Logic_Document

Smart Assign – Logic Overview

Purpose:

Smart Assign is designed to intelligently distribute tasks among users in the collaborative Kanban board, ensuring a balanced workload and preventing user overload. It automates the assignment process, reducing manual effort and bias.

Implementation Summary:

- When a user triggers Smart Assign for a task, the system evaluates all registered users and counts their currently assigned tasks that are either in 'Todo' or 'In Progress' status.
- The user with the fewest such active tasks is selected as the assignee for the new task.
- This approach is rule-based and deterministic, always choosing the least-loaded user at the moment of assignment.
- Factors considered:
  - Number of active (incomplete) tasks per user
  - Only tasks in 'Todo' or 'In Progress' are counted (completed tasks are ignored)
  - No consideration of user availability, time slots, or task priority in the current logic

Example:

If 3 users are available, and one has fewer active tasks, the system assigns the next incoming task to that user to maintain balanced workload.

Conflict Handling – Logic Overview

Purpose:

The system handles conflicts that arise from concurrent task edits, such as two users attempting to modify the same task simultaneously (race conditions, data overwrites).

Conflict Scenarios:

- If a user tries to update a task that is currently being edited by another user, a conflict is detected.
- The backend checks a task's `isBeingEdited` flag and `editedBy` field to determine if another user is editing the task.
- Example 1: User A opens a task for editing. User B tries to edit the same task before User A finishes. The system detects the conflict and blocks User B's update.

- Example 2: Two users submit changes to the same task nearly simultaneously. The system prompts the second user with a conflict resolution dialog.

Resolution Strategy:

- When a conflict is detected, the user is presented with a modal dialog showing both the current (server) version and their own (local) version of the task.
- The user can:
  - Overwrite the server version with their changes
  - Manually merge fields from both versions (field-by-field selection)
  - Cancel their changes
- The system uses timestamps and edit flags to detect and manage conflicts, but resolution is user-driven for transparency and control.

Example:

If two users are assigned overlapping tasks, the system checks timestamped logs and keeps the earlier entry while prompting the second user with a resolution dialog.

☑ Design Decision Highlights

- Why this logic?

  - Smart Assign ensures fair, transparent, and automated task distribution, reducing bottlenecks and manual bias.
  - Conflict handling prioritizes data integrity and user awareness, preventing silent overwrites and supporting collaborative editing.

- Benefits:

  - Performance: Simple, efficient queries for assignment and conflict checks.
  - UX: Clear feedback and resolution options for users, reducing frustration.
  - Scalability: Logic can be extended to consider more factors (e.g., user skills, availability) as the app grows.

- Future Scope:

  - Incorporate user availability, task priority, or skill matching into Smart Assign.
  - Add auto-rescheduling or notification-based conflict resolution.
  - Enhance real-time collaboration with live editing indicators.