Logic Document

Smart Assign Logic

Objective:

Automatically assign a task to the user with the fewest number of active (incomplete) tasks.

How it works:

1. When a task needs to be auto-assigned, we trigger the smartAssignTask API.
2. This API internally calls an aggregation query to **group all active tasks** (status !== "Done") by assignedTo field.
3. It then **counts the number of tasks per user**.
4. Among these users, we select the one with the **lowest task count**.
5. This user is assigned to the task, and an **action log is recorded** showing who triggered the smart assign and what changes were made.
6. If no users have active tasks (i.e., all tasks are unassigned), the system simply picks the first user from the user list.

Conflict Handling Logic

Objective:

Prevent accidental overwrites when **multiple users edit the same task simultaneously.**

How it works:

1. When a user opens a task to edit, we store the updatedAt timestamp.
2. When the user submits an update, we **include this original timestamp** (clientUpdatedAt) in the API request.
3. On the backend, we compare the updatedAt timestamp from the database with the one sent by the client — **if they don't match, a conflict is detected**.
4. In case of a conflict, backend responds with a **409 Conflict status** which includes the **latest task from the server**.
5. On the frontend:
    - We notify the user of the conflict.
    - The user is presented with **two versions**:
        - Their own edited version.
        - The latest version from the server.
    - The user can choose to:
        - **Overwrite** (force update).
        - **Merge manually**.
        - **Cancel** to review changes.

Example:

- User A opens Task X at 12:00 PM.
- User B opens Task X at 12:01 PM.
- User B updates and saves Task X at 12:02 PM.
- User A (who hasn’t refreshed) tries to save their version at 12:03 PM.
- Since Task X’s updatedAt changed, a conflict is detected and User A is warned before overwriting.

Outcome:

- Prevents data loss from simultaneous edits.
- Gives users control over how to resolve conflicts.
- Works seamlessly with real-time sync for a collaborative experience.