Help me develop a minimal, fast-to-use task management tool for handling many small tasks in a fast-paced environment, with easy maintenance and strong tracking for weekly or monthly reporting. I need a web-based tool for now, but may want to create a desktop version at a later time.

The primary view of the tool needs to be clean and compact, with a list of tasks (one on each row), a status icon (more details below), a priority flag, and a due date that may be blank very often. Simply adding new task by Clicking the end of the list and starting typing task name. The tasks can be opened in a new window with a double click to manually edit all the features, but it should be a rare action. Most of the activities like creating task, updating status, set due dates etc. should be able to be done in the primary list view. And features like task starting, completion date etc. need to be recorded automatically to save maintaining effort. The tasks could be categorized into different subjects/areas (help me name this level), and can have sub-tasks. The higher level category don’t need the features like status, dates. It’s only to help organize. When hover over a task, a text bubble will pop out to keep more detailed notes of this task.

Task status would be To Do when a new task is created. A status button with different color/icon to show and also change the status. One click of the button when it’s To Do, the status will change to In-Progress and record a start date. Then another click will change to Pause and record a pause date. Click again change back to In-progress and record a resume date. It can keep switching between Paus and In-Progress with one click. And with a double click on any status, it changes to Done and record a completion date. Long-press of the button opens a drop down menu of all the status to select manually. The dates mentioned should be recorded as well with the manual change of status. There many be multiple pause date and resume date. Help me design a data structure that would record these dates effectively. There would also be a last modified date that records any kind of activity except renaming. status sorting order : in-progress -> to do -> paused -> done.

I really want to have the pause status for tasks that doesn’t need any effort for now but need to circle back on a later time. Help me figure out how to design the date features so it reflects that during the pause period these tasks were not worked on. But if the pause related functions are too complicated, we can focus on the others as a starting point and add Pause status back later.

The sub-task status would be nested to parent task status, meaning when any sub-task is in-progress, the parent task status would automatically change to in-progress. When all the sub-tasks are paused, the parent task would automatically change to paused. But when all the sub-tasks are done, the parent task will not change accordingly and would need user to confirm the task is done. Ideally pop up a question asking if the task is done now that all the sub-tasks are done. But this can be a later enhancement if not easy to do.

Priority is Normal by default. Use red, yellow, green, gray flag icons to show Priorities Urgent, High, Normal, Low. When a task is due in the next business day, automatically set the priority to be Urgent.

On another side, the tool needs to be able to use as a work log that can easily track what has been done already. The dates recorded above are used to sort or filter tasks, like in a sql table. It would be very helpful for weekly team update etc. So create two view tabs by default, one view as planner that hide completed tasks by default (ideally have the ability to unhide). Sort the tasks first by status, then by priority, the rest by last modified date. The other view as tracker that shows the tasks that have been worked on in last x days (x=7 by default but user can change). Show completed tasks by default, and sort the tasks by area, status (reserve order), last modified date (descending order).

user should be able to sort or filter in any view flexibly. The sort or filter can combine multiple levels, for example sort by status in ascending order then sort by the last modified date. And filters can also combine multiple features with a mix of AND/OR/NOT.

all the task info is stored in a sql-like table on the back end. And user could pull that table out if needed.

Ideally user can create simple summaries like total number of tasks done by week/monthly, or total/average number of days spend on each task, etc. But this is not needed for now.

Your requirements have been translated into a modern, maintainable, and fast-to-use web-based task management tool design. Here’s a summary of what’s been implemented and how the system is structured:

**1. Data Structure & Backend**

* **Areas** (your “subjects/areas”): Used for high-level organization. No status/dates, just a name.
* **Tasks**: Each task has a title, description, area, parent (for sub-tasks), status, priority, due date, and automatic timestamps (created, last modified, start, completion).
* **Task History**: Every status change (including pause/resume) is logged with a timestamp and optional note. This allows for accurate reporting and tracking of work periods, including multiple pause/resume cycles.
* **Status Flow**: To Do → In Progress ↔ Paused → Done (with all transitions and dates tracked).
* **Priority**: Urgent (red), High (yellow), Normal (green), Low (gray). Due-tomorrow tasks are auto-marked Urgent.
* **Parent/Sub-task Logic**: Parent status auto-updates based on sub-tasks (in-progress/paused), but not to Done (user confirmation required).
* **SQL Database**: All data is stored in SQLite, easily exportable for reporting.

**2. Frontend (React + TypeScript + Tailwind CSS)**

* **Primary View**: Clean, compact list. Each row: status button, priority flag, task name, due date.
* **Quick Add**: Click at the end of the list and start typing to add a new task.
* **Status Button**:
* Click: To Do → In Progress → Paused → In Progress (cycles)
* Double-click: Mark as Done
* Long-press: Dropdown for manual status selection (future enhancement)
* **Dates**: Start, pause, resume, and completion dates are all recorded automatically.
* **Sub-tasks**: Nested under parent, with status logic as described.
* **Hover**: Shows a tooltip with task notes/description.
* **Edit Modal**: Double-click a task to open a modal for full editing (rare action).
* **Views**:
* **Planner**: Hides completed tasks by default, sorts by status > priority > last modified.
* **Tracker**: Shows tasks worked on in the last X days (default 7), including completed, sorted by area > status (reverse) > last modified.
* **Flexible Sorting/Filtering**: UI supports multi-level sort/filter (future: AND/OR/NOT logic).
* **Export**: All task data can be exported as JSON.

**3. How Pause/Resume is Tracked**

* Every time a task is paused or resumed, a record is added to the task\_history table with the timestamp and status.
* For reporting, you can calculate “active work time” by subtracting all paused intervals from the total time between start and completion.

**4. Naming for High-Level Category**

* “Area” is used in the codebase (can be changed to “Project”, “Subject”, etc. if you prefer).

**5. Setup & Next Steps**

* All code and configuration files are in place.
* To run locally:

1. npm run install-all (from the root) to install all dependencies.
2. npm run dev to start both backend and frontend.
3. Open http://localhost:3000 in your browser.

**If you want to add the Pause/Resume logic for reporting or need a desktop version (Electron), let me know!**