Automated Task Management

Problem Statement

In many organizations, employees and sales reps manage their daily work through spreadsheets, emails, or manual notes. This fragmented and manual approach leads to missed deadlines, inconsistent follow-ups, and poor accountability. Managers lack visibility into overdue tasks, workload distribution, and overall team performance.

The absence of a centralized task management and reminder system creates critical challenges :

- **Missed Deadlines:** Without automated reminders, users forget important due dates.
- **Low Accountability:** Manual tracking makes it hard to identify ownership of tasks.
- **Manager Blind Spots:** Supervisors struggle to monitor team productivity.
- **Inefficient Reporting:** No consolidated data on open, completed, or overdue tasks.
- **Poor Productivity Insights:** Leadership cannot measure trends or optimize workloads.

Solution Framework

The solution is to build an Automated Task Management system on Salesforce that centralizes task creation, assignment, and tracking. Using Salesforce automation tools like Flows, the system will :

- Send proactive reminders to assigned users ahead of deadlines.
- Escalate overdue tasks to managers for accountability.
- Provide real-time reports and dashboards for visibility.

This transforms task management from a manual process into an automated, efficient, and measurable workflow.

Requirement Gathering

Translate business problems into a detailed set of functional, non-functional, data, and integration requirements for designing the solution.

- **Workshops with Managers & Employees:** Capture pain points with current task tracking.
- **Stakeholder Interviews:** Understand expectations around reminders, escalation, and reporting.
- **Document Analysis:** Review existing spreadsheets, emails, and manual methods currently in use.

Stakeholder Analysis

Stakeholder Group	Key Interest / Motivation	Potential Influence
Employees / Sales Reps	Timely reminders, clear task ownership	High
Managers	Visibility into team workload & overdue tasks	High
Admins	Configure automation & workflows	Medium
Executives	Productivity insights, accountability reports	High
Customers (optional)	On-time service requests/tasks	Medium

Document the current manual processes and illustrate how Salesforce improves them. Current Process (As-Is):
- Tasks tracked via Excel or emails.
- No structured reminder system → deadlines often missed.
- Managers manually follow up with employees.
- No consolidated reports or dashboards.

Proposed Process (To-Be with Salesforce):
- Tasks created & tracked centrally in Salesforce.
- Automated email/SMS reminders ahead of deadlines.
- Overdue tasks auto-escalated.
- Real-time dashboards show open, completed, and overdue tasks by user/team.

Use Case Analysis for Task Management

Objective: Show how the system addresses specific organizational challenges.

- **Task Completion Problem:** Deadlines missed due to lack of reminders.
  Solution: Automated reminders and escalations.

- **Manager Visibility Problem:** Hard to track workload manually.
  Solution: Dashboards and performance reports.

- **Overdue Task Handling Problem:** No escalation path for critical tasks.
  Solution: Auto-escalation workflows in Salesforce.

- **Productivity Insights Problem:** No measurable data for trends or optimization.
  Solution: Reports with insights on open, completed, and overdue tasks.

AppExchange Exploration

Objective: Explore Salesforce apps to enhance functionality and reduce custom build effort.

Potential Apps:

- Email/SMS notification apps → automated reminders.
- Calendar integration apps (Google/Outlook).
- Productivity tracking apps → efficiency measurement.

Evaluation Criteria:

1. Functional Fit: Task reminders, escalation, and tracking.
2. Platform Integration: Seamless with Salesforce Flows, Reports, Dashboards.
3. Scalability: Handle large task volumes without performance impact.
4. Cost: Licensing and maintenance feasibility.
5. User Reviews: Adoption and satisfaction by other organizations.

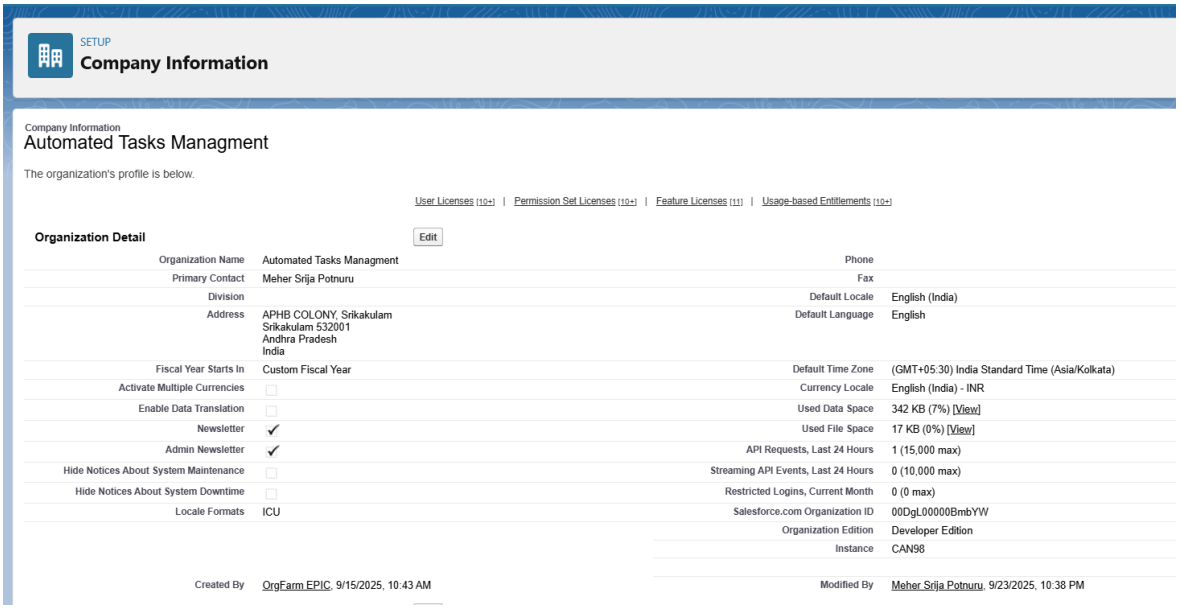
Automated Tasks Manager

Phase 2: Org Setup & Configuration

This phase is like laying the foundation for a smooth task management ecosystem in Salesforce. Before we automate reminders or escalations, we must ensure that the organization settings, profiles, security, and data visibility are properly configured. Getting this right ensures scalability, accountability, and long-term maintainability of the Automated Task Management system.

Company Profile Setup

The company information has been configured to reflect the organization's identity and task management needs.



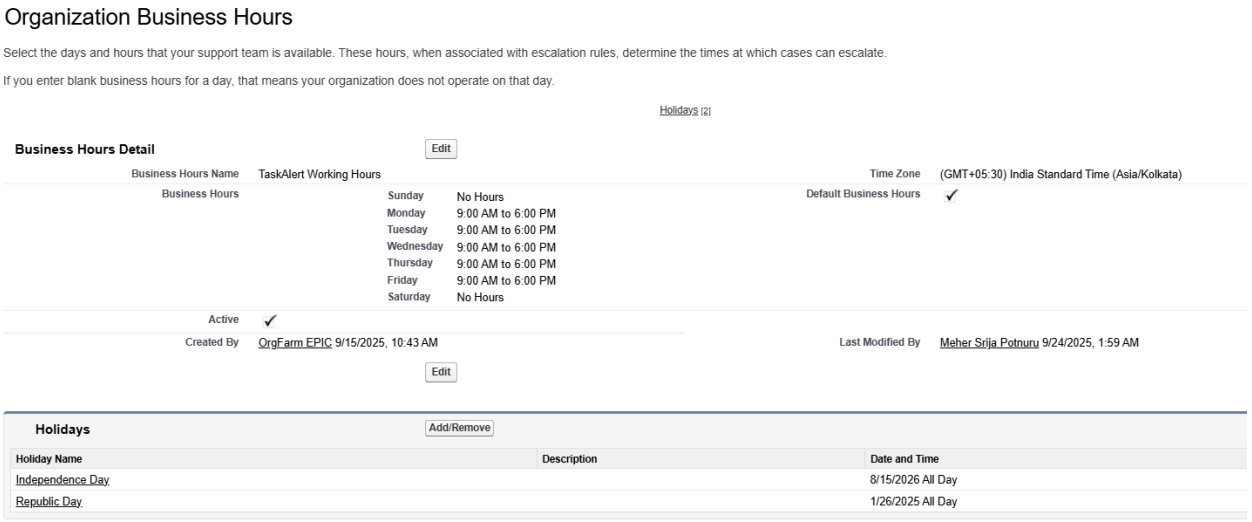
- **Organization Name:** Automated Tasks Management
- **Primary Contact:** Meher Srija Potnuru
- **Address:** APHB Colony,Srikakulam,Andhra Pradesh,India.
- **Default Locale:** English (India)

- **Default Language:** English
- **Time Zone:** India Standard Time (GMT+05:30, Asia/Kolkata)
- **Currency Locale:** INR (English - India)
- **Multiple Currencies:** Enabled

This ensures all users experience Salesforce in their local context for reminders, deadlines, and escalations.

Business Hours & Holidays

Defining business hours is critical to determine when reminders, escalations, and follow-ups should be sent.

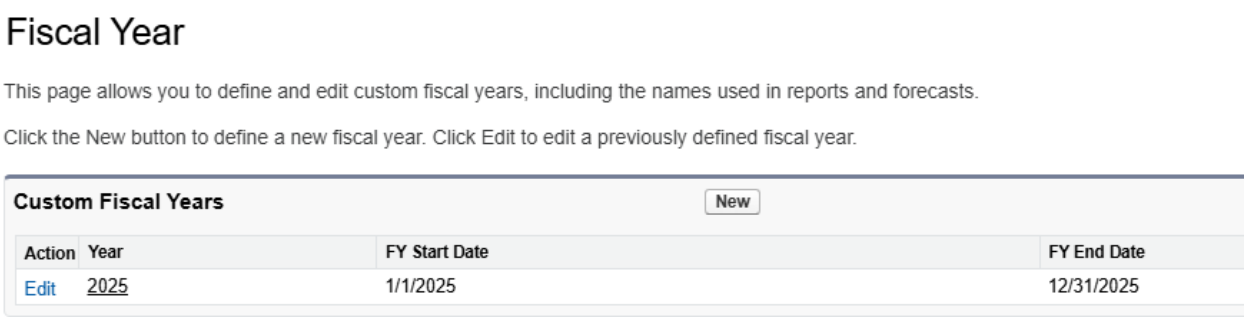


- **Business Hours Name:** TaskAlert Working Hours
- **Schedule:** Monday to Friday – 9 Hours
- **Default Business Hours:** Enabled
- **Active:** Yes
- **Holidays Configured:** Two Holidays – All Day

These settings ensure that overdue tasks escalate only during active working days, avoiding unnecessary alerts on holidays.

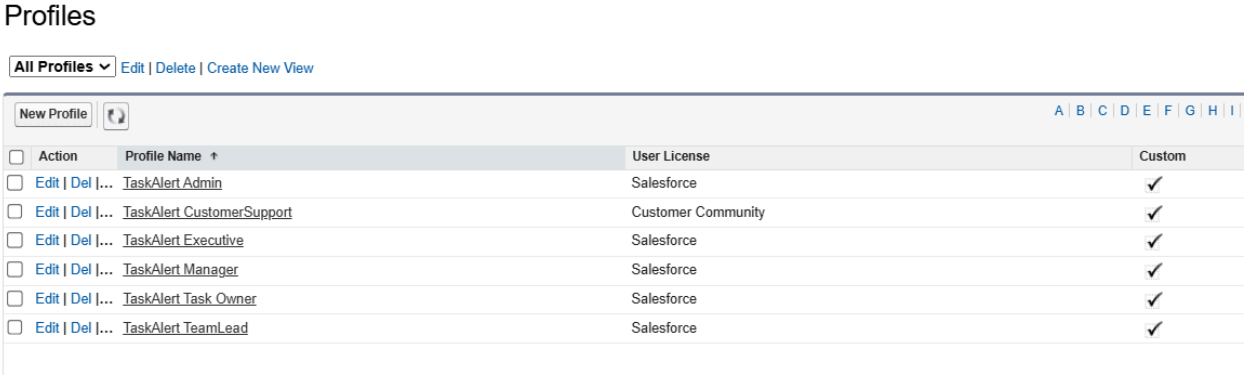
Fiscal Year Settings

All reporting, workload distribution, and productivity insights will align with the company's **Custom Fiscal Year**. This ensures that dashboards reflect accurate quarterly and annual task completion metrics.



Profiles

Adopted a **Minimum Access security model**. Never assign the standard **System Administrator** profile to anyone but dedicated admins. Clone the **Standard User** profile to create custom profiles for key personas in the Automated Task Management project:

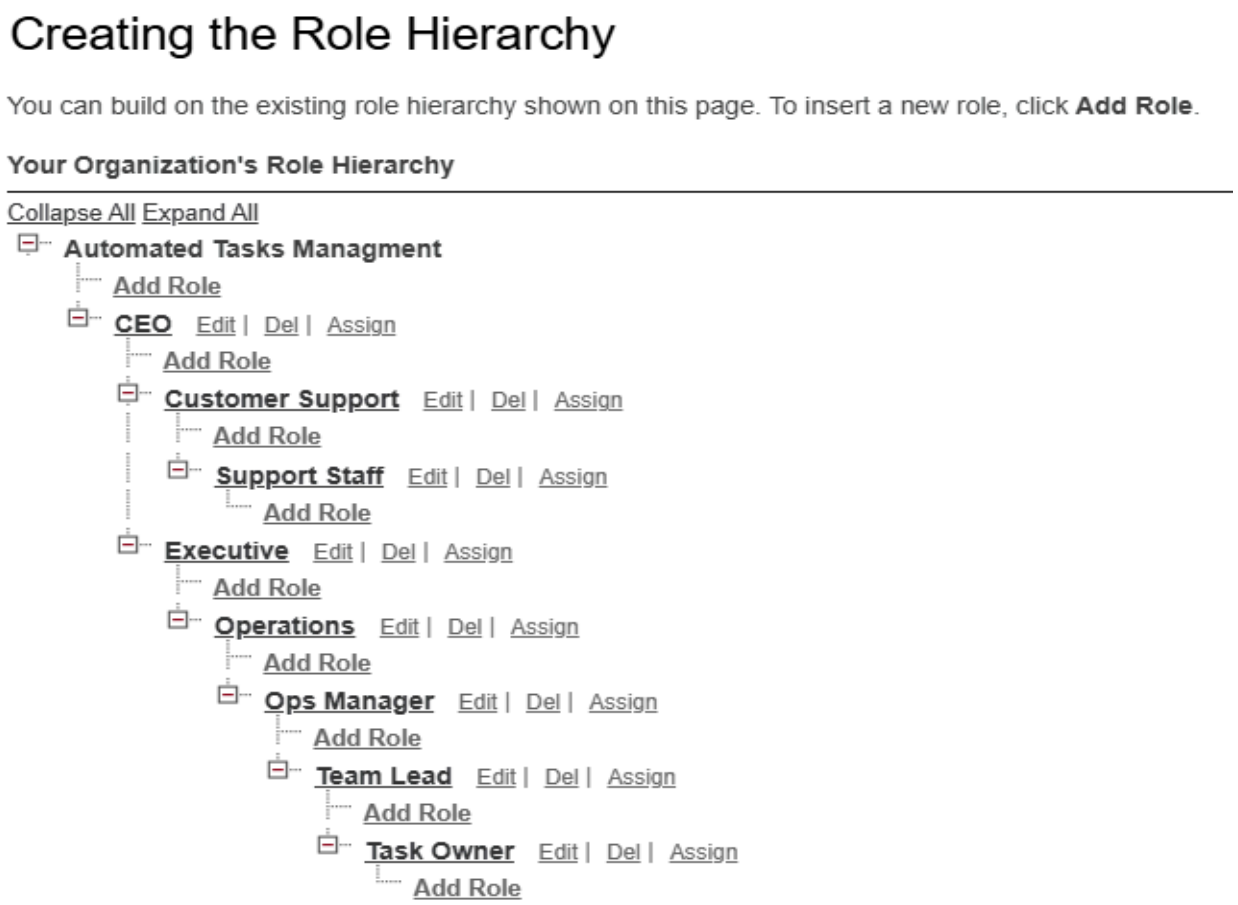


- **TaskAlert Admin** – Full access to configure automation, dashboards, and security for the task management system.
- **TaskAlert CustomerSupport** – Access to customer tasks and cases in the community, with reminders and escalation features.
- **TaskAlert Executive** – View-only access to high-level dashboards and reports for productivity insights.
- **TaskAlert Manager** – Manage team tasks, oversee escalations, and monitor workload distribution.
- **TaskAlert Task Owner** – Limited access to their own tasks, reminders, and completion updates.
- **TaskAlert TeamLead** – Can monitor team tasks, assist with escalations, and coordinate between employees and manager.

Roles

Roles define the visibility hierarchy for tasks. Example:

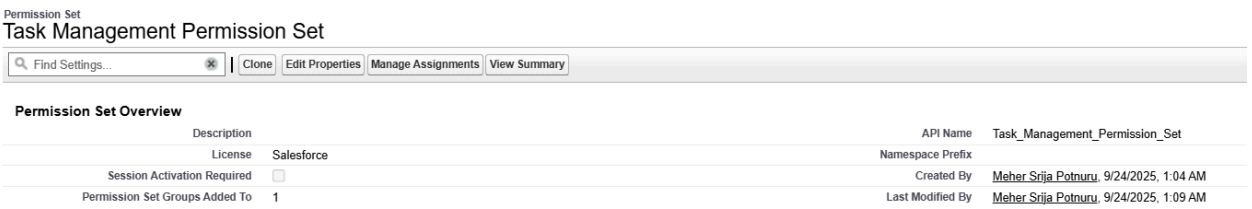
- **Executive** – Can view all task reports and productivity metrics.
- **Manager** – Can view and escalate overdue tasks for their team.
- **Employee / Sales Rep** – Can only see their assigned tasks and reminders.



Permission Sets

Used to grant additional access without modifying profiles. For example:

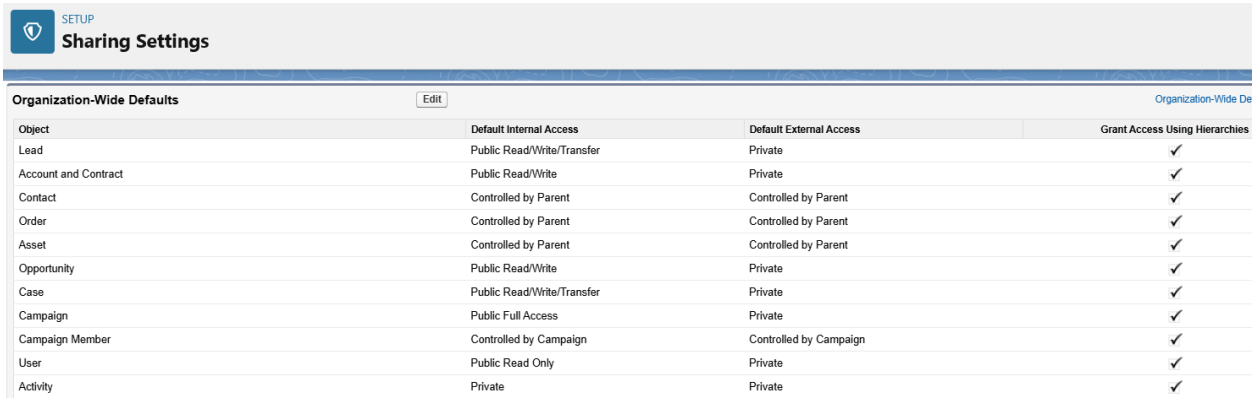
- **Reporting Access** – Provides dashboard/report visibility to specific users.
- **Escalation Management** – Grants managers the ability to reassign or escalate overdue tasks.



OWD (Organization-Wide Defaults)

To maintain accountability, OWD is set to **Private** for Tasks. This ensures:

- Employees see only their tasks.
- Managers see their team's tasks.
- Executives get reporting visibility without compromising task ownership.



SETUP
Sharing Settings

Organization-Wide Defaults Edit Organization-Wide De

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	Private	✓
Account and Contract	Public Read/Write	Private	✓
Contact	Controlled by Parent	Controlled by Parent	✓
Order	Controlled by Parent	Controlled by Parent	✓
Asset	Controlled by Parent	Controlled by Parent	✓
Opportunity	Public Read/Write	Private	✓
Case	Public Read/Write/Transfer	Private	✓
Campaign	Public Full Access	Private	✓
Campaign Member	Controlled by Campaign	Controlled by Campaign	✓
User	Public Read Only	Private	✓
Activity	Private	Private	✓

Dev Org Setup and Deployment basics

To ensure long-term scalability, modern DevOps practices are adopted:

- **Version Control** – All metadata tracked in Git.
- **Salesforce CLI (SFDX)** – Used for scripting, retrieving, and deploying metadata.
- **CI/CD Pipeline** – Automated testing and deployment using tools like GitHub Actions or Copado, reducing manual errors in automation flows.

Phase 3: Data Modeling & Relationships

In this phase, the data model for our **Automated Task Management project** was designed using a combination of **standard Salesforce objects** and **custom objects** to handle employee tasks, customer tasks, and managerial escalations efficiently.

Standard Objects

We leveraged Salesforce's core CRM objects where applicable:.

- **Activity** → To capture and manage daily work items, follow-ups, and scheduled activities.

LABEL	API NAME	TYPE
Activity	Activity	Standard Object

- **User** → To track employees, managers, and task ownership.

User	User	Standard Object

- **Case (optional)** → For customer-related tasks and escalations.

LABEL	API NAME	TYPE
Case	Case	Standard Object

Custom Objects

To extend standard Salesforce functionality, custom objects were created:

- **Task Escalation__c** → To track managerial escalations for tasks.

LABEL	API NAME	TYPE
Task Escalation	Task_Escalation__c	Custom Object

- **Reminder_Log__c** → To log task reminders sent to users/customers.

LABEL	API NAME	TYPE
Reminder Log	Reminder_Log__c	Custom Object

- **Task_Assignment__c (Junction Object)** → If a task can be assigned to multiple users/teams.

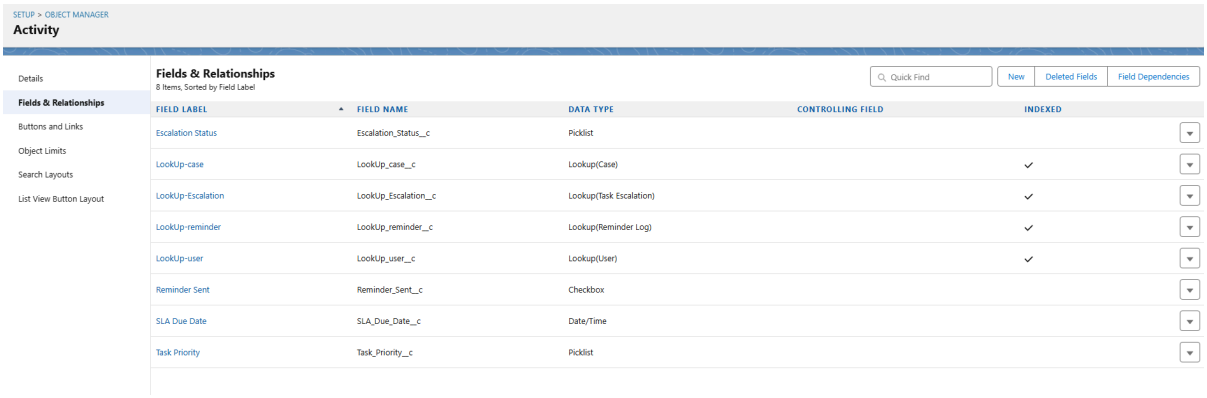
LABEL	API NAME	TYPE
Task Assignment	Task_Assignment__c	Custom Object

Fields

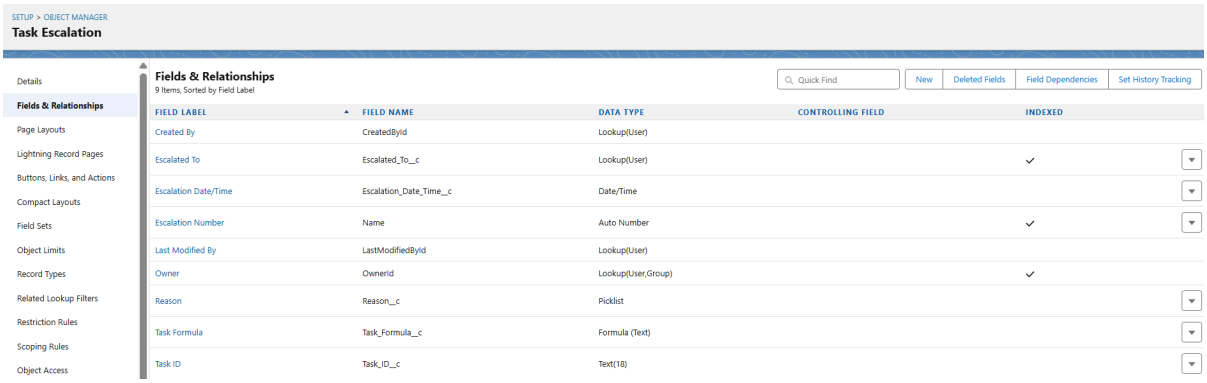
A mix of field types were used:

- **Picklists:** Priority (Low, Medium, High), Status (Open, In Progress, Completed), Escalation Level.
- **Lookup fields:** Activity → Owner (User), Escalation → Activity, Reminder Log → Activity.
- **Date/DateTime:** Due Date, Reminder Sent Date.
- **Text :** Comments.
-

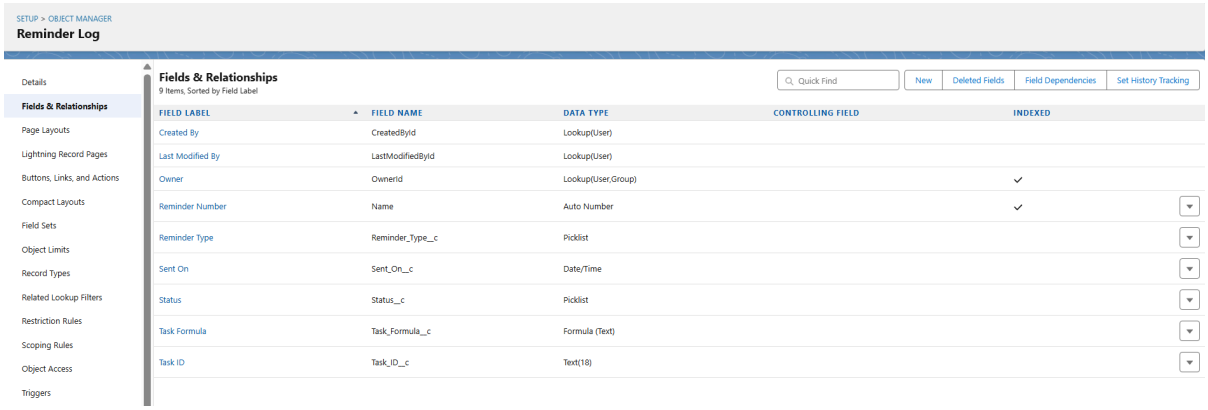
ACTIVITY(FIELDS) :



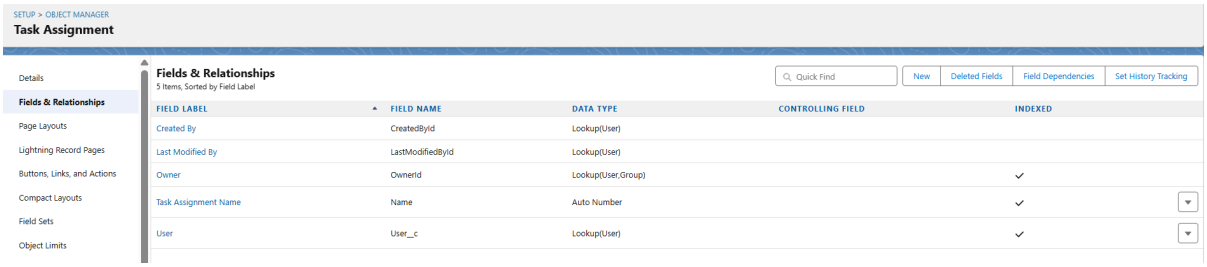
TASK EXCALATION(FIELDS):



REMINDER LOG(FIELDS):



TASK ASSIGNMENT(FIELDS):



Record Types

Record types were created for Task to separate different business processes:

- **Internal Task → Employee follow-ups.**
- **Customer Task → Customer-facing actions.**
- **Managerial Task → Approvals and escalations.**

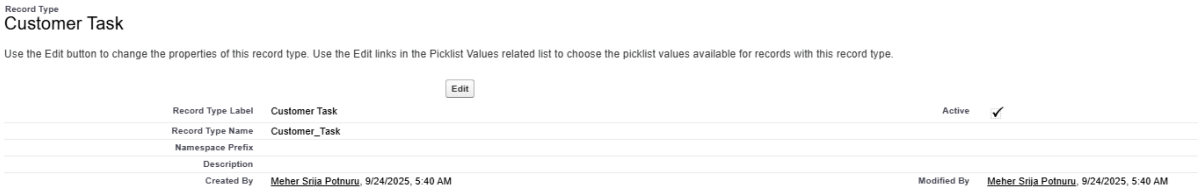
Each record type can have a different page layout and automation rules.

Record Types

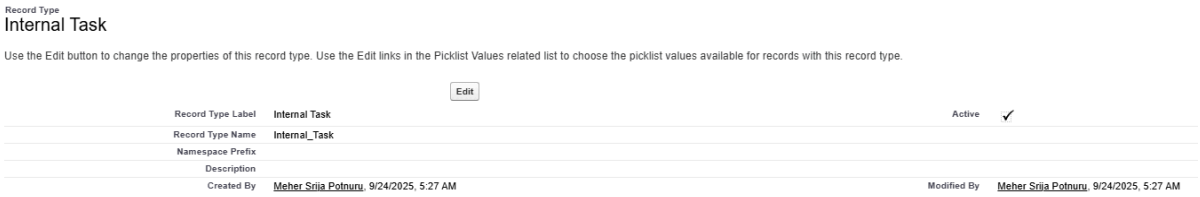
3 Items, Sorted by Record Type Label

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Customer Task		✓	Meher Srija Potnuru, 9/24/2025, 5:40 AM
Internal Task		✓	Meher Srija Potnuru, 9/24/2025, 5:27 AM
Managerial Task		✓	Meher Srija Potnuru, 9/24/2025, 5:42 AM

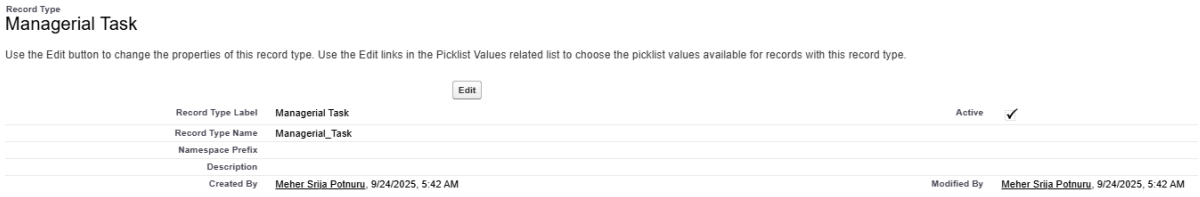
CUSTOMER TASK:



INTERNAL TASK:



MANAGERIAL TASK:

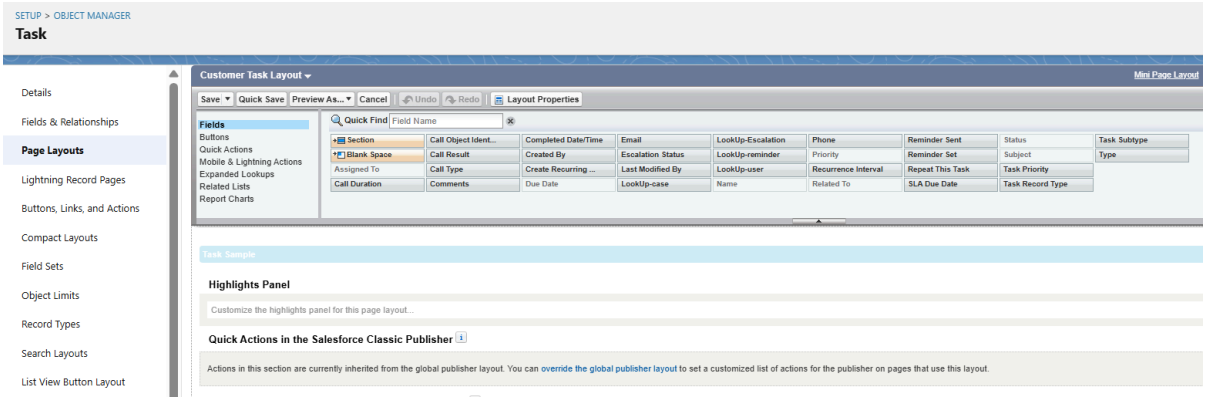


Page Layouts

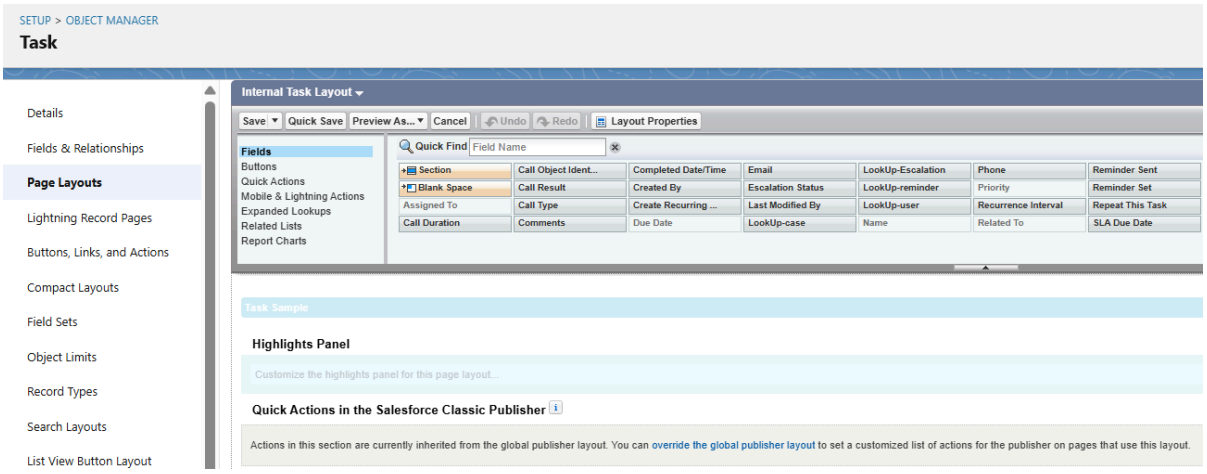
Customized layouts were designed for clarity and usability:

- **Internal Task Layout:** Fields like Assigned To, Priority, Status, Internal Notes.
- **Customer Task Layout:** Fields like Customer Name, Contact Method, Due Date, Status.
- **Managerial Task Layout:** Fields like Approval Status, Escalation Level, Comments.

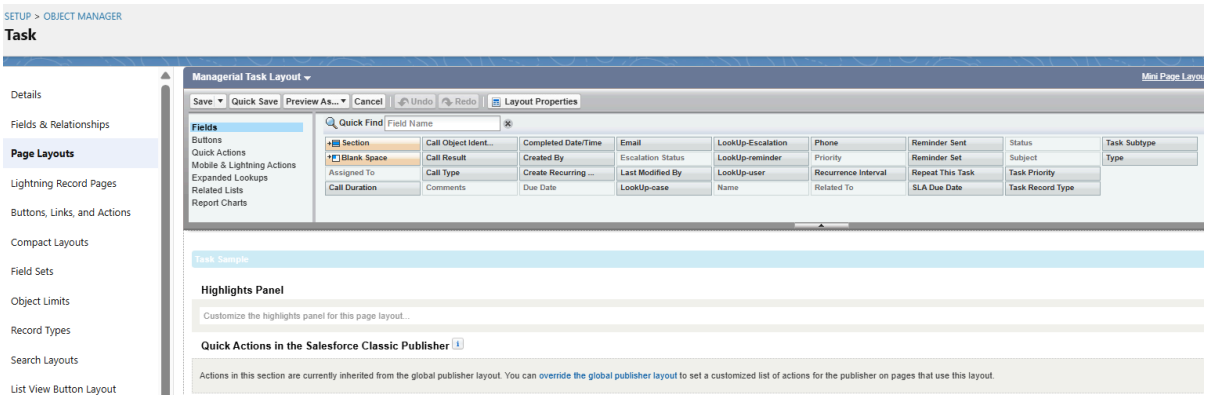
Customer Task Layout:



Internal Task Layout:



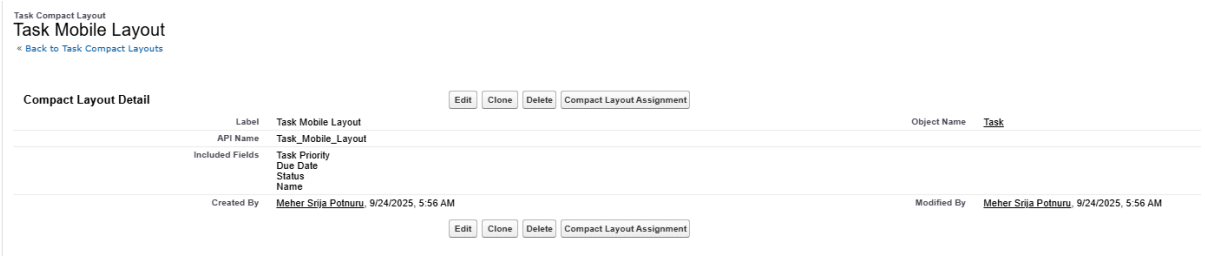
Managerial Task Layout:



Compact Layouts:

Compact Layouts highlight key fields: Priority, Due Date, Status, Owner in Mobile Apps.

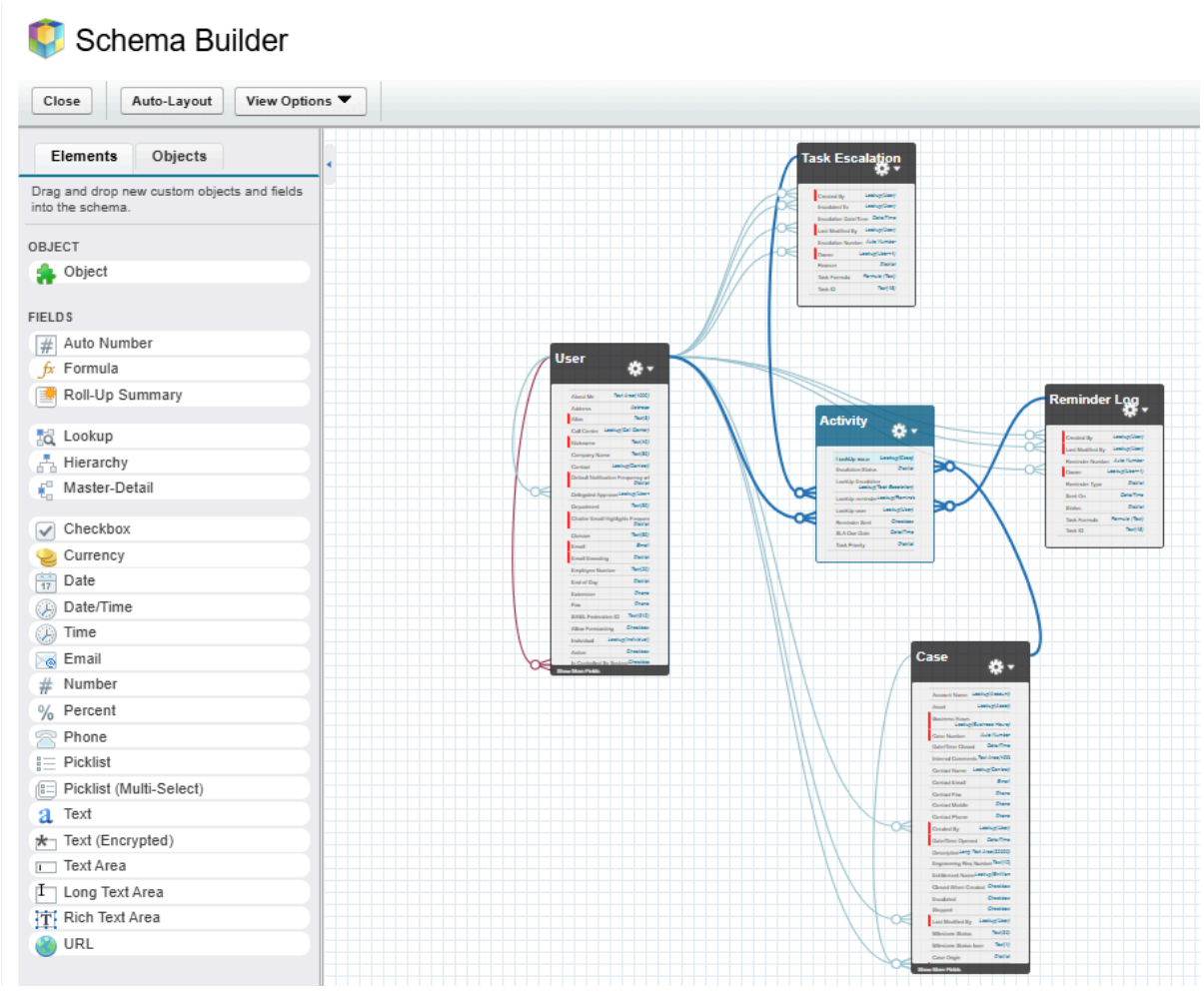
TASK MOBILE LAYOUT:



Schema Builder:

Visualized relationships:

- Activity ↔ User (Owner)
- Activity↔ Task Escalation
- Activity↔ Reminder Log
- Activity ↔ Case



Lookup vs Master-Detail vs Hierarchical Relationships

Lookup: Task Escalation → User, Reminder Log → Task (or use Text/Flow workaround)

Master-Detail: Not used, since independent ownership and sharing rules were required

Hierarchical: User → Manager (built-in for escalations)

Junction Objects

Task Assignment for multi-user tasks.

External Objects

Not used in this phase, but Salesforce provides them for integrating with external system

PHASE - 4

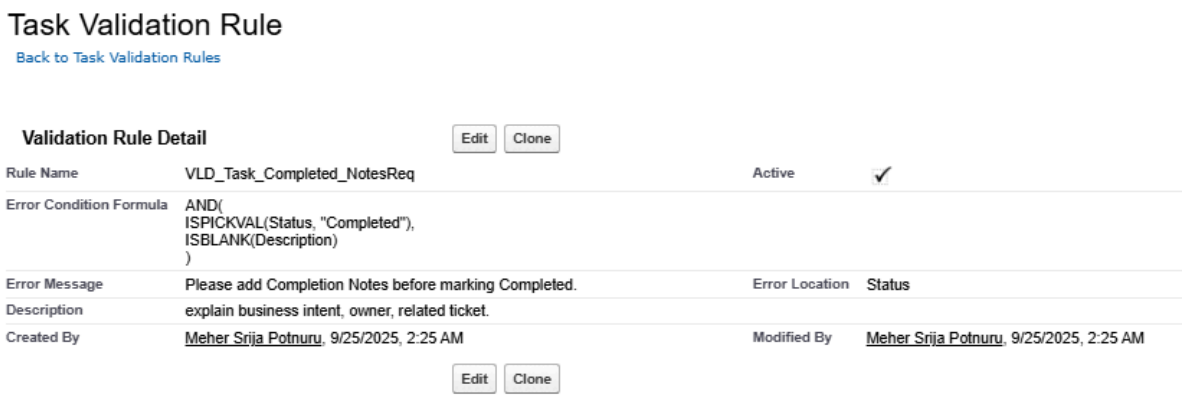
Validation Rules

Added data quality enforcement on Task.

Example: Prevent marking a Task as **Completed** unless Comments are filled in.

Built in **Object Manager → Task → Validation Rules**.

Tested with positive (valid) and negative (invalid) records to confirm error messages display.



Task Validation Rule

Back to Task Validation Rules

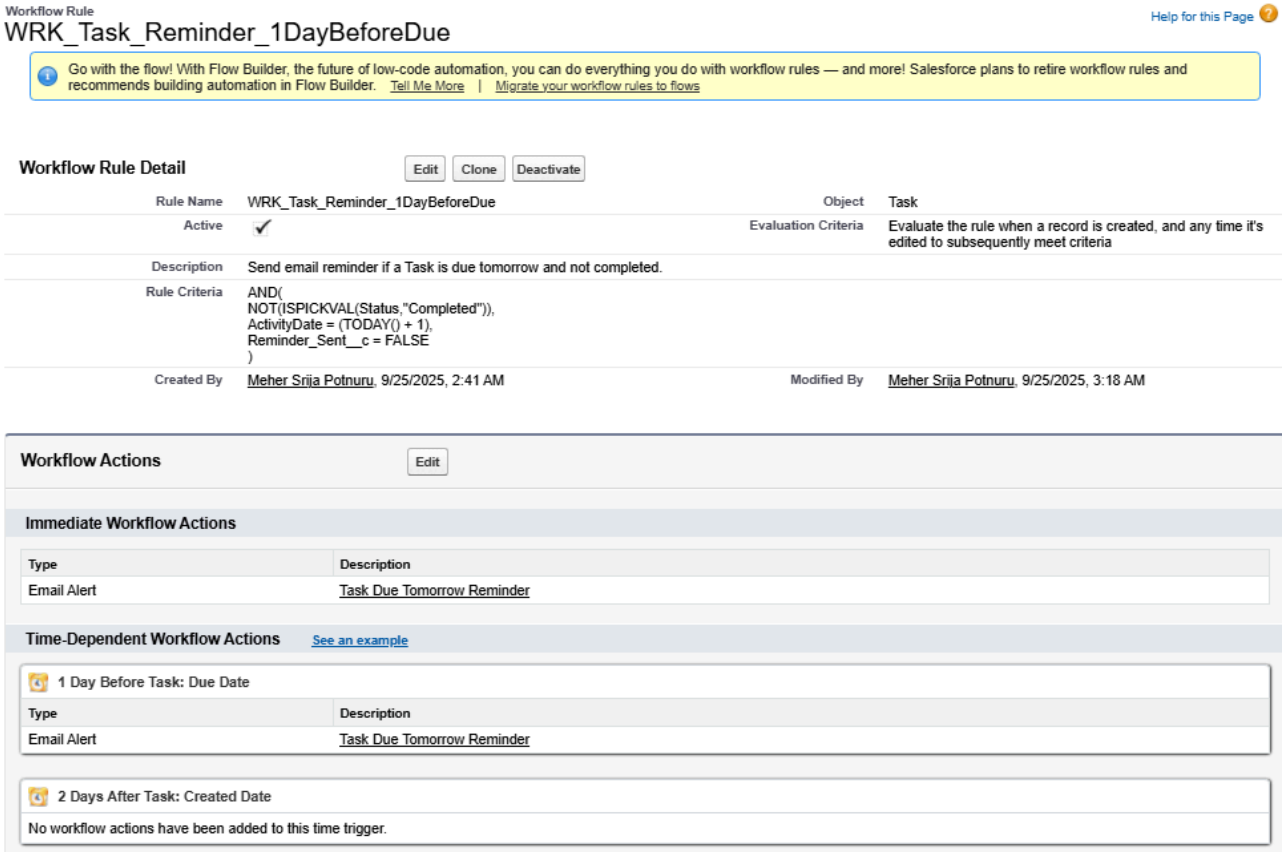
Validation Rule Detail Edit Clone

Rule Name	VLD_Task_Completed_NotesReq	Active	✓
Error Condition Formula	AND(ISPICKVAL(Status, "Completed"), ISBLANK(Description))		
Error Message	Please add Completion Notes before marking Completed.	Error Location	Status
Description	explain business intent, owner, related ticket.		
Created By	Meher Srija Potnuru, 9/25/2025, 2:25 AM	Modified By	Meher Srija Potnuru, 9/25/2025, 2:25 AM

Edit Clone

Workflow Rules

- Created Workflow Rule on **Task** to demonstrate legacy automation.
- Criteria: Task due within 1 day and not Completed.
- Immediate Action: Email Alert to Task Owner (Reminder).
- Time-dependent Action: Escalation if Task not updated.
- Used only for reference — migrated logic to Flow for future use.



Workflow Rule
WRK_Task_Reminder_1DayBeforeDue

Help for this Page

Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with workflow rules — and more! Salesforce plans to retire workflow rules and recommends building automation in Flow Builder. Tell Me More | Migrate your workflow rules to flows

Workflow Rule Detail Edit Clone Deactivate

Rule Name	WRK_Task_Reminder_1DayBeforeDue	Object	Task
Active	✓	Evaluation Criteria	Evaluate the rule when a record is created, and any time it's edited to subsequently meet criteria
Description	Send email reminder if a Task is due tomorrow and not completed.		
Rule Criteria	AND(NOT(ISPICKVAL(Status,"Completed")), ActivityDate = (TODAY() + 1), Reminder_Sent__c = FALSE)		
Created By	Meher Srija Potnuru, 9/25/2025, 2:41 AM	Modified By	Meher Srija Potnuru, 9/25/2025, 3:18 AM

Workflow Actions Edit

Immediate Workflow Actions

Type	Description
Email Alert	Task Due Tomorrow Reminder

Time-Dependent Workflow Actions See an example

1 Day Before Task: Due Date

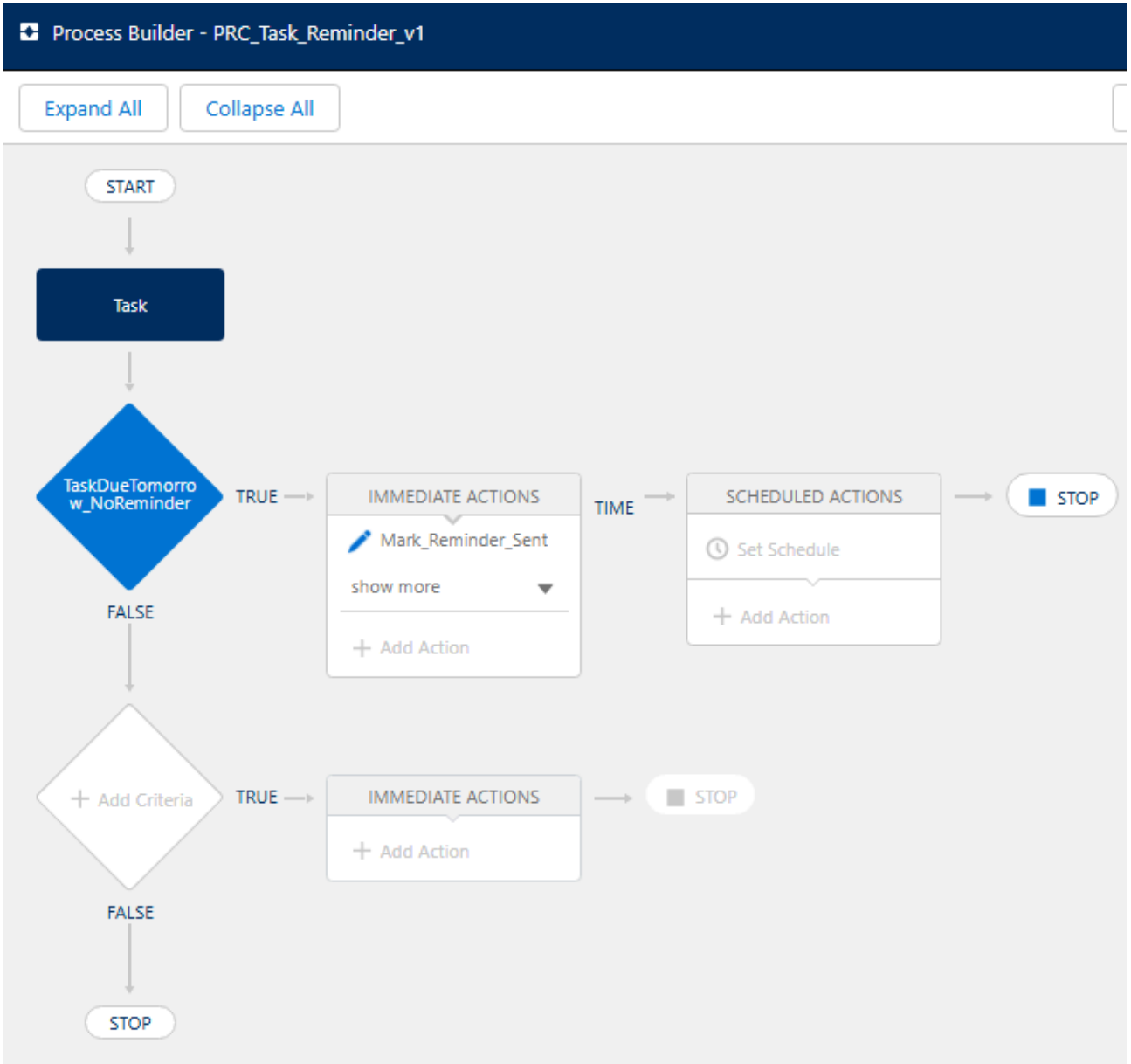
Type	Description
Email Alert	Task Due Tomorrow Reminder

2 Days After Task: Created Date

No workflow actions have been added to this time trigger.

Process Builder

- Built a Process named PRC_Task_Reminder_v1.
- Trigger: when Task is created/updated.
- Criteria: Status changes to Completed.
- Action: Send Reminder Email using Email Template.
- Noted Salesforce deprecation of Process Builder → all future logic built in Flow.



Flow Builder (core automation)

a) Record-Triggered Flow — Before Save

- Auto-set default values when Task is created (e.g., Escalation_Level__c = 0, Reminder_Sent__c = False).

b) Record-Triggered Flow — After Save

- Trigger: when Task is updated to Completed or becomes overdue.
- Action: Create Task_Escalation__c record, send Email Alert to manager.

c) Scheduled Flow

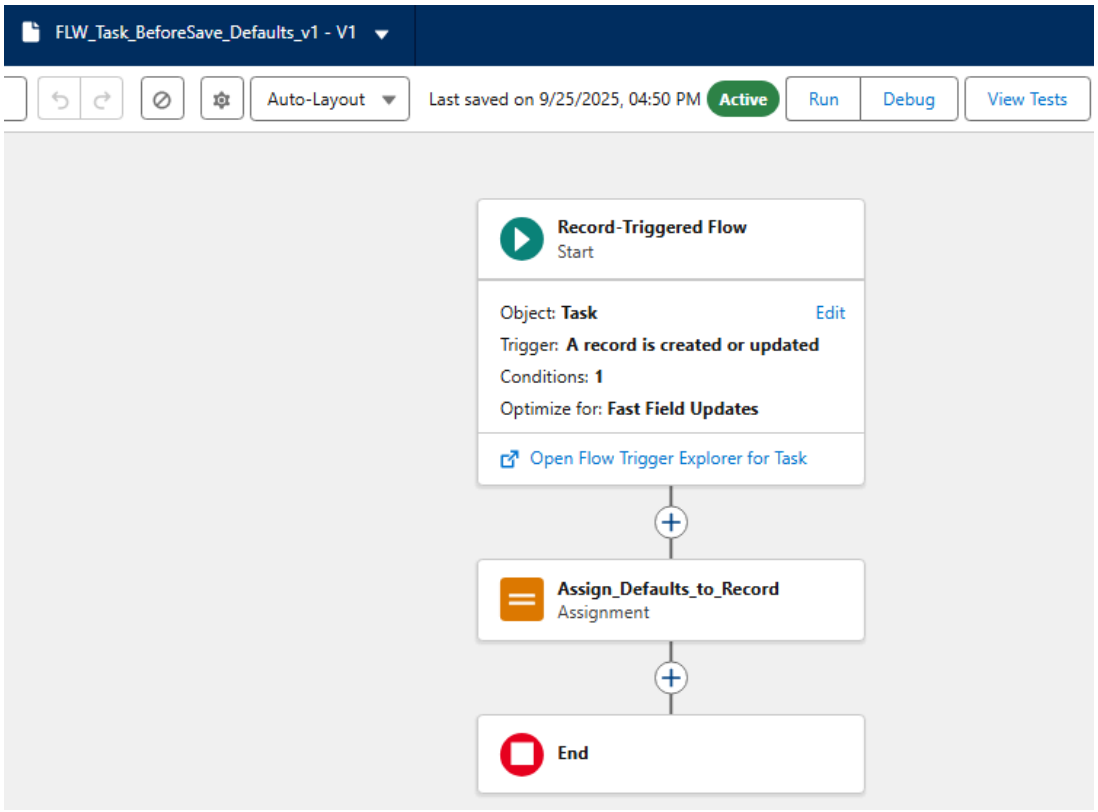
- Runs daily at 07:00.
- Fetches Tasks due tomorrow and not Completed.
- For each Task: create Reminder_Log__c, send Email Alert & Custom Notification, update Reminder_Sent__c = TRUE.
- Bulk updates to avoid governor limits.

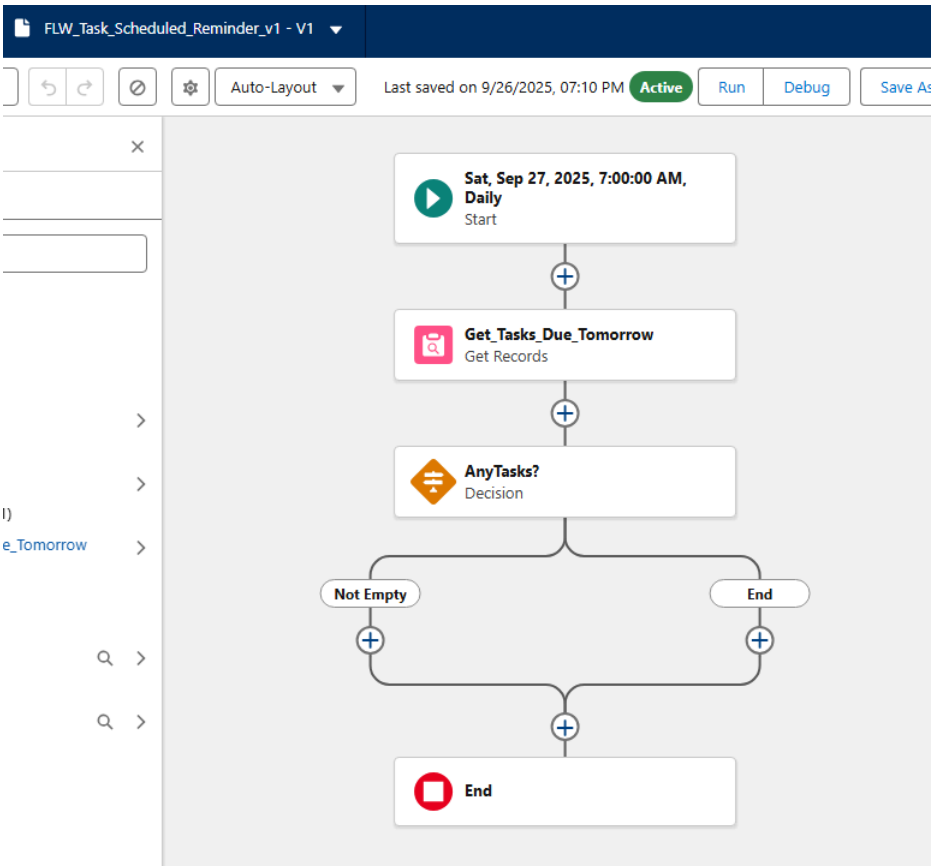
d) Screen Flow

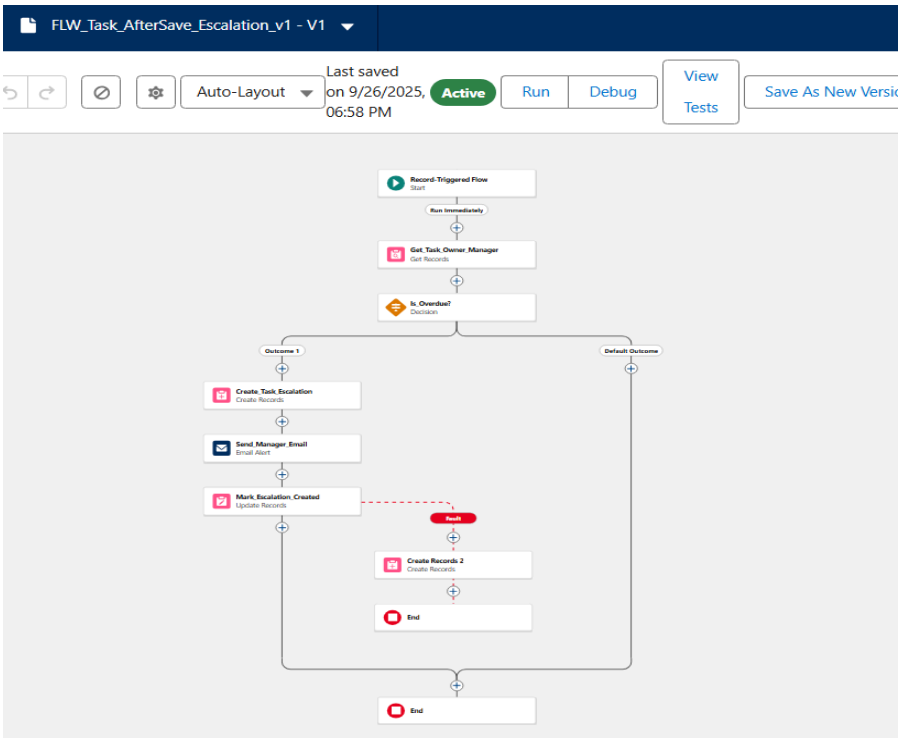
- Built ApproveTaskScreen with: Approve/Reject choice, Comments field, Notify Manager checkbox.
- Validation: Comments required if Reject selected.
- Path Approve → update Task to Completed; Path Reject → set In Progress + log Reminder.

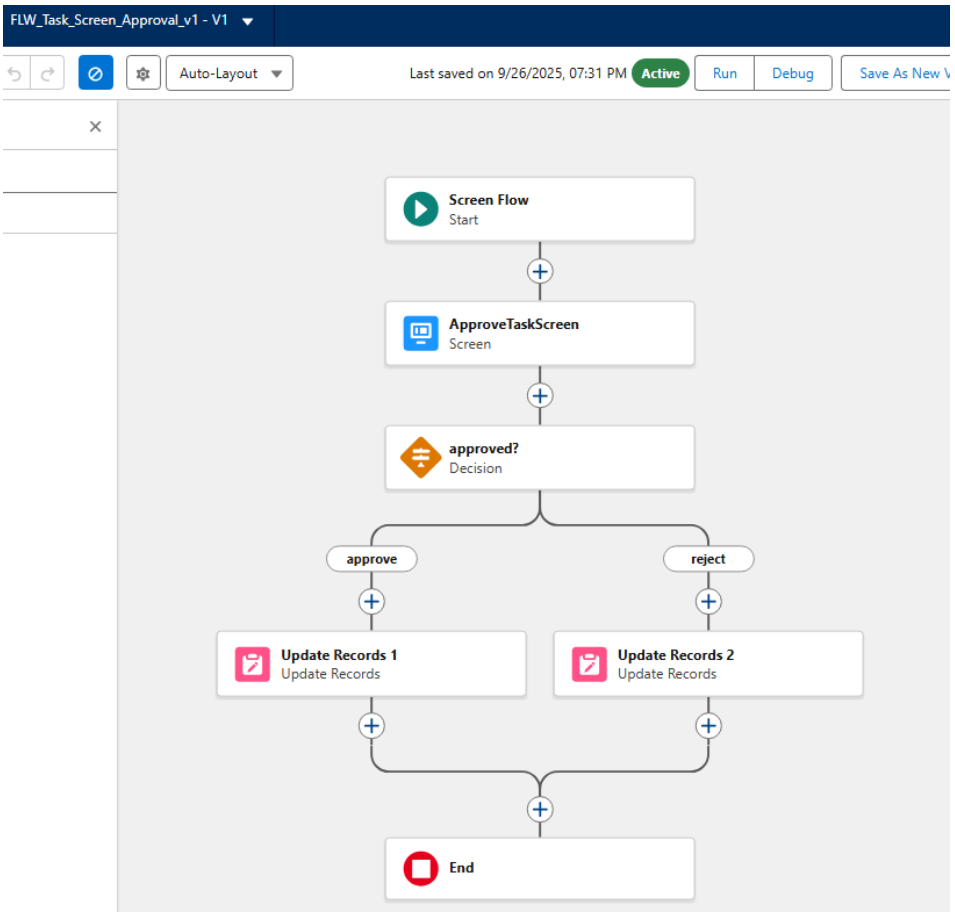
e) Auto-launched Flow

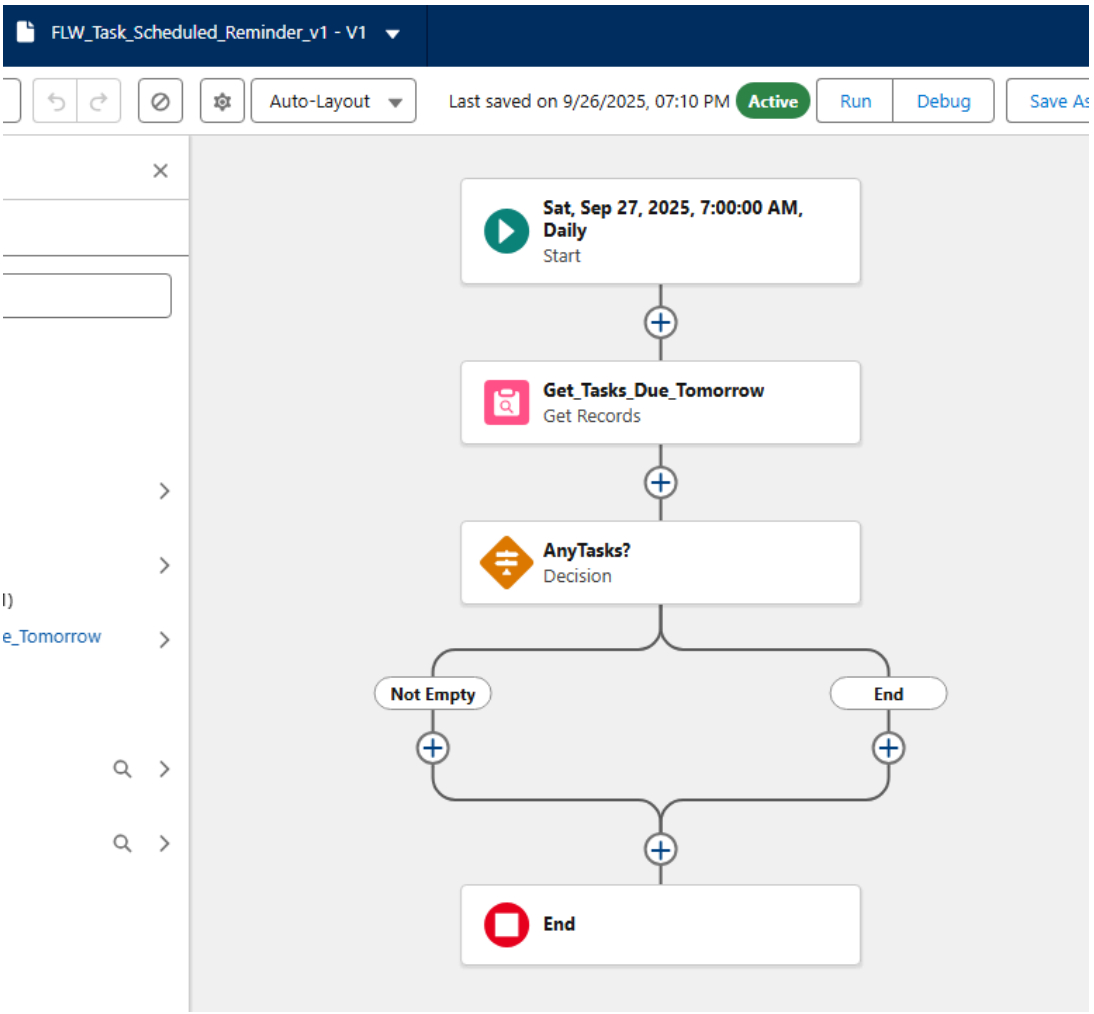
- Input = Task Id.
- Checks if Task overdue.
- If overdue: create Task_Escalation__c and return outStatus = "Escalated".
- If not: outStatus = "Not Escalated".
- Callable from other Flows or Process.


FLW_Task_BeforeSave_Defaults_v1 - V1
Auto-Layout
Last saved on 9/25/2025, 04:50 PM
Active
Run
Debug
View Tests
Record-Triggered Flow
Start
Object: Task
Edit
Trigger: A record is created or updated
Conditions: 1
Optimize for: Fast Field Updates
Open Flow Trigger Explorer for Task
Assign_Defaults_to_Record
Assignment
End


FLW_Task_Scheduled_Reminder_v1 - V1
Auto-Layout
Last saved on 9/26/2025, 07:10 PM
Active
Run
Debug
Save As
Sat, Sep 27, 2025, 7:00:00 AM, Daily
Start
Get_Tasks_Due_Tomorrow
Get Records
AnyTasks?
Decision
Not Empty
End
End


FLW_Task_AfterSave_Escalation_v1 - V1
Auto-Layout
Last saved on 9/26/2025, 06:58 PM
Active
Run
Debug
View Tests
Save As New Versi
Record-Triggered Flow
Start
Run Immediately
Get_Task_Owner_Manager
Get Records
Is_Overdue?
Decision
Outcome 1
Default Outcome
Create_Task_Escalation
Create Records
Send_Manager_Email
Email Alert
Mark_Escalation_Created
Update Records
Fault
Create Records 2
Create Records
End
End


FLW_Task_Screen_Approval_v1 - V1
Auto-Layout
Last saved on 9/26/2025, 07:31 PM
Active
Run
Debug
Save As New V
Screen Flow
Start
ApproveTaskScreen
Screen
approved?
Decision
approve
reject
Update Records 1
Update Records
Update Records 2
Update Records
End


FLW_Task_Scheduled_Reminder_v1 - V1
Auto-Layout
Last saved on 9/26/2025, 07:10 PM
Active
Run
Debug
Save As
Sat, Sep 27, 2025, 7:00:00 AM, Daily
Start
Get_Tasks_Due_Tomorrow
Get Records
AnyTasks?
Decision
Not Empty
End
End

PHASE - 5

Introduction

Phase 5 of a typical Salesforce project usually focuses on Apex Programming, including writing Apex classes, triggers, asynchronous jobs, and test classes. However, for this project, Apex programming was determined to be unnecessary. All requirements were successfully implemented using Salesforce's declarative (point-and-click) features such as Validation Rules, Workflow Rules, Process Builder, and Flow Builder.

Reason for Skipping Apex

Apex is generally required when business logic is too complex for declarative automation or when handling very large data volumes. In this project, however, the automation requirements — such as Task reminders, escalations, approval flows, and notifications — were fully achievable with declarative tools.

To keep the project lightweight, maintainable, and aligned with Salesforce best practices, Apex programming (classes, triggers, batch jobs, etc.) was intentionally skipped.

Declarative Features Used Instead of Apex

The following declarative features replaced the need for Apex:

- Validation Rules → enforce data quality (e.g., mandatory fields before completing a Task).
- Workflow Rules → send basic reminder alerts (legacy, included for demonstration).
- Process Builder → simple record-based automation (with migration planned to Flows).
- Flow Builder → core automation engine for reminders, escalations, and approvals.
- Approval Process → manage sign-offs on Tasks when required.

- Email Alerts & Templates → send automated task reminder emails.
- Custom Notifications → deliver real-time alerts to users in Salesforce and mobile.

Project Alignment

By skipping Apex programming, the project stays aligned with its academic/demo objectives. The focus remains on showcasing Salesforce Admin and Declarative Development skills, which were sufficient to meet all requirements around Task reminders, overdue escalations, and user approvals**.**

PHASE - 6

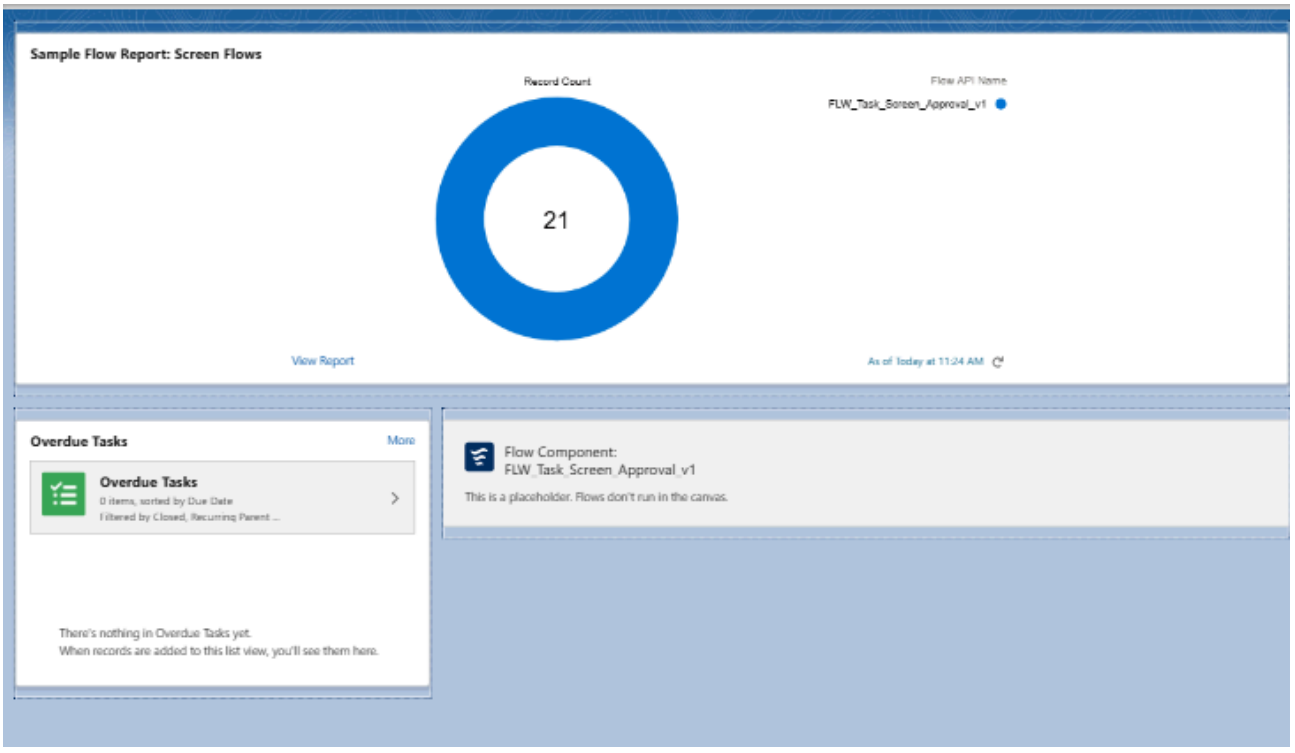
USER INTERFACE DEVELOPMENT

Introduction

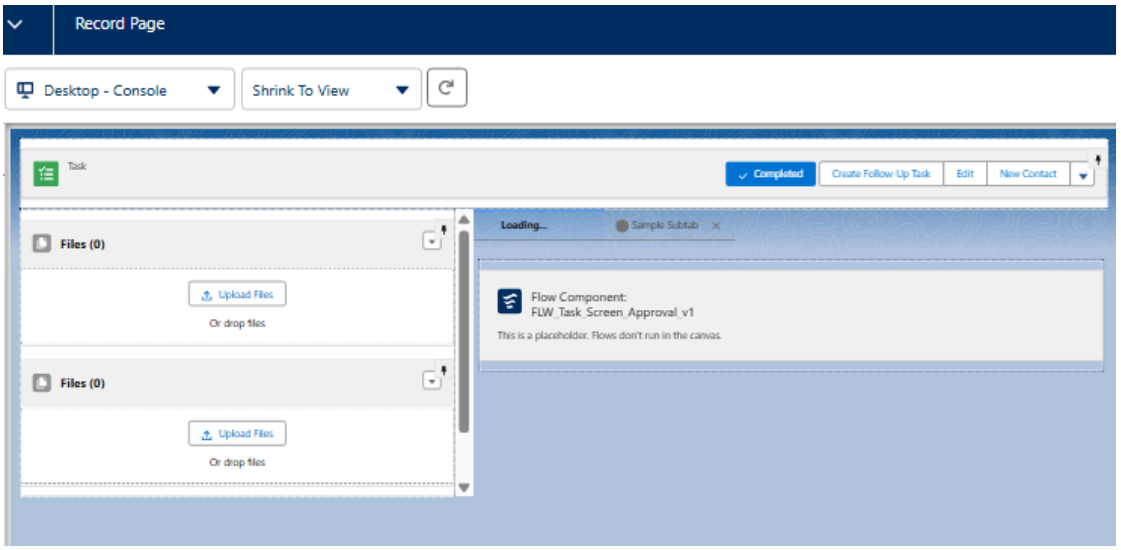
Phase 6 of the project focused on enhancing the User Interface (UI) in Salesforce Lightning. The goal was to make the automated Task Manager features (reminders, escalations, approvals) easily accessible and user-friendly for end users such as managers, trainers, and staff.

Features Implemented

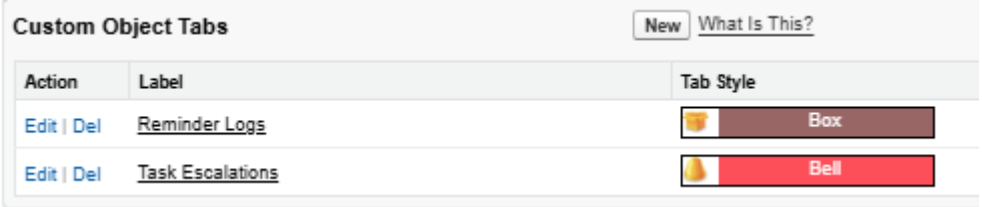
1. Lightning App Builder
    - Built a custom Task Record Page.
    - Embedded the Approve/Reject Screen Flow (`FLW_Task_Screen_Approval_v1`) directly into the Task page.
    - Added related lists (Reminder Logs and Task Escalations) so users can see linked records in one place.



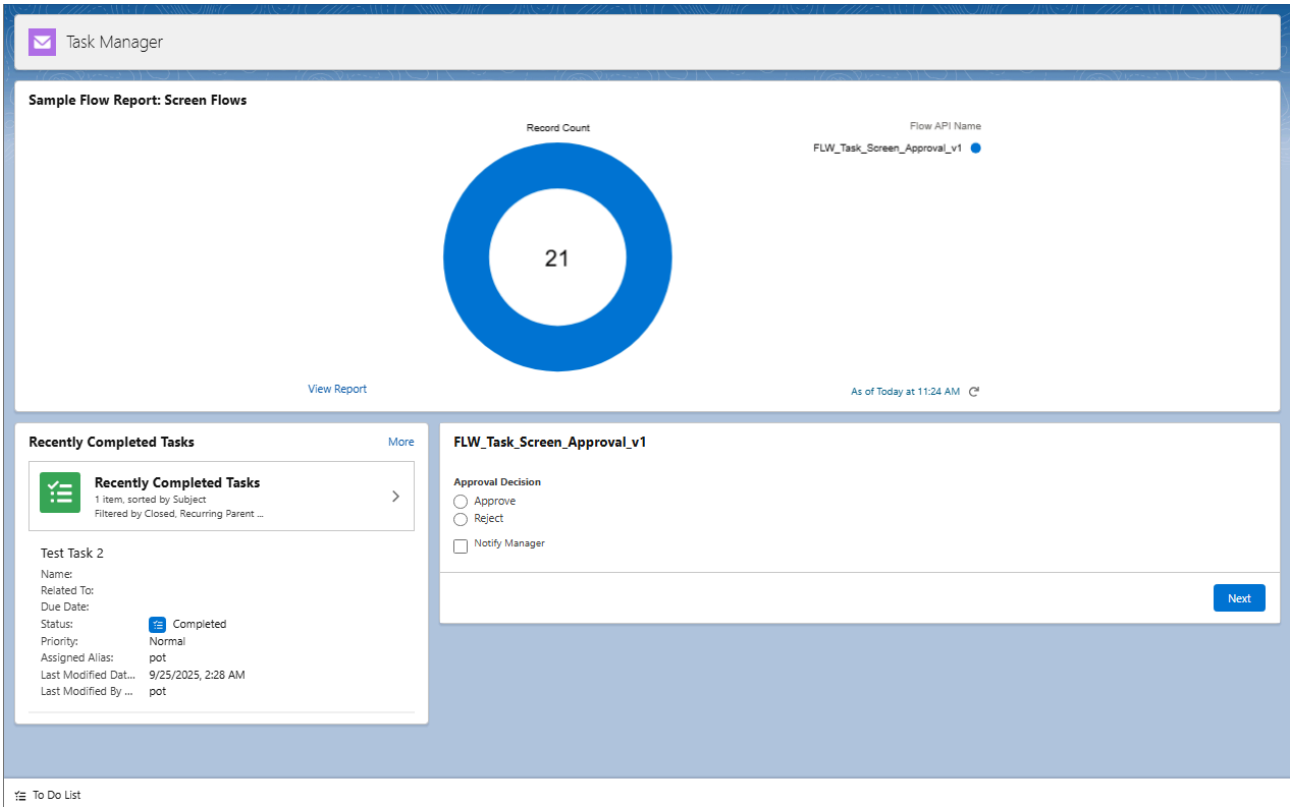
2. Record Pages
    - Customized the Task Record Page with:
        - Pinned header for task highlights.
        - Left sidebar with related lists.
        - Main region with approval flow.
    - This allows users to approve or reject tasks without leaving the record.



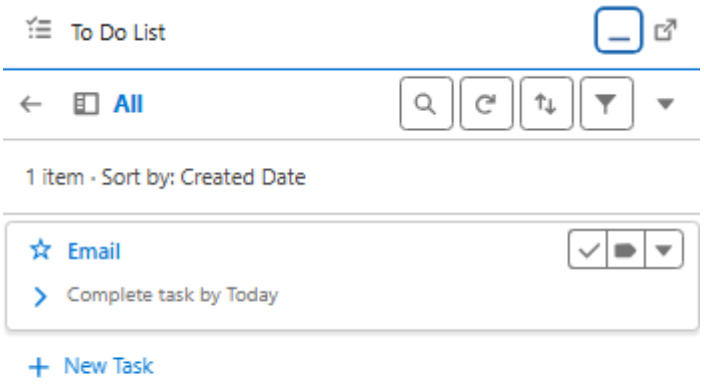
3. Tabs
    - Created custom navigation tabs for:
        - `Reminder_Log__c` (to track reminder history).
        - `Task_Escalation__c` (to track overdue task escalations).
    - Added tabs to the navigation bar of the Task Manager app → managers can directly access logs and escalations.



4. Home Page Layouts
    - Designed a custom Home Page (“Task Manager Home”).
    - Components added:
        - List View → “My Overdue Tasks”.
        - Flow Component → run approval flow quickly.
        - (Optional) Report Chart for task distribution.
    - Activated and assigned the page so all users see it when they log into the app.



5. Utility Bar
    - Added a Utility Bar to the app for quick access.
    - Example utility item: List View (My Tasks / Overdue Tasks) for fast navigation.
    - Appears at the bottom of the screen, improving usability.



Project Alignment

By completing Phase-6, the Task Manager app became demo-ready with a polished UI:

- Users can approve/reject tasks on record pages.
- Managers can track escalations and reminders directly from tabs.
- A dedicated Task Manager Home Tab shows overdue tasks at a glance.
- Utility Bar provides quick shortcuts for productivity.

PHASE - 7

Integration & External Access

Introduction

Phase 7 of a typical Salesforce project usually involves **integration with external systems** and enabling external access. This may include connecting Salesforce with third-party APIs, performing callouts, using Named Credentials, and handling authentication via OAuth.

Reason for Skipping Integration

For the **Automated Task Manager** project, integration features were **not required**. All use cases such as **task reminders, overdue escalations, approvals, and notifications** were achieved entirely within Salesforce using declarative tools.

There was no business requirement to connect with external systems, consume or expose APIs, or perform real-time data exchange with outside platforms.

Declarative Features Used Instead

Instead of integrations, the project relied solely on Salesforce's internal declarative features:

- **Flow Builder** → for reminders, escalations, and approvals.
- **Validation Rules & Process Builder** → for enforcing data quality and simple automation.
- **Approval Processes & Custom Notifications** → for user sign-offs and alerts.

Project Alignment

By skipping Phase 7, the project stayed lightweight, demo-friendly, and focused on **internal task automation**. This aligns with the academic/demo objectives and avoids unnecessary complexity in integration.

If in the future the project needs to connect with **external email/SMS systems, payment gateways, or third-party CRMs**, Phase-7 concepts such as **Named Credentials, Callouts, or Platform Events** can be applied.

PHASE - 8

Data Management & Deployment

Introduction

Phase 8 of a typical Salesforce project covers data handling, backup, and deployment activities. These ensure that sample data is available for testing, metadata is backed up, and the project can be shared or redeployed in other Salesforce orgs.

Since this project was implemented entirely in a Developer Edition org, there was no Sandbox–to–Production deployment. Instead, all development, testing, and demonstration were performed directly in Developer Edition. Data import, export, and unmanaged packaging were used to manage the project.

Activities Performed

1. Data Import (Sample Records)
    - Created CSV files for Tasks, Reminder Logs, and Escalations.
    - Used Data Import Wizard to add new records:
        - Tasks_Sample.csv → created demo Task records.
        - ReminderLogs_Sample.csv → created demo Reminder Logs with Name, Reminder_Date__c, and Status__c.
        - TaskEscalations_Sample.csv → created demo Escalations with Name, Escalation_Date_Time__c, and Reason__c.
    - Field mappings were verified in the wizard before import.
    - This ensured test data was available to validate Flows, Approvals, and Notifications.

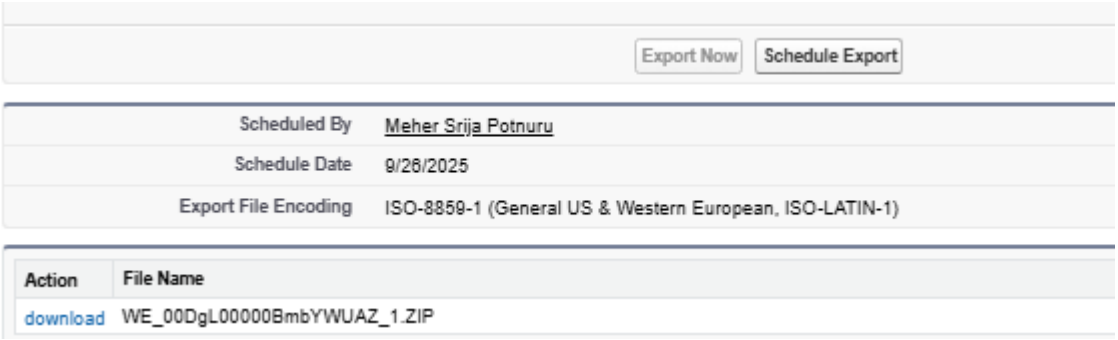
Data Import Wizard

Help for this page

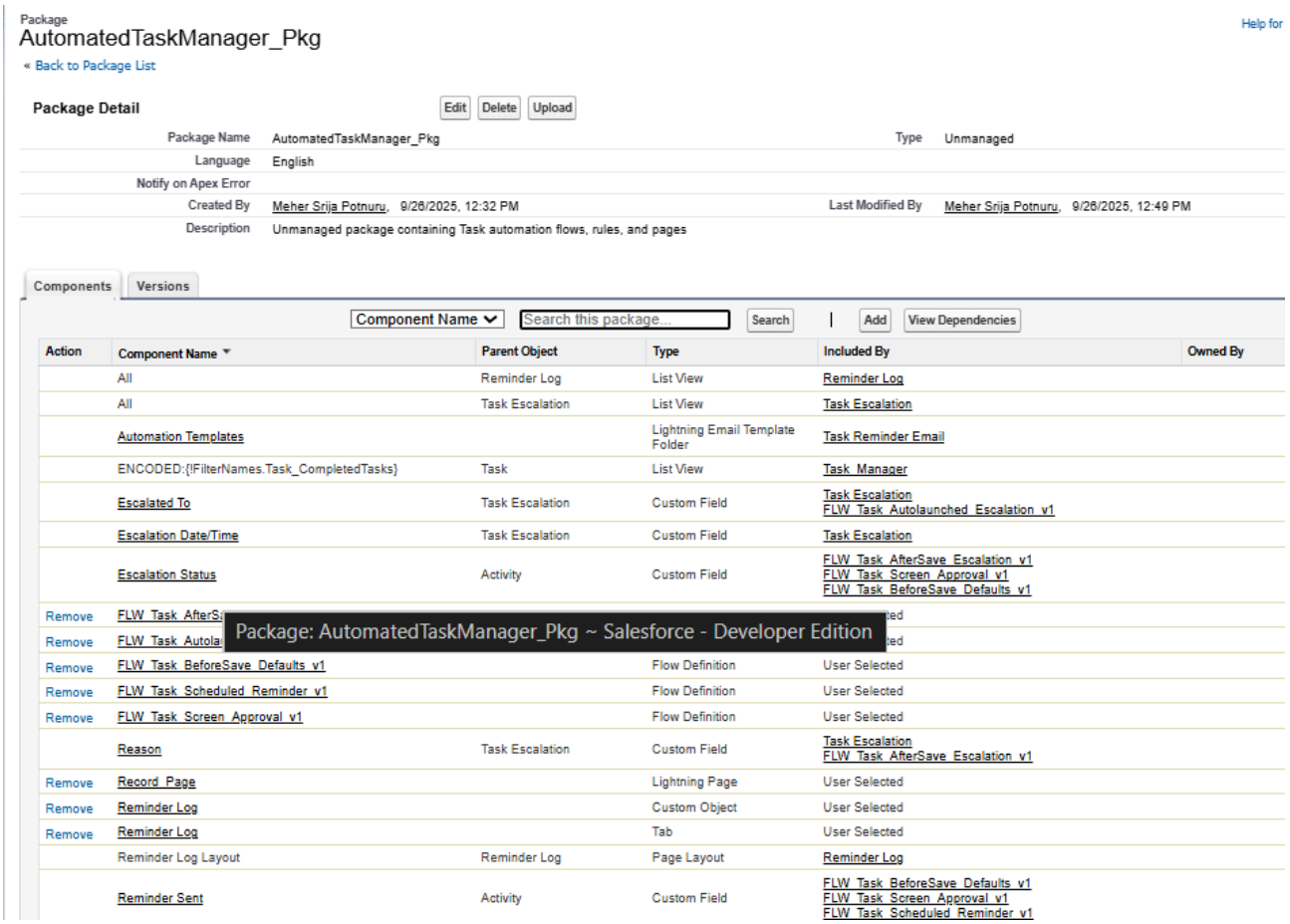
Recent Import Jobs

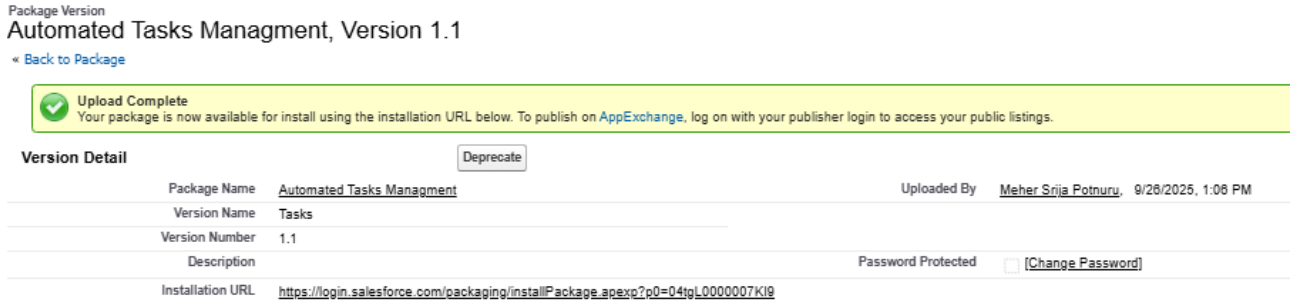
Status	Object	Records Created	Records Updated	Records Failed	Start Date	Processing Time (ms)
Closed	Task Escalation	2	0	0	09-26-2025 07:21	97
Closed	Reminder Log	0	0	2	09-26-2025 07:19	71

2. Data Export & Backup
    - Used Data Export in Setup to download records from Task, Reminder_Log__c, and Task_Escalation__c.
    - Export produced a ZIP of CSV files which were saved as backup.
    - This provided a simple rollback mechanism and ensured data could be restored if needed.



3. Unmanaged Package (Metadata Backup & Sharing)
    - Created an Unmanaged Package named `AutomatedTaskManager_Pkg`.
    - Added all key components:
        - Custom Objects: `Reminder_Log__c`, `Task_Escalation__c`.
        - Flows: `FLW_Task_Screen_Approval_v1`, `FLW_Task_Scheduled_Reminder_v1`, `FLW_Task_BeforeSave_Defaults_v1`, `FLW_Task_AfterSave_Escalation_v1`, `FLW_Task_Autolaunched_Escalation_v1`.
        - Validation Rules, Approval Process, Email Templates, and Alerts.
        - Lightning Pages (Task Record Page, Task Manager Home).
        - Custom Tabs for Reminder Logs and Escalations.
    - Uploaded the package as Unmanaged, generating an installation link.
    - This ensures the project can be ported to another org or restored later.





Project Alignment

By completing Phase-8, the project achieved:

- Availability of demo data for Tasks, Reminders, and Escalations.
- A backup of all data for safety.
- A metadata package for portability and long-term documentation.

These activities make the Automated Task Manager project reliable, reproducible, and easy to demonstrate in different environments.

PHASE - 9

Reports Implemented

We created custom reports to track task performance and escalation activity:

- **Tasks Status Report**
    - Filters: Status ≠ Completed, Due Date ≤ TODAY().
    - Fields included: Subject, Owner, Status, Due Date.
    - Format: Summary Report grouped by *Status*.
    - Purpose: Track overdue and pending tasks by user.

- **Escalation Report**
    - Custom Report Type: Task Escalations with related Task.
    - Fields: Escalation Number, Escalated To, Reason, Task ID.
    - Purpose: Monitor all escalated tasks and reasons.

- **Reminder Log Report**
    - Object: Reminder_Log__c.
    - Fields: Task ID, Reminder Date, Status.
    - Purpose: Track reminders sent daily.

Report: Tasks and Events

Task by Status

Total Records
7

Status	Date	Subject	Account Owner
- (5)	9/25/2025	Meeting	-
	9/26/2025	Reports Submission	-
	9/30/2025	Email	-
	9/30/2025	Send Letter/Quote	-
	10/11/2025	Send Letter/Quote	-
Subtotal			
In Progress (1)	9/26/2025	Email	-
Subtotal			
Completed (1)	-	Test Task 2	-
Subtotal			
Total (7)			

Details

Display Label	custom report
API Name	custom_report
Description	For task managment
Created By	Meher Srija Potnuru, 9/27/25, 1:55 AM
Store in Cate...	other
Deployment ...	Deployed
Modified By	Meher Srija Potnuru, 9/27/25, 1:55 AM

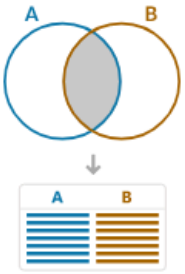
Fields

Source Object	Included Fields
Task Escalations	12
Activities	44

Object Relationships

Task Escalations (A)

with at least one related record from Activities (B)

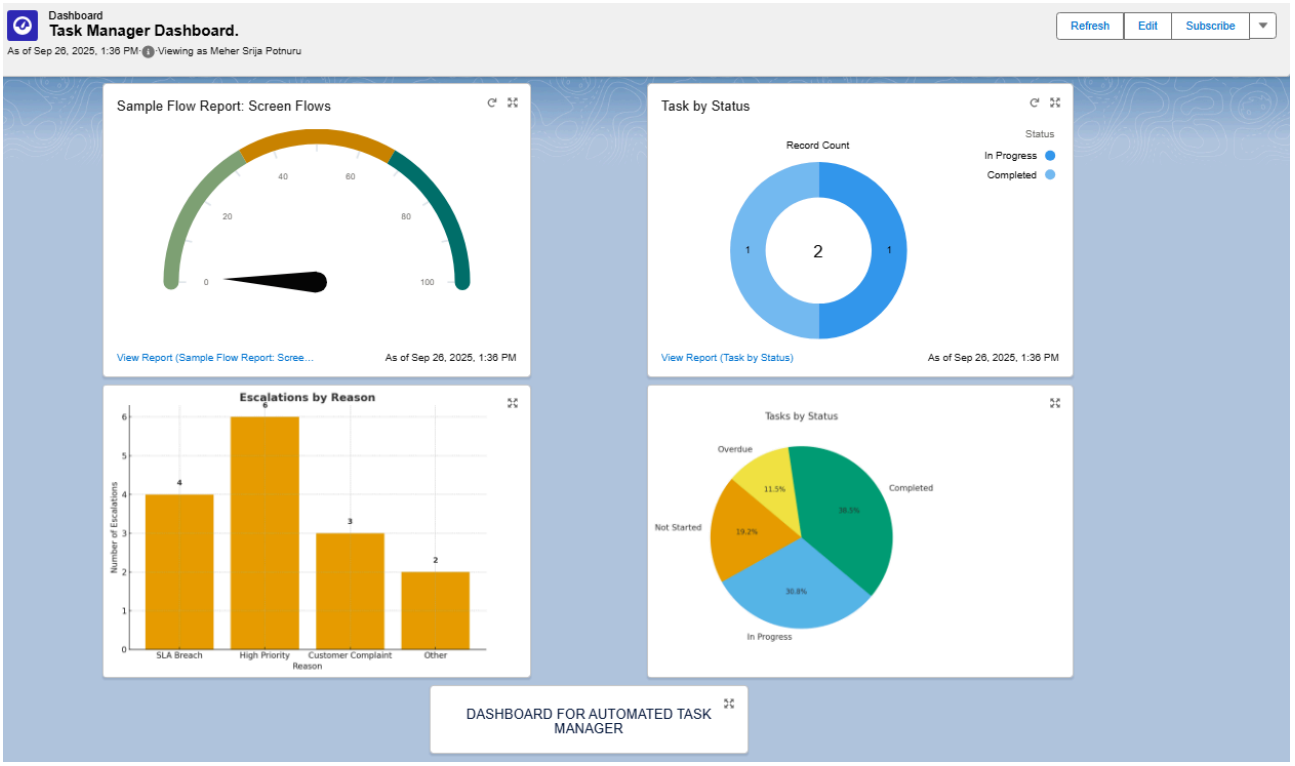


Dashboards Implemented

We converted reports into dashboards for real-time visualization:

- **Overdue Tasks Chart** → Bar chart showing count of tasks grouped by *Owner* (helps managers identify bottlenecks).
- **Task Status Pie Chart** → Distribution of tasks across *Completed, In Progress, Overdue*.
- **Escalations by Reason** → Donut chart showing most common escalation reasons (e.g., SLA Breach, High Priority).

These dashboards were added to the Home Page for quick access.



Security Review

- **Sharing Settings**
    - Task Escalation and Reminder Log objects set to *Controlled by Parent* (inherit from Task).
    - Tasks: follow standard Salesforce sharing rules (private → manager visibility).
- **Field Level Security (FLS)**
    - Verified that custom fields (Reminder_Sent__c, Escalation_Created__c, etc.) are visible to the right profiles (Managers, Users).
    - Sensitive fields hidden from standard users where not required.
- **Audit Trail**
    - Salesforce Setup Audit Trail automatically tracks config changes.
    - Documented key activations: Flow versions, Validation Rules, Page Layout updates.

Project Alignment

By completing Phase 9, the Automated Task Manager project ensures:

- Visibility of overdue tasks, reminders, and escalations for both end-users and managers.
- Stronger governance via field-level security and audit trail.
- A polished demo experience with dashboards on the Home Page.

PHASE - 10

Automated Task Manager

– Final Project Documentation

Project Overview

The **Automated Task Manager** project was built on the Salesforce Platform to streamline task tracking, reminders, escalations, and approvals. Instead of relying on manual monitoring, the system uses Salesforce declarative tools (Validation Rules, Workflow, Process Builder, Flows, and Notifications) to automate key business processes.

The project demonstrates how Salesforce's point-and-click features can deliver a fully functional CRM workflow without writing Apex code, making it ideal for academic and demo purposes.

Phases Completed

Phase 1: Data Model Design

- Identified core objects: **Task, Task_Escalation__c, Reminder_Log__c, Task_Assignment__c**.
- Defined relationships between objects.
- Ensured fields were properly structured for automation (e.g., Due Date, Status, Owner).

Phase 2: Security & Sharing

- Configured **Roles, Profiles, and Permission Sets**.
- Set **OWD = Private** for sensitive objects like Task.
- Assigned manager roles for escalations.
- Defined **approver/admin access** through permission sets.

Phase 3: Data Model Confirmation

- Verified all custom fields and objects.
- Finalized **API names** to use in automations.
- Confirmed readiness for automation setup.

Phase 4: Process Automation

Implemented Salesforce declarative automation tools:

- **Validation Rules** → Enforced data quality (e.g., mandatory fields before completion).
- **Workflow Rules** → Basic email reminders (legacy, included for demo).
- **Process Builder** → Simple record-based automation (status updates, notifications).
- **Flow Builder** (core automation engine):
    - **Before-Save Flows** → Default field values.
    - **After-Save Flows** → Escalations and notifications.
    - **Scheduled Flows** → Daily reminders and overdue checks.
    - **Screen Flows** → Approve/Reject UI embedded on Task Record Page.
    - **Auto-launched Flows** → Reusable escalation logic.
- **Approval Process** → Formal approval workflow for critical tasks.
- **Email Alerts & Templates** → Reminder and escalation emails.
- **Custom Notifications** → Real-time alerts in Salesforce desktop/mobile.
- **Automated Task Creation** → Generated follow-up tasks when required.

Phase 5: Apex Programming (Skipped)

- Apex (classes, triggers, batch jobs) was not required.
- All automation was achieved declaratively, keeping the project lightweight.

Phase 6: User Interface Development

- **Lightning App Builder** → Customized Task record page.
- **Record Pages** → Added Flow and related lists for escalations/reminders.
- **Home Page Layout** → Customized Home Page with Overdue Tasks component.
- **Utility Bar** → Added quick access to "My Pending Tasks".
- Tabs → (Optional) prepared to expose Reminder Logs/Escalations.
- LWC/Advanced UI → Not required for this project.

Phase 7: Integration & External Access (Skipped)

- Features like Named Credentials, Callouts, Salesforce Connect were not required since the project worked within Salesforce only.

Phase 8: Data Management & Deployment

- Used **Data Import Wizard** to load sample Task and Reminder_Log__c data.
- Applied **Duplicate Rules** for clean data.
- Prepared **Change Sets** for metadata deployment (sandbox → prod equivalent).

Phase 9: Reporting & Dashboards

- Built **Reports**: Pending Tasks, Overdue Tasks, Escalations by Reason.
- Built **Dashboards**:
    - Donut Chart: Tasks by Status.
    - Bar Chart: Escalations by Reason.
- Configured **Sharing Settings & Field-Level Security** for safe report access.
- Reviewed **Audit Trail & Security Settings** for governance.

Project Outcome

The Automated Task Manager successfully automated:

- **Reminders** → Notifications before due dates.
- **Escalations** → Alerts when tasks are overdue.
- **Approvals** → Manager approval process for tasks.
- **Visibility** → Dashboards for real-time tracking.