

# LearnSmart CRM

Salesforce-Based Education Management & Student Engagement System

## Problem Statement

Educational institutions face challenges in managing student admissions, tracking performance, maintaining attendance, and ensuring effective communication between teachers, students, and parents. Many schools and colleges rely on manual processes or separate systems, leading to fragmented data, inefficiencies, and communication delays. Without a centralized CRM, educational management struggles with poor visibility into student progress, delayed decision-making, and inconsistent engagement. To address these challenges, a Salesforce-based solution, LearnSmart CRM, is proposed to streamline academic processes, automate communication, and enhance student experience.

## Requirement Gathering

- Institutions struggle with manual student enrollment and admission tracking.
- Teachers lack integrated tools to monitor student performance and attendance.
- Parents and students require better communication channels.
- Manual grading and reporting cause delays in academic assessments.
- School administrators need dashboards to monitor admissions, attendance, and performance trends.
- Lack of reminders for exams, fees, and assignments reduces engagement.

## Objectives

The primary objectives of LearnSmart CRM are to:

- Centralize student, teacher, course, and attendance data within Salesforce.
- Automate admission workflows and communication alerts for smoother onboarding.
- Provide teachers with tools to track and evaluate student performance.
- Offer a parent–student portal for transparency in academics and communication.
- Enable management to monitor real-time academic and administrative metrics.

- Integrate email/SMS notifications for reminders, results, and announcements.

## Stakeholder Analysis

- Students: Need access to schedules, assignments, and grades.
- Teachers: Require performance tracking and communication tools.
- Parents: Need visibility into student progress and attendance.
- Administrators: Manage admissions, academic reports, and overall analytics.
- System Admins: Configure Salesforce, manage users, and maintain security.

## Business Process Mapping

### ■ Current Process:

- Admissions and fee payments handled manually.
- Attendance and grades recorded on spreadsheets.
- Communication via notice boards or phone calls.
- Reports prepared manually.

### ■ Proposed Salesforce Process:

- Students register online through Experience Cloud.
- Teachers update attendance and grades digitally.
- Automated reminders sent for classes, exams, and fees.
- Real-time dashboards display academic and operational insights.

## Industry-Specific Use Case Analysis

- Education sector requires digital transformation for improved learning outcomes.
- Salesforce CRM helps streamline academic workflows and student engagement.
- Automation reduces manual workload for teachers and administrators.
- Reports and dashboards enable data-driven educational decisions.
- Integrated communication enhances collaboration between schools, students, and parents.

### AppExchange Exploration

- Salesforce Education Cloud: Provides education data model but limited customization

for smaller institutions.

- Blackboard CRM: Offers learning management but lacks Salesforce integration.
- Decision: Develop a custom Salesforce CRM solution tailored for schools and colleges focusing on admissions, performance tracking, and communication.

## Conclusion

LearnSmart CRM provides a Salesforce-powered solution for schools and colleges to manage academic, administrative, and communication processes efficiently. By centralizing student data, automating workflows, and enabling digital collaboration, LearnSmart CRM improves the overall education experience for students, parents, and educators. The platform ensures transparency, enhances engagement, and supports data-driven academic management—helping institutions deliver quality education in an organized, connected, and measurable way.

# Phase 2: Org Setup & Configuration

## 1.Salesforce Edition

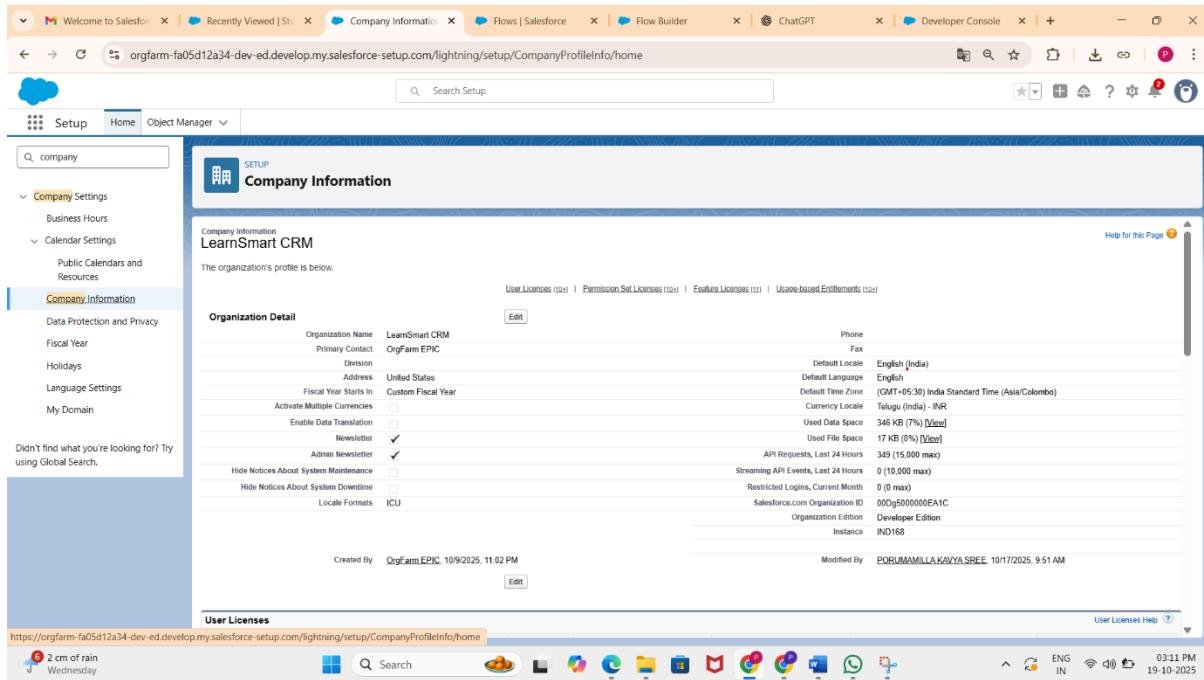
Choose the **Salesforce Developer Edition** for your project.

- Provides all tools needed for practice, development, and testing.
- Free to use, supports all key features, and is ideal for student projects.

The image shows two screenshots from a Windows 10 desktop. The top screenshot is a web browser window titled 'Developer Edition with Agentforce' showing the sign-up process for a Salesforce Developer Edition. It features a blue-themed interface with a central form for entering first name (PORUMAMILLA), last name (KAVYA SREE), job title (Student), work email (kavyasree3223529@genforce.com), company (LearnSmart EduCRM), and country/region (India). Below the form is a checkbox for agreeing to the terms and conditions. A large button at the bottom right says 'Sign Me Up'. The bottom screenshot is a Gmail inbox window titled 'Welcome to Salesforce: Reset your password'. It shows an email from support@salesforce.com with the subject 'Welcome to your Developer Edition'. The email contains a welcome message, a 'Reset Password' button, and a URL (https://orgfarm-fa05d12a34-dev-ed.develop.my.salesforce.com). It also includes a note about the username (kavyasree3223529578@genforce.com) and a message about the org's active status.

## 2. Company Profile Setup

- Navigate to **Setup → Company Information**.
- Update your organization details:
  - Organization Name
  - Time Zone: **GMT +05:30**
  - Currency: **INR**
  - Default Language: **English (India)**
- Confirm the main contact details.



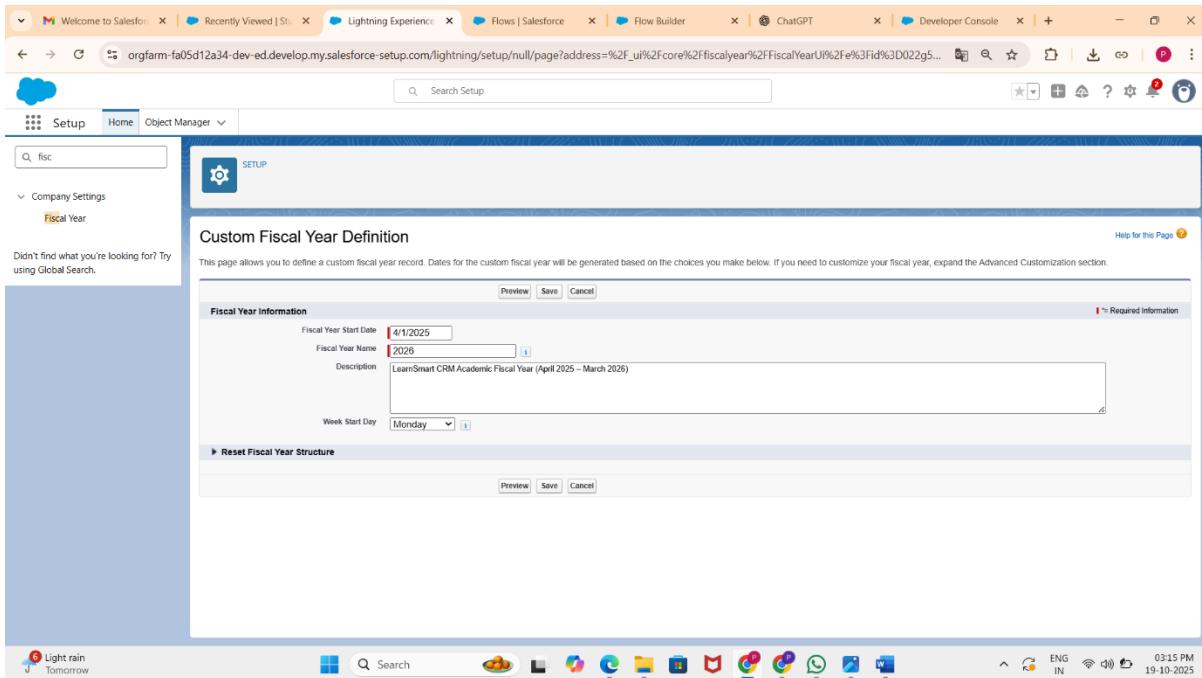
## 3. Business Hours & Holidays

- **Business Hours:** Setup → Business Hours → New
  - Working Days: Monday to Saturday
  - Working Hours: 9:00 AM – 6:00 PM
- **Holidays:** Setup → Holidays → New
  - Add important holidays like Independence Day and Diwali.
  - Ensures workflows and escalation rules respect non-working days.

## 4. Fiscal Year Settings

- Navigate to Setup → Fiscal Year
- Select Standard Fiscal Year
- Set Start Month: April, End Month: March

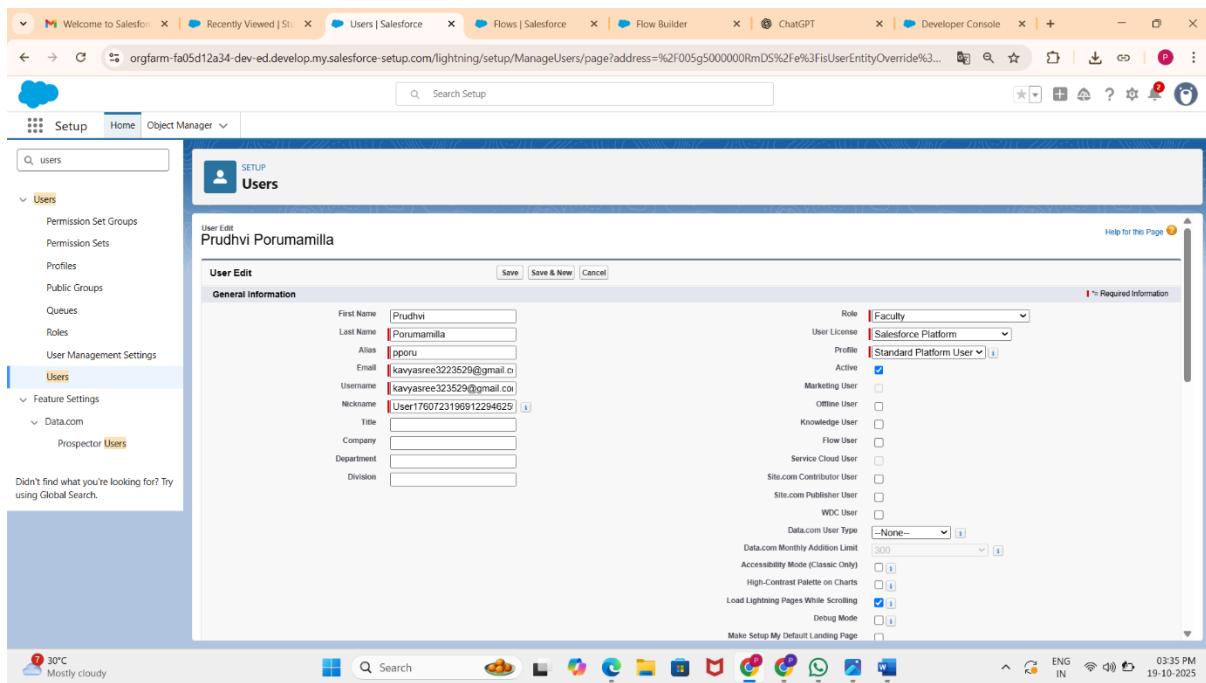
- Matches the Indian government's financial reporting cycle.



## 5. User Setup & Licenses

- Go to **Setup** → **Users** → **New User**
- Create accounts for:
  - **College Management**
  - **Principal**
  - **Hod**
  - **Faculty**
- Assign appropriate **Salesforce licenses** and **roles** to each user.

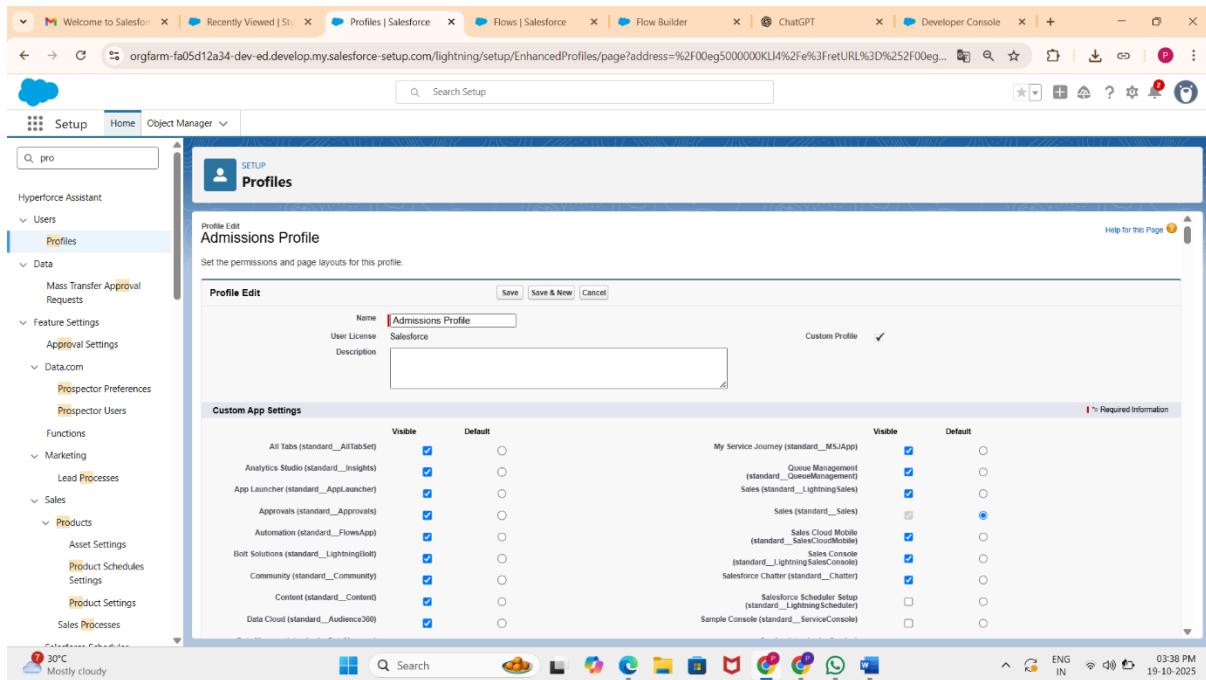
User	Role	Recommended Profile	License
<b>Principal</b>	Principal	Custom Principal Profile	Salesforce Platform
<b>HOD</b>	Hod	Custom HOD Profile	Salesforce Platform
<b>Faculty</b>	Faculty	Faculty Profile	Salesforce Platform



## 6. Profiles

- Navigate to **Setup → Profiles**
- Clone the **Standard User** profile and rename it.
- Create profiles such as:
  - Management Profile
  - HOD Profile
  - Faculty Profile
- Assign **object-level** and **field-level permissions** according to user responsibilities.

Profile	Access Level	Permissions
<b>Management Profile</b>	Full	View, Create, Edit, Delete, Modify All
<b>HOD Profile</b>	Moderate	Manage students/faculty in dept
<b>Faculty Profile</b>	Limited	Mark attendance, view course/student info



## 7. Roles

- Go to **Setup → Roles → Set Up Roles**
- Define a data visibility hierarchy:

System Admin

→ College Management

→ Principal

→ Hod

→ Faculty

- Ensures data flows securely upward through the organization.

Role	Reports To	Data Visibility	Example User
<b>Management</b>	System Admin	Can see all under Principal	Management
<b>Principal</b>	Management	Can see all under HOD	Principal
<b>HOD</b>	Principal	Can see all under Faculty	HOD

**Role Detail**

	Label	College Management	Role Name	College_Management
This role reports to	None		Role Name as displayed on reports	
Modified By	PORUMAMILLA KAYA SREE	10/17/2025, 9:16 AM	Sharing Groups	Role, Role and Internal Subordinates
Opportunity Access	Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities			
Case Access	Users in this role can edit all cases associated with accounts that they own, regardless of who owns the cases			

**Users in College Management Role**

Action	Full Name	Alias	Username	Active
Edit	Jani Mehra	jmehra	jmehra@leamsmart.com	<input type="checkbox"/>
Edit	Kavya Sree	ksree	kavyasree322352@gmail.com	<input checked="" type="checkbox"/>

**Creating the Role Hierarchy**

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**

- Collapse All Expand All
- LearnSmart CRM
  - + Add Role
  - + Add Role
  - + CEO Edit | Del | Assign
    - + Add Role
  - + COO 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
  - + College Management Edit | Del | Assign
    - + Add Role
    - + Principal Edit | Del | Assign
      - + Add Role
    - + Admissions Officer Edit | Del | Assign
      - + Add Role
    - + Department Head Edit | Del | Assign
      - + Add Role

## 8. Permission Sets

- Navigate to **Setup → Permission Sets → New**
- Create additional privileges such as:
  - Opportunity Access
  - Data Export Access

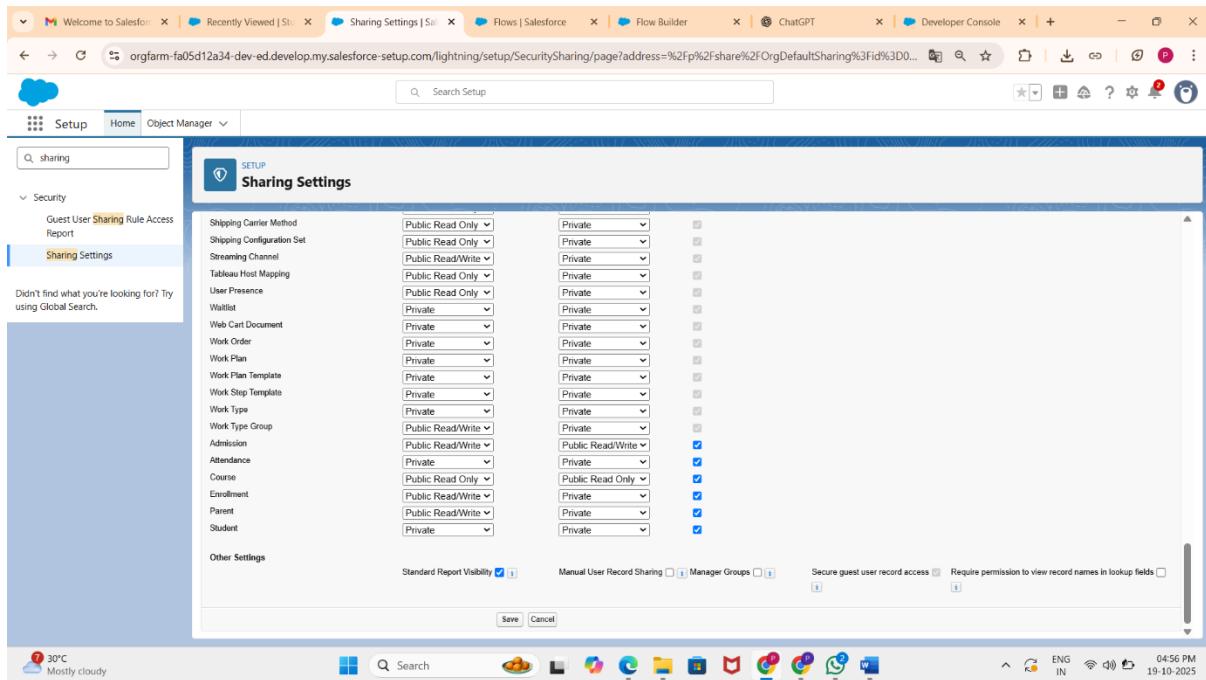
- Assign permission sets to users who require extra access without changing their profiles.

The screenshot shows the Salesforce Setup interface with the 'Permission Sets' page open. The left sidebar shows navigation options like 'Setup', 'Home', and 'Object Manager'. The main content area displays a permission set named 'Grade Manager'. It includes fields for 'API Name' (Grade\_Manager), 'Description', 'License', 'Session Activation Required', and 'Permission Set Groups Added To'. Below this, the 'Apps' section lists various app permissions such as 'Assigned Apps', 'Assigned Connected Apps', 'Object Settings', 'App Permissions', 'Apex Class Access', 'Visualforce Page Access', 'External Data Source Access', and 'Flow Access'. The bottom of the screen shows the Windows taskbar with icons for search, file explorer, and other applications, along with system status indicators like battery level and network connection.

## 9. Organization-Wide Defaults (OWD)

- Go to **Setup → Sharing Settings → Edit**
- Configure default record access:
  - Management: **Private**
  - Principal: **Private**
  - Hod: **Private**
  - Student: **Controlled by Parent**
- Ensures data security and proper sharing control.

Object	Default Internal Access	Purpose
<b>Management</b>	Private	Restrict access to Management data
<b>Principal</b>	Private	Restrict access to Principal data
<b>HOD</b>	Private	Restrict access to HOD data
<b>Student</b>	Controlled by Parent	Student inherits access from parent



## 10. Sharing Rules

- Navigate to **Setup** → **Sharing Settings** → **New Rule**
- Create sharing rules for collaboration:
  - Share **Management** records with **Principal** (Read Only)
  - Share **Principal** with **Faculty** (Read/Write)
- Allows teamwork without compromising data security.

### Rule 1: Share Management records with Principal (Read Only)

- Object: **Management**
- Rule Type: **Based on record owner**
- Select “Records owned by users in” → choose **Management Role**.
- Share with → select **Principal Role**.
- Access Level → **Read Only**.
- Click **Save** and **Recalculate** if prompted.

### Rule 2: Share Principal records with Faculty (Read/Write)

- Scroll to the **Principal** object section.
- Click **New** under **Sharing Rules**.
- Rule Type: **Based on record owner**.
- Select “Records owned by users in” → choose **Principal Role**.
- Share with → select **Faculty Role**.
- Access Level → **Read/Write**.

- Click Save and Recalculate.

Object	Access Type	Setting
Waitlist	Private	
Web Cart Document	Private	
Work Order	Private	
Work Plan	Private	
Work Plan Template	Private	
Work Step Template	Private	
Work Type	Private	
Work Type Group	Public ReadWrite	
Admission	Public ReadWrite	
Attendance	Private	
Course	Public Read Only	
Enrollment	Public ReadWrite	
Parent	Public ReadWrite	
Student	Private	

**Other Settings**

- Manager Groups:
- Secure guest user record access:
- Require permission to view record names in lookup fields:

Sharing Rules | Lead Sharing Rules | New | Recalculate | Lead Sharing Rules Help ?

## 11. Login Access Policies

- Go to Setup → Login Access Policies
- Enable Grant Login Access to Administrators
- Helps administrators troubleshoot login or data issues while maintaining transparency.

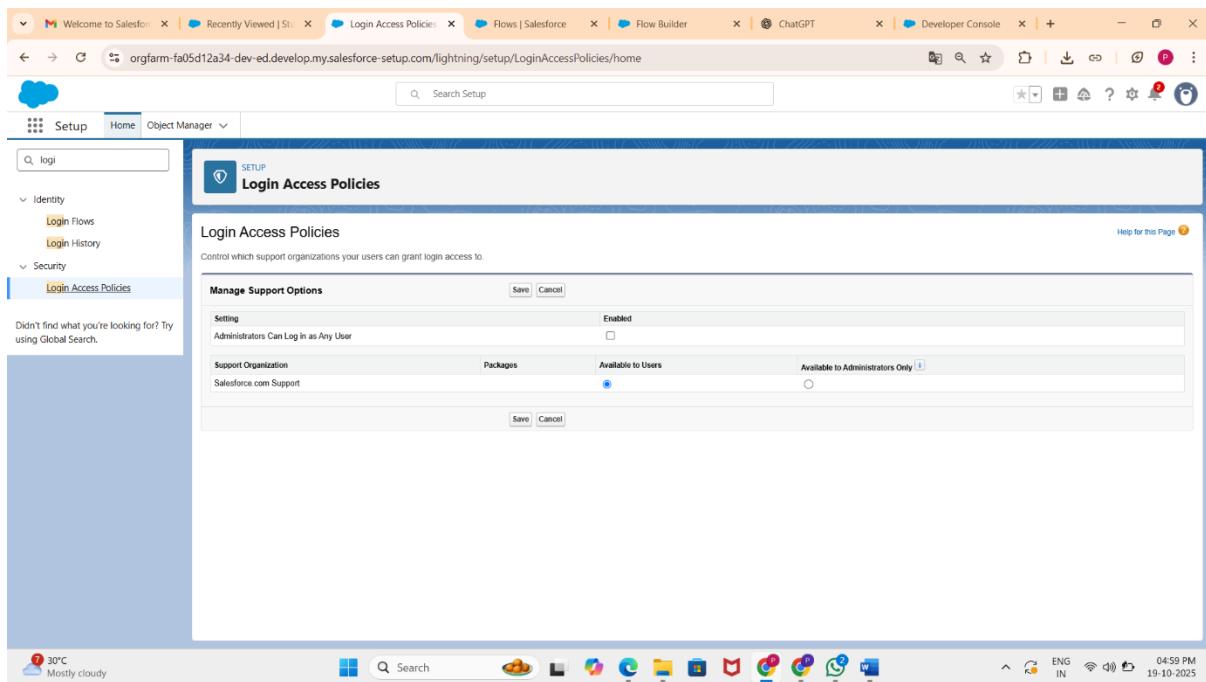
**Administrators can temporarily log in as a user to troubleshoot:**

- Data visibility issues
- Permission or record access problems
- Login or profile-related errors
- All such logins are **logged by Salesforce** to ensure **accountability and transparency**.

Grant Login Access to Administrators is enabled, allowing secure troubleshooting by system administrators without compromising user data privacy.

### ✓ Outcome:

- Administrators can log in as users for troubleshooting.
- All login activities are tracked for transparency and security.



## 12. Developer Org Setup

- Sign up for a free **Salesforce Developer Org** at [developer.salesforce.com](https://developer.salesforce.com)
- Navigate to **Setup → App Manager → New Lightning App**
- Create your app named **LearnSmart CRM**
- This serves as the main environment for customization and testing.

### ✓ Outcome:

- A **LearnSmart CRM app** is now created in your Developer Org.
- The environment is ready for:
  - Custom objects
  - Workflows and Flows
  - Sharing rules and permissions
  - Testing automation and reports
  - Safe Testing Environment
  - Customization Flexibility
  - User Role Simulation
  - Automation & Workflow Testing
  - Data Integration Practice

Welcome to Salesforce | PORUMAMILLA KAVYA SREE | Login Access Policies | Flows | Salesforce | Flow Builder | ChatGPT | Developer Console

orgfarm-fa05d12a34-dev-ed.lightning.force.com/lightning/r/Student\_c/a00g5000001620AAQ/view

LearnSmart CRM Students Courses Attendances Admissions Reports Enrollments

Search... ...

Student PORUMAMILLA KAVYA SREE

New Contact Edit New Opportunity

Related Details

Student Name: PORUMAMILLA KAVYA SREE

Owner: PORUMAMILLA KAVYA SREE

Student Admission Number: 3,939

Date of Birth: 10/3/2007

Grade: XII

Gender: Female

Phone: (798) 982-8257

Email: kavayasree3223529@gmail.com

Student Status: Applied

Created By: PORUMAMILLA KAVYA SREE, 10/19/2025, 4:31 AM

Last Modified By: PORUMAMILLA KAVYA SREE, 10/19/2025, 4:31 AM

Activity

Filters: All time • All activities • All types ...

Refresh • Expand All • View All

Upcoming & Overdue

No activities to show.

Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

30°C Mostly cloudy

Search

Windows Taskbar icons

ENG IN 05:01 PM 19-10-2025

# Phase 3: Data Modeling & Relationships – Instructions

## Objective:

To design and implement the data structure for LearnSmart CRM by creating objects, fields, record types, and relationships that enable efficient data management and reporting.

### 1 Standard & Custom Objects

Objects store records in Salesforce. Standard objects come pre-built (e.g., Account, Contact, Lead), while custom objects are project-specific (e.g., Student, Course, Enrollment).

#### Step-by-Step:

1. Go to **Setup → Object Manager**.
2. To create a Custom Object:
  - o Click Create → Custom Object.
  - o Enter Label & Plural Label (e.g., Student / Students).
  - o Enter Object Name (API Name).
  - o Choose Record Name (Text or Auto Number).
  - o Enable optional features: Allow Reports, Activities, Track Field History.
3. Save the object.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. Under the 'Student' object, the 'Fields & Relationships' tab is active. A table lists the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Date of Birth	Date_of_Birth_c	Date		
Email	Email_c	Email (Unique)		
Gender	Gender_c	Picklist		
Grade	Grade_c	Picklist		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone	Phone_c	Phone		
Student Admission Number	Student_Admission_Number_c	Number(10, 0)		
Student Name	Name	Text(80)		
Student Status	Student_Status_c	Picklist		

### 2 Fields

Fields store specific data within an object.

## Step-by-Step:

1. Go to Object Manager → [Object] → Fields & Relationships → New.
2. Select Field Type: Text, Number, Date, Picklist, Checkbox, Lookup, Formula, etc.
3. Enter Field Details:
  - Label
  - Length / Values
  - Required or Optional
4. Set Field-Level Security for profiles.
5. Add the field to Page Layouts.

## Example (Student Object):

- First Name → Text
- Last Name → Text
- Email → Email
- Enrollment Date → Date
- Status → Picklist (Active, Inactive, Graduated)

The screenshot shows the Salesforce Setup interface for creating a new field. The URL in the browser is <https://orgfarm-fa05d12a34-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01lg500000AFYr/FieldsAndRelationships/Name/view>. The main content area displays the 'Student Name' field details, including its label, data type, and validation rules. The sidebar on the left lists various configuration options for the 'Student' object.

### **3 Record Types**

Record Types allow different business processes for the same object (e.g., Individual vs Corporate Student).

#### **Step-by-Step:**

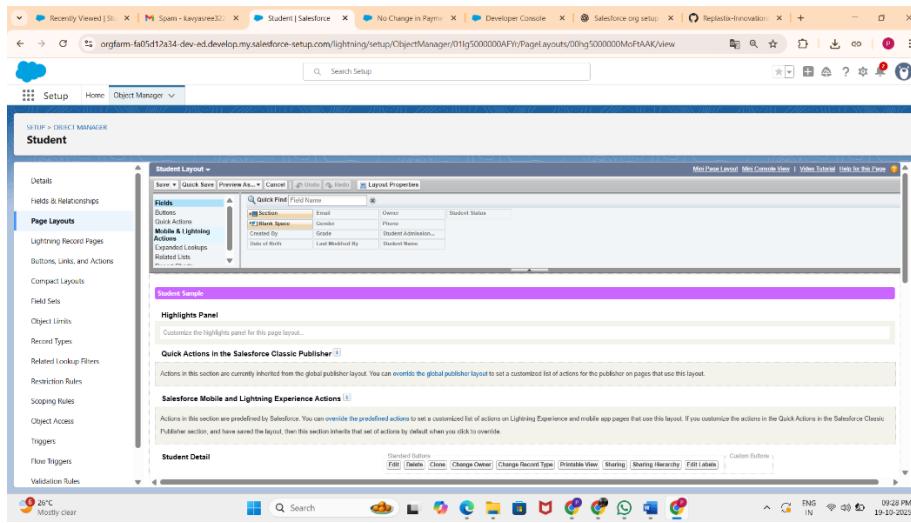
1. Go to Object Manager → [Object] → Record Types → New.
2. Enter Record Type Label & Active Status.
3. Assign to Profiles.
4. Select which Page Layout to use for each record type.
5. Save.

### **4 Page Layouts**

Page Layouts control the arrangement of fields, sections, and related lists on a record page.

#### **Step-by-Step:**

1. Go to Object Manager → Student
2. Page Layouts → New/Edit.
3. Drag and drop:
  - Email
  - Gender
  - Grade
  - Phone
  - Student Admission number
  - Student Name
  - Student Status
4. Assign layout to profiles or record types.
5. Save.



## 5 Compact Layouts

Display key fields in mobile or highlights panel for quick access.

### Step-by-Step:

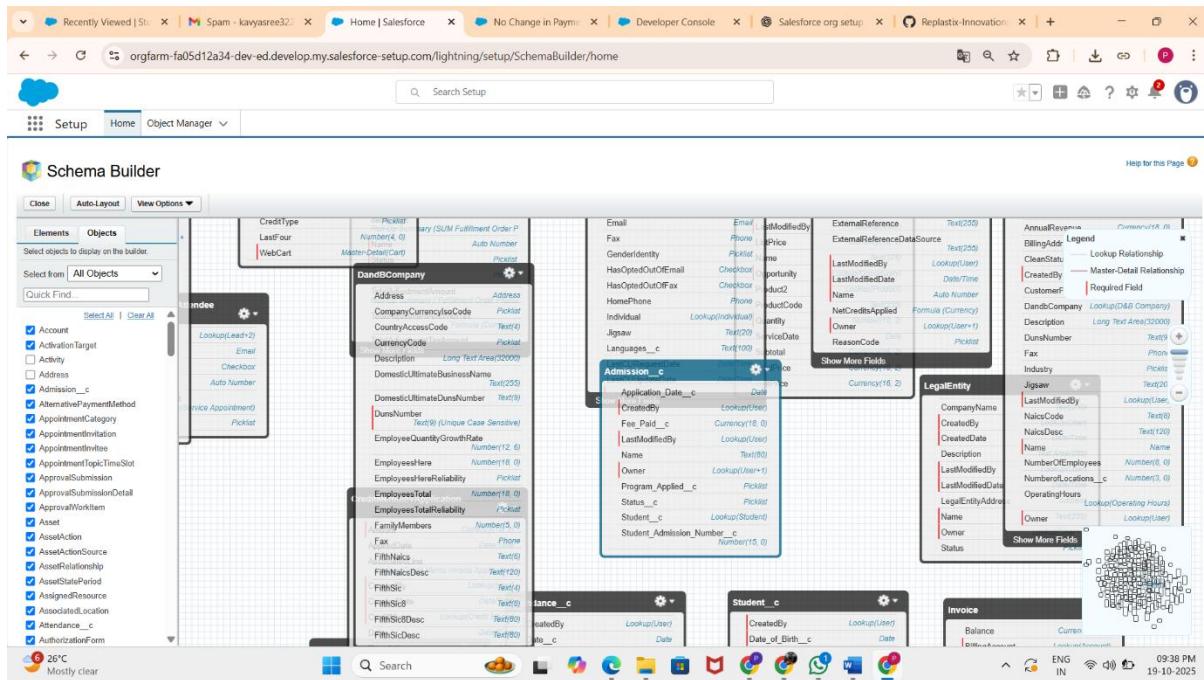
1. Go to Object Manager → [Object] → Compact Layouts → New.
2. Select the fields to display.
3. Set as Primary Compact Layout.
4. Save.

## 6 Schema Builder

Visual tool to design and understand objects and relationships.

### Step-by-Step:

1. Go to Setup → Schema Builder.
2. Add objects to the canvas.
3. Drag to view relationships between objects.
  - Admission Number
  - Attendance
  - Student
4. Use this for planning and validating your data model.



## 7 Relationships

Relationships link objects to reflect real-world associations.

## Step-by-Step:

1. Go to Object Manager → [Object] → Fields & Relationships → New.
  2. Choose relationship type:
    - Lookup: Loose connection; child can exist independently.
    - Master-Detail: Strong parent-child; supports roll-up summary fields.
    - Hierarchical: Only for User object (manager-subordinate).
  3. Define Related Object and settings.
  4. Save.

## Example:

- Student → Course (Lookup)
  - Enrollment → Student & Course (Master-Detail)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course Fee	Course_Fee_c	Currency(16, 2)		
Created By	CreatedById	Lookup(User)		
Enrollment Date	Enrollment_Date_c	Date		
Enrollment Name	Name	Text(80)		
Enrollment Status	Enrollment_Status_c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Payment Status	Payment_Status_c	Picklist		
Student	Student_c	Lookup(Student)		

## 8 Junction Objects

Enable many-to-many relationships between two objects.

### Step-by-Step:

1. Create a custom object (e.g., Enrollment).
2. Add two Master-Detail relationships linking the related objects (Student, Course).
3. Configure Page Layouts and Related Lists.
4. Save.

## 9 External Objects

Allow Salesforce to access data stored outside the platform without copying it into Salesforce.

### Step-by-Step:

1. Go to Setup → External Data Sources → New.
2. Choose Data Source Type (e.g., OData).
3. Configure Authentication.

4. Create External Objects that map to external tables.

5. Save and validate access.

The screenshot shows the Salesforce Object Manager interface for the 'Enrollment' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main content area is titled 'Fields & Relationships' and displays a table of fields. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course Fee	Course_Fee__c	Currency(16, 2)		
Created By	CreatedById	Lookup(User)		
Enrollment Date	Enrollment_Date__c	Date		
Enrollment Name	Name	Text(80)		
Enrollment Status	Enrollment_Status__c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Status	Payment_Status__c	Picklist		
Student	Student__c	Lookup(Student)		✓

## Phase 4: Process Automation (Admin) — Learn Smart CRM Project

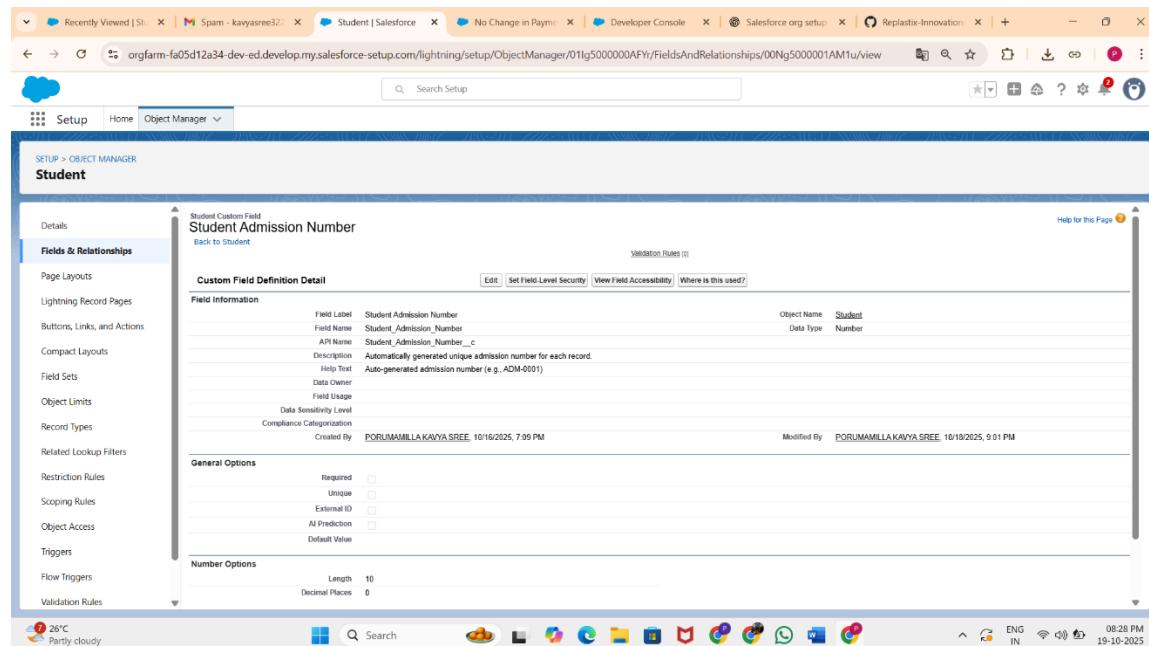
Process Automation in Salesforce helps streamline business operations, enforce business rules, and save time by automating repetitive tasks.

### 1 Validation Rules

- Validation Rules ensure **data integrity** by enforcing conditions before a record is saved.
- Example: Prevent a Student from being enrolled without an Admission Number.

#### Step-by-Step

1. Setup → Object Manager → Select object (e.g., **Student**).
2. Click **Validation Rules** → New.
3. Enter **Rule Name** (e.g., `AdmissionNumber_Required`).
4. Enter **Error Condition Formula**:
5. `ISBLANK(Admission_Number__c)`
6. Enter **Error Message**: "Admission Number cannot be blank."
7. Select **Error Location** (Field or Top of Page).
8. Click **Save**.



## 2 Workflow Rules (Classic Automation)

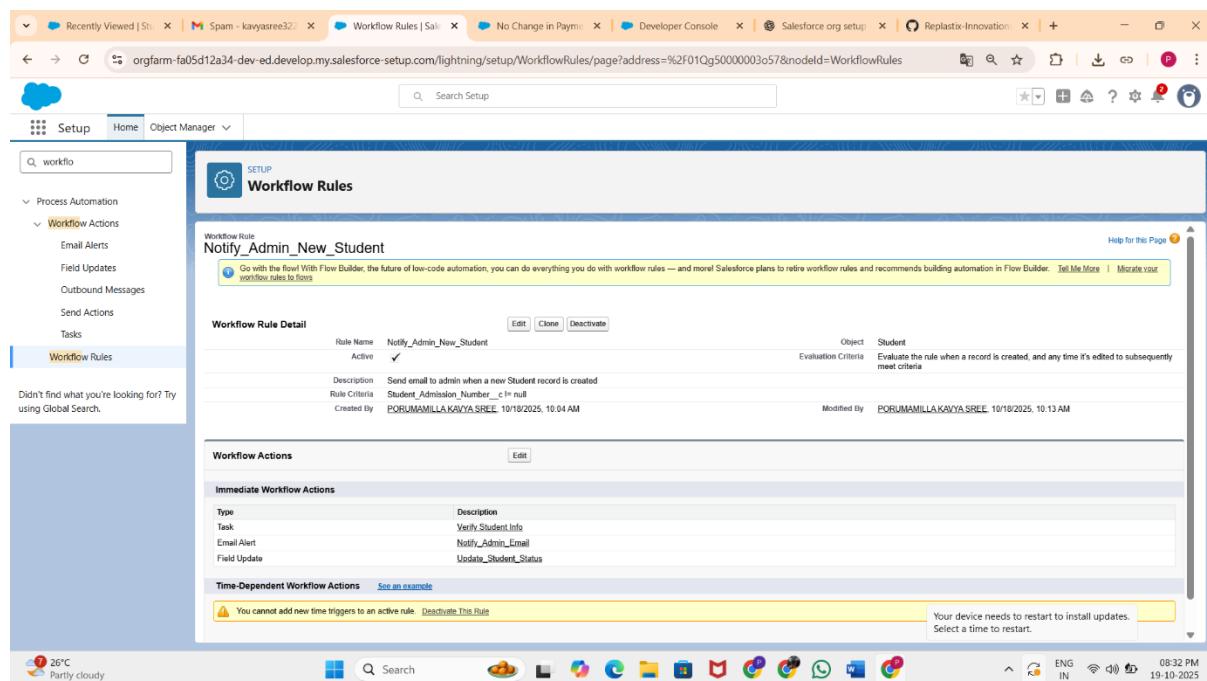
- Automates **actions** like emails, field updates, tasks when conditions are met.
- Example: Send an email to admin when a new Student record is created.

### Step-by-Step

1. Setup → Workflow Rules → **New Rule**.
2. Select Object (e.g., Student) → **Next**.
3. Enter **Rule Name** → Set **Evaluation Criteria** (Created, Created & Edited).
4. Enter **Rule Criteria** (e.g., Admission\_Number\_\_c != null).
5. Click **Save & Next**.

### Add Workflow Actions

- **Email Alert:** Notify admin.
  - **Field Update:** Update a field automatically.
  - **Task:** Assign a task to a user.
6. Activate the workflow.

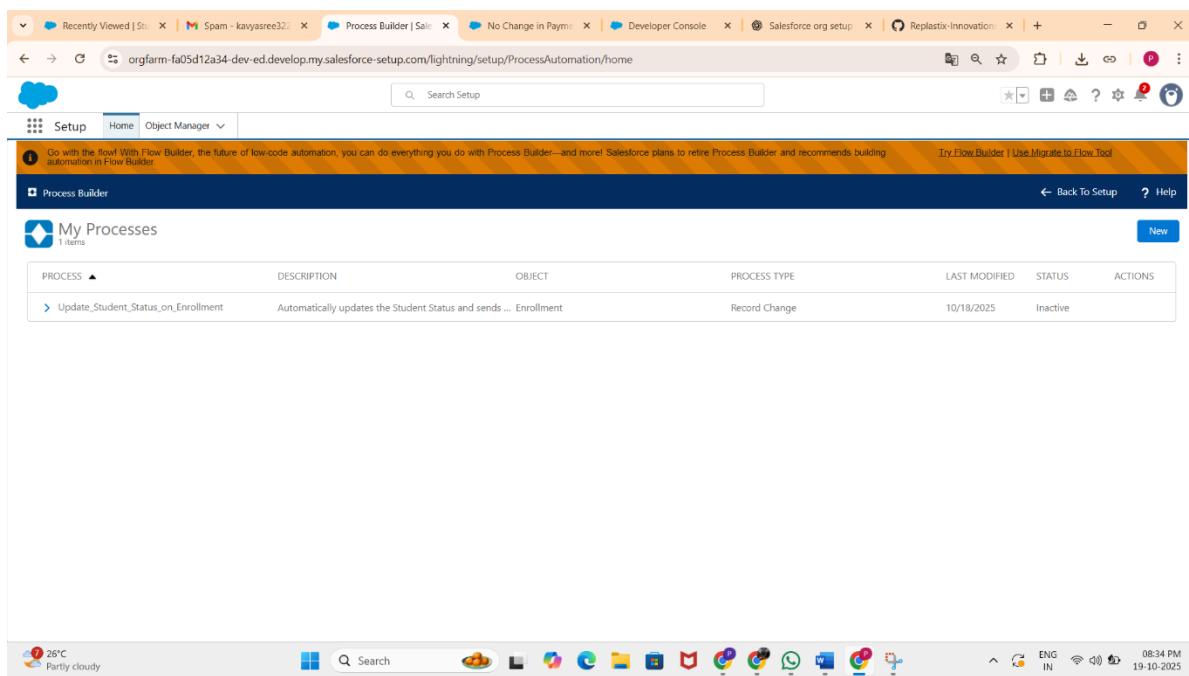


### 3 Process Builder (Point-and-Click Automation)

- More advanced than Workflow Rules. Allows multiple **if-then actions**.
- Example: Automatically update Student status when Enrollment is created.

#### Step-by-Step

1. Setup → Process Builder → New → Enter Name & Description → Save.
2. Add **Object** (e.g., Enrollment) → Start the process **when a record is created or edited**.
3. Define **Criteria** (e.g., Enrollment\_Status\_\_c = "Completed").
4. Add **Immediate Actions**:
  - Field Update: Update Student Status to “Enrolled”.
  - Email Alert: Notify Student or Admin.
5. Click **Activate**.



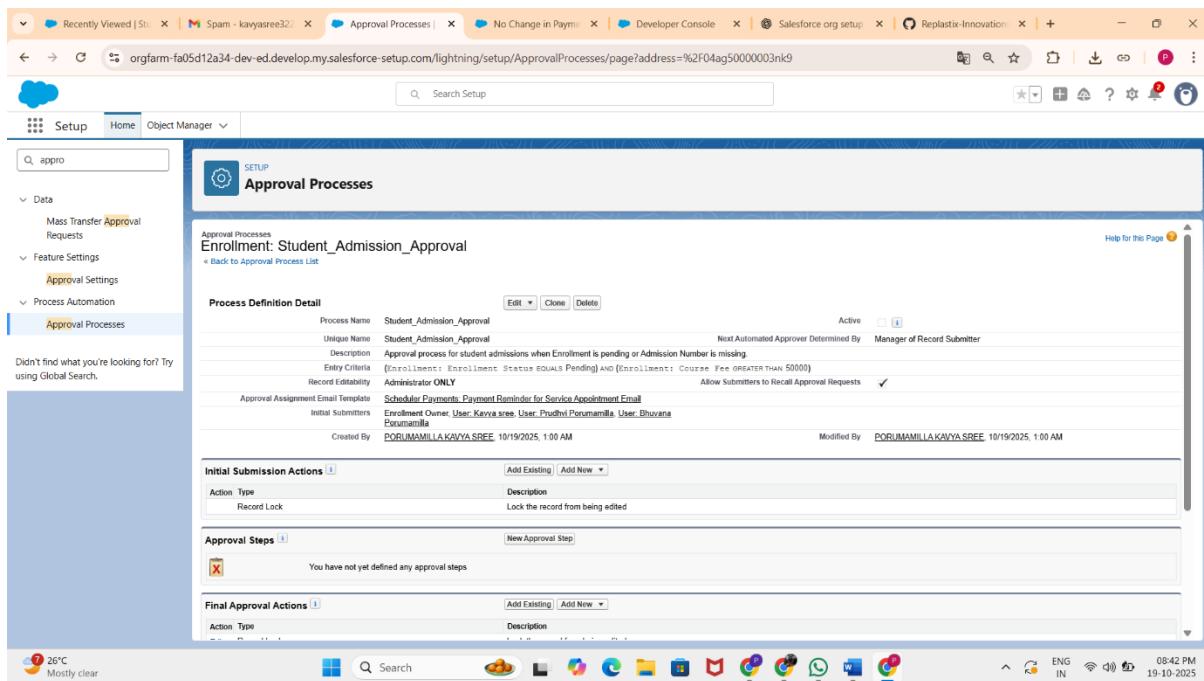
### 4 Approval Process

#### What & Why

- Automates record approvals based on conditions.
- Example: Approval required when Student Admission Number is missing or special approval for high-value courses.

## Step-by-Step

1. Setup → Approval Processes → **Create New Approval Process** → Use Standard Setup Wizard.
2. Select Object (e.g., Student).
3. Define:
  - Name of approval process
  - Entry criteria (e.g., Enrollment\_Status = “Pending”)
4. Assign **Approver** (User or Queue).
5. Define **Approval Actions** (Field Updates, Email Alerts) and **Rejection Actions**.
6. Activate process.



## 5 Flow Builder

- Powerful Salesforce automation tool. Handles **complex multi-step automations**.

### Types of Flows

1. **Screen Flow** → Interactive user flow (e.g., Student registration wizard).
2. **Record-Triggered Flow** → Triggered when a record is created/updated.
3. **Scheduled Flow** → Run at a specific date/time (e.g., daily updates).

4. **Auto-launched Flow** → Triggered by other flows, processes, or Apex.

### Step-by-Step for Record-Triggered Flow

1. Setup → Flows → **New Flow** → Select **Record-Triggered Flow** → Create.
2. Select Object (e.g., Student) → Trigger **on Create or Update**.
3. Define **Conditions** (e.g., Admission\_Number\_\_c is not null).
4. Add **Actions**:
  - Update Records: Set Status = “Enrolled”.
  - Send Email Alert.
  - Create Task for admin.
5. Connect elements → **Save** → **Activate**.

## 6 Email Alerts

- Sends email notifications automatically based on Workflow, Process Builder, or Flow.

### Step-by-Step

1. Setup → Email Alerts → **New Email Alert**.
2. Select Object (e.g., Student).
3. Enter **Name, Description**.
4. Choose **Email Template** (create if needed).
5. Select **Recipients** (User, Role, or Field).
6. Save → Use in Workflow/Process/Flow.

## 7 Field Updates

- Automatically updates field values based on conditions.

### Step-by-Step

1. Workflow → Field Update → **New**.
2. Enter Name → Choose Field (e.g., Student Status).
3. Set **New Value** (e.g., “Enrolled”).
4. Save and activate.

## Tasks

- Automatically assigns tasks to users. Example: Admin to verify Student info.

### Step-by-Step

1. Workflow/Process/Flow → **New Task Action**.
2. Enter **Assigned To, Subject, Due Date, Priority**.
3. Save → Activate.

## Custom Notifications

- Sends in-app notifications to users. Works in Lightning Experience and Mobile App.

### Step-by-Step

1. Setup → **Notification Builder** → **Custom Notifications** → New.
2. Enter **Name, API Name, Supported Channels** (Desktop, Mobile).
3. Save → Use in Flow, Process, or Workflow to notify users when needed

## Phase 5: Apex Programming (Developer)

Apex is Salesforce's object-oriented programming language that allows developers to write complex business logic and automate processes that go beyond declarative tools (like Flows or Process Builder). It runs on the Salesforce platform and helps implement customized, scalable, and efficient solutions.

### 1 Apex Classes & Utility Classes

#### Purpose:

- Create reusable code to handle business logic.
- Automate notifications, calculations, or custom logic.

#### Use Cases:

- Create a utility class to calculate recycled product cost.
- Build classes for automated notifications or custom business logic.

### 2 Apex Triggers

#### Purpose:

- Automatically execute Apex code when records are created, updated, deleted, or undeleted.

#### Trigger Types:

Type	Purpose	Example Use Case
Before Insert/Update	Validate or modify records before saving	Ensure total waste collected is correct before insert
After Insert/Update/Delete	Execute logic after records are saved	Notify recycling center when order status changes

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for Recently Viewed, Spam, Student, No Change in Payment, Developer Console, SalesForce org setup, Replaxt-Innovation, and others. The main area displays two Apex classes: `EnrollmentPaymentHandler.apxc` and `EnrollmentPaymentTrigger.apxc`. The `EnrollmentPaymentHandler.apxc` class contains methods for handling insertions and updates of enrollment records, setting status messages based on payment status. The `EnrollmentPaymentTrigger.apxc` class contains triggers for the `Enrollment` object. Below the code editor is a progress bar showing deployment status for various runs, with a summary table at the bottom indicating deployment times and errors.

```

1  public class EnrollmentPaymentHandler {
2
3    public static void handleBeforeInsert(List<Enrollment__c> newEnrollments){
4      for(Enrollment__c e : newEnrollments){
5        if(e.Payment_Status__c == 'Paid'){
6          e.Status_Message__c = 'Your payment is successfully received!';
7        } else if(e.Payment_Status__c == 'Pending'){
8          e.Status_Message__c = 'Your payment is pending. Please complete it soon.';
9        } else if(e.Payment_Status__c == 'Failed'){
10          e.Status_Message__c = 'Your payment has failed. Please try again.';
11        } else if(e.Payment_Status__c == 'Refunded'){
12          e.Status_Message__c = 'Your payment has been refunded successfully.';
13        }
14      }
15    }
16
17    public static void handleBeforeUpdate(List<Enrollment__c> updatedEnrollments, Map<Id, Enrollment__c> oldMap){
18      for(Enrollment__c e : updatedEnrollments){
19        Enrollment__c oldE = oldMap.get(e.getId());
20
21      }
22    }
23  }
24
25  public trigger EnrollmentPaymentTrigger on Enrollment__c {
26    after insert {
27      EnrollmentPaymentHandler.handleBeforeInsert(newList);
28    }
29    after update {
30      EnrollmentPaymentHandler.handleBeforeUpdate(updatedList, map);
31    }
32  }

```

ReqId	Nice	Order	Description	Status	Start	End	Duration (m)	Handler Err	Ajax Err	Delay
12	0	10	Getting members of ApexClassMember for containerId=1dgc5000000ME17AAG	Finished	8:13:54	8:13:54	400			
11	-2	9	Getting deployment for id=1dgc500000002NKA2	Finished	8:13:53	8:13:54	301			
10	-1	8	Creating deployment for containerId=1dgc5000000ME17AAG Save=false runTests=false	Finished	8:13:52	8:13:52	260			
9	-1	8	Creating or Updating containerMember for containerId=1dgc5000000ME17AAG	Finished	8:13:51	8:13:52	235			
8	-1	7	Creating deployment for containerId=1dgc5000000ME17AAG Save=false runTests=false	Group Failed	8:13:40	8:13:40	0			
7	-1	7	Creating or Updating containerMember for containerId=1dgc5000000ME17AAG	Failed	8:13:39	8:13:40	213			

### 3 Trigger Design Pattern

#### Purpose:

- Maintain clean, reusable, and scalable trigger logic.

#### Best Practices:

- Only one trigger per object.
- Delegate logic to handler classes.
- Ensure triggers are bulkified (handle multiple records efficiently).

#### Use Case:

- Maintain organized logic for Plastic Waste automation through a central handler class.

### 4 SOQL & SOSL

#### Purpose:

- SOQL: Fetch data from one or more Salesforce objects.
- SOSL: Search data across multiple objects and fields.

#### Use Cases:

- Retrieve all orders placed by a specific customer (SOQL).
- Search for a keyword in multiple objects (SOSL).

## 5 Collections

## Purpose:

- Store and manage multiple data values efficiently.

Collection Description	Example Use Case
List	Ordered collection of elements List of Orders or Waste records
Set	Unordered, unique elements Unique Waste Types
Map	Key-value pairs Map of Customer ID → Order List

## 6 Control Statements

## Purpose:

- Execute code conditionally or repeatedly.

## **Common Statements:**

- Conditional: if, else, switch
  - Loops: for, while, do-while
  - Others: break, continue

## 7 Asynchronous Processing

## Purpose:

- Perform background tasks that don't block the main execution thread.

## Types & Use Cases:

Type	Purpose	Example Use Case
Batch Apex	Process large datasets asynchronously	Update thousands of recycled product records nightly
Queueable Apex	Run async operations with more complex logic	Send confirmation emails after order approval
Scheduled Apex	Run Apex code at defined times	Generate weekly recycled product reports

Type	Purpose	Example Use Case
Future Methods	Execute code asynchronously, useful for callouts	Notify customers about shipment updates

## 8 Exception Handling

### Purpose:

- Catch and handle errors gracefully to prevent transaction failures.

### Use Cases:

- Handle record insert/update exceptions.
- Log errors for admin review.

## 9 Test Classes

### Purpose:

- Validate Apex code and ensure proper functionality before deployment.
- Salesforce requires at least 75% code coverage for production deployment.

### Use Cases:

- Test triggers, classes, and async logic.
- Simulate user actions and validate outcomes.

# Phase 6: User Interface Development

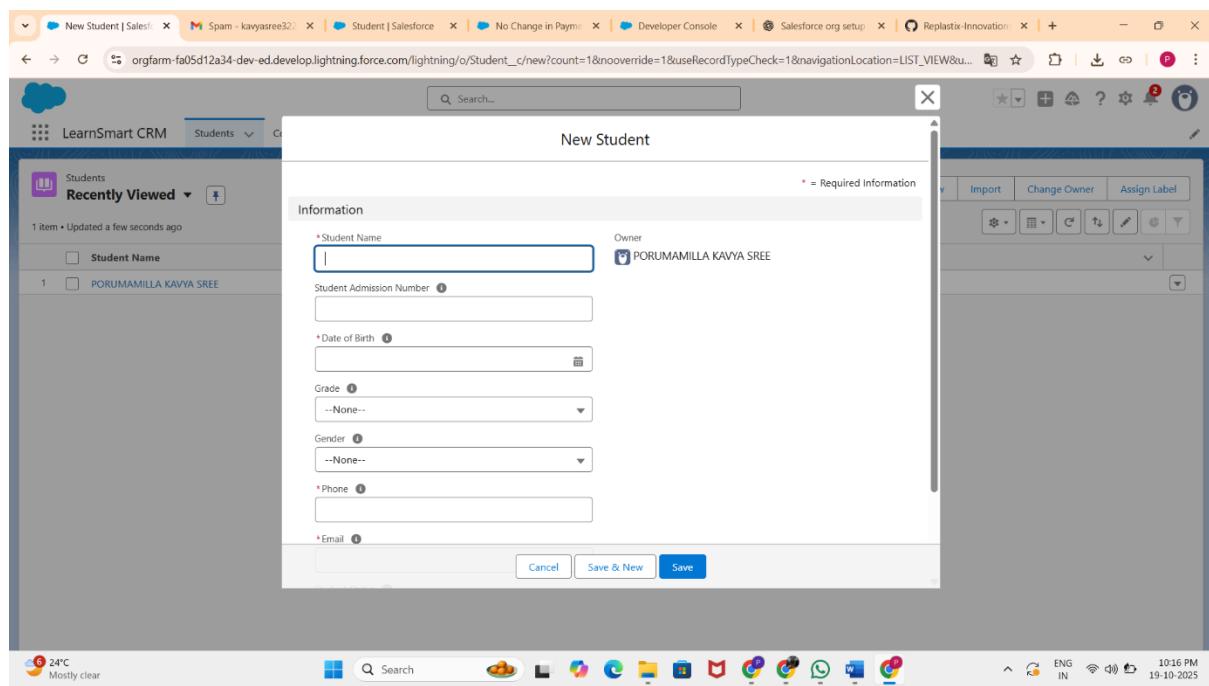
The goal of this phase is to create an intuitive and functional UI for LearnSmart CRM using Salesforce Lightning features.

## 1. Lightning App Builder

**Objective:** Create custom Lightning apps and pages tailored for LearnSmart CRM.

**Steps:**

1. Go to **Setup** → **App Builder** → **Lightning App Builder**.
2. Click **New** → select **App Page**, **Record Page**, or **Home Page** depending on the requirement.
3. Name the app/page as per your module (e.g., "Student Management Dashboard").
4. Drag and drop components (standard or custom) onto the page layout.
5. Save and activate the page for the required **user profiles**.



## 2. Record Pages

**Objective:** Customize the layout for individual records (Accounts, Leads, Contacts, or custom objects).

**Steps:**

1. Go to **Setup** → **Object Manager** → [Object Name] → **Lightning Record Pages**.

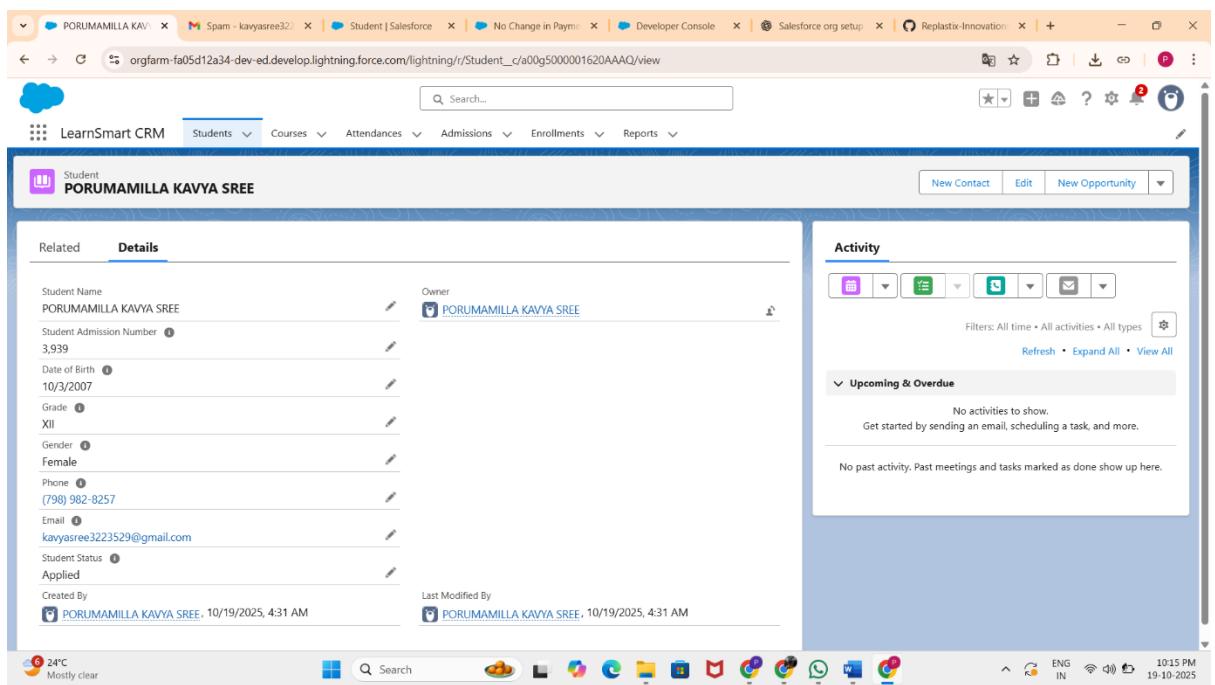
2. Click **New** → **Record Page**.
3. Add relevant sections: details, related lists, custom LWC components.
4. Assign this page to **specific apps, profiles, or as org default**.

### 3. Tabs

**Objective:** Create tabs to organize objects, Visualforce pages, or Lightning Components in your app.

**Steps:**

1. Setup → Tabs → Click **New**.
2. Choose type: **Custom Object Tab**, **Lightning Page Tab**, **Web Tab**.
3. Assign a **tab label and icon**.
4. Add tab to your app via **App Manager** → **Edit Lightning App** → **Navigation Items**.



### 4. Home Page Layouts

**Objective:** Customize home pages for users to provide dashboards, reports, or key components.

**Steps:**

1. Go to **Setup** → **Lightning App Builder** → **Home Page**.

2. Create a **New Home Page** and name it (e.g., “Sales Team Dashboard”).
3. Drag components such as **Reports, Lists, or Custom LWC components**.
4. Activate the home page for specific **profiles or the org default**.

## 5. Utility Bar

**Objective:** Provide easy access to frequently used tools.

**Steps:**

1. Go to **Setup → App Manager → Edit App → Utility Bar**.
2. Click **Add Utility Item** (e.g., **Notes, Recent Records, Omni-Channel**).
3. Set **visibility and labels** for each item.
4. Save and activate the app.

## 6. Lightning Web Components (LWC)

**Objective:** Create reusable, dynamic UI components for LearnSmart CRM.

**Instructions:**

1. Set up VS Code with Salesforce Extension Pack.
2. Create a new Lightning Web Component (LWC) for each module, e.g., Students, Courses, Enrollments.
3. Design the component layout using HTML and CSS for UI elements.
4. Implement component logic in JavaScript.
5. Deploy the LWC to Salesforce.
6. Add the deployed LWC to Lightning pages via Lightning App Builder.

The screenshot shows the LearnSmart CRM interface. On the left, there's a sidebar with navigation links: Students, Courses, Attendances, Admissions, Enrollments, and Reports. The main content area displays a 'Course' record for 'AIML'. The 'Details' tab is selected, showing fields like Course Name (AIML), Course Title (Mathematics), Credits (2), Department (CSE), and Description (In this specialization mathematics can have 2 credits). It also shows the 'Created By' field (PORUMAMILLA KAVYA SREE) and the 'Last Modified By' field (PORUMAMILLA KAVYA SREE, 10/19/2025, 1:41 AM). To the right, there's an 'Activity' section with a search bar and filter options (All time, All activities, All types). It shows a message: 'No activities to show. Get started by sending an email, scheduling a task, and more.' The system status bar at the bottom indicates it's 24°C, mostly clear, and shows the date and time as 10:19 PM on 19-10-2025.

This screenshot shows the LearnSmart CRM interface again, but this time it's displaying an 'Attendance' record for 'Regular'. The 'Details' tab is selected, showing fields like Attendance Name (Regular), Student (PORUMAMILLA KAVYA SREE), Date (9/27/2025), Status (Present), Remarks (GOOD), and User (Prudhvi Porumamilla). The 'Last Modified By' field is shown as PORUMAMILLA KAVYA SREE, 10/19/2025, 7:24 AM. The right side of the screen shows the same 'Activity' section as the previous screenshot, indicating no activities. The system status bar at the bottom remains the same, showing 24°C, mostly clear, and the date and time as 10:20 PM on 19-10-2025.

LearnSmart CRM

Admission 229Y1A3939

**Related Details**

- Student Admission Name: 229Y1A3939
- Owner: PORUMAMILLA KAVYA SREE
- Application Date: 6/6/2025
- Program Applied: CSE
- Status: Applied
- Fee Paid: ₹50,000
- Student Admission Number: 3,939
- Created By: PORUMAMILLA KAVYA SREE, 10/19/2025, 7:25 AM

Last Modified By: PORUMAMILLA KAVYA SREE, 10/19/2025, 7:27 AM

**Activity**

Filters: All time • All activities • All types

No activities to show.

Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

LearnSmart CRM

Enrollment College admission

**Related Details**

- Enrollment Name: College admission
- Owner: PORUMAMILLA KAVYA SREE
- Enrollment Status: Enrolled
- Course Fee: ₹50,000.00
- Payment Status: Paid
- Enrollment Date: 6/6/2024
- Student: PORUMAMILLA KAVYA SREE

Created By: PORUMAMILLA KAVYA SREE, 10/19/2025, 6:01 AM

Last Modified By: PORUMAMILLA KAVYA SREE, 10/19/2025, 7:28 AM

**Activity**

Filters: All time • All activities • All types

No activities to show.

Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

## 7. Apex with LWC

**Objective:** Connect UI components with backend Salesforce logic.

**Instructions:**

1. Create Apex classes to handle backend operations like fetching, updating, or deleting records.
2. Connect the Apex classes to your LWC to display and manipulate data dynamically.

3. Ensure that data is displayed correctly in the LWC interface.

## 8. Events in LWC

**Objective:** Enable communication between parent and child components.

**Instructions:**

1. Define events in child components to notify parent components when an action occurs.
2. Configure parent components to listen and respond to these events.
3. Use events to update UI or trigger further actions within the CRM.

## 9. Wire Adapters

**Objective:** Fetch Salesforce data reactively.

**Instructions:**

1. Connect LWCs to Salesforce data sources using reactive data fetching.
2. Ensure the component automatically updates when the underlying data changes.
3. Display the retrieved data in a structured and user-friendly manner.

## 10. Imperative Apex Calls

**Objective:** Call Apex methods on demand (e.g., when a button is clicked).

**Instructions:**

1. Configure LWCs to call Apex methods only when required by user action.
2. Handle the data returned from Apex and update the UI dynamically.
3. Manage errors or exceptions gracefully to ensure a smooth user experience.

## 11. Navigation Service

**Objective:** Navigate between pages, records, or URLs.

**Instructions:**

1. Set up navigation in LWCs to allow users to move between different CRM pages.
2. Configure navigation for records, objects, or external URLs.

3. Ensure navigation actions are intuitive and consistent with the CRM workflow.

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

This phase enables LearnSmart CRM to connect with external systems or APIs.

### **1. Named Credentials:**

Securely store external system authentication for use in Apex callouts.

### **2. External Services:**

Register external APIs for declarative access in Flows and Process Builder.

### **3. Web Services (REST/SOAP):**

Expose or consume APIs to integrate LearnSmart CRM with other systems.

### **4. Callouts:**

Connect to external APIs via HTTP requests and handle responses.

### **5. Platform Events:**

Enable event-driven communication between Salesforce and external apps.

### **6. Change Data Capture:**

Track record changes and trigger automation in Salesforce.

### **7. Salesforce Connect:**

Access and display external data in Salesforce without storing it locally.

### **Result of Phase 7:**

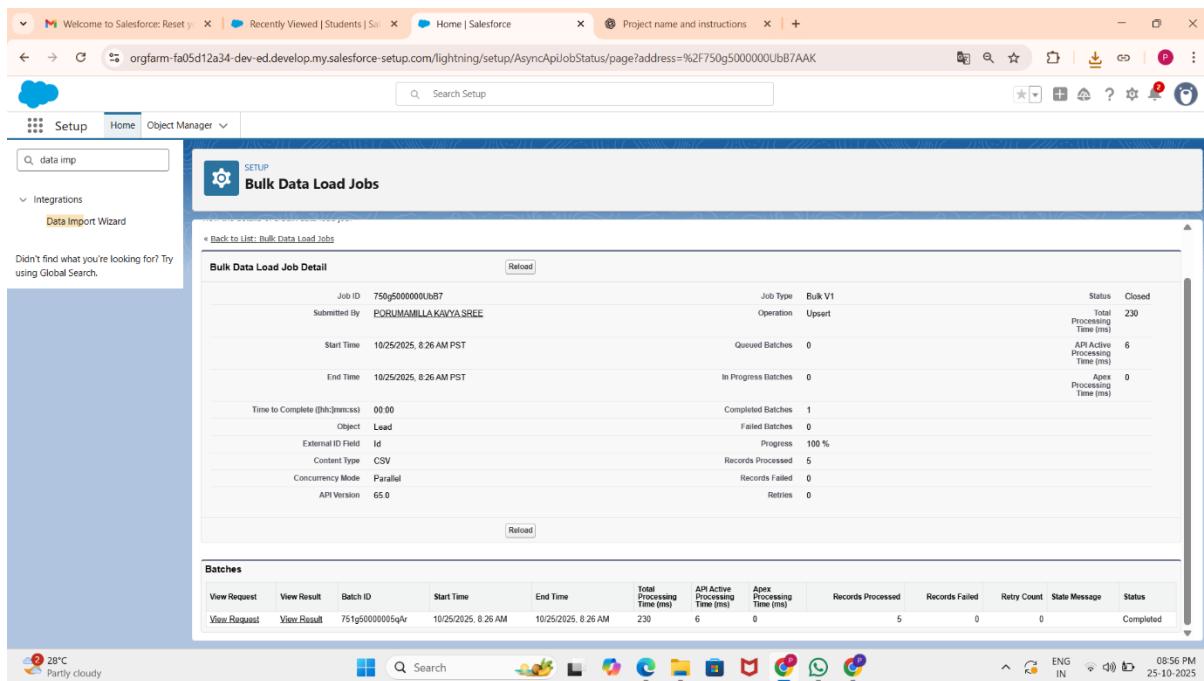
- Integration & External Access enables LearnSmart CRM to securely connect with external systems, APIs, and services.
- It allows real-time data exchange, event-driven updates, and seamless access to external data without storing it in Salesforce.

## Phase 8: Data Management & Deployment

Manage, migrate, and secure Salesforce data while preparing the org for deployment.

### Data Import Wizard

Tool to easily import records like Leads, Contacts, and Accounts into Salesforce.



The screenshot shows the Salesforce Setup interface with the Bulk Data Load Jobs page open. The job details are as follows:

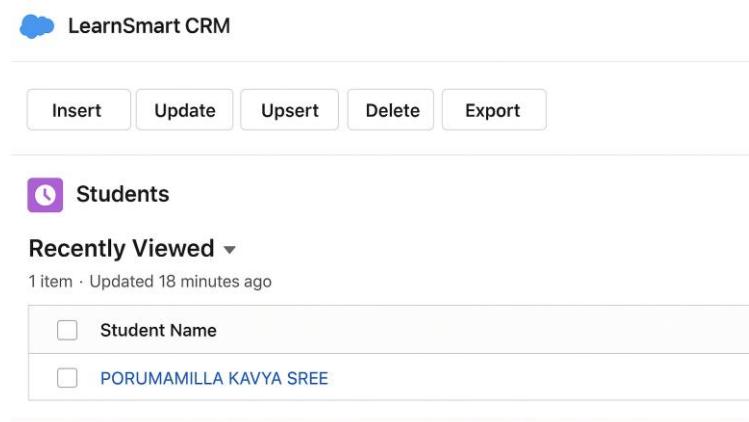
Job ID	750g5000000Ub87	Submitted By	PORUMAMILLA KAVYA SREE	Job Type	Bulk V1	Status	Closed
Start Time	10/25/2025, 8:26 AM PST			Operation	Upsert	Total Processing Time (ms)	230
End Time	10/25/2025, 8:26 AM PST			Queued Batches	0	API Active Processing Time (ms)	6
Time to Complete (hh:mm:ss)	00:00			In Progress Batches	0	Ajax Processing Time (ms)	0
Object	Lead			Completed Batches	1		
External ID Field	Id			Failed Batches	0		
Content type	CSV			Progress	100 %		
Concurrency Mode	Parallel			Records Processed	5		
API Version	65.0			Records Failed	0		
				Retries	0		

**Batches**

View Request	View Result	Batch ID	Start Time	End Time	Total Processing Time (ms)	API Active Processing Time (ms)	Ajax Processing Time (ms)	Records Processed	Records Failed	Retry Count	State Message	Status
View Request	View Result	751g50000005qAr	10/25/2025, 8:26 AM	10/25/2025, 8:26 AM	230	6	0	5	0	0	Completed	

### Data Loader

A client application for bulk import, update, export, and delete of Salesforce records.



The screenshot shows the LearnSmart CRM Data Loader interface. The top navigation bar includes buttons for Insert, Update, Upsert, Delete, and Export. Below the navigation is a section titled "Students" with a "Recently Viewed" dropdown. Underneath is a table with two rows:

<input type="checkbox"/>	Student Name
<input type="checkbox"/>	PORUMAMILLA KAVYA SREE

## Duplicate Rules

Set rules to prevent or alert users about duplicate records in the system.

The screenshot shows the 'Duplicate Rules' setup page for the 'Lead' object. The 'Rule Details' section includes a 'Rule Name' field set to 'Email' and an 'Object' dropdown also set to 'Lead'. Under 'Actions', 'Action On Create' is set to 'Allow' with 'Alert' checked and 'Report' unchecked. 'Action On Edit' is set to 'Allow' with both 'Alert' and 'Report' checked. An 'Alert Text' field contains the placeholder 'Use one of these records?'. The 'Matching Rules' section shows 'Compare Leads With' set to 'Leads' and 'Matching Rule' set to 'Standard Lead Matching Rule'. The 'Conditions' section contains a table with four rows, each with a 'Field' column (containing 'Email'), an 'Operator' column (containing 'Is Not'), and a 'Value' column (containing 'True'). The 'Conditions' table has three 'AND' operators between the rows. At the bottom of the page are 'Save', 'Save & New', and 'Cancel' buttons.

## Data Export & Backup

Regularly export Salesforce data for backup, compliance, and recovery purposes.

The screenshot shows the 'Data Export' setup page. The 'Monthly Export Service' section indicates that a data export is currently in progress. It provides options to 'Export Now' or 'Schedule Export'. Below this, a message states that the export has been queued and will be completed with an email notification. The export details listed are: 'Scheduled By' (PORUMAMILLA KAVYA SREE), 'Schedule Date' (10/25/2025), and 'Export File Encoding' (ISO-8859-1 (General US & Western European, ISO-LATIN-1)). The bottom of the page includes standard browser navigation and status icons.

## Change Sets

Deploy metadata between Salesforce orgs without manual configuration.

### Unmanaged vs Managed Packages

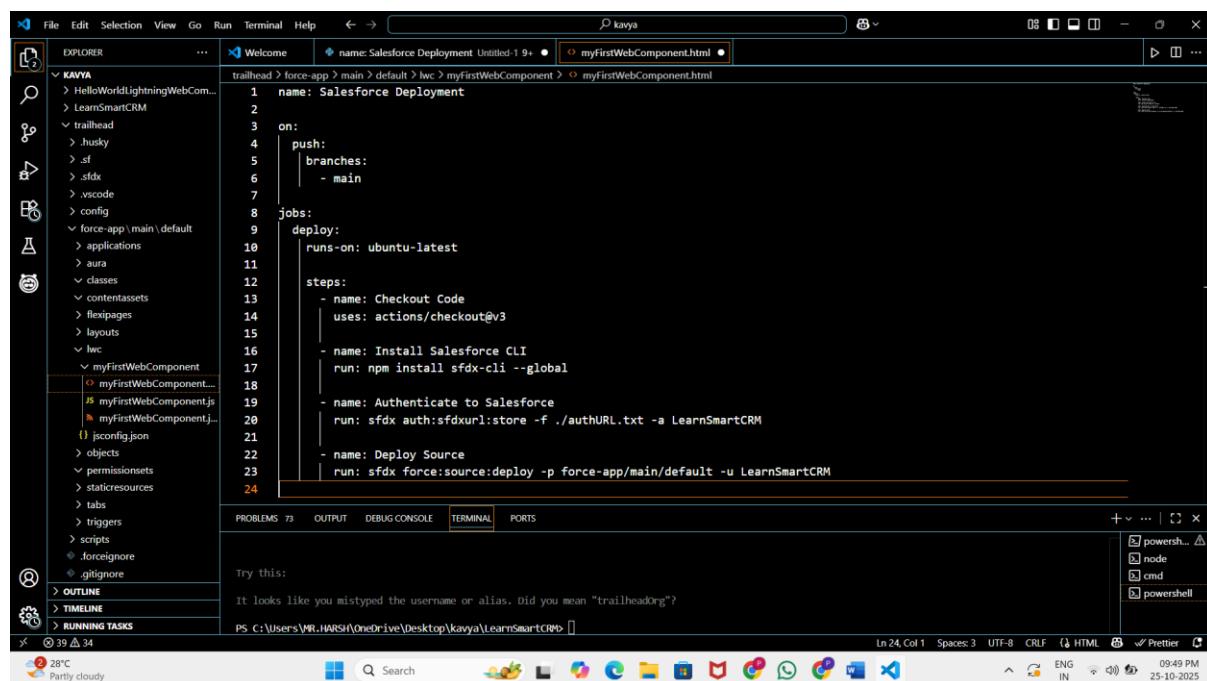
Unmanaged packages are for sharing code without versioning; managed packages allow upgrades and licensing.

## ANT Migration Tool

Command-line utility for deploying and retrieving metadata in Salesforce.

## VS Code & SFDX

Use Visual Studio Code with Salesforce DX for development, testing, and deployment automation.



```
name: Salesforce Deployment Untitled-1 9+ ● myFirstWebComponent.html ●
trailing-edge > force-app > main > default > lwc > myFirstWebComponent > myFirstWebComponent.html
1 name: Salesforce Deployment
2
3 on:
4   push:
5     branches:
6       - main
7
8   jobs:
9     deploy:
10    runs-on: ubuntu-latest
11
12    steps:
13      - name: Checkout Code
14        uses: actions/checkout@v3
15
16      - name: Install Salesforce CLI
17        run: npm install sfdx-cli --global
18
19      - name: Authenticate to Salesforce
20        run: sfdx auth:sfdxurl:store -f ./authURL.txt -a LearnSmartCRM
21
22      - name: Deploy Source
23        run: sfdx force:source:deploy -p force-app/main/default -u LearnSmartCRM
24
```

## Result of Phase 8:

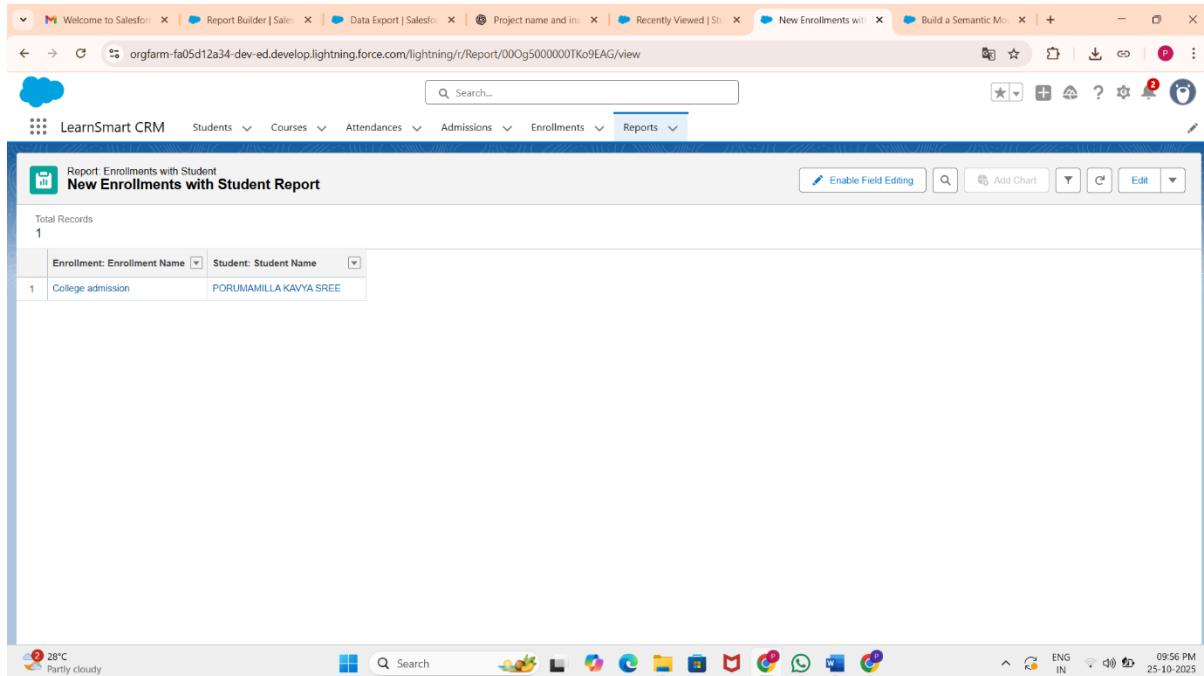
- Data is securely imported, exported, and backed up while metadata and org configurations are efficiently deployed across environments.
- Duplicates are controlled, and development tools like VS Code, SFDX, and ANT ensure smooth migration and package management.

## Phase 9: Reporting, Dashboards & Security Review

Analyze data and ensure proper access controls while creating insightful reports and dashboards.

### Reports (Tabular, Summary, Matrix, Joined)

Different formats to display Salesforce data in customizable views for analysis.

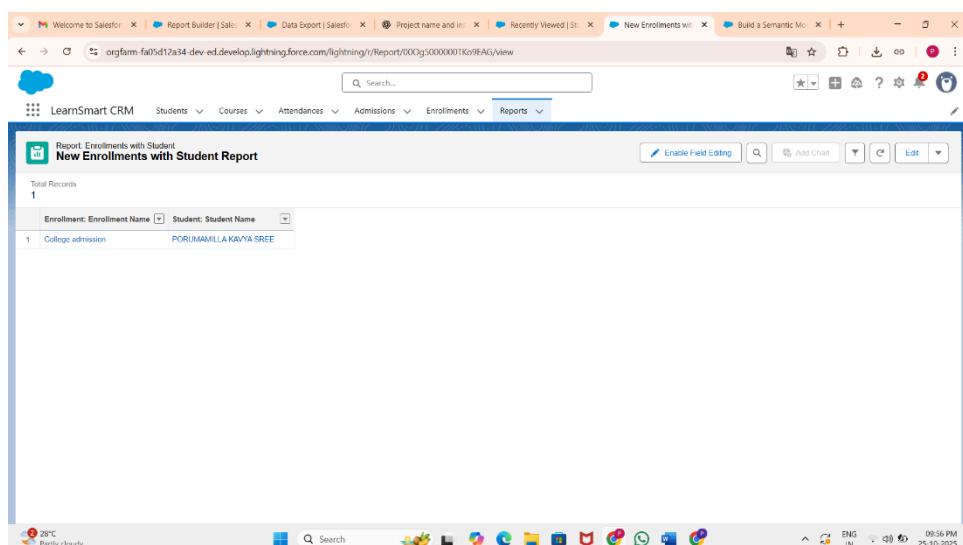


The screenshot shows a Salesforce report titled "Report: Enrollments with Student" and "New Enrollments with Student Report". The report displays a single record with the following details:

Enrollment: Enrollment Name	Student: Student Name
College admission	PORUMAMILLA KAVYA SREE

### Report Types

Define which objects and fields are available for creating reports.



The screenshot shows a Salesforce report titled "Report: Enrollments with Student" and "New Enrollments with Student Report". The report displays a single record with the following details:

Enrollment: Enrollment Name	Student: Student Name
College admission	PORUMAMILLA KAVYA SREE

## Dashboards

Visual representation of key metrics and reports for better decision-making.

New Enrollments with Student Report

Enrollment Name	Student Name	College admission
	PORUMAMILLA KAVYA SREE	

Your device needs to restart to install updates.  
Select a time to restart.

## Dynamic Dashboards

Dashboards that display data according to the logged-in user's access level.

## Sharing Settings

Control record-level access for users and groups in Salesforce.

Sharing Settings

Object	Default Internal Access	Default External Access	Granular Access Using Hierarchies
Lead	Public Read/Write/Transfer	Private	✓
Contact	Public Read/Write	Private	✓
Case	Controlled by Parent	Controlled by Parent	✓
Opportunity	Controlled by Parent	Controlled by Parent	✓
Asset	Public Read/Write	Private	✓
Campaign	Public Read/Write/Transfer	Private	✓
Campaign Member	Public Full Access	Private	✓
User	Controlled by Campaign	Controlled by Campaign	✓
	Public Read Only	Private	✓

## Field Level Security

Set visibility and edit permissions for individual fields across profiles and users.

The screenshot shows the Salesforce Object Manager interface. The left sidebar is titled 'Object Manager' and lists various configuration options under 'Fields & Relationships'. The main content area is titled 'Phone' and shows the 'Custom Field Definition Detail' for the 'Student' object. The 'Field Information' section includes details like Field Label (Phone), Field Name (Phone), API Name (Phone\_\_c), Description (Contact phone number of student or parent), Help Text (Enter 10-digit contact number), Data Owner (Field Usage), and Data Sensitivity Level (Compliance Categorization). The 'General Options' section shows 'Required' checked and 'Default Value' as blank. The 'Validation Rules' section indicates 'No validation rules defined'. The top navigation bar shows the current page as 'SETUP > OBJECT MANAGER Student'. The browser address bar shows the URL: <https://orgfarm-fa05d12a34-dev-ed.my.salesforce.com/lightning/setup/ObjectManager/01lg5000000AFYr/FieldsAndRelationships/00Ng5000001Do1U/view>. The bottom status bar shows the date and time as 10/17/2025, 8:03 AM.

## Result of Phase 9:

Comprehensive reports and dashboards provide actionable insights tailored to user access levels.

Security settings, including sharing rules and field-level permissions, ensure data protection and controlled visibility.