

Leave Tracker App CRM Project

Problem Statement

In many organizations, managing employee leave requests and approvals is still handled through manual methods such as emails, spreadsheets, or paper forms. This leads to inefficiencies, lack of visibility, and communication gaps between employees, managers, and HR teams.

- **Employees** often struggle to track their leave balances, request history, and approval status.
- **Managers** face challenges in reviewing and approving leave requests promptly, which may cause delays and disrupt workforce planning.
- **HR teams** spend significant time consolidating leave data, ensuring policy compliance, and generating reports for payroll and management.

The absence of a centralized, automated system results in:

- Delayed approvals and miscommunication.
- Limited visibility into team availability and workforce planning.
- Difficulty in enforcing leave policies consistently.
- Increased administrative burden on HR.

To address these challenges, an automated **Leave Tracking System in Salesforce CRM** is required. This solution will streamline the entire leave management lifecycle—leave application, approval workflow, balance tracking, policy enforcement, and reporting—providing transparency, efficiency, and improved employee experience.

Use Cases for Leave Tracking App

1. Submit Leave Request

- **Actor:** Employee
 - **Description:** Employee logs into the Salesforce app and submits a leave request by selecting leave type (sick leave, vacation, casual, etc.), start and end dates, and reason.
 - **Outcome:** Leave request is saved and routed to the reporting manager for approval.
-

2. View Leave Balance

- **Actor:** Employee
 - **Description:** Employee checks available leave balance (earned, sick, casual, etc.) before submitting a request.
 - **Outcome:** Employee has visibility into their current leave entitlement and can plan accordingly.
-

3. Approve/Reject Leave Request

- **Actor:** Manager
Description: Manager receives a notification when an employee submits a leave request. Manager reviews the request and either approves or rejects it.
 - **Outcome:** Employee is notified of the decision, and records are updated.
-

4. View Team Leave Calendar

- **Actor:** Manager
 - **Description:** Manager views a calendar of team members' approved leaves to avoid scheduling conflicts.
 - **Outcome:** Helps in resource planning and workload management.
-

5. Track Leave History

- **Actor:** Employee
- **Description:** Employee views their past leave applications and statuses.
- **Outcome:** Provides transparency and helps employees track patterns.

6. Generate Leave Reports

- **Actor:** HR Administrator
 - **Description:** HR runs reports (monthly/quarterly/yearly) to analyze leave trends, absenteeism, and compliance with policies.
 - **Outcome:** Data supports payroll processing and workforce planning.
-

7. Define & Manage Leave Policies

- **Actor:** HR Administrator
 - **Description:** HR sets leave entitlements, carry-forward rules, and approval hierarchies in Salesforce.
 - **Outcome:** Policies are consistently applied across the organization.
-

8. Automated Notifications & Reminders

- **Actor:** System
- **Description:** Salesforce automatically sends notifications to managers when a leave request is submitted and reminders for pending approvals. Employees get notified of approval/rejection.
- **Outcome:** Reduces delays in the process.

Phase 1: Problem Understanding and Industry Analysis

1. Requirement Gathering

The requirements are divided into three categories:

- **Functional Requirements:**

- Employees can submit leave requests (type, duration, reason).
- Employees can view their leave balance and leave history.
- Managers can approve/reject leave requests with comments.
- Team leave calendar for managers to check overlapping leaves.
- HR can define leave policies (entitlements, carry forward, encashment rules).
- HR can generate reports on leave usage, trends, and compliance.
- System sends automated notifications and reminders.

- **Non-Functional Requirements:**

- Mobile-friendly interface for employees and managers.
- Role-based access control (Employees, Managers, HR, Admin).
- Scalability to handle growing workforce.
- Integration with Payroll/Attendance systems.

- **Reporting Requirements:**

- Leave usage by employee, department, or period.
- Trend analysis (sick leaves, absenteeism).
- Pending leave approvals.

2. Stakeholder Analysis

- **Employees** (apply for leave, check balances, track status).
- **Managers** (approve/reject leave, monitor team availability).
- **HR Team** (set leave policies, track compliance, generate reports).
- **Executives** (analyze absenteeism, plan workforce).
- **System Admins** (configure Salesforce app, manage customization).

3. Business Process Mapping

Step-by-step Workflow:

1. **Employee Submits Request** → Select leave type, dates, reason.
2. **System Validates** → Checks leave balance, policy compliance.
3. **Manager Approval Workflow** → Manager reviews and approves/rejects.
4. **Notifications Triggered** → Employee notified; HR updated.
5. **Leave Balance Updates** → Automatically deducts approved leave.
6. **Reporting & Payroll Integration** → HR pulls reports and aligns with payroll.

4. Industry-Specific Use Case Analysis

- **IT Services / Consulting:** Project managers need visibility into resource availability for client delivery timelines.
- **Healthcare:** Staffing schedules must ensure minimum workforce coverage, especially during critical shifts.
- **Manufacturing:** Shift supervisors need leave data to maintain production line continuity.
- **Retail:** Store managers need leave tracking to avoid understaffing during peak sales seasons.
- **Education:** Academic institutions require structured leave policies for faculty and staff to avoid academic disruptions.

The Salesforce-based leave tracking app ensures adaptability across industries by allowing configurable policies, custom workflows, and role-based access.

5. AppExchange Exploration

Before building from scratch, exploring **Salesforce AppExchange** provides insights into existing solutions and accelerators:

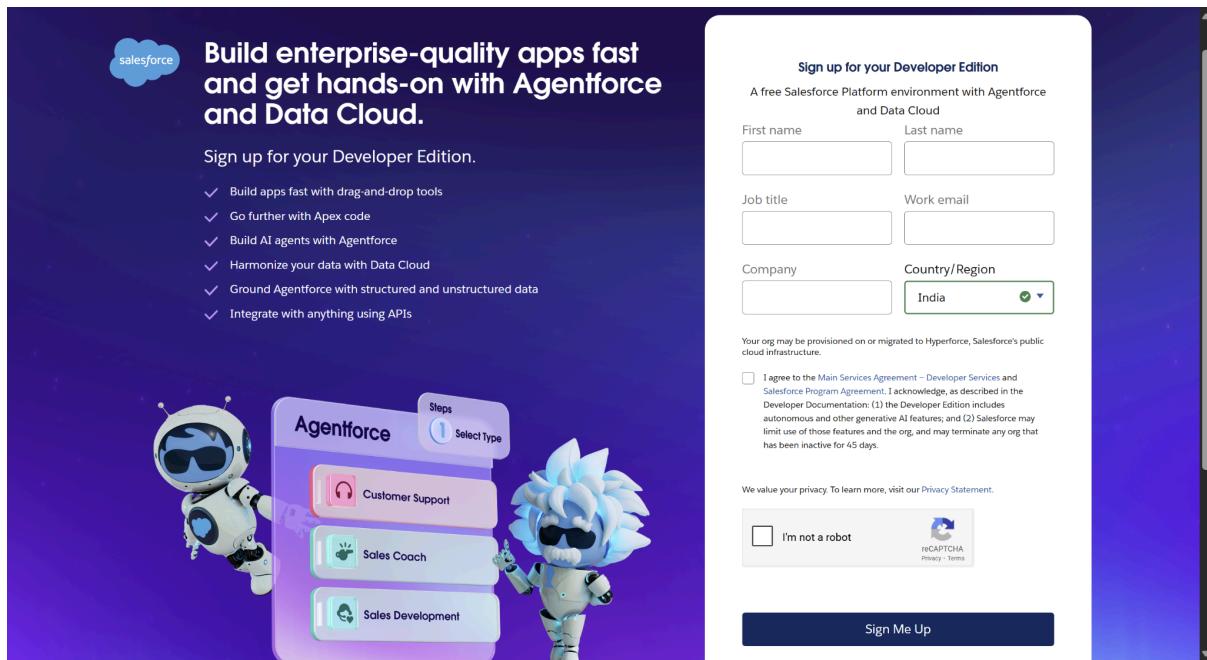
- **Existing Leave Management Apps:**
 - *Leave Management System (LMS)* – Basic leave request/approval flow.
 - *HRMS Solutions* – Broader HR apps with leave as one module.
 - *Attendance & Absence Trackers* – Focus on time/attendance integrations.
- **Key Learnings from Exploration:**
 - Most apps provide **standard workflows**, but lack **deep customization** for industry-specific policies.
 - Integration with existing Salesforce objects (Users, HR data) is often limited.
 - Many charge additional licensing costs, making a custom solution more cost-effective.

- **Decision:** Build a **custom Leave Tracking App on Salesforce CRM** with flexibility, while keeping AppExchange apps in mind for potential integrations (e.g., payroll, attendance).

Phase 2 : Org Setup & Configuration for Leave Tracking App

1. Org Preparation

- Spin up a **Developer Edition / Sandbox / Trailhead Playground**.
- Enable necessary features:
 - **Approvals**
 - **Reports & Dashboards**
 - **Email-to-User Notifications**
 - **Chatter**



2. Custom Objects & Fields

Create the objects needed to track leave requests and balances.

Objects

1. Leave Request

- Fields:
 - Employee (Lookup → User/Employee)
 - Leave Type (Picklist: Sick, Casual, Earned, etc.)
 - Start Date (Date)
 - End Date (Date)
 - Total Days (Formula)
 - Status (Picklist: Draft, Submitted, Approved, Rejected, Cancelled)
 - Reason (Long Text Area)

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Leave Request'. Below the title, there's a search bar and several global buttons. The main content area displays a table titled 'Fields & Relationships' with 10 items, sorted by Field Label. The table has columns for 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The rows show various fields like 'Created By', 'From Date', 'Last Modified By', etc., with their corresponding field names and data types. On the left side of the main content area, there's a sidebar with links to other setup categories such as Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, Triggers, Flow Triggers, Validation Rules, and Conditional Field Formatting.

3. Relationships

- **User ↔ Leave Request:** Lookup (many leave requests per user).
- **User ↔ Leave Balance:** Lookup (one record per leave type).
- **Leave Request ↔ Leave Balance:** Lookup (optional, for validation).

4. Automation

Validation Rules

- End Date ≥ Start Date.
- Total Days ≤ Available Balance.

Flows

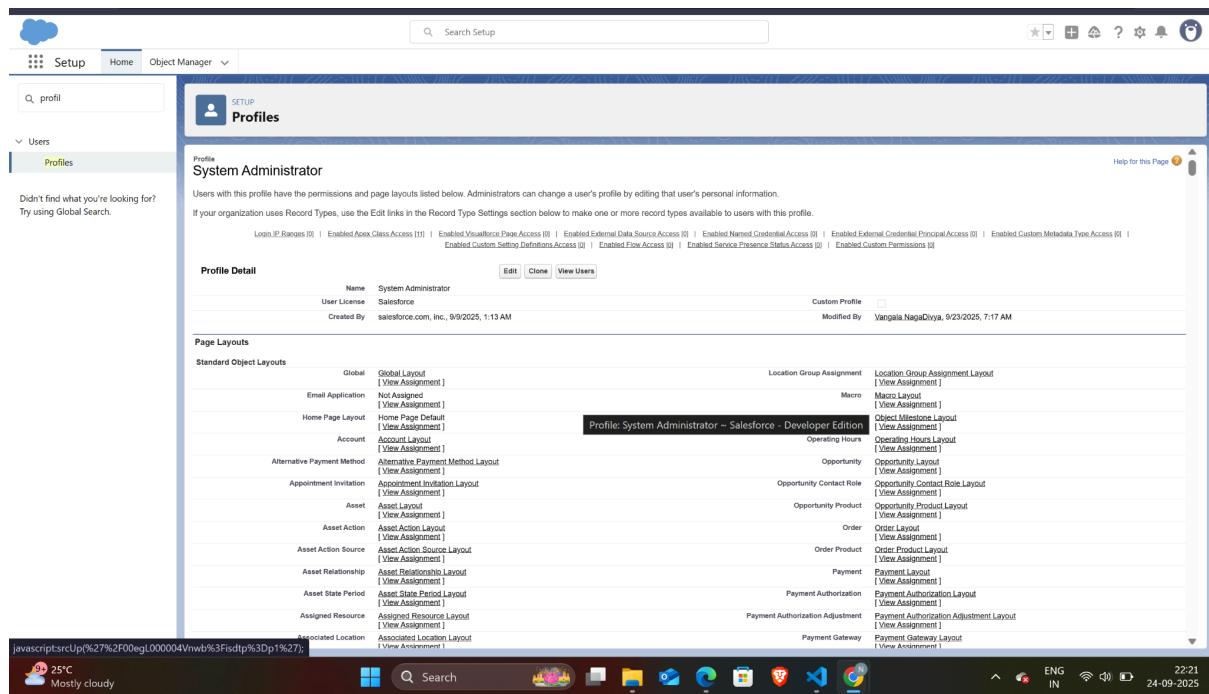
- **Submit Leave Flow:**
 - Auto-calculate total days.
 - Check leave balance.
 - Update balance on approval.
- **Approval Process:**
 - Manager approves/rejects.
 - Notification sent to employee.

Record-Triggered Flows

- On leave approval → Send Approval Email.
- On rejection → Send Rejected Email.

5. Security & Access

- **Profiles / Permission Sets**
 - Employee: Create/View own Leave Requests, View balances.
 - Manager: Approve/Reject team requests.



- **Sharing Rules**

- Leave Requests visible only to employee, manager, and HR.

6. UI Configuration

- **Lightning App Builder**

- Custom **Leave Management App** with navigation tabs:
 - Leave Requests
 - Leave Balances
 - Reports & Dashboards
 - Approvals

- **Record Pages**

- Employee-friendly leave request form.
- Manager view with team's availability panel.

7. Reports & Dashboards

- Reports:

- Leave Requests by Employee.
- Leave Balances Remaining.
- Approved vs Rejected Leaves.

- Dashboard:

- HR Overview (Total Leaves Taken, Team Availability).

8. Testing & Deployment

- Create sample users (Employee, Manager, HR).
- Test leave request submission, approval workflow, and balance deduction.
- Move to UAT → Production.

Phase 3: Data Modeling & Relationships – Leave Tracking App

1. Standard & Custom Objects

- **Standard Objects:**
 - **User** → Represents employees, managers, and HR/Admin.
- **Custom Objects:**
 - **Leave Request** → Stores employee leave applications.

2. Fields

- **Leave Request:** Employee (Lookup to User), Leave Type, Start Date, End Date, Total Days (Formula), Status, Reason.

The screenshot shows the Salesforce Object Manager interface for the 'Leave Request' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main area displays the 'Fields & Relationships' section, which lists 10 items sorted by Field Label. The table columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are: Created By (CreatedBy), From Date (From_Date__c), Last Modified By (LastModifiedBy), Leave Request Id (Name), Manager Comment (Manager_Comment__c), Owner (OwnerId), Reason (Reason__c), Status (Status__c), To Date (To_Date__c), and User (User__c). The 'Owner' field is indexed, while others are not.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
From Date	From_Date__c	Date		
Last Modified By	LastModifiedBy	Lookup(User)		
Leave Request Id	Name	Auto Number		✓
Manager Comment	Manager_Comment__c	Text Area(255)		
Owner	OwnerId	Lookup(User,Group)		✓
Reason	Reason__c	Text Area(255)		
Status	Status__c	Picklist		
To Date	To_Date__c	Date		
User	User__c	Lookup(User)		✓

3. Record Types

- **Leave Request Record Types:**
 - Sick Leave
 - Casual Leave
 - Earned Leave
 - Special Leave(ex:Maternity,Paternity)
 - Work from home

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit Del Deactivate	Work from home	Work from home	<input type="checkbox"/>	Assigned dynamically	Ganga.dev, 9/15/2025, 4:23 AM
<input type="checkbox"/> Edit Del Deactivate	Compensation off	Compensation off	<input type="checkbox"/>	Assigned dynamically	Ganga.dev, 9/15/2025, 4:23 AM
<input type="checkbox"/> Edit Del Deactivate	Privilege Leave	Privilege Leave	<input type="checkbox"/>	Assigned dynamically	Ganga.dev, 9/15/2025, 4:23 AM
<input type="checkbox"/> Edit Del Deactivate	Casual Leave	Casual Leave	<input type="checkbox"/>	Assigned dynamically	Ganga.dev, 9/15/2025, 4:23 AM
<input type="checkbox"/> Edit Del Deactivate	Sick Leave	Sick Leave	<input type="checkbox"/>	Assigned dynamically	Ganga.dev, 9/15/2025, 4:23 AM
<input type="checkbox"/> Edit Del Deactivate	Paid Leave	Paid Leave	<input type="checkbox"/>	Assigned dynamically	Ganga.dev, 9/15/2025, 4:23 AM

- Record Types allow different **picklist values, page layouts, and approval processes** for each leave type.

4. Page Layouts

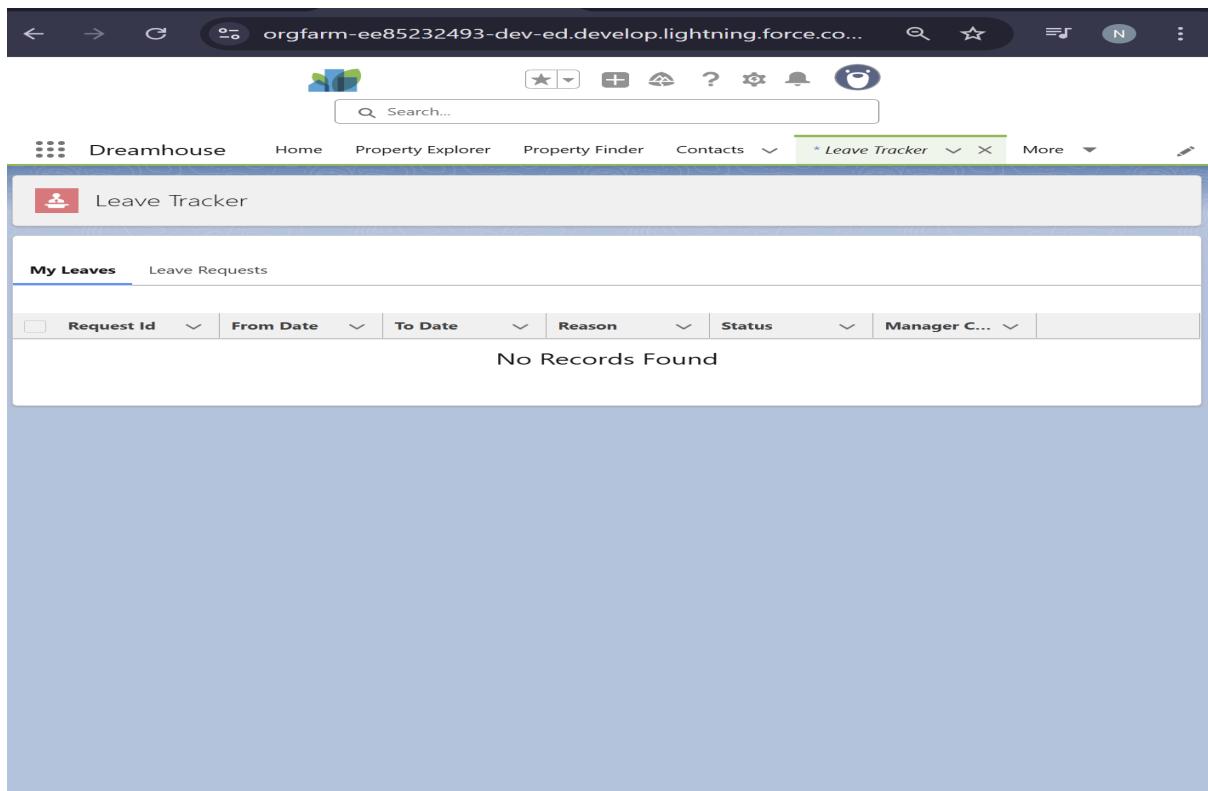
- Employee Layout:** Simple form for applying leave.
- Manager Layout:** Includes approval section and team view.

5. Compact Layouts

- For **Leave Request** (Mobile / Highlights Panel):
 - Employee, Leave Type, Start Date, End Date, Status.
- For **Leave Balance**:
 - Employee, Leave Type, Remaining Balance.

6. Schema Builder

- Use **Schema Builder** to:
 - Visually design object relationships (User ↔ Leave Request ↔ Leave Balance).
 - Add fields & relationships quickly.
 - Check dependencies across objects.



7. Lookup vs Master-Detail vs Hierarchical

- **Lookup Relationship:**
 - Used between **Leave Request** ↔ **User** and **Leave Balance** ↔ **User** (loose coupling).
- **Master-Detail Relationship:**
 - Could be used between **Leave Balance** ↔ **Leave Request** if you want cascading delete and roll-up summaries.
- **Hierarchical Relationship** (only on User):
 - Can define Manager → Employee relationship for approvals.

8. Junction Objects

- Useful if you want **many-to-many relationships**, e.g.:
 - **Employee** ↔ **Project** ↔ **Leave Requests** (if leaves are project-specific).
- For basic Leave Management, junction objects are **not mandatory**.

9. External Objects

- If company holidays or employee data is stored in an **external HR system (ERP/Payroll)**, use **External Objects** (via Salesforce Connect).
- Allows referencing external data in real-time without storing in Salesforce.

Phase 3 Outcome

- A **complete data structure** for Leave Management is defined.
- **Standard & Custom Objects** ensure flexibility and scalability.
- **Record Types & Page Layouts** provide tailored experiences for employees, managers, and HR.
- **Schema Builder** helps visualize and maintain relationships.
- Clear use of **Lookup vs Master-Detail** ensures proper data integrity.
- Future scalability supported with **Junction Objects & External Objects** if business expands.

Phase 4: Process Automation (Admin)

1. Validation Rules

Ensures correct data entry and prevents invalid records.

- **Date Validation**
End Date must always be later than Start Date.
`End_Date__c < Start_Date__c`
→ Error message: "End Date cannot be before Start Date."
- **Reason Required**
Employees must provide a reason before submitting their leave request.

Request Id	User	From Date	To Date	Reason	Status	Manager Comment
A0007	John Mikaelson	2025-09-25	2025-09-26	test	Approved	<button>Edit</button>
A0006	John Mikaelson	2025-09-25	2025-09-26	test	Pending	<button>Edit</button>
A0005	John Mikaelson	2025-09-25	2025-09-29	test	Pending	<button>Edit</button>

2. Workflow Rules

Automatically handles updates when a leave request is created.

- **Auto-update Status:**
On submission, the request status is set to “**Under Review**.”
- **Actions:**
 - Send email notification to the Manager.
 - Update Status field to “**Submitted**.”

3. Process Builder

Handles conditional automation beyond workflow rules.

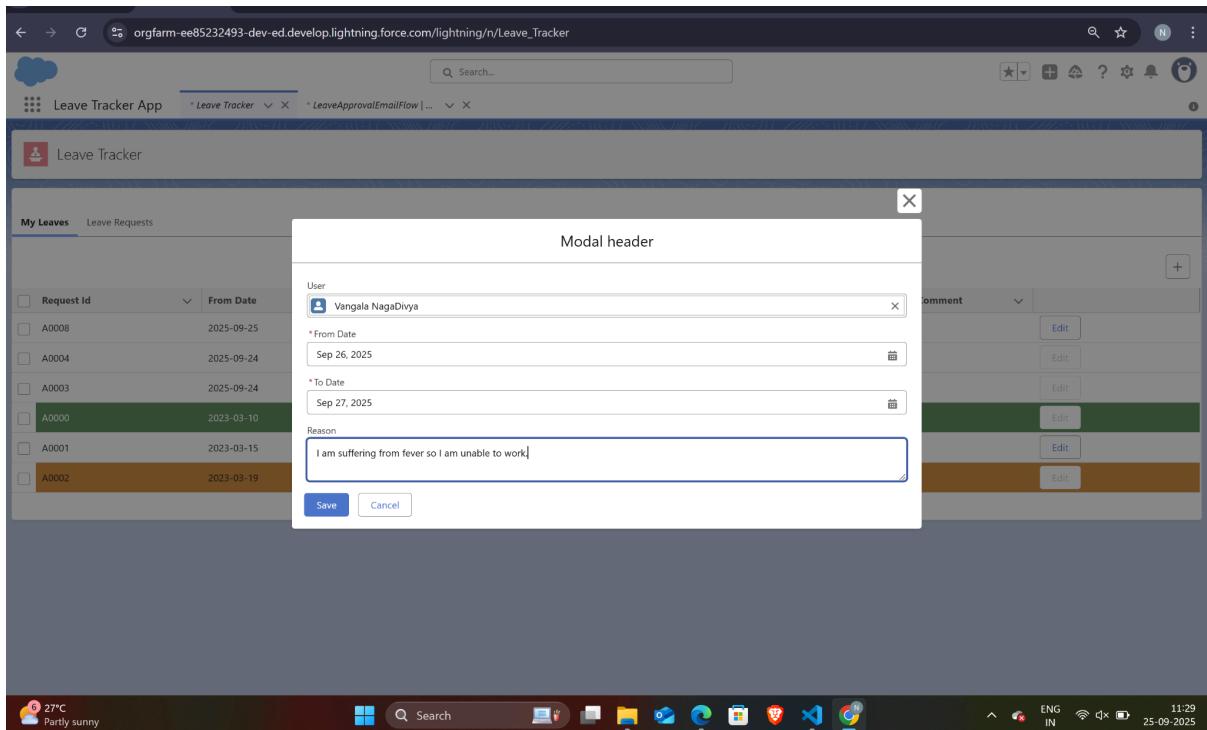
- If Status = **Approved** → Send email to employee with approval confirmation.
- If Status = **Rejected** → Send email with rejection reason and manager comments.

4. Flow Builder

Screen flows guide employees through leave request submission.

- **Steps:**

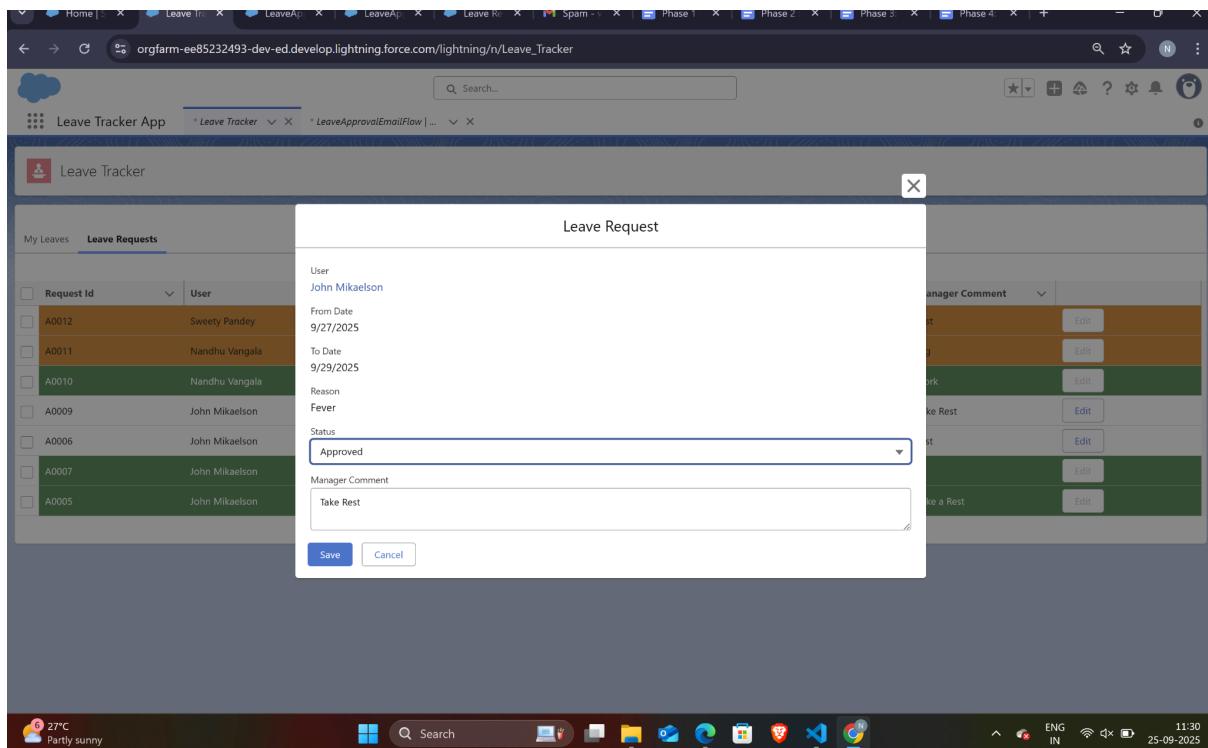
1. Select Leave Type.
2. Choose Start and End Dates.
3. Provide Reason for leave.
4. Review details & Submit.



5. Approval Process

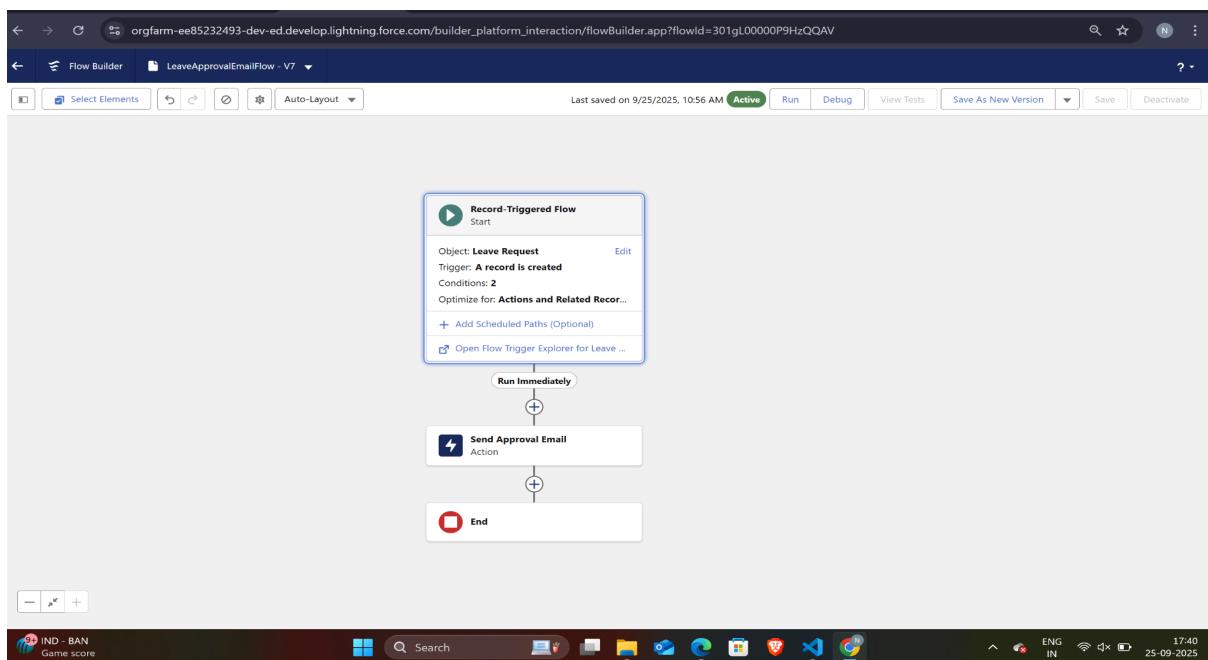
Streamlines request approval.

1. Employee submits request → Status = **Pending**.
2. Manager receives approval request.
3. Manager approves or rejects.
4. If approved → Email sent to employee.
5. If rejected → Rejection email sent to employee.



6.Email Flows

- Click Setup ()→ Quick Find→ Flows New Flow.
- Select Record-Triggered Flow.
- Choose Object: Leave_Request_Status_Notification → Trigger: When record is created→ After Save.
- Add Action Send Email Select Welcome Email Template.
- Save and Activate.



7. Email Alerts

Automated notifications keep everyone updated.

- Leave Request Submitted → Sent to Manager.
- Leave Request Approved → Sent to Employee.

The screenshot shows a Gmail inbox with a search bar at the top containing "in:spam". A single email message is visible, which has been flagged as spam. The subject of the email is "Your Leave Request has been Approved". The sender is listed as "Vangala NagaDivya via 4rs76c4npk9d5g.gl-bchpiuat.can98.bnc.salesforce.com to me". The timestamp is "10:52 (0 minutes ago)". The message body contains a note from the HR Team stating: "Hello Nandhu Vangala, Your leave request from September 25, 2025 to September 27, 2025 has been approved by your manager. Regards, HR Team". Below the message are standard Gmail interaction buttons: Reply, Forward, and a smiley face icon.

- Leave Request Rejected → Sent to Employee.

The screenshot shows a Gmail inbox with a search bar at the top containing "in:spam". A single email message is visible, which has been flagged as spam. The subject of the email is "Your Leave Request has been Rejected". The sender is listed as "Vangala NagaDivya via 4rs76c4nk98d5g-gl-bchpiuat.can98.mesara to me". The timestamp is "10:52 (0 minutes ago)". The message body contains a note from the HR Team stating: "Hello Nandhu Vangala, Your leave request from September 25, 2025 to September 27, 2025 has been rejected by your manager. Regards, HR Team". Below the message are standard Gmail interaction buttons: Reply, Forward, and a smiley face icon.

- Leave Balance Reminder → Monthly email to Employee.

8. Field Updates

Ensures status fields stay in sync with actions.

- On Approval: **Status** → **Approved**.
- On Rejection: **Status** → **Rejected**.
- On Cancellation: **Status** → **Cancelled**.

9. Tasks & Custom Notifications

Keeps managers and employees on track.

- Task auto-created for managers to review requests.
- Custom Notification (via Salesforce App) → Push notification when a request is assigned.

Phase 4 Outcome:

- All leave management processes (submission, approval, rejection, cancellation) automated.
- Real-time notifications for employees and managers.
- HR/Admin enjoy complete visibility with minimal manual work.

Phase 5: Apex Programming (Developer)

Goal: Add advanced logic and automation for Leave Tracking App.

1. Classes & Objects

- Create a **LeaveService** class to handle reusable logic like leave validation, balance calculation, and notifications.
- Example: `LeaveService.calculateRemainingLeaves(UserId)`

2. Apex Triggers

- On **LeaveRequest Insert** → Prevent overlapping leave requests for the same employee.
- On **LeaveRequest Update** → Ensure only Pending requests can be edited.

3. Trigger Design Pattern

- Implement a **LeaveRequestTriggerHandler** class.
- Trigger file only calls the handler methods → keeps code clean and testable.

4. SOQL & SOSL

Example SOQL:

```
SELECT Id, Status__c, From_Date__c, To_Date__c  
FROM LeaveRequest__c  
WHERE User__c = :UserInfo.getUserId()  
AND Status__c = 'Approved'
```

- Use to fetch approved leaves and check for overlaps.

5. Collections: List, Set, Map

- Use **Set<Date>** to store applied leave dates and prevent duplicates.
- Use **Map<Id, LeaveRequest__c>** to handle bulk triggers efficiently.

6. Control Statements

Example:

```
IF (newLeave.From_Date__c <= existingLeave.To_Date__c  
&& newLeave.To_Date__c >= existingLeave.From_Date__c) {  
    throw new LeaveException('Leave dates overlap with an existing request.');//  
}
```

7. Batch Apex

- Nightly job to **mark expired leave requests** (where **To_Date__c < TODAY** and **Status = Pending** → **Status = Expired**).

```
submitHandler(event){ event.preventDefault();  
const fields={...event.detail.fields}; fields.Status__c='Pending';  
// fields.User__c=this.currentUserId;  
if (new Date(fields.From_Date__c)>new Date(fields.To_Date__c))  
{  
    this.showToast('From date should be grater than to date','Error','error');  
}  
else if(new Date()>new Date(fields.From_Date__c)){  
    this.showToast('From date should not be less than Today','Error','error');  
}
```

```
else{
    this.refs.leaveRequestFrom.submit(fields);
}
}
```

8. Future Methods

- Call external **HR/Payroll API** asynchronously when a leave request is approved (to sync leave balances with payroll system).

@AuraEnabled

```
public static void sendNotification(String email, String subject, String body) {
    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    mail.setToAddresses(new String[] { email });
    mail.setSubject(subject);
    mail.setPlainTextBody(body);
    Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });
}
```

9. Exception Handling

- Wrap logic in **try-catch** blocks.

Example:

```
try {
    LeaveService.validateLeave(newLeave);
} catch (Exception e) {
    throw new AuraHandledException('Error: ' + e.getMessage());
}
```

10. Test Classes

- Create sample employees and leave requests.
- Insert overlapping leave request → ensure trigger throws error.
- Approve leave → ensure balance reduces correctly.
- Coverage goal: > 85%.

Phase 5 Outcome:

- Robust Apex code with reusable classes and triggers.
- Asynchronous processing ensures performance.
- Proper exception handling + test coverage makes the system reliable and deployment-ready.

Phase 6: User Interface Development

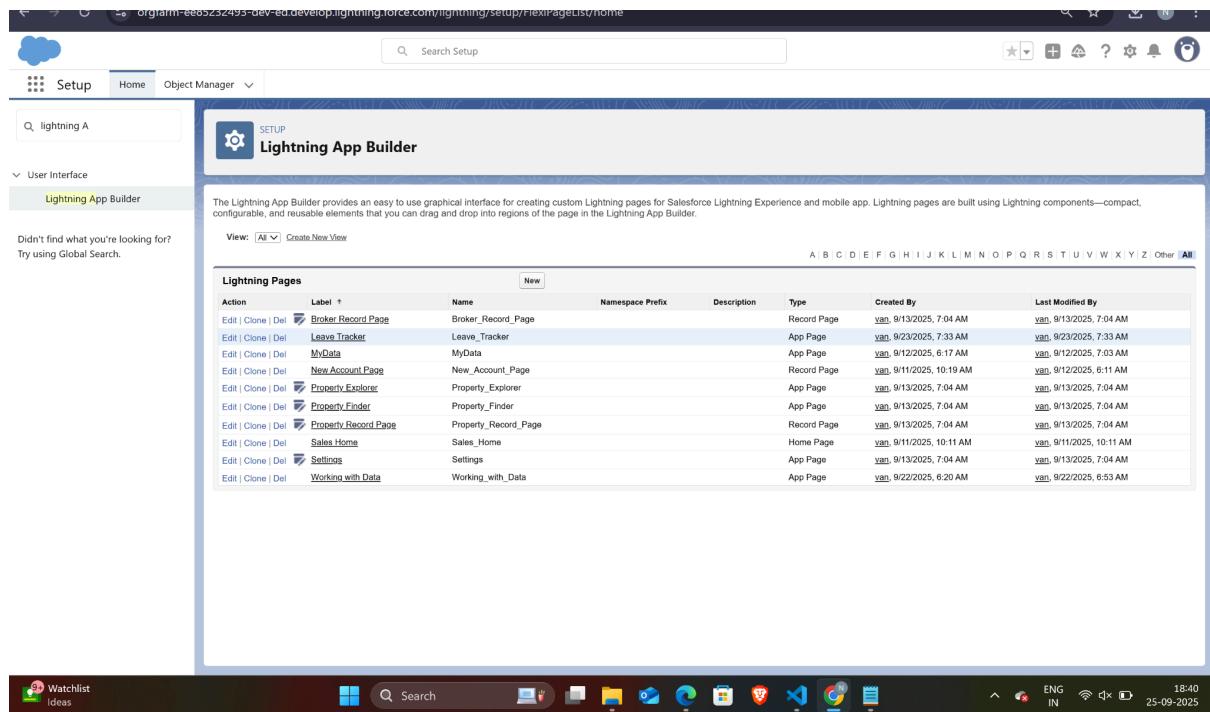
Goal: Deliver an intuitive and interactive UI for the Leave Tracking application.

1. Lightning App Builder

Created a dedicated **Leave Tracking App** in Salesforce.

The app provides navigation tabs for:

- **Home** (overview dashboard & notifications)
- **My Leaves** (employee's personal leave history)
- **Team Requests** (manager view for approvals)
- **Analytics** (reports and dashboards)



The screenshot shows the Salesforce Lightning App Builder interface. The top navigation bar includes 'SETUP' and 'Lightning App Builder'. The left sidebar has a search bar ('lightning A') and a 'User Interface' section with 'Lightning App Builder' selected. A message says 'Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Lightning App Builder' and contains a table of 'Lightning Pages'. The table has columns: Action, Label, Name, Namespace Prefix, Description, Type, Created By, and Last Modified By. The table lists several pages, including 'Broker_Record_Page', 'Leave_Tracker', 'MyData', 'New_Account_Page', 'Property_Explorer', 'Property_Finder', 'Property_Record_Page', 'Sales_Home', 'Settings', and 'Working_with_Data'. The 'Last Modified By' column shows the user 'yan' with various dates and times. The bottom of the screen shows the Windows taskbar with icons for File Explorer, Task View, File, Print, Mail, Edge, OneDrive, and others, along with system status indicators like battery level, signal strength, and date/time (18:40, 25-09-2025).

Action	Label	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Edit Clone Del	<input checked="" type="checkbox"/> Broker_Record_Page	Broker_Record_Page			Record Page	yan, 9/13/2025, 7:04 AM	yan, 9/13/2025, 7:04 AM
Edit Clone Del	<input checked="" type="checkbox"/> Leave_Tracker	Leave_Tracker			App Page	yan, 9/23/2025, 7:33 AM	yan, 9/23/2025, 7:33 AM
Edit Clone Del	<input checked="" type="checkbox"/> MyData	MyData			App Page	yan, 9/12/2025, 6:17 AM	yan, 9/12/2025, 7:03 AM
Edit Clone Del	<input checked="" type="checkbox"/> New_Account_Page	New_Account_Page			Record Page	yan, 9/11/2025, 10:19 AM	yan, 9/12/2025, 6:11 AM
Edit Clone Del	<input checked="" type="checkbox"/> Property_Explorer	Property_Explorer			App Page	yan, 9/13/2025, 7:04 AM	yan, 9/13/2025, 7:04 AM
Edit Clone Del	<input checked="" type="checkbox"/> Property_Finder	Property_Finder			App Page	yan, 9/13/2025, 7:04 AM	yan, 9/13/2025, 7:04 AM
Edit Clone Del	<input checked="" type="checkbox"/> Property_Record_Page	Property_Record_Page			Record Page	yan, 9/13/2025, 7:04 AM	yan, 9/13/2025, 7:04 AM
Edit Clone Del	<input checked="" type="checkbox"/> Sales_Home	Sales_Home			Home Page	yan, 9/11/2025, 10:11 AM	yan, 9/11/2025, 10:11 AM
Edit Clone Del	<input checked="" type="checkbox"/> Settings	Settings			App Page	yan, 9/13/2025, 7:04 AM	yan, 9/13/2025, 7:04 AM
Edit Clone Del	<input checked="" type="checkbox"/> Working_with_Data	Working_with_Data			App Page	yan, 9/22/2025, 6:20 AM	yan, 9/22/2025, 6:53 AM

The screenshot shows the Salesforce Lightning App Builder. In the top navigation bar, there are links for 'Home' and 'Object Manager'. A search bar is present at the top right. The main area is titled 'Lightning App Builder' and contains a sub-section 'g Pages'. A table lists various pages with columns for Action, Label, Name, Namespace Prefix, Description, Type, Created By, and Last Modified By. The table includes rows for 'Broker Record Page', 'Leave Tracker', 'MyData', 'New Account Page', 'Property Explorer', 'Property Finder', 'Property Record Page', 'Sales Home', 'Settings', and 'Working with Data'. The status bar at the bottom shows system information like 'Very humid Now', a taskbar with various icons, and a date/time stamp '25-09-2025'.

2. Record Pages

Customized the **LeaveRequest__c Record Page** to display:

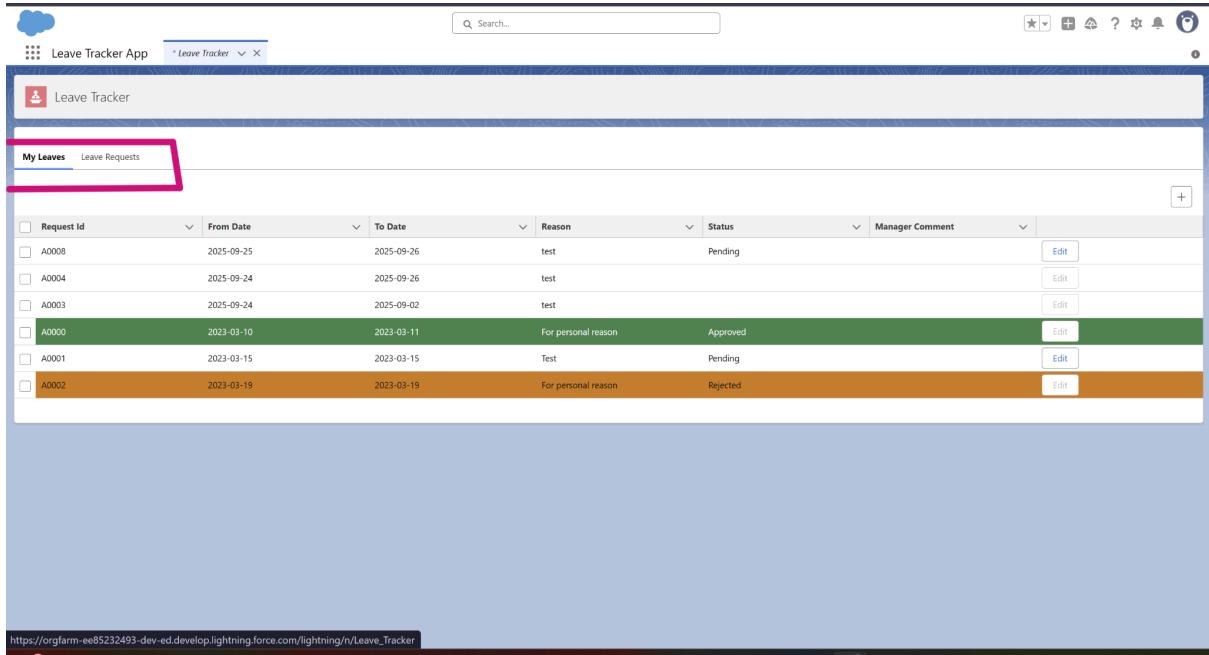
- Key employee details (Name, Department)
- Leave type, dates, and submitted reason
- Manager's comments and decision section
- Quick Actions for **Approve** and **Reject**
- Compact Layout for faster glance at status

The screenshot shows the customized 'LeaveRequest__c Record Page' for the 'Leave Tracker' app. The page has a header with 'Leave Tracker' and a sub-header 'My Leaves'. Below is a table titled 'Leave Requests' with columns: Request Id, From Date, To Date, Reason, Status, and Manager Comment. The table lists several entries, with one row ('A0000') highlighted in green and another ('A0002') in orange. Each row has an 'Edit' button. The status bar at the bottom shows 'Trending videos Cat Thinks Gues...', a taskbar with icons, and a date/time stamp '25-09-2025'.

3. Tabs

Configured custom tabs for:

- **Leave Request** (object tab for direct access)
- **My Leaves** (Lightning page with user Leaves)



The screenshot shows a Salesforce Lightning page titled "Leave Tracker". At the top, there are two tabs: "My Leaves" and "Leave Requests", with "My Leaves" being the active tab and highlighted with a pink border. Below the tabs is a table with the following data:

Request Id	From Date	To Date	Reason	Status	Manager Comment	Action
A0008	2025-09-25	2025-09-26	test	Pending		<button>Edit</button>
A0004	2025-09-24	2025-09-26	test			<button>Edit</button>
A0003	2025-09-24	2025-09-02	test			<button>Edit</button>
A0000	2023-03-10	2023-03-11	For personal reason	Approved		<button>Edit</button>
A0001	2023-03-15	2023-03-15	Test	Pending		<button>Edit</button>
A0002	2023-03-19	2023-03-19	For personal reason	Rejected		<button>Edit</button>

At the bottom of the page, the URL is visible: https://orgfarm-ee85232493-dev-ed.lightning.force.com/lightning/n/Leave_Tracker.

4. Home Page Layouts

Designed a tailored home page with:

- **Leave Balance Chart** (available vs. used leave days)
- **Quick Action: Request Leave**
- **Recent Leave Submissions** list view

Request Id	From Date	To Date	Reason	Status	Manager Comment
A0013	2025-09-26	2025-09-27	Fever	Pending	<button>Edit</button>
A0008	2025-09-25	2025-09-26	test	Pending	<button>Edit</button>
A0004	2025-09-24	2025-09-26	test	<button>Edit</button>	
A0003	2025-09-24	2025-09-02	test	<button>Edit</button>	
A0000	2023-03-10	2023-03-11	For personal reason	Approved	<button>Edit</button>
A0002	2023-03-19	2023-03-19	For personal reason	Rejected	<button>Edit</button>
A0001	2023-03-15	2023-03-15	Test	Pending	<button>Edit</button>

5. Utility Bar

Added shortcuts for:

- **Apply for Leave** (record form popup)
- **Ask HR** (case creation for leave-related queries)

6. Lightning Web Components (LWC)

Developed multiple LWCs to improve user experience:

applyLeave

Form for employees to submit leave requests.

Fields include: Leave Type, From Date, To Date, Reason.

Validates input before sending to Apex.

```
// applyLeave.js
import { LightningElement, track } from 'lwc';
import createLeave from '@salesforce/apex/LeaveRequestController.createLeave';

export default class ApplyLeave extends LightningElement {
    @track leaveType;
    @track fromDate;
    @track toDate;
    @track reason;

    handleSubmit() {
        if(!this.leaveType || !this.fromDate || !this.toDate) {
            alert('Please complete all required fields');
        }
    }
}
```

```

        return;
    }
    createLeave({
        leaveType: this.leaveType,
        fromDate: this.fromDate,
        toDate: this.toDate,
        reason: this.reason
    })
    .then(() => {
        alert('Leave submitted successfully');
    })
    .catch(error => {
        console.error(error);
    });
}
}

```

myLeaves

- Shows logged-in user's past and upcoming leave requests.
- Data fetched using `@wire(getMyLeaves)`.
- Status highlighted with colors:
 - Approved = Green
 - Pending = Orange
 - Rejected = Red

teamRequests (Manager View)

- Displays all pending leave requests from direct reports.
- Includes **Approve** / **Reject** buttons.
- On action, calls Apex to update record and sends email notifications.

```

// teamRequests.js
import { LightningElement, wire } from 'lwc';
import getTeamRequests from
'@salesforce/apex/LeaveRequestController.getTeamRequests';
import updateRequestStatus from
'@salesforce/apex/LeaveRequestController.updateRequestStatus';

export default class TeamRequests extends LightningElement {
    @wire(getTeamRequests) requests;

    handleAction(event) {
        const leaveId = event.target.dataset.id;

```

```

    const status = event.target.dataset.status;

    updateRequestStatus({ leaveld, status })
      .then(() => {
        alert(`Leave ${status}`);
      })
      .catch(error => {
        console.error(error);
      });
  }
}

```

7. Apex with LWC

- **Imperative Apex Calls** → used in `applyLeave` and `teamRequests` for inserts/updates.

```
import {refreshApex} from '@salesforce/apex';
```

- **Wire Adapters** → used in `myLeaves` to auto-refresh data.

```

@wire(getMyLeaves)
wiredMyLeaves(result){
  this.myLeavesWireResult=result;
  if(result.data){
    this.myLeaves=result.data.map(a=>(
      ...a,
      cellClass:a.Status__c == 'Approved' ? 'slds-theme_success': a.Status__c ==
      'Rejected' ? 'slds-theme_warning' : '',
      isEditDisabled:a.Status__c!= 'Pending'
    )));
  }
  if(result.error){
    console.log('Error occurred while fetching my leaves- ',result.error);
  }
}

```

8. Navigation Service

- After submitting a request → navigate to **Leave Record page**.
- After manager approval/rejection → redirect to **Team Requests dashboard**.

- Used Salesforce **NavigationMixin** for seamless transitions.

Phase 6 Outcome

- Built a **modern Lightning interface** with reusable LWCs.
- Employees can submit, track, and cancel leave requests effortlessly.
- Managers can quickly review and act on pending requests.
- The app aligns with Salesforce Lightning UX standards and ensures productivity.

Phase 7: Integration & External Access

Goal: Enable the Leave Tracking app to connect with external systems and services securely.

1. Named Credentials

- Created **Named Credentials** to securely store API keys and endpoints.
- Example: HRMS API credentials to validate employee leave balances in an external HR system.

2. External Services

- Integrated Salesforce with **HR / Payroll systems** for leave balance validation.
- Used **External Services** to import schema and auto-generate Apex classes.

Sample Code (HTTP Callout):

```
public with sharing class LeaveIntegrationService {
    public static void getLeaveBalance(String employeeId) {
        Http http = new Http();
        HttpRequest req = new HttpRequest();
        req.setEndpoint('callout:HR_System/leavebalance/' + employeeId);
```

```

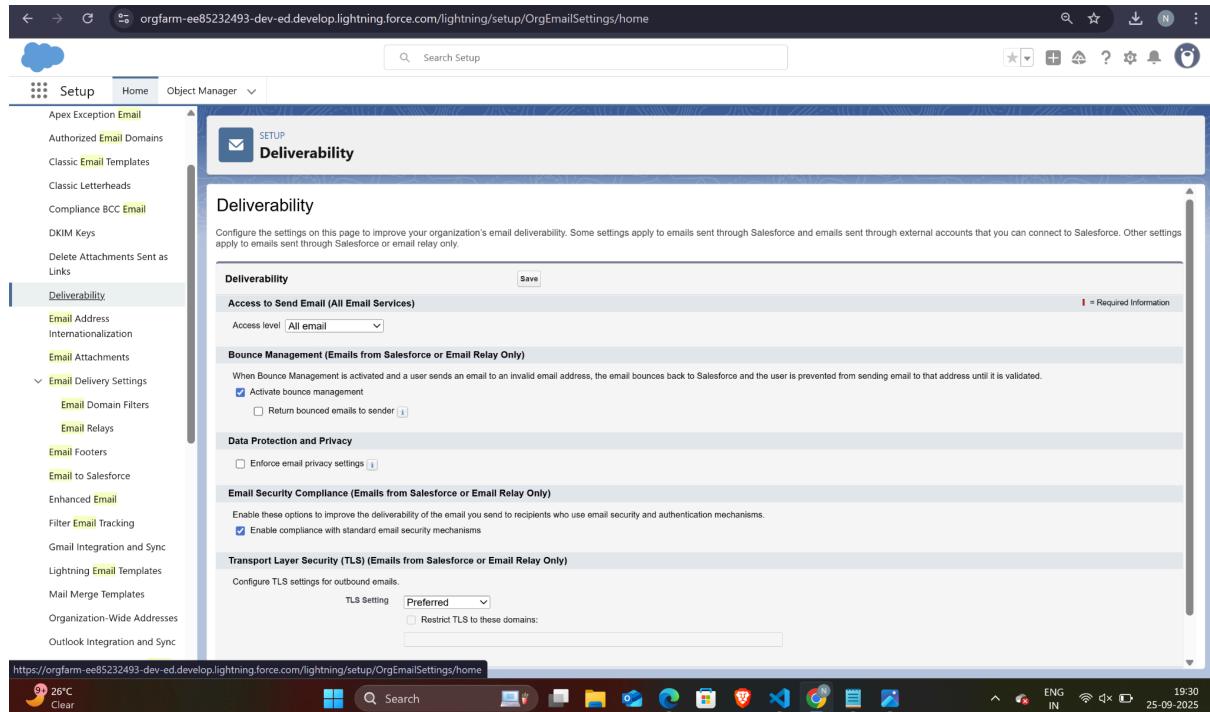
req.setMethod('GET');

HttpResponse res = http.send(req);
if(res.getStatusCode() == 200){
    System.debug('Leave Balance: ' + res.getBody());
} else {
    System.debug('Error: ' + res.getStatus());
}
}
}
}

```

3. Email Service Integration

- Salesforce sends email notifications when leave is approved/rejected.
- Can also integrate with Gmail/Outlook API for two-way sync.



4. Platform Events

- Published a **LeaveRequestEvent__e** when a new leave request is submitted.
- External payroll/attendance systems can subscribe and update their records automatically.

Sample Platform Event Trigger:

```

trigger LeaveEventTrigger on LeaveRequest__c (after update) {
    for(LeaveRequest__c lr : Trigger.new){

```

```

        if(lr.Status__c == 'Approved'){
            Leave_Status_Event__e eventMsg = new Leave_Status_Event__e(
                LeaveId__c = lr.Id,
                Status__c = 'Approved',
                EmployeeEmail__c = lr.User__r.Email
            );
            Database.SaveResult result = EventBus.publish(eventMsg);
        }
    }
}

```

5. Change Data Capture (CDC)

- Enabled **CDC** for the `LeaveRequest__c` object.
- When leave status changes (Approved / Rejected), external HR tools receive updates instantly.

6. Salesforce Connect

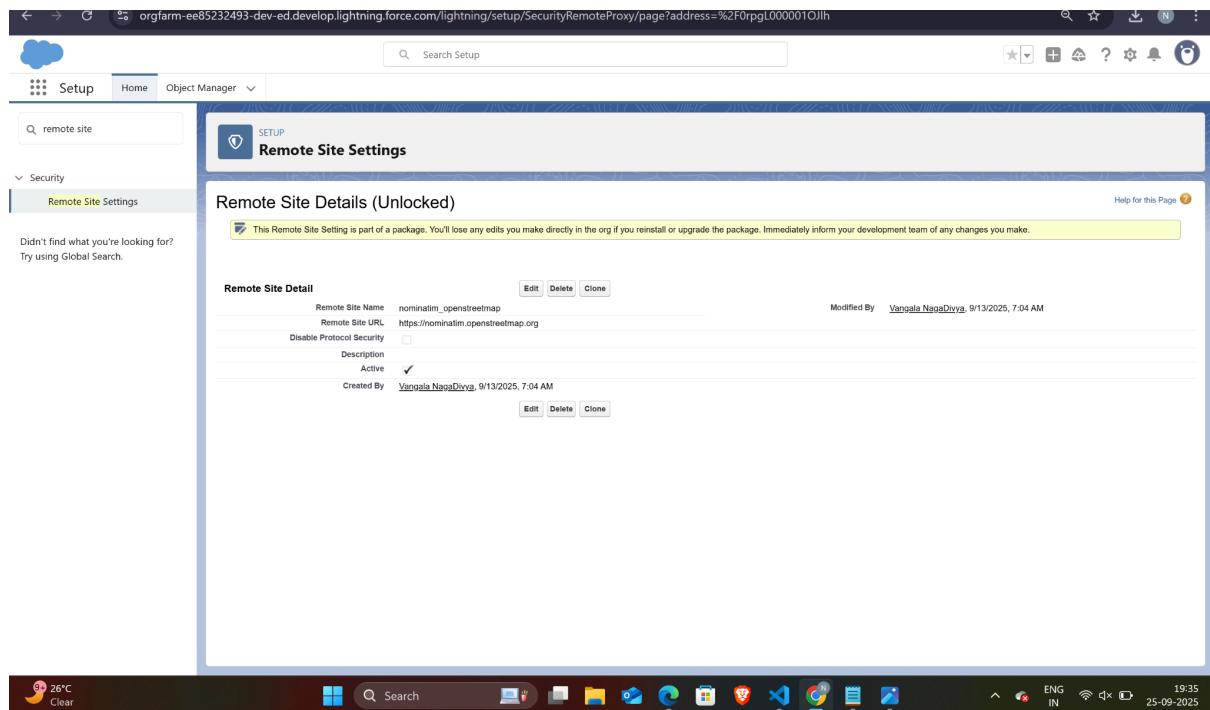
- Configured **Salesforce Connect** to link an **external HR database**.
- Employees can view leave balances stored outside Salesforce **without data duplication**.

7. OAuth & Authentication

- Implemented **OAuth 2.0** for secure login when employees access the **Employee Self-Service Portal**.
- Portal users authenticate with company's Single Sign-On (SSO).

8. Remote Site Settings

- Configured **Remote Site Settings** to allow callouts to HR/Payroll domains (e.g., <https://hr.company.com>).



Phase 7 Outcome

- Leave requests are **validated and synced** with external HR systems in real time.
- Managers see accurate leave balances across systems.
- Employees have a seamless experience whether using Salesforce or external portals.
- The app is now **integration-ready and scalable** for enterprise needs.

Phase 8: Data Management & Deployment

Goal: Manage leave-related data efficiently and ensure smooth deployment between environments.

1. Data Import Wizard

- Use Salesforce **Data Import Wizard** to import small sets of data.
- Example: Import 50 demo **Employee LeaveRequest__c** records for testing.
- Simple UI-based tool → no technical setup required.

2. Data Loader

- For bulk operations (insert, update, delete).
- Example: Import **thousands of leave records** from an external HR system.
- Supports CSV-based upload and scheduled automation.

Example CLI Command:

```
sfdx force:data:tree:import --plan leaveRequestPlan.json
```

3. Duplicate Rules

- Prevents multiple **leave requests** for the same employee on the same dates.
- Example: Duplicate Rule → block if *From_Date__c* and *To_Date__c* overlap with existing leave request for the same employee.

Example Rule:

```
Block duplicate LeaveRequest__c if Employee = same and From Date/To Date overlaps
```

4. Data Export & Backup

- Weekly export of all **LeaveRequest__c** and **Employee__c** data.
- Backup ensures HR can recover records if any accidental deletion occurs.

5. Change Sets

- Move **LWC components (applyLeave, myLeaves, leaveRequest)**, validation rules, approval processes, and workflows from **Sandbox** → **Production**.
- Ensures configuration changes are properly tracked and deployed.
- Used Change Sets for deploying metadata between environments (Sandbox → Production).
- **Included:**

Custom Objects (LeaveRequest__c)

Apex Classes (LeaveRequestController, Triggers)

Lightning Web Components (applyLeave, myLeaves, leaveRequest)

Email Templates & Flows

6. Unmanaged vs Managed Packages

- **Unmanaged Package:** Share code/configurations internally within the organization.
- **Managed Package:** Use if you plan to publish the **Leave Management App** on Salesforce AppExchange for external customers.

7. ANT Migration Tool

- Command-line based deployment tool.
- Example: Migrate **LeaveRequest__c custom object, Apex classes, and LWCs** between environments.
- Useful for automated deployments in CI/CD pipelines.

8. VS Code & SFDX

- Use **Salesforce DX + VS Code** for developer-friendly deployments.
- Version control with Git.
- Automates pushing/pulling metadata between **scratch orgs, sandboxes, and production**.

Prerequisites

Install Visual Studio Code.

Install Salesforce CLI (SFDX).

Install Salesforce Extension Pack in VS Code.

Connect Org → sf杏 force:auth:web:login -d -a DevHub.

```
leave-tracker-project/
├── force-app/
│   └── main/
│       └── default/
│           ├── classes/      (Apex Classes)
│           ├── lwc/          (Lightning Web Components)
│           ├── objects/      (Custom Objects)
│           ├── triggers/     (Apex Triggers)
│           └── email/        (Email Templates)
└── sfdx-project.json
└── .gitignore
```

Sample Deployment Commands

Retrieve metadata from source org
sfdx force:source:retrieve -m ApexClass,CustomObject,LWC
Deploy metadata to target org
sfdx force:source:deploy -p force-app/main/default
Run all tests before deployment
sfdx force:apex:test:run --resultformat human --codecoverage

Deployment Checklist:

- Code Quality Check (PMD, Prettier for LWC).
- Apex Test Coverage $\geq 75\%$.
- Run Validation Deployment (test without committing).
- Backup Production metadata before final push.
- Post-deployment steps (activate Flows, verify Email Deliverability).

Phase 8 Outcome:

- Reliable data import/export for leave records.
- Duplicate prevention ensures clean data.
- Backup strategy protects against data loss.
- Deployment tools (Change Sets, ANT, SFDX) streamline moving changes from Sandbox → Production.
- Leave Tracking App is secure, scalable, and ready for enterprise deployment.

Phase 9: Reporting, Dashboards & Security Review

Goal: Provide HR & Managers with insights while ensuring secure access to leave data.

1. Reports

- **Employee Leave Report:** Total leaves taken by each employee in a period.
- **Leave Type Report:** Breakdown of Sick Leave, Casual Leave, etc.
- **Pending Requests Report:** Leaves still awaiting manager approval.

The screenshot shows a Salesforce Lightning interface with a report titled "Leave Requests Report". The report displays a grid of leave requests with columns for Status, Leave Request Id, Leave Request: Created By, Leave Request: Created Date, From Date, To Date, Manager Comment, User, and Reason. The data is grouped by status: Pending (4), Approved (5), and Rejected (3). Each group has a subtotal row. The total number of records is 14.

Status	Leave Request Id	Leave Request: Created By	Leave Request: Created Date	From Date	To Date	Manager Comment	User	Reason
Pending (4)								
	A0003	Vangala NagaDivya	9/23/2025	9/24/2025	9/26/2025	-	Vangala NagaDivya	test
	A0004	Vangala NagaDivya	9/23/2025	9/24/2025	9/26/2025	-	Vangala NagaDivya	test
	A0005	Vangala NagaDivya	9/23/2025	9/25/2025	9/26/2025	-	Vangala NagaDivya	Test
	A0006	Vangala NagaDivya	9/24/2025	9/25/2025	9/27/2025	-	Vangala NagaDivya	test
	A0007	Vangala NagaDivya	9/24/2025	9/25/2025	9/26/2025	-	John Mikaelson	Fever
	A0008	Vangala NagaDivya	9/24/2025	9/25/2025	9/27/2025	Take Rest	John Mikaelson	Fever
	A0009	Vangala NagaDivya	9/24/2025	9/27/2025	9/29/2025	-	Nandhu Vangala	Going to family trip
Approved (5)								
	A0010	Vangala NagaDivya	9/23/2025	3/10/2023	3/11/2023	-	Vangala NagaDivya	For personal reason
	A0011	Vangala NagaDivya	9/24/2025	9/25/2025	9/26/2025	-	John Mikaelson	test
	A0012	Vangala NagaDivya	9/24/2025	9/25/2025	9/29/2025	Take a Rest	John Mikaelson	test
	A0013	Vangala NagaDivya	9/24/2025	9/25/2025	9/26/2025	test	John Mikaelson	test
Rejected (3)								
	A0014	Vangala NagaDivya	9/23/2025	3/19/2023	3/19/2023	-	Vangala NagaDivya	For personal reason
	A0015	Vangala NagaDivya	9/24/2025	9/26/2025	9/27/2025	Jhg	Nandhu Vangala	dfg
	A0016	Vangala NagaDivya	9/24/2025	9/27/2025	9/28/2025	test	Sweety Pandey	test
Total (14)								

2. Report Types

- Create a **custom report type** combining **Employee (User)** and **LeaveRequest__c**.
- Example: “Employee Leave Requests by Department” → shows department-wise leave usage.

3. Dashboards

Dashboards visually display leave metrics.

A. Create Dashboard

Go to App Launcher → Dashboards → New

Add components from leave reports:

Metric: Count of Pending Leaves

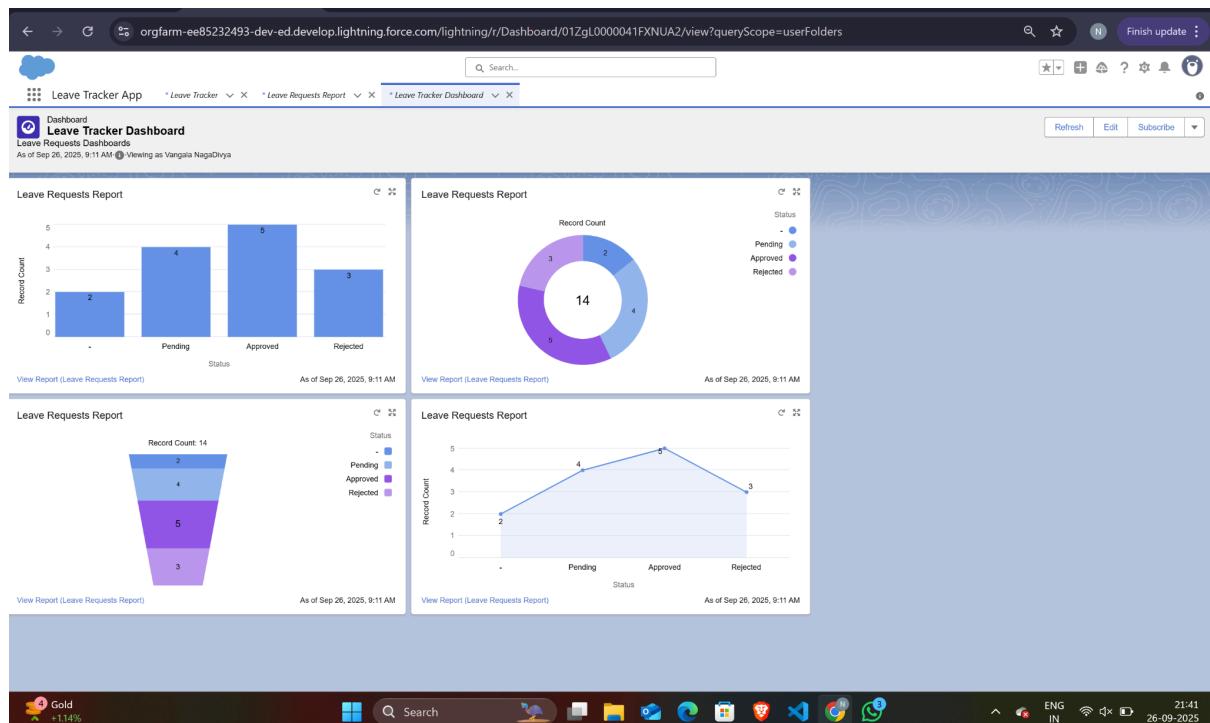
Pie Chart: Leaves by Type

Bar Chart: Leaves by Status

Line Chart: Leaves Over Time

B. Dynamic Dashboards

Managers → See only their team's data. HR/Admins → See complete organization-level data.



4. Sharing Settings

- **Leave Requests:** Private (only employee, manager, and HR can see).
- **Employee Directory:** Public read-only (basic details only).

The screenshot shows the 'Sharing Settings' page under the 'SETUP' tab. On the left sidebar, 'Sharing' is selected under 'Security'. The main content area displays various sharing rule sections: 'Work Plan Sharing Rules', 'Work Plan Template Sharing Rules', 'Work Step Template Sharing Rules', 'Work Type Sharing Rules', 'Work Type Group Sharing Rules', 'Broker Sharing Rules', 'Leave Request Sharing Rules' (which is highlighted with a red box), and 'Property Sharing Rules'. Each section has 'New' and 'Recalculate' buttons.

The screenshot shows the Salesforce Sharing Settings page. The left sidebar has a search bar and navigation links for Setup, Home, and Object Manager. Under Security, it lists 'Guest User Sharing Rule Access Report' and 'Sharing Settings'. A note says 'Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Sharing Settings' and contains a table of sharing rules. The table has two columns: 'Object' and 'Sharing Rule'. The 'Object' column lists various Salesforce objects, and the 'Sharing Rule' column shows sharing levels (e.g., Public Read/Write, Private) and checkboxes for 'Public Read Only' and 'Private'. The 'Leave Request' row is highlighted with a red box. At the bottom, there are 'Other Settings' options like Standard Report Visibility, Manual User Record Sharing, Manager Groups, Secure guest user record access, and Require permission to view record names in lookup fields. Buttons for Save and Cancel are at the bottom right.

5. Session Settings

- Configure session timeout to **30 minutes** of inactivity.
- Prevents unauthorized access on shared devices.

6. Audit Trail

- Track all changes to **LeaveRequest__c** records.
- Example: Who approved/rejected a leave, and when.
- Provides compliance visibility for HR audits.

The screenshot shows the 'View Setup Audit Trail' page in the Salesforce Setup interface. The table displays the following data:

Date	User	Source Namespace Prefix	Action	Section	Delegate User
9/24/2025, 11:23:06 PM PDT	vangalanagadiya123447@agentforce.com		Changed myLeaves Lightning Web Component	Lightning Components	
9/24/2025, 11:23:06 PM PDT	vangalanagadiya123447@agentforce.com		Changed leaveTracker Lightning Web Component	Lightning Components	
9/24/2025, 11:23:06 PM PDT	vangalanagadiya123447@agentforce.com		Changed leaveRequests Lightning Web Component	Lightning Components	
9/24/2025, 11:22:10 PM PDT	vangalanagadiya123447@agentforce.com		Changed myLeaves Lightning Web Component	Lightning Components	
9/24/2025, 11:22:10 PM PDT	vangalanagadiya123447@agentforce.com		Changed leaveTracker Lightning Web Component	Lightning Components	
9/24/2025, 11:22:10 PM PDT	vangalanagadiya123447@agentforce.com		Changed leaveRequests Lightning Web Component	Lightning Components	
9/24/2025, 11:16:17 PM PDT	vangalanagadiya123447@agentforce.com		Changed myLeaves Lightning Web Component	Lightning Components	
9/24/2025, 11:16:17 PM PDT	vangalanagadiya123447@agentforce.com		Changed leaveTracker Lightning Web Component	Lightning Components	
9/24/2025, 11:16:17 PM PDT	vangalanagadiya123447@agentforce.com		Changed leaveRequests Lightning Web Component	Lightning Components	
9/24/2025, 11:29:29 PM PDT	vangalanagadiya123447@agentforce.com		Changed myLeaves Lightning Web Component	Lightning Components	
9/24/2025, 11:29:29 PM PDT	vangalanagadiya123447@agentforce.com		Changed leaveTracker Lightning Web Component	Lightning Components	
9/24/2025, 10:55:38 PM PDT	vangalanagadiya123447@agentforce.com		Changed leaveRequests Lightning Web Component	Lightning Components	
9/24/2025, 10:55:38 PM PDT	vangalanagadiya123447@agentforce.com		Deleted flow version #7 "LeaveApprovalEmailFlow" for flow with Unique Name "LeaveApprovalEmailFlow"	Flows	
9/24/2025, 10:55:38 PM PDT	vangalanagadiya123447@agentforce.com		Created flow version #7 "LeaveApprovalEmailFlow" for flow with Unique Name "LeaveApprovalEmailFlow"	Flows	
9/24/2025, 10:26:52 PM PDT	vangalanagadiya123447@agentforce.com		Deactivated flow version #6 "LeaveApprovalEmailFlow" for flow with Unique Name "LeaveApprovalEmailFlow"	Flows	
9/24/2025, 10:26:52 PM PDT	vangalanagadiya123447@agentforce.com		Activated flow version #7 "LeaveApprovalEmailFlow" for flow with Unique Name "LeaveApprovalEmailFlow"	Flows	
9/24/2025, 10:26:48 PM PDT	vangalanagadiya123447@agentforce.com		Deleted flow version #7 "LeaveApprovalEmailFlow" for flow with Unique Name "LeaveApprovalEmailFlow"	Flows	
9/24/2025, 10:26:46 PM PDT	vangalanagadiya123447@agentforce.com		Created flow version #7 "LeaveApprovalEmailFlow" for flow with Unique Name "LeaveApprovalEmailFlow"	Flows	
9/24/2025, 10:25:32 PM PDT	vangalanagadiya123447@agentforce.com		Activated flow version #6 "LeaveApprovalEmailFlow" for flow with Unique Name "LeaveApprovalEmailFlow"	Flows	
9/24/2025, 10:22:18 PM PDT	vangalanagadiya123447@agentforce.com		Activated flow version #7 "LeaveApprovalEmailFlow" for flow with Unique Name "LeaveApprovalEmailFlow"	Flows	

Download setup audit trail for last six months (Excel .csv file)

Phase 9 Outcome:

- Real-time insights into leave trends and approval status.
- Dashboards for employees, managers, and HR for tailored reporting.
- Strong security with FLS, IP restrictions, and audit trails.
- Ensures both **data visibility** and **data protection** across the Leave Tracking App.

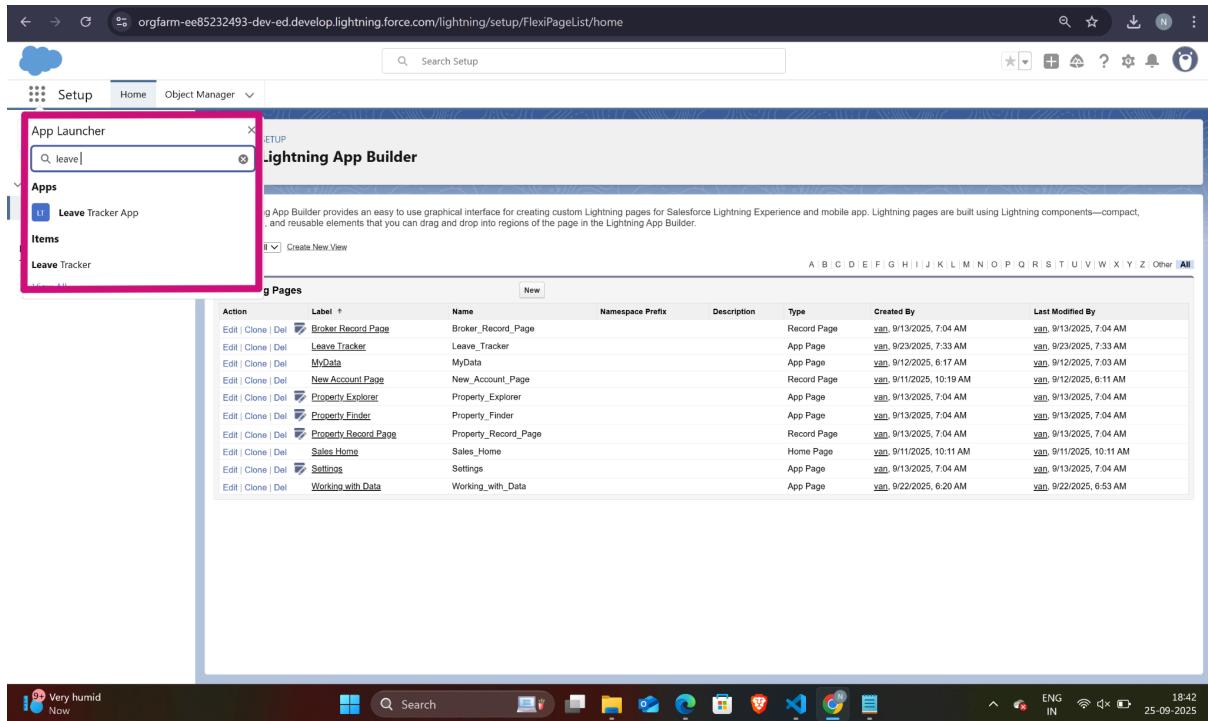
Phase 10: Final Presentation & Demo Day

Goal: Wrap up the project like a real-world delivery by showcasing functionality, value, and documentation.

1. Pitch Presentation

- **Problem:**
 - Manual leave requests cause delays.
 - Employees lack transparency on leave balance/status.
 - Managers spend time tracking approvals via emails or spreadsheets.
- **Solution:**
 - A Salesforce-based **Leave Tracker App** with:
 - Online leave request submission.

- Manager approval/rejection workflow.
- Automated email notifications.
- Real-time dashboards and reports.



- **Benefits:**

- Streamlined leave approval process.
- Improved accuracy and reduced manual errors.
- Instant visibility into employee leave trends.
- Secure & scalable within Salesforce.

The screenshot shows a web browser window for the 'Leave Tracker App' on a Salesforce Lightning page. The URL is https://orgfarm-ee85232493-dev-ed.lightning.force.com/lightning/n/Leave_Tracker. The page title is 'Leave Tracker'. The main content is a table titled 'Leave Requests' with columns: Request Id, User, From Date, To Date, Reason, Status, Manager Comment, and Edit buttons. One row for request A0010 is highlighted with a green background. A red rectangular box highlights this row. The status for A0010 is 'Approved' with a comment 'work'. The browser's taskbar at the bottom shows various pinned icons and the date 25-09-2025.

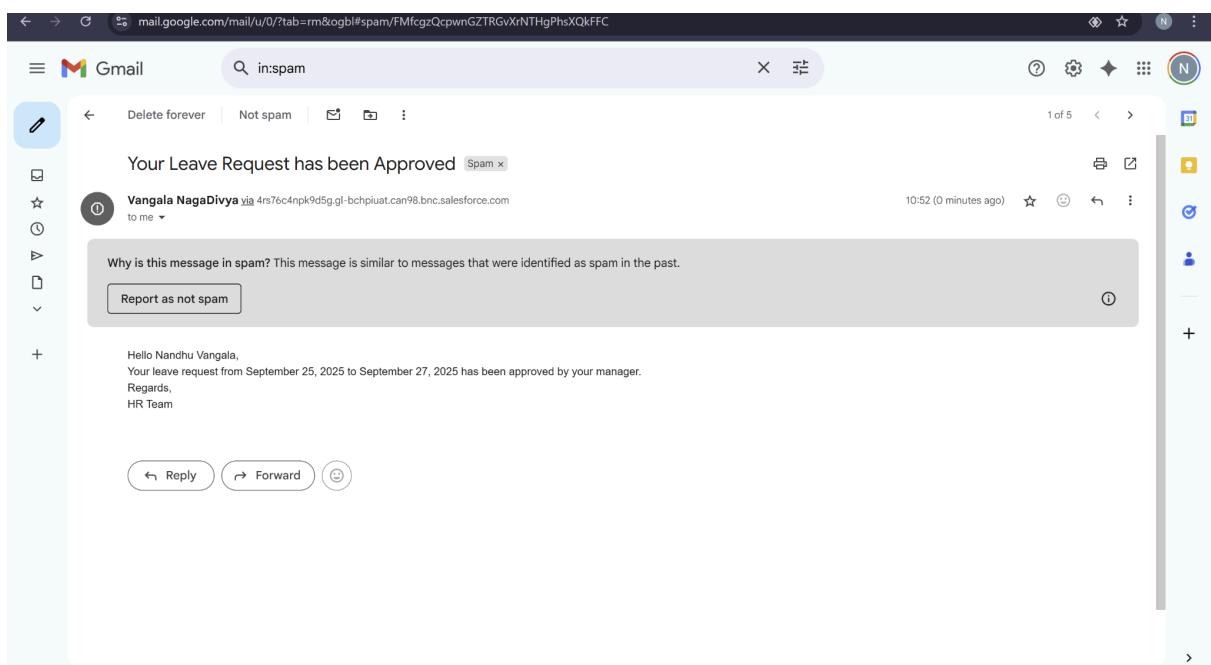
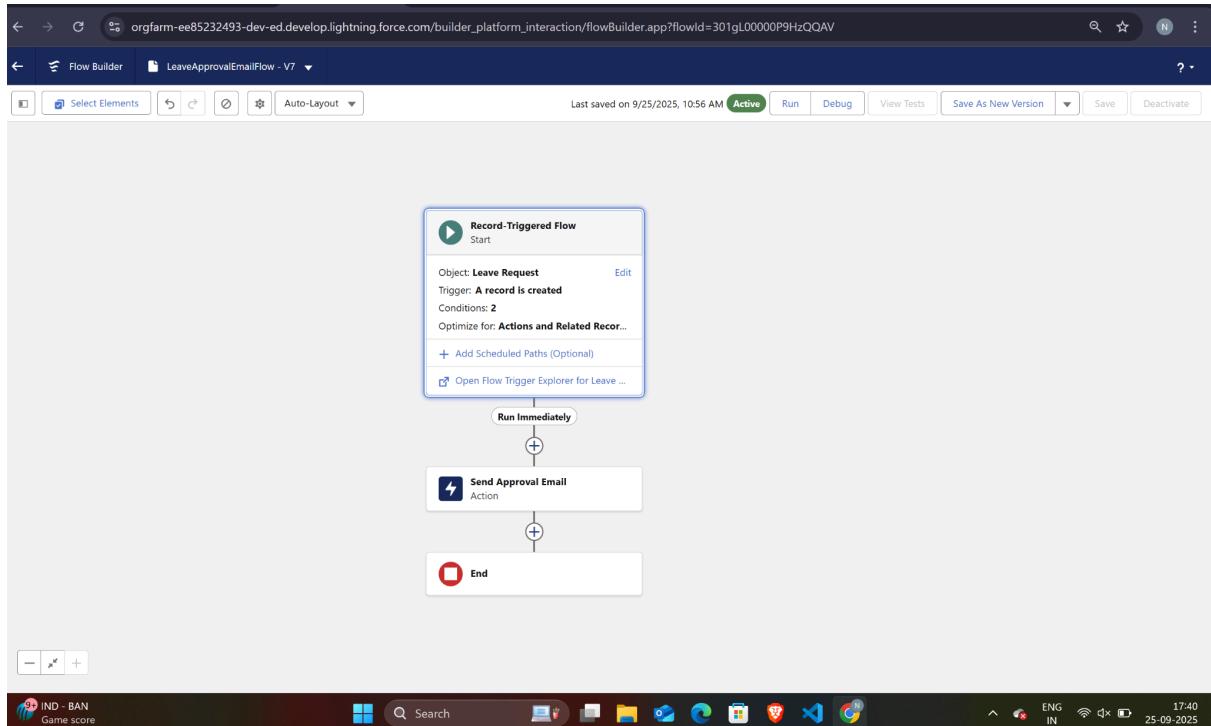
2. Demo Walkthrough

- **Employee Workflow:**
 - Log in and click **Apply Leave** (LWC form).
 - Fill details → Submit → Leave Request record created.
 - Validation ensures proper date entries.

The screenshot shows a modal dialog box over the 'Leave Requests' list. The modal header says 'Modal header'. It contains fields for 'User' (Vangala NagaDivya), 'From Date' (Sep 26, 2025), 'To Date' (Sep 27, 2025), and 'Reason' (I am suffering from fever so I am unable to work). At the bottom are 'Save' and 'Cancel' buttons. The background shows the same leave request list as the previous screenshot, with request A0000 highlighted. The browser's taskbar at the bottom shows various pinned icons and the date 25-09-2025.

- **Manager Workflow:**

- Open **Leave Requests** tab.
- Review pending requests.
- Approve or Reject directly from the record page or LWC.
- Automatic email sent to employee.



- **Reports & Dashboards:**

- "My Leave Requests" report for employees.
- Manager dashboard showing Leaves.

- o Status charts: Approved , Pending , Rejected.

Leave Requests Report

Leave Request: Owner Name	Leave Request: Leave Request Id	Leave Request: Created By	Leave Request: Created Date	From Date	To Date	Manager Comment	User	Reason
Vangala NagaDivya (14)	A0000	Vangala NagaDivya	9/23/2025	3/10/2023	3/11/2023	-	Vangala NagaDivya	For personal reason
	A0001	Vangala NagaDivya	9/23/2025	3/15/2023	3/15/2023	-	Vangala NagaDivya	Test
	A0002	Vangala NagaDivya	9/23/2025	3/19/2023	3/19/2023	-	Vangala NagaDivya	For personal reason
	A0003	Vangala NagaDivya	9/23/2025	9/24/2025	9/2/2025	-	Vangala NagaDivya	test
	A0004	Vangala NagaDivya	9/23/2025	9/24/2025	9/26/2025	-	Vangala NagaDivya	test
	A0008	Vangala NagaDivya	9/24/2025	9/25/2025	9/26/2025	-	Vangala NagaDivya	test
	A0013	Vangala NagaDivya	9/25/2025	9/26/2025	9/27/2025	-	Vangala NagaDivya	Fever
	A0007	Vangala NagaDivya	9/24/2025	9/25/2025	9/26/2025	-	John Mikaelson	test
	A0005	Vangala NagaDivya	9/24/2025	9/25/2025	9/29/2025	Take a Rest	John Mikaelson	test
	A0009	Vangala NagaDivya	9/24/2025	9/27/2025	9/29/2025	Take Rest	John Mikaelson	Fever
	A0006	Vangala NagaDivya	9/24/2025	9/25/2025	9/26/2025	test	John Mikaelson	test
	A0010	Vangala NagaDivya	9/24/2025	9/25/2025	9/27/2025	work	Nandhu Vengala	Going to family trip
	A0011	Vangala NagaDivya	9/24/2025	9/26/2025	9/27/2025	jhg	Nandhu Vengala	dfg
	A0012	Vangala NagaDivya	9/24/2025	9/27/2025	9/28/2025	test	Sweetly Pandey	test
Subtotal								
Total (14)								

Row Counts Detail Rows Subtotals Grand Total

24°C Mostly cloudy 23:27 ENG IN 25-09-2025

Leave Requests Report

Record Count: 14

Leave Request: Owner Name: Vangala NagaDivya

View Report (Leave Requests Report) As of Sep 25, 2025, 11:05 AM

Leave Requests Report

Record Count: 14

Leave Request: Owner Name: Vangala NagaDivya

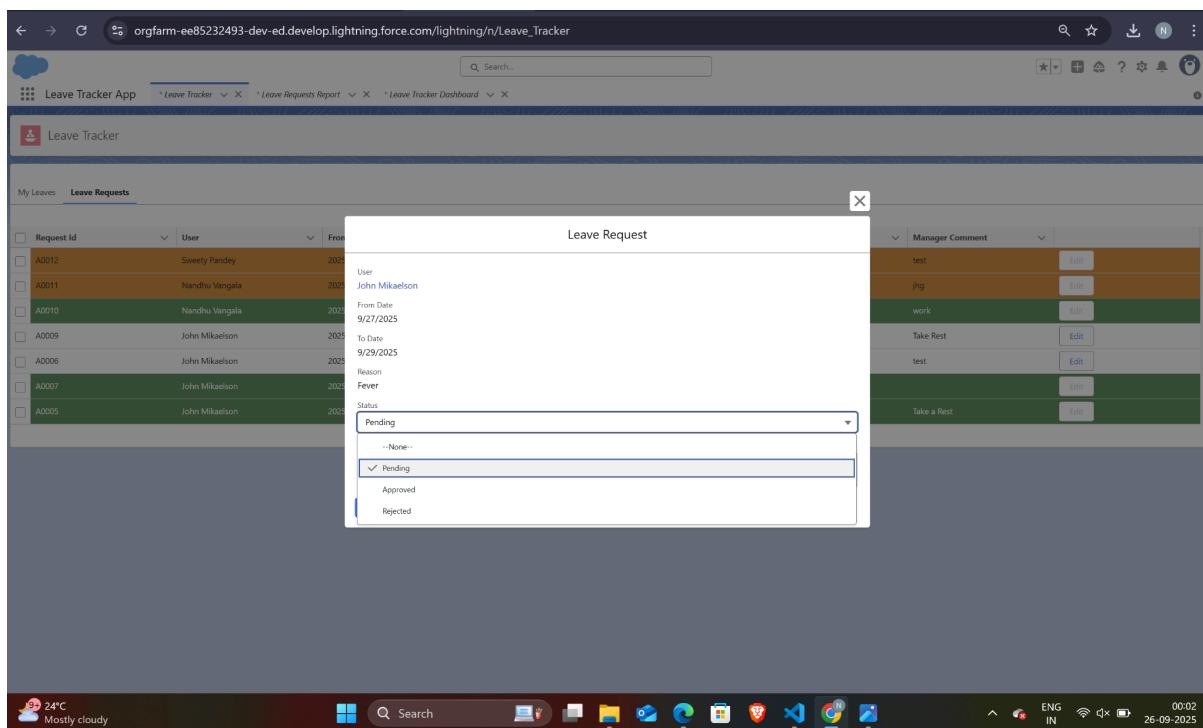
View Report (Leave Requests Report) As of Sep 25, 2025, 11:05 AM

Leave Requests Report

Leave Request: Leave Request Id	Leave Request: Created By	Leave Request: Create Date	Pro...	To D...
A0000	Vangala NagaDivya	9/23/2025	3/10/21	3/11/21
A0001	Vangala NagaDivya	9/23/2025	3/15/21	3/15/21
A0002	Vangala NagaDivya	9/23/2025	3/19/21	3/19/21
A0003	Vangala NagaDivya	9/23/2025	9/24/21	9/22/20
A0004	Vangala NagaDivya	9/23/2025	9/24/21	9/26/21
A0005	Vangala NagaDivya	9/24/2025	9/25/21	9/26/21
A0006	Vangala NagaDivya	9/24/2025	9/25/21	9/26/21

View Report (Leave Requests Report) As of Sep 25, 2025, 11:05 AM

24°C Mostly cloudy 23:36 ENG IN 25-09-2025

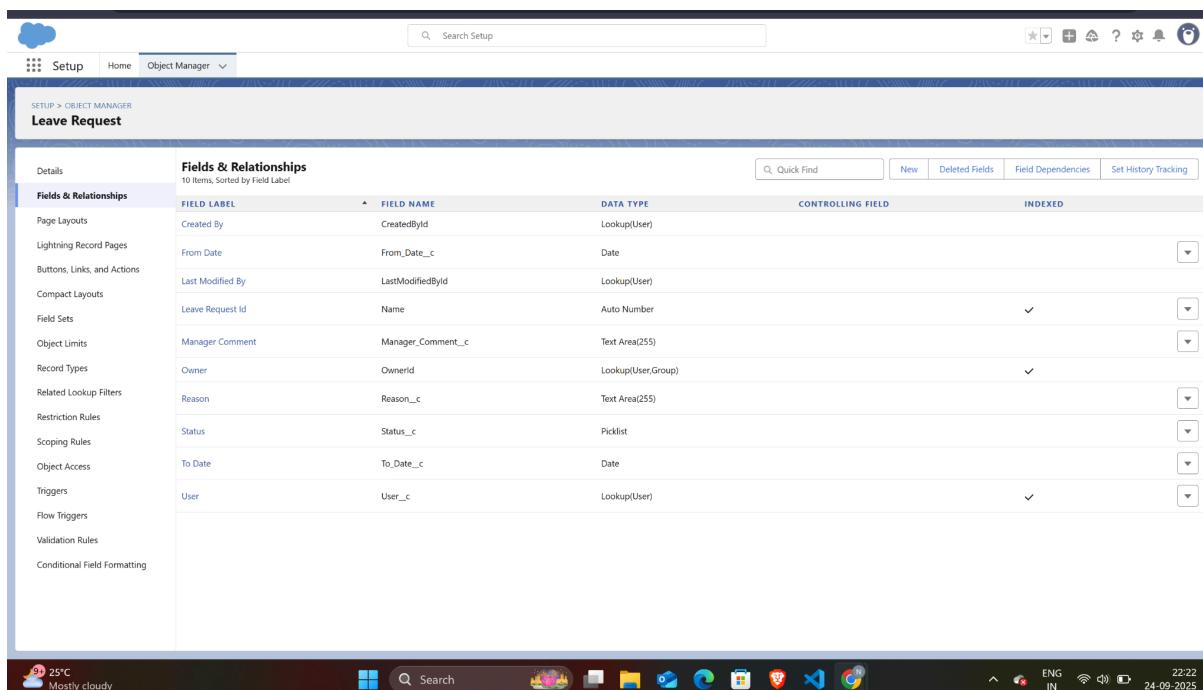


3. Handoff Documentation

Deliverables prepared for smooth transition:

- **System Design Document:**

- Object Model (LeaveRequest__c, fields, relationships).
- Flow diagrams (Approval process, Notifications).
- Security & sharing rules.



The screenshot shows the Salesforce Sharing Settings page. The left sidebar has a search bar and navigation links for Setup, Home, and Object Manager. Under the Security section, 'Sharing Settings' is selected. A message says 'Did you find what you're looking for? Try using Global Search.' The main area displays a table of sharing rules for various objects. One row for 'Leave Request' is highlighted with a red box. The table columns include the object name, sharing rule type (e.g., Public Read/Write, Private), and access level (e.g., Private). At the bottom, there are 'Save' and 'Cancel' buttons, and checkboxes for 'Standard Report Visibility', 'Manual User Record Sharing', 'Manager Groups', 'Secure guest user record access', and 'Require permission to view record names in lookup fields'. The status bar at the bottom shows 'PAK - BAN Live', system icons, and the date '25-09-2025'.

- **User Guide:**
 - How employees apply leave.
 - How managers approve/reject requests.
 - How admins monitor reports & dashboards.
- **Admin Notes:**
 - Adding new fields or modifying approval flows.
 - Managing email templates & notifications.
 - Deployment steps using Change Sets/SFDX.

4.Demo Video Link:

Leave Tracking App Demo Video

https://drive.google.com/file/d/1PoZ1Ymzcv0aMV0SwW4eDw-JZz8aD-wDr/view?usp=drive_link

Phase 10 Outcome:

- A polished **Leave Tracker App** ready for real users.
- Stakeholders clearly see the **value delivered**.
- Project closed with **complete demo + documentation** for long-term success.

Done By
Vangala NagaDivya