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 Hub 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

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.

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.

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.

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.

Track Leave History

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

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.

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.

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).

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.

3. 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.

4. AppExchange Exploration

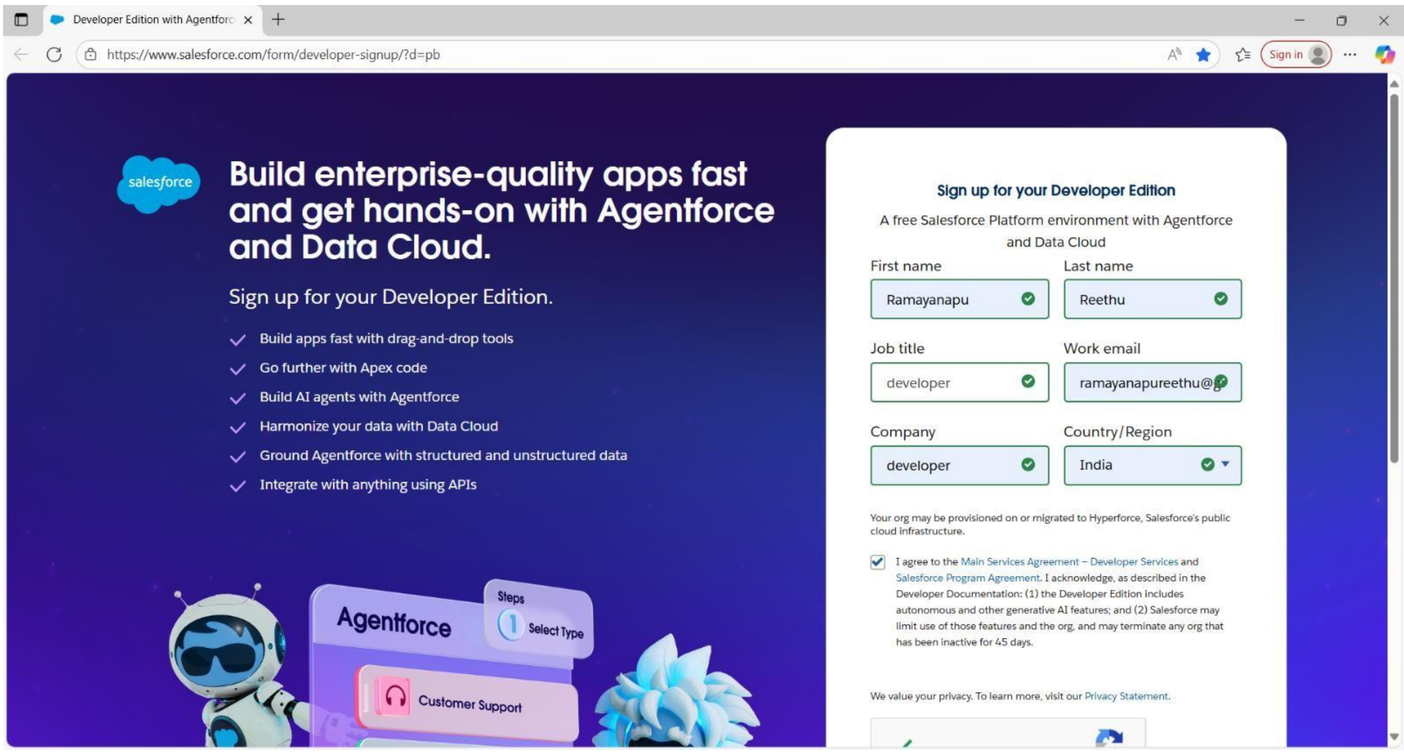
Before building from scratch, exploring **Salesforce AppExchange** provides insights into existing solutions and accelerators:

- **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 costeffective.
- **Decision:** Build a **custom Leave Hub 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 Hub App

OrgPreparation:
Spin up a **Developer Edition / Sandbox / Trailhead Playground**.


Developer Edition with Agentforc

https://www.salesforce.com/form/developer-signup/?d=pb

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- Build apps fast with drag-and-drop tools
- Go further with Apex code
- Build AI agents with Agentforce
- Harmonize your data with Data Cloud
- Ground Agentforce with structured and unstructured data
- Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Ramayanapu	Last name: Reethu
Job title: developer	Work email: ramayanapureethu@g
Company: developer	Country/Region: India

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

☑ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our Privacy Statement.

2.Company Setup:

- Organization Name: Leave Tracker (example for your project).
- Default Time Zone: IST (India Standard Time) to match employee location.
- Default Currency: INR (₹) for leave-related reporting (if linked with payroll).
- Language Settings: English (default), with multilingual support possible if required.

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)
        ■ Status (Picklist: Draft, Submitted, Approved, Rejected, Cancelled)
        ■ Reason (Long Text Area)

2. **Leave Balance**

   Fields:

        ■ Employee (Lookup → User/Employee)
        ■ Leave Type (Picklist)
        ■ Available Balance (Number)
        ■ Taken (Number)
        ■ Remaining Balance (Formula)

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.

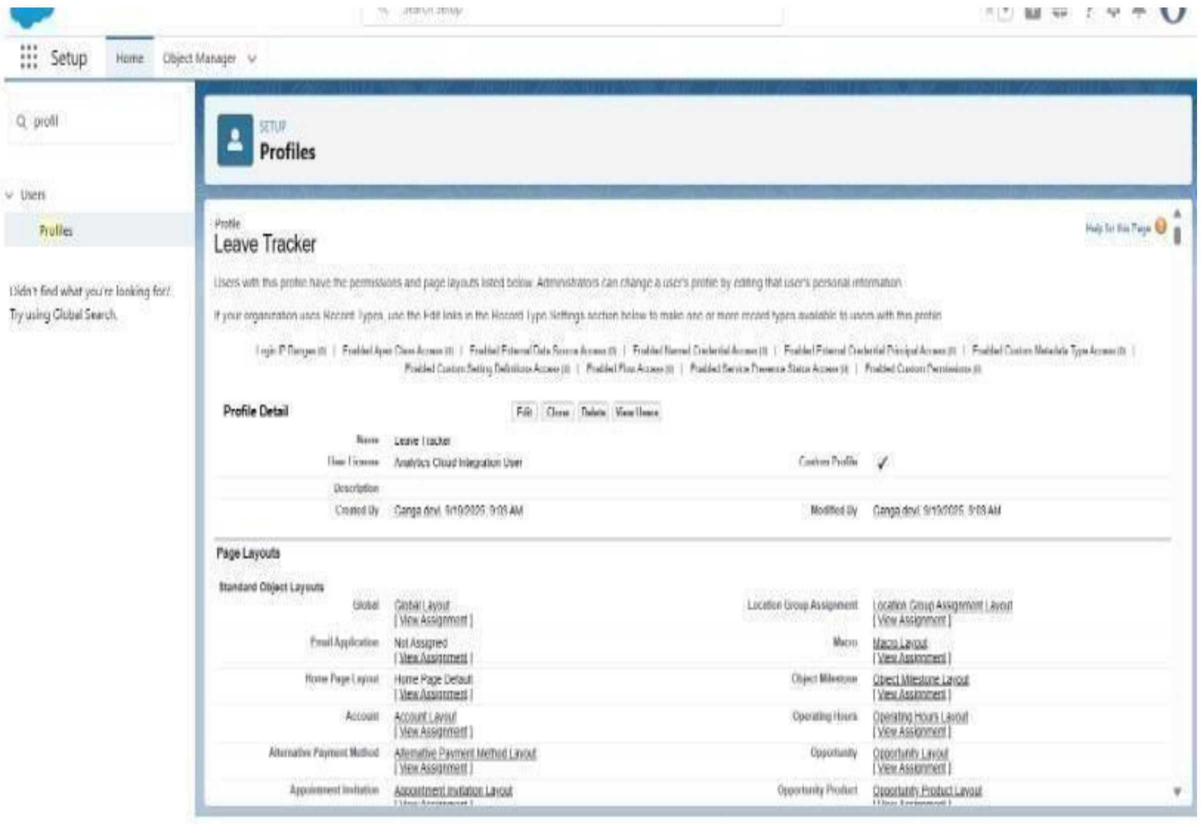
Record-Triggered Flows

- On leave approval → Deduct from Leave Balance.
- On rejection → Leave Balance remains unchanged.

5.Security & Access

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

○ HR/Admin: Full access.


Setup Home Object Manager

profil

Users
Profiles

Didn't find what you're looking for? Try using Global Search.

SETUP
Profiles

Profile
Leave Tracker

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges (0) | Enabled Apex Class Access (0) | Enabled External Data Source Access (0) | Enabled Named Credential Access (0) | Enabled External Credential Principal Access (0) | Enabled Custom Metadata Type Access (0) | Enabled Custom Setting Definitions Access (0) | Enabled Flow Access (0) | Enabled Service Presence Status Access (0) | Enabled Custom Permissions (0)

Profile Detail Edit Clone Delete View Users

Name	Leave Tracker		
User License	Analytics Cloud Integration User	Custom Profile	✓
Description			
Created By	Ganga devi, 9/10/2025, 9:03 AM	Modified By	Ganga devi, 9/10/2025, 9:03 AM

Page Layouts

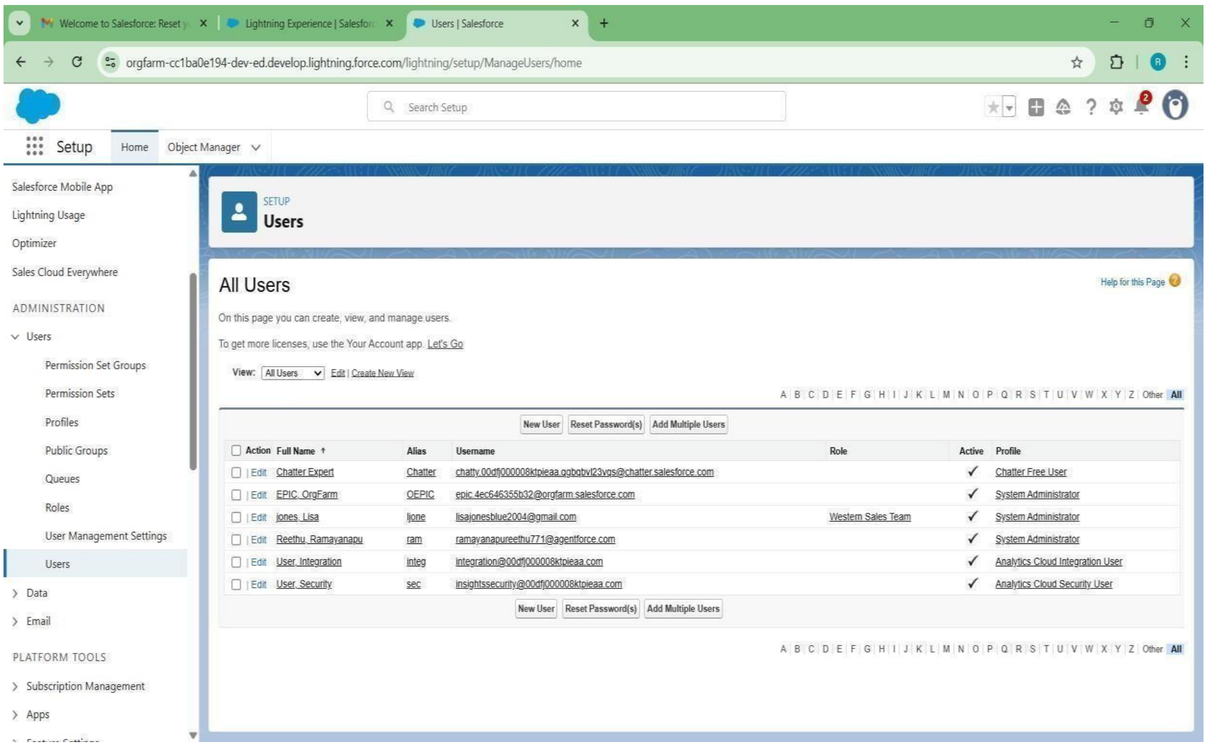
Standard Object Layouts

Global	Global Layout [View Assignment]	Location Group Assignment	Location Group Assignment Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Macro	Macro Layout [View Assignment]
Home Page Layout	Home Page Default [View Assignment]	Object Milestone	Object Milestone Layout [View Assignment]
Account	Account Layout [View Assignment]	Operating Hours	Operating Hours Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Opportunity	Opportunity Layout [View Assignment]
Appointment Invitation	Appointment Invitation Layout	Opportunity Product	Opportunity Product Layout

- **Users:**

○ Used to test the profiles, Roles, permission sets.

○ The user is used to edit the leave request to accepted , Rejected.



All Users

On this page you can create, view, and manage users.

To get more licenses, use the Your Account app. Let's Go

View: All Users Edit | Create New View

Action	Full Name ↑	Alias	Username	Role	Active	Profile
Edit	Chatter Expert	Chatter	chatty.00dfj000008ktpieaa.qgbqbvl23vqs@chatter.salesforce.com		✓	Chatter Free User
Edit	EPIC, OrgFarm	OEPIC	epic.4ec646355b32@orgfarm.salesforce.com		✓	System Administrator
Edit	jones, Lisa	ljone	lisajonesblue2004@gmail.com	Western Sales Team	✓	System Administrator
Edit	Reethu, Ramayanapu	ram	ramayanapureethu771@agentforce.com		✓	System Administrator
Edit	User, Integration	integ	integration@00dfj000008ktpieaa.com		✓	Analytics Cloud Integration User
Edit	User, Security	sec	insightssecurity@00dfj000008ktpieaa.com		✓	Analytics Cloud Security User

● **Sharing Rules** ○ Leave Requests visible only to employee, manager, and HR.

○ HR/Admin has access to all employee records.

5. 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.

6. Reports & Dashboards

- Reports:
  - Leave Requests by Employee.
  - Leave Balances Remaining.
  - Approved vs Rejected Leaves.
- Dashboard:
  - HR Overview (Total Leaves Taken, Team Availability).

7. 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 Hub

1. Standard & Custom Objects

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

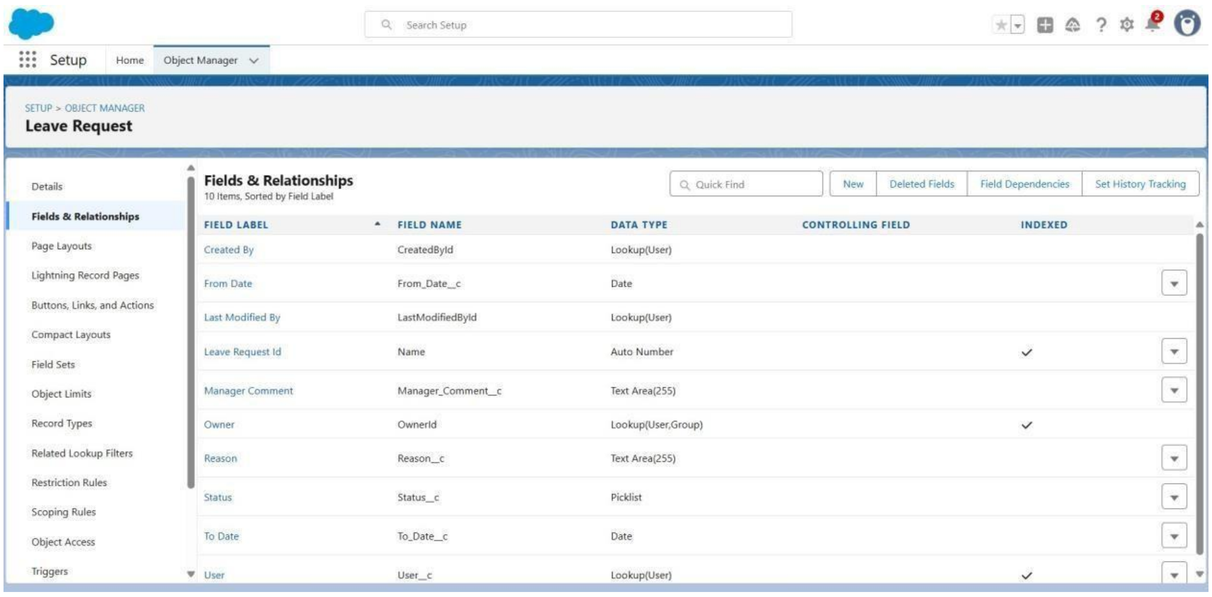
2. Fields

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

3. Record Types

- **Leave Request Record Types**:
  - Sick Leave
  - Casual Leave

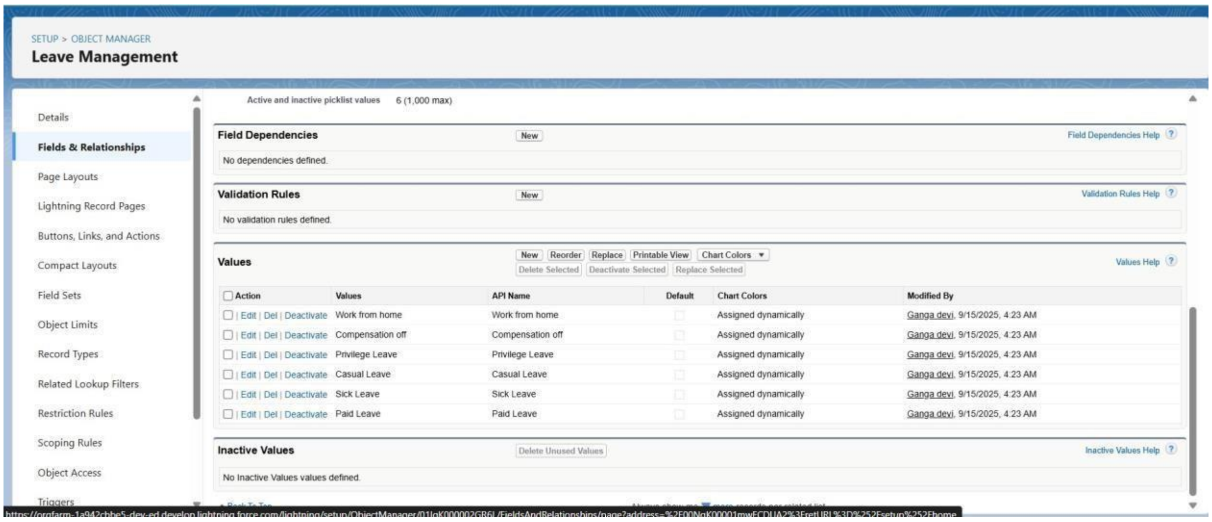
    ○ Earned Leave


SETUP > OBJECT MANAGER
Leave Request

Fields & Relationships
10 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
From Date	From_Date__c	Date		
Last Modified By	LastModifiedById	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)		✓

    ○ Special Leave(ex:Maternity,Paternity)
    ○ Work from home


SETUP > OBJECT MANAGER
Leave Management

Active and inactive picklist values 6 (1,000 max)

Field Dependencies
No dependencies defined.

Validation Rules
No validation rules defined.

Values

Action	Values	API Name	Default	Chart Colors	Modified By
Edit | Del | Deactivate	Work from home	Work from home		Assigned dynamically	Ganga devi, 9/15/2025, 4:23 AM
Edit | Del | Deactivate	Compensation off	Compensation off		Assigned dynamically	Ganga devi, 9/15/2025, 4:23 AM
Edit | Del | Deactivate	Privilege Leave	Privilege Leave		Assigned dynamically	Ganga devi, 9/15/2025, 4:23 AM
Edit | Del | Deactivate	Casual Leave	Casual Leave		Assigned dynamically	Ganga devi, 9/15/2025, 4:23 AM
Edit | Del | Deactivate	Sick Leave	Sick Leave		Assigned dynamically	Ganga devi, 9/15/2025, 4:23 AM
Edit | Del | Deactivate	Paid Leave	Paid Leave		Assigned dynamically	Ganga devi, 9/15/2025, 4:23 AM

Inactive Values
No Inactive Values values defined.

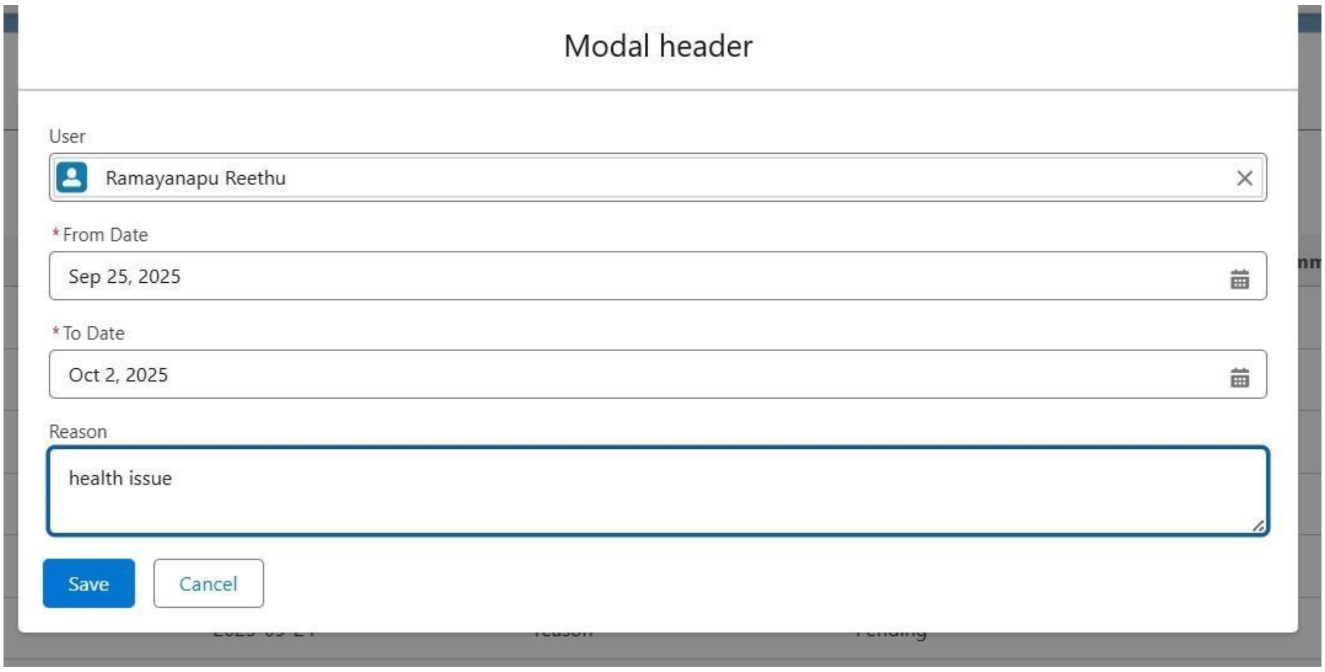
- Record Types allow different **picklist values, page layouts, and approval processes** for each leave type.

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.


Modal header

User

Ramayanapu Reethu

*From Date

Sep 25, 2025

*To Date

Oct 2, 2025

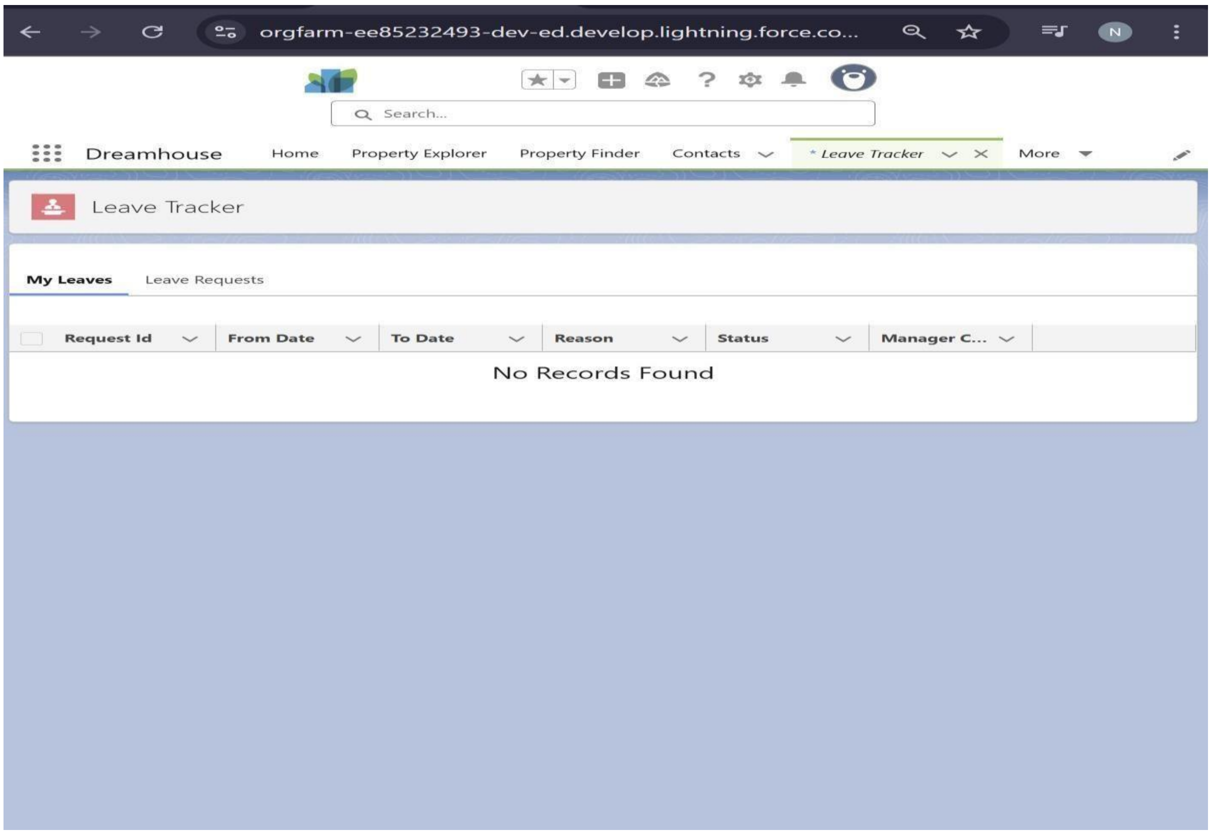
Reason

health issue

- 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.



Request Id	From Date	To Date	Reason	Status	Manager C...

No Records Found

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 *(Optional)*

- 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 4: Process Automation (Admin)

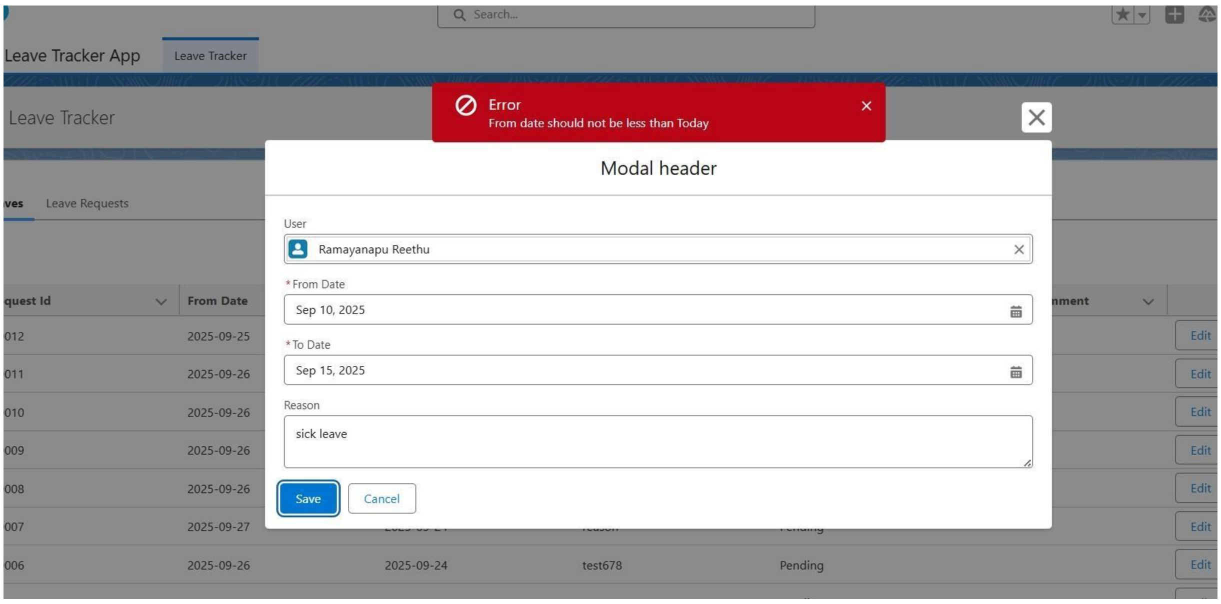
1. Validation Rules:

1. From Date should not be in the past

- **Rule Name:** From_Date_Not_Past
- **Error Message:** “End Date must be later than Start Date.” • **Error Condition Formula:**

```
if (new Date (fields.From_Date__c)>new Date(fields.To_Date__c)){
this.ShowToast('From date should not be grater than to date','Error'
,'error');          }
```

- **Error Location:** Field →Top



Leave Tracker App | Leave Tracker

Leave Tracker

Error
From date should not be less than Today

Modal header

User
Ramayanapu Reethu

*From Date
Sep 10, 2025

*To Date
Sep 15, 2025

Reason
sick leave

Save | Cancel

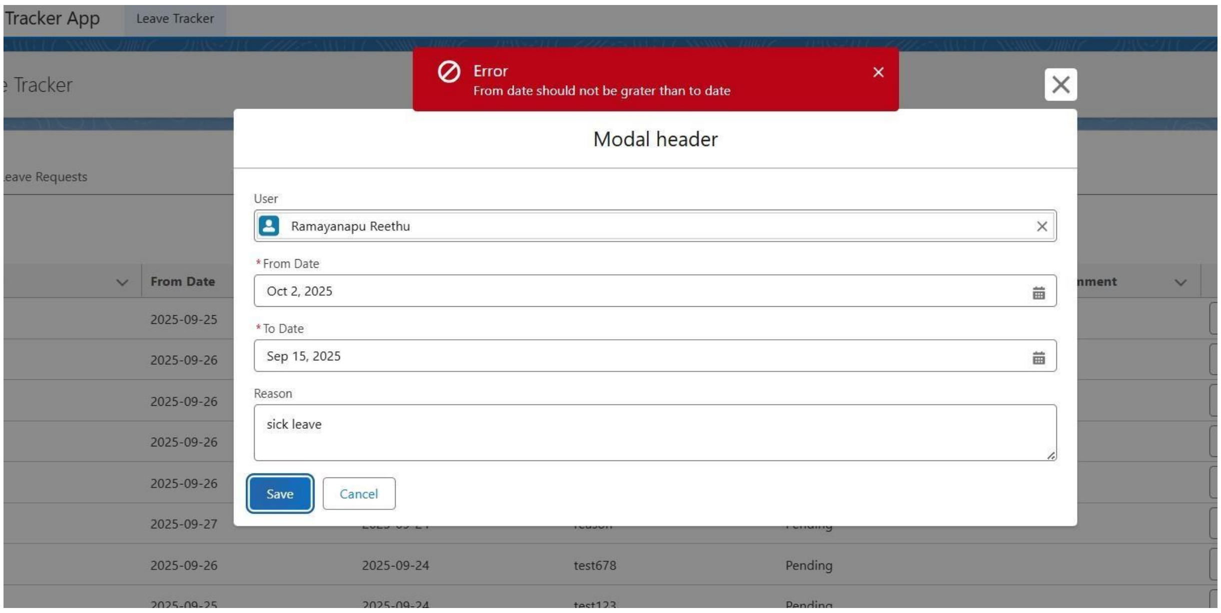
quest Id	From Date				nment	
012	2025-09-25					Edit
011	2025-09-26					Edit
010	2025-09-26					Edit
009	2025-09-26					Edit
008	2025-09-26					Edit
007	2025-09-27					Edit
006	2025-09-26	2025-09-24	test678	Pending		Edit

2. From Date should not be leassthan today:

- **Rule Name:** From_Date_Not_be lessthan today.
- **Error Message:** “End Date must be later than Start Date.”
- **Error Condition Formula:**

```
else if(new  Date()>new Date(fields.From_Date__c)){
            this.ShowToast('From date should not be less than Today','Error','error');
        }
```

Error Location: Field →Top


Tracker App | Leave Tracker

Error
From date should not be grater than to date

Modal header

User: Ramayanapu Reethu

*From Date: Oct 2, 2025

*To Date: Sep 15, 2025

Reason: sick leave

From Date			
2025-09-25			
2025-09-26			
2025-09-26			
2025-09-26			
2025-09-26			
2025-09-27			
2025-09-26	2025-09-24	test678	Pending
2025-09-25	2025-09-24	test123	Pending

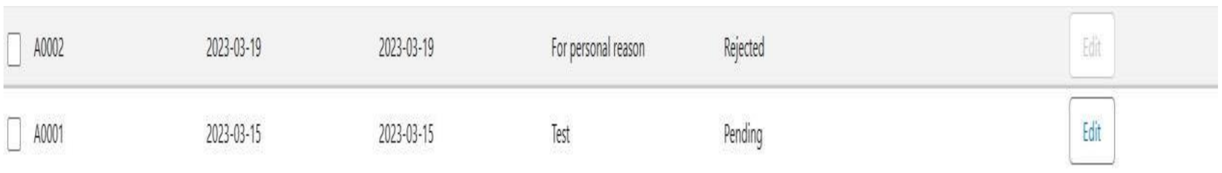
2. Workflow Rules:

1.Workflow for Approved Status

- **Purpose:** Notify the manager when a leave request is created and status is Approved.
- **Evaluation Criteria:** created, and every time it's edited to meet criteria

2. Workflow for Rejected Status

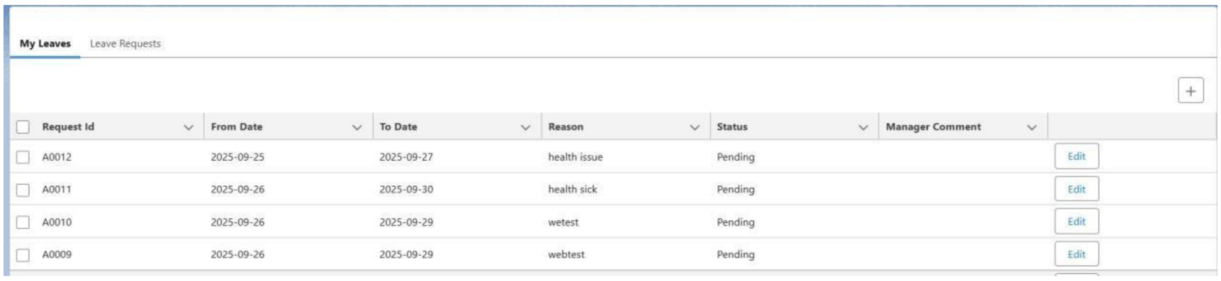
- **Purpose:** Notify the employee if their leave request is **Rejected**.
- **Evaluation Criteria:** created, and every time it's edited to meet criteria.



A0002	2023-03-19	2023-03-19	For personal reason	Rejected	Edit
A0001	2023-03-15	2023-03-15	Test	Pending	Edit

3. Workflow for Pending Status

- **Purpose:** Notify the employee if their leave request is **Pending**.
- **Evaluation Criteria:** created, and every time it's edited to meet criteria



Request Id	From Date	To Date	Reason	Status	Manager Comment	
A0012	2025-09-25	2025-09-27	health issue	Pending		Edit
A0011	2025-09-26	2025-09-30	health sick	Pending		Edit
A0010	2025-09-26	2025-09-29	wetest	Pending		Edit
A0009	2025-09-26	2025-09-29	webtest	Pending		Edit

```
 @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.error('Error occurred while fetching my leaves: ', result.error);         }
   }
```

3. Process Builder

On Leave Request Submission:

- Process Builder is used for conditional automation that cannot be handled by simple workflow rules. Example:
- If Reason = Valid -> Leave is Approved
- If Reason = InValid -> Leave is Rejected

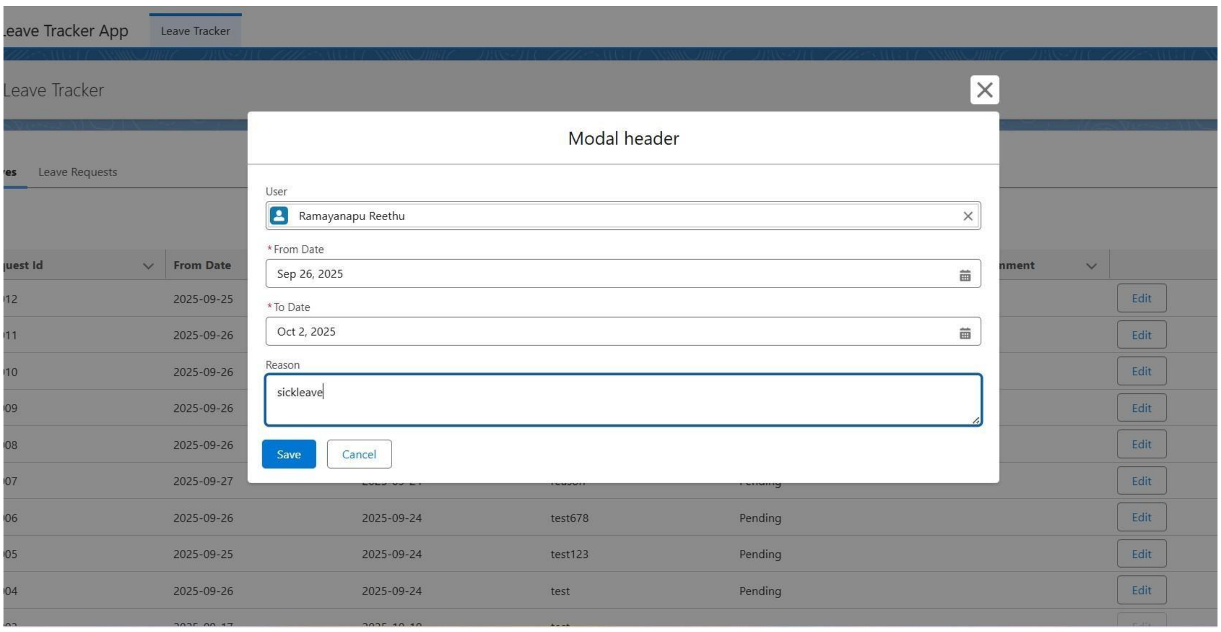
4. Flow Builder

Screen Flows (for employees):

Guided process to apply for leave.

- Step 1: User.

- Step 2: Pick Dates.
- Step 3: Provide Reason.
- Step 4: Review & Submit.



Modal header

User
Ramayanapu Reethu

* From Date
Sep 26, 2025

* To Date
Oct 2, 2025

Reason
sickleave

Save | Cancel

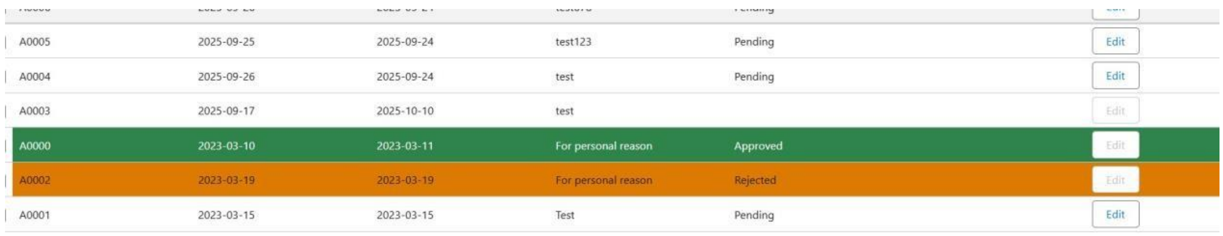
uest Id	From Date				nment	
12	2025-09-25					Edit
11	2025-09-26					Edit
10	2025-09-26					Edit
09	2025-09-26					Edit
08	2025-09-26					Edit
07	2025-09-27					Edit
06	2025-09-26	2025-09-24	test678	Pending		Edit
05	2025-09-25	2025-09-24	test123	Pending		Edit
04	2025-09-26	2025-09-24	test	Pending		Edit

5. Approval Process

- Step 1: Employee submits request → Status = Pending.
- Step 2: Manager receives approval request.
- Step 3: Manager approves or rejects.
- Step 4: If approved → Send email to employee.
- Step 5: If rejected → Send email to employee.

6.Field Updates

- On Approval: Status → Approved.
- On Rejection: Status → Rejected.
- On Cancellation: Status → Pending.



A0005	2025-09-25	2025-09-24	test123	Pending	Edit
A0004	2025-09-26	2025-09-24	test	Pending	Edit
A0003	2025-09-17	2025-10-10	test		Edit
A0000	2023-03-10	2023-03-11	For personal reason	Approved	Edit
A0002	2023-03-19	2023-03-19	For personal reason	Rejected	Edit
A0001	2023-03-15	2023-03-15	Test	Pending	Edit

Phase 5: Apex Programming (Developer)

1. Classes & Objects

- LeaveRequestController: Handles leave application logic (submit, update, approve, reject).

- Utility Classes: For reusable logic like date validation, string formatting, and error handling. **2. Apex Triggers**

Before Insert/Update:

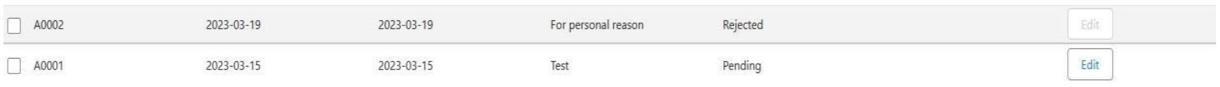
- Validate leave dates (From_Date ≤ To_Date).
- Prevent overlapping leave requests.

After Insert:

- Notify manager of new leave request submission.

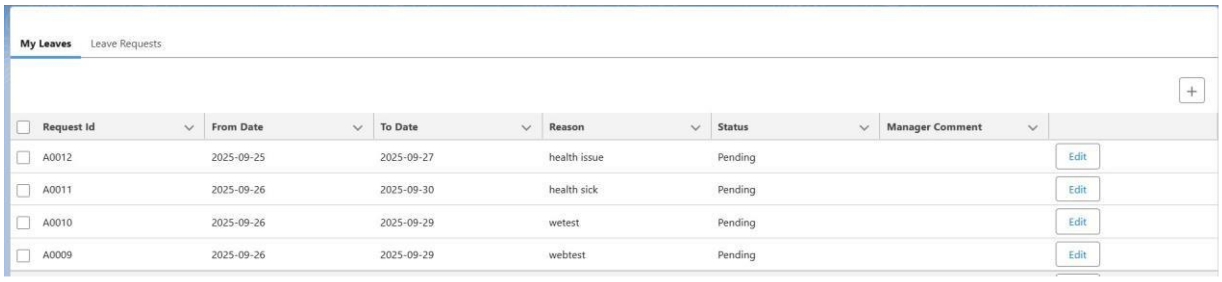
After Update:

- If Leave is approved change the status of the leave whether it is approved.
- If Leave is rejected change the status of the leave whether it is rejected.



A0002	2023-03-19	2023-03-19	For personal reason	Rejected	Edit
A0001	2023-03-15	2023-03-15	Test	Pending	Edit

- If Leave is in pending change the status of the leave whether it is Pending.



My Leaves | Leave Requests

Request Id	From Date	To Date	Reason	Status	Manager Comment	
A0012	2025-09-25	2025-09-27	health issue	Pending		Edit
A0011	2025-09-26	2025-09-30	health sick	Pending		Edit
A0010	2025-09-26	2025-09-29	wetest	Pending		Edit
A0009	2025-09-26	2025-09-29	webtest	Pending		Edit

3. Trigger Design Pattern

Handler Class Pattern followed:

- One trigger per object (OnsubmitHandler).
- Delegates logic to OnsubmitHandler class.
- Improves readability, reusability, and testability.

4. SOQL & SOSL Usage:

SOQL (Salesforce Object Query Language) is used to fetch records from Salesforce objects. In your code:

```
<lightning-datatable key-field="Id" data={myLeaves}
columns={columns}onrowaction={rowActionHandler}></lightning-datatable>
```

- **data={myLeaves}** → This property is populated by a **JavaScript @wire or Apex method** that fetches leave request records.
- Example Apex method:

```
@AuraEnabled(cacheable=true) public static List<LeaveRequest__c>

getUserLeaves(Id userId) {

  return [

    SELECT Id, From_Date__c, To_Date__c, Status__c, Reason__c

    FROM LeaveRequest__c

    WHERE User__c = :userId

     ORDER BY From_Date__c DESC

  ];

}
```

- This SOQL query fetches all leave requests for the current user (`User__c = :userId`) to display in the datatable.
- The results populate `myLeaves` in the LWC and show each record in a row.

SOSL :

SOSL (Salesforce Object Search Language) is used for **searching across multiple objects or fields**, typically with a search term.

- In your current template, there is **no direct SOSL usage**.
- SOSL would be relevant if you had a **search bar** and wanted to find leave requests by keyword in fields like **Reason__c**, **Status__c**, or **Employee Name**.

Example SOSL in Apex :

```
@AuraEnabled(cacheable=true) public static List<LeaveRequest__c> searchLeaves(String

searchTerm) {

   List<List<SObject>> searchResults = [FIND :searchTerm IN ALL FIELDS RETURNING LeaveRequest__c(Id,
From_Date__c, To_Date__c, Reason__c, Status__c)];     return (List<LeaveRequest__c>)searchResults[0];

}
```

- Searches LeaveRequest__c records across all fields for the term entered by the user.
- Useful for a global search feature in your LWC.

3. Record Creation / Update:

```
   <lightning-record-edit-form object-api-name={objectApiName} record-id={recordId}
onsuccess={successHandler} onsubmit={submitHandler}>

   <lightning-input-field field-name="User__c" value={currentUserId}></lightning-inputfield>

   <lightning-input-field field-name="From_Date__c"></lightning-input-field>

   <lightning-input-field field-name="To_Date__c"></lightning-input-field>

   <lightning-input-field field-name="Reason__c"></lightning-input-field>

</lightning-record-edit-form>
```

What it does:

- Allows the user to **create or edit** a Leave Request record.
- record-id={recordId} → If empty, creates a new record; if filled, updates an existing record.
- onsuccess={successHandler} → Called after save to refresh data.

Relation to SOQL/SOSL:

- Not direct, but **under the hood**, Salesforce uses **SOQL** to fetch field data for the record when editing.
- Upon submission, Salesforce performs a **DML operation** (insert/update) on the object.

Control Statements:

- If-Else: Approve vs Reject logic.
- Checks if **From_Date__c > To_Date__c**.
- If true, shows an **error toast** → “From date should not be greater than To date.”

  Prevents invalid date ranges.

- Compares today’s date (new Date()) with the From Date.
- If From Date is **earlier than today**, shows an **error toast** → “From date should not be less than Today.”
- If both validations pass:
- Submits the form with updated fields (including Status__c = 'Pending').
- Uses this.refs.leaveRequestFrom (reference to the form) to perform the save operation.

```
 if (new Date(fields.From_Date__c)>new Date(fields.To_Date__c)){
               this.ShowToast('From date should not 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);
          }
```

5. Asynchronous Apex:

Automatically runs when the component loads or when reactive parameters change.

Salesforce handles caching and re-fetching.

You don’t write .then() or await; data comes through the result.

Best for read-only data that should auto-refresh (like a list of leave requests in a table).

```
@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.error('Error occurred while fetching my leaves:

', result.error);

    }
  }
```

6.Exception Handling

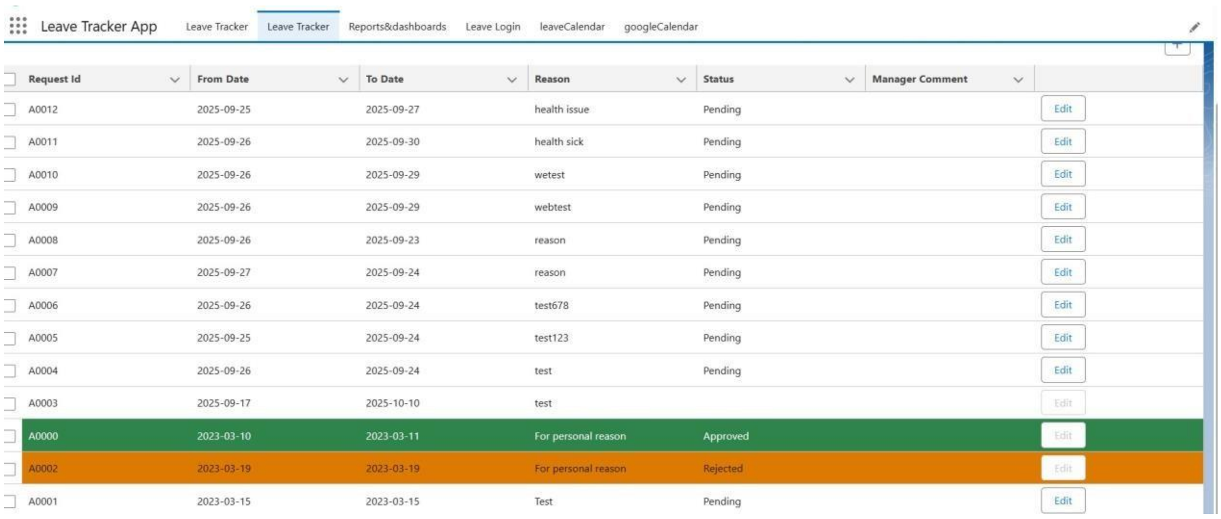
- Try-Catch Blocks: Handle DML and SOQL exceptions.

```
 try {
    update leaveRecord;   }
catch(DmlException e) {
      System.debug('Error: ' + e.getMessage());
 }
```

▪ Custom Exceptions: For business rules like “Insufficient Leave Balance.” ▪ Error Logging: Store errors in a custom object Error_Log__c for admin review.

Phase 6: User Interface Development 1. Lightning App Builder

- Designed a custom Leave Management App in Salesforce.
- App includes navigation tabs:
    - Home
    - My Leaves
    - Leave Requests
    - Reports & Dashboards
    - Leaves Calender
    - Google Calender



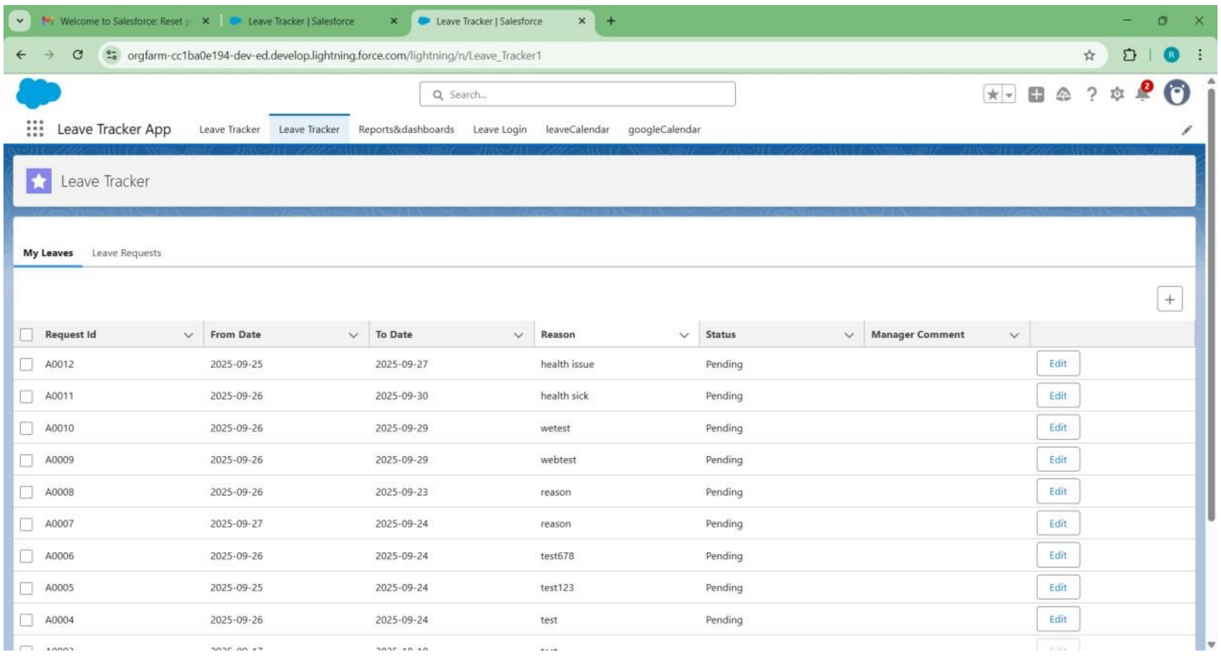
Request Id	From Date	To Date	Reason	Status	Manager Comment	
A0012	2025-09-25	2025-09-27	health issue	Pending		Edit
A0011	2025-09-26	2025-09-30	health sick	Pending		Edit
A0010	2025-09-26	2025-09-29	wetest	Pending		Edit
A0009	2025-09-26	2025-09-29	webtest	Pending		Edit
A0008	2025-09-26	2025-09-23	reason	Pending		Edit
A0007	2025-09-27	2025-09-24	reason	Pending		Edit
A0006	2025-09-26	2025-09-24	test678	Pending		Edit
A0005	2025-09-25	2025-09-24	test123	Pending		Edit
A0004	2025-09-26	2025-09-24	test	Pending		Edit
A0003	2025-09-17	2025-10-10	test			Edit
A0000	2023-03-10	2023-03-11	For personal reason	Approved		Edit
A0002	2023-03-19	2023-03-19	For personal reason	Rejected		Edit
A0001	2023-03-15	2023-03-15	Test	Pending		Edit

2. Record Pages

- Customized LeaveRequest__c Record Page to show:
    - Employee details
    - Leave type, dates, and reason
    - Manager comments
    - Approval/Rejection buttons (Quick Actions)
    - Compact layouts added for quick visibility of status.

3.Tabs

- Created custom tabs for:
    - Leave Request
    - My Leaves


Welcome to Salesforce: Reset y
Leave Tracker | Salesforce
Leave Tracker | Salesforce
orgfarm-cc1ba0e194-dev-ed.develop.lightning.force.com/lightning/n/Leave_Tracker1
Search...
Leave Tracker App
Leave Tracker
Leave Tracker
Reports&dashboards
Leave Login
leaveCalendar
googleCalendar

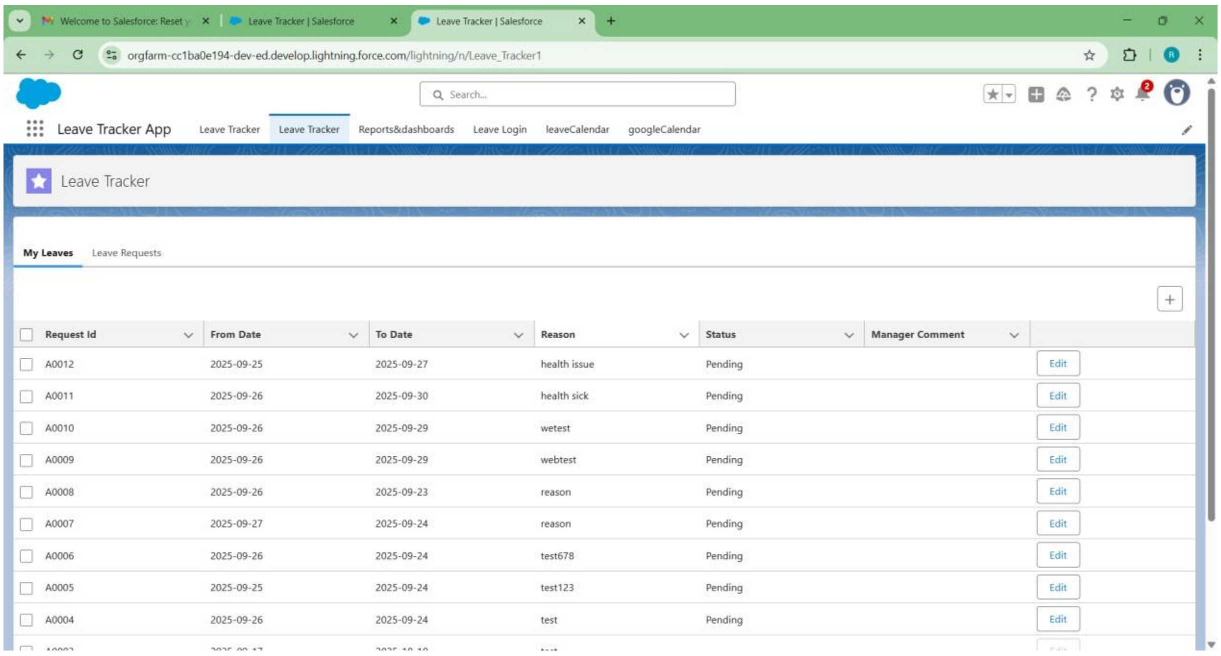
Leave Tracker

My Leaves | Leave Requests

Request Id	From Date	To Date	Reason	Status	Manager Comment	
A0012	2025-09-25	2025-09-27	health issue	Pending		Edit
A0011	2025-09-26	2025-09-30	health sick	Pending		Edit
A0010	2025-09-26	2025-09-29	wetest	Pending		Edit
A0009	2025-09-26	2025-09-29	webtest	Pending		Edit
A0008	2025-09-26	2025-09-23	reason	Pending		Edit
A0007	2025-09-27	2025-09-24	reason	Pending		Edit
A0006	2025-09-26	2025-09-24	test678	Pending		Edit
A0005	2025-09-25	2025-09-24	test123	Pending		Edit
A0004	2025-09-26	2025-09-24	test	Pending		Edit

4. Home Page Layouts

- Customized with:
    - Employee Leave Summary Chart
    - Quick Action: Apply for Leave
    - Recent Leave Requests list view


Leave Tracker

My Leaves | Leave Requests

Request Id	From Date	To Date	Reason	Status	Manager Comment	
A0012	2025-09-25	2025-09-27	health issue	Pending		Edit
A0011	2025-09-26	2025-09-30	health sick	Pending		Edit
A0010	2025-09-26	2025-09-29	wetest	Pending		Edit
A0009	2025-09-26	2025-09-29	webtest	Pending		Edit
A0008	2025-09-26	2025-09-23	reason	Pending		Edit
A0007	2025-09-27	2025-09-24	reason	Pending		Edit
A0006	2025-09-26	2025-09-24	test678	Pending		Edit
A0005	2025-09-25	2025-09-24	test123	Pending		Edit
A0004	2025-09-26	2025-09-24	test	Pending		Edit

5. Lightning Web Components (LWC)

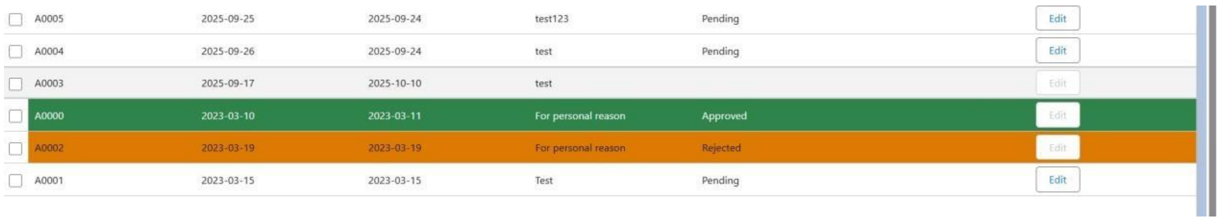
Built multiple LWCs to handle leave functionality:

MyLeaves

- Displays logged-in employee's leave history.
- Uses @wire to fetch data fromLeaveRequestController.getMyLeaves().

Color-coded statuses:

- `Approved  = Green`
- `Rejected  = Red`



A0005	2025-09-25	2025-09-24	test123	Pending	Edit
A0004	2025-09-26	2025-09-24	test	Pending	Edit
A0003	2025-09-17	2025-10-10	test		Edit
A0000	2023-03-10	2023-03-11	For personal reason	Approved	Edit
A0002	2023-03-19	2023-03-19	For personal reason	Rejected	Edit
A0001	2023-03-15	2023-03-15	Test	Pending	Edit

LeaveRequest

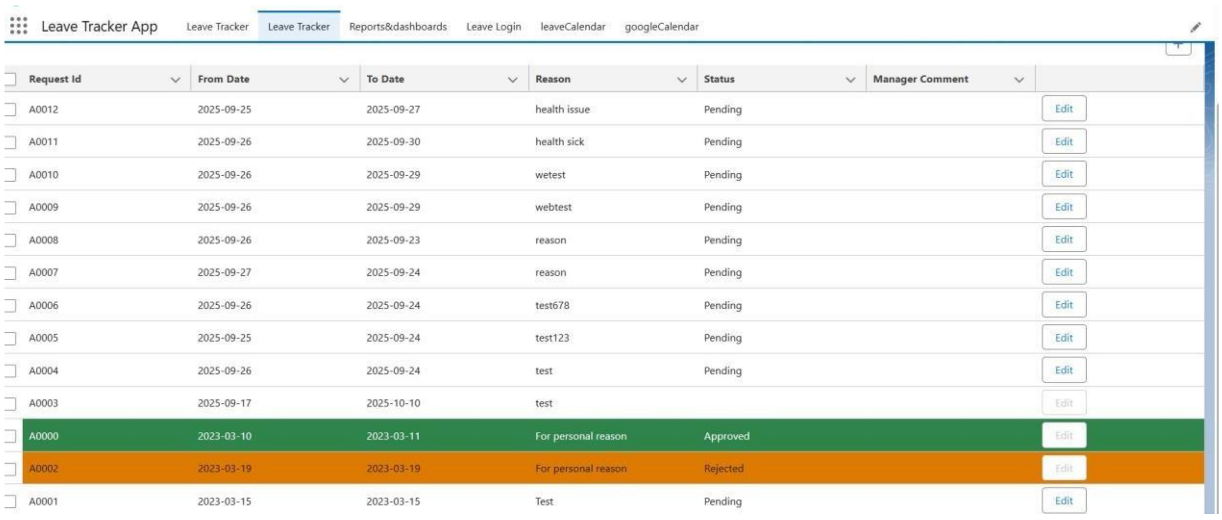
- Managers can view and act on leave requests.
- Buttons for Approve / Reject directly from the LWC.
- Sends updates via Apex controller + email notifications.

6. Apex with LWC

- Imperative Apex Calls: Used in applyLeave and leaveRequest for record insert/update.
- Wire Adapters: Used in myLeaves to fetch employee's leaves dynamically.

7. Navigation

- Used in LWC to navigate between:
  - Leave request record page
  - Leave dashboard reports
  - Apply Leave form
  - Login Page
  - Reports and dashboards
  - Leave Calender
  - Google Calender

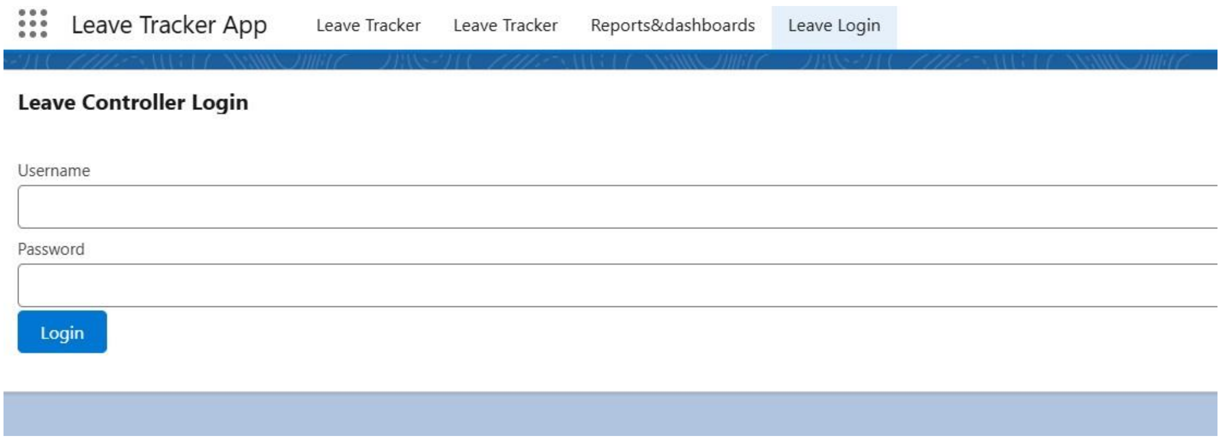


Request Id	From Date	To Date	Reason	Status	Manager Comment	
A0012	2025-09-25	2025-09-27	health issue	Pending		Edit
A0011	2025-09-26	2025-09-30	health sick	Pending		Edit
A0010	2025-09-26	2025-09-29	wetest	Pending		Edit
A0009	2025-09-26	2025-09-29	webtest	Pending		Edit
A0008	2025-09-26	2025-09-23	reason	Pending		Edit
A0007	2025-09-27	2025-09-24	reason	Pending		Edit
A0006	2025-09-26	2025-09-24	test678	Pending		Edit
A0005	2025-09-25	2025-09-24	test123	Pending		Edit
A0004	2025-09-26	2025-09-24	test	Pending		Edit
A0003	2025-09-17	2025-10-10	test			Edit
A0000	2023-03-10	2023-03-11	For personal reason	Approved		Edit
A0002	2023-03-19	2023-03-19	For personal reason	Rejected		Edit
A0001	2023-03-15	2023-03-15	Test	Pending		Edit

Phase 7: Integration & External Access

Integrations:

Login Page:



Leave Controller Login

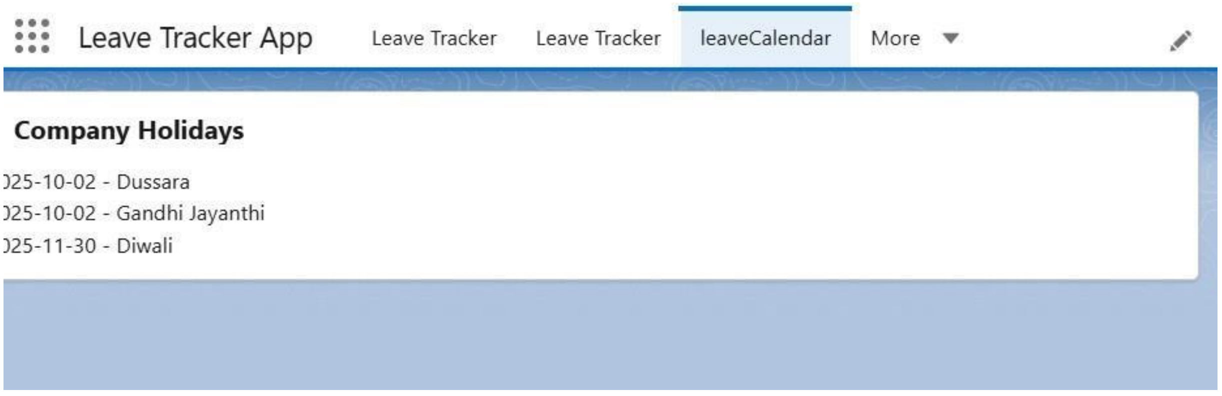
Username

Password

Login

Calendar Integrations:

- Google Calendar / Outlook Tab: Embed or sync employee calendars inside Salesforce.
- Purpose: Employees can see leave requests alongside their personal/work calendar to avoid conflicts.



Company Holidays

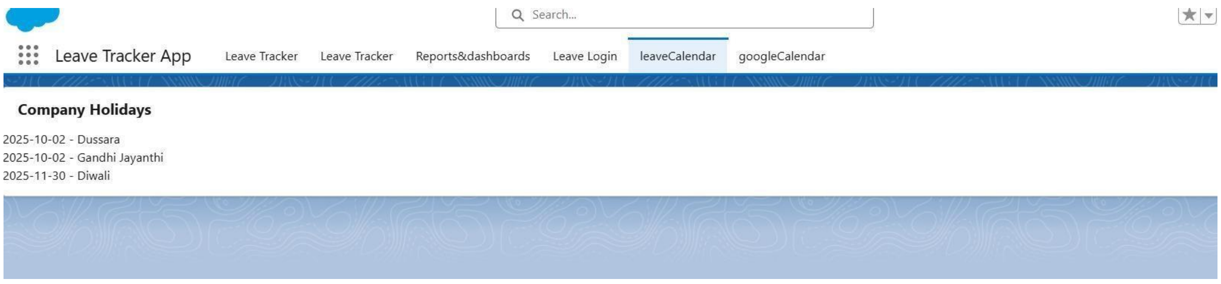
)25-10-02 - Dussara
)25-10-02 - Gandhi Jayanthi
)25-11-30 - Diwali

API-Based Integrations (Advanced)

- **Google Calendar API / Microsoft Graph API**:

  Automatically push approved leaves to external calendars using Apex or middleware.

- **Purpose:** Real-time syncing of leave events with company calendars.



Company Holidays

2025-10-02 - Dussara
2025-10-02 - Gandhi Jayanthi
2025-11-30 - Diwali

Named Credentials Setup

- Define Named Credentials in Salesforce to securely store external service URLs + authentication.
- Example use case: Connecting Salesforce to an HR system API for employee details.

External Services & Callouts

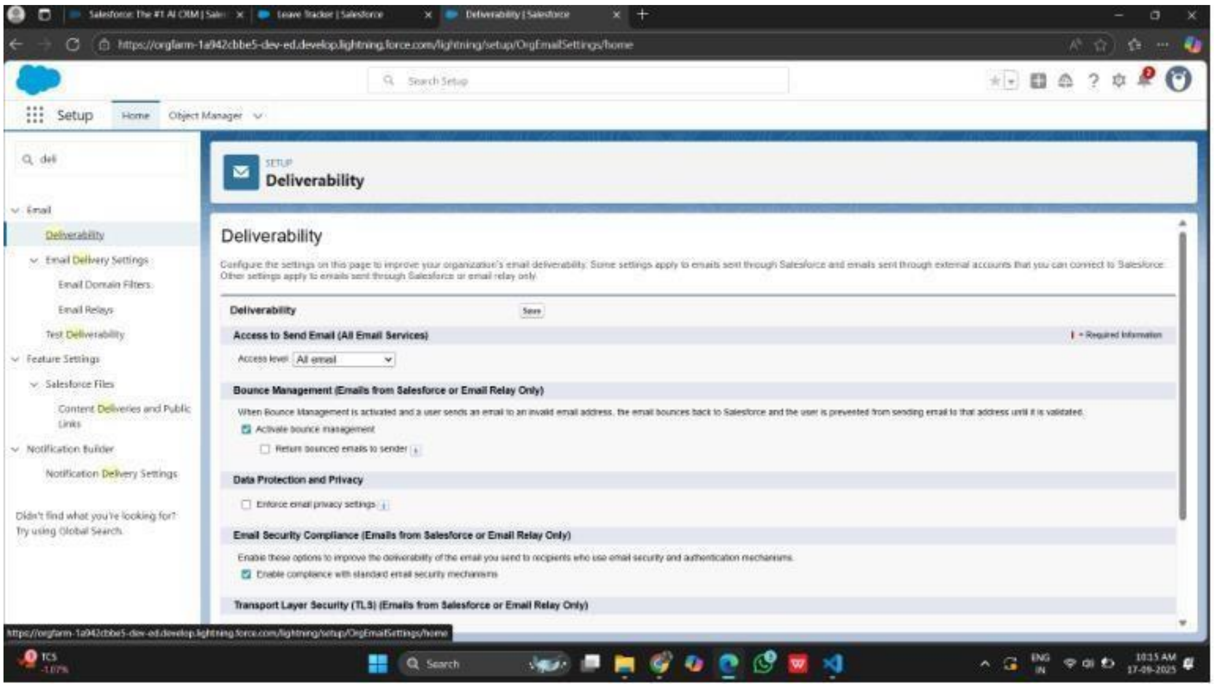
Demonstrate Apex HTTP Callout to fetch leave balance from an external HR system.

Sample Code (HTTP Callout):

```
 public with sharing class LeaveIntegrationService {
    public static void getLeaveBalance(String employeeId) {         Http
 http = new Http();

               HttpRequestreq.setEndpoint('callout:HR_System/leavebalance/' req = new

 HttpRequest();   + 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());        }

    } }
```

Email Service Integration

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



SETUP
Deliverability

Deliverability

Configure the settings on this page to improve your organization's email deliverability. Some settings apply to emails sent through Salesforce and emails sent through external accounts that you can connect to Salesforce. Other settings apply to emails sent through Salesforce or email relay only.

Deliverability

Access to Send Email (All Email Services) | = Required Information

Access level: All email

Bounce Management (Emails from Salesforce or Email Relay Only)

When Bounce Management is activated and a user sends an email to an invalid email address, the email bounces back to Salesforce and the user is prevented from sending email to that address until it is validated.

☑ Activate bounce management

☐ Return bounced emails to sender

Data Protection and Privacy

☐ Enforce email privacy settings

Email Security Compliance (Emails from Salesforce or Email Relay Only)

Enable these options to improve the deliverability of the email you send to recipients who use email security and authentication mechanisms.

☑ Enable compliance with standard email security mechanisms

Transport Layer Security (TLS) (Emails from Salesforce or Email Relay Only)

Platform Events for Real-time Updates

- Use Platform Events to notify external systems (like HR Payroll) when leave status changes.
- 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);
      }
   }
}
```

OAuth & Authentication

- If external API requires OAuth 2.0, connect via Named Credentials → OAuth.
- Secure token handling ensures safe communication.

Phase 8: Data Management & Deployment

Objective:

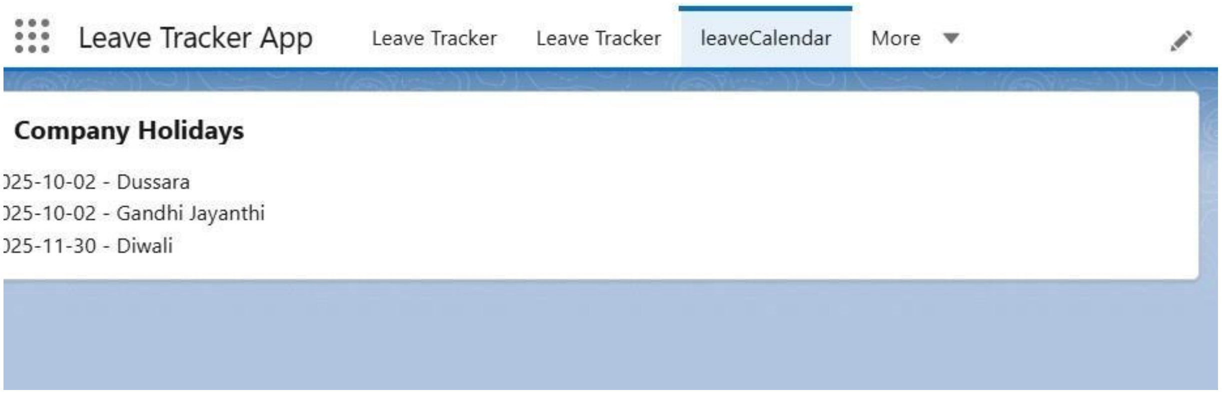
To ensure smooth and error-free deployment of Leave Tracker customizations from development to higher environments (UAT, Production) using best practices and tools.

Data Import Wizard

- Used to load initial employee data (Users, basic LeaveRequest__c records).
- Supports CSV files.
- Best for small/medium data volumes (< 50,000 records).

Calendar:

- Google Calendar / Outlook Tab: Embed or sync employee calendars inside Salesforce.
- Purpose: Employees can see leave requests alongside their personal/work calendar to avoid conflicts.

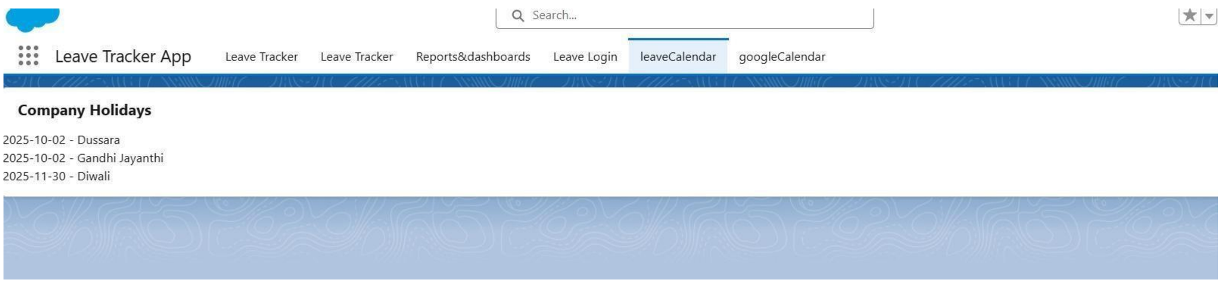


Company Holidays

)25-10-02 - Dussara
)25-10-02 - Gandhi Jayanthi
)25-11-30 - Diwali

API-Based Integrations (Advanced)

- Google Calendar API / Microsoft Graph API: Automatically push approved leaves to external calendars using Apex or middleware.
- Purpose: Real-time syncing of leave events with company calendars.

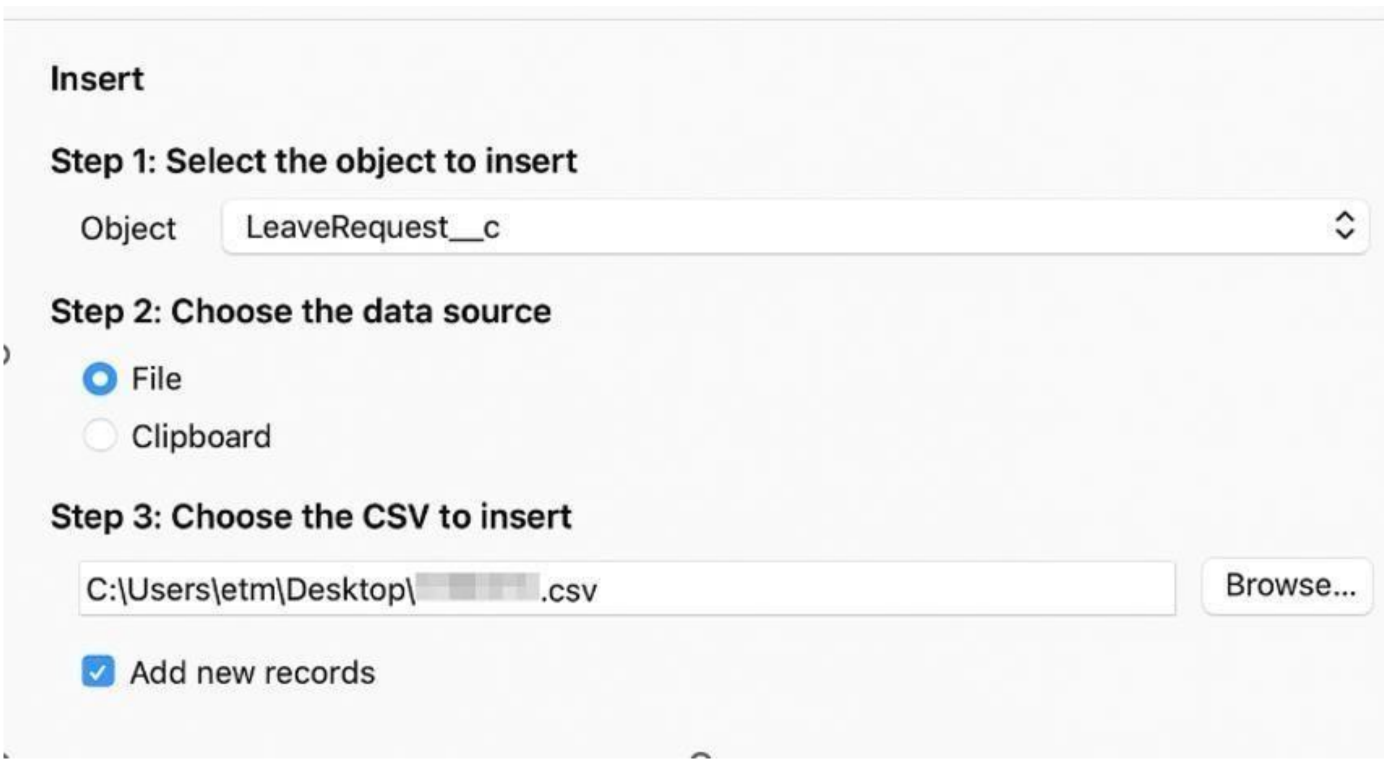


Company Holidays

2025-10-02 - Dussara
2025-10-02 - Gandhi Jayanthi
2025-11-30 - Diwali

Data Loader

- Used to perform bulk operations (insert, update, delete, upsert) for large datasets.
- Example: Bulk upload of historical leave records.
- Supports command-line interface for automation.



Insert

Step 1: Select the object to insert

Object: LeaveRequest__c

Step 2: Choose the data source

[✓ File] Clipboard

Step 3: Choose the CSV to insert

C:\Users\etm\Desktop\[redacted].csv

[✓] Add new records

Change Sets

Used Change Sets for deploying metadata between environments (Sandbox → Production).

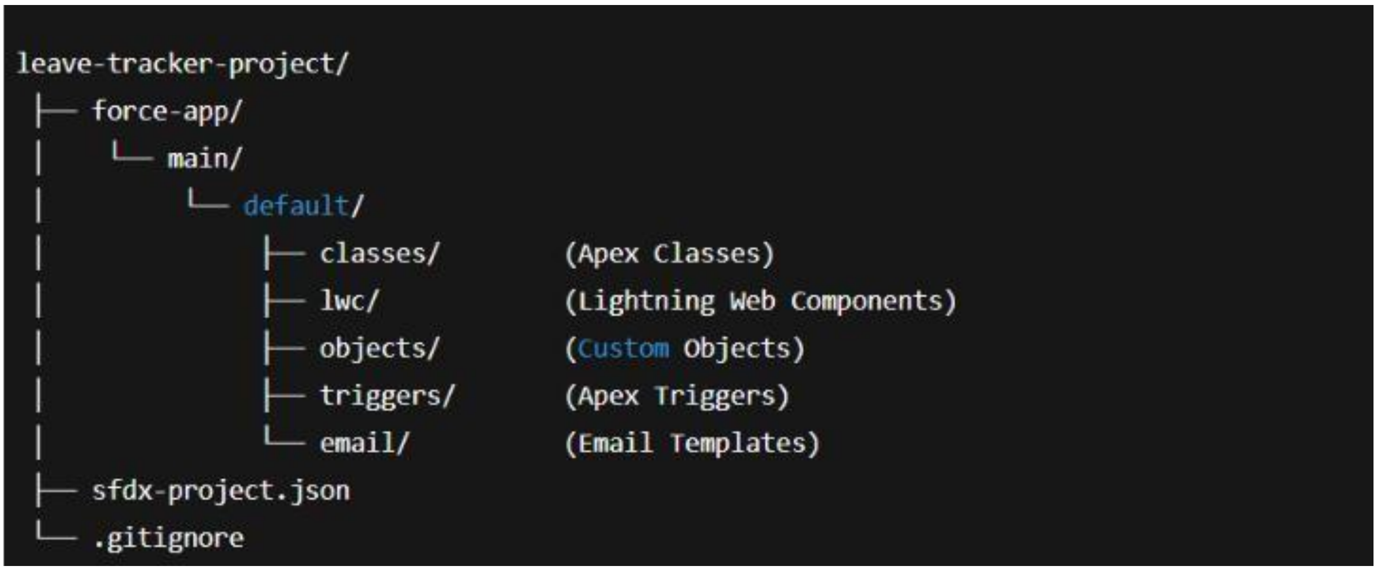
Included:

- Custom Objects (LeaveRequest__c)
- Apex Classes (LeaveRequestController, Triggers)
- Lightning Web Components (applyLeave, myLeaves, leaveRequest)
- Email Templates & Flows

VS Code Deployment Setup

Prerequisites

- Install Visual Studio Code.
- Install Salesforce CLI (SFDX).
- Install Salesforce Extension Pack in VS Code.



```
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
```

- Connect Org → sfdx force:auth:web:login -d -a DevHub.
- Sample Project Structure (VS Code)
- Retrieve metadata from source org ○ sfdx force:source:retrieve -m ApexClass,CustomObject,LWC
- Deploy metadata to target org ○ sfdx force:source:deploy -p forceapp/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 ≥ 75%.
- Run Validation Deployment (test without committing).

- Backup Production metadata before final push.
- Post-deployment steps (activate Flows, verify Email Deliverability).

Phase 9: Reporting, Dashboards & Security Review

Goal

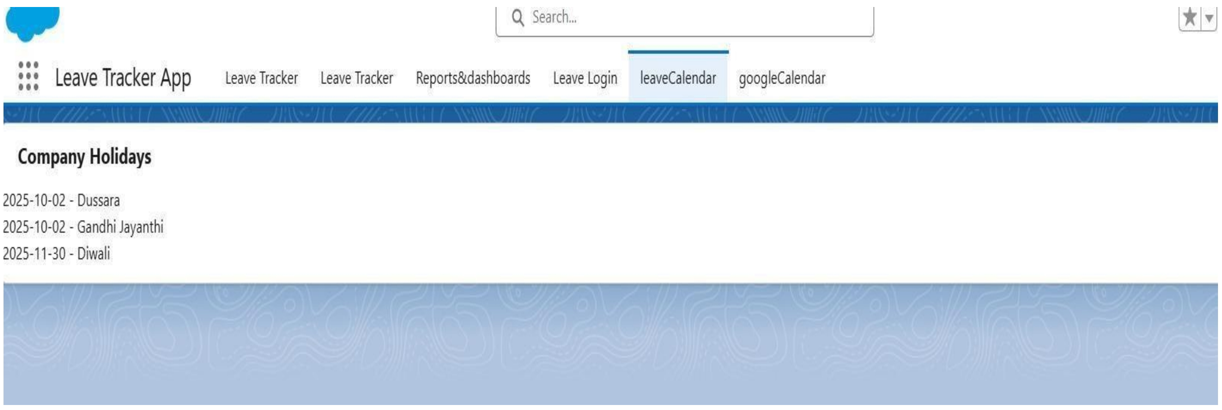
- Track leave requests and approvals with reports & dashboards.
- Protect sensitive employee leave data from unauthorized access.
- Ensure managers, HR, and employees see the right information in real

time. **Reports**

Created different report formats to track leave usage:

Report Types

- Created a Custom Report Type:
- Primary Object: LeaveRequest__c
- Related Object: User
- Calender: Leave Calender
- Calender:Google Calender



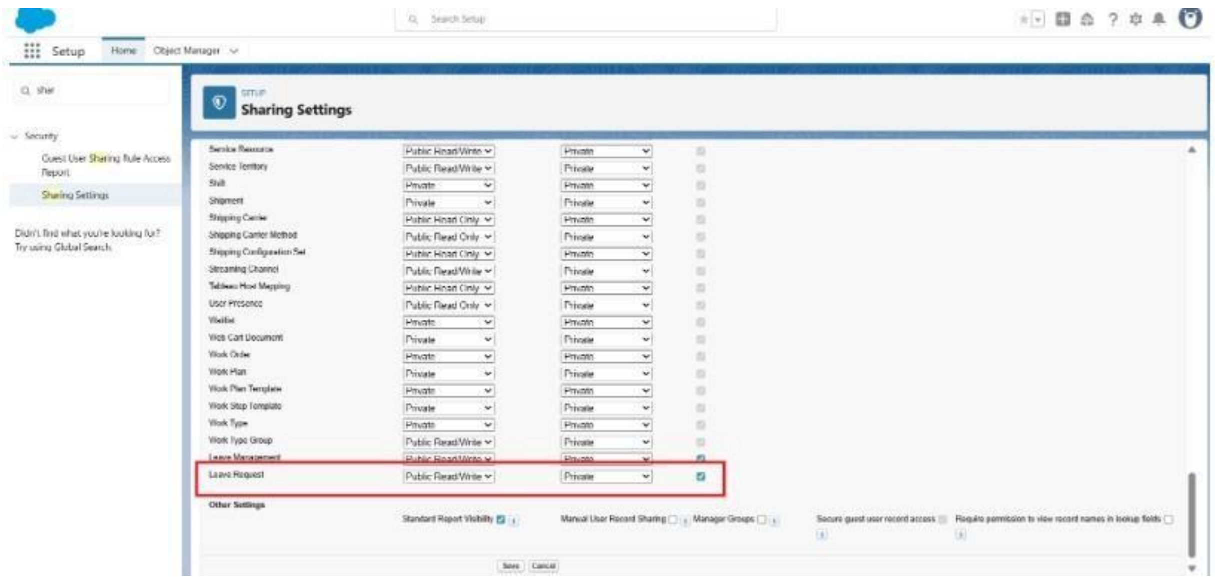
Company Holidays

2025-10-02 - Dussara
2025-10-02 - Gandhi Jayanthi
2025-11-30 - Diwali

Security

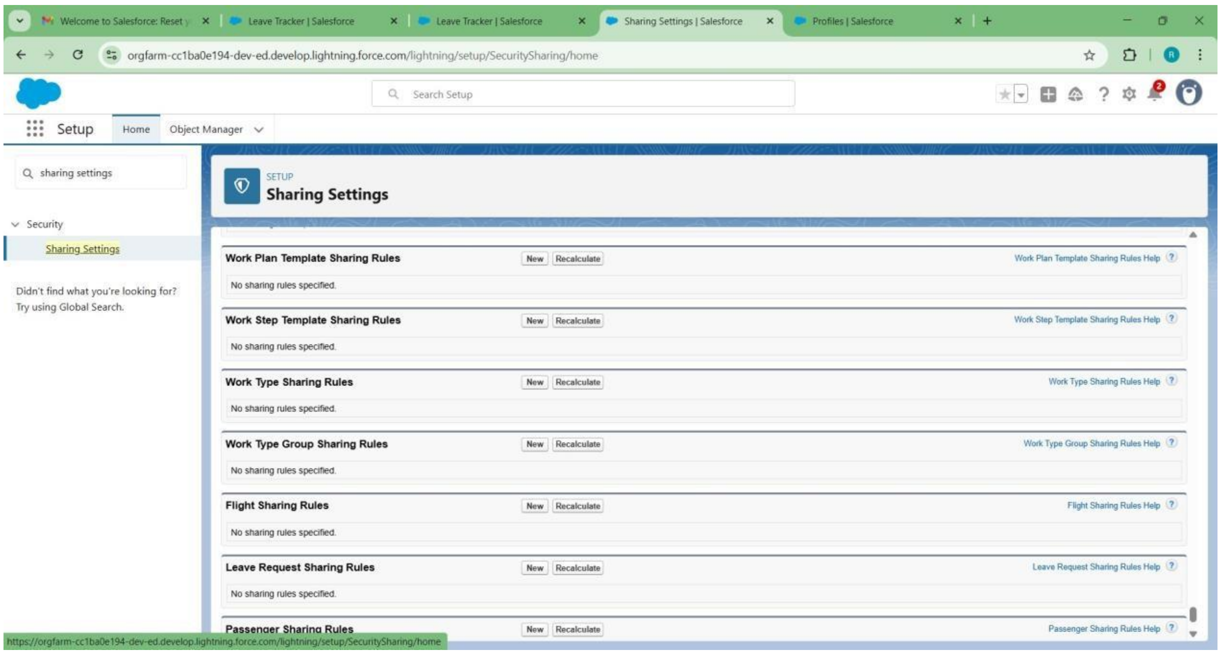
A. Sharing Settings

- Org-Wide Defaults (OWD): Leave Requests = Private (only owner, manager, HR can see).
- Sharing Rules: Managers can access their team’s requests.
- Role Hierarchy: Employee → Manager → HR → Admin.


SETUP
Sharing Settings

Service Resource	Public Read/Write	Private	
Service Territory	Public Read/Write	Private	
Shift	Private	Private	
Shipment	Private	Private	
Shipping Carrier	Public Read Only	Private	
Shipping Carrier Method	Public Read Only	Private	
Shipping Configuration Set	Public Read Only	Private	
Streaming Channel	Public Read/Write	Private	
Tableau Host Mapping	Public Read Only	Private	
User Presence	Public Read Only	Private	
Waitlist	Private	Private	
Web Cart Document	Private	Private	
Work Order	Private	Private	
Work Plan	Private	Private	
Work Plan Template	Private	Private	
Work Step Template	Private	Private	
Work Type	Private	Private	
Work Type Group	Public Read/Write	Private	
Leave Management	Public Read/Write	Private	✓
Leave Request	Public Read/Write	Private	✓

Other Settings
Standard Report Visibility ✓ Manual User Record Sharing ☐ Manager Groups ☐ Secure guest user record access ☐ Require permission to view record names in lookup fields ☐


orgfarm-cc1ba0e194-dev-ed.develop.lightning.force.com/lightning/setup/SecuritySharing/home

SETUP
Sharing Settings

Work Plan Template Sharing Rules
No sharing rules specified.

Work Step Template Sharing Rules
No sharing rules specified.

Work Type Sharing Rules
No sharing rules specified.

Work Type Group Sharing Rules
No sharing rules specified.

Flight Sharing Rules
No sharing rules specified.

Leave Request Sharing Rules
No sharing rules specified.

Passenger Sharing Rules

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

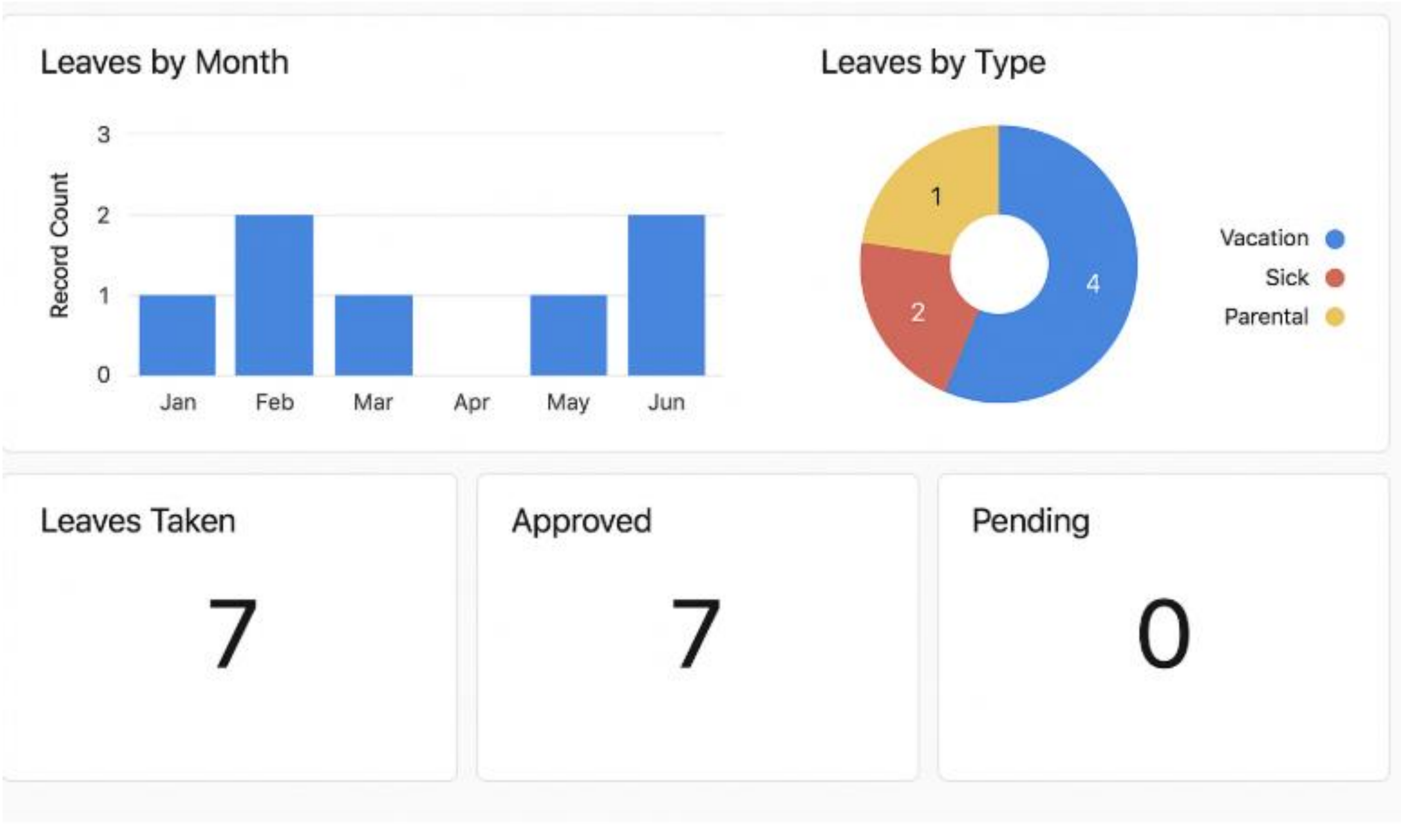
○ Pie Chart: Leaves by Type

○ Bar Chart: Leaves by Status

○ Line Chart: Leaves Over Time

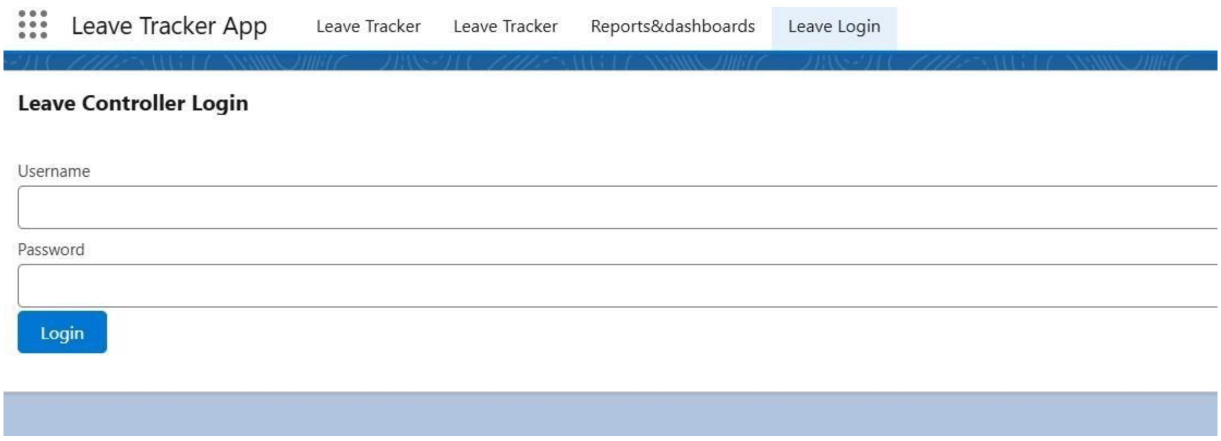
B. Dynamic Dashboards

Managers → See only their team's data. HR/Admins → See complete organizationlevel data.



Field Level Security (FLS)

- Employees: Can see only their leave details.
- Managers: Can see team leaves + approval fields.
- Sensitive fields (like Manager Comments) hidden from employees.



Leave Controller Login

Username

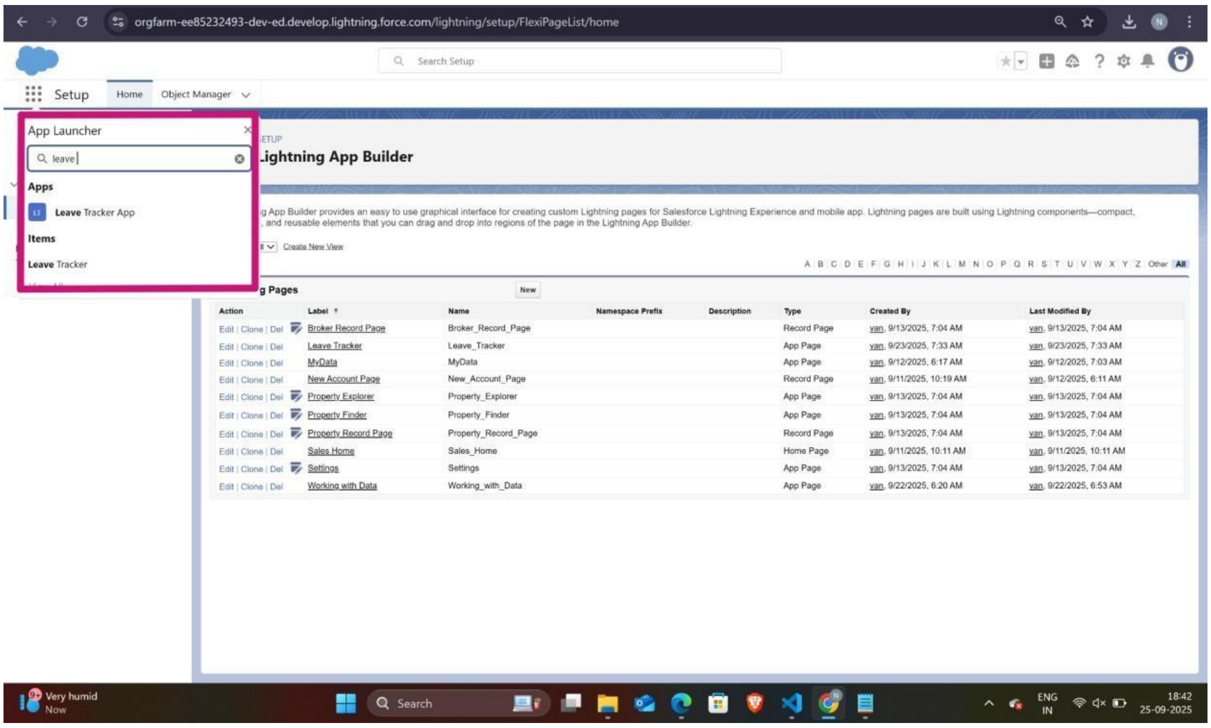
Password

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 Hub App** with: ■ Online leave request submission.
    - ■ Manager approval/rejection workflow.
    - ■ Automated email notifications.
    - ■ Real-time dashboards and reports.
    - ■ Login Page for User Security

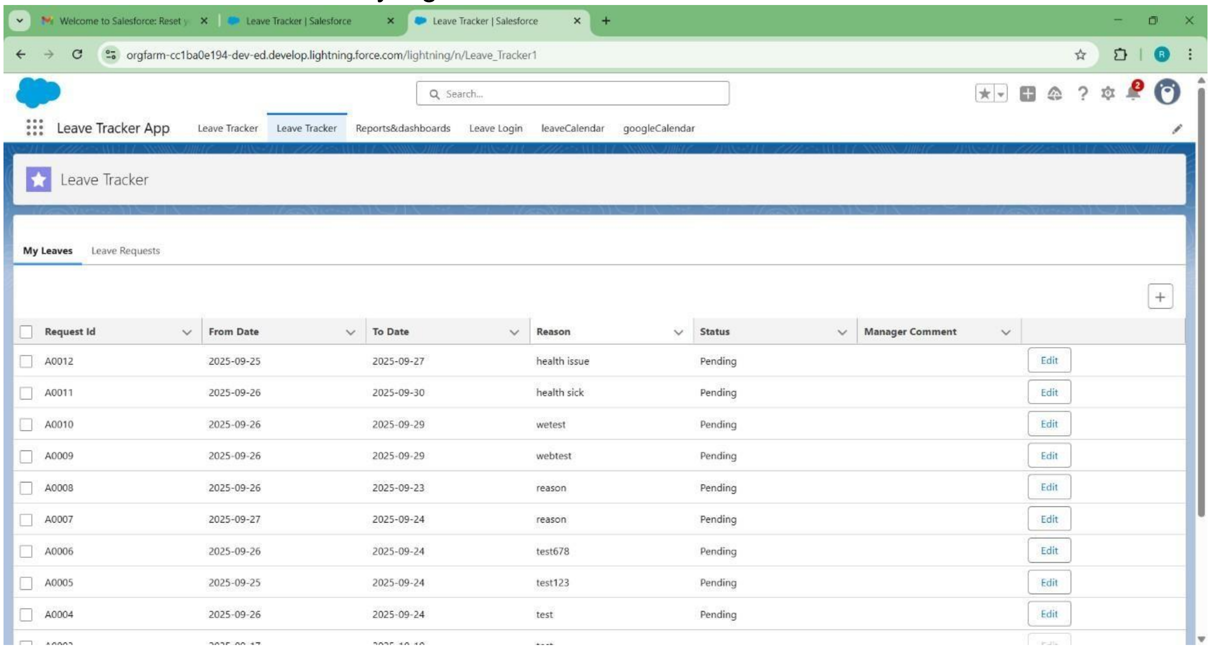


Lightning App Builder provides an easy to use graphical interface for creating custom Lightning pages for Salesforce Lightning Experience and mobile app. Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of the page in the Lightning App Builder.

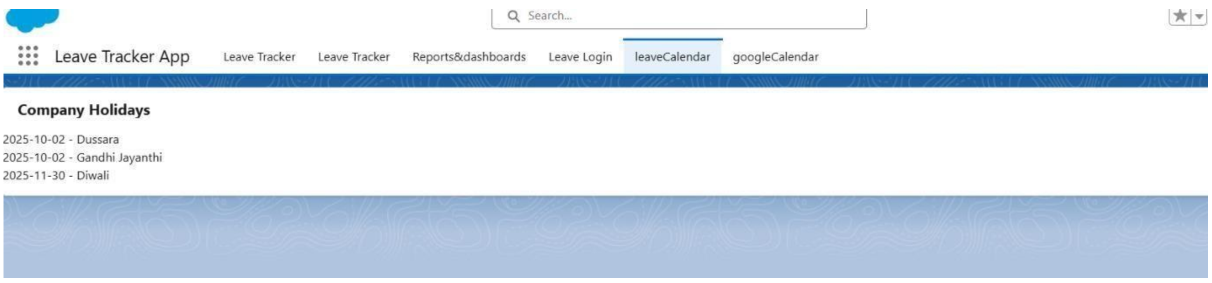
Lightning Pages

Action	Label	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Edit | Clone | Del	Broker Record Page	Broker_Record_Page			Record Page	van, 9/13/2025, 7:04 AM	van, 9/13/2025, 7:04 AM
Edit | Clone | Del	Leave Tracker	Leave_Tracker			App Page	van, 9/23/2025, 7:33 AM	van, 9/23/2025, 7:33 AM
Edit | Clone | Del	MyData	MyData			App Page	van, 9/12/2025, 6:17 AM	van, 9/12/2025, 7:03 AM
Edit | Clone | Del	New Account Page	New_Account_Page			Record Page	van, 9/11/2025, 10:19 AM	van, 9/12/2025, 6:11 AM
Edit | Clone | Del	Property Explorer	Property_Explorer			App Page	van, 9/13/2025, 7:04 AM	van, 9/13/2025, 7:04 AM
Edit | Clone | Del	Property Finder	Property_Finder			App Page	van, 9/13/2025, 7:04 AM	van, 9/13/2025, 7:04 AM
Edit | Clone | Del	Property Record Page	Property_Record_Page			Record Page	van, 9/13/2025, 7:04 AM	van, 9/13/2025, 7:04 AM
Edit | Clone | Del	Sales Home	Sales_Home			Home Page	van, 9/11/2025, 10:11 AM	van, 9/11/2025, 10:11 AM
Edit | Clone | Del	Settings	Settings			App Page	van, 9/13/2025, 7:04 AM	van, 9/13/2025, 7:04 AM
Edit | Clone | Del	Working with Data	Working_with_Data			App Page	van, 9/22/2025, 6:20 AM	van, 9/22/2025, 6:53 AM

- **Benefits:**
  - Streamlined leave approval process.
  - Improved accuracy and reduced manual errors.
  - Instant visibility into employee leave trends.
  - Secure & scalable within Salesforce.
  - User Security login


Welcome to Salesforce: Reset y
Leave Tracker | Salesforce
Leave Tracker | Salesforce
orgfarm-cc1ba0e194-dev-ed.develop.lightning.force.com/lightning/n/Leave_Tracker1
Search...
Leave Tracker App
Leave Tracker
Leave Tracker
Reports&dashboards
Leave Login
leaveCalendar
googleCalendar
Leave Tracker
My Leaves
Leave Requests
+

Request Id	From Date	To Date	Reason	Status	Manager Comment	
A0012	2025-09-25	2025-09-27	health issue	Pending		Edit
A0011	2025-09-26	2025-09-30	health sick	Pending		Edit
A0010	2025-09-26	2025-09-29	wetest	Pending		Edit
A0009	2025-09-26	2025-09-29	webtest	Pending		Edit
A0008	2025-09-26	2025-09-23	reason	Pending		Edit
A0007	2025-09-27	2025-09-24	reason	Pending		Edit
A0006	2025-09-26	2025-09-24	test678	Pending		Edit
A0005	2025-09-25	2025-09-24	test123	Pending		Edit
A0004	2025-09-26	2025-09-24	test	Pending		Edit

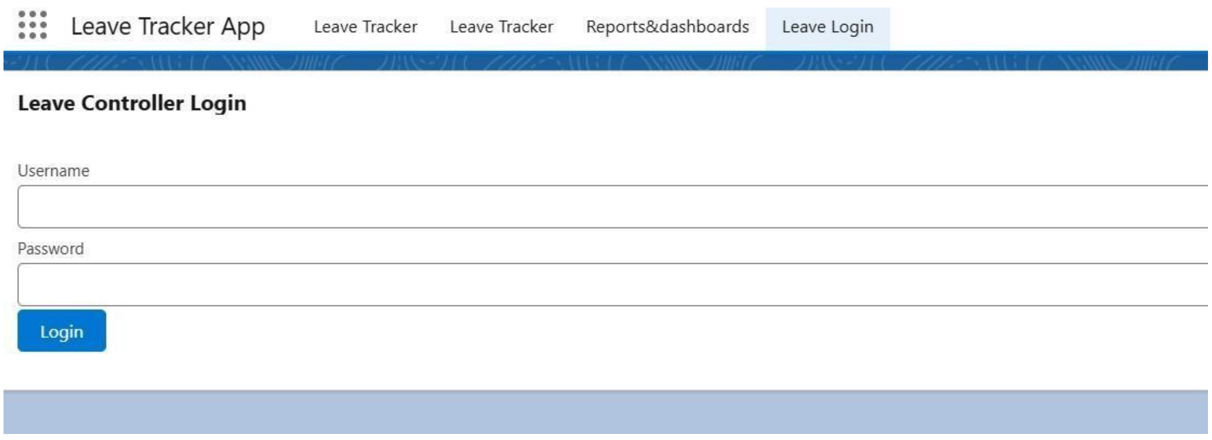


Company Holidays

2025-10-02 - Dussara
2025-10-02 - Gandhi Jayanthi
2025-11-30 - Diwali

2. Demo Walkthrough

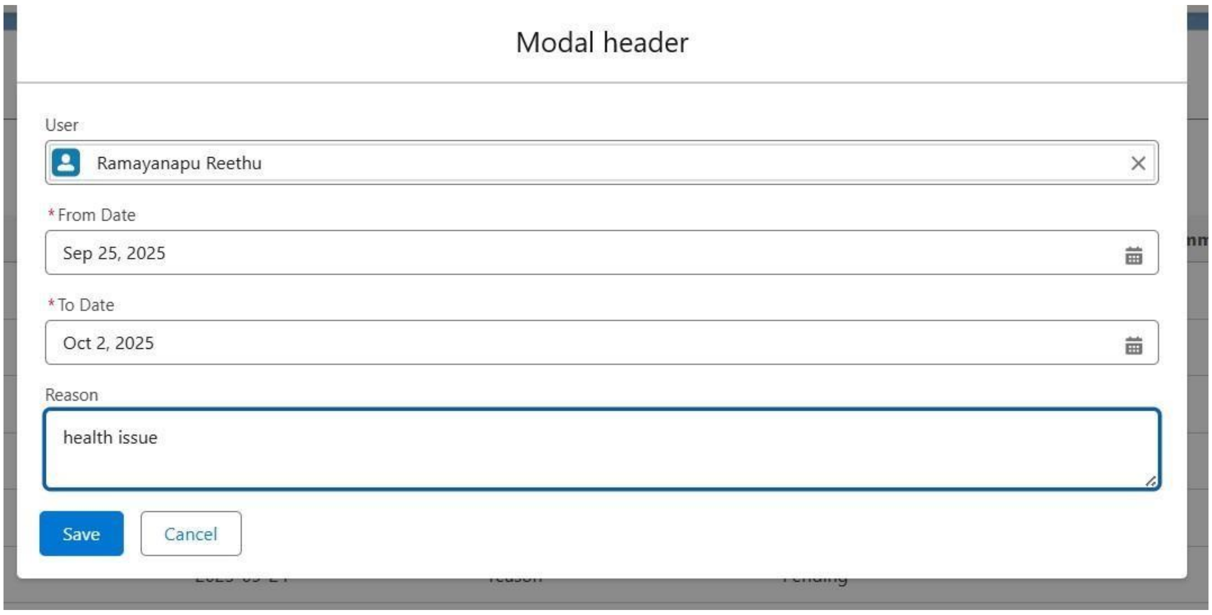
- **Employee Workflow:**
    - Log in and click **Apply Leave** (LWC form).
    - Fill details → Submit → Leave Request record created.
    - Validation ensures proper date entries.
    - User Login



Leave Controller Login

Username

Password



Modal header

User

Ramayanapu Reethu

*From Date

Sep 25, 2025

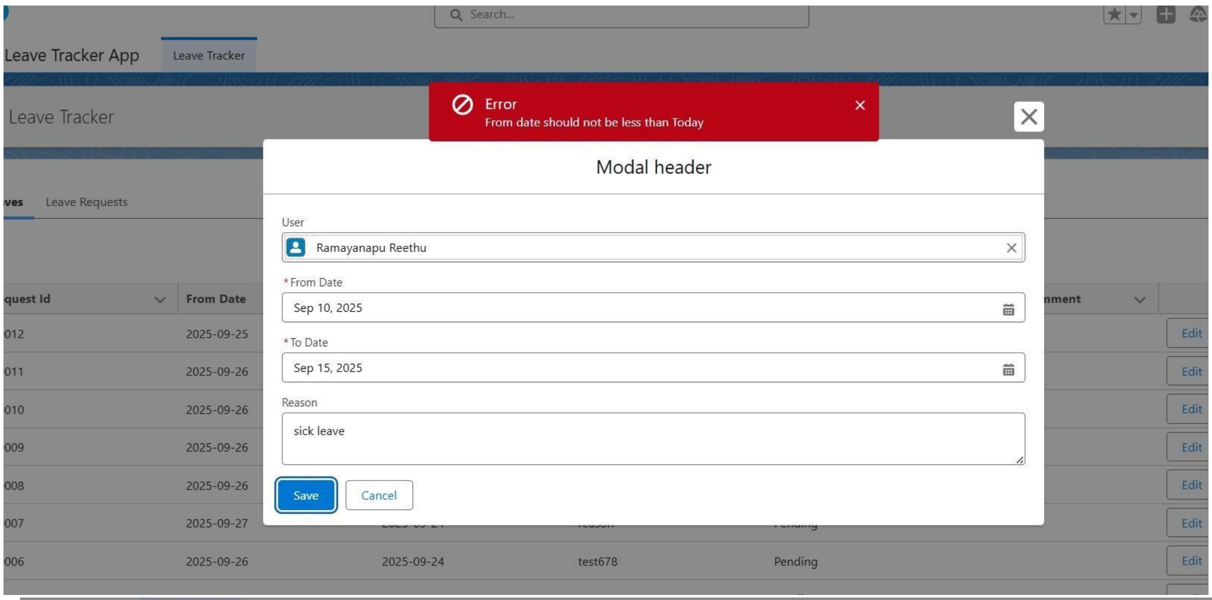
*To Date

Oct 2, 2025

Reason

health issue

Save Cancel



Leave Tracker App | Leave Tracker

Leave Tracker

Error
From date should not be less than Today

Modal header

User

Ramayanapu Reethu

*From Date

Sep 10, 2025

*To Date

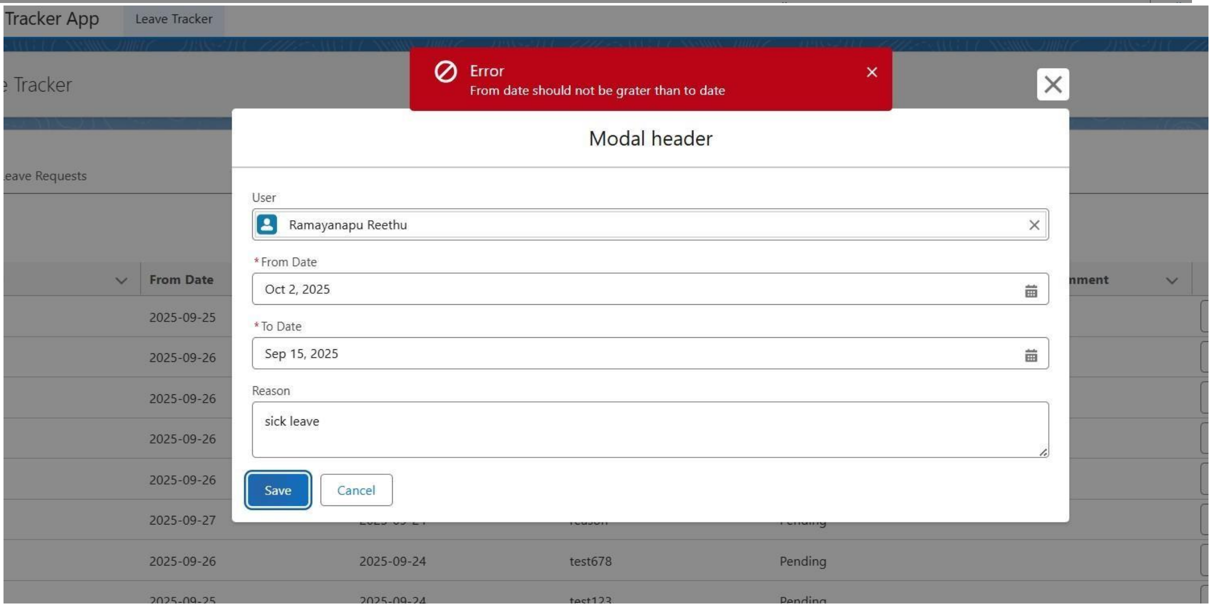
Sep 15, 2025

Reason

sick leave

Save Cancel

quest Id	From Date				
012	2025-09-25				Edit
011	2025-09-26				Edit
010	2025-09-26				Edit
009	2025-09-26				Edit
008	2025-09-26				Edit
007	2025-09-27				Edit
006	2025-09-26	2025-09-24	test678	Pending	Edit



Tracker App | Leave Tracker

Error
From date should not be grater than to date

Modal header

User

Ramayanapu Reethu

*From Date

Oct 2, 2025

*To Date

Sep 15, 2025

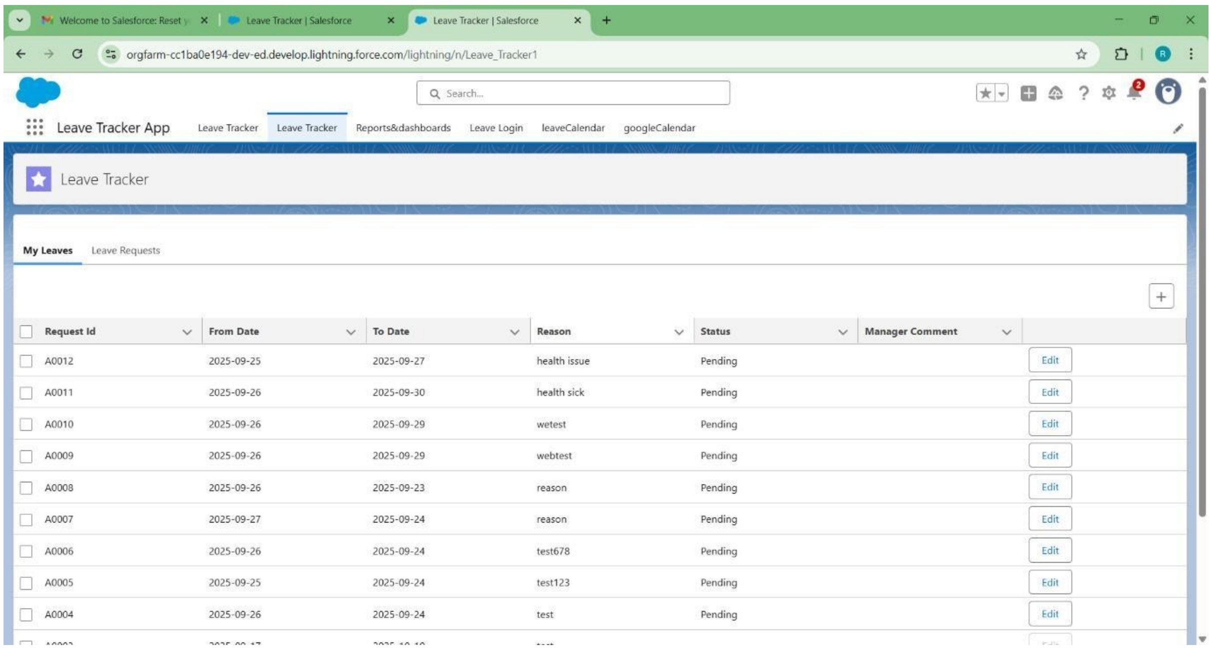
Reason

sick leave

Save Cancel

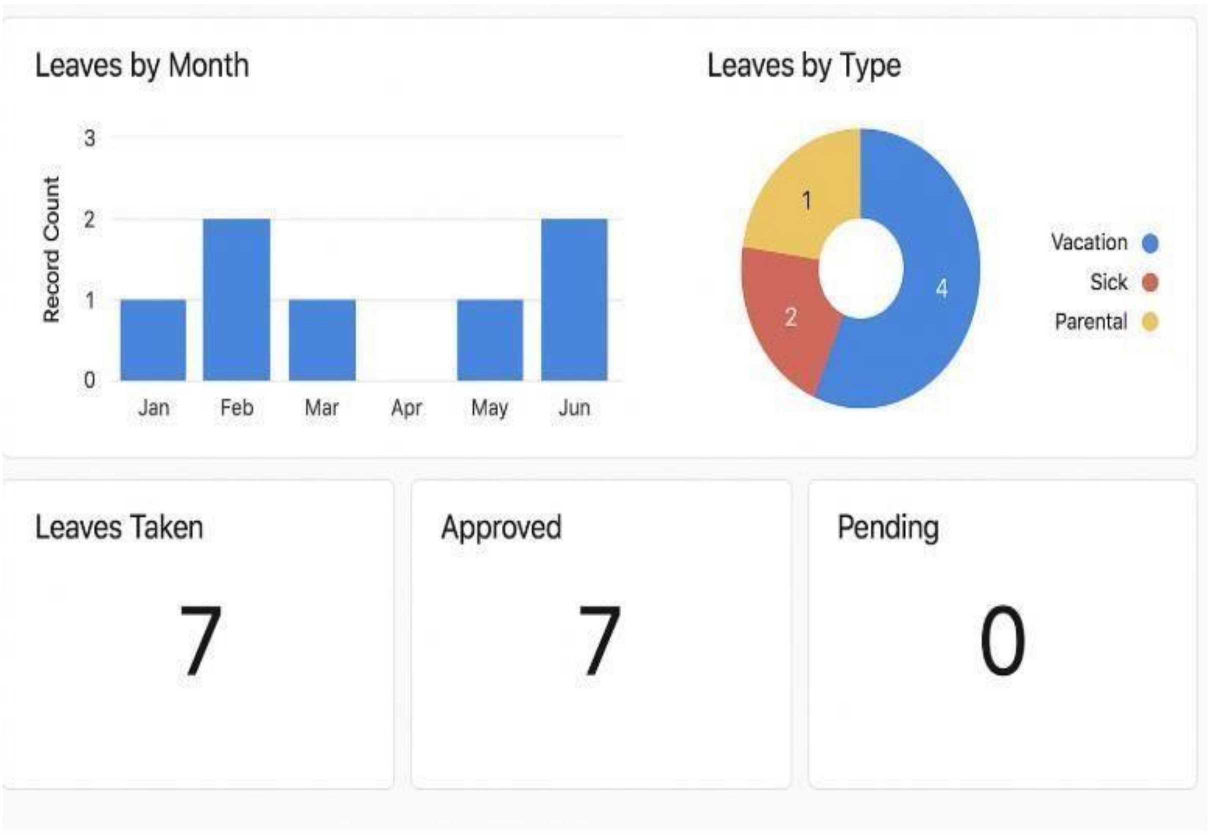
From Date			
2025-09-25			
2025-09-26			
2025-09-26			
2025-09-26			
2025-09-26			
2025-09-27			
2025-09-26	2025-09-24	test678	Pending
2025-09-25	2025-09-24	test123	Pending

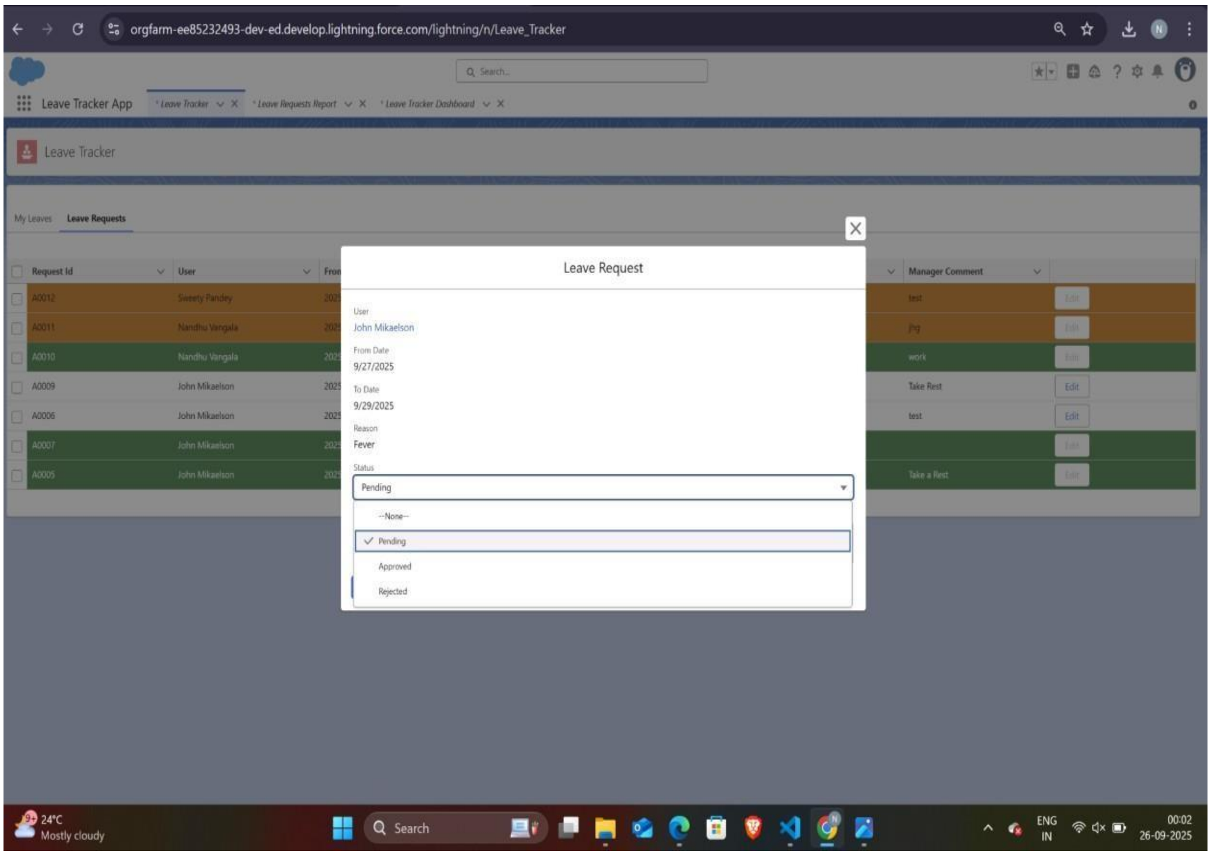
- **Manager Workflow:**
  - Open **Leave Requests** tab.
  - Review pending requests.
  - Approve or Reject directly from the record page or LWC.
  - Automatic email sent to employee.
- **Reports & Dashboards:**
  - "My Leave Requests" report for employees.
  - Manager dashboard showing Leaves.
  - Status charts: Approved , Pending , Rejected.



orgfarm-cc1ba0e194-dev-ed.develop.lightning.force.com/lightning/n/Leave_Tracker1

Request Id	From Date	To Date	Reason	Status	Manager Comment	
A0012	2025-09-25	2025-09-27	health issue	Pending		Edit
A0011	2025-09-26	2025-09-30	health sick	Pending		Edit
A0010	2025-09-26	2025-09-29	wetest	Pending		Edit
A0009	2025-09-26	2025-09-29	webtest	Pending		Edit
A0008	2025-09-26	2025-09-23	reason	Pending		Edit
A0007	2025-09-27	2025-09-24	reason	Pending		Edit
A0006	2025-09-26	2025-09-24	test678	Pending		Edit
A0005	2025-09-25	2025-09-24	test123	Pending		Edit
A0004	2025-09-26	2025-09-24	test	Pending		Edit




orgfarm-ee85232493-dev-ed.develop.lightning.force.com/lightning/n/Leave_Tracker
Search...
Leave Tracker App
* Leave Tracker
* Leave Requests Report
* Leave Tracker Dashboard
Leave Tracker
My Leaves
Leave Requests

Request Id	User	Manager Comment	
A0012	Sweety Pandey	test	Edit
A0011	Nandhu Vangala	jjg	Edit
A0010	Nandhu Vangala	work	Edit
A0009	John Mikaelson	Take Rest	Edit
A0006	John Mikaelson	test	Edit
A0007	John Mikaelson		Edit
A0005	John Mikaelson	Take a Rest	Edit

Leave Request

User
John Mikaelson

From Date
9/27/2025

To Date
9/29/2025

Reason
Fever

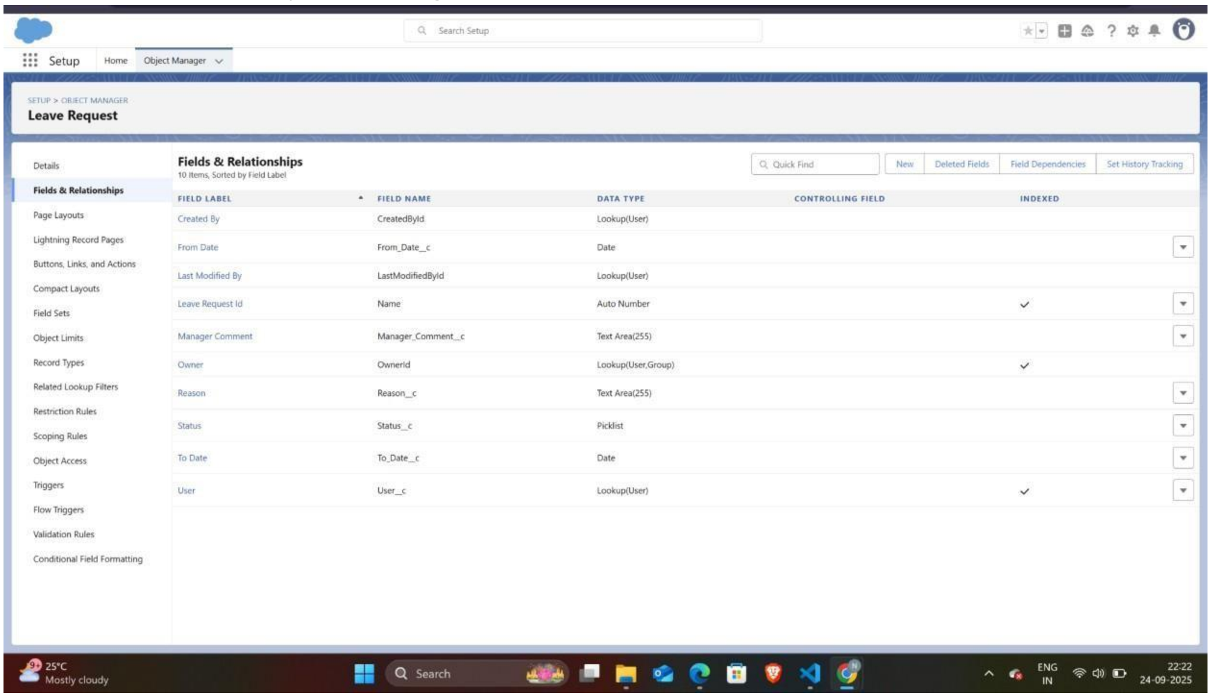
Status
Pending

--None--
✓ Pending
Approved
Rejected

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.

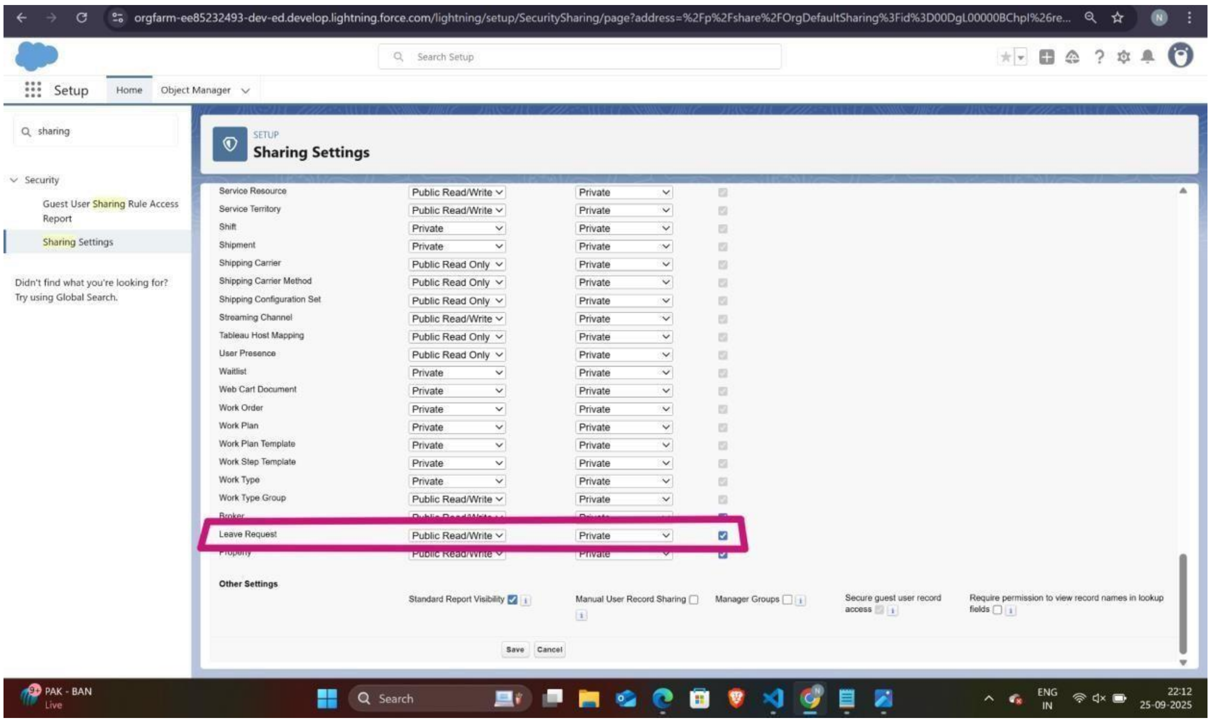

SETUP > OBJECT MANAGER

Leave Request

Fields & Relationships

10 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
From Date	From_Date__c	Date		
Last Modified By	LastModifiedById	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)		✓


orgfarm-ee85232493-dev-ed.develop.lightning.force.com/lightning/setup/SecuritySharing/page?address=%2Fp%2Fshare%2FOrgDefaultSharing%3Fid%3D00DgL00000BChpl%26re...

SETUP

Sharing Settings

Service Resource	Public Read/Write	Private	✓
Service Territory	Public Read/Write	Private	✓
Shift	Private	Private	✓
Shipment	Private	Private	✓
Shipping Carrier	Public Read Only	Private	✓
Shipping Carrier Method	Public Read Only	Private	✓
Shipping Configuration Set	Public Read Only	Private	✓
Streaming Channel	Public Read/Write	Private	✓
Tableau Host Mapping	Public Read Only	Private	✓
User Presence	Public Read Only	Private	✓
Waitlist	Private	Private	✓
Web Cart Document	Private	Private	✓
Work Order	Private	Private	✓
Work Plan	Private	Private	✓
Work Plan Template	Private	Private	✓
Work Step Template	Private	Private	✓
Work Type	Private	Private	✓
Work Type Group	Public Read/Write	Private	✓
Broker	Public Read/Write	Private	✓
Leave Request	Public Read/Write	Private	✓
Property	Public Read/Write	Private	✓

Other Settings

Standard Report Visibility ✓ Manual User Record Sharing ☐ Manager Groups ☐ Secure guest user record access ✓ Require permission to view record names in lookup fields ☐

- **User Guide:**
  - How employees apply leave.
  - Employee Logins
  - 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.

Phase 10 Outcome:

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