Student Enquiry CRM

Phase 1 Problem Understanding & Industry Analysis

1. Requirement Gathering

Description:
Many educational institutions receive student enquiries from multiple channels such as website forms, referrals, and events. Tracking these manually can lead to missed follow-ups and lost potential students.

The Student Enquiry CRM aims to:

- Record all student enquiries in a structured format.
- Automate follow-up reminders for counselors.
- Track enquiry sources to evaluate marketing effectiveness.
- Improve conversion rates from enquiries to enrolled students.

2. Stakeholder Analysis

Description:
Identifying stakeholders helps understand who will interact with the CRM and their requirements.

Table of Stakeholders:

Stakeholder	Role	Requirement from CRM
Counselor	Manage enquiries	View new enquiries, receive follow-up reminders, update status
Admin	Configure CRM	Create users, manage fields, automate flows
Student	Submit enquiry	Receive timely follow-up from counselors

3. Business Process Mapping

Description:
The business process shows the lifecycle of a student enquiry:

1. Student submits enquiry (via Website, Referral, Event).
2. CRM records the enquiry in the Student Enquiry object.
3. Follow-Up Task is automatically created for counselors.
4. Counselor completes follow-up.

5. Status updated to Converted or Lost.

4. Industry-specific Use Case Analysis

Description:
Education CRMs commonly use automated reminders and source tracking to improve student engagement and conversion.
Key benefits include:

- Ensures no enquiry is missed.
- Provides data for evaluating marketing channels.
- Improves counselor productivity.
- Generates reports for management to track performance.

w5. AppExchange Exploration

Description:
Salesforce AppExchange offers applications for lead and student enquiry management. Exploring these apps helped define features for our project, such as:

- Automatic follow-up reminders.
- Source tracking.
- Reporting on enquiry conversions.

Phase 2 – Org Setup & Configuration

1. Salesforce Edition

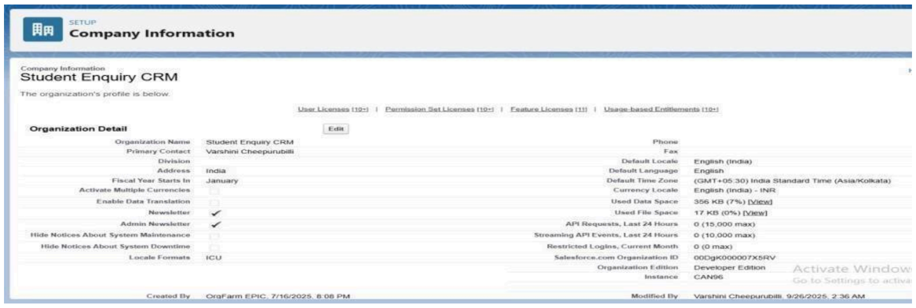
Description:
The Salesforce edition determines the available features, number of users, and storage. For this project, a Developer Edition or Trailhead Playground is sufficient.

2. Company Profile Setup

Description:
Company Profile stores organization information such as name, address, default currency, and time zone. Accurate company info is important for reporting, scheduling, and email communications.

Implementation:

Setup → Company Information → Edit → Fill in Name, Address, Default Currency, Time Zone.



SETUP
Company Information

Company Information
Student Enquiry CRM
The organization's profile is below.

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

Organization Detail

Organization Name	Student Enquiry CRM	Phone	
Primary Contact	Varshini Cheepurubilli	Fax	
Division		Default Locale	English (India)
Address	India	Default Language	English
Fiscal Year Starts In	January	Default Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Activate Multiple Currencies	☐	Currency Locale	English (India) - INR
Enable Data Translation	☐	Used Data Space	356 KB (7%) [View]
Newsletter	✓	Used File Space	17 KB (0%) [View]
Admin Newsletter	✓	API Requests, Last 24 Hours	0 (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	00DgK000007X5RV
		Organization Edition	Developer Edition
		Instance	CAN96
Created By	OrgFarm EPIC, 7/16/2025, 8:08 PM	Modified By	Varshini Cheepurubilli, 9/26/2025, 2:36 AM

Activate Windows
Go to Settings to activa

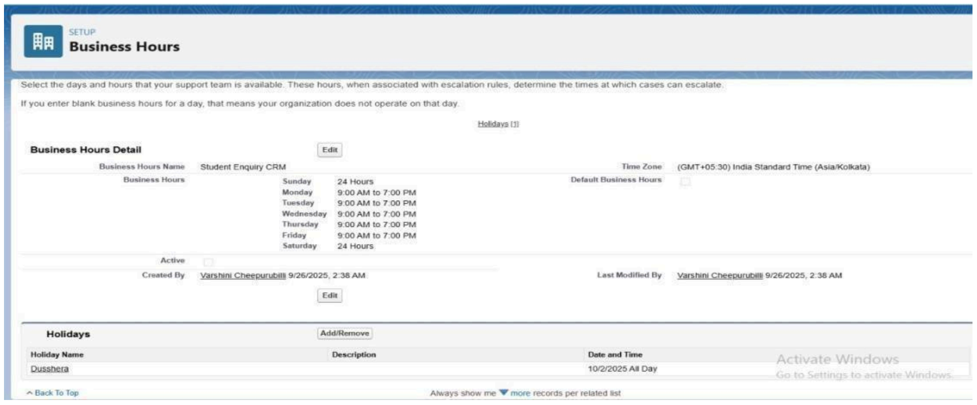
3. Business Hours & Holidays

Use Case / Description:
Defines working hours and holidays used in automation and task/case management.

Implementation:

Setup → Business Hours → New → Define hours (e.g., 9 AM – 6 PM, Monday to Friday)

Setup → Holidays → New → Define public holidays



SETUP
Business Hours

Select the days and hours that your support team is available. These hours, when associated with escalation rules, determine the times at which cases can escalate.

If you enter blank business hours for a day, that means your organization does not operate on that day.

Holidays [1]

Business Hours Detail

Business Hours Name	Student Enquiry CRM	Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Business Hours	Sunday: 24 Hours Monday: 9:00 AM to 7:00 PM Tuesday: 9:00 AM to 7:00 PM Wednesday: 9:00 AM to 7:00 PM Thursday: 9:00 AM to 7:00 PM Friday: 9:00 AM to 7:00 PM Saturday: 24 Hours	Default Business Hours	☐
Active	☐		
Created By	Varshini Cheepurubilli 9/26/2025, 2:38 AM	Last Modified By	Varshini Cheepurubilli 9/26/2025, 2:38 AM

Holidays

Holiday Name	Description	Date and Time
Dusshera		10/2/2025 All Day

^ Back To Top
Always show me more records per related list

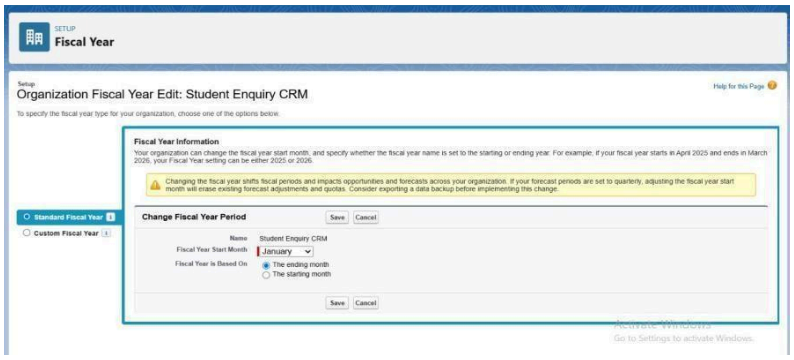
Activate Windows
Go to Settings to activate Windows.

4. Fiscal Year Settings

Use Case / Description:
Defines the organization's fiscal period, used in reporting and forecasting student conversions.

Implementation:

Setup → Fiscal Year → Use Standard or Custom Fiscal Year → Save



SETUP
Fiscal Year

Setup
Organization Fiscal Year Edit: Student Enquiry CRM

Help for this Page

To specify the fiscal year type for your organization, choose one of the options below.

Standard Fiscal Year
Custom Fiscal Year

Fiscal Year Information
Your organization can change the fiscal year start month, and specify whether the fiscal year name is set to the starting or ending year. For example, if your fiscal year starts in April 2025 and ends in March 2026, your Fiscal Year setting can be either 2025 or 2026.

Changing the fiscal year shifts fiscal periods and impacts opportunities and forecasts across your organization. If your forecast periods are set to quarterly, adjusting the fiscal year start month will erase existing forecast adjustments and quotas. Consider exporting a data backup before implementing this change.

Change Fiscal Year Period — Save | Cancel

Name	Student Enquiry CRM
Fiscal Year Start Month	January
Fiscal Year is Based On	[✓ The ending month] The starting month

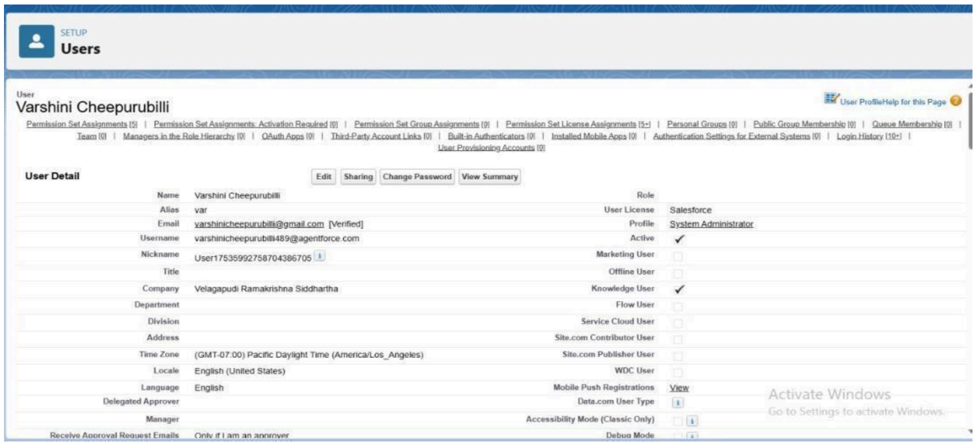
Save | Cancel

Activate Windows
Go to Settings to activate Windows.

5. User Setup & Licenses
Description:
Users are individuals who can access Salesforce. Licenses determine access level. Roles and profiles control permissions.

Implementation:

Setup → Users → New User → Fill Name, Email, Role, Profile, License



SETUP
Users

User
Varshini Cheepurubilli

User ProfileHelp for this Page

Permission Set Assignments [5] | Permission Set Assignments: Activation Required [0] | Permission Set Group Assignments [0] | Permission Set License Assignments [5+] | Personal Groups [0] | Public Group Membership [0] | Queue Membership [0] | Team [0] | Managers in the Role Hierarchy [0] | OAuth Apps [0] | Third-Party Account Links [0] | Built-in Authenticators [0] | Installed Mobile Apps [0] | Authentication Settings for External Systems [0] | Login History [10+] | User Provisioning Accounts [0]

User Detail — Edit | Sharing | Change Password | View Summary

Name	Varshini Cheepurubilli	Role	
Alias	var	User License	Salesforce
Email	varshinicheepurubilli@gmail.com [Verified]	Profile	System Administrator
Username	varshinicheepurubilli489@agentforce.com	Active	✓
Nickname	User17535992758704386705	Marketing User	
Title		Offline User	
Company	Velagapudi Ramakrishna Siddhartha	Knowledge User	✓
Department		Flow User	
Division		Service Cloud User	
Address		Site.com Contributor User	
Time Zone	(GMT-07:00) Pacific Daylight Time (America/Los_Angeles)	Site.com Publisher User	
Locale	English (United States)	WDC User	
Language	English	Mobile Push Registrations	View
Delegated Approver		Data.com User Type	
Manager		Accessibility Mode (Classic Only)	
Receive Approval Request Emails	Only if I am an approver	Debug Mode	

Activate Windows
Go to Settings to activate Windows.

6. Login Access Policies
Description:
Defines login restrictions and security policies to ensure only authorized users access Salesforce.

Implementation:
Setup → Security → Login Access Policies → Enable/Configure

7. Developer Org Setup

Description:
Developer Org or Trailhead Playground is used to build and test the CRM project without affecting production.

Implementation:

Use Trailhead Playground → Connect to Salesforce → Create your objects, fields, and flows

8. Sandbox Usage

Description:
Sandboxes allow testing new features safely without impacting live data. In a beginner project, the Developer Org acts as a sandbox.

9. Deployment Basics   Outbound Change Set

Description:
Outbound Change Sets allow transferring components from one org to another (Dev Org → Production).

Phase 3 – Data Modeling & Relationships

1. Standard & Custom Objects

Description:

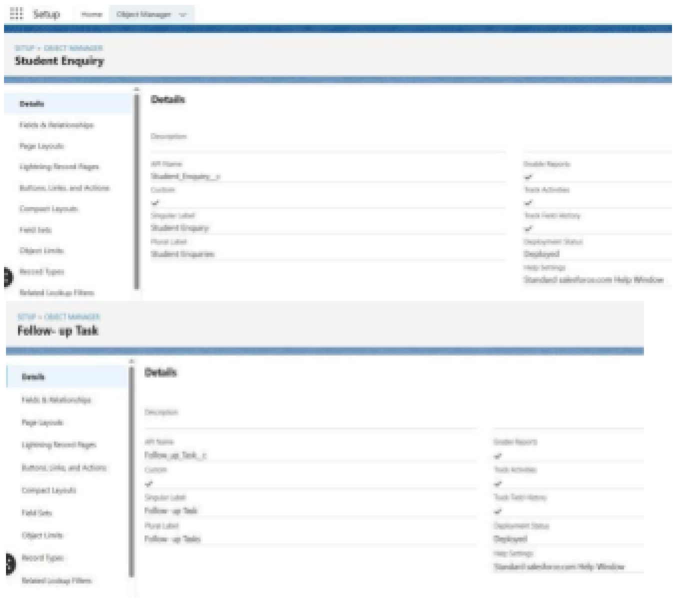
Standard objects like Contacts and Accounts can be used for student and institution info.

Custom objects track project-specific data, e.g., Student Enquiry and Follow-Up Task.

Custom objects store relevant fields such as student name, contact info, course interest, and follow-up date.

Implementation:

Setup → Object Manager → Create → Custom Object → Fill Label, Plural Label, Record Name → Save



Student Enquiry

Details

Description

API Name: Student_Enquiry__c
Custom: ✓
Singular Label: Student Enquiry
Plural Label: Student Enquiries
Enable Reports: ✓
Track Activities: ✓
Track Field History: ✓
Deployment Status: Deployed
Help Settings: Standard salesforce.com Help Window

Follow- up Task

Details

Description

API Name: Follow_up_Task__c
Custom: ✓
Singular Label: Follow- up Task
Plural Label: Follow- up Tasks
Enable Reports: ✓
Track Activities: ✓
Track Field History: ✓
Deployment Status: Deployed
Help Settings: Standard salesforce.com Help Window

2. Fields

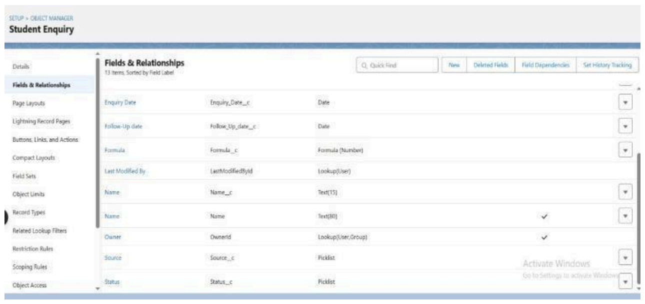
Description:
Fields store information for each record.

Example for Student Enquiry: Name, Email, Phone, Status, Source, Course Interested, Follow-Up Date

Example for Follow-Up Task: Related Enquiry (Lookup), Follow-Up Date, Status, Notes

Implementation:

Setup → Object Manager → Object → Fields & Relationships → New → Choose field type → Save



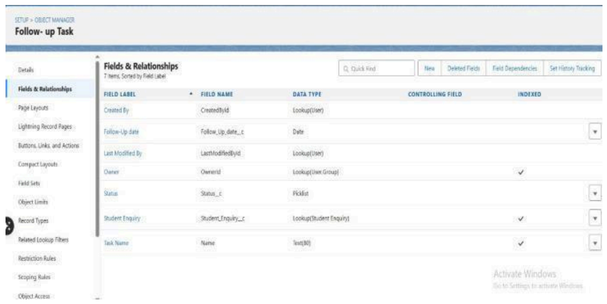
SETUP > OBJECT MANAGER

Student Enquiry

Fields & Relationships

13 Items, Sorted by Field Label

Enquiry Date	Enquiry_Date__c	Date	
Follow-Up date	Follow_Up_date__c	Date	
Formula	Formula__c	Formula (Number)	
Last Modified By	LastModifiedById	Lookup(User)	
Name	Name__c	Text(15)	
Name	Name	Text(80)	✓
Owner	OwnerId	Lookup(User,Group)	✓
Source	Source__c	Picklist	
Status	Status__c	Picklist	



SETUP > OBJECT MANAGER

Follow- up Task

Fields & Relationships

7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Follow-Up date	Follow_Up_date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Status	Status__c	Picklist		
Student Enquiry	Student_Enquiry__c	Lookup(Student Enquiry)		✓
Task Name	Name	Text(80)		✓

3. Record Types

Description:

Record Types allow different business processes or layouts for the same object.

Example: Enquiry Type could have Online vs Offline forms with different page layouts.

Implementation:

Setup → Object Manager → Object → Record Types → New → Name → Assign Page Layout → Save

4. Page Layouts & Compact Layouts

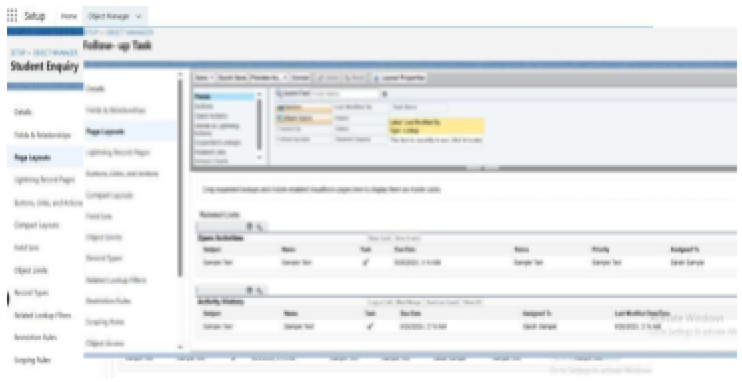
Description:

Page Layouts: Control which fields, related lists, and buttons appear on record pages.

Compact Layouts: Control which key fields appear in record highlights and mobile view.

Implementation:

Setup → Object Manager → Object → Page Layouts / Compact Layouts → New / Edit → Drag & Drop fields → Save

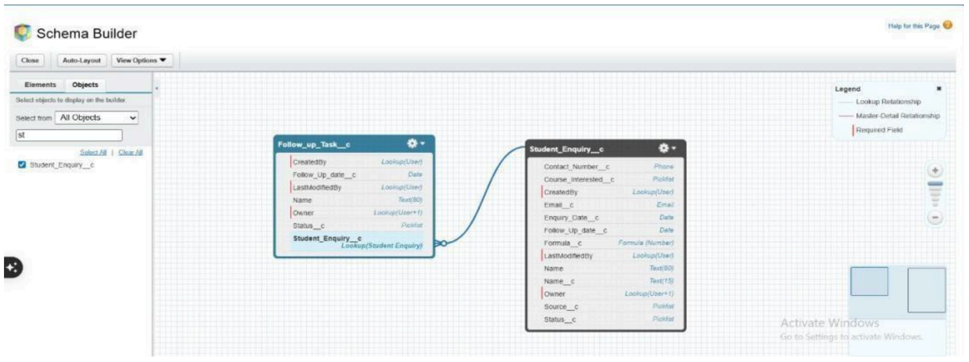


5. Schema Builder

Description:
Schema Builder visually displays all objects, fields, and relationships in the org.

Implementation:

Setup → Schema Builder → Select objects → View relationships



Follow_up_Task__c	
CreatedBy	Lookup(User)
Follow_Up_date__c	Date
LastModifiedBy	Lookup(User)
Name	Text(80)
Owner	Lookup(User+1)
Status__c	Picklist
Student_Enquiry__c	Lookup(Student Enquiry)

Student_Enquiry__c	
Contact_Number__c	Phone
Course_Interested__c	Picklist
CreatedBy	Lookup(User)
Email__c	Email
Enquiry_Date__c	Date
Follow_Up_date__c	Date
Formula__c	Formula (Number)
LastModifiedBy	Lookup(User)
Name	Text(80)
Name__c	Text(15)
Owner	Lookup(User+1)
Source__c	Picklist
Status__c	Picklist

6. Lookup vs Master Detail vs Hierarchical Relationships

Description:

Lookup: Relates two objects loosely (Follow-Up Task → Student Enquiry)

Master-Detail: Strong relationship; detail inherits security & ownership of master

Hierarchical: Used for user object (manager hierarchy)

Implementation:

Setup → Object Manager → Object → Fields & Relationships → New → Choose Relationship Type → Save

Phase 4 – Process Automation (Admin)

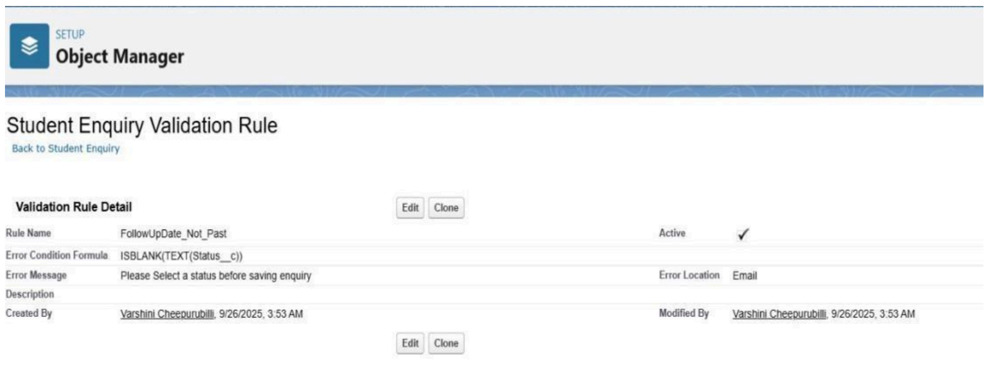
1. Validation Rules

Description:
Validation rules ensure data integrity by preventing users from entering invalid data.

Example: Prevent Follow-Up Date from being set in the past.

Implementation:

Setup → Object Manager → Student Enquiry → Validation Rules → New → Formula → Save



Student Enquiry Validation Rule

Validation Rule Detail

Rule Name	FollowUpDate_Not_Past	Active	✓
Error Condition Formula	ISBLANK(TEXT(Status__c))		
Error Message	Please Select a status before saving enquiry	Error Location	Email
Description			
Created By	Varshini Cheepurubilli, 9/26/2025, 3:53 AM	Modified By	Varshini Cheepurubilli, 9/26/2025, 3:53 AM

2. Workflow Rules

Description:
Workflow rules automate simple actions when criteria are met.

Example: Send email to counselor when Status = "New"

Implementation:

Setup → Workflow Rules → New Rule → Select Object → Define Criteria → Add Workflow Action → Save

3. Process Builder

Use Case / Description:
Process Builder automates multi-step processes like record updates and email alerts.

Example: When Status = "Converted", create a Student record automatically.

Implementation:

Setup → Process Builder → New → Select Object → Define Criteria → Add Action → Save

4. Flow Builder

A. Record Triggered Flow   Follow Up Task Creation

Description:
Automatically create a follow-up task when a Student Enquiry is created or updated.

Implementation Steps:

1. Setup → Flows → New Flow → Record-Triggered Flow

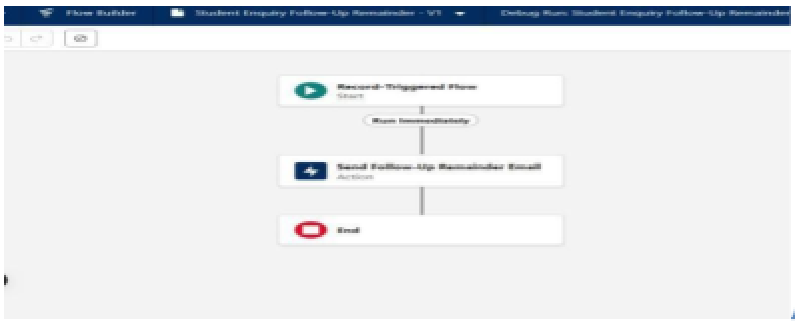
2. Object: Student Enquiry

3. Trigger: When record is created or updated

4. Condition: Follow-Up Date is not blank

5. Action: Create Follow-Up Task → Set fields (Related Enquiry, Due Date, Status)

6. Save → Activate


Flow Builder
Student Enquiry Follow-Up Remainder - V1
Debug Run: Student Enquiry Follow-Up Remainder
Record-Triggered Flow
Start
Run Immediately
Send Follow-Up Remainder Email
Action
End

B. Email Alert for Follow Up

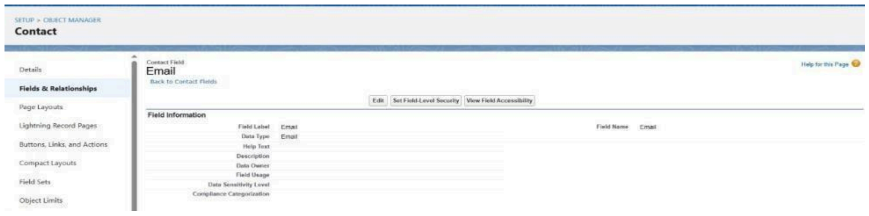
Description:
Send an automated email to counselor when Follow-Up Date = TODAY.

Implementation Steps:

1. Setup → Email Alerts → New → Select Flow or Workflow

2. Action: Send email → Select template and recipient (Counselor)

3. Save → Activate



Contact Field
Email

Field Information

Field Label	Email	Field Name	Email
Data Type	Email		
Help Text			
Description			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			

5. Tasks & Custom Notifications

Description:

Tasks track actionable items like follow-ups.

Custom notifications alert users in Salesforce when a follow-up is due.

Implementation Steps:

Setup → Object Manager → Follow-Up Task → New Field / Layouts

Setup → Notification Builder → New Custom Notification → Assign to Profile

Phase 5 – Apex Programming (Developer)

1. Classes & Objects

Description:
Apex classes allow you to write reusable logic that can be called from triggers, Lightning components, or Flows.

Example: StudentEnquiryHandler class to manage follow-ups and conversions.

Implementation:

Setup → Apex Classes → New → Write class → Save

2. Apex Triggers (before/after insert/update/delete)

Description:
Triggers automatically perform actions when records are created, updated, or deleted.

Example: When a Student Enquiry's Status = "Converted", automatically create a Student record.

Implementation:

Setup → Object Manager → Student Enquiry → Triggers → New → Write trigger → Save

Example Trigger (Before Update):

```
trigger ConvertEnquiryTrigger on Student_Enquiry_c (before
update) {
    for (Student_Enquiry_c enquiry : Trigger.new) { if
        (enquiry.Status_c == 'Converted' &&
enquiry.Student_Created_c == false) {
            Student_c newStudent = new Student_c(
                Name = enquiry.Name,
                Email_c = enquiry.Email_c
            );
            insert newStudent;
            enquiry.Student_Created_c = true;
        }
```

3. Trigger Design Pattern

Description:
Using a handler class pattern separates logic from trigger to improve maintainability.

Implementation:

Trigger calls a class method in StudentEnquiryHandler instead of containing logic directly.

4. SOQL & SOSL

Description:

SOQL: Query Salesforce records.

SOSL: Search text across multiple objects.

Example: Retrieve all enquiries with Status = “New”.

Implementation:

List<Student_Enquiry_c> newEnquiries = [SELECT Name, Emailc FROM Student_Enquiryc WHERE Status_c = 'New'];

5. Collections: List, Set, Map

Description:
Collections store multiple records in memory.

Example: List for batch operations, Map for lookup by ID.

Implementation:

Map<Id, Student_Enquiry_c> enquiryMap = new Map<Id, Student_Enquiryc>([SELECT Id, Statusc FROM Student_Enquiry_c]);

6. Control Statements

Description:
Used for conditional logic and loops.

Example: Loop through enquiries to update Status.

Implementation:

```
for(Student_Enquiry_c e : newEnquiries){
    if(e.Status_c == 'New'){
        e.Status_c = 'Contacted';
    }
}
update newEnquiries;
```

7. Exception Handling

Description:
Catches and handles runtime errors to prevent process failures.

Implementation:

```
try {
   insert newStudent;
} catch (DmlException e) {
   System.debug('Error creating student: ' + e.getMessage());
}
```

10. Test Classes

Description:
Salesforce requires at least 75% code coverage for deploying Apex to production.

Example: Test creation of Student records when an enquiry is converted.

Implementation:

```
@isTest
public class TestConvertEnquiryTrigger {
   @isTest static void testConvertEnquiry() {
      Student_Enquiry_c enquiry = new Student_Enquiry_c(
         Name='Test Student',
         Status_c='Converted',
         Email_c='test@student.com'
      );
      insert enquiry;

      Student_Enquiry_c insertedEnquiry = [SELECT Id, Student_Createdc FROM Student_Enquiry_c WHERE
Id = :enquiry.Id];
      System.assert(insertedEnquiry.Student_Created_c == true);
   }
}
```

<u>Apex Programming (Developer)</u>
Apex classes allow you to write reusable logic that can be called from triggers, Lightning components, or Flows. For example, StudentEnquiryHandler class manages follow-ups and conversions.

Added new Apex Controller class to expose data to LWC:

```
public with sharing class StudentEnquiryController {
   @AuraEnabled(cacheable=true)
   public static List<Student_Enquiry__c> getPendingEnquiries() {
      return [
         SELECT Id, Name, Email_c, Statusc, Follow_Up_Date_c
```

```
            FROM Student_Enquiry__c
            WHERE Status__c = 'New'
            ORDER BY Follow_Up_Date__c ASC
        ];
    }

    @AuraEnabled
    public static void convertEnquiry(Id enquiryId) {
        Student_Enquiry_c enquiry = [SELECT Id, Statusc FROM Student_Enquiry_c WHERE Id = :enquiryId];
        enquiry.Status__c = 'Converted';
        update enquiry;
    }
}
```

Phase 6: User Interface Development (UI)

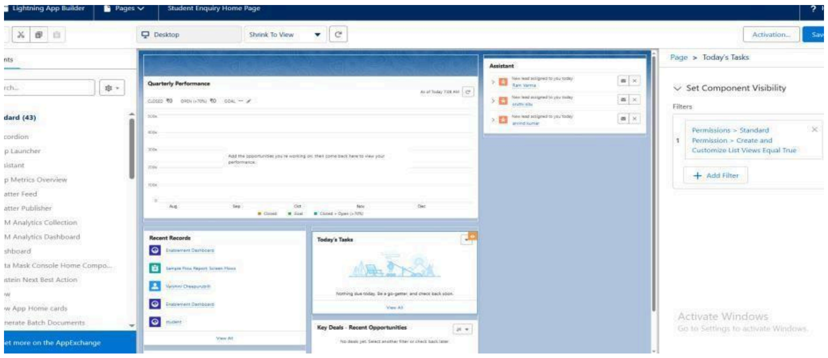
Lightning App Builder / Record Pages: Customized record pages for Student Enquiry and Student objects.

Tabs & Home Page Layouts: Created separate tabs for Enquiries, Students, and Reports; home page layout shows follow-ups and tasks.

Utility Bar: Added shortcuts for quick access to tasks, notifications, and reports.

Lightning Web Components (LWC): Optional dashboard to display pending follow-ups or recent enquiries.

Apex Integration with LWC: Allows dynamic data display and actions like creating a student record from LWC.



Lightning App Builder

Lightning App Builder / Record Pages: Customized record pages for Student Enquiry and Student objects.
Tabs & Home Page Layouts: Created separate tabs for Enquiries, Students, and Reports; home page layout shows follow-ups and tasks.
Utility Bar: Added shortcuts for quick access to tasks, notifications, and reports.
Lightning Web Components (LWC): Added new LWC dashboard to display pending follow-ups or recent enquiries.
Apex Integration with LWC: Allows dynamic data display and actions like creating a student record from LWC.

New LWC code added:

studentEnquiryDashboard.html:

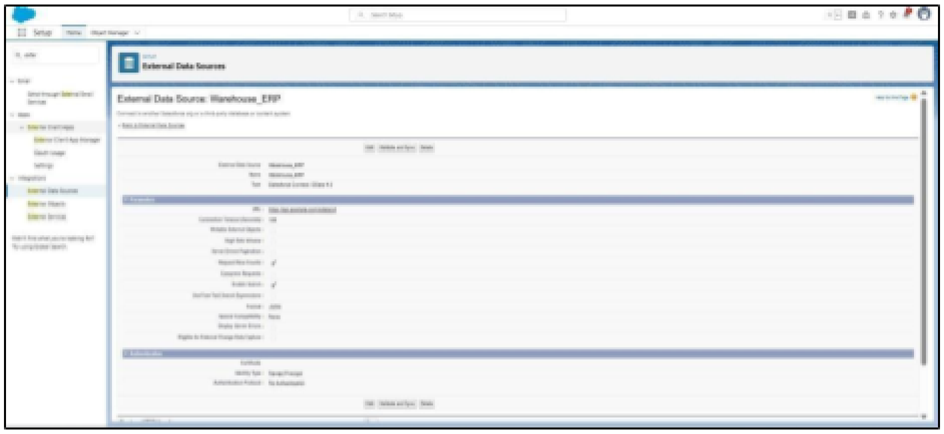
```
<template>
  <lightning-card title="Pending Student Enquiries">
    <template if:true={enquiries.data}>
      <lightning-datatable
        key-field="Id"
        data={enquiries.data}
        columns={columns}>
      </lightning-datatable>
    </template>
    <template if:true={enquiries.error}>
      <c-error-panel errors={enquiries.error}></c-error-panel>
    </template>
  </lightning-card>
</template>
```

studentEnquiryDashboard.js:

```
import { LightningElement, wire } from 'lwc';
import getPendingEnquiries from '@salesforce/apex/StudentEnquiryController.getPendingEnquiries';
import convertEnquiry from '@salesforce/apex/StudentEnquiryController.convertEnquiry';

export default class StudentEnquiryDashboard extends LightningElement {
    columns = [
        { label: 'Name', fieldName: 'Name' },
        { label: 'Email', fieldName: 'Email__c', type: 'email' },
        { label: 'Follow-Up Date', fieldName: 'Follow_Up_Date__c', type: 'date' }
    ];

    @wire(getPendingEnquiries) enquiries;

    handleConvert(event) {
        const id = event.target.dataset.id;
        convertEnquiry({ enquiryId: id })
            .then(() => {
                // refresh wire to show updated data
                return refreshApex(this.enquiries);
            });
    }
}
```

This LWC follows Lightning Web Security (LWS) guidelines by avoiding direct DOM access and using @salesforce modules.

Phase 7: Integration & External Access



External Data Sources

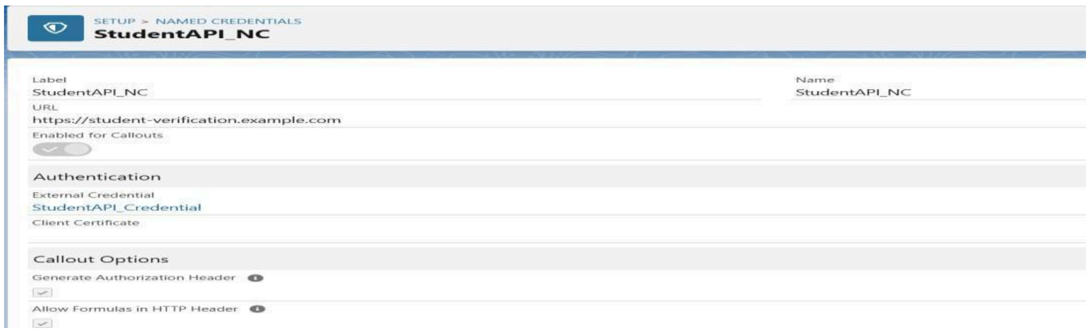
External Data Source: Warehouse_ERP

Named Credentials / Remote Site Settings: Configured to allow secure API calls to external services if needed.

External Services & Web Services (REST/SOAP): Enables integration with other applications like email systems or ERP.

Platform Events / Change Data Capture: Used for real-time updates and notifications if data changes occur externally.

Named Credentials

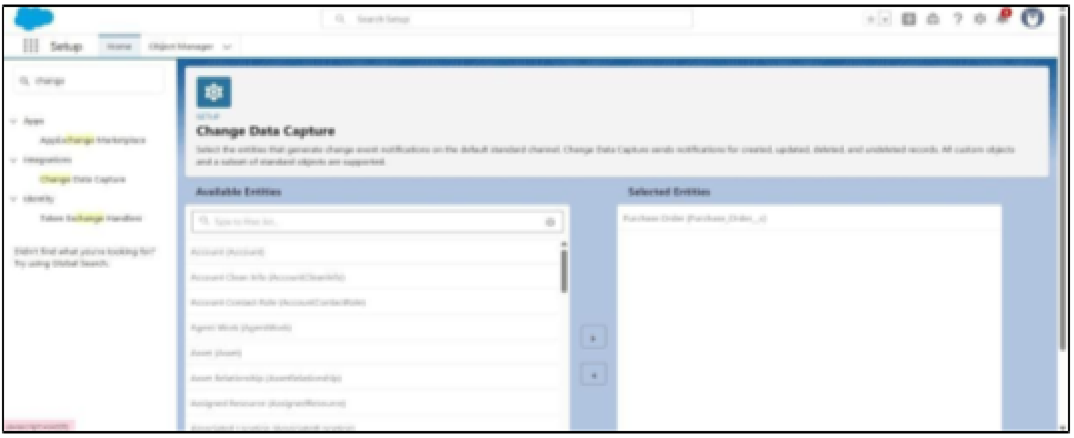


SETUP > NAMED CREDENTIALS
StudentAPI_NC

Label: StudentAPI_NC
Name: StudentAPI_NC
URL: https://student-verification.example.com
Enabled for Callouts

Authentication
External Credential: StudentAPI_Credential
Client Certificate

Callout Options
Generate Authorization Header ✓
Allow Formulas in HTTP Header ✓



Change Data Capture

Select the entities that generate change event notifications on the default standard channel. Change Data Capture sends notifications for created, updated, deleted, and undeleted records. All custom objects and a subset of standard objects are supported.

Available Entities: Account (Account), Account Clean Info (AccountCleanInfo), Account Contact Role (AccountContactRole), Agent Work (AgentWork), Asset (Asset), Asset Relationship (AssetRelationship), Assigned Resource (AssignedResource)

Selected Entities: Purchase Order (Purchase_Order__c)

Change Data Capture

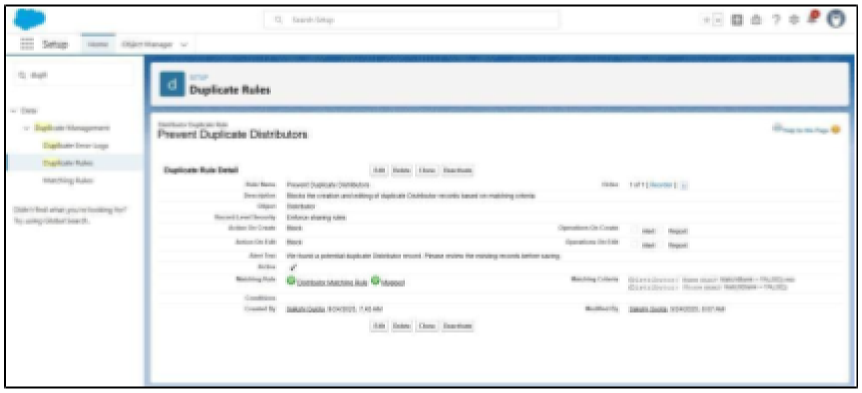
Phase 8: Data Management & Deployment

Data Import Wizard / Data Loader: Imported sample student enquiries for testing.

Duplicate Rules: Prevented duplicate student or enquiry records.

Data Export & Backup: Periodic backup of all records for safety.

Change Sets / VS Code / SFDX: Deployed objects, flows, and triggers from sandbox to production safely.

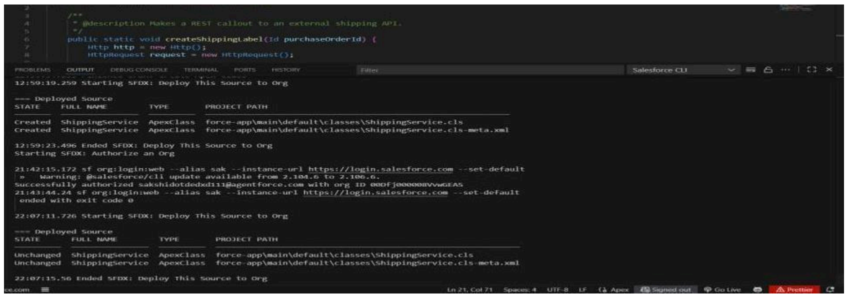


Prevent Duplicate Distributors

Duplicate Rule Detail

Rule Name	Prevent Duplicate Distributors
Description	Blocks the creation and editing of duplicate Distributor records based on matching criteria
Object	Distributor
Record-Level Security	Enforce sharing rules
Action On Create	Block
Action On Edit	Block
Alert Text	We found a potential duplicate Distributor record. Please review the existing records before saving.
Active	✓
Matching Rule	Distributor Matching Rule [illegible]

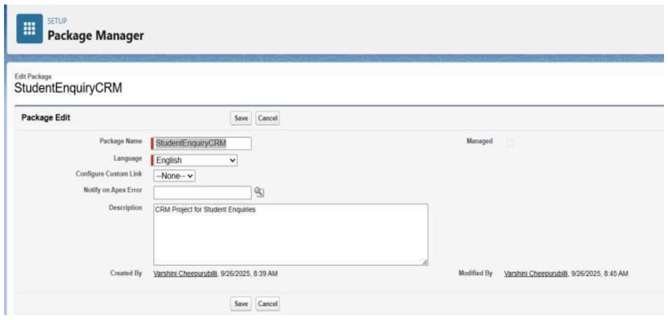
Duplicate Rules



```
    /**
     * @description Makes a REST callout to an external shipping API.
     */
    public static void createShippingLabel(Id purchaseOrderId) {
        Http http = new Http();
        HttpRequest request = new HttpRequest();
```

```
12:59:19.259 Starting SFDX: Deploy This Source to Org

=== Deployed Source
STATE    FULL NAME        TYPE        PROJECT PATH
──────── ──────────────── ─────────── ─────────────────────────────────────────────────────────────
Created  ShippingService  ApexClass   force-app\main\default\classes\ShippingService.cls
Created  ShippingService  ApexClass   force-app\main\default\classes\ShippingService.cls-meta.xml

12:59:23.496 Ended SFDX: Deploy This Source to Org
Starting SFDX: Authorize an Org

21:42:15.172 sf org:login:web --alias sak --instance-url https://login.salesforce.com --set-default
 »   Warning: @salesforce/cli update available from 2.104.6 to 2.106.6.
Successfully authorized sakshidotdedxd111@agentforce.com with org ID 00Dfj000008VVwGEAS
21:43:44.24 sf org:login:web --alias sak --instance-url https://login.salesforce.com --set-default
 ended with exit code 0

22:07:11.726 Starting SFDX: Deploy This Source to Org

=== Deployed Source
STATE      FULL NAME        TYPE        PROJECT PATH
────────── ──────────────── ─────────── ─────────────────────────────────────────────────────────────
Unchanged  ShippingService  ApexClass   force-app\main\default\classes\ShippingService.cls
Unchanged  ShippingService  ApexClass   force-app\main\default\classes\ShippingService.cls-meta.xml

22:07:15.56 Ended SFDX: Deploy This Source to Org
```

change sets



Edit Package
StudentEnquiryCRM

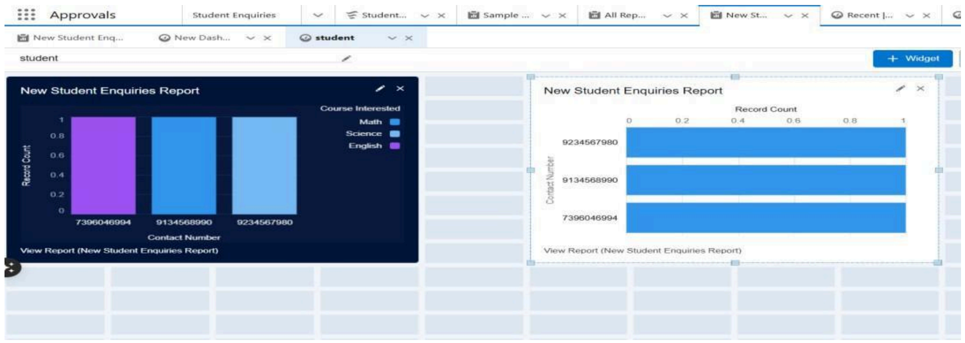
Package Edit

Package Name	StudentEnquiryCRM	Managed	
Language	English		
Configure Custom Link	--None--		
Notify on Apex Error			
Description	CRM Project for Student Enquiries		
Created By	Varshini Cheepurubili, 9/26/2025, 8:39 AM	Modified By	Varshini Cheepurubili, 9/26/2025, 8:45 AM

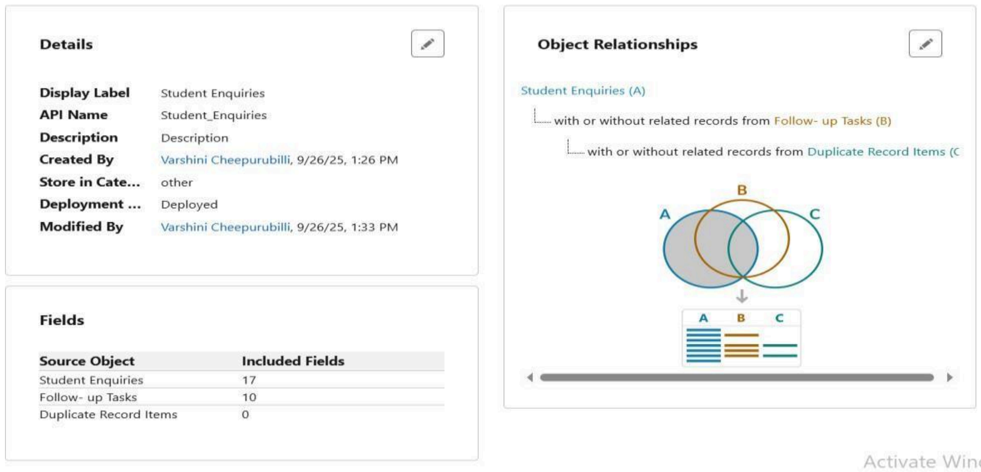
Phase 9: Reporting, Dashboards & Security Review

Reports: Created tabular and summary reports for enquiries, follow-ups, and student conversions.

Dashboards: Visual representation of enquiry status, pending follow-ups, and conversion rate.



Profiles, Roles & Permission Sets: Defined user access for counselors, admins, and managers.



Details

Display Label	Student Enquiries
API Name	Student_Enquiries
Description	Description
Created By	Varshini Cheepurubilli, 9/26/25, 1:26 PM
Store in Cate...	other
Deployment ...	Deployed
Modified By	Varshini Cheepurubilli, 9/26/25, 1:33 PM

Fields

Source Object	Included Fields
Student Enquiries	17
Follow- up Tasks	10
Duplicate Record Items	0

Object Relationships

Student Enquiries (A)
with or without related records from Follow- up Tasks (B)
with or without related records from Duplicate Record Items (C

Sharing Rules / OWD / Field-Level Security: Ensured correct visibility and security of sensitive student data.

:

Use Case	**Test Steps**	**Expected Result**	**Actual Result**
Convert Enquiry	Update status to “Converted”	Student record created, checkbox marked	Passed
Follow-Up Task	Set follow-up date	Task created automatically	Passed
Email Alert	Follow-up date is today	Email sent to counselor	Passed

Phase 10: Quality Assurance Testing

Test Cases: Created test cases for all major functionalities including:

-Student Enquiry creation and validation rules

-Follow-Up task automation

-Email alerts for pending follow-ups

-Conversion trigger from enquiry to student record

Sample Test Table:

Conclusion

The Student Enquiry Management System project demonstrates the complete lifecycle of a Salesforce implementation, starting from problem understanding to testing and deployment. By dividing the project into multiple phases, we were able to cover both administrative and developmental aspects of Salesforce, ensuring a well-rounded learning experience.

Key highlights include:

Data Modeling & Relationships: Designed standard and custom objects such as Student Enquiry and Student, with proper fields, record types, and relationships to support the use case.

Process Automation: Implemented validation rules, flows, and triggers to automate repetitive tasks such as enquiry conversion and student creation.

User Interface Enhancements: Configured record pages, home page layouts, and list views for a user-friendly experience.

Integration & Deployment: Used named credentials and change sets to prepare the system for real-world extensibility and migration.

Reports & Dashboards: Enabled stakeholders to track enquiries, conversions, and follow-ups effectively with interactive charts and reports.

Quality Assurance Testing: Validated every automation and trigger through systematic test cases and Apex test classes, ensuring reliability.

Overall, the project illustrates how Salesforce can streamline enquiry-to-admission processes in an educational setup by improving efficiency, data accuracy, and decision-making through automation and analytics.

For future enhancements, this system can be extended with AI-driven lead scoring, chatbot integration, and Einstein Analytics to make the enquiry process even smarter and more predictive.