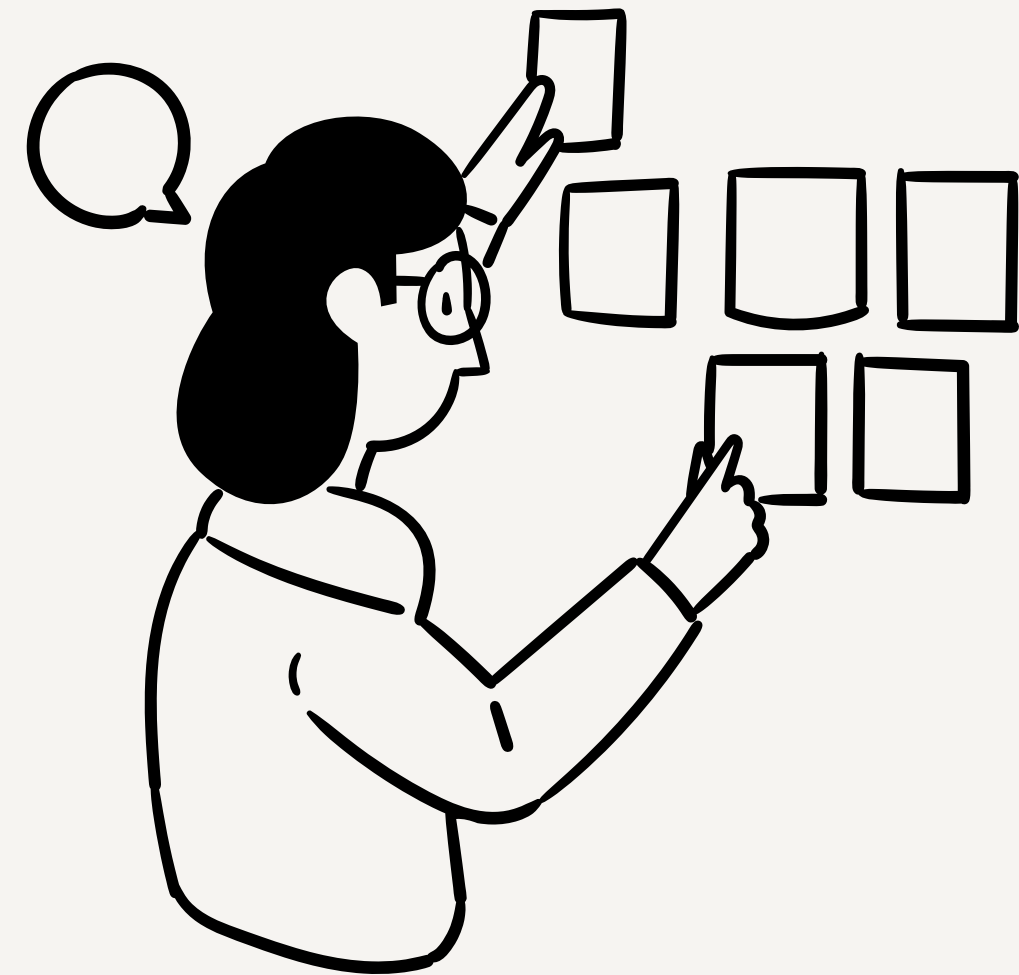- Local-only, simple storage (in-browser local storage).

PLANNED
FEATURES

- Task CRUD (create, edit, delete)

- Status management (To Do / In Progress / Done)

- Due date + overdue indicator

- Filtering by status and deadline buckets (Overdue, Today, Upcoming)

- Sorting (by due date, then status order)

- Responsive UI

- Storage (localStorage)

TASK
BREAKDOWN

- Project setup (Vite/React, TS, Tailwind, ESLint)

- Data model + localStorage service (CRUD)

- Add Task form (description + due date validation)

- Task list UI (rows with status pill + due badge)

- Status interactions (quick toggle / select)

- Overdue logic (labels for Overdue/Today/This Week)

- Filters panel (status multi-select + deadline buckets)

- Sorting

AI
TOOLS

V0.dev (Application Development)

Prompt used: Create a simple React+ Tailwind + Typescript application with the following features,

- I want to create, update, and remove tasks from a list.
- Each task should have a status (e.g., To Do, In Progress, Done), a due date, and a short description.
- I want to filter tasks based on their status and deadlines.
- Do not need to use a backend and a database.
- Use local storage to store the data

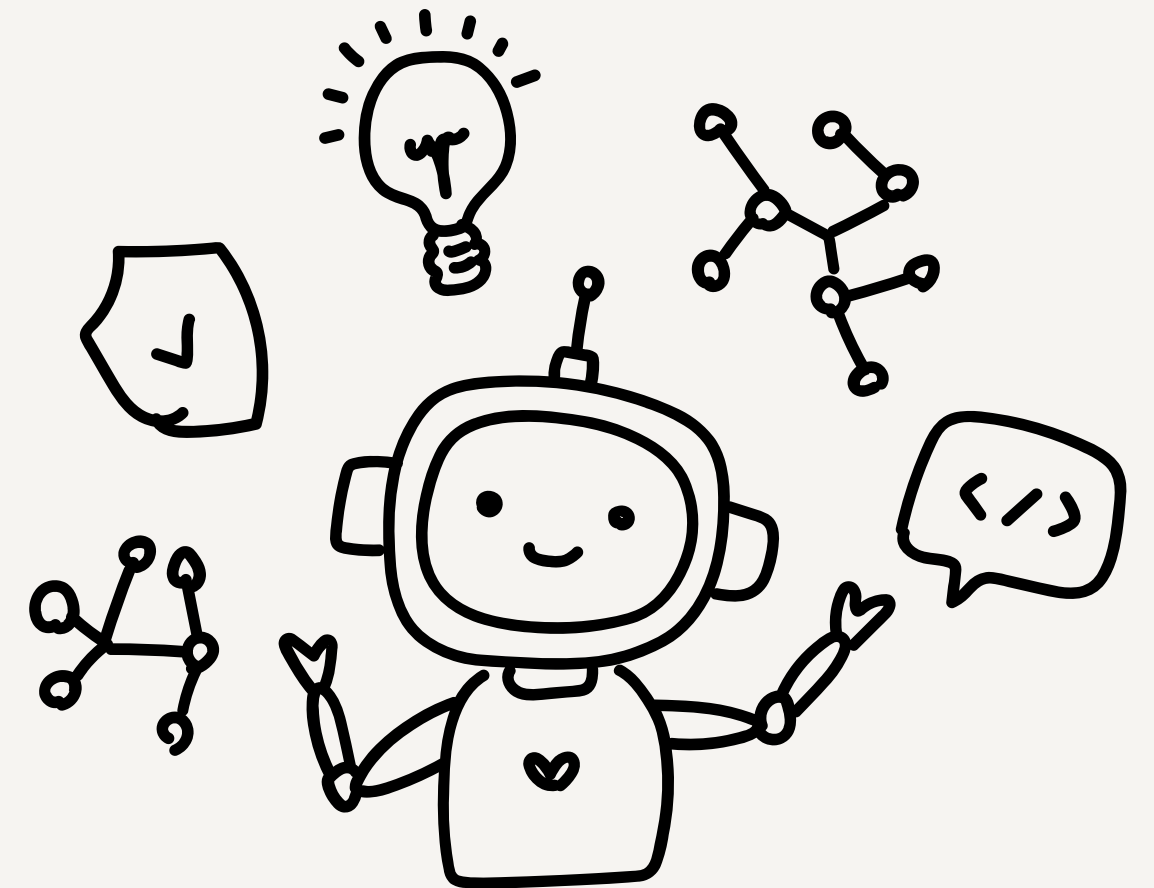**ChatGPT (Documentation)**
DiagramGPT (Architecture Diagram)

Figure 1

LEGEND

- UI component
- React state
- Persistence
- Data flow
- UI control

COMPONENTS ARCHITECTURE

- DashboardStats
- TaskFilters
- TASKLIST
  - TaskCard (xN)
- TaskForm
- app/page.tsx (root)

Counts, progress, upcoming due

Status/date/search/sort

set filters

Filtered list

Edit, delete, status

modal/edit flags

Create, edit, status

controlled inputs

DATA FLOW & LOGIC

- localStorage
  - load
  - auto-save on change
- Filter State
- Tasks State
- UI State
- Form State

metrics

update/delete

filtered list

filtered list

create/update

modal/edit flags