**1. Project Overview**

The Plan & Task Tracker is a dynamic, single-page web application designed for streamlined project and task management. Its core purpose is to provide a clear, real-time view of project progress where tasks are sequential and dependent on one another.

The application automatically calculates and adjusts project timelines based on a working-day calendar (excluding weekends), user inputs, and delays. It serves as a centralized hub for tracking multiple plans, logging all changes for auditing purposes, and capturing task-specific feedback. Built with HTML, Tailwind CSS, and a Firebase Firestore backend, the application is both responsive and highly interactive.

**2. Key Features**

* **Dashboard View:** A central screen displaying all created plans as interactive cards, showing high-level information like Plan Name, Status, Completion Percentage, and Tentative End Date.
* **Plan Creation:** Users can create new plans by providing a name and a target start date for the first task. The application automatically blocks the selection of past dates and weekends.
* **Automatic Sequential Scheduling:** Based on predefined rules (e.g., Task 2 takes 2 working days), the system instantly calculates the target dates for all subsequent tasks and the final plan end date.
* **Dynamic Timeline Recalculation:** If a task is completed late or a target date is manually changed, the application automatically recalculates and pushes the target dates for all future dependent tasks, providing a constantly accurate forecast.
* **Task Management:**
  + **Mark Completion:** Users can mark the currently active task as complete.
  + **Change Target Date:** The target date for the active task can be manually changed, which reschedules the rest of the plan.
  + **Action Locking:** To ensure a logical workflow, actions (like "Mark Complete", "Change Date", "Comments") for a task are disabled until the preceding task is completed.
* **Status Tracking:** Each plan has a mutable status (TBS, In-Progress, On Hold, Done) that can be changed at any time from the detail view.
* **Task-Specific Comments:** Instead of a general comment log, each task has its own dedicated section for adding and viewing remarks, allowing for more organized feedback.
* **Comprehensive History Log:** Every significant action—plan creation, task completion, date changes, status updates—is automatically recorded in a time-stamped, read-only log for complete auditability.
* **Excel Export:** Users can download a detailed Excel file for any plan. The file contains multiple sheets: a plan summary, a full task list with all dates, a consolidated list of all task comments, and the complete history log.

**3. Use Cases**

The application is ideal for processes that follow a consistent, sequential workflow.

* **Client Onboarding:** Tracking the standard set of tasks required to onboard a new client, from initial contract to final setup.
* **Small Project Management:** Managing simple projects with dependent phases where one phase must be complete before the next begins.
* **Standard Operating Procedures (SOPs):** Ensuring a standard, multi-step process is followed consistently every time it's executed.
* **Personal Goal Tracking:** Breaking down a large personal goal (e.g., "Complete a certification") into a series of sequential steps and tracking progress.

**4. User Stories**

* **As a Project Manager, I want to** create a new plan with a specific start date **so that** I can get an automatically generated, realistic timeline for the entire project.
* **As a Project Manager, I want to** see all my active plans on a single dashboard **so that** I can quickly assess the status and progress of each one at a glance.
* **As a Team Member, I want to** mark a task as complete **so that** the project manager is notified and the next task in the sequence becomes active.
* **As a Project Manager, I want to** be able to change the target date of an upcoming task **so that** I can adjust the plan's schedule in response to new information, with all future dates updating automatically.
* **As a Team Member, I want to** add comments directly to a specific task **so that** my feedback is contextual and easy for others to find.
* **As a Project Manager, I want to** view a complete history log for a plan **so that** I can audit all changes, delays, and completions that have occurred.
* **As a Project Manager, I want to** download a complete summary of a plan as an Excel file **so that** I can share detailed progress reports with stakeholders or for offline analysis.
* **As a User, I want** the system to prevent me from taking actions on future tasks **so that** I don't accidentally alter parts of the plan that are not yet active.

**5. User Flow**

1. **Landing & Dashboard:** The user arrives on the **Dashboard View**, which displays a grid of all existing plans. If no plans exist, a message prompts them to create one.
2. **Plan Creation:**
   * The user clicks the "+ Create New Plan" button, which opens a modal.
   * They enter a "Plan Name" and select a "Task 1 Target Date" from a calendar that disallows past dates and weekends.
   * Upon submission, the system creates the plan, calculates all subsequent task dates, and adds the new plan card to the dashboard.
3. **Viewing a Plan:**
   * The user clicks on a plan card from the dashboard.
   * They are navigated to the **Plan Detail View** for that specific plan.
4. **Interacting with a Plan:**
   * In the detail view, the user can see the overall plan progress, change the plan's status, or download the details to Excel.
   * They see a list of all 5 tasks. Only the first incomplete task (the "active" task) has its action buttons ("Mark Complete", "Change Date", "Comments") enabled. All subsequent tasks are locked.
5. **Task Completion & Progression:**
   * The user clicks "Mark Complete" on the active task. A modal appears asking for the "Actual Completion Date".
   * If the date is late, the timeline for all future tasks is automatically pushed back.
   * The completed task is now marked "Completed", and the *next* task in the sequence becomes active, enabling its action buttons.
6. **Making Adjustments:**
   * The user can click "Change Date" on the active task to reschedule it, which also reschedules all subsequent tasks.
   * The user can click "Comments" on any active or completed task to open a modal where they can read existing comments or add new ones for that task.
7. **Reviewing History:** The user can review the "History Log" on the right side of the detail view to see a full, time-stamped record of all activities.
8. **Returning to Dashboard:** The user clicks the "Back to Dashboard" button to return to the main view of all plans.

**6. Technical Details**

**Technology Stack**

* **Frontend:**
  + **HTML5:** For the core structure and content of the application.
  + **Tailwind CSS:** A utility-first CSS framework for rapid, responsive UI development.
  + **Vanilla JavaScript (ES6+):** For all client-side logic, including DOM manipulation, event handling, and data processing.
* **Backend (BaaS - Backend as a Service):**
  + **Google Firebase:** A comprehensive platform for web and mobile applications.
    - **Firestore:** A flexible, scalable NoSQL cloud database for storing all plan and task data in real-time.
    - **Firebase Authentication:** Used for managing user sessions (including anonymous sign-in).
* **Libraries:**
  + **SheetJS (xlsx):** A JavaScript library used for generating and downloading .xlsx Excel files directly from the browser.

**Database Schema (Firestore)**

The database is structured in a hierarchical NoSQL format to logically group related data.

* **Root Collection:** artifacts/{appId}/public/data/plans
  + This is the main collection that holds all individual plan documents.
* **plans Document:**
  + **Document ID:** A unique, auto-generated ID for each plan.
  + **Fields:**
    - planName (String): The user-defined name of the plan.
    - status (String): The current status (e.g., "In-Progress", "Done").
    - completionPercentage (Number): The calculated progress percentage.
    - tentativeEndDate (Timestamp): The projected end date of the last task.
    - createdAt (Timestamp): The server timestamp when the plan was created.
* **tasks Sub-collection:** (Path: /plans/{planId}/tasks)
  + A sub-collection within each plan document.
  + **Document ID:** A unique, auto-generated ID for each task.
  + **Fields:**
    - taskName (String): The name of the task (e.g., "Task 1").
    - sequence (Number): The order of the task (1-5).
    - durationInWorkingDays (Number): The business rule for how long the task takes.
    - originalTargetDate (Timestamp): The initially calculated target date.
    - revisedTargetDate (Timestamp): The new target date after a change.
    - actualCompletionDate (Timestamp): The date the user confirmed completion.
    - isCompleted (Boolean): true if the task is done, otherwise false.
* **comments Sub-collection:** (Path: /plans/{planId}/tasks/{taskId}/comments)
  + A sub-collection within each task document to store task-specific comments.
  + **Document ID:** A unique, auto-generated ID for each comment.
  + **Fields:**
    - commentText (String): The content of the user's comment.
    - author (String): The UID of the user who commented.
    - timestamp (Timestamp): The server timestamp when the comment was added.
* **logs Sub-collection:** (Path: /plans/{planId}/logs)
  + A sub-collection within each plan document to store the history.
  + **Document ID:** A unique, auto-generated ID for each log entry.
  + **Fields:**
    - message (String): A human-readable description of the event.
    - timestamp (Timestamp): The server timestamp of the event.