Project Title: Smart Leave Tracker – Employee Leave Management SystemPhase 2: Org Setup & Configuration

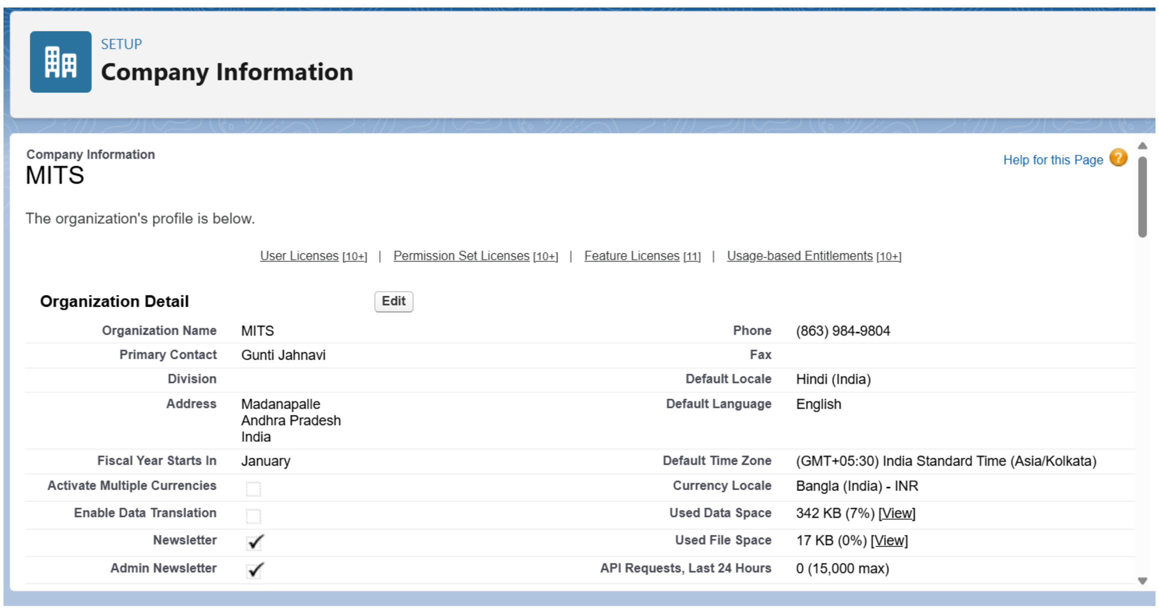
1. Salesforce Edition & Developer Org Setup

Objective: Create a dedicated development environment for the Leave Tracker Project.

Action Items:

- Create new Salesforce Developer Org
- Name: LeaveTracker_DevOrg
- Enable all necessary features for HR automation

2. Company Profile Setup



Navigation: Setup → Company Information

Configure the following settings:

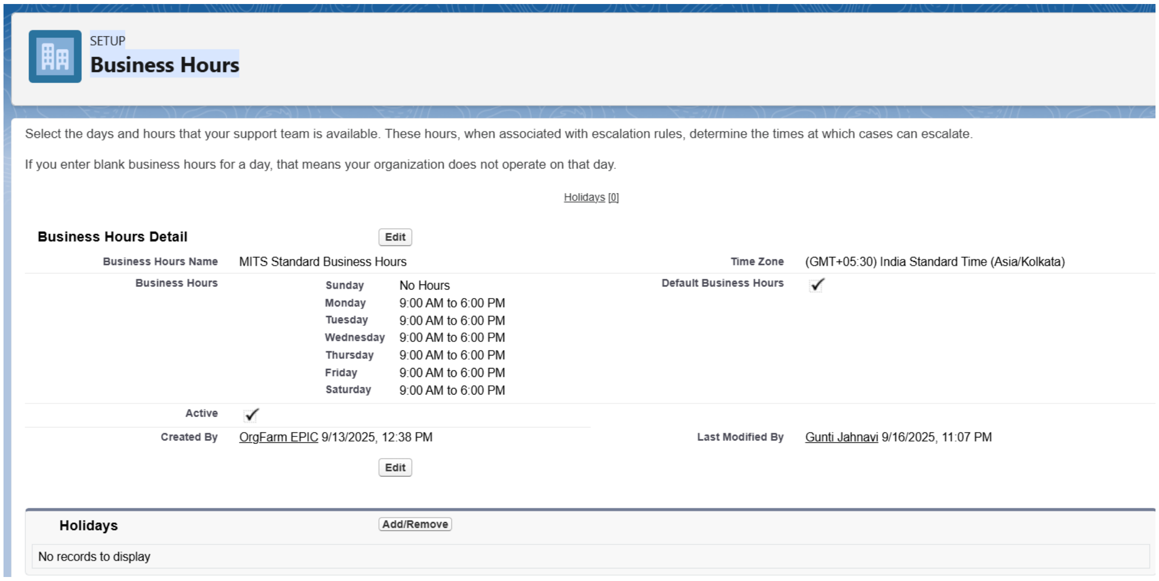
- Company Name: Your Company – Leave Tracker CRM

- Primary Contact: Manager• Default Locale: English (India) • Currency: INR (₹)• Timezone: Asia/Kolkat

3. Business Hours & Holidays Setup

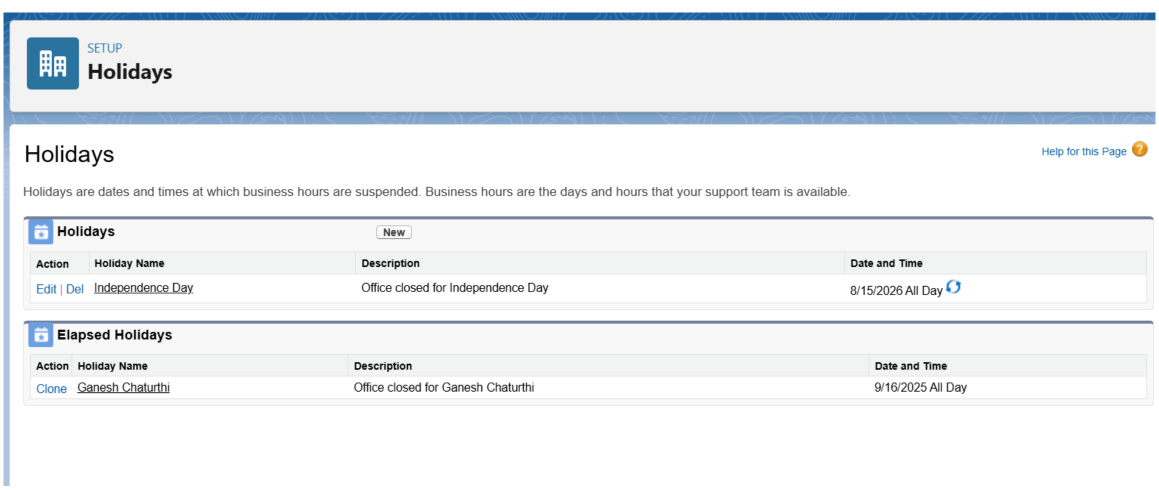
- Business Hours:

Navigation: Setup → Company Settings → Business Hours. Create 'Standard Business Hours': Monday-Friday, 9:00 AM – 6:00 PM.



- Holidays:

Navigation: Setup → Company Settings → Holidays. Add relevant national and regional holidays for India.

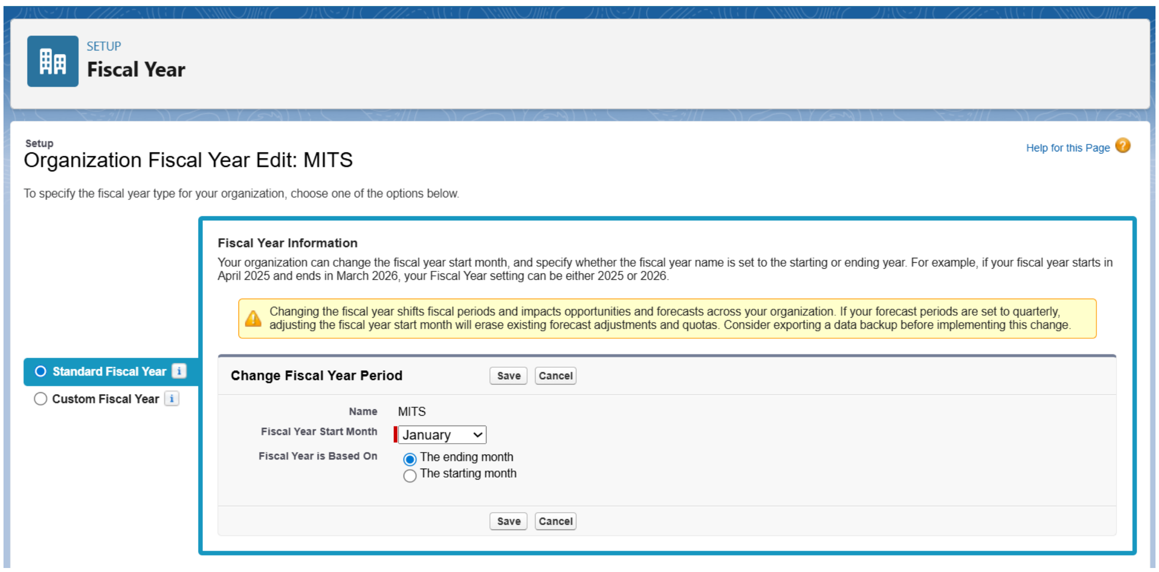


4. Fiscal Year Settings

Navigation: Setup → Company Settings → Fiscal Year

Configuration: Type: Standard Fiscal Year. Start: April. End:

March (Common fiscal year in India).

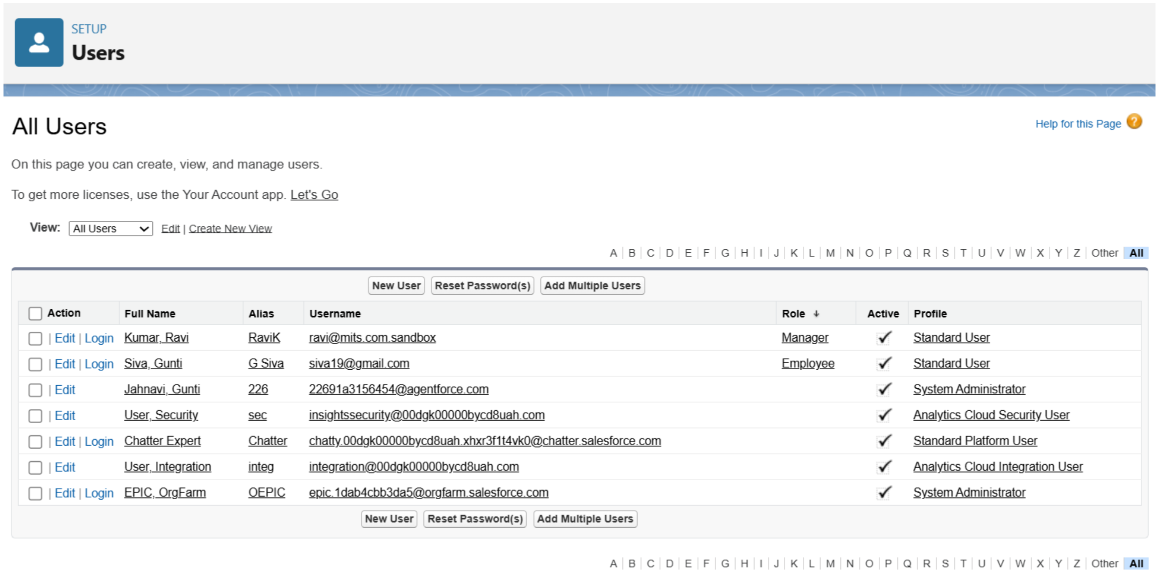


5. User Setup & Licenses

Navigation: Setup → Users → New User

For this project, only two user types are configured:

- Manager: Salesforce License, Manager Profile
- Employee: Salesforce Platform License, Employee Profile

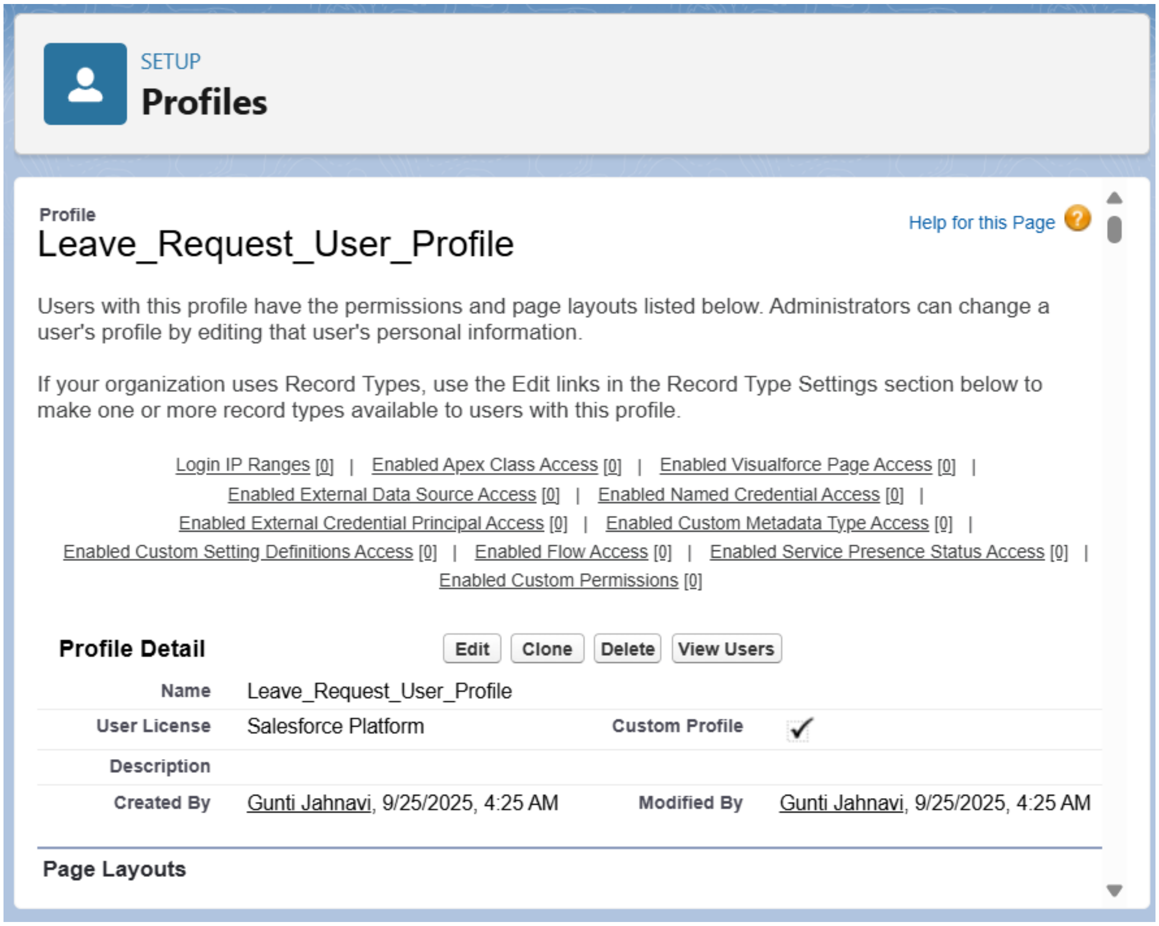


6. Profiles Configuration

Navigation: Setup → Profiles → Clone Standard Profile

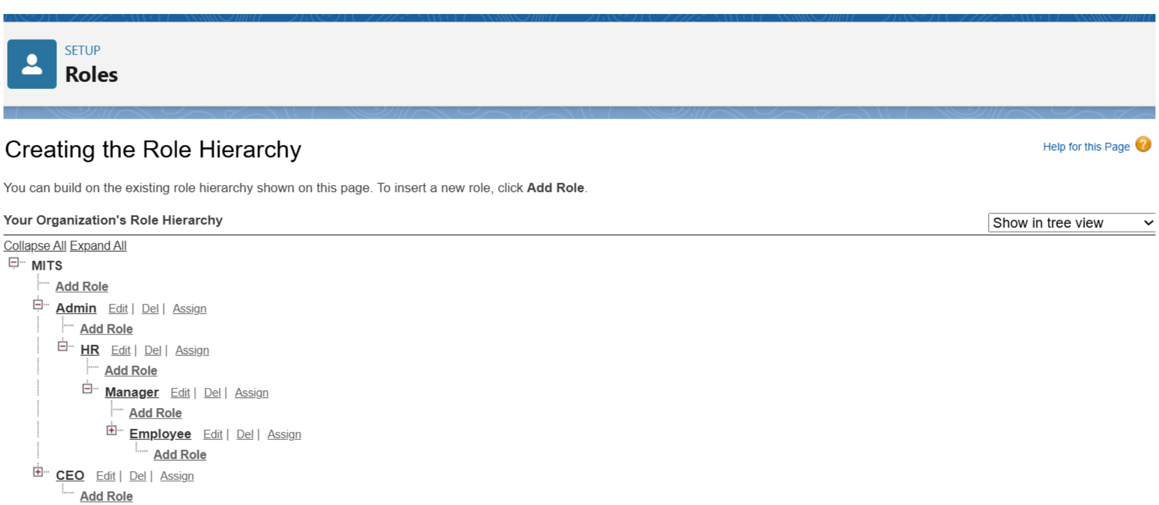
Custom Profiles to Create:

• Manager Profile: Read/Write access to team Leave Request• Employee Profile: Read access only to own Leave Requests



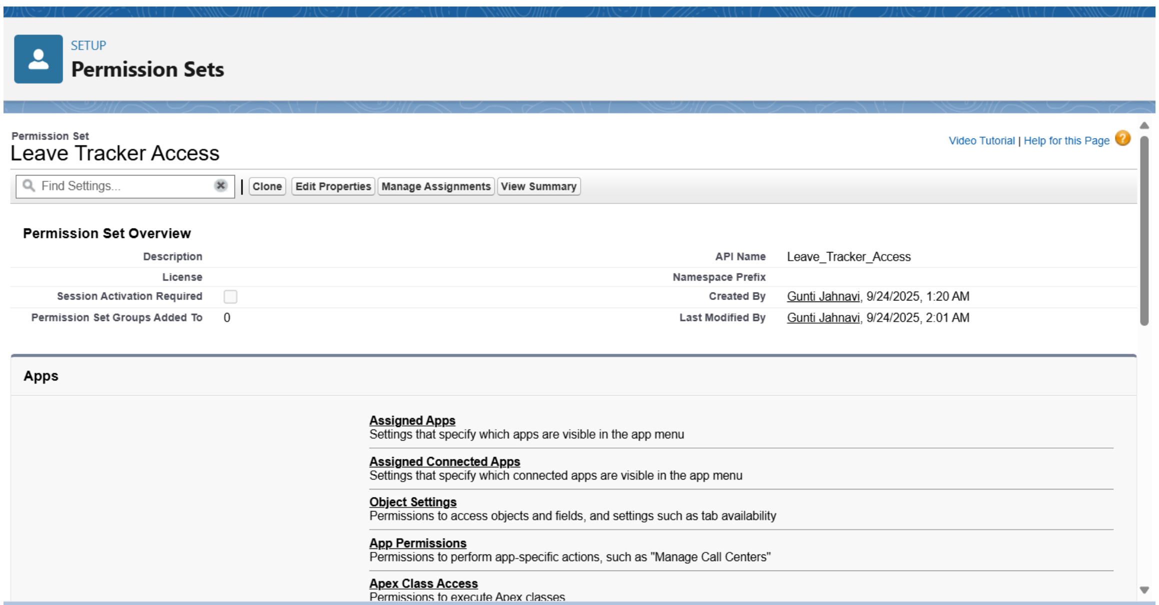
7.Roles

1. Click the **Setup (gear icon)** in Salesforce.
2. In the **Quick Find** box, type **Roles**.
3. Click on **Roles** under **Users**.



7.Create Permission Set

1. Go to **Setup → Permission Sets → New**.
2. Name it **Leave Tracker Access**.
3. Leave User License **--None--** (applies to all users).
4. Click **Save**.

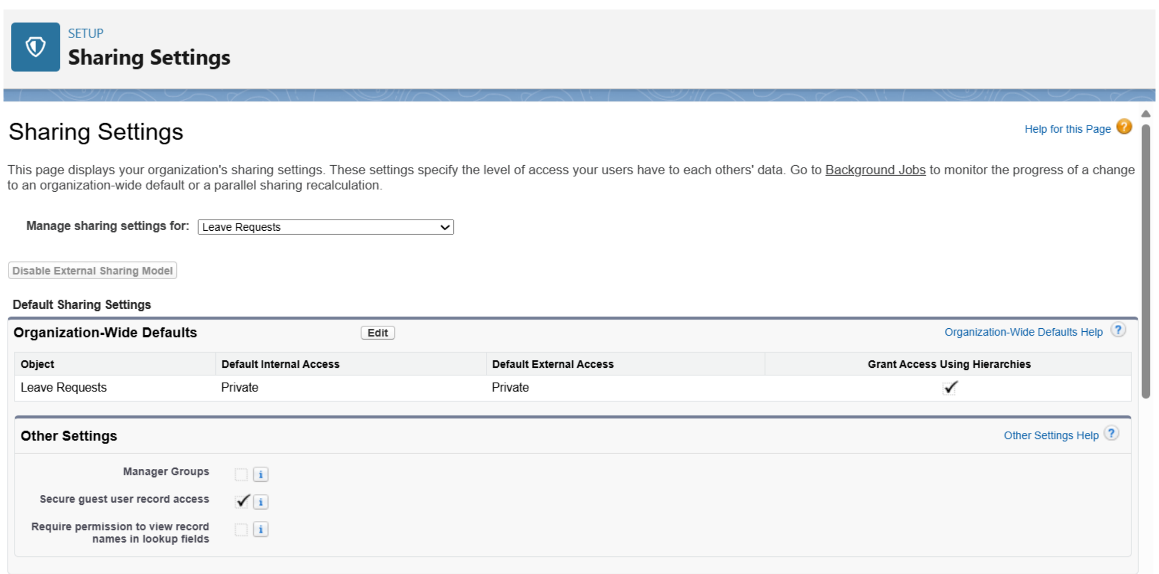


8. OWD

Purpose:
Controls **baseline access** to records for all users in Salesforce.

Steps:

1. Go to **Setup → Security → Sharing Settings**.
2. Find your **Leave Requests object**.
3. Set the **Default Internal Access** and **Default External Access**:
   - **Private** → Users can only see records they own. *(Recommended for sensitive data like Leave Requests)*
   - **Public Read Only** → All users can view but not edit.
   - **Public Read/Write** → All users can view and edit.
4. Click **Save**.



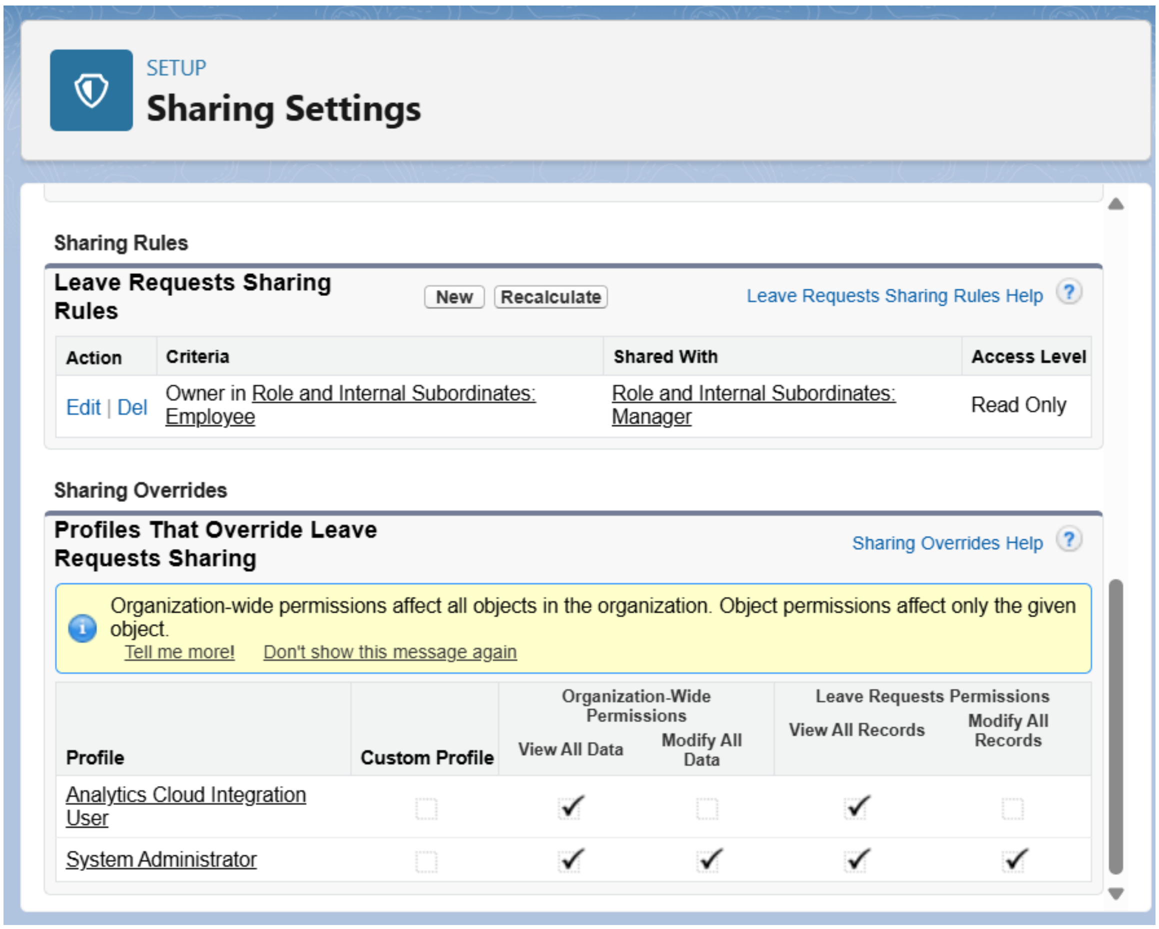
9 Sharing Rules

Purpose:
Allows you to **open access beyond OWD**, e.g., let Managers see their team's leave requests.

Steps:

1. Go to **Setup → Security → Sharing Settings → Leave Requests → Sharing Rules → New**

2. **Rule Type:** Based on Role, Public Group, or Criteria
3. Define **Access Level**:
    - Read Only → Can view records
    - Read/Write → Can edit records
4. Click **Save**.



10. Login Access Policies

Purpose:
Controls whether **admins or support users can log in as other users**.

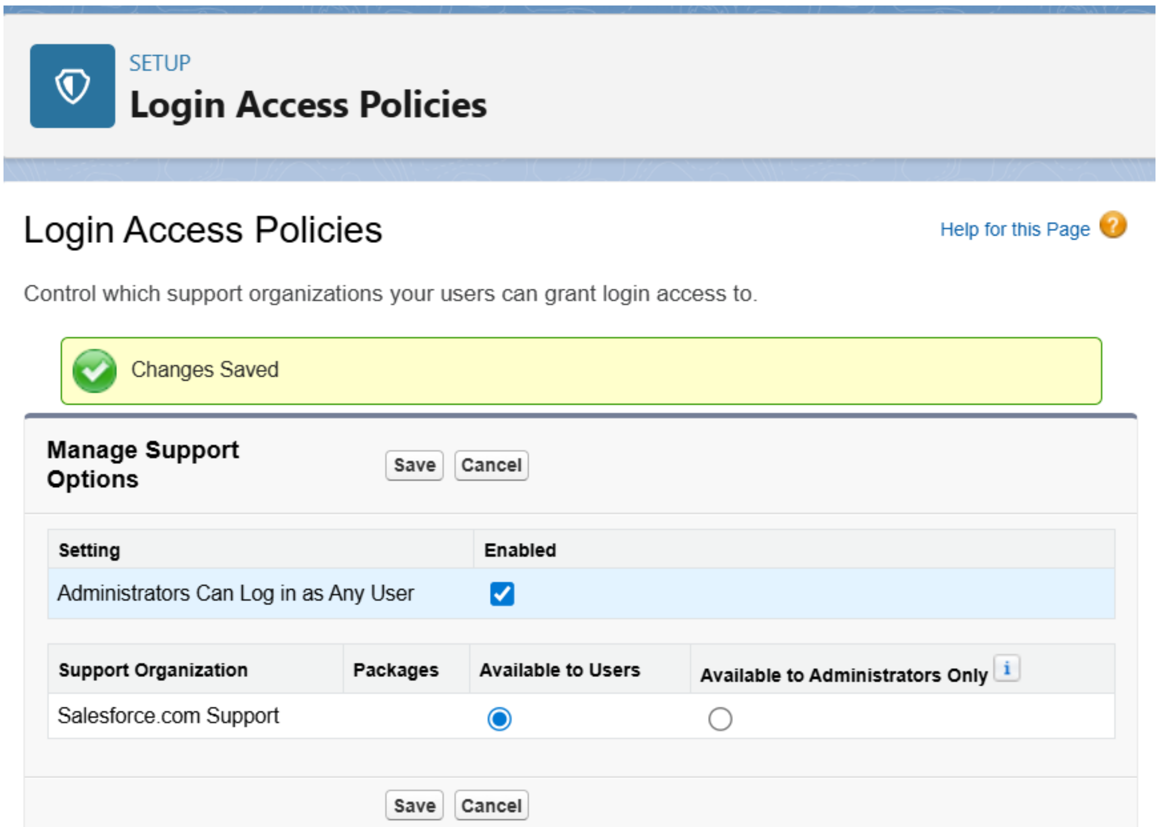
Steps:

1. Go to **Setup → Security → Login Access Policies**.
2. Options:**Administrators Can Log in as Any User** → Gives admin full access.

Users Can Grant Login Access → Users explicitly allow access for support.

3. Check boxes as per your org policy.

4. Click **Save**.



11. Dev Org Setup

Purpose:
Configures **developer orgs** for testing and building customizations.

Steps:

1. Go to **Setup → Company Settings → Company Information**.

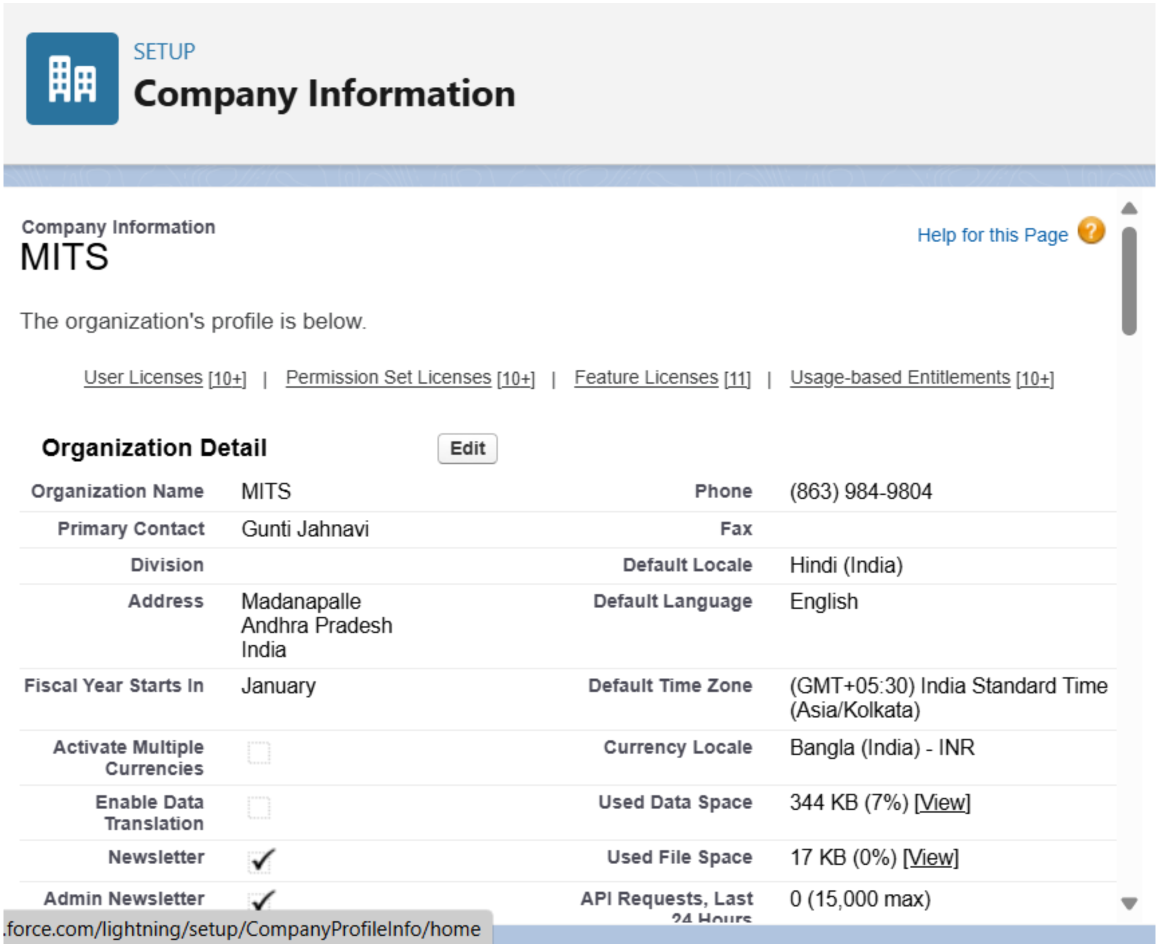
2. Check your **Org Edition**: Developer Edition, Enterprise Edition, etc.

3. Verify:

    - Users, Profiles, Roles are set up

    - OWD, Sharing Rules configured

- Sandbox access if needed



12. Sandbox Usage

1. Sandbox is a safe copy of your org for testing.
2. Create it via **Setup → Sandboxes** and choose type (Developer, Partial, Full).
3. Test changes here before touching Production.

13. Deployment Basics

1. Deployment moves changes from Sandbox to Production.
2. Use **Change Sets**: add components in Sandbox → upload → deploy in Production.
3. Always test after deploying to confirm it works