Goal Getters

A ToDo app

For all of your task tracking needs

Solves Organizational Problems by organizing and Creating Tasksasks

Team Rules

- Communicate Clearly
- Hold yourself accountable
- Follow KISS
- No Surprises

Team Members & Roles & Sprint Plants

Dillon Carpenter: Team Leader

- Coordinates the Team
- Does weekly progress reports and presentations
- Other Team Leader responsibilities

David-Michael Davies: Team Member

Sprint 1

- Week 4: Starting the First Feature
  - Set up the schema
  - Connect to the database
  - Ensure these elements update properly
- Week 5: Finish the First Feature, Start second feature
  - Ensure the elements update properly
  - Set up the Sign up/in/out pages
- Week 6: Continue Second Feature
  - Set up authentication functionality
  - Implement Password Hashing

Sprint 2

- Week 10: Start the Third Feature
  - Search bar and filtering options
  - Ensure these elements respond to user input
- Week 11: Finish the Third Feature
  - Clear search option
  - Ensure that the clear search option resets the search results
- Week 12: Start the Last Feature
  - Set up recurring task option
  - Set up recurrence schedule
  - Ensure recurrence schedule is stored in the database
- Week 13: Finish the Last feature
  - Ensure recurrence schedule is stored in the database

Nabou Diouf: Team Member

Sprint 1

- Week 4 (Sprint 1–1) — Start Feature 1: Completion Status
  - Add button to toggle completion
  - Save completion status and load it on refresh
- Done when: toggle works + persists after refresh.

- Week 5 (Sprint 1–2) — Finish Feature 1
  - Completed tasks look visually different
  - Allow un-completing tasks (completed to incomplete)
  - Quick manual test checklist
- Done when: completion status and UI persistence are all correct.

- Week 6 (Sprint 1–3) — Build Feature 2: Categories/Tags
  - Add category field
  - Category select on “Create Task”
  - Color mapping for categories (consistent)
  - Show colored badge/label in the list
  - Persist category after refresh
- Done when: same category always same color + persists.
- Week 7 (Sprint 1–4) — Finish Feature 2 + Sprint 1 polish
  - Category editing for existing tasks
  - UI polish
  - Edge cases: prevent empty/invalid category
- Done when: Sprint 1 is demo-ready (Feature 1 + 2 complete)

Sprint 2

- Week 10 (Sprint 2–1) — Start Feature 3: Due Date Input + Storage
    - Add due Date field to tasks
    - Due date input on create/edit
    - Display due date in list
    - Persist due date after refresh
- Done when: due dates store correctly and survive refresh.

- Week 11 (Sprint 2–2) — Finish Feature 3: Overdue + Due Soon Alerts
    - Overdue label and highlight when past due
    - Due Soon label (choose window: next 24 hours OR next 7 days)
    - Alerts update automatically based on current date/time
- Done when: overdue/due soon labels are correct and update reliably.

- Week 12 (Sprint 2–3) — Start Feature 4: Sorting
    - Sort by due date (Soonest to Latest, Latest to Soonest)
    - Sort by priority (High to Low, Low to High)
- Done when: sorting works cleanly and doesn’t break alerts.

- Week 13 (Sprint 2–4) — Finish Feature 4: Filtering + Combine Sort/Filter
    - Filters: Category, Priority, Status (Completed/Incomplete), Overdue
    - Combine sort and filters
    - Clear all filters button
- Done when: combined filtering/sorting produces correct results.

- Week 14 (Sprint 2–5) — Integration and Testing
    - Handle tasks with no due date safely (sorting and alerts)
    - Decide consistent placement for “no due date” (bottom recommended)
    - Fix bugs and complete regression checklist
- Done when: stable in all normal.

Sam Stoeckel: Team Member

Sprint 1

- Week 4: Beginning of frontend for CRUD implementation
- Week 5: Integration for simple /add, /delete, edit functions with integration with Database
- Week 6: Basic implementation of list view
- Week 7: Added functionality for each task on the list view.

Sprint 2

- Week 10: Implement frontend routing between all current views.
- Week 11: General UI/UX Navigation improvements. Bug fixes.
- Week 12: Beginning implementation of the calendar view.
- Week 13: Complete calendar view interactivity.
- Week 14: Flex Week

Total Features and Requirements

- Total Features: 12
- Total Requirements: 46
- Total Projected Tests: 36