

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.