Leave Tracker App

Industry: Corporate / General Business

Project Type: Salesforce LWC implementation for a Leave Management App to streamline employee leave requests and automate approvals.

Target Users:

- **Employees:** Need to submit new leave requests and view the status of their past requests.
- **Managers:** Need to approve or reject their direct reports' leave requests.
- **HR/Admins:** Need to see a comprehensive overview of all employee leave and generate reports.

Problem Statement: Currently, the company handles employee leave requests through emails and spreadsheets. This causes mistakes like double bookings, lost requests, and no real-time tracking. A centralized automated system is needed to make leave management easier and more accurate.

Phase 1: Problem Understanding & Requirements

- **Goal:** Define the purpose and scope of the Leave Management App.

- **Requirement Gathering:**
  - Track all employee leave requests.
  - Show leave balances for each employee.
  - Allow managers to approve or reject leave requests.
  - Prevent employees from submitting overlapping leave requests.
  - Generate a report of all approved leaves.

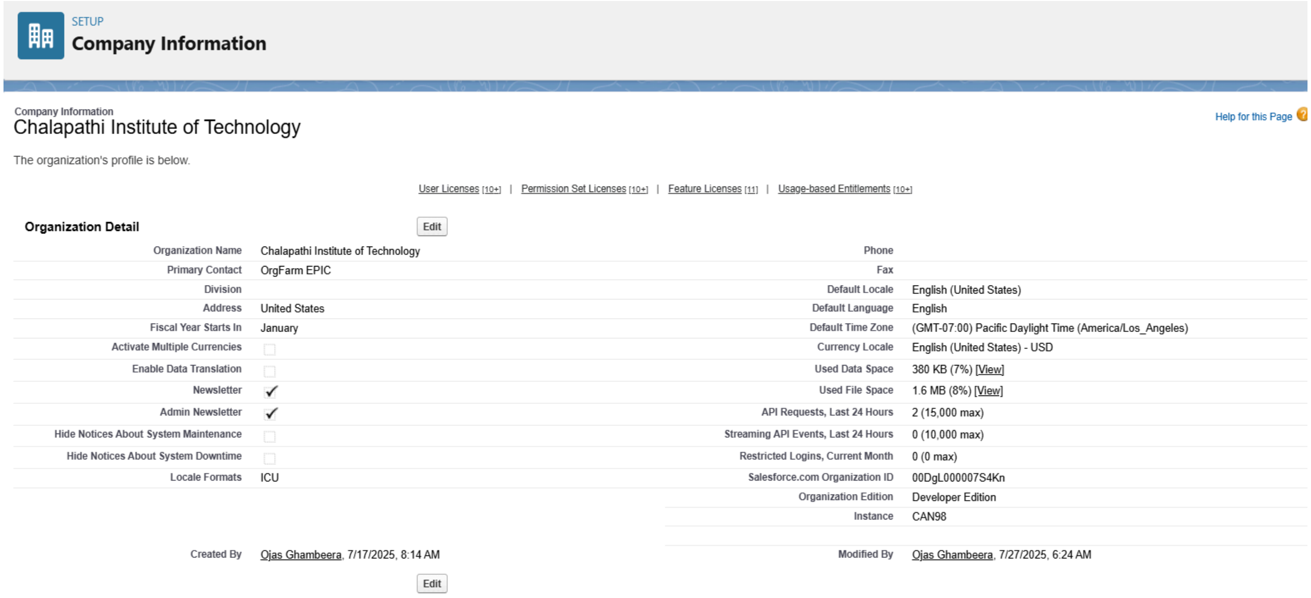
- **Stakeholder Analysis**:
  - Admin: Manages the system setup.
  - Employees: Submit and view their own leave requests.

    - Managers: Approve or reject leave requests and view team calendars.
    - HR: Oversees leave policies and reports.
- **Business Process Mapping**:
    - Draw a flow: Employee submits request → Manager receives notification → Manager approves/rejects → Employee is notified of the decision.
- **Industry Analysis:**
    - Consider industry-specific use cases, such as different types of leave (sick, vacation, etc.) and company-specific policies.

Phase 2: Org Setup & Configuration:

Goal: Prepare your Salesforce Developer Edition environment.

Company Profile Setup: Set up your company's information, local time zone, and currency in Salesforce.



Company Information

Company Information

Chalapathi Institute of Technology

Help for this Page

The organization's profile is below.

User Licenses [10+] | Permission Set Licenses [10+] | Feature Licenses [11] | Usage-based Entitlements [10+]

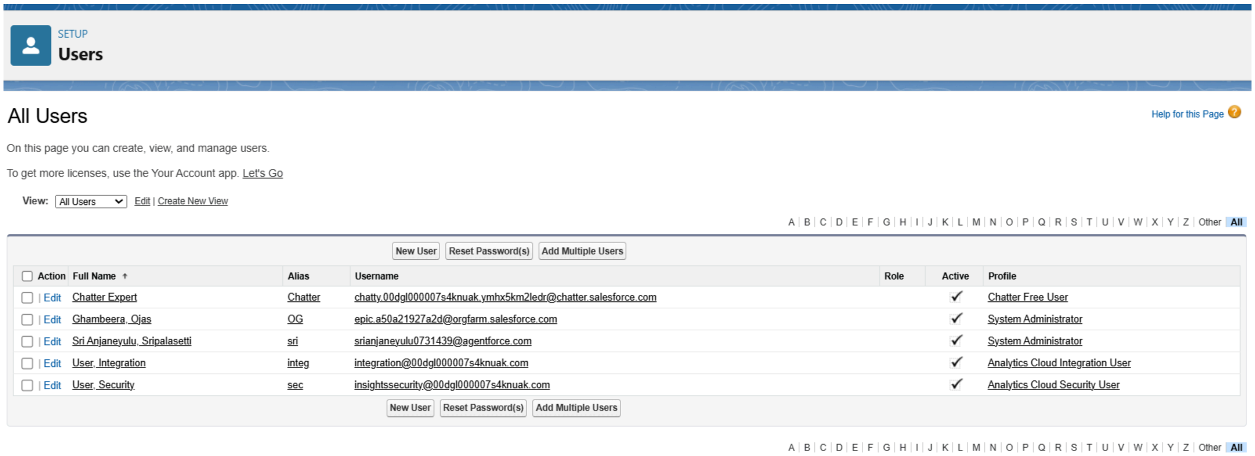
Organization Detail Edit

Organization Name	Chalapathi Institute of Technology	Phone	
Primary Contact	OrgFarm EPIC	Fax	
Division		Default Locale	English (United States)
Address	United States	Default Language	English
Fiscal Year Starts In	January	Default Time Zone	(GMT-07:00) Pacific Daylight Time (America/Los_Angeles)
Activate Multiple Currencies	□	Currency Locale	English (United States) - USD
Enable Data Translation	□	Used Data Space	380 KB (7%) [View]
Newsletter	✓	Used File Space	1.6 MB (8%) [View]
Admin Newsletter	✓	API Requests, Last 24 Hours	2 (15,000 max)
Hide Notices About System Maintenance	□	Streaming API Events, Last 24 Hours	0 (10,000 max)
Hide Notices About System Downtime	□	Restricted Logins, Current Month	0 (0 max)
Locale Formats	ICU	Salesforce.com Organization ID	00DgL000007S4Kn
		Organization Edition	Developer Edition
		Instance	CAN98
Created By	Ojas Ghambeera, 7/17/2025, 8:14 AM	Modified By	Ojas Ghambeera, 7/27/2025, 6:24 AM

Edit

User Setup & Profiles:

Create users for employees and managers. Create custom profiles or permission sets for different access levels. For example, employees can create their own requests, while managers have approval access.



SETUP

Users

All Users

Help for this Page

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

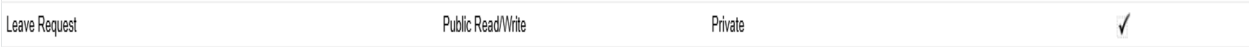
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
New User | Reset Password(s) | Add Multiple Users

□ Action	Full Name ↑	Alias	Username	Role	Active	Profile
□ | Edit	Chatter Expert	Chatter	chatty.00dgl000007s4knuak.ymhx5km2ledr@chatter.salesforce.com		✓	Chatter Free User
□ | Edit	Ghambeera, Ojas	OG	epic.a50a21927a2d@orgfarm.salesforce.com		✓	System Administrator
□ | Edit	Sri Anjaneyulu, Sripalasetti	sri	srianjaneyulu0731439@agentforce.com		✓	System Administrator
□ | Edit	User, Integration	integ	integration@00dgl000007s4knuak.com		✓	Analytics Cloud Integration User
□ | Edit	User, Security	sec	insightssecurity@00dgl000007s4knuak.com		✓	Analytics Cloud Security User

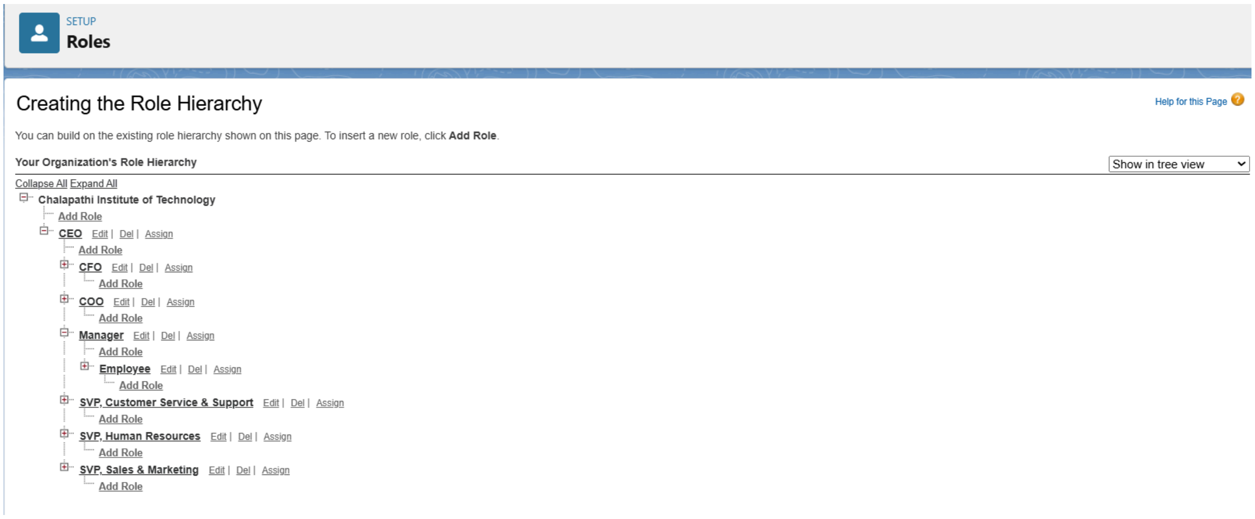
New User | Reset Password(s) | Add Multiple Users
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

Roles & Sharing:

The organization's reporting structure was established using a role hierarchy. A "Manager" role was created above the "Employee" role to ensure that managers can automatically see the leave requests of their direct reports. Additionally, the Organization-Wide Defaults (OWD) for the `Leave_Request__c` object were set to "Private." This is a critical security measure that prevents employees from seeing other employees' leave requests by default, ensuring data privacy. Sharing rules will be used to open up access as needed.



Leave Request	Public Read/Write	Private	✓



SETUP

Roles

Creating the Role Hierarchy

Help for this Page

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy

Show in tree view

Collapse All Expand All

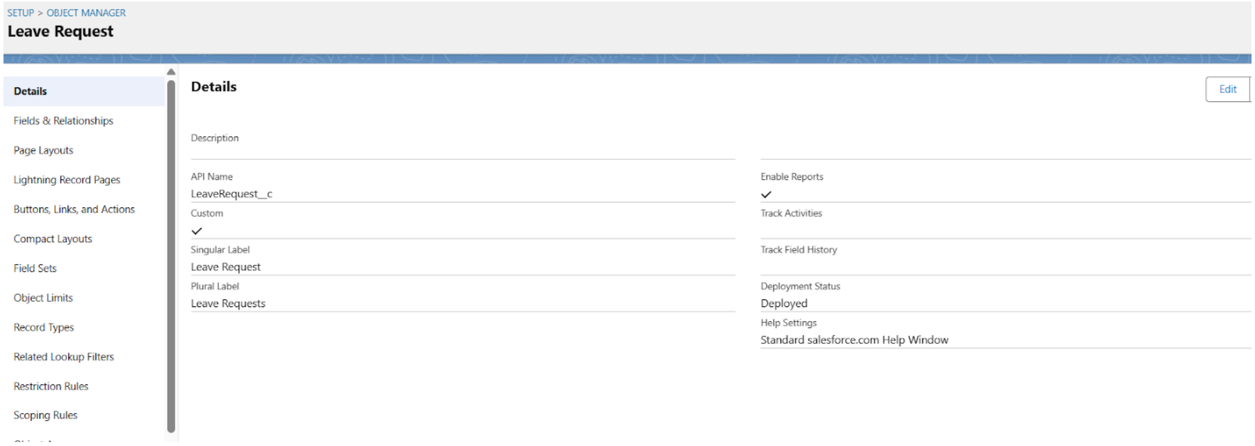
- Chalapathi Institute of Technology
  - Add Role
  - CEO Edit | Del | Assign
    - Add Role
    - CFO Edit | Del | Assign
      - Add Role
    - COO Edit | Del | Assign
      - Add Role
    - Manager Edit | Del | Assign
      - Add Role
      - Employee Edit | Del | Assign
        - Add Role
    - SVP, Customer Service & Support Edit | Del | Assign
      - Add Role
    - SVP, Human Resources Edit | Del | Assign
      - Add Role
    - SVP, Sales & Marketing Edit | Del | Assign
      - Add Role

Dev Org Setup:

A Salesforce Developer Org was used as the primary development environment. This approach provides a safe, isolated sandbox for building and testing all custom components, Apex code, and automations without affecting a live production environment.

Phase 3: Data Modeling & Relationships

- **Goal**: Design the data structure for the app.
- **Objects**:
  - Standard: Use the `User` object to represent employees and managers.
  - Custom: Create a `Leave_Request__c` object to store leave information.



Leave Request

Details

Description

API Name
LeaveRequest__c

Custom
✓

Singular Label
Leave Request

Plural Label
Leave Requests

Enable Reports
✓

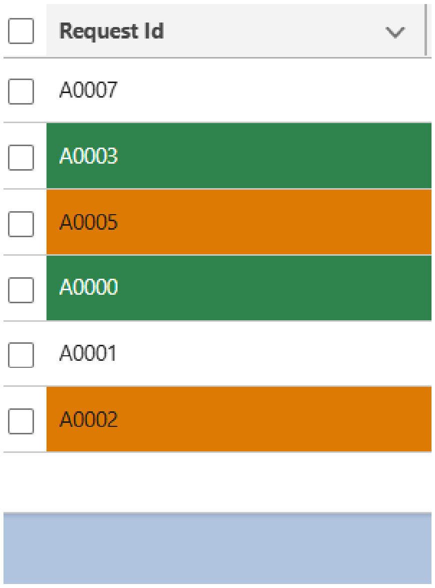
Track Activities

Track Field History

Deployment Status
Deployed

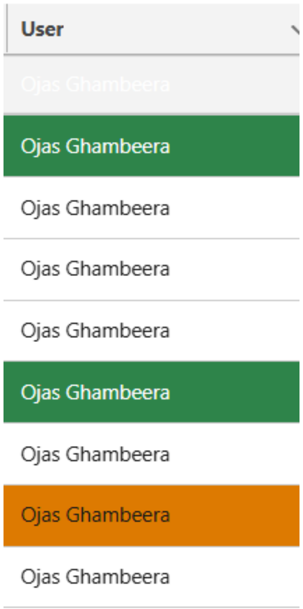
Help Settings
Standard salesforce.com Help Window

`Request Id`: A system-generated identifier for each record.



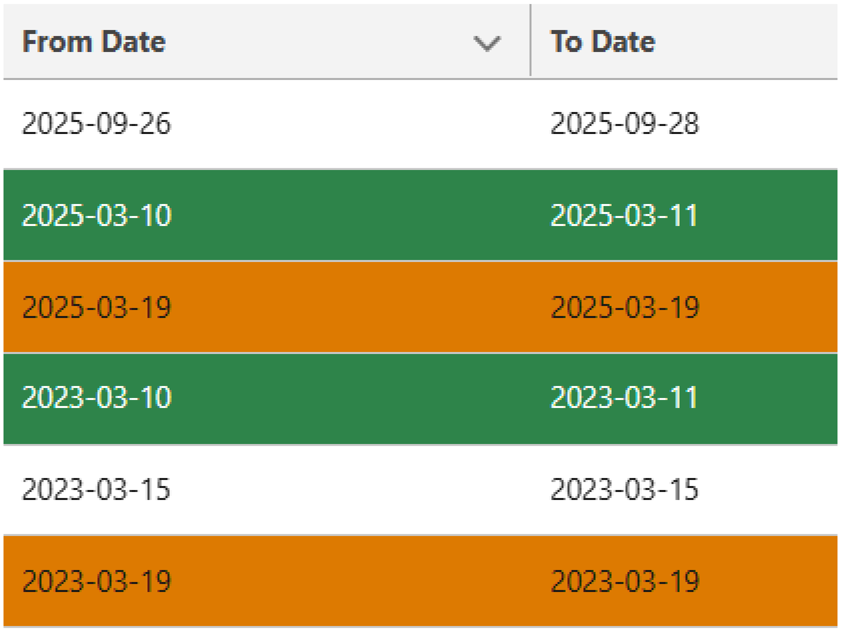
□	Request Id
□	A0007
□	A0003
□	A0005
□	A0000
□	A0001
□	A0002

`User`: A lookup relationship to the `User` object, correctly linking the leave request to the employee who submitted it.



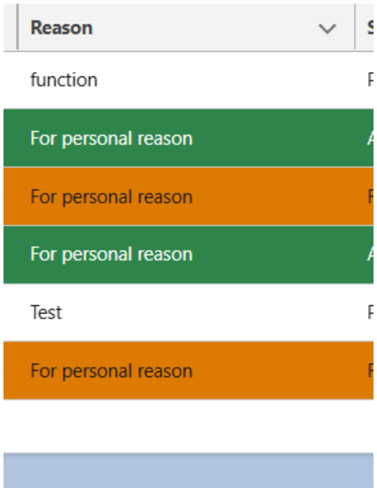
User
Ojas Ghambeera
Ojas Ghambeera
Ojas Ghambeera
Ojas Ghambeera
Ojas Ghambeera
Ojas Ghambeera
Ojas Ghambeera
Ojas Ghambeera
Ojas Ghambeera

`From Date` and `To Date`: Date fields that define the duration of the leave.



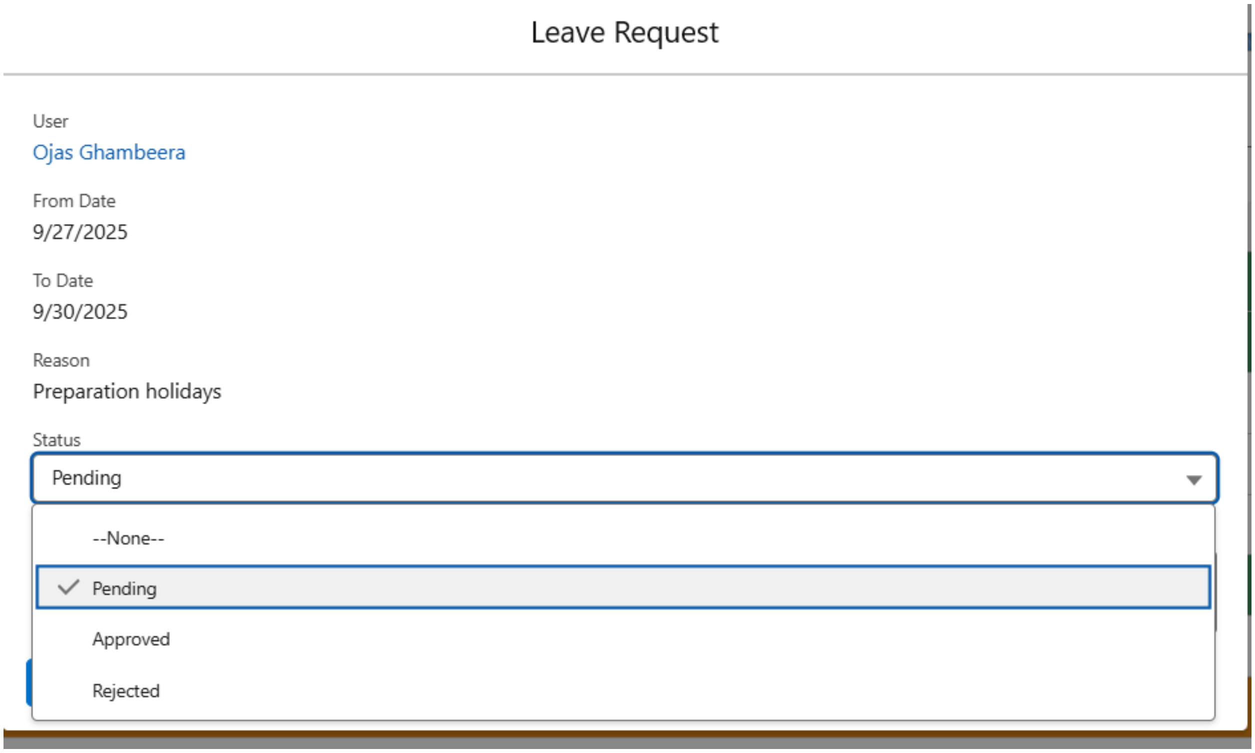
From Date	To Date
2025-09-26	2025-09-28
2025-03-10	2025-03-11
2025-03-19	2025-03-19
2023-03-10	2023-03-11
2023-03-15	2023-03-15
2023-03-19	2023-03-19

Reason: A text field to capture the employee's reason for the leave.



Reason	S
function	F
For personal reason	A
For personal reason	F
For personal reason	A
Test	F
For personal reason	F

Status: A picklist field that tracks the current state of the request (e.g., Pending, Approved, Rejected).



Leave Request

User
Ojas Ghambeera

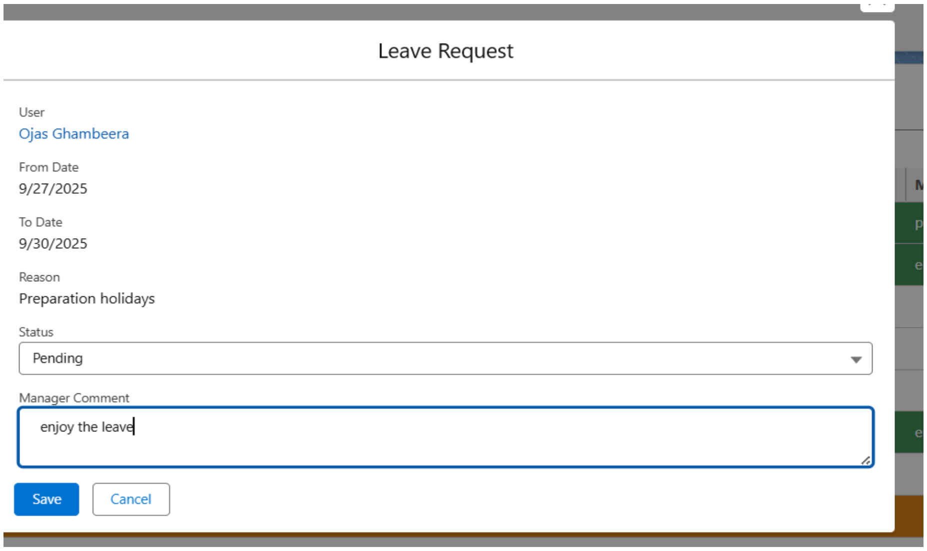
From Date
9/27/2025

To Date
9/30/2025

Reason
Preparation holidays

Status
Pending

Manager Comment: A text field for the manager to add notes during the approval process.



Leave Request

User
Ojas Ghambeera

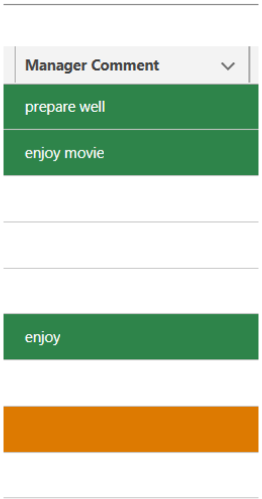
From Date
9/27/2025

To Date
9/30/2025

Reason
Preparation holidays

Status
Pending

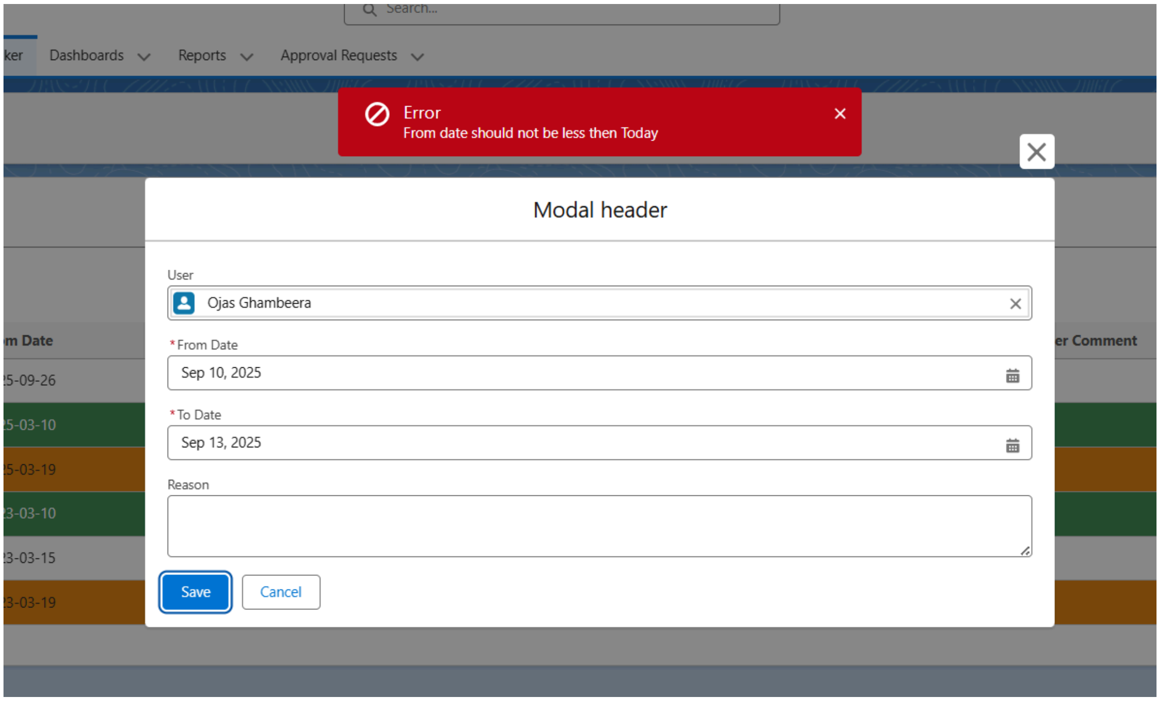
Manager Comment
enjoy the leave



Manager Comment
prepare well
enjoy movie
enjoy

Phase 4: Process Automation (Admin)

- **Goal**: Automate key business processes.
- **Validation Rules**:
  - Create a rule to ensure the End Date is after the Start Date.
  - Create a rule to prevent an employee from submitting a request with a past start date.



Error
From date should not be less then Today

Modal header

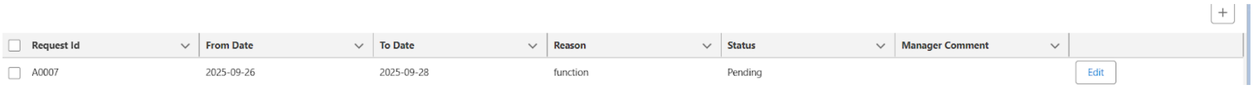
User
Ojas Ghambeera

*From Date
Sep 10, 2025

*To Date
Sep 13, 2025

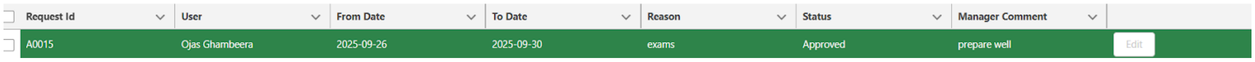
Reason

- **Flow Builder**:
  - Use a Record-Triggered Flow to automatically set the `Status` to 'Pending' when a new leave request is created.
  - Use a Screen Flow to create a custom form for employees to submit a new request.



□ Request Id	From Date	To Date	Reason	Status	Manager Comment	
□ A0007	2025-09-26	2025-09-28	function	Pending		Edit

- **Approval Process**:
    - Set up an approval process on the `Leave_Request__c` object to route requests to the assigned manager.

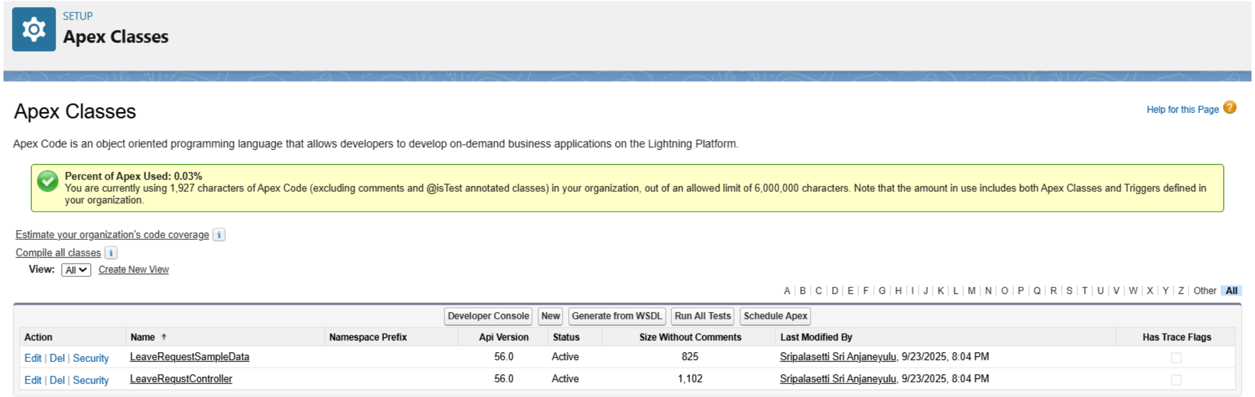


☐	Request Id	User	From Date	To Date	Reason	Status	Manager Comment	
☐	A0015	Ojas Ghambeera	2025-09-26	2025-09-30	exams	Approved	prepare well	Edit

---

Phase 5: Apex Programming (Developer)

- **Goal**: Add custom, complex logic.
- **Apex Triggers**:
    - Create an Apex Trigger on the `Leave_Request__c` object to prevent overlapping bookings for the same employee.



SETUP
Apex Classes

Apex Classes

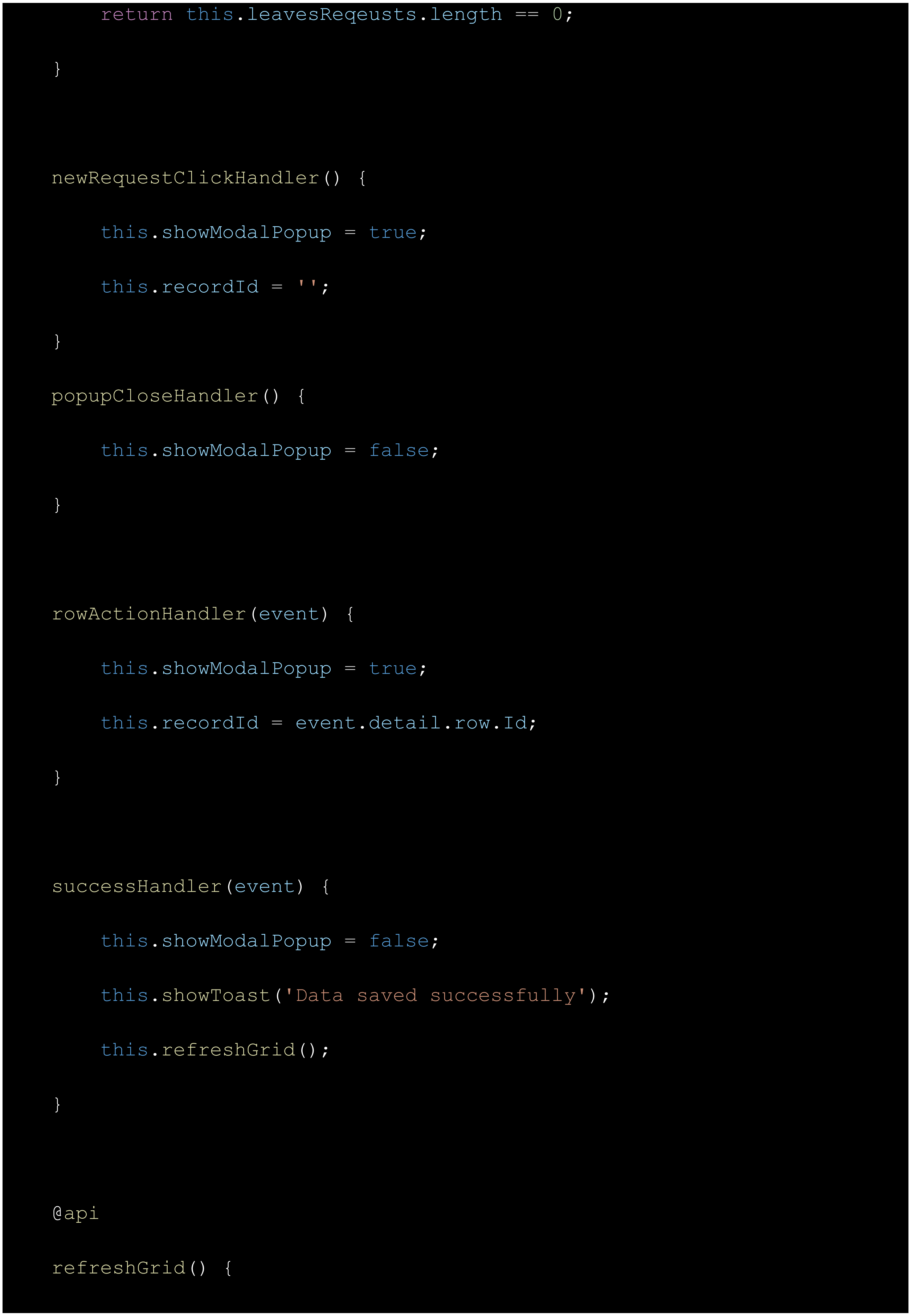
Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.03%
You are currently using 1,927 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage
Compile all classes
View: All
Create New View
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
Developer Console
New
Generate from WSDL
Run All Tests
Schedule Apex

Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit | Del | Security	LeaveRequestSampleData		56.0	Active	825	Sripalasetti Sri Anjaneyulu, 9/23/2025, 8:04 PM	☐
Edit | Del | Security	LeaveRequstController		56.0	Active	1,102	Sripalasetti Sri Anjaneyulu, 9/23/2025, 8:04 PM	☐

```
import { api, LightningElement, wire } from 'lwc';

import getLeaveRequests from
'@salesforce/apex/LeaveRequstController.getLeaveRequests';

import { ShowToastEvent } from 'lightning/platformShowToastEvent';

import Id from '@salesforce/user/Id';

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



const COLUMNS = [
    { label: 'Request Id', fieldName: 'Name', cellAttributes: { class: {
fieldName: 'cellClass' } } },

    { label: 'User', fieldName: 'userName', cellAttributes: { class: {
fieldName: 'cellClass' } } },

    { label: 'From Date', fieldName: 'From_Date__c', cellAttributes: {
class: { fieldName: 'cellClass' } } },

    { label: 'To Date', fieldName: 'To_Date__c', cellAttributes: { class:
{ fieldName: 'cellClass' } } },

    { label: 'Reason', fieldName: 'Reason__c', cellAttributes: { class: {
fieldName: 'cellClass' } } },

    { label: 'Status', fieldName: 'Status__c', cellAttributes: { class: {
fieldName: 'cellClass' } } },

    { label: 'Manager Comment', fieldName: 'Manager_Comment__c',
cellAttributes: { class: { fieldName: 'cellClass' } } },

    {

        type: "button", typeAttributes: {

            label: 'Edit',

            name: 'Edit',

            title: 'Edit',

            value: 'edit',

            disabled: { fieldName: 'isEditDisabled' }

        }, cellAttributes: { class: { fieldName: 'cellClass' } }

    }

];

export default class LeaveRequests extends LightningElement {

    columns = COLUMNS;
    leavesReqeusts = [];
    leavesReqeustsWireResult;
    showModalPopup = false;
    objectApiName = 'LeaveRequest__c';
    recordId = '';
    currentUserId = Id;
    @wire(getLeaveRequests)
    wiredMyLeaves(result) {
        this.leavesReqeustsWireResult = result;
        if (result.data) {
            this.leavesReqeusts = result.data.map(a => ({
                ...a,
                userName: a.User__r.Name,
                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 occured while fetching my leaves- ', result.error);
        }
    }

    get noRecordsFound() {
```



```
        return this.leavesReqeusts.length == 0;
    }

    newRequestClickHandler() {
        this.showModalPopup = true;
        this.recordId = '';
    }
    popupCloseHandler() {
        this.showModalPopup = false;
    }

    rowActionHandler(event) {
        this.showModalPopup = true;
        this.recordId = event.detail.row.Id;
    }

    successHandler(event) {
        this.showModalPopup = false;
        this.showToast('Data saved successfully');
        this.refreshGrid();
    }

    @api
    refreshGrid() {
        refreshApex(this.leavesReqeustsWireResult);

    }

    showToast(message, title = 'success', variant = 'success') {

        const event = new ShowToastEvent({

            title,

            message,

            variant

        });

        this.dispatchEvent(event);

    }

}
```

- **Apex Classes**:
    - Write a class with methods to query and return leave requests for a specific user.
    - Develop a class to handle the custom logic of the approval process if needed.

LeaveRequestSampleData.cls

```
public with sharing class LeaveRequestSampleData {

    public static void createData(){

        Id currentUserId=UserInfo.getUserId();

        List<LeaveRequest__c> leaves=new List<LeaveRequest__c>();

        leaves.Add(new
LeaveRequest__c(User__c=currentUserId,From_Date__c=Date.newInstance(2025,
03, 10),To_Date__c=Date.newInstance(2025, 03, 11),Reason__c='For personal
reason',Status__c='Approved'));

        leaves.Add(new
LeaveRequest__c(User__c=currentUserId,From_Date__c=Date.newInstance(2025,
03, 15),To_Date__c=Date.newInstance(2025, 03,
15),Reason__c='Test',Status__c='Pending'));

        leaves.Add(new
LeaveRequest__c(User__c=currentUserId,From_Date__c=Date.newInstance(2025,
03, 19),To_Date__c=Date.newInstance(2025, 03, 19),Reason__c='For personal
reason',Status__c='Rejected'));



        insert leaves;

    }

}
```

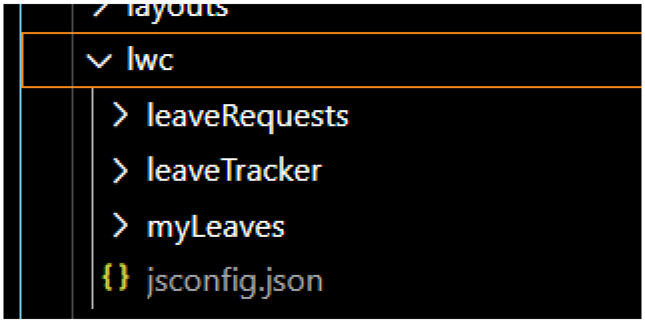
LeaveRequestController.cls

```
public with sharing class LeaveRequstController {

    @AuraEnabled(cacheable=true)

    public static List<LeaveRequest__c> getMyLeaves(){

        try {

            List<LeaveRequest__c> myLeaves=new List<LeaveRequest__c>();

            myLeaves=[SELECT
Id,Name,From_Date__c,To_Date__c,Reason__c,Status__c,Manager_Comment__c
FROM LeaveRequest__c WHERE User__c=:UserInfo.getUserId() ORDER BY
CreatedDate DESC];

            return myLeaves;
        } catch (Exception e) {
            throw new AuraHandledException(e.getMessage());
        }
    }

    @AuraEnabled(cacheable=true)
    public static List<LeaveRequest__c> getLeaveRequests(){
        try {
            List<LeaveRequest__c> myLeaves=new List<LeaveRequest__c>();
            myLeaves=[SELECT Id,Name,From_Date__c,To_Date__c,Reason__c,Status__c,Manager_Comment__c,User__r.ManagerId,User__r.Name FROM LeaveRequest__c
            WHERE User__r.ManagerId=:UserInfo.getUserId() ORDER BY CreatedDate DESC];
            return myLeaves;
        } catch (Exception e) {
            throw new AuraHandledException(e.getMessage());
        }
    }
}
```

- **Test Classes**:
    - ○ Write test classes to ensure your trigger logic works correctly and meets the required code coverage.

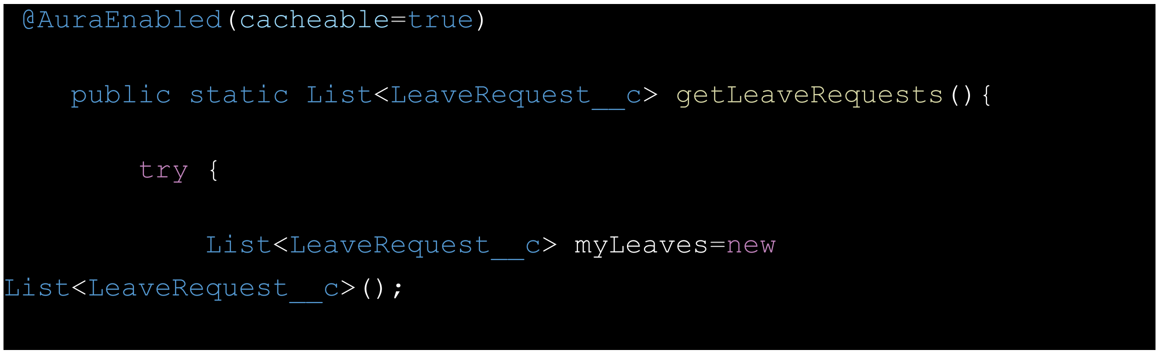
Phase 6: User Interface Development

- **Goal**: Create a user-friendly front end with Lightning Web Components (LWC).
- **LWC Development**:
  - Build an LWC to display a list of an employee's leave requests using a `lightning-datatable`.
  - Create another LWC for a custom form to submit a new leave request.



```
> layouts
∨ lwc
  > leaveRequests
  > leaveTracker
  > myLeaves
  {} jsconfig.json
```

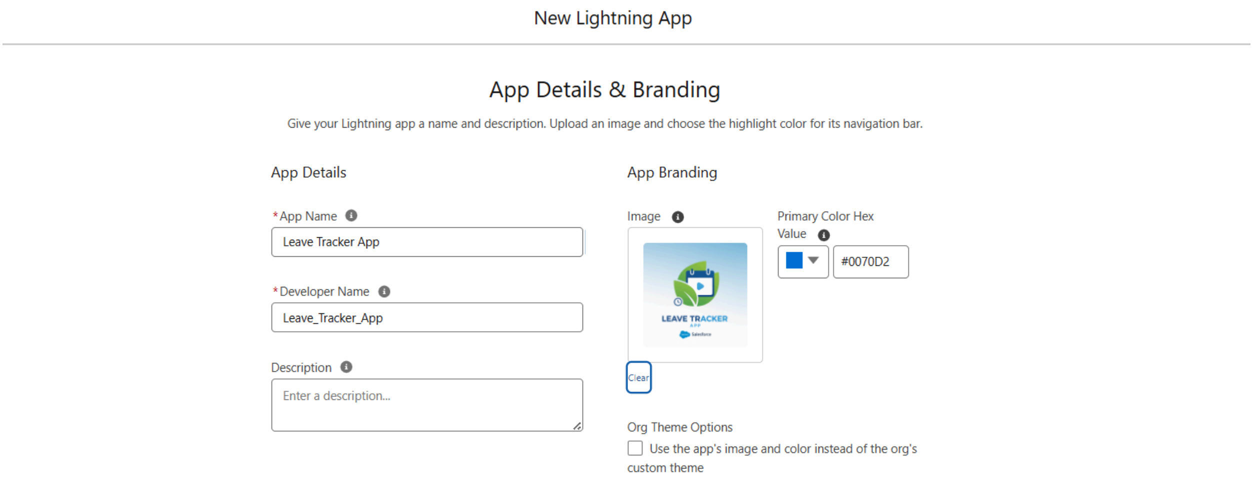
- **Apex with LWC**:
  - Use `@wire` to reactively fetch an employee's leave history from an Apex controller.
  - Use imperative Apex calls to submit new leave requests when the "Submit" button is clicked.



```
 @AuraEnabled(cacheable=true)

    public static List<LeaveRequest__c> getLeaveRequests(){

        try {

            List<LeaveRequest__c> myLeaves=new
List<LeaveRequest__c>();
```

```
            myLeaves=[SELECT Id,Name,From_Date__c,To_Date__c,Reason__c,Status__c,Manager_Comment__c,User__r.ManagerId,User__r.Name FROM LeaveRequest__c

            WHERE User__r.ManagerId=:UserInfo.getUserId() ORDER BY CreatedDate DESC];

            return myLeaves;

        } catch (Exception e) {

            throw new AuraHandledException(e.getMessage());

        }

    }
```

- **Lightning App Builder**:
    - Create a "Leave Tracker " app and add your custom LWC to a Lightning App Page or the Home Page.



New Lightning App

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: Leave Tracker App

*Developer Name: Leave_Tracker_App

Description: Enter a description...

App Branding

Image

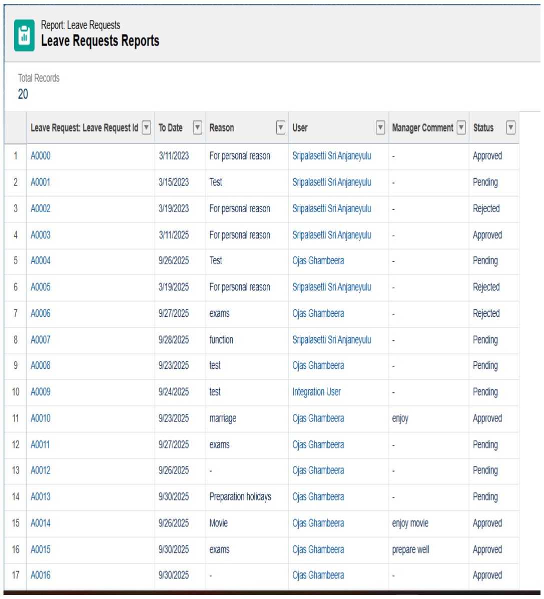
Primary Color Hex Value: #0070D2

Org Theme Options

□ Use the app's image and color instead of the org's custom theme

Phase 7: Reporting, Dashboards & Security

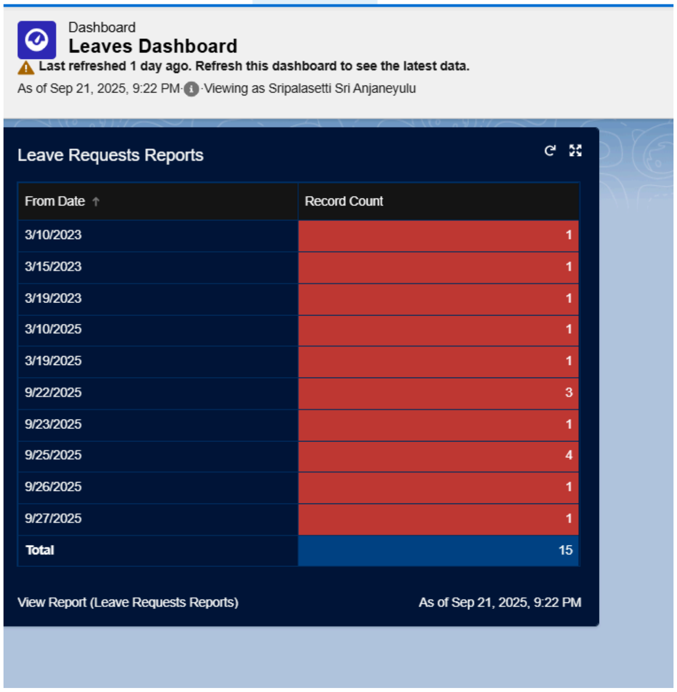
- **Goal**: Provide business insights and secure the application.
- **Reports & Dashboards**:
    - Create a report to show "Leave Requests by Status" (Pending, Approved, Rejected).
    - Build a dashboard for managers to visualize their team's upcoming leave.
    - Use a dynamic dashboard to show each manager only their own team's data.



Report: Leave Requests
Leave Requests Reports

Total Records
20

	Leave Request: Leave Request Id	To Date	Reason	User	Manager Comment	Status
1	A0000	3/11/2023	For personal reason	Sripalasetti Sri Anjaneyulu	-	Approved
2	A0001	3/15/2023	Test	Sripalasetti Sri Anjaneyulu	-	Pending
3	A0002	3/19/2023	For personal reason	Sripalasetti Sri Anjaneyulu	-	Rejected
4	A0003	3/11/2025	For personal reason	Sripalasetti Sri Anjaneyulu	-	Approved
5	A0004	9/26/2025	Test	Ojas Ghambeera	-	Pending
6	A0005	3/19/2025	For personal reason	Sripalasetti Sri Anjaneyulu	-	Rejected
7	A0006	9/27/2025	exams	Ojas Ghambeera	-	Rejected
8	A0007	9/28/2025	function	Sripalasetti Sri Anjaneyulu	-	Pending
9	A0008	9/23/2025	test	Ojas Ghambeera	-	Pending
10	A0009	9/24/2025	test	Integration User	-	Pending
11	A0010	9/23/2025	marriage	Ojas Ghambeera	enjoy	Approved
12	A0011	9/27/2025	exams	Ojas Ghambeera	-	Pending
13	A0012	9/26/2025	-	Ojas Ghambeera	-	Pending
14	A0013	9/30/2025	Preparation holidays	Ojas Ghambeera	-	Pending
15	A0014	9/26/2025	Movie	Ojas Ghambeera	enjoy movie	Approved
16	A0015	9/30/2025	exams	Ojas Ghambeera	prepare well	Approved
17	A0016	9/30/2025	-	Ojas Ghambeera	-	Approved



Dashboard
Leaves Dashboard
⚠ Last refreshed 1 day ago. Refresh this dashboard to see the latest data.
As of Sep 21, 2025, 9:22 PM · Viewing as Sripalasetti Sri Anjaneyulu

Leave Requests Reports

From Date ↑	Record Count
3/10/2023	1
3/15/2023	1
3/19/2023	1
3/10/2025	1
3/19/2025	1
9/22/2025	3
9/23/2025	1
9/25/2025	4
9/26/2025	1
9/27/2025	1
Total	15

View Report (Leave Requests Reports)
As of Sep 21, 2025, 9:22 PM

- **Security:**
    - Use Field-Level Security to hide sensitive information from employees or managers.
    - Set login IP ranges or hours if required by the company's policy.

Phase 8: Data Management & Deployment

1. **Data Management**

- Data Model Design

    - Custom Object: Leave_Request__c (to store leave applications).

    - Key Fields:

        - Employee Name (Lookup → User)

        - From Date

        - To Date

        - Reason

        - Status (Picklist: Pending, Approved, Rejected)

        - Manager Comment

```
import { api, LightningElement, wire } from 'lwc';

import getLeaveRequests from
'@salesforce/apex/LeaveRequstController.getLeaveRequests';

import { ShowToastEvent } from 'lightning/platformShowToastEvent';
import Id from '@salesforce/user/Id';
import { refreshApex } from '@salesforce/apex';

const COLUMNS = [
    { label: 'Request Id', fieldName: 'Name', cellAttributes: { class: { fieldName: 'cellClass' } } },
    { label: 'User', fieldName: 'userName', cellAttributes: { class: { fieldName: 'cellClass' } } },
    { label: 'From Date', fieldName: 'From_Date__c', cellAttributes: { class: { fieldName: 'cellClass' } } },
    { label: 'To Date', fieldName: 'To_Date__c', cellAttributes: { class: { fieldName: 'cellClass' } } },
    { label: 'Reason', fieldName: 'Reason__c', cellAttributes: { class: { fieldName: 'cellClass' } } },
    { label: 'Status', fieldName: 'Status__c', cellAttributes: { class: { fieldName: 'cellClass' } } },
    { label: 'Manager Comment', fieldName: 'Manager_Comment__c', cellAttributes: { class: { fieldName: 'cellClass' } } },
    {
        type: "button", typeAttributes: {
            label: 'Edit',
            name: 'Edit',
            title: 'Edit',
            value: 'edit',
            disabled: { fieldName: 'isEditDisabled' }
        }, cellAttributes: { class: { fieldName: 'cellClass' } }
    }
```

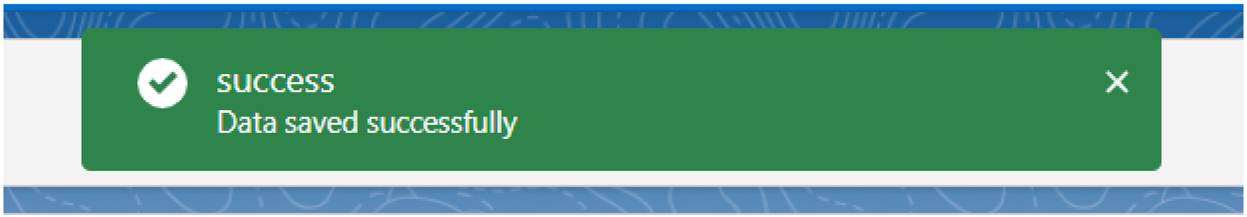
```
];

export default class LeaveRequests extends LightningElement {

    columns = COLUMNS;


    leavesReqeusts = [];

    leavesReqeustsWireResult;

    showModalPopup = false;

    objectApiName = 'LeaveRequest__c';

    recordId = '';

    currentUserId = Id;

    @wire(getLeaveRequests)

    wiredMyLeaves(result) {

        this.leavesReqeustsWireResult = result;

        if (result.data) {

            this.leavesReqeusts = result.data.map(a => ({

                ...a,

                userName: a.User__r.Name,

                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 occured while fetching my leaves- ',
result.error);
```

```
        }

    }


    get noRecordsFound() {

        return this.leavesReqeusts.length == 0;

    }


    newRequestClickHandler() {

        this.showModalPopup = true;

        this.recordId = '';

    }

    popupCloseHandler() {

        this.showModalPopup = false;

    }


    rowActionHandler(event) {

        this.showModalPopup = true;

        this.recordId = event.detail.row.Id;

    }


    successHandler(event) {

        this.showModalPopup = false;

        this.showToast('Data saved successfully');

        this.refreshGrid();
    }


    @api

    refreshGrid() {

        refreshApex(this.leavesReqeustsWireResult);

    }

    showToast(message, title = 'success', variant = 'success') {

        const event = new ShowToastEvent({

            title,

            message,

            variant

        });

        this.dispatchEvent(event);

    }

}
```

- Data Security
    - Applied Field-Level Security (FLS) to hide sensitive fields like Manager Comments from employees.

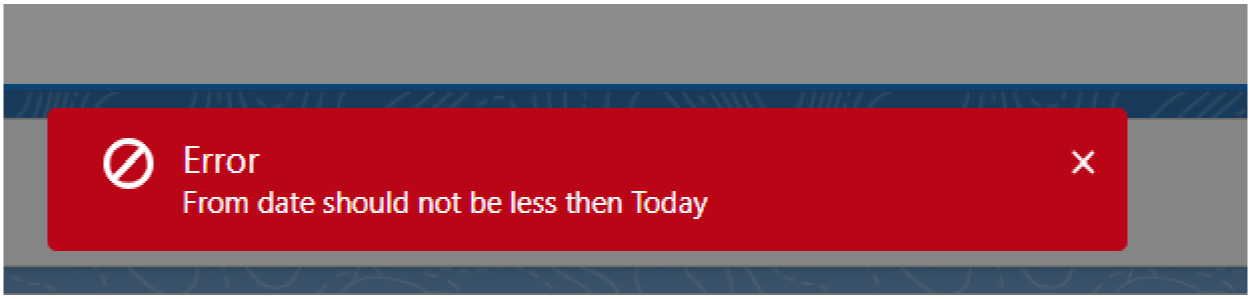
- Used Profiles & Permission Sets to restrict access (Employee vs Manager roles).

- Data Loading
  - Used Salesforce Data Import Wizard / Data Loader to upload sample employee and leave records for testing.



success
Data saved successfully

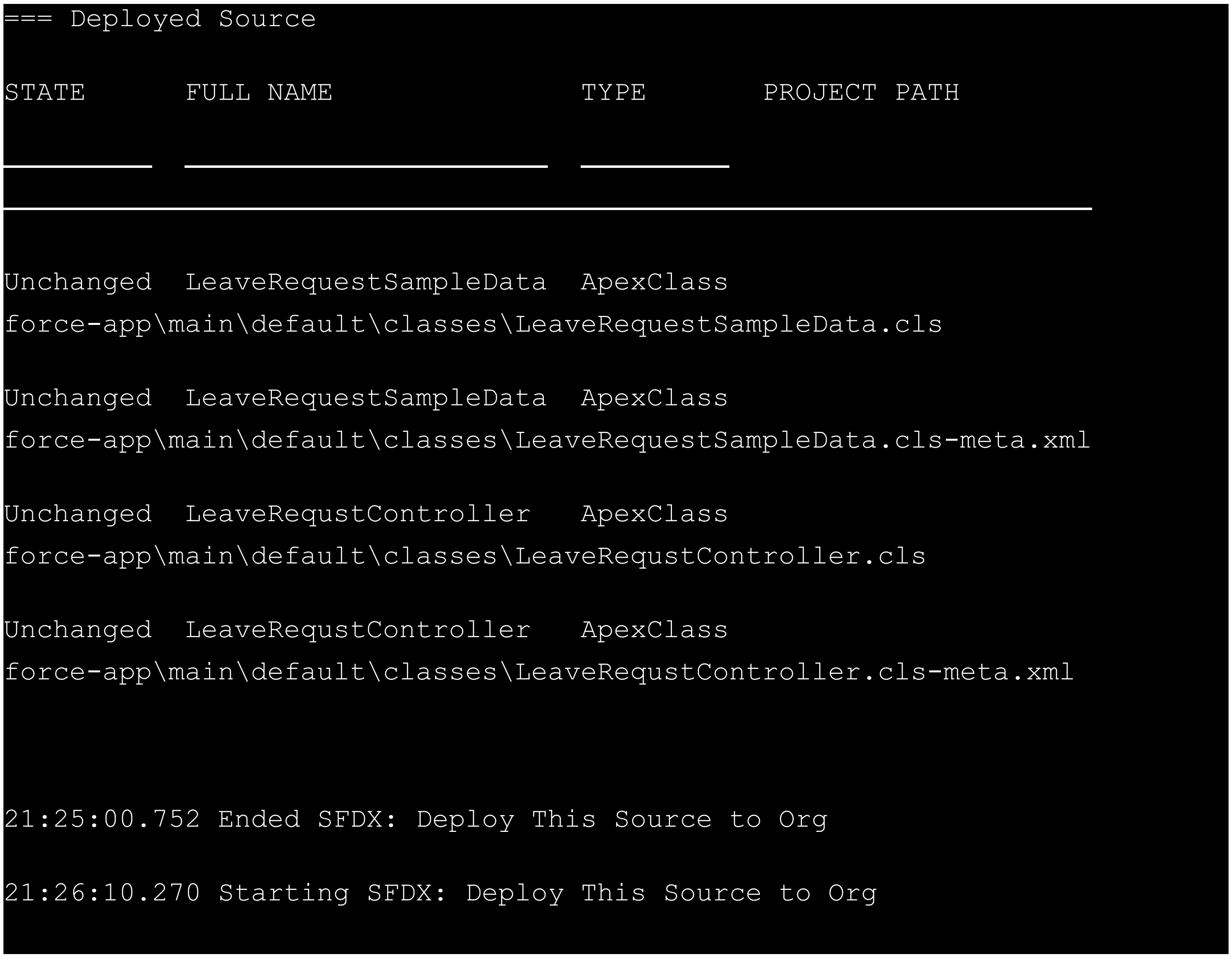
- Data Validation
  - Validation Rule: To ensure *From Date* < *To Date*.
  - Prevented overlapping leave requests for the same employee.



Error
From date should not be less then Today

2. **Deployment Process**

- Development Environment
  - Built and tested the app in Salesforce Developer Org using VS Code + SFDX.



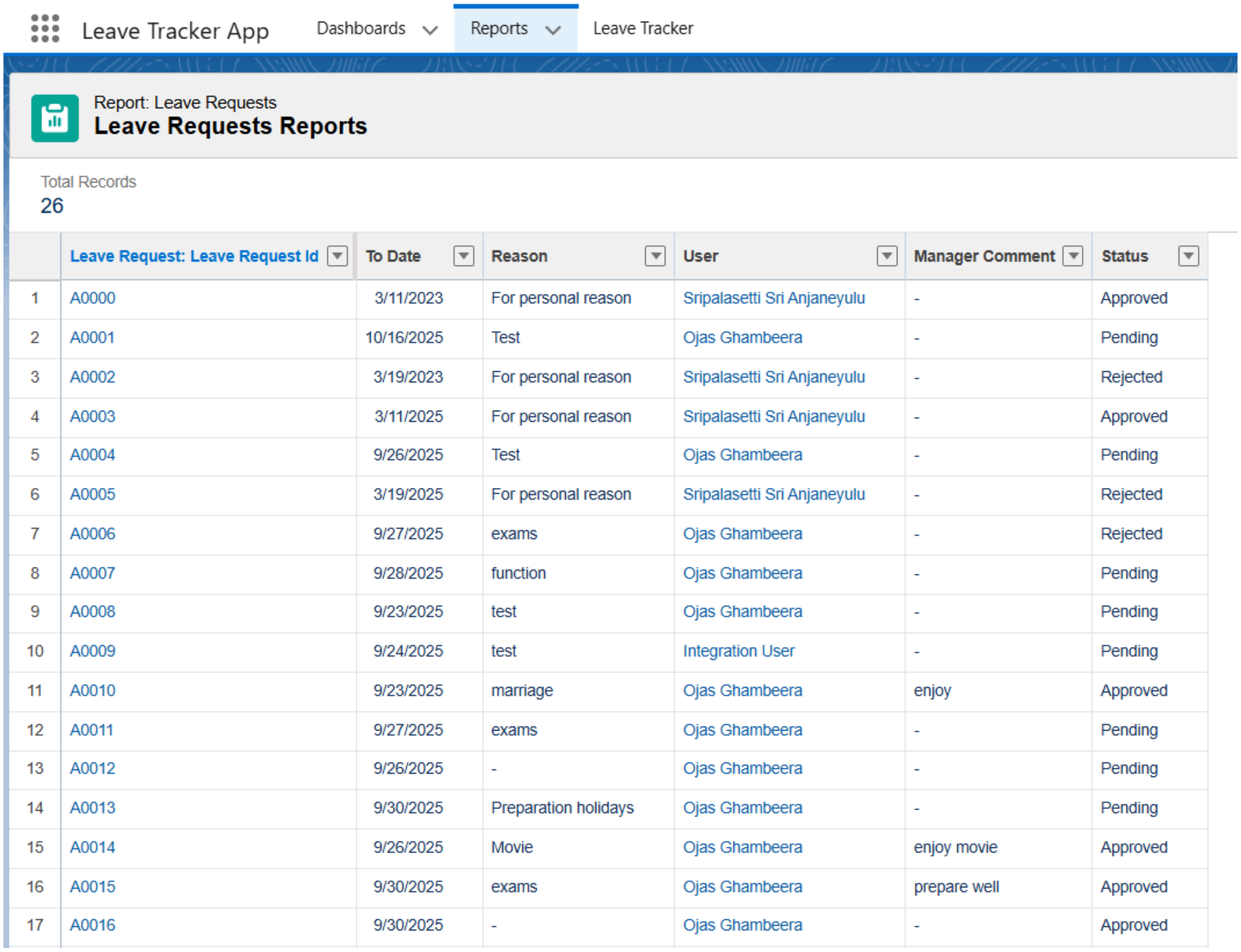
```
=== Deployed Source

STATE      FULL NAME               TYPE       PROJECT PATH

─────────  ──────────────────────  ─────────
──────────────────────────────────────────────────────────────────

Unchanged  LeaveRequestSampleData  ApexClass
force-app\main\default\classes\LeaveRequestSampleData.cls

Unchanged  LeaveRequestSampleData  ApexClass
force-app\main\default\classes\LeaveRequestSampleData.cls-meta.xml

Unchanged  LeaveRequstController   ApexClass
force-app\main\default\classes\LeaveRequstController.cls

Unchanged  LeaveRequstController   ApexClass
force-app\main\default\classes\LeaveRequstController.cls-meta.xml



21:25:00.752 Ended SFDX: Deploy This Source to Org

21:26:10.270 Starting SFDX: Deploy This Source to Org
```

- Change Set Deployment
  - Created an Outbound Change Set with Apex Classes, LWC components, Custom Objects, and Profiles.
  - Deployed the app to Test/Sandbox Org for UAT (User Acceptance Testing).

- Testing
  - Verified that leave requests can be created, approved, and tracked.
  - Ensured UI works properly in Lightning App.
  - Security testing for restricted fields.
- Final Deployment
  - Deployed to Production Org after successful UAT.
  - Migrated sample data using Data Loader for initial setup.
- Post-Deployment Activities
  - Assigned app permission sets to employees and managers.
  - Shared documentation and demo with users.

Phase 9: Reporting, Dashboards & Security Review

- **Goal:** Create a robust system for monitoring business insights and securing sensitive data.
- **Reports:**
  - Create a report to track "Leave Requests by Status" (e.g., Pending, Approved, Rejected).
  - Build a report on "Total Leave Taken by Employee" to help HR monitor leave balances.
  - Create a "Departmental Leave Calendar" report to help managers visualize team availability.



Report: Leave Requests

Leave Requests Reports

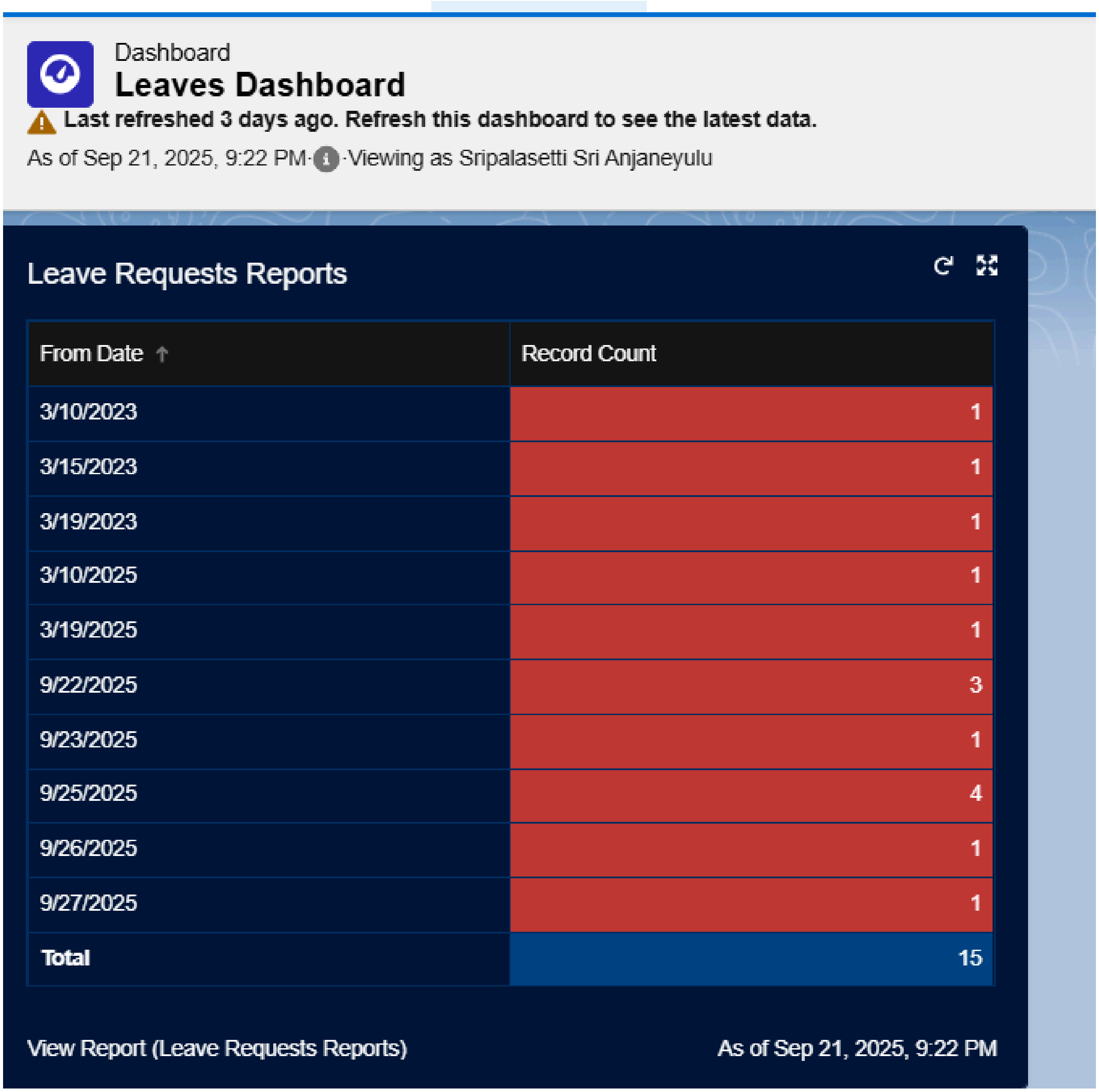
Total Records

26

	Leave Request: Leave Request Id	To Date	Reason	User	Manager Comment	Status
1	A0000	3/11/2023	For personal reason	Sripalasetti Sri Anjaneyulu	-	Approved
2	A0001	10/16/2025	Test	Ojas Ghambeera	-	Pending
3	A0002	3/19/2023	For personal reason	Sripalasetti Sri Anjaneyulu	-	Rejected
4	A0003	3/11/2025	For personal reason	Sripalasetti Sri Anjaneyulu	-	Approved
5	A0004	9/26/2025	Test	Ojas Ghambeera	-	Pending
6	A0005	3/19/2025	For personal reason	Sripalasetti Sri Anjaneyulu	-	Rejected
7	A0006	9/27/2025	exams	Ojas Ghambeera	-	Rejected
8	A0007	9/28/2025	function	Ojas Ghambeera	-	Pending
9	A0008	9/23/2025	test	Ojas Ghambeera	-	Pending
10	A0009	9/24/2025	test	Integration User	-	Pending
11	A0010	9/23/2025	marriage	Ojas Ghambeera	enjoy	Approved
12	A0011	9/27/2025	exams	Ojas Ghambeera	-	Pending
13	A0012	9/26/2025	-	Ojas Ghambeera	-	Pending
14	A0013	9/30/2025	Preparation holidays	Ojas Ghambeera	-	Pending
15	A0014	9/26/2025	Movie	Ojas Ghambeera	enjoy movie	Approved
16	A0015	9/30/2025	exams	Ojas Ghambeera	prepare well	Approved
17	A0016	9/30/2025	-	Ojas Ghambeera	-	Approved

- **Report Types:**
  - Create a custom report type that joins the `User` object with the `Leave_Request__c` object.
- **Dashboards:**
  - Build a "Manager's Leave Dashboard" that includes a pie chart of requests by status and a component showing upcoming leave.

- **Dynamic Dashboards:**
  - Implement a dynamic dashboard for managers so that each manager only sees data for their direct reports.



Dashboard

Leaves Dashboard

⚠ Last refreshed 3 days ago. Refresh this dashboard to see the latest data.

As of Sep 21, 2025, 9:22 PM · ℹ · Viewing as Sripalasetti Sri Anjaneyulu

Leave Requests Reports

From Date ↑	Record Count
3/10/2023	1
3/15/2023	1
3/19/2023	1
3/10/2025	1
3/19/2025	1
9/22/2025	3
9/23/2025	1
9/25/2025	4
9/26/2025	1
9/27/2025	1
Total	15

View Report (Leave Requests Reports)

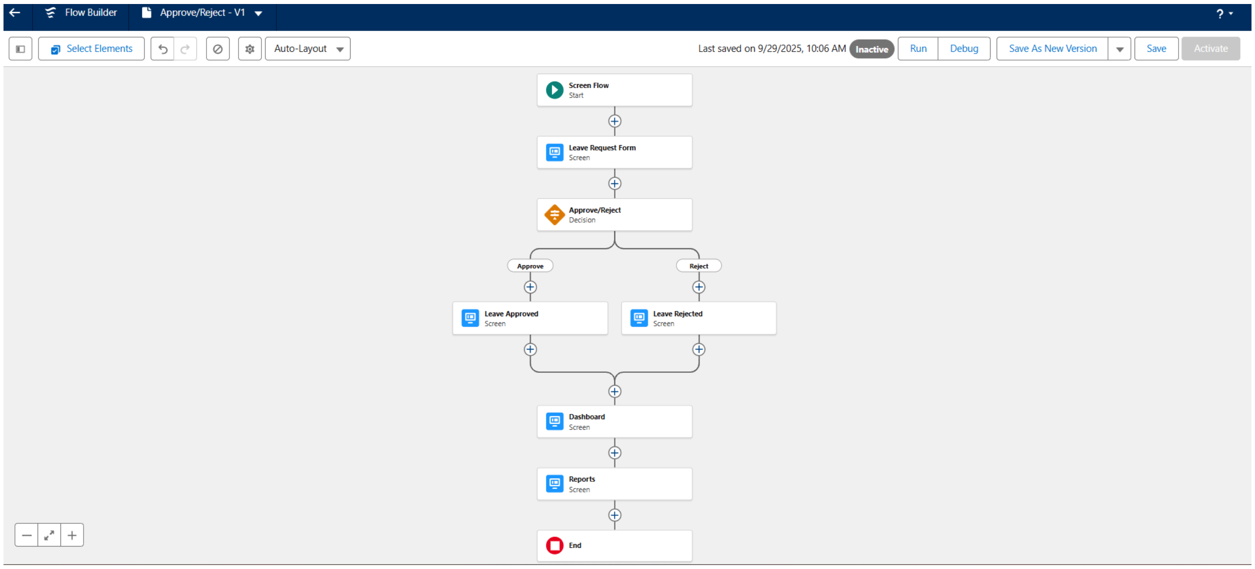
As of Sep 21, 2025, 9:22 PM

Phase 10: Final Presentation & Demo Day

Goal: Deliver the final product and prepare for a professional handoff.

- **Pitch Presentation**: Outline the initial business problem, the Salesforce LWC solution, and key benefits (time savings, accuracy, improved visibility).

- **Demo Walkthrough**: Conduct a step-by-step demo:
  1. Employee submits a new leave request.
  2. Manager reviews and approves/rejects with notifications.
  3. Dashboard displays real-time team leave status.
- **Testing & Validation**: All workflows and automations were tested with sample data to confirm correctness and compliance with requirements.



Last saved on 9/29/2025, 10:06 AM

- **Handoff Documentation**: Provide a comprehensive user guide (for employees/managers) and a technical guide (for admins/future developers).

Demo Video Link: https://drive.google.com/file/d/1Eu2EwCYPHxUjq5PkXK0R2lyeG-rrKHm_/view?usp=drive_link