# Task Creation

From where to access Task Creation –

1. Dashboard – The dashboard must have a button / a tile from where the person can create a task. There will be different buttons for different type of tasks to be created, with different colors but of same, symmetric sizes.
2. Calendar View – Another view from where, if user clicks the task can be created for the selected date.
3. Task View / All Task – Task creation button will have “Create Task” written with a small drop down integrated with the button. The user clicks on create task; it creates a simple task (A simple task can be changed to any type of other task other than sub-task before it is saved from the create task drawer). If the drop-down arrow is clicked, it opens options of creating Task, Recurring Task, Approval Task, Milestone task etc.
4. Task (in the task view grid) – A Sub-Task can be created from the task itself. The task, if have sub tasks in it, it will show a different icon . Such task row if clicked will open the drill down of other rows (sub-tasks). Which can be clicked to open the sub-task details. Which can then be edited / updated and other operations can be done.

### 🔐 **Permissions Matrix (Role-Based)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Role | View | Edit | Reassign | Delete | Add Comments | Add Files | Change Status |
| Creator | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| Assignee | ✅ | ✳️ (own fields) | ✅ (if allowed) | ❌ | ✅ | ✅ | ✅ |
| Collaborator | ✅ | ❌ | ❌ | ❌ | ✅ | ✅ | ❌ |
| Admin | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |

✳️ Assignee / The person the task is assigned to can work on a task but can not edit key fields like – Due Date, Snooze, priority and visibility etc.

Example, If a manager creates a task and assign it to it’s subordinate, he should work on that task without manipulating the due date, Snooze settings, priority and visibility of the task. In case of such a scenario where editing these is required, it will be done by the manager himself.

# ✅ ALL TASK - TASK VIEW PAGE (Unified Full-Page View for a Task/Sub-task)

### 🎯 **Purpose**

A centralized workspace to **view, manage, and act** on a task/sub-task.

### 🧩 **Sections & Functionality – Task Data Grid**

|  |  |
| --- | --- |
| Section | Description |
| **Header Bar** | Task Title, Status (dropdown), Priority, Assignee, Task Type (Normal/Milestone/Approval), Tags, color coding etc. |
| **Quick Actions – from the row quick links (three dots)** | 🟢 Create Subtask, ❌ Delete, ⏸️ Snooze, 🧠 Mark Risk, ✅ Mark Done etc. |

Further once the user is in the task, all the options, action items are displayed in the best possible way to support the user friendliness.

|  |  |
| --- | --- |
| Section | Description |
| Core Info Panel | Description, Start/Due Date, Time Estimate, Parent/Linked Tasks, Creator, Reminders |
| Subtasks Tab | Expandable list, inline status change, filterable |
| Milestone/Approval Info | Visual cues (color/icon) if milestone condition is met or pending approval logic |
| Activity Feed | Real-time chronological log of status changes, comments, edits, reassignment, reminders, etc. |
| Comments | Threaded, mentions, tags, reactions, attachments |

### 🔹 **Main Task Creation Modal (Sliding Drawer from the right side)**

This is the primary popup/modal shown when a user clicks **“Create Task”**.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Type | Required | Behavior & Description |
| **Task Name** | Text (Single line, max 20 characters) | ✅ Yes | Short, clear title. Editable directly in-place. Character limit shown live. |
| **Description** | Multiline Rich Text (Rich) | ❌ No | Add details, context, or notes. Supports formatting: bullet points, bold text, etc. |
| **Assigned To** | Searchable Dropdown (Single Select) | ✅ Yes | Default = Self. User can select another if desired. For solo users, field is fixed. |
| **Priority** | Dropdown (Low, Medium, High, Critical) | ✅ Yes | Default = Low. Selecting a priority dynamically sets Due Date based on system rules. |
| **Due Date** | Date Picker (Auto/Manual) | ✅ Yes\* | Auto-filled based on Priority unless user chooses a different date. User can select the due date by the help of a calendar control. |
| **Visibility** | Toggle (Private, Public) | ✅ Yes | Default = Private. For single users, only Private is shown. For organizations, both shown. |
| **Labels / Tags** | Multiselect (Chips) | ❌ No | Helps categorize or filter tasks. User can create or choose from existing tags.  User can enter small tags, labels here – Example #TasksetuProject, #Finance Team |
| **Attachments** | File Upload (multi-file) | ❌ No | Attach any file up to 5 MB all inclusive. All standard document types supported. (Post MVP – Can have a admin panel to decide the file limit) |

🟨 At the bottom / or somewhere else of this modal, there is a **[More Options]** button / icon. Clicking this opens the Advanced Fields modal.

### 🔹 **"More Options" Modal (Advanced Fields)**

These are less frequently used fields but important for teams working in structured environments.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Type | Required | Behavior & Description |
| **Reference Process** | Dropdown (Searchable) | ❌ No | Link this task to a predefined process (like a standard operating procedure).   * Reference to be done from other task/process already saved as template (include one task and sub-tasks) which includes the custom forms attached to the process. (Published process) |
| **Custom Form** | Dropdown (Searchable) | ❌ No | Select a form template to collect specific data with the task. Cannot create forms here.   * If user want to have reference from another form which is made from custom form module. * User can connect multiple custom form (one for one task/sub-task) and create it’s own process. |
| **Dependencies** | Multiselect Dropdown | ❌ No | Choose existing tasks that must be completed before this one starts.   * To help set a sequence to the task. * Example – If a task is created (Onboard Employee) then its dependency can be set as another task ex – Employee finalization. * Having this set, this task can not be completed till the dependent task is completed. |
| **Task Type** | Dropdown (Simple, Recurring, Approval) | ✅ Yes | Default = Simple. Defines the kind of task being created.  Other Options – Recurring, Approval, Milestone. The options, fields will toggle based on what is selected. |

## 🔁 **Dynamic Behaviour (System-Controlled)**

### 🎯 **Priority-to-Due-Date Logic (Managed by Backend Table – both priority & priority vs due date configuration)**

Instead of hardcoding the number of days between creation and default due date, this is **stored in a master table**, allowing admins to later adjust the logic without code changes.

|  |  |
| --- | --- |
| Priority | Default Days to Due Date (T+x) |
| Low | T + 30 days – Show Due Date |
| Medium | T + 14 days – Show Due Date |
| High | T + 7 days – Show Due Date |
| Critical | T + 2 days – Show Due Date |

* T = Today's date (date of task creation)
* If the user selects a Due Date manually, it **overrides** the default logic and put the date as selected by the user.

### 🧠 **User Experience Design Considerations**

|  |  |
| --- | --- |
| UX Feature | Description |
| **Inline Task Name Edit** | Like Asana — user clicks on task name and directly edits it. Auto-saves when clicked outside. |
| **Live Feedback & Validation** | Real-time alerts if required fields are missed or input formats are invalid. |
| **Searchable Drop-downs** | All user-selection fields are searchable with live filtering. |
| **Default Autofill** | Smart pre-fills (e.g., Priority = Low, Due Date = +30 days) to save time. |
| **Context-Aware Visibility** | Only relevant fields are shown depending on whether the user is in an organization or a single-user environment. |

### 🧪 **Example User Flow (Task Creation)**

**Scenario: Solo User creates a task**

1. Clicks **"Create Task"**
2. Enters **Task Name: “Follow up on client feedback”**
3. **Assigned To:** Auto-selected as self
4. Leaves **Priority** as “Low” → Due Date set to **T + 30 days**
5. Leaves **Visibility** as “Private”
6. Clicks **Save**

**Scenario: Org User creates a structured task**

1. Clicks **"Create Task"**
2. Enters **Task Name**, selects another user in **Assigned To**
3. Sets **Priority = High** → Due Date defaults to **T + 7 days**
4. Opens **More Options**
5. Links to **Reference Process: “Onboarding”**
6. Attaches **Form Template: “New Hire Details”**
7. Selects **Dependency: “HR Interview” task**
8. Clicks **Save**

## **Create Sub-Task**

### 🔍 Overview: What is a Sub-task?

A **Sub-task** is a smaller, actionable unit of work under a parent task.  
It inherits some context but has its own fields like assignee, due date, and status. Sub-tasks **do not have sub-sub-tasks** in the MVP — we will restrict to **one-level nesting** for simplicity.

Also, A sub-task can only be created under a Simple Task and not for approval task, milestone task & recurring task. This means, A simple task can be parent to sub-task but Approval, Milestone & Recurring tasks can not be parents to any sub-task.

Sub-Task can only be a simple task.

### 🧭 Where & How Sub-tasks Are Accessed

#### A. **From Task View Screen** (Primary)

* A collapsible "Sub-tasks" section appears at the bottom.
* Shows all the sub-tasks which are created already for the Task.
* Shows a summarized list:
  + Sub-task name
  + Due date
  + Assignee avatar
* Button: + Add Sub-task
* Sub-Tasks can be opened one by one. When clicked on any sub-task, a section appears from below and open sub-task details, hiding the already opened sub-task (if opened already)

#### B. **From Task List View**

* Parent task shows a small icon or count (3 Sub-tasks)
* Clicking it opens the task view with the sub-task section focused.

#### C. **From Quick Action Menu (3-dot / right-click menu)**

* Options:
  + Add Sub-task
  + View Sub-tasks

### 📋 Fields in Sub-task Creation (from Modal or Inline Add)

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Mandatory (M/O) | Default | Behavior |
| Sub-task Name | M | New Sub-Task | Short name, max 60 chars. Appears inline, editable on click. |
| Assigned To | M | Self | Searchable dropdown (single select). Self-assigned by default. |
| Due Date | M | Based on priority | Pre-filled to parent due date unless priority overrides. |
| Priority | O | Low | Low, Medium, High, Critical. Changes due date if not set manually. |
| Status | M | To Do | To Do, In Progress, Blocked, Completed. Visual cue in list. |
| Visibility | M | Inherit from Parent | Can override if company user. |
| Notes/Description | O | – | Rich text editor for internal use. |
| Attachments | O | – | Upload files or drag/drop. |

### ✏️ Sub-task Creation UX – Flow

#### **Step 1**: User opens a Task (Task View screen)

* Sees Sub-tasks section
* Clicks on + Add Sub-task

#### **Step 2**: Sub-task Input Modal or Inline Add opens

* User enters Sub-task Name
* Assigns user (search dropdown)
* Selects priority (optional)
* Due date auto-suggested based on priority and today's date (from backend config)
* Clicks Create

#### **Step 3**: Sub-task appears under parent

* Collapsed by default (expandable)
* Assigned person gets notified
* If a collaborator was added in the parent, they also get update (if notification rules apply)

### 🔁 Inheritance Logic

|  |  |  |
| --- | --- | --- |
| Field | Inherits from Parent | Can Override? |
| Visibility | Yes | Yes (for company users) |
| Priority | No | Yes |
| Due Date | Suggested from Parent | Yes |
| Assignee | No | Yes |

### 🔐 Permissions

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Self (Individual) | Company User | Company Admin |
| Create Sub-task | ✅ | ✅ | ✅ |
| Assign to others | ❌ | ✅ (team only) | ✅ |
| View sub-tasks | ✅ | ✅ | ✅ |
| Edit Sub-task | ✅ (if assigned) | ✅ (if assigned/collaborator) | ✅ (always) |
| Delete Sub-task | ✅ (if created) | ✅ (if created) | ✅ |

### 🧠 User Friendliness Considerations

* **Inline Quick Add:** Mirrors Asana / Notion style – task appears instantly as you type.
* **Click-to-edit fields:** Task name, due date, and priority can be edited inline on hover.
* **Visual Indicators:** Overdue sub-tasks show in red, completed are struck-through.
* **Keyboard Shortcuts:** Allow Enter to add sub-task, Tab to move to next field.

## **Recurring Task**

### 🔹 Purpose:

Allow users to create tasks that repeat automatically based on defined recurrence rules — reducing manual effort for repetitive workflows like daily check-ins, weekly reports, monthly reviews, etc.

#### 🧩 Where Recurring Logic Appears

##### 📍 Task Creation Modal

* Section: **“Make this a recurring task”**
  + Checkbox: “Repeat this task”
    - When checked, recurrence options appear

#### 📋 Recurrence Options (Modal or Inline Expandable Section)

|  |  |  |
| --- | --- | --- |
| Field | Type | Behavior/Options |
| Frequency | Dropdown | Daily, Weekly, Monthly, Custom |
| Repeat Every | Input | Numeric (e.g. every 2 days, every 3 weeks) |
| Repeat On (for weekly) | Checkboxes | Mon, Tue, Wed... |
| Start Date | Date | Defaults to today |
| End Condition | Radio | • Never Ends • Ends after N occurrences • Ends by specific date |
| Time of Creation | Optional | If defined, system will use this to auto-create task on that time (else default) |

#### 🤖 System Behaviour

##### 🔄 How It Works

* System **generates a new task** based on the recurrence pattern.
* Each recurrence is treated as **a new standalone task**.
* The new task **copies fields** like: name, description, assignee, priority, tags, etc.
* It auto-updates the **due date** based on pattern.

#### 🔐 Permissions & Editing

|  |  |
| --- | --- |
| User Type | Rights |
| Task Creator | Can create, edit, or stop recurrence |
| Assignee | Cannot change recurrence, but can edit individual task instance |
| Admin | Can view all recurrences across company (if needed for reports or corrections) |

#### 🧠 System Data Model (Backend)

##### RecurringTaskMaster Table

|  |  |
| --- | --- |
| Field Name | Description |
| id | Unique ID |
| creator\_user\_id | Who created this pattern |
| base\_task\_id | The task from which recurrence started |
| frequency | Daily / Weekly / Monthly / Custom |
| repeat\_every | Interval number |
| repeat\_on\_days | If weekly, which days (comma list) |
| start\_date | From when recurrence begins |
| end\_condition\_type | Never / AfterN / EndDate |
| end\_value | e.g. 5 (occurrences) or 2025-12-31 |
| active | Boolean |

#### 🔔 Notifications

* When a new instance is created via recurrence logic:
  + The assignee receives a normal **task creation notification**.
  + If collaborators were defined in base task, they're notified too.

#### 🖼️ UX Considerations

|  |  |
| --- | --- |
| Area | Behavior |
| Calendar View | Only upcoming instance(s) visible, not all future repeats |
| Task List | Shows each instance as a separate task with recurrence icon (🔁) |
| Task Detail View | Shows “Recurring from Task #1234” if it’s part of a recurring pattern |
| Manage Recurring Tasks | New tab under "My Tasks" or "Company Tasks" to list and manage patterns |
| Editing a Recurring Task | Only the **master** pattern can be edited for future instances |
| Stopping Recurrence | Button: "Stop recurrence" – removes only future generation, not past |

#### ⚠️ Edge Cases & Fail Safes

|  |  |
| --- | --- |
| Case | Handling |
| User deletes master task | Prompt: “This will stop recurrence. Are you sure?” |
| Company disables recurrence | Existing ones remain, but no new ones can be created |
| Assignee changed mid-recurring | Only affects future instances post-change |
| Weekend creation logic (optional) | Provide toggle: “Skip weekends” or “Adjust to next business day” |

#### ✅ Example Use Cases

|  |  |
| --- | --- |
| Use Case | How It Works |
| Weekly Team Sync Task | Repeats every Monday at 10:00 AM with same assignees & forms |
| Monthly Vendor Invoice Audit | Repeats on 1st of every month, ends after 12 cycles |
| Daily Stand-up | Task created every weekday, due same day, auto-closed if not updated |
| Annual Self-Appraisal Form | Recurs every year with a form attached, sent to employee email |

### 🎯 **If Task Type is selected as “Recurring”**

* Due Date should become blank and un-editable.
* Somewhere near task type, a pop-up/ modal/card should open.
* Showing Calendar – multi select dates.
* Drop Down showing Dates “Every – DDL” of month.
* Drop Down showing days “Every – DDL” of week.
* Only one of the above three should be selectable.
* System captures this information which will be used further in the system.

|  |  |
| --- | --- |
| Priority | Default Days to Due Date (T+x) |
| Low | T + 30 days |
| Medium | T + 14 days |
| High | T + 7 days |
| Critical | T + 2 days |

* T = Today's date (date of task creation)
* If the user selects a Due Date manually, it **overrides** the default logic.

## **Approval Task**

### 🧩 **What Is an Approval Task?**

An **Approval Task** is a special task type where the **primary goal is to obtain a decision** (approval or rejection) from designated approver(s). It also supports **auto-approval logic** if no action is taken within a timeline.

### 🔧 **Creation Flow & Fields**

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Default / Behavior |
| **Task Name** | Short text | Mandatory |
| **Approval Task?** | Toggle/Checkbox | Mandatory toggle |
| **Approver** | Dropdown | Default = Creator, but can be changed |
| **Approval Mode** | Dropdown | Options: “Any One”, “All Must Approve”, “Sequential” |
| **Due Date** | Date | Mandatory |
| **Auto-Approve After** | Numeric Input | Optional (Disabled by default). Format: Due Date + X Days |
| **Description** | Long Text | Optional (for justification, background, etc.) |
| **Attachments** | File Upload | Optional. Users can upload documents or evidence for approval |
| **Collaborators** | Dropdown | Optional. For notification only |
| **Visibility** | Private/Public | Same logic as regular tasks |

### ✅ **Approval Flow**

|  |  |
| --- | --- |
| Action Point | System Behavior |
| Approver opens task | Sees “Approve” / “Reject” buttons + comments + file preview if uploaded |
| Approval is given | Status becomes “Approved”, timestamp + user name stored in history |
| Rejected | Status becomes “Rejected”, comments optional but encouraged |
| **Auto-Approval** enabled, time elapsed | Status marked “Auto-Approved (No Response)” |
| In sequential approval mode | Only current approver sees action buttons; next is notified post action |

### 🎯 **UI/UX Considerations**

* Approval tasks have a **checkmark or decision icon**.
* Inline approval buttons (email or app).

### 📋 **Use Cases**

* Legal contract approvals
* Budget & cost approvals
* Hiring panel decisions
* Policy changes approvals

### 🛡️ **Permissions**

* Only assigned approver can act.
* Task creator can edit due dates and change approver if needed.
* Admin can override in special cases (with comment trail).

## **Milestone Task**

### 🧩 **What Is a Milestone Task?**

A **Milestone Task** is a **checkpoint-type task** that represents a significant goal or event in a process. It can either:

* Exist **as a standalone task**.
* Or be **logically linked** to one or more other tasks/subtasks (dependencies).

### 🔧 **Creation Flow & Fields**

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Default / Behavior |
| **Task Name** | Short text | Mandatory |
| **Milestone Toggle** | Checkbox/Toggle | Mandatory toggle to mark task as Milestone |
| **Milestone Type** | Dropdown | Standalone or Linked to Tasks |
| **Link to Tasks** | Multi-select | Visible only if Linked is selected — choose tasks/subtasks to monitor |
| **Due Date** | Date | Mandatory |
| **Assigned To** | Dropdown | Mandatory, Default creator |
| **Description** | Long Text | Optional (for background, criteria, purpose) |
| **Visibility** | Private/Public | Default: Private |
| **Priority** | Dropdown | Default: Medium |
| **Collaborators** | Dropdown | Optional (for updates & comments visibility) |

### 🎯 **Behavior and Logic**

|  |  |
| --- | --- |
| Scenario | Behavior |
| Milestone is **linked** to task(s) | It **monitors** progress of those dependencies. |
| When **all linked tasks/subtasks** are completed | Milestone status icon **changes color**, e.g., turns **blue/green** |
| When **immediate upper task** (e.g., parent) is done | Visual cue (e.g., **flashing dot**, status tag changes) to prompt action |
| User must **manually confirm** Milestone is “Achieved” | With a “Mark Milestone Achieved” action |
| Milestone not marked after all dependencies complete | System can optionally notify the owner after configurable time window |

### 🎨 **UI/UX Enhancements**

* Milestones in list view show a **star icon (⭐)** or **flag**.
* Timeline/Dashboard views show them as **checkpoints or vertical markers**.
* In Task View, linked tasks appear under a "Dependency" tab.

### 📋 **Use Cases**

* “Product Beta Launch Complete”
* “Vendor Onboarded & Verified”
* “Training Cycle Finished”
* “Phase 1 Closure Milestone”

### 🛡️ **Permissions**

* Only assigned user or Admin can mark milestone as completed.
* Editing milestone dependencies is restricted to task creator or admin.

## **Task Update -**

### 🔄 Task Update – **Where & How Edit Happens**

Modern users expect **speed, minimal clicks**, and **in-context editing**. So, instead of sending users to a separate "Edit Page," Tasksetu will support **contextual editing** from wherever the user is interacting with tasks:

*Task Update is not task life cycle, it is updating/editing the task details.*

Task Status Update – in every view

#### **1. Task List View (Table/Grid or Kanban)**

|  |  |
| --- | --- |
| Trigger | Action |
| Click on Task Name or ID | Opens **Task Detail Modal** (Modal Coming in from Right) for full edit |
| Hover on editable fields (e.g. Assignee, Due Date, Priority) | Shows pencil/edit icon → click to **inline-edit** in place |
| Bulk Edit Option (Future Enhancement) | Checkbox-select multiple rows → shows “Bulk Edit” toolbar |

✅ **Why?** Fastest way for users working with multiple tasks  
🧠 **Design Tip:** Show minimal indicators unless user hovers or selects

#### **2. Task Detail Modal (Default Edit Container)**

|  |  |
| --- | --- |
| Where | Action |
| Fields like Task Name, Priority, Due Date, Assignee, Description, etc. | Editable directly from this modal. “Save” button |
| “More” Section | Opens a second modal (drawer or popover) for advanced settings like custom form, dependencies, related process |
| Collaborators | Added via searchable multi-select inside this modal |

✅ **Why?** Familiar interaction, full field visibility, fewer distractions  
🧠 **Design Tip:** Split into collapsible sections for better readability

#### **3. Kanban / Calendar / My Day View (Mini Update)**

|  |  |
| --- | --- |
| Trigger | Action |
| Click on Task Card | Opens **Task Quick View** drawer from right side, showing key info (assignee, due date, status, description) |
| Click “Edit” or icon inside | Converts into **inline editing controls** or opens full Task Modal |
| Drag-drop in calendar / Kanban – (Future Enhancement) | Updates date or status directly, without opening anything |

✅ **Why?** Great for quick updates without breaking workflow  
🧠 **Design Tip:** Use soft transitions, auto-save where possible

#### **Notification or Activity Feed (Jump-to-Edit) – (Future Enhancement)**

|  |  |
| --- | --- |
| Trigger | Action |
| User clicks on task reference in notification or feed | Opens Task Detail Modal in editable mode, directly focusing on the field they were notified about (e.g., overdue date) |

✅ **Why?** Brings the user exactly to where they need to act  
🧠 **Design Tip:** Highlight the affected field for 2-3 seconds to draw attention.

#### 🧭 Final Flow (User-Friendly Summary)

1. **User clicks anywhere they see the task** (board, list, calendar, search result)
2. **Task Detail Modal opens** with editable fields.
3. **Inline fields** like Task Name, Assignee, and Due Date become editable on click.
4. **Advanced settings** accessed via “More” → opens a layered modal.

## **Task Status Management**

### ✅ TASK STATUS MANAGEMENT

#### 🧩 Core Idea

* Every task and sub-task in Tasksetu must have a **Status** at all times.
* Status defines the **current state** of work on the task (e.g., "Open", "In Progress", "Completed").
* Each company can define its own statuses and the sequence/logic between them.

### 🔁 DEFAULT STATUS FLOW (System Default for All Companies)

These will apply only until the company defines their own.

|  |  |  |
| --- | --- | --- |
| Status Code | Label | Description |
| OPEN | Open | Task is created but not yet started |
| INPROGRESS | In Progress | Task is being worked on |
| ONHOLD | On Hold | Task is paused |
| DONE | Completed | Task has been completed |
| CANCELLED | Cancelled | Task was terminated intentionally |

### 🏢 COMPANY-LEVEL CONFIGURABLE STATUSES

Each company admin can:

* Define their **own statuses**, label them, order them.
* Choose which ones are “active” vs “final” statuses.
* Map status change rules (e.g., "On Hold" can go back only to "Open" or "In Progress").
* Set **default status** for newly created tasks/sub-tasks.

#### UI for Admin (Company Settings > Task Settings > Status Configuration):

* Add New Status (Text, Color Picker, Description)
* Drag-drop to reorder
* Mark as “Default” or “Final Status”
* Option to “Hide system statuses” (if fully custom)

\*\*\*\* The same kind of logic shall be applicable for priority settings as well \*\*\*\*

##### 🧱 **1. System-Defined Statuses (Tasksetu Core Layer)**

These are required to support core application logic, analytics, and consistency across features.

Example system statuses:

* Not Started
* In Progress
* On Hold
* Completed
* Cancelled

System status codes (e.g. SYS\_NOT\_STARTED) will never change, and **must** be used internally.

##### 🏢 **2. Company-Defined Statuses (Custom Layer)**

Companies can configure their own statuses via Admin Panel.  
Examples: The company defined status will be mapped with the system status codes only. The Admin must do this mapping in the settings.

* Pending → mapped to Not Started
* Working On → mapped to In Progress
* Blocked → mapped to On Hold
* Closed, Done → both mapped to Completed

🧩 **Rules:**

* **One system status can have multiple company statuses.**
* **Each company status must be mapped to a valid system status.**
* Mapping is **mandatory** at the time of status creation or update.
* Company can **edit names, colors, order**, but **not delete a system status mapping**.

##### ⚙️ **3. Admin Status Configuration Flow**

**Step-by-step:**

1. Admin goes to Company Settings → Status Management
2. Sees existing company-defined statuses (editable inline)
3. Can click “+ Add New Status”
   * Enters Status Name
   * Selects Color
   * Must map to a system status from dropdown
4. On Save, system checks:
   * **All system statuses have at least one mapping**
   * No duplication of status names
   * System prompts: “Changing a status will affect existing tasks”

##### 🔄 **4. What Happens to Existing Tasks?**

If a company **renames** or **remaps** a status:

* Existing tasks remain unchanged in database (linked to system code)
* UI dynamically renders using updated company-defined label
* Activity feed records:

"Task status updated display from ‘Pending’ to ‘New Pending’"

🛑 If a company **tries to delete a status**, system prevents it if it’s being used — or offers a **merge option** (move tasks to a different company status mapped to same system status).

##### 🌐 **6. UI Considerations**

* Status shown in dropdowns (e.g., task cards, Kanban board) will show company-defined statuses with color tags
* Admin view will have table:

|  |  |  |  |
| --- | --- | --- | --- |
| Company Status | Color | System Mapping | Tasks Using |
| Working On | Blue | In Progress | 452 |

### ⚙️ BEHAVIOR IN TASK VIEW / UPDATE

#### WHERE & HOW USER SEES / UPDATES STATUS:

* **In Task List View**:
  + Small status pill/badge
  + Hover → shows quick dropdown (editable for assigned users)
* **In Task Detail Modal**:
  + Inline dropdown for status (editable, modern style)
  + Change triggers confirmation pop-up for “final” statuses like "Completed" or "Cancelled"
  + Activity log records status changes (Status changed from OPEN to INPROGRESS by Priya)

### 🧠 RULES & LOGIC

#### Default Assignment:

* New task → Gets **company-defined default** status (else "Open")

#### Edit Permissions:

* Only:
  + Task assignee
  + Collaborators (if allowed)
  + Admins

#### Bulk Update:

* Multiple tasks can have status changed in Task List (bulk-select)

#### Status Logic for Sub-Tasks:

* Parent task can’t be marked “Completed” until all sub-tasks are either:
  + Done
  + Cancelled (optional setting)

### 🔄 HANDLING STATUS CHANGES (Retroactive)

When a company changes its status list (adds, renames, removes):

|  |  |
| --- | --- |
| Scenario | Behavior |
| A status is renamed | All tasks with old label auto-update display label (no data change) |
| A status is deleted | System prompts admin to **map old status to a new one** |
| A new default is set | All future tasks use the new default. Existing tasks are unaffected. |

#### Data Integrity:

* All statuses (even retired ones) are preserved in task history (activity logs).
* System keeps a hidden mapping table for legacy status values to prevent breaking reports.

### 🌈 USER FRIENDLINESS

* **Status Pill Colors**: Color-coded status labels (configurable per status)
* **Quick Status Change**: Dropdown shows only valid next statuses.
* **Autosave**: Inline changes auto-save when user clicks outside.
* **Tooltips**: Hover over status explains what it means (for clarity)

### 🔧 BACKEND DESIGN OVERVIEW (High-Level)

* **Table: status\_master**
  + status\_id, label, description, color, is\_final, is\_default, company\_id (null for global), active\_flag
* **Table: task\_status\_history**
  + task\_id, previous\_status\_id, new\_status\_id, changed\_by, changed\_at

### ✅ SUMMARY

|  |  |  |
| --- | --- | --- |
| Feature | Supported | Configurable |
| Default System Statuses | ✅ | ✅ |
| Company-Defined Statuses | ✅ | ✅ |
| Status Edit from Task List & Detail | ✅ | - |
| Retroactive Handling of Deleted Status | ✅ | - |
| Activity Feed Logging | ✅ | - |
| Sub-task Blocking Final Status | ✅ | ✅ (optional) |

## **Task Deletion**

### 🔹 Purpose:

Allow users to delete tasks they created or were assigned to, in a way that prevents accidental loss of data and maintains integrity of related task structures (like subtasks, dependencies, and history).

### 🧠 Deletion Logic

|  |  |
| --- | --- |
| Scenario | Action |
| Task has no subtasks/dependencies | Delete directly after confirmation. |
| Task has subtasks | Prompt: “This task has subtasks. Deleting it will delete all subtasks.” |
| Task is a subtask | Same as normal — delete with confirmation. |
| Task is part of a dependency | Prompt: “This task is referenced in another task. Are you sure you want to delete?” |
| Task has forms or attachments | Notify: “All linked forms and files will also be deleted.” |
| Task created by another user | Allow deletion **only if** current user is an admin or task assignee. |

### 🔐 **Permissions & Roles**

|  |  |
| --- | --- |
| User Role | Deletion Rights |
| Individual User | Can delete only their own created tasks or ones assigned to them. |
| Team User | Can delete tasks they created/are assigned to (same as above). |
| Company Admin | Can delete any task within their company’s workspace. |

### 🧩 **User Interface (UI) Flow**

1. **Access Point:**
   * Delete icon (🗑️) shown in:
     + Task card (hover or tap)
     + Task detail view (top-right menu)
     + Subtask listing (inline)
2. **Confirmation Prompt:**
   * Modal:

Are you sure you want to delete this task?

[ ] Also delete all subtasks and attached forms

This action is irreversible.

[Cancel] [Delete]

1. **UI Behavior:**
   * On delete:
     + Spinner / Loading indication
     + Snackbar/Toast: “Task deleted successfully”
     + Parent view refreshes to remove the deleted task

### 🧮 **Database Considerations**

* **Soft Delete Strategy** (Recommended):
  + Add is\_deleted, deleted\_by, and deleted\_at columns in the task table.
  + All queries exclude is\_deleted = true unless explicitly needed.
  + Enables potential future archival/restore functionality.
* **Cascade Deletes (on subtasks, form linkages):**
  + Soft-delete all related sub-tasks.
  + Flag file/form links for deletion (or leave orphaned clean-up to background job).

### 🔄 **Impact on Other Modules**

|  |  |
| --- | --- |
| Affected Module | Consideration |
| Subtasks | Ensure all nested subtasks are handled recursively. |
| Notifications | Suppress all pending notifications for the deleted task. |
| Analytics | Exclude deleted tasks from metrics by default (unless admin view). |
| Activity Feed | Log task deletion action (“Task deleted by {User} on {Date}”). |

### ⚠️ **Edge Case Handling**

|  |  |
| --- | --- |
| Case | Action |
| Task already marked deleted (race condition) | Show toast: “Task already deleted or inaccessible.” |
| Unauthorized deletion attempt | Block with alert: “You don’t have permission to delete this task.” |

### ✅ **Summary:**

* Deletion is confirmed, permission-based, and handled via soft-delete to allow data recovery in the future.
* Subtask/dependency/form relationships are handled with proper user prompts.
* Consistent UI pattern across desktop and mobile (icon + modal + toast).

## **Comments & Activity Feed**

#### 🎯 Purpose:

To allow users to collaborate, document decisions, and maintain a transparent history of actions on every task. The comments and activity feed should be available for all the tasks, sub-tasks.

#### 🔄 Where Accessible:

* **Task View Modal**  
  Always visible in a right-hand collapsible section (like Slack or Notion).  
  Tabs:
  + 🗨️ **Comments**
  + 🕓 **Activity Feed**

#### 🗨️ Comments

|  |  |
| --- | --- |
| Aspect | Description |
| Input Box | Always shown at bottom of Comments tab. Placeholder: “Leave a comment…” |
| Mentions | Use @ to mention any user in the workspace. Autocomplete supports names. Names should come from the user master for that company. |
| Formatting | Rich text support (Bold, Italic, Bullets, Code block, Line breaks) |
| Attachments | Drag & drop or click to attach (images, docs, etc.) within a comment |
| Emojis | Emoji picker and reaction support |
| Edit/Delete | Only allowed for own comments. Deleted comments show: “This comment was deleted.” |
| Notification | Mentioned users and assignee get notified via app and/or email |
| Timestamp | Each comment shows user name and time (hover to see exact datetime) |

#### 🕓 Activity Feed

|  |  |
| --- | --- |
| Action Type | Tracked |
| Task Created | “John created this task” |
| Field Updated | “Due Date changed from X to Y” |
| Subtask Added | “Subtask ‘X’ added by Jane” |
| Status/Priority Change | “Status updated to ‘In Progress’” |
| Assignment Change | “Task assigned to Sarah by Admin” |
| Comment Added/Edited | “Jane commented...” (links to comment) |
| File Attached/Removed | “Report.docx added by Anil” |
| Recurrence Updated | “Recurring pattern updated” |
| Form Attached/Submitted | “Form ‘Vendor Review’ submitted” |
| Task Completed/Reopened | “Task marked Complete by Ajay” / “Reopened by Neha” |

All activities are **read-only** and **timestamped**, with the username highlighted.

### 📎 Attachments & Links

#### 📍 Location:

* In the **Task Detail** (center or top section), there’s a **tab or attachment area** where all files and links are managed.
* Attachments added via:
  + Dedicated "Attach Files" button.
  + Drag & drop into task body or comment.
  + Date Time & User captured for identification.

#### 📂 Attachment Features

|  |  |
| --- | --- |
| Feature | Behavior |
| File Types Allowed | Images (JPG, PNG), Docs (PDF, DOCX), Sheets, Presentations, ZIPs, etc. – File type configurable at org level by the Admin |
| Upload Limits | Up to 02 MB per file (configurable at org level by The Admin) |
| Drag & Drop Support | Across all supported screens |
| Inline Preview | Images and PDFs open inside a lightbox. Others download by default. |
| Delete Option | Only by uploader or task owner/admin |
| Versioning (Future) | To track changes to same-named files |

#### 🔗 Links

|  |  |
| --- | --- |
| Feature | Behavior |
| Add Link | Click “+ Add Link” → Paste URL & optional title |
| Preview | Smart previews for known domains (YouTube, Google Docs, Figma, etc.) |
| Deletion | Only link adder or task owner can remove |
| Uses | External docs, dashboards, related tasks, vendor portals, etc. |

### 🔐 Permissions

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Comments | Activity Feed | Attachments & Links |
| Assignee | View/Add/Edit Own | View All | Add/Remove Own |
| Collaborator | View/Add | View All | Add (not remove) |
| Viewer | View Only | View Only | View Only |
| Admin | Full Rights | Full Rights | Full Rights |

### ⚠️ Edge Cases / Fail Safes

|  |  |
| --- | --- |
| Case | Handling |
| Link contains malware | Virus scan integration (future enhancement) |
| User deletes file in comment | File stays if referenced in other places, else removed |
| Mentioned user removed from task | They’ll still see the comment but won't get further updates |

### 🧠 UX Enhancements (Modern Experience)

* **Collapsible Side Panel**: To focus on task vs discussion.
* **Infinite Scroll with Lazy Load**: For long comment/activity threads.
* **Live Updates**: Comment added by other users appear instantly (socket-based).
* **Comment Bookmarking (Future)**: For key decisions.
* **Filters in Activity Feed**: View only status updates, comments, etc.

### ✅ Sample Scenario

Imagine a user adds a comment:

@Ritu please attach the final pricing sheet. See draft below.  
(attaches file)

Then the assignee changes the due date.  
In activity feed:

* “Ritu was mentioned in a comment”
* “Due date updated from Apr 30 to May 2”

## **Task Snooze / Postpone**

#### 🎯 **Purpose:**

Allows a user to temporarily pause or delay focus on a task **without rescheduling its due date**. This is useful when the user wants to remove the task from immediate attention but doesn’t want to forget it.

#### 🧩 **Where This Feature Lives:**

* Available on the **task card** (mini-menu or hover actions)
* Inside the **task detail view**
* Optionally, on **calendar view** or **My Tasks**

#### ⚙️ **Field Logic & Behaviors:**

|  |  |  |
| --- | --- | --- |
| Field / Control | Type | Behavior |
| **Snooze Until** | Date + Time Picker | Default: Next day 9 AM  Options: Today Eve, +1 day, +3 days, Next Monday, Pick a date |
| **Note/Reason (Optional)** | Text field | To capture user's intent |
| **Restore on Wake-up** | Auto | Task comes back to “My Tasks” dashboard / notification |

#### 🪄 **What Happens Behind the Scenes:**

* Task remains **assigned**, due date is **unchanged**
* Task becomes hidden from “My Tasks” dashboard until the snooze date
* **Activity log** shows snooze
* **Email/app notification** sent at “wake-up” time

#### 🔒 **Permissions:**

Only the **assigned user** can snooze the task.  
Admins can **unsnooze** or see snoozed tasks if needed.

### ✅ **Notifications & Reminders**

#### 🎯 **Purpose:**

Keeps users informed about their tasks, deadlines, updates, and collaboration. Ensures nothing slips through the cracks.

#### 🔔 **Types of Notifications:**

|  |  |  |
| --- | --- | --- |
| Type | Trigger Point | Sent To |
| **Task Assignment** | Task created or reassigned | Assignee |
| **Due Date Reminder** | Configurable (e.g., 3 days before, 1 day before) | Assignee, Collaborators |
| **Overdue Reminder** | Daily or every 3 days after due date | Assignee |
| **Comment Mention** | @mention in comment | Mentioned user |
| **Status Change** | Task status updated | Assignee, Collaborators |
| **Custom Reminder** | User-set reminders | Self or others |
| **Snooze Wake-up** | Snoozed task is due to wake | Assigned user |

#### 📬 **Delivery Channels:**

* **In-app notifications** (top right bell icon with number badge)
* **Email** (for important actions, summarized daily/weekly if preferred)
* **Push notification** (if mobile app is integrated later)
* **Integrations**: Outlook / Google Calendar sync for reminders

#### ⚙️ **Reminder Settings (User Configurable):**

* Go to: Settings > Notifications
* Options to:
  + Turn on/off specific notification types
  + Set custom reminder intervals
  + Choose delivery method (email/app/both)

#### 🧠 UX Considerations:

* Default reminders should **not overwhelm**
* Users can mark “Do Not Disturb” windows
* Smart Reminder Suggestion: If user hasn't interacted with a task 2 days before due date, send a subtle nudge.

### ✨ Visual Cues on Task UI:

|  |  |
| --- | --- |
| Cue | Purpose |
| 🔕 Icon + Tooltip | Task is snoozed |
| ⏰ Clock Icon | Reminder is active |
| 🔴 Overdue Badge | Task past due date |

### 📝 Audit & Activity Trail:

* **Snooze actions**, **notifications triggered**, and **reminders sent** all logged in task’s **Activity Feed** with timestamps.

# ✅ HOME / DASHBOARD PAGE (Unified Experience)

### 🎯 **Purpose**

Single landing hub for employees and managers with a **configurable, intelligent view** of tasks, analytics, and calendars.

### 📌 **Universal Design Principles**

* Same page = **Home + Dashboard**
* **Role-specific data loading** (Employee/Manager/Admin)
* Mobile-friendly layout with collapsible tiles and charts
* Option to customize **columns, filters, sort order**, and **column sequence**
* **Global Search bar** with deep filters (assignee, tag, status, priority, form data, etc.) - Integrated

### 👨‍💼 Employee Home/Dashboard View

|  |  |
| --- | --- |
| Section | Description |
| **My Tasks Table** | Filterable, sortable, customizable columns (status, due date, tag, etc.) |
| **Quick Filters / Tabs** | Due Today, Overdue, Snoozed, Completed This Week, Assigned by Me |
| **Task Calendar View** | Month/week/day view of task due dates, drag-drop to reschedule |
| **Reminders & Alerts** | Comment mentions, pending actions, snoozed reactivations |
| **Approvals Needed** | Approval tasks waiting for your action |
| **Pinned Tasks** | Bookmarked or frequently accessed tasks |
| **Create New** | Task, Sub-task, Recurring Task, Form |

### 👨‍👩‍👧 Manager Home/Dashboard View (Superset of Employee View)

|  |  |
| --- | --- |
| Section | Description |
| **My Tasks** | Just like employee view |
| **Team Tasks Matrix** | Table showing team members vs # tasks: Assigned, Overdue, At Risk, etc. |
| **Overdue & Risky Tasks** | Highlighted section with “no activity for X days”, past due date, blocked |
| **Team Milestones** | Timeline chart for upcoming and overdue milestones |
| **Delegated Tasks** | Tasks created by manager but assigned to others |
| **Approval Workflows** | View & act on approval tasks pending within team |

### 🧱 Dashboard Tiles & Visuals (Shown on Both Views Based on Role)

|  |  |
| --- | --- |
| Tile / Chart | Description |
| **🟡 My Pending Tasks** | Count + filtered view of your pending tasks |
| **🔴 Overdue Tasks** | Pie/Count with drill-down capability |
| **🟢 Completed This Week** | Line chart with day-wise completion trend |
| **🧩 Milestones This Month** | Timeline view (color-coded: pending, due, achieved) |
| **✅ Approvals Awaiting** | Count of pending approval actions |
| **📊 Task by Priority** | Pie/Column chart (Critical, High, Medium, Low) |
| **📅 Task Calendar** | Embedded drag-drop calendar to reschedule or assign tasks |
| **📈 Team Activity Trend** | Bar chart of tasks completed/overdue across team (for managers) |

### 🔄 Column Configuration

Each employee (and manager) can:

* Select the columns they want in their Task Table view.
* Reorder column sequence via drag-drop.
* Save and reset column layout per preference.

Optionally export filtered view (CSV, XLS).