

# Leave Tracker Management – Salesforce CRM Project

---

## Overview

The **Leave Tracker Management App** is a Salesforce CRM-based solution developed for organizations to track employee leave requests, balances, and approvals.

It helps organizations automate leave management, maintain centralized records, and generate real-time insights into leave usage and trends.

---

## Problem

Organizations face challenges in managing employee leaves:

- Manual leave application and tracking.
  - No centralized system to monitor leave balances and history.
  - Difficulty generating reports on leave trends and pending approvals.
  - Lack of automation in notifying managers and employees.
- 

## Solution

The **Leave Tracker Management App** provides a digital, automated solution leveraging Salesforce CRM features:

- **Custom Leave Objects** linked with Employee records.
- **Automation via Apex Triggers** to update leave balances and prevent overlapping leaves.
- **Lightning Web Component (LWC)** to apply for leaves, view history, and check balances.

- **Batch Apex & Scheduler** to auto-calculate leave accruals and send reminders.
  - **Reports & Dashboards** for managers and HR to analyze leave trends and approvals.
- 

## Objectives

- ✓ Maintain a centralized leave management system.
  - ✓ Automate leave status updates and balance calculations.
  - ✓ Provide managers with real-time visibility into leave requests.
  - ✓ Reduce manual intervention and improve efficiency.
  - ✓ Enhance transparency for employees and HR.
- 

## Stakeholders

- **Employees:** Apply for leave, track status, view balances.
  - **Managers:** Approve/reject leave requests, monitor team leave.
  - **HR/Admins:** Configure leave policies, manage leave data, generate reports.
- 

## How Salesforce CRM is Used

- **Custom Objects & Fields:** Leave Request object (Start Date, End Date, Leave Type, Status); Leave Policy object (Max Leaves, Accrual Rate).
- **Apex Trigger & Handler:** Automatically update leave balance and prevent overlapping leaves.
- **Lightning Web Component (LWC):** Submit leave requests, view balances, and history.
- **Batch Apex & Scheduler:** Auto-calculate leave accruals and send reminders for pending approvals.
- **Reports & Dashboards:** Visualize leave usage trends, pending approvals, and departmental summaries.

---

## Problem Statement

To develop a Salesforce-based CRM solution that:

- Tracks leave requests for employees.
  - Automatically updates leave balances and status.
  - Provides a user-friendly LWC interface for leave applications.
  - Implements automation using Apex (Triggers, Batch, Scheduler).
- 

## Tech Stack

- **Platform:** Salesforce CRM
  - **Development:** Apex, Lightning Web Components (LWC)
  - **Automation:** Triggers, Batch Apex, Schedulable Apex
  - **Analytics:** Reports & Dashboards
- 

## Expected Impact

- ✓ Streamlined leave management with centralized records.
  - ✓ Automated workflows → saving time for HR and managers.
  - ✓ Improved transparency and self-service for employees.
  - ✓ Scalable solution suitable for organizations of any size.
- 

## 10 Phases of the Project

### Phase 1: Problem Understanding & Industry Analysis

- Requirement gathering with HR, managers, and employees.

- Identify challenges like manual tracking, lack of automation, and reporting issues.
- Map leave application, approval, and tracking processes.

## **Phase 2: Org Setup & Configuration**

- Set up Salesforce Developer Org.
- Configure users, roles, profiles, and permission sets.
- Define working hours, holidays, org-wide defaults, and sharing rules.

## **Phase 3: Data Modeling & Relationships**

- Custom Objects: Leave Request, Leave Policy.
- Fields: Leave Type, Start/End Date, Status, Max Leaves, Accrual Rate.
- Relationships: Employee ↔ Leave Request, Leave Policy ↔ Leave Request.

## **Phase 4: Process Automation (Admin)**

- Validation rules: End date after start date, check leave balance.
- Approval process for leaves exceeding threshold.
- Flows to update leave balances and notify managers/employees.
- Email alerts and reminders for pending approvals.

## **Phase 5: Apex Programming (Developer)**

- Triggers to prevent overlapping leaves.
- Apex classes for reusable leave logic.
- Scheduled Apex for leave accrual and monthly summaries.
- Exception handling and test classes to ensure data integrity.

## **Phase 6: User Interface Development**

- Lightning App Builder → “Leave Tracker” app.

- Record pages for leave requests and policies.
- Tabs: Leave Requests, Policies, Reports.
- LWC for applying leave, checking balances, and viewing history.
- Dashboards for managers and HR.

### **Phase 7: Integration & External Access**

- REST/SOAP callouts to integrate with payroll or external holiday calendars.
- Platform Events to notify HR about critical leave overlaps.
- OAuth & authentication for external portals.

### **Phase 8: Data Management & Deployment**

- Data import wizard and Data Loader for bulk employee/leave data.
- Duplicate rules to avoid redundant records.
- Backups and deployment via Change Sets or VS Code/SFDX.

### **Phase 9: Reporting, Dashboards & Security Review**

- Reports: leave taken, pending approvals, leaves per department.
- Dashboards for managers and HR.
- Dynamic dashboards → employees see only their data.
- Field-level security for sensitive information and audit trails.

### **Phase 10: Final Presentation & Demo Day**

- Present the problem, solution, and benefits.
- Demo leave application, approval flow, notifications, and dashboards.
- Share documentation, setup guides, and user manuals.
- Showcase the project on LinkedIn/GitHub portfolio.

