

# Institute Management System

---

## 1. Project Overview

The Institute Management System is a comprehensive Salesforce solution designed to efficiently manage student records, course offerings, enrollments, and attendance tracking. This system automates repetitive tasks, ensures data integrity, and provides meaningful insights through dynamic reports and dashboards. By leveraging Salesforce's automation and security features, the system delivers an intuitive and secure platform for administrators to manage educational workflows.

### Key Highlights:

- Centralized management of Students, Courses, Enrollments, and Attendance
  - Automation of attendance creation using Record-Triggered Flows
  - Validation rules to prevent duplicate enrollments
  - Interactive dashboards and detailed reports for analytics
  - Role-based access control and field-level security for data protection
- 

## 2. Project Objectives

The primary objectives of this project are:

1. Streamline the management of student and course information.
  2. Automate attendance tracking and reduce manual errors.
  3. Provide real-time insights through reports and dashboards.
  4. Enforce secure, role-based access to sensitive data.
  5. Ensure consistency, accuracy, and maintainability of records.
-

## **Phase 1: Problem Understanding & Industry Analysis**

### **1. Requirement Gathering**

Institute Management require a system to manage student details, course enrollments, attendance, grades, and fees in a centralized manner. The requirements were gathered from administrators, teachers, and students to identify challenges in manual management.

### **2. Stakeholder Analysis**

- **Students** – Need easy enrollment, grade tracking, and fee updates.
- **Teachers** – Need to record attendance, manage grades, and monitor performance.
- **Administrators** – Require centralized control over student data, reporting, and fee collection.
- **Parents/Guardians (Future)** – Require visibility into student performance and attendance.

### **3. Business Process Mapping**

- **Student Admission** → Creation of student record.
- **Course Enrollment** → Mapping students to multiple courses.
- **Attendance Tracking** → Daily attendance marking per course.
- **Grade Management** → Entering marks and calculating GPA.
- **Fee Management** → Recording payments, balances, and due alerts.
- **Reporting & Analytics** → Generating performance, fee, and attendance reports.

### **4. Industry-specific Use Case Analysis**

- Current systems in schools and colleges rely on **spreadsheets/manual records**, causing redundancy and errors.
- Industry trend is shifting to **cloud-based CRMs** like Salesforce for data accuracy and accessibility.
- Similar use cases:
- University student portals.
- Training centers managing course enrollments.
- Coaching institutes tracking performance.

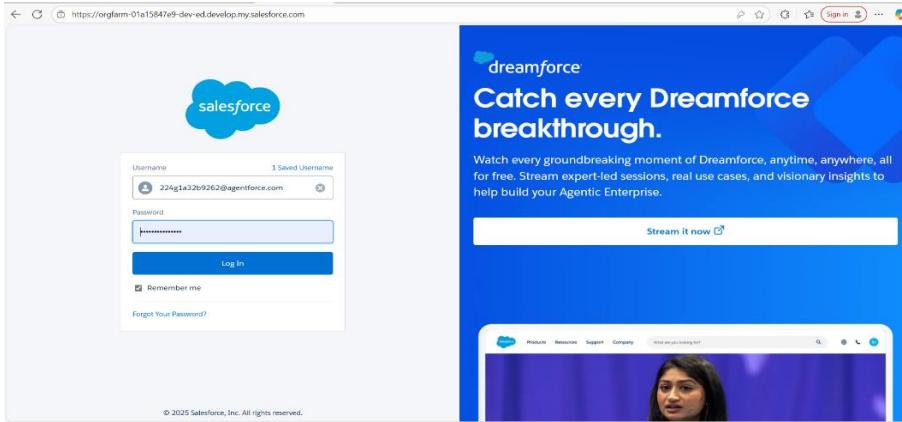
## 5. AppExchange Exploration

- Explored Salesforce **AppExchange** solutions for education management.
- Found apps like:
  - **Salesforce Education Cloud** – Comprehensive education CRM.
  - **Attendance & Fee Management Apps** – Useful but often paid solutions.
- Decided to **build a custom Institute Management System** tailored for specific institutional needs instead of relying fully on prebuilt paid apps.

## Phase 2: Org Setup & Configuration – Institute Management System

### 1. Salesforce Edition

- Developer Edition (Dev Org) used for project setup.



### 2. Company Profile Setup

- Configured Institution Name, Address, Default Time Zone, Currency, and Locale.

A screenshot of the Salesforce Setup interface under 'Company Settings'. The left sidebar shows 'Company Information' is selected. The main area is titled 'Company Information' and displays the organization's name: 'SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY'. It includes sections for 'Organization Detail' (Primary Contact: OrgAdmin RNC), 'Phone' (Default Locale: English (India)), and various configuration options like 'API Requests, Last 24 Hours' (0.15,000 max). The page also shows 'Created By: OrgAdmin RNC' and 'Modified By: GenerateVishwanath' with timestamps.

### 3. Business Hours & Holidays

- Defined Standard Business Hours (9 AM – 6 PM, Mon–Sat).

A screenshot of the Salesforce Setup interface under 'Business Hours'. The left sidebar shows 'Business Hours' is selected. The main area is titled 'Business Hours' and shows 'Organization Business Hours'. It details the 'Business Hours Detail' for 'Business Hours Name: Institution Standard Business Hours' and 'Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)'. The 'Active' checkbox is checked. The 'Last Modified By: GenerateVishwanath' timestamp is shown as '10/19/2025, 8:21 AM'. Below this is a 'Holidays' section which is currently empty.

- Added **holidays** (e.g., Independence Day, New Year, Republic Day).

The screenshot shows the 'Holidays' setup page in Salesforce. It lists two entries:

Action	Holiday Name	Description	Date and Time
Link	Independence Day		8/15/2025 All Day
Link	New Year		1/1/2025 All Day
Edit	Independence Day		
Edit	New Year		
Delete	Independence Day		
Delete	New Year		

## 4. Fiscal Year Settings

- Configured **Standard Fiscal Year** (Jan–Dec).

The screenshot shows the 'Organization Fiscal Year Edit' page for SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY. It is set to 'Standard Fiscal Year'. The 'Change Fiscal Year Period' section is open, showing the fiscal year starts in January and ends in December.

Name	SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
Fiscal Year Start Month	January
Fiscal Year Is Based On	<input checked="" type="radio"/> The ending month

## 5. User Setup & Licenses

- Created sample users with **Salesforce licenses**:
- System Administrator** – Full access.
- Faculty User** – Limited teaching access.
- Student User** – Read-only/student data access.

## 6. Profiles

- Created/Modified Profiles:
- **System Administrator** (default full access).

**Profile Detail**

Name	User License	Created By	Custom Profile	Modified By
System Administrator	Salesforce	salesforce.com, Inc.		Gavala Vyshnavi 10/17/2025, 8:34 PM

**Page Layouts**

Object Type	Standard Object Layout	Location Group Assignment
Lead	Global [View Assignment]	[View Assignment]
Account	Not Assigned [View Assignment]	Macro [View Assignment]
Opportunity	Object Milestone Layout [View Assignment]	Operating Hours Layout [View Assignment]
Appointment	Appointment Layout [View Assignment]	Opportunity Product Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Opportunity Quota Layout [View Assignment]

- **Teacher Profile** – Access to Students, Courses, Attendance, Grades.

**Profile Detail**

Name	User License	Created By	Custom Profile	Modified By
Teacher Profile	Salesforce	Gavala Vyshnavi		Gavala Vyshnavi 10/18/2025, 8:07 PM

**Page Layouts**

Object Type	Standard Object Layout	Location Group Assignment
Lead	Global [View Assignment]	[View Assignment]
Account	Not Assigned [View Assignment]	Macro [View Assignment]
Opportunity	Object Milestone Layout [View Assignment]	Operating Hours Layout [View Assignment]
Appointment	Appointment Layout [View Assignment]	Opportunity Product Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Opportunity Quota Layout [View Assignment]

- **Student Profile** : To check the student details in the read only mode.

**Profile Detail**

Name	User License	Created By	Custom Profile	Modified By
Student Profile	Read Only	Gavala Vyshnavi		Gavala Vyshnavi 10/18/2025, 8:09 PM

**Page Layouts**

Object Type	Standard Object Layout	Location Group Assignment
Lead	Global [View Assignment]	[View Assignment]
Account	Not Assigned [View Assignment]	Macro [View Assignment]
Opportunity	Object Milestone Layout [View Assignment]	Operating Hours Layout [View Assignment]
Appointment	Appointment Layout [View Assignment]	Opportunity Product Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Opportunity Quota Layout [View Assignment]

- **Finance Profile – Access to Fees object.**

The screenshot shows the Salesforce Setup interface under the 'Profiles' section. A specific profile, 'Finance Profile', is selected. The page lists several permissions such as 'Login IP Range', 'Enabled Apex Class Access', 'Enabled Visualforce Page Access', etc. It also shows 'Page Layouts' for various objects like Global, Home Page, Account, Alternative Payment Method, and Appointment Invitation. Each layout has a 'View Assignment' link next to it.

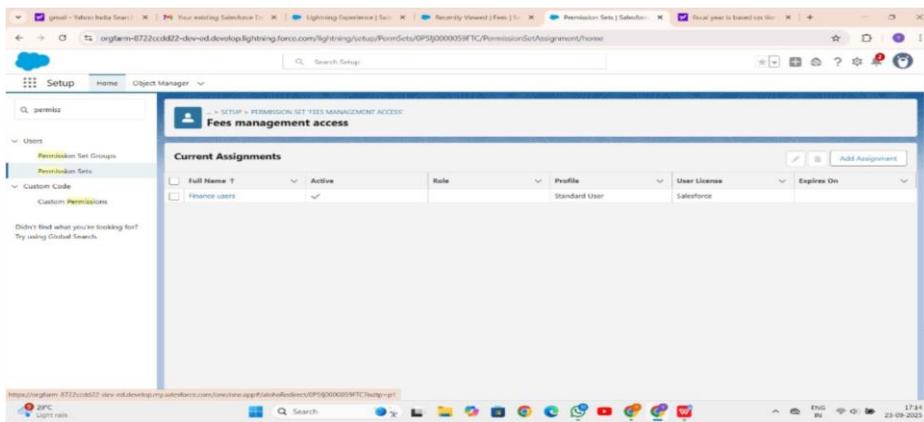
## 7. Roles

- Defined Role Hierarchy:
  - **Principal (Top Role)**
    - HOD
    - Finance Officer
    - Teacher

The screenshot shows the Salesforce Setup interface under the 'Roles' section. It displays a hierarchical tree of roles for 'SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY'. The root node is 'SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY', which branches into 'CEO', 'Finance Officer', 'HOD', and 'Teachers'. Each role node has an 'Edit' button next to it.

## 8. Permission Sets

- **Created Permission Sets for additional access:**
  - **Report Access** – Extra reporting permissions.
  - **Fee Management Access** – Extended access to Fees object.



## 9. Objects Created

- Student (Custom Object)** – Manage student details.
- Course (Custom Object)** – Store course information.
- Enrollment (Junction Object)** – Relationship between Student & Course.
- Attendance (Custom Object)** – Track daily student attendance.
- Grades (Custom Object)** – Store marks, grades, GPA.
- Fees (Custom Object)** – Manage payments and pending balances.

<https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01II000002tTHF/Details/view>

Setup > Object Manager  
**Enrollment**

**Details**

Description

API Name: **Enrollment\_c**  
Custom ✓  
Singular Label: Enrollment  
Plural Label: Enrollments

Enable Reports  
Track Activities  
Track Field History  
Deployment Status  
Deployed  
Help Settings  
Standard Salesforce.com Help Window

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Restriction Rules  
Scoping Rules  
Object Access  
Triggers

<https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01II000002z/kP/Details/view>

Setup > Object Manager  
**Attendance**

**Details**

Description

API Name: **Attendance\_c**  
Custom ✓  
Singular Label: Attendance  
Plural Label: Attendances

Enable Reports  
✓  
Track Activities  
Track Field History  
Deployment Status  
Deployed  
Help Settings  
Standard Salesforce.com Help Window

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout  
Restriction Rules  
Scoping Rules

<https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01II000003Vs4P/Details/view>

Setup > Object Manager  
**Grade**

**Details**

Description

API Name: **Grade\_c**  
Custom ✓  
Singular Label: Grade  
Plural Label: Grades

Enable Reports  
✓  
Track Activities  
Track Field History  
Deployment Status  
Deployed  
Help Settings  
Standard Salesforce.com Help Window

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout  
Restriction Rules  
Scoping Rules

<https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01II000003Vw61/Details/view>

Setup > Object Manager  
**Fee**

**Details**

Description

API Name: **Fee\_c**  
Custom ✓  
Singular Label: Fee  
Plural Label: Fees

Enable Reports  
✓  
Track Activities  
Track Field History  
Deployment Status  
Deployed  
Help Settings  
Standard Salesforce.com Help Window

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout  
Restriction Rules  
Scoping Rules

## 10. OWD (Org-Wide Defaults)

- **Student** → Private
- **Course** → Public Read Only
- **Enrollment** → Private
- **Attendance** → Private
- **Grades** → Private
- **Fee** → Private

The screenshot shows the 'Sharing Settings' page in the Salesforce Setup. The left sidebar has 'Sharing' selected under 'Security'. The main area displays a grid of objects and their sharing defaults. Most objects have 'Private' selected as the default sharing level. 'Attendance' is set to 'Public Read Only'. 'Grade' is set to 'Public ReadWrite'. 'Fee' is set to 'Private'. 'Work Type Group' is set to 'Public ReadWrite'. Other objects like 'Student' and 'Teacher' also have 'Private' selected. At the bottom, there are sections for 'Other Settings' including 'Standard Report Visibility', 'Manual User Record Sharing', 'Manager Groups', 'Secure guest user record access', and 'Require permission to view record names in lookup fields'.

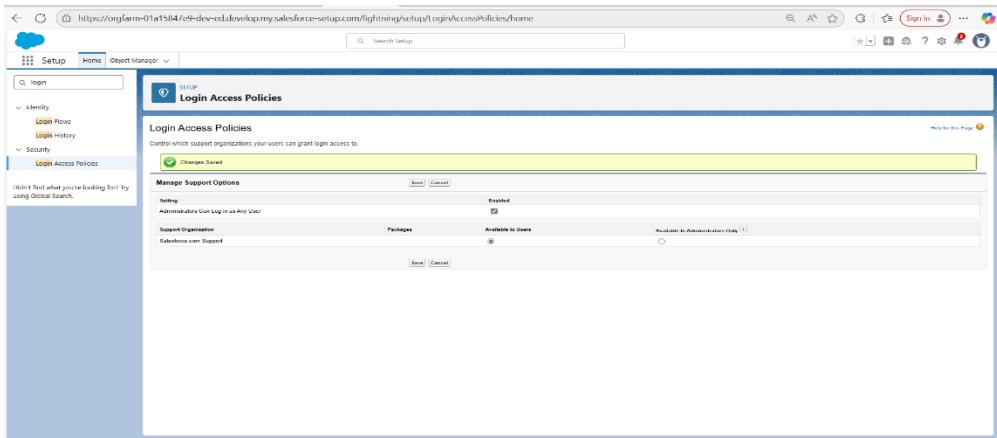
## 11. Sharing Rules

- Shared **Courses** with all Faculty.
- Shared **Student Records** read-only with Faculty (except Finance objects).
- Finance users granted access to **Fees object**.

The screenshot shows the 'Sharing Rules' page in the Salesforce Setup. The left sidebar has 'Sharing Rules' selected under 'Sharing settings'. The main area lists several sharing rule categories: 'Attendance Sharing Rules', 'Course Sharing Rules', 'Enrollment Sharing Rules', 'Fee Sharing Rules', 'Grade Sharing Rules', 'Status Sharing Rules', and 'Student Sharing Rules'. Each category has a 'New' button and a 'Recycle Bin' button. Under each category, there are rows for different actions. For example, in 'Attendance Sharing Rules', there is a row for 'Criteria' where 'Owner in All Internal Users' is mapped to 'Role: Teachers' with 'ReadWrite' access. In 'Fee Sharing Rules', there is a row for 'Criteria' where 'Owner in All Internal Users' is mapped to 'Role: Finance Officer' with 'ReadWrite' access. Other categories show similar mappings for different roles like 'Role: Teachers' and 'Role: Admin'.

## 12. Login Access Policies

- Enabled **Administrators Can Log in as Any User**.
- Configured **Trusted IP Ranges** for secure login.



## 13. Dev Org Setup

- Enabled **Dev Hub** in Developer Org.
- Connected CLI:  
sf org login web --set-default-dev-hub --alias DevHub

## 14. Sandbox Usage

- **Not available** in Developer Edition.
- Documented sandbox usage for real-world projects (Dev, Test, UAT).

## 15. Deployment Basics

- Practiced deployment commands:
- sf project retrieve start --manifest manifest/package.xml -o DevHub
- sf project deploy start --target-org ScratchOrg
- Source code and configuration stored in **GitHub repo** for version control.

# Phase 3: Data Modeling & Relationships

## 1. Standard & Custom Objects

In this project, we used standard objects like Account and Contact, and created custom objects such as Student, Course, and Enrollment.

Custom objects were designed to represent our business data model, where Students can enroll in multiple Courses through the Enrollment object.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. A search bar at the top right contains 'Search Setup'. Below it, a navigation bar has 'Setup' and 'Object Manager' selected. The main area is titled 'SETUP > OBJECT MANAGER' and shows 'Student' as the selected object. On the left is a sidebar with various tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The 'Details' tab is selected. The main content area displays the following details for the 'Student' object:

Details	
Description	
API Name	Student_c
Custom	✓
Singular Label	Student
Plural Label	Students
Enable Reports	✓
Track Activities	✓
Task Field History	✓
Deployment Status	In Development
Help Settings	Standard salesforce.com Help Window

At the bottom right of the main area are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. A search bar at the top right contains 'Search Setup'. Below it, a navigation bar has 'Setup' and 'Object Manager' selected. The main area is titled 'SETUP > OBJECT MANAGER' and shows 'Course' as the selected object. On the left is a sidebar with various tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, and Triggers. The 'Details' tab is selected. The main content area displays the following details for the 'Course' object:

Details	
Description	
API Name	Course_c
Custom	✓
Singular Label	Course
Plural Label	Courses
Enable Reports	
Track Activities	
Task Field History	
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

At the bottom right of the main area are 'Edit' and 'Delete' buttons.

https://orgfarm-01a1584/e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01lJ00000z1Hr/Details/view

The screenshot shows the Salesforce Setup interface for the 'Object Manager' section. The object being viewed is 'Enrollment'. The left sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, and Triggers. The main details pane shows the following fields:

Field	Value
Description	
API Name	Enrollment_c
Custom	✓
Singular Label	Enrollment
Plural Label	Enrollments
Enable Reports	
Track Activities	
Track Field History	
Deployment Status	
Deployed	
Help Settings	
Standard Salesforce.com Help Window	

Buttons at the top right of the details pane are 'Edit' and 'Delete'.

https://orgfarm-01a1584/e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01lJ00000z7kP/Details/view

The screenshot shows the Salesforce Setup interface for the 'Object Manager' section. The object being viewed is 'Attendance'. The left sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules, and a URL 'https://orgfarm-01a1584/e9-dev-ed.develop.my.salesforce-setup.com/one/one.app#setup...'. The main details pane shows the following fields:

Field	Value
Description	
API Name	Attendance_c
Custom	✓
Singular Label	Attendance
Plural Label	Attendances
Enable Reports	✓
Track Activities	
Track Field History	
Deployment Status	
Deployed	
Help Settings	
Standard Salesforce.com Help Window	

Buttons at the top right of the details pane are 'Edit' and 'Delete'.

https://orgfarm-01a1584/e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01lJ000003w4P/Details/view

The screenshot shows the Salesforce Setup interface for the 'Object Manager' section. The object being viewed is 'Grade'. The left sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main details pane shows the following fields:

Field	Value
Description	
API Name	Grade_c
Custom	✓
Singular Label	Grade
Plural Label	Grades
Enable Reports	✓
Track Activities	
Track Field History	
Deployment Status	
Deployed	
Help Settings	
Standard Salesforce.com Help Window	

Buttons at the top right of the details pane are 'Edit' and 'Delete'.

The screenshot shows the Salesforce Object Manager interface for the 'Fee' object. The 'Details' tab is selected. On the left, a sidebar lists various setup options like Page Layouts, Buttons, and Record Types. The main pane displays the object's details, including its API name 'Fee\_\_c' and singular/plural labels 'Fee' and 'Fees' respectively.

## 2. Fields

Custom fields were added to capture additional data for each object. Examples include:

- Student: Student Name, Last Name, Email, Phone, Department (Picklist), Student Id, Schoolership, Status(Picklist).
- Course: Course Name, Course Code, Duration. –
- Enrollment: Enrollment Date, Status.
- Attendance: Date, Attendance Name.
- Fee: Fee Id, Mode Of Transaction, Payment date, Payment Status.
- Grade: CGPA, Grade, Marks.

The screenshot shows the Salesforce Object Manager interface for the 'Student' object. The 'Fields & Relationships' tab is selected. The sidebar lists various setup options. The main pane displays a table of fields, each with its field label, name, data type, controlling field, and indexing status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Department	Department__c	Text		
Email	Email__c	Email		
Last Modified By	LastModifiedBy	Lookup(User)		
Last Name	Last_Name__c	Text(60)		
Owner	OwnerId	Lookup(User/Group)		
Phone	Phone__c	Phone		
Scholarship	Scholarship__c	Checkbox		
Status	Status__c	Picklist		
Student Id	Student_Id__c	Auto Number		
Student Name	Name	Text(60)		

<https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce.com/lightning/setup/ObjectManager/01fj000002rSh/FieldsAndRelationships/view>

SETUP > OBJECT MANAGER  
Course

Details Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course Code	Course_Code__c	Text(6)		✓
Course Name	Course_Name__c	Text(20)		✓
Created By	CreatedBy	Lookup(User)		✓
Grade	Grade__c	Number(4, 0)		✓
Duration	Duration__c	Number(13, 0)		✓
End Date	End_Date__c	Date		✓
Instructor	Instructor__c	Lookup(User)		✓
Last Modified By	LastModifiedBy	Lookup(User)		✓
Owner	OwnerId	Lookup(Owner Group)		✓
Start Date	Start_Date__c	Date		✓

<https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce.com/lightning/setup/ObjectManager/01fj000002rSh/FieldsAndRelationships/view>

SETUP > OBJECT MANAGER  
Enrollment

Details Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course	Course__c	Lookup(Course)		✓
Created By	CreatedBy	Lookup(User)		✓
Enrollment State	Enrollment_Status__c	Text		✓
Enrollment Name	Name	Text(20)		✓
Grade	Grade__c	Text(2)		✓
Last Modified By	LastModifiedBy	Lookup(User)		✓
Owner	OwnerId	Lookup(Owner/Group)		✓
Status	Status__c	Text		✓
Student	Student__c	Lookup(Student)		✓

<https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce.com/lightning/setup/ObjectManager/01fj000002rSh/FieldsAndRelationships/view>

SETUP > OBJECT MANAGER  
Attendance

Details Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Attendance Name	Name	Text(20)		✓
Course	Course__c	Lookup(Course)		✓
Created By	CreatedBy	Lookup(User)		✓
Date	Date__c	Date		✓
Enrollment	Enrollment__c	Lookup(Enrollment)		✓
Last Modified By	LastModifiedBy	Lookup(User)		✓
Owner	OwnerId	Lookup(Owner/Group)		✓
Status	Status__c	Text		✓
Student	Student__c	Lookup(Student)		✓

<https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce.com/lightning/setup/ObjectManager/01fj000002rSh/FieldsAndRelationships/view>

SETUP > OBJECT MANAGER  
Fee

Details Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		✓
Fee ID	Fee_Id__c	Auto Number		✓
Fee Name	Name	Text(20)		✓
Last Modified By	LastModifiedBy	Lookup(User)		✓
Mode Of Transaction	Mode_of_Transaction__c	Text		✓
Owner	OwnerId	Lookup(Owner/Group)		✓
Payment Date	Payment_Date__c	Date		✓
Payment Status	Payment_Status__c	Picklist		✓
Pending Balance	Pending_Balance__c	Currency(18, 0)		✓
Student	Student__c	Lookup(Student)		✓

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
CGPA	CGPA__c	Number(5, 2) (Unique)		✓
Course	Course__c	Lookup(Course)		✓
Created By	Createdbyid	Lookup(User)		✓
Grade	Grade__c	Picklist		✓
Grade Name	Name	Text(20)		✓
Last Modified By	LastModifiedbyid	Lookup(User)		✓
Marks	Marks__c	Number(5, 2)		✓
Owner	Owner	Lookup(User/Group)		✓
Student	Student__c	Lookup(Student)		✓

### 3. Record Types

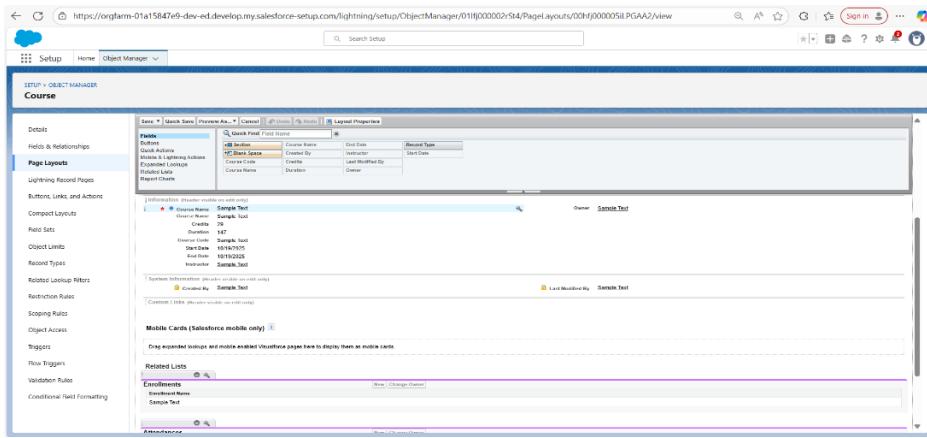
- Record types were created on the Course object to differentiate between Online and Offline courses. Each record type uses a specific page layout to show relevant fields.

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Offline	For Offline Course	✓	Gavala Vysheen, 10/19/2022, 7:52 AM
Online	For Online Course	✓	Gavala Vysheen, 10/19/2022, 12:21 AM

### 4. Page Layouts

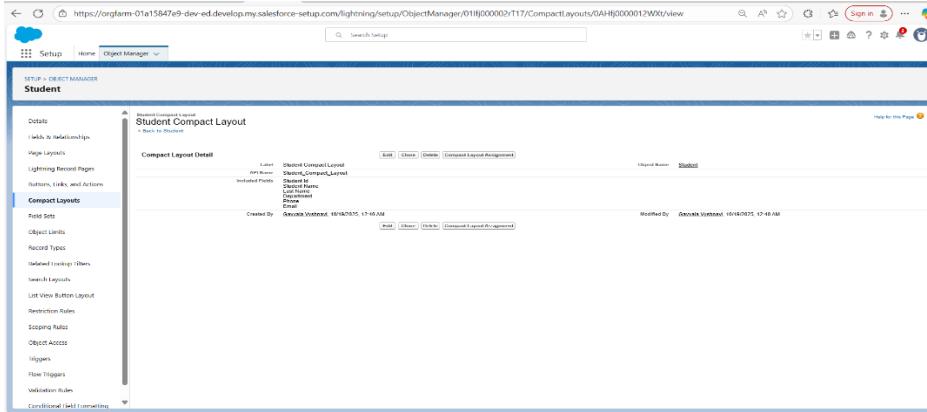
- Page layouts were customized to organize fields and related lists in a user-friendly way. Different page layouts were assigned based on record type.

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Grade Layout	Gavala Vysheen, 10/19/2022, 9:29 PM	Gavala Vysheen, 10/19/2022, 12:16 AM



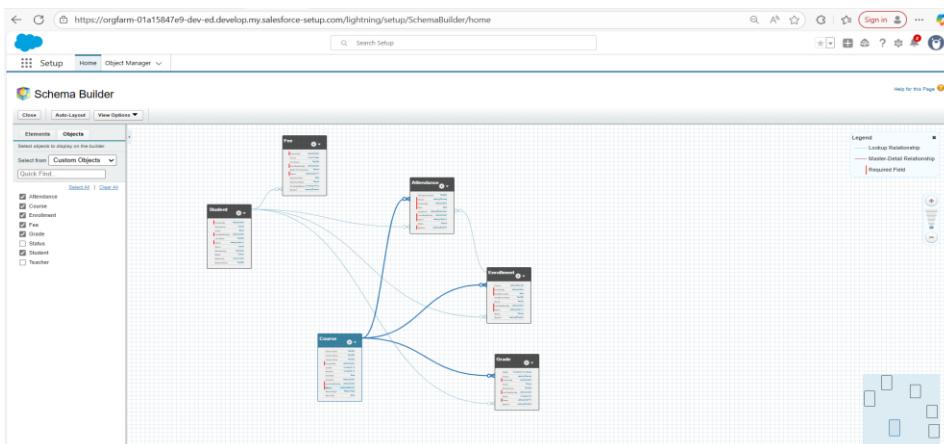
## 5. Compact Layouts

- Compact layouts were configured to display key information (Name, Email, Department, Phone for Student) in the highlights panel.



## 6. Schema Builder

- Schema Builder was used to visualize relationships between Student, Course, and Enrollment objects, making it easier to validate the data model.



## 7. Lookup vs Master-Detail vs Hierarchical Relationships

Relationships were created as follows:

- Student ↔ Course: Many-to-Many relationship implemented using the Enrollment junction object.
- Lookup relationships were used where child records can exist independently.
- Master-Detail relationships were used for Enrollment to ensure cascading delete behavior and roll-up summary fields.

## 8. Junction Objects

Enrollment object was created as a junction object with two master-detail fields:

- Student (Master-Detail)
- Course (Master-Detail)

## 9. External Objects

- An external object was configured to fetch reference data about available training materials from an external system, allowing real-time access without data duplication.

The screenshot shows the Salesforce Setup interface with the 'External Objects' page open. The left sidebar navigation includes 'Email', 'Send through External Email Services', 'Apps', 'External Client Apps' (with 'External Client App Manager', 'OAuth Usage', and 'Settings'), 'Integrations', 'External Data Sources', 'External Objects' (which is selected and highlighted in blue), and 'External Services'. The main content area has a title 'External Objects' with a sub-instruction: 'Use external objects to virtually represent external data as Salesforce objects. External objects map to a table in a data source outside the Salesforce org and enable access to that data via custom tabs and search. Each external object requires an external data source definition for connection details.' Below this is a table titled 'New External Object' with four rows:

Action	Label	Deployed	External Data Source	Description
Edit   Erase	Certification	✓	Resources	
Edit   Erase	Library Resource	✓	Resources	
Edit   Erase	Partner Course	✓	Resources	
Edit   Erase	Training material	✓	Resources	

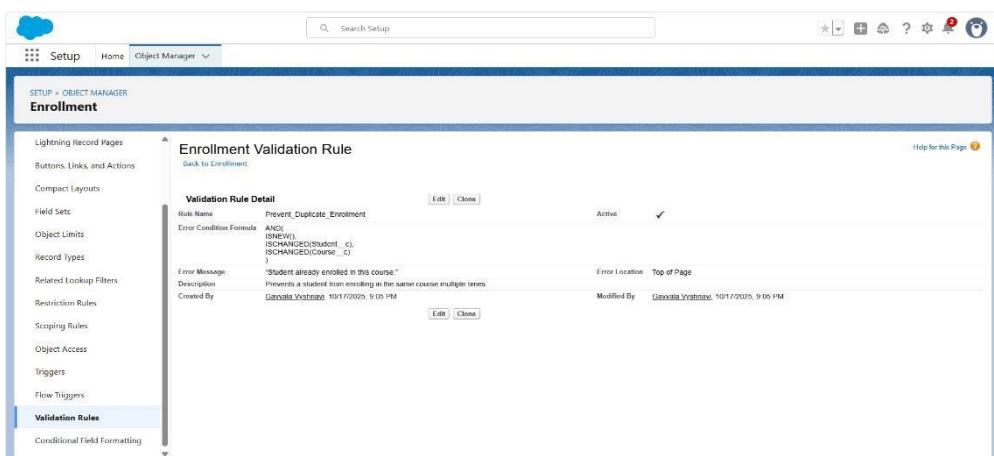
# Phase 4: Process Automation (Admin)

**Goal:** The goal of this phase is to automate repetitive administrative tasks in the **Institute Management System**. Using Salesforce automation tools, we ensure student records are accurate, status updates are automatic, reminders are sent on time, and approval requests are streamlined without manual effort.

## 1. Validation Rules

**Object:** Enrollment\_\_c

- **Rule:** Prevents duplicate enrollments for the same student and course.
- **Condition:** AND (ISNEW(), ISCHANGED(Student\_\_c), ISCHANGED(Course\_\_c))
- **Error Message:** “Student already enrolled in this course.”

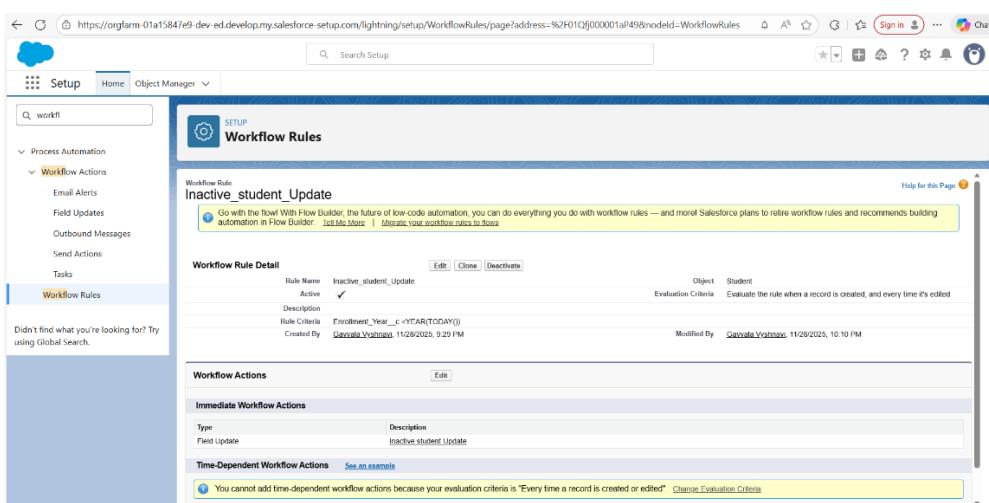


The screenshot shows the Salesforce Object Manager interface for the 'Enrollment' object. On the left, a sidebar lists various configuration options like Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, Triggers, Flow Triggers, Validation Rules, and Conditional Field Formatting. The 'Validation Rules' option is selected. In the main content area, a validation rule named 'Enrollment\_Duplicate\_Enrollment' is displayed. The rule has an active status and an error condition formula: AND (ISNEW(), ISCHANGED(Student\_\_c), ISCHANGED(Course\_\_c)). The error message is "Student already enrolled in this course." and it prevents a student from enrolling in the same course multiple times. The rule was created by 'Gavota\_Vyshnavi' on 10/17/2025 at 9:09 PM and modified by the same user on 10/17/2025 at 9:09 PM.

## 2. Workflow Rules

**Object:** Student\_\_c

- **Rule Name:** Inactive Student Update
- **Criteria:** Enrollment\_Year\_\_c < YEAR(TODAY())
- **Action:** Field Update → Status\_\_c = "Inactive"



The screenshot shows the Salesforce Setup page for 'Workflow Rules'. The sidebar on the left includes sections for Process Automation, Workflow Actions (Email Alerts, Field Updates, Outbound Messages, Send Actions, Tasks), and Workflow Rules. The 'Workflow Rules' section is selected. A specific workflow rule named 'Inactive\_student\_Update' is displayed. The rule has an active status and an evaluation criteria of 'Enrollment\_Year\_\_c < YEAR(TODAY())'. It was created by 'Gavota\_Vyshnavi' on 11/28/2025 at 9:29 PM and modified by the same user on 11/29/2025 at 10:10 PM. The rule triggers an immediate workflow action: a 'Field Update' where the 'Status\_\_c' field is set to 'Inactive'.

https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/WorkflowRules/page?address=%2F01Qj000001aPQj&modelId=WorkflowRules

## Workflow Rules

Workflow Rule **Send\_welcome\_email**

Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with workflow rules — and more! Salesforce plans to retire workflow rules and recommends building automation in Flow Builder. Tell Me More | Migrate your workflow rules to flows

**Workflow Rule Detail**

Rule Name	Send_welcome_email	Object	Student
Active	✓	Evaluation Criteria	Evaluate the rule when a record is created, and every time it's edited
Description			
Rule Criteria	TRUE	Created By	Gavalka Vyshnavi, 11/28/2025, 9:39 PM
		Modified By	Gavalka Vyshnavi, 11/28/2025, 9:43 PM

**Workflow Actions**

**Immediate Workflow Actions**

Type	Description
Email Alert	Welcome_email_student

**Time-Dependent Workflow Actions** See an example

You cannot add time-dependent workflow actions because your evaluation criteria is "Every time a record is created or edited". Change Evaluation Criteria

https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/WorkflowRules/page?address=%2F01Qj000001aP5&modelId=WorkflowRules

## Workflow Rules

Workflow Rule **fee\_payment\_remainder**

Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with workflow rules — and more! Salesforce plans to retire workflow rules and recommends building automation in Flow Builder. Tell Me More | Migrate your workflow rules to flows

**Workflow Rule Detail**

Rule Name	fee_payment_remainder	Object	Fee
Active	✓	Evaluation Criteria	Evaluate the rule when a record is created, and every time it's edited
Description			
Rule Criteria	Fee: Payment Status EQUALS Pending	Created By	Gavalka Vyshnavi, 11/28/2025, 10:01 PM
		Modified By	Gavalka Vyshnavi, 11/28/2025, 10:04 PM

**Workflow Actions**

**Immediate Workflow Actions**

Type	Description
Email Alert	fee_remainder_alert_to_student

**Time-Dependent Workflow Actions** See an example

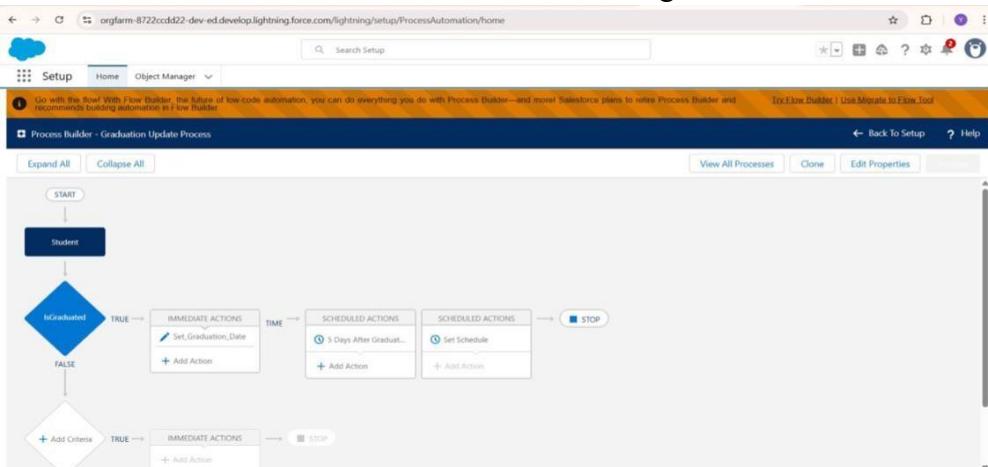
You cannot add time-dependent workflow actions because your evaluation criteria is "Every time a record is created or edited". Change Evaluation Criteria



### 3 .ProcessBuilder

**Object:** Student\_\_c

- **Process Name:** Graduation Update
- **Criteria:** When Status\_\_c = "Graduated"
- **Action 1:** Update Field → Graduation\_Date\_\_c = TODAY()
- **Action 2:** Send Email Alert → “Graduation Congratulations Email” to student.



**Object:** Student\_\_c

- **Process Name:** Scholarship Approval
- **Entry Criteria:** Scholarship\_Requested\_\_c = TRUE
- **Steps:**
  1. Initial Submission → Lock record.
  2. Assigned To → Finance Staff Queue.
  3. Final Approval → Unlock record + Update Scholarship\_Status\_\_c = "Approved".
  4. Rejection → Update Scholarship\_Status\_\_c = "Rejected".

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions   Edit	1	Step 1	Record Lock		Queue Finance Staff	Final Rejection
Action	Type		Description			
Edit	Record Lock		Unlock the record for editing			
Edit   Remove	Field Update		final_approval_scholarship			
Action	Type		Description			
Edit	Record Lock		Unlock the record for editing			
Edit   Remove	Field Update		final_rejection_scholarship			

### 5. Flow Builder

#### a) Screen Flow: Student Registration

- **Inputs:** First Name, Last Name, Email, Enrollment Year, Status.

- **Action:** Create Student record in Student \_\_c.

### b) Record-Triggered Flow: Auto Assign Status

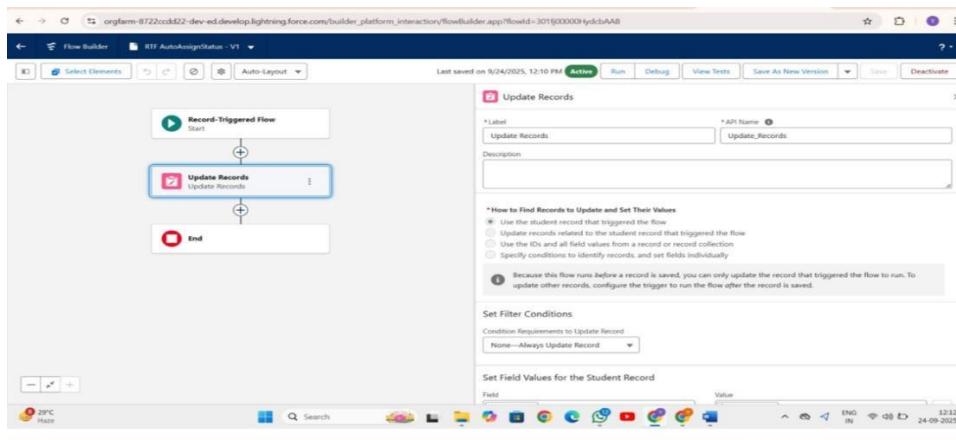
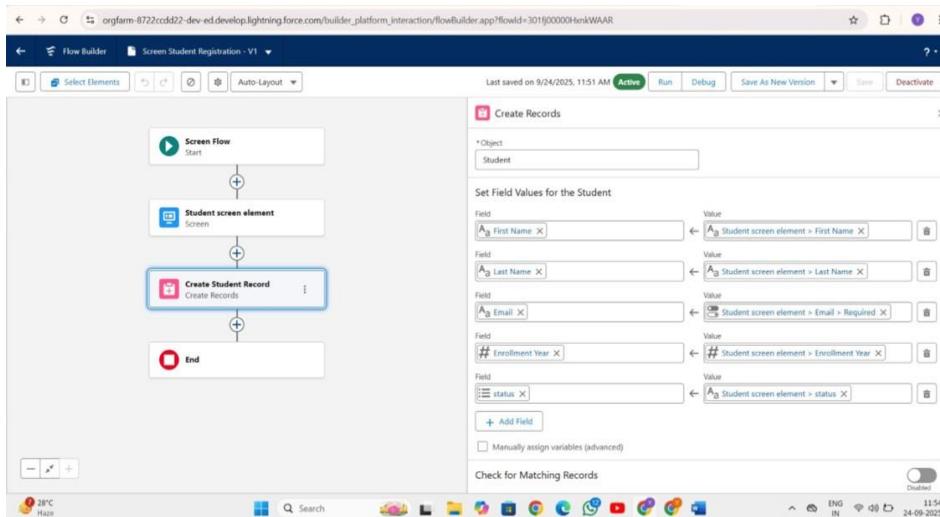
- **When:** A new Student record is created.
- **Condition:** If Enrollment\_Year\_\_c = T(YEAR(TODAY()))
- **Action:** Update Status\_\_c = "Enrolled".

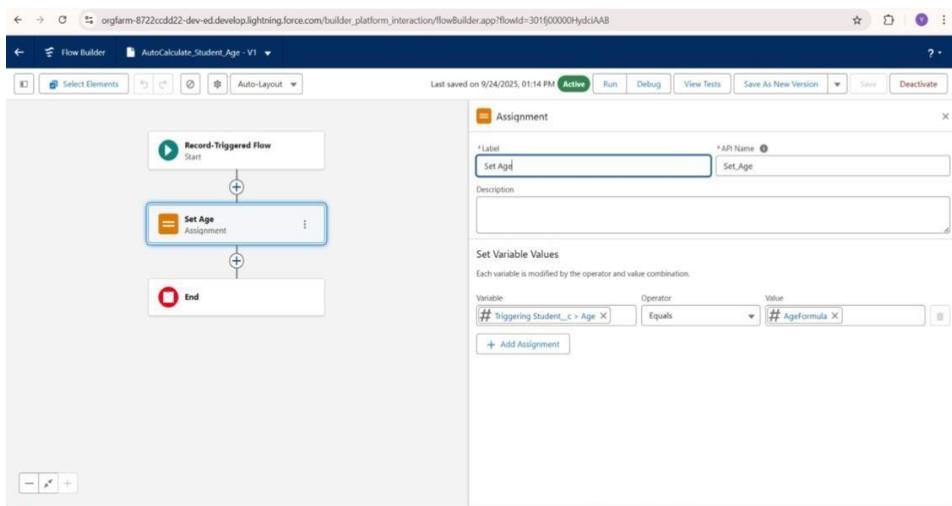
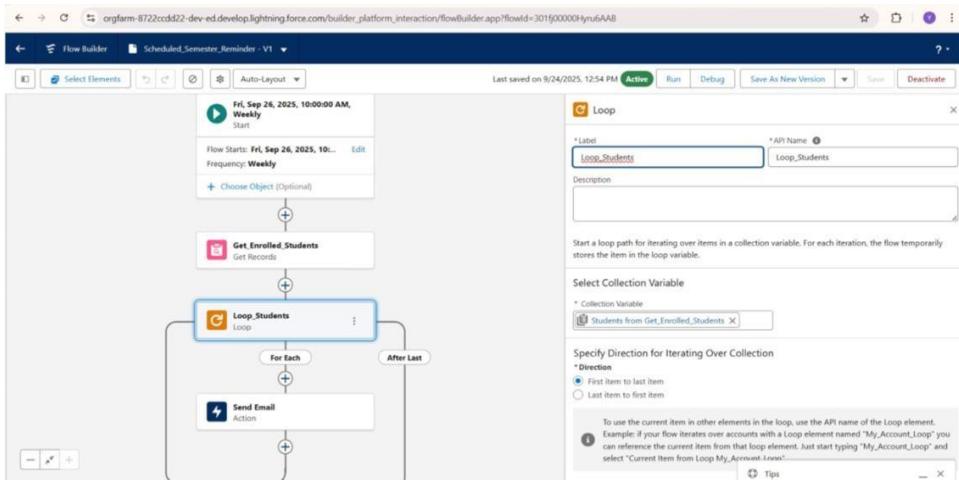
### c) Scheduled Flow: Semester Reminder

- **When:** Every semester start date (Scheduled path).
- **Action:** Send Email Alert to all Student \_\_c records with Status \_\_c = "Enrolled".

### d) Auto-Launched Flow: Calculate Student Age

- **Formula:** TODAY() - Date\_of\_Birth\_\_c
- **Action:** Update Age\_\_c field in Student\_\_c.





## 6. Email Alerts

**Object:** Student\_\_c

- **Templates Created:**
- Welcome Student Email (after registration).
- Fee remainder For Students.

The screenshot shows the Salesforce Setup interface with the 'Email Alerts' page selected. The alert is titled 'Welcome email student'. The 'Email Alert Detail' section includes fields for Description (Welcome\_email\_student), Unique Name (Welcome\_email\_student), From Email Address (Current User's email address), Recipients (User\_Impersonation User, User\_Security User, User\_Gavvalla.Vyshnavi, Email Field: Email), and Additional Emails. The 'Email Template' is set to 'Group Service Appointments Enrollment Confirmation Email' for the 'Student' object. The alert was created by Gavvalla Vyshnavi on 11/28/2025 at 9:42 PM and modified by the same user on the same date at 9:42 PM. Below the detail section are three tabs: 'Rules Using This Email Alert', 'Approval Processes Using This Email Alert', and 'Entitlement Processes Using This Email Alert'. The 'Rules Using This Email Alert' tab shows one rule named 'Send\_welcome\_email' with the action 'Edit | Del | Deactivate'. The 'Object' column is 'Student' and the 'Active' checkbox is checked. The 'Approval Processes Using This Email Alert' and 'Entitlement Processes Using This Email Alert' tabs both show a message stating 'This alert is currently not used by any approval processes' or 'entitlement processes'.

The screenshot shows the Salesforce Setup interface with the 'Email Alerts' page selected. The alert is titled 'fee\_remainder\_alert\_to\_student'. The 'Email Alert Detail' section includes fields for Description (fee\_remainder\_alert\_to\_student), Unique Name (fee\_remainder\_alert\_to\_student), From Email Address (Current User's email address), Recipients (Role and Internal Subordinates: Finance Officer, Role and Internal Subordinates: HO), and Additional Emails. The 'Email Template' is set to 'Scheduled Payments\_Payment Reminder for Service Appointment Email' for the 'Fee' object. The alert was created by Gavvalla Vyshnavi on 11/28/2025 at 10:04 PM and modified by the same user on the same date at 10:25 PM. Below the detail section are three tabs: 'Rules Using This Email Alert', 'Approval Processes Using This Email Alert', and 'Entitlement Processes Using This Email Alert'. The 'Rules Using This Email Alert' tab shows one rule named 'fee\_payment\_reminder' with the action 'Edit | Del | Deactivate'. The 'Object' column is 'Fee' and the 'Active' checkbox is checked. The 'Approval Processes Using This Email Alert' and 'Entitlement Processes Using This Email Alert' tabs both show a message stating 'This alert is currently not used by any approval processes' or 'entitlement processes'.

## 7. Field Updates

### Objects & Fields:

- Student\_\_c.Status\_\_c → "Inactive" (Workflow Rule).
- Student\_\_c.Graduation\_Date\_\_c → TODAY() (Process Builder).
- Student\_\_c.Age\_\_c → Formula result (Flow).

## 8. Tasks

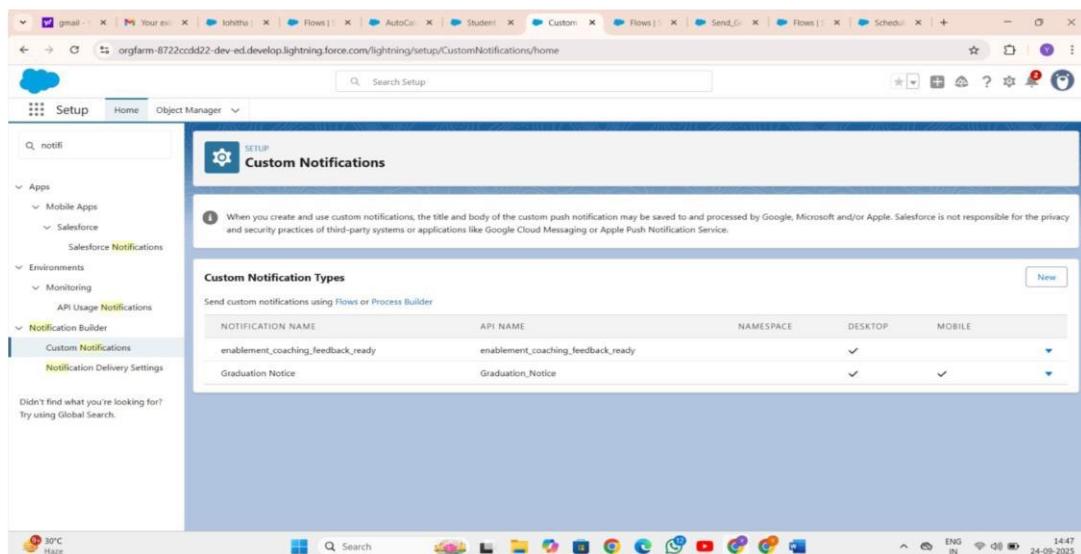
### Object: Student\_\_c

- **Example:** Checks whether a student is having low CGPA than the standard eligibility CGPA.

## 9. Custom Notifications

### Object: Student\_\_c

- **Example:** When a student is marked as “Graduated”, send a **Custom Notification** to the Advisor/Staff to initiate alumni registration.



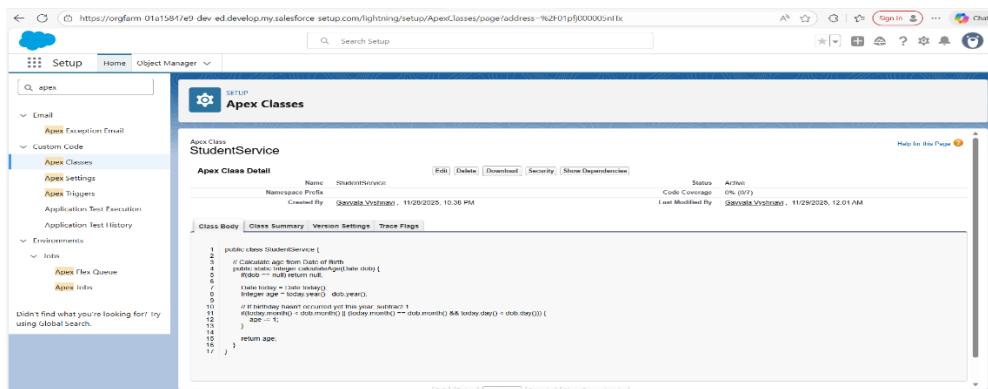
# Phase 5: Apex Programming (Developer)

## Goal of this Phase

The goal of Phase 5 was to implement **Apex programming features** in Salesforce to support the Student Management System. This included creating **classes, triggers, SOQL/SOSL queries, collections, asynchronous processing, exception handling, and test classes** to ensure the application is efficient, scalable, and reliable.

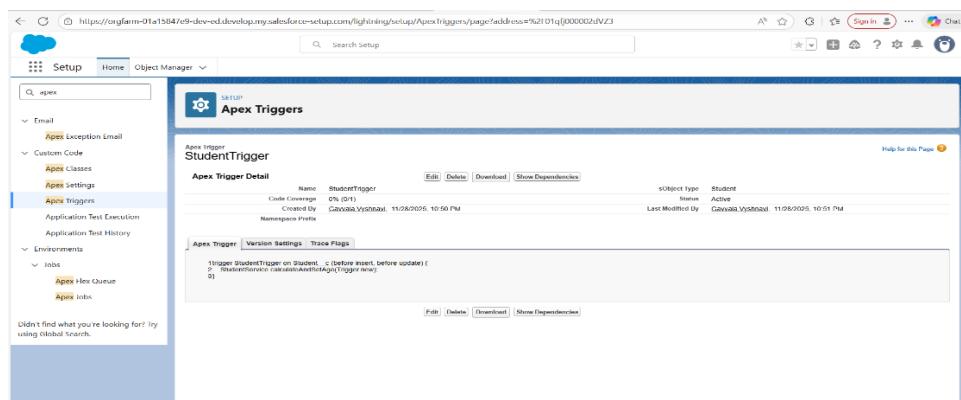
## 1. Classes & Objects

- Created **StudentService** Apex class to handle business logic.
- Added methods for calculating student age based on Date of Birth and updating records.



## 2. Apex Triggers (Before/After Insert/Update/Delete)

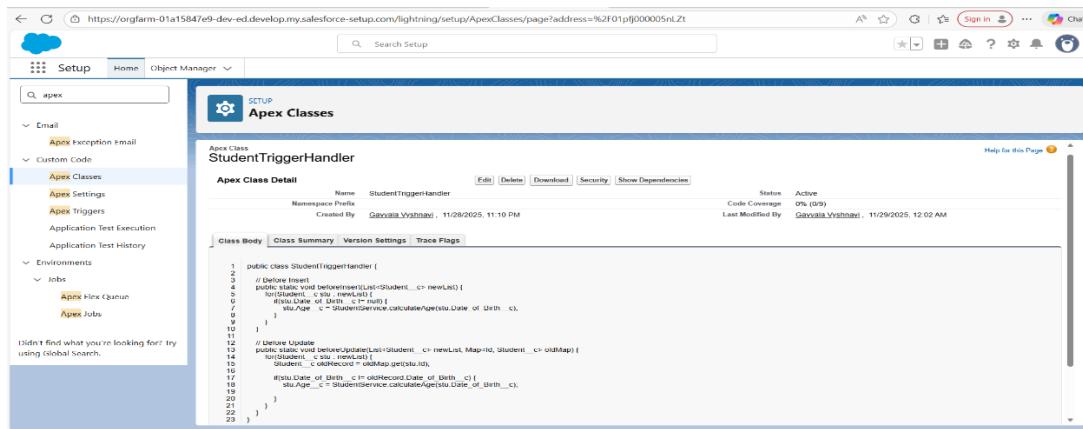
- Implemented a **StudentTrigger** to:
  - Calculate Age before insert/update when Date of Birth is set.
  - Prevent invalid updates using before update triggers.
  - Maintain consistency when student records are deleted.



### 3. Trigger Design Pattern

- Applied Handler Class Pattern:

- Trigger delegates logic to StudentTriggerHandler class.
- Ensures clean separation of logic, reusable code, and easier maintenance.

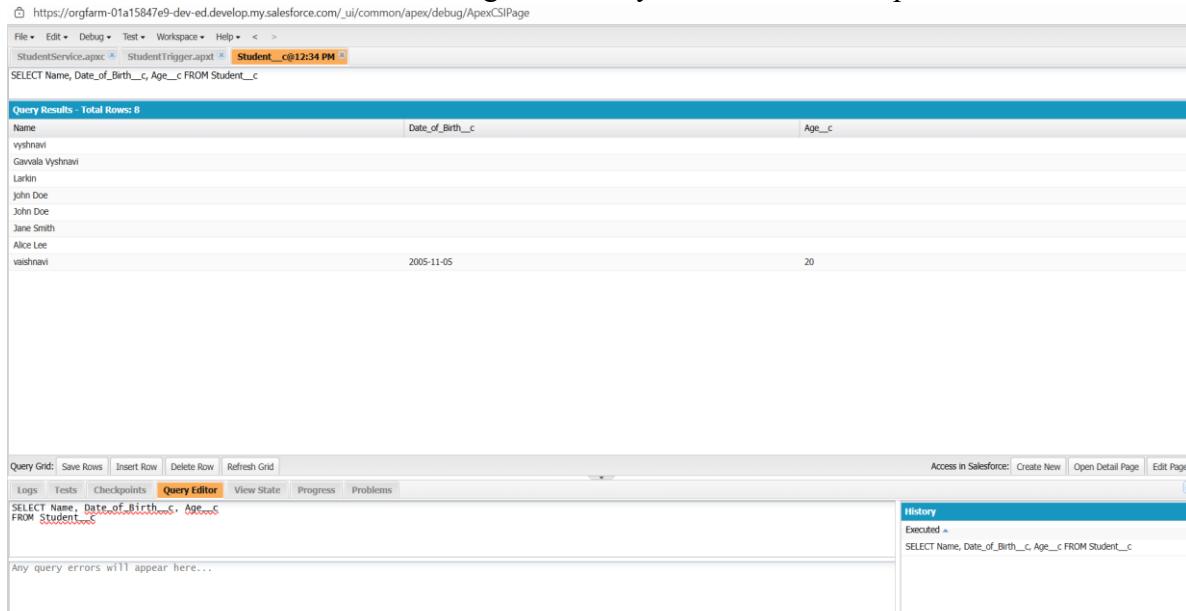


The screenshot shows the Salesforce Setup interface with the search bar set to "apex". The left sidebar has sections for Email, Custom Code, and Environments. Under Custom Code, Apex Classes is selected, showing sub-options like Apex Settings, Apex Triggers, Application Test Execution, and Application Test History. The main content area is titled "Apex Classes" and shows the "StudentTriggerHandler" class. The class body contains the following code:

```
1 public class StudentTriggerHandler {
2     // ...
3     public static void beforeInsert(List<Student__c> newList) {
4         for(Student__c stu : newList) {
5             stu.Date_of_Birth__c = null;
6             stu.Age__c = StudentService.calculateAge(stu.Date_of_Birth__c);
7         }
8     }
9 }
10 // Update
11 public static void beforeUpdate(List<Student__c> newList, Map<Id, Student__c> oldMap) {
12     for(Student__c stu : newList) {
13         if(stu.Date_of_Birth__c != null) {
14             stu.Age__c = StudentService.calculateAge(stu.Date_of_Birth__c);
15         }
16     }
17 }
18 public static void afterUpdate(List<Student__c> newList, Map<Id, Student__c> oldMap, Map<Id, Student__c> newMap) {
19     for(Student__c stu : newList) {
20         stu.Age__c = StudentService.calculateAge(stu.Date_of_Birth__c);
21     }
22 }
23 }
```

### 4. SOQL & SOSL

- Used **SOQL queries** to fetch Student records with fields like Id, Date\_of\_Birth\_\_c, and Age\_\_c.
- Demonstrated **SOSL** for searching students by name across multiple fields.



The screenshot shows the Salesforce Query Editor. At the top, it says "https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSPage". The tabs include Logs, Tests, Checkpoints, **Query Editor**, View State, Progress, and Problems. The Query Editor contains the following SOQL query:

```
SELECT Name, Date_of_Birth__c, Age__c FROM Student__c
```

The "Query Results - Total Rows: 8" section displays the following data:

Name	Date_of_Birth__c	Age__c
vyshnavi		
Gavala Vyshnavi		
Larkin		
John Doe		
John Smith		
Alice Lee		
vaishnavi	2005-11-05	20

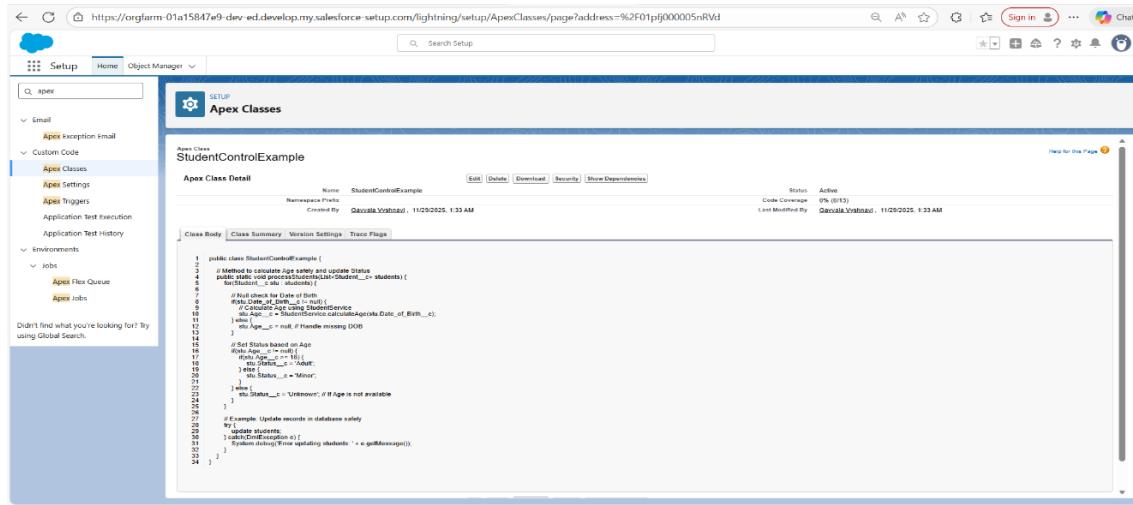
At the bottom, there is a note: "Any query errors will appear here..."

## 5. Collections (List, Set, Map)

- Used **List<Student\_\_c>** for batch updates.
- Applied **Set<Id>** to handle unique record IDs.
- Implemented **Map<Id, Student\_\_c>** to efficiently compare old and new trigger contexts.

## 6. Control Statements

- Used **If-Else conditions** for null checks.
- Applied **For loops** to process multiple student records.
- Incorporated **Try-Catch blocks** for exception handling.



The screenshot shows the Salesforce Setup Apex Classes page. The URL is https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce.com/lightning/setup/ApexClasses/page?address=%2F01p0j000005nRvD. The page title is "Apex Classes". On the left sidebar, under "Custom Code", "Apex Classes" is selected. The main content area displays the code for "StudentControlExample".

```
Apex Class Detail
Name: StudentControlExample
Namespace Prefix: 
Created By: Gavazla Vrashabha | 11/29/2025, 1:33 AM
Status: Active
Code Coverage: 9% (0/13)
Last Modified By: Gavazla Vrashabha | 11/29/2025, 1:33 AM

Class Body | Class Summary | Version Settings | Trace Flags

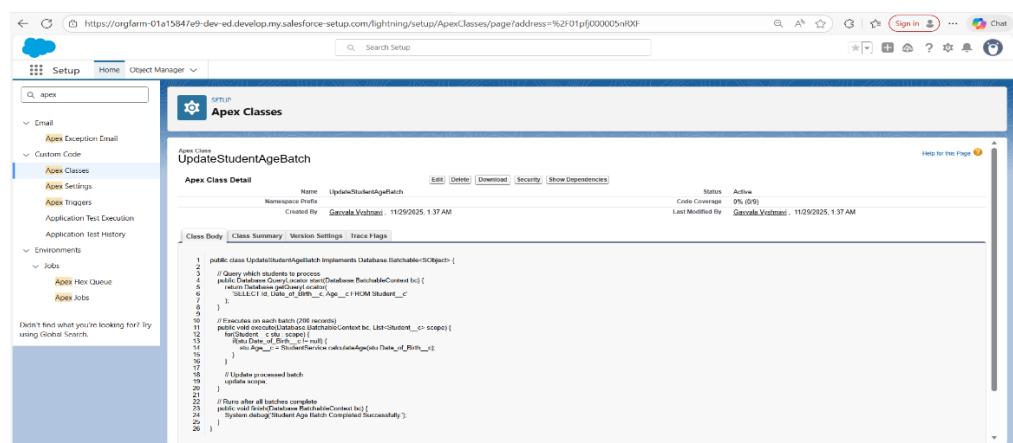
public class StudentControlExample {
    // Method to calculate Age and update Status
    public static void preProcessBatch(List<Student__c> students) {
        for (Student__c student : students) {
            if (student.Date_of_Birth__c == null) {
                StudentService.calculateAge(student);
            } else {
                student.Age__c = StudentService.calculateAge(student.Date_of_Birth__c);
            }
        }
    }

    // Set Status based on Age
    public static void postProcessBatch(List<Student__c> students) {
        for (Student__c student : students) {
            if (student.Age__c < null) {
                student.Status__c = 'Missing';
            } else if (student.Age__c == null) {
                student.Status__c = 'Unknown';
            } else {
                student.Status__c = 'Adult';
            }
        }
    }

    // Example: Update records in database safely
    public static void updateStudents() {
        List<Student__c> students = new List<Student__c>();
        System.debug('Before updating students: ' + students);
        try {
            Database.update(students, false);
        } catch (DatabaseException e) {
            System.debug('Error updating students: ' + e.getMessage());
        }
    }
}
```

## 7. Batch Apex

- Created **UpdateStudentAgeBatch** class implementing `Database.Batchable<SObject>`.
- Processes students in batches to calculate and update Age.
- Scheduled and monitored using **Apex Jobs**.



The screenshot shows the Salesforce Setup Apex Classes page. The URL is https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce.com/lightning/setup/ApexClasses/page?address=%2F01p0j000005nRxF. The page title is "Apex Classes". On the left sidebar, under "Custom Code", "Apex Classes" is selected. The main content area displays the code for "UpdateStudentAgeBatch".

```
Apex Class Detail
Name: UpdateStudentAgeBatch
Namespace Prefix: 
Created By: Gavazla Vrashabha | 11/29/2025, 1:37 AM
Status: Active
Code Coverage: 0% (0/0)
Last Modified By: Gavazla Vrashabha | 11/29/2025, 1:37 AM

Class Body | Class Summary | Version Settings | Trace Flags

public class UpdateStudentAgeBatch implements Database.Batchable<Student__c> {
    // Query
    public Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator('SELECT Id, Date_of_Birth__c, Age__c FROM Student__c');
    }

    // Process
    public void execute(Database.BatchableContext bc, List<Student__c> scope) {
        for (Student__c student : scope) {
            if (student.Date_of_Birth__c != null) {
                student.Age__c = StudentService.calculateAge(student.Date_of_Birth__c);
            }
        }
    }

    // Finish
    public void finish(Database.BatchableContext bc) {
        Database.update(scope);
    }
}

// Debug message after all batches complete
public void finish(Database.BatchableContext bc) {
    System.debug('Student Age Batch Completed Successfully');
}
```

## 8. Queueable Apex

- Developed **UpdateStudentAgeQueueable** class implementing Queueable.
- Allows background execution of student age updates.
- Can be chained for sequential execution.

The screenshot shows the Salesforce Setup Apex Classes page. The URL is https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ApexClasses/page?address=%2F01pfj000005nRyr. The page title is "Apex Classes". The left sidebar has a search bar and navigation links for Email, Custom Code, Apex Classes, Apex Settings, Apex Triggers, Application Test Execution, Application Test History, Environments, and Jobs. Under "Jobs", "Apex Flex Queue" and "Apex Jobs" are listed. The main content area shows the "Apex Class Detail" for "UpdateStudentAgeQueueable". The class body contains the following code:

```
1 public class UpdateStudentAgeQueueable implements Queueable {
2     public void execute(QueueableContext context) {
3         //Query all student records
4         List<Student__c> students = [
5             SELECT Id, Date_of_Birth__c, Age__c
6             FROM Student__c
7         ];
8
9         //Process each student
10        for(Student__c stu : students) {
11            if(stu.Date_of_Birth__c != null) {
12                stu.Age__c = StudentService.calculateAge(stu.Date_of_Birth__c);
13            }
14        }
15
16        //Update students
17        update students;
18
19        System.debug('Queueable Job Completed Successfully');
20    }
21 }
22 }
```

## 9. Scheduled Apex

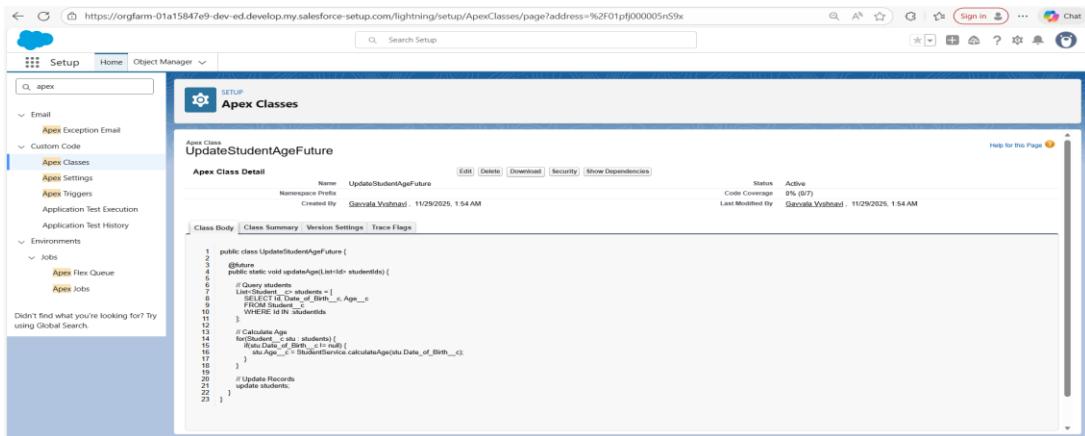
- Created **ScheduleUpdateStudentAgeBatch** class implementing Scheduledable.
- Scheduled batch jobs to run at predefined times.

The screenshot shows the Salesforce Setup Apex Classes page. The URL is https://orgfarm-01a15847e9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ApexClasses/page?address=%2F01pfj000005nS3V. The page title is "Apex Classes". The left sidebar is identical to the previous screenshot. The main content area shows the "Apex Class Detail" for "ScheduleUpdateStudentAgeBatch". The class body contains the following code:

```
1 public class ScheduleUpdateStudentAgeBatch implements Scheduledable {
2     public void execute(SchedulableContext sc) {
3         // Call your Batch Class
4         UpdateStudentAgeBatch b = new UpdateStudentAgeBatch();
5         Database.executeBatch(b, 200);
6     }
7 }
```

## 10. Future Methods

- Implemented **UpdateStudentAgeFuture** class with `@future` annotation.
- Runs background updates asynchronously for lightweight processing.



The screenshot shows the Salesforce Setup Apex Classes page. The search bar at the top has "apex" typed into it. On the left, there's a sidebar with sections like Email, Custom Code, Apex Classes, Apex Settings, Apex Triggers, Application Test Execution, Application Test History, Environments, and Jobs. The main area is titled "Apex Class Detail" for "UpdateStudentAgeFuture". It shows the class code:

```
1 public class UpdateStudentAgeFuture {
2     @future
3     public static void updateAge(List<Student__c> students) {
4         List<Student__c> students = [
5             SELECT Age__c, Date_of_Birth__c FROM Student__c
6             WHERE Id IN :ids
7         ];
8
9         // Calculate Age
10        for(Student__c student : students) {
11            if(student.Date_of_Birth__c != null) {
12                student.Age__c = StudentService.calculateAge(student.Date_of_Birth__c);
13            }
14        }
15
16        // Update Records
17        update students;
18    }
19 }
```

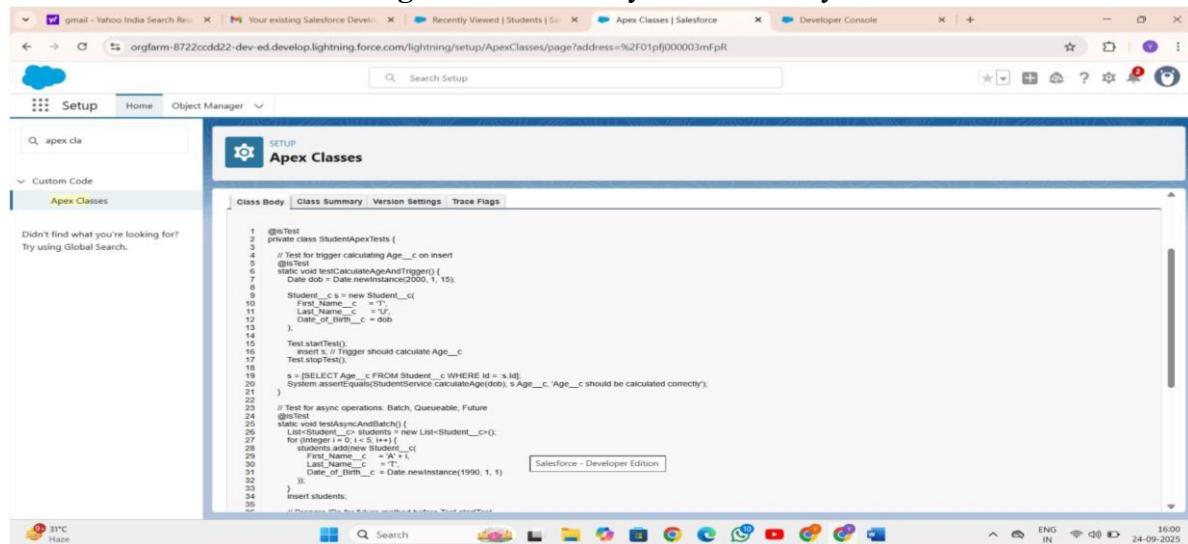
Below the code, there are tabs for Class Summary, Version Settings, and Trace Flags. At the bottom right, there are buttons for Edit, Delete, Download, Security, and Show Dependencies. The status bar indicates "Status: Active", "Code Coverage: 0% (0/7)", and "Last Modified By: Gacala.Vishnud - 11/29/2025, 1:54 AM".

## 11 .ExceptionHandling

- Wrapped logic in **try-catch** blocks.
- Handled null Date of Birth scenario to prevent runtime errors.
- Ensured no unhandled exceptions crash the system.

## 12. Test Classes

- Created **TestStudentService** and other test classes with `@isTest`.
- Verified Batch, Queueable, Future, and Exception Handling methods.
- Achieved **code coverage** and validated system reliability.



The screenshot shows the Salesforce Setup Apex Classes page. The search bar at the top has "apex cla" typed into it. On the left, there's a sidebar with sections like Custom Code, Apex Classes, Apex Settings, Apex Triggers, Application Test Execution, Application Test History, Environments, and Jobs. The main area is titled "Class Body" for "StudentApexTests". It shows the class code:

```
1 @isTest
2 private class StudentApexTests {
3
4     // Test for trigger calculating Age__c on insert
5     @isTest
6     static void testCalculateAgeAndTrigger() {
7         Date dob = Date.newInstance(2000, 1, 15);
8
9         Student__c s = new Student__c();
10        s.Last_Name__c = 'V';
11        s.Date_of_Birth__c = dob;
12
13        Test.startTest();
14        insert s; // Trigger should calculate Age__c
15        Test.stopTest();
16
17        s = [SELECT Age__c FROM Student__c WHERE Id = :s.Id];
18        System.assertEqual(StudentService.calculateAge(dob), s.Age__c, 'Age__c should be calculated correctly');
19    }
20
21
22    // Test for async operations: Batch, Queueable, Future
23    @isTest
24    static void testAsyncOperations() {
25        List<Student__c> students = new List<Student__c>();
26
27        for(Integer i = 0; i < 1000; i++) {
28            students.add(new Student__c(
29                Last_Name__c = 'A' + i,
30                Last_Name__c = 'B' + i,
31                Date_of_Birth__c = Date.newInstance(1990, 1, 1)
32            ));
33        }
34        insert students;
35    }
36}
```

Below the code, there are tabs for Class Summary, Version Settings, and Trace Flags. At the bottom right, there are buttons for Edit, Delete, Download, Security, and Show Dependencies. The status bar indicates "Salesforce - Developer Edition".

### **13. Asynchronous Processing**

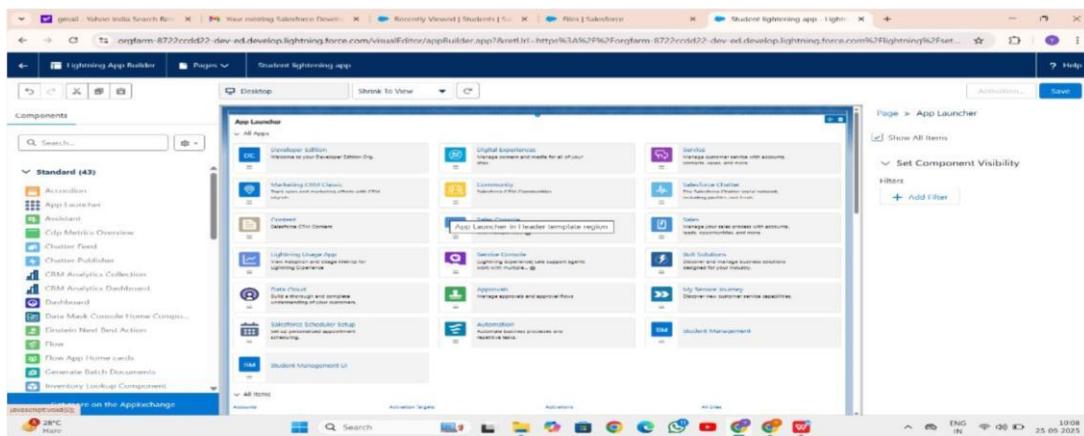
- Implemented and tested all four types: **Batch, Queueable, Scheduled, and Future Methods.**
- Verified job execution via **Apex Jobs** in Setup.
- Ensured scalability for handling large volumes of Student records.

# Phase 6: User Interface Development

**Goal :** To design and implement a user-friendly interface in Salesforce using Lightning tools, LWCs, and Apex integration, enabling users to interact with data efficiently and navigate seamlessly through the application.

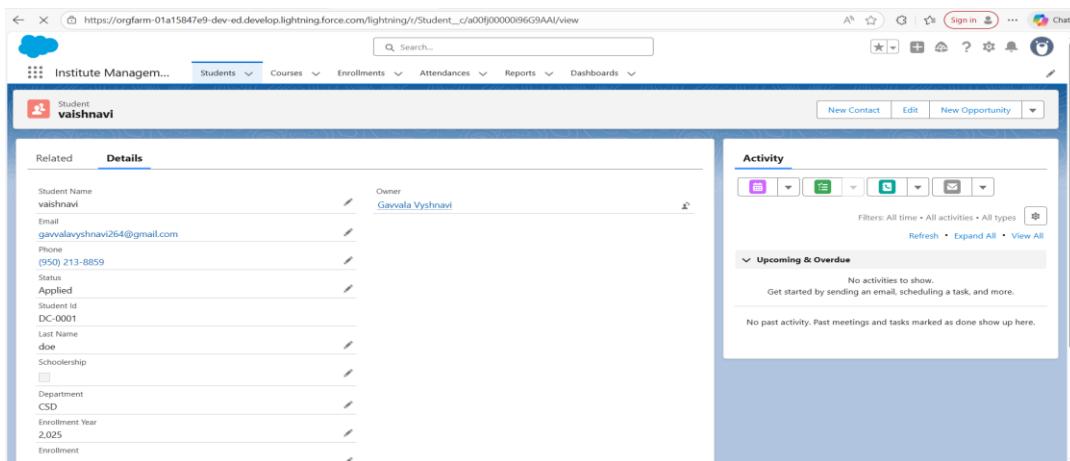
## 1.Lightning App Builder

- Purpose:** Drag-and-drop tool to create custom pages without coding.
- Key Uses:** Build App Pages, Home Pages, and Record Pages.
- Details:** Place standard components or custom LWCs, save, and activate for users.



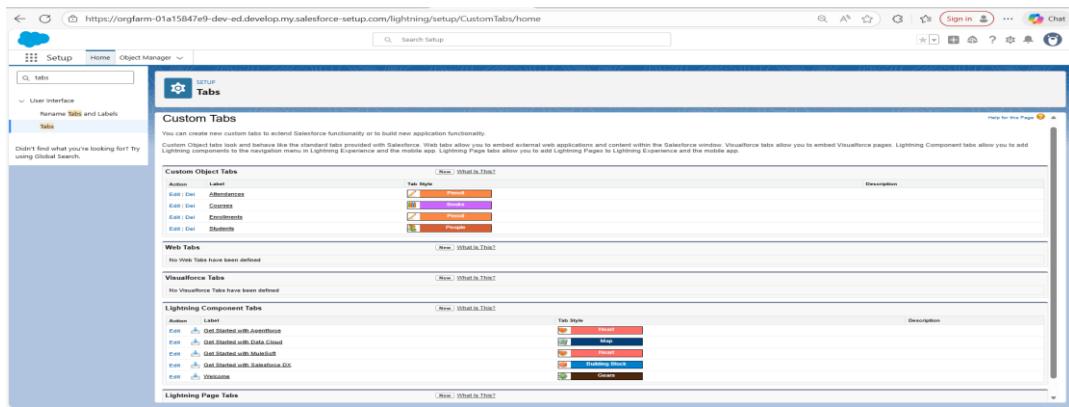
## 2.Record Pages

- Purpose:** Customize the layout of individual records.
- Key Uses:** Display key fields, related lists, and custom LWCs for objects like Student or Course.
- Details:** Assign pages to apps or profiles to control visibility.



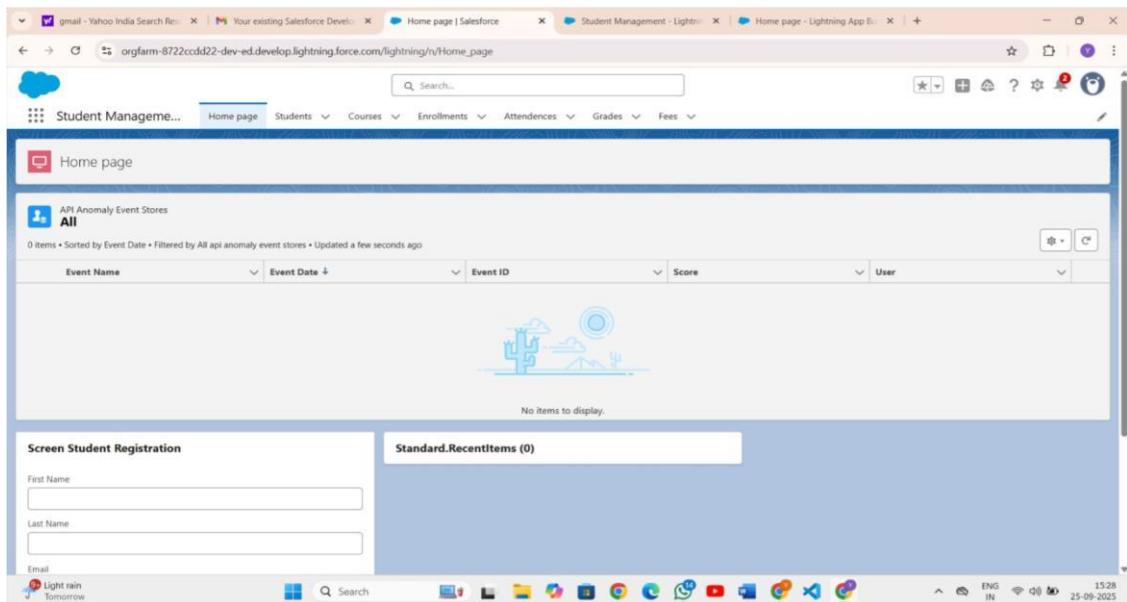
### 3. Tabs

- **Purpose:** Organize objects, pages, or components into tabs for easy navigation.
- **Key Uses:** Access **Students**, **Courses**, **Reports**, or custom pages quickly.
- **Details:** Tabs can host objects, Visualforce pages, web pages, or LWCs.



### 4. Home Page Layouts

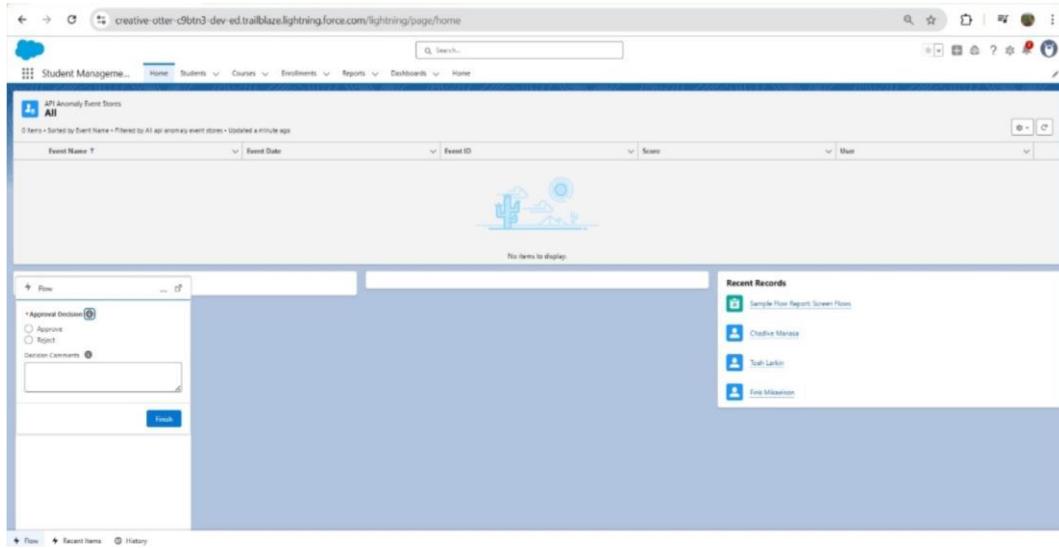
- **Purpose:** Customize the Home Page for better user experience.
- **Key Uses:** Show dashboards, reports, navigation buttons, and custom LWCs.
- **Details:** Layouts define the structure and visibility of components on the Home Page.



### 5. Utility Bar

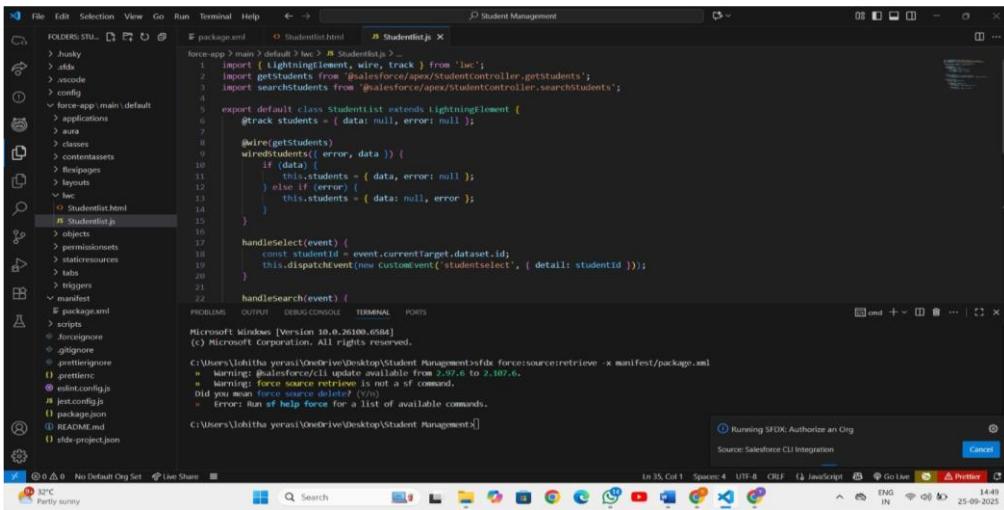
- **Purpose:** Quick access to frequently used tools at the bottom of the app.
- **Key Uses:** Add Notes, Messages, or custom LWCs for easy reach.

- **Details:** Can configure behavior, size, and component order.



## 6. Lightning Web Components (LWC)

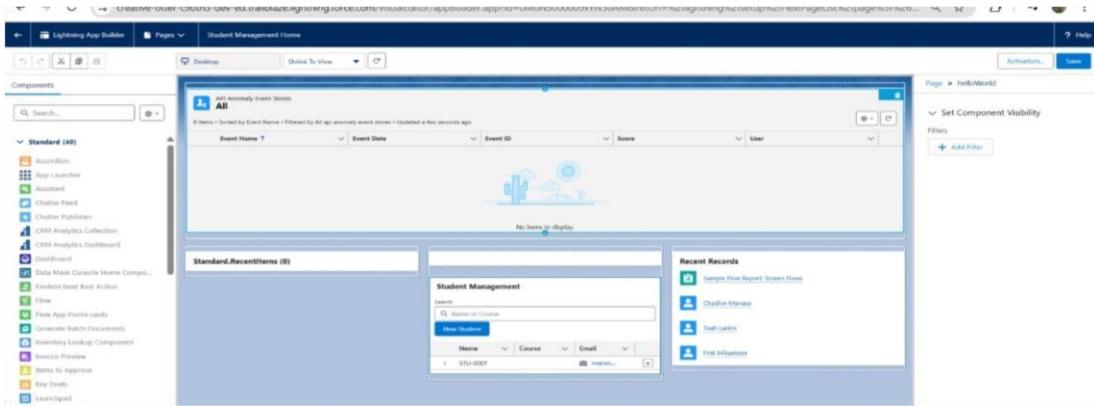
- **Purpose:** Build reusable, interactive components for the UI.
  - **Key Uses:** Display student lists, details, assignments, certificates, and navigation buttons.
  - **Details:** Components consist of HTML (template), JS (functionality), CSS (styling), and meta file (configuration).



## 7. Apex with LWC

- **Purpose:** Fetch or manipulate Salesforce data using Apex in LWCs.
  - **Key Uses:** Display dynamic data like students or courses.

- **Details:** Use **@AuraEnabled** methods in Apex, then call them in JS (Wire or Imperative).



## 8. Events in LWC

- **Purpose:** Enable communication between components.
- **Key Uses:** Pass data from child to parent, or across sibling components.
- **Details:** Use **Custom Events** (`dispatchEvent`) or **Lightning Message Service** for cross-component communication.

## 9. Wire Adapters

- **Purpose:** Automatically fetch Salesforce data reactively.
- **Key Uses:** Display records or lists without manual refresh.
- **Details:** Example: `@wire(getRecord, { recordId: '$recordId', fields }) record;`

## 10. Imperative Apex Calls

- **Purpose:** Call Apex methods manually from JS when needed.
- **Key Uses:** Fetch or update data on button click or action.
- **Details:** Example:

```
import getStudents from '@salesforce/apex/StudentController.getStudents';
handleLoadStudents() {
    getStudents()
        .then(result => { this.students = result; })
        .catch(error => { this.error = error; });
}
```

## 11. Navigation Service

- **Purpose:** Programmatically navigate to records, lists, or external URLs.
- **Key Uses:** Navigate users to **Student details, Course list, or external websites**.
- **Details:** Use NavigationMixin.Navigate() inside LWC methods. Works on **App Pages and Record Pages**, but not standard Home Pages.

# Phase 7: Integration & External Access – Institute Management System

## Goal of This Phase

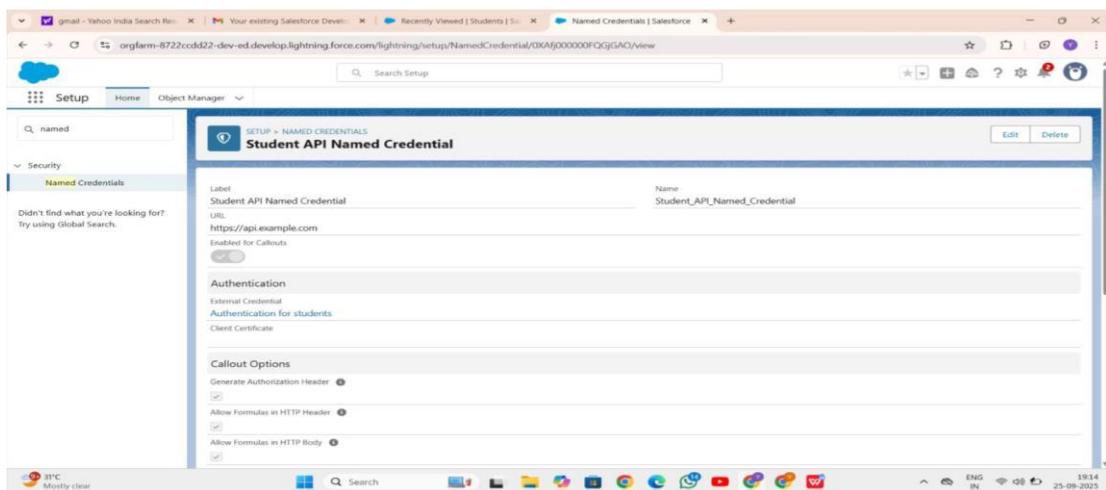
The primary goal of this phase was to enable **seamless data exchange and interaction** between Salesforce and external systems.

This ensures that the Institute Management System can:

- Synchronize student data with other platforms (ERP, Learning Systems, etc.)
- Trigger real-time updates when changes happen inside Salesforce.
- Provide secure and reliable API endpoints for external applications.
- Enforce best practices for authentication, callouts, and integration limits.

### 1. Named Credentials

- **Created Named Credential:** Student\_API\_Named\_Credential
- Configured with:
  - Base URL of the external Student API.
  - Authentication type (if required – basic or OAuth).
  - Purpose: Simplified authentication and callout management without exposing credentials in Apex code.



### 2. External Services

- **Registered External Schema:** Connected to the external Student REST API using the schema definition.
- Auto-generated Apex classes for:
  - Fetching student details.
  - Creating or updating student records externally.

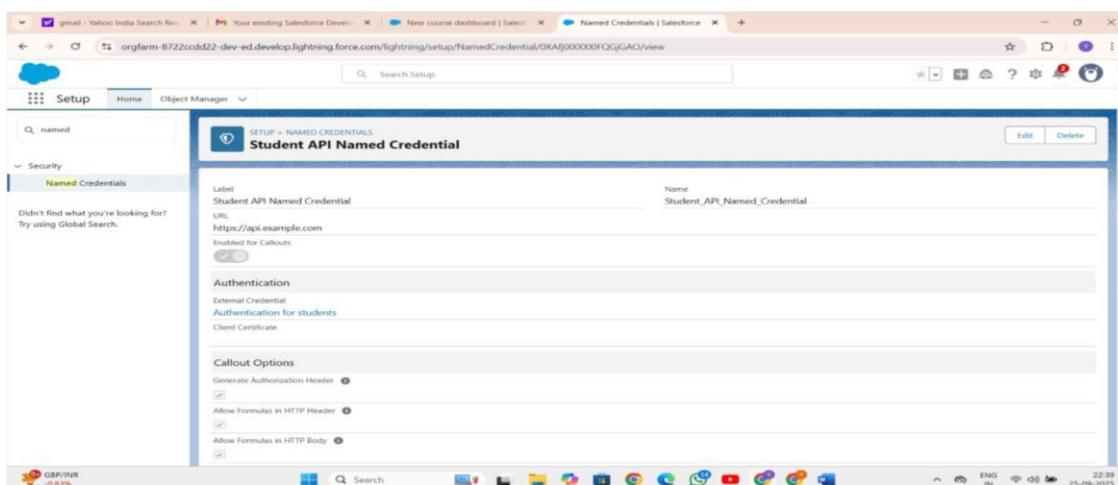
- Purpose: Allow declarative integration (Flow & Apex) without manually writing callout code.



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

- Developed REST Apex Class: StudentRESTService

- Exposed endpoints:
- /students → Create or update student records.
- /students/{id} → Fetch student details
- Student\_ID\_c.
- Added proper authentication and exception handling.
- Purpose: Allow external systems to push and pull student data securely.



## 4. Callouts

- Create Apex Callout Class: CalloutExample
- Added test class with HttpCalloutMock to enable test coverage.
- Verified callout works with Remote Site Settings and Named Credentials.
- Purpose: Send student updates from Salesforce to external systems when a record is created or updated.

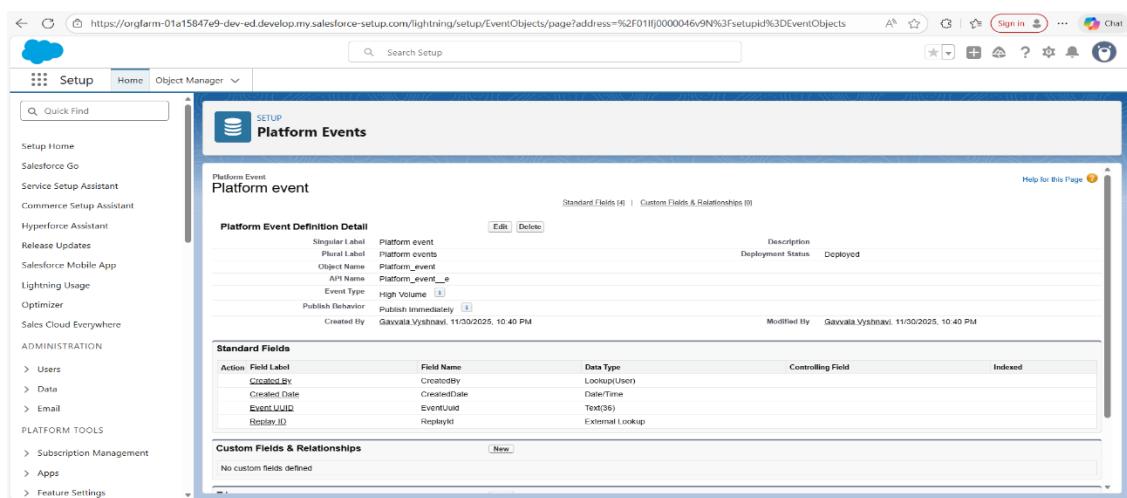
## 5. Platform Events

- **Created Platform Event:** Student\_Event\_\_e

- **Custom Fields:**

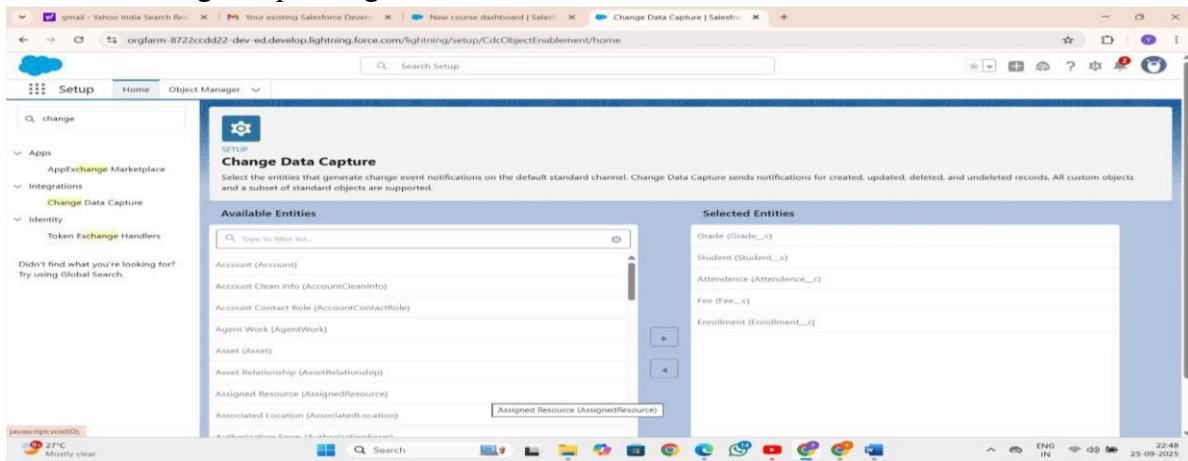
- Student\_ID\_\_c (Text)
- First\_Name\_\_c (Text)
- Last\_Name\_\_c (Text)
- Course\_Program\_\_c (Text)
- Grade\_Marks\_\_c (Text)
- Semester\_\_c (Number)
- Status\_\_c (Checkbox)
- Timestamp\_\_c (Date/Time)

- Purpose: Publish student data changes to subscribers (internal or external) in real time.



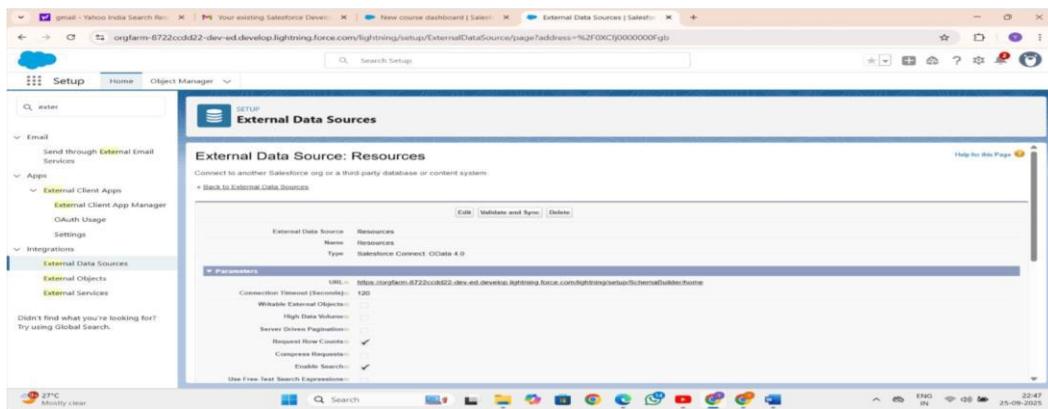
## 6. Change Data Capture (CDC)

- Enabled CDC for Student\_\_c object
- Created Apex Trigger: StudentChangeTrigger
  - Logs change type, changed fields, and commit timestamp.
  - Takes action when Grade\_Marks\_\_c is updated.
- Purpose: Monitor and respond to changes in real time without writing complex logic.



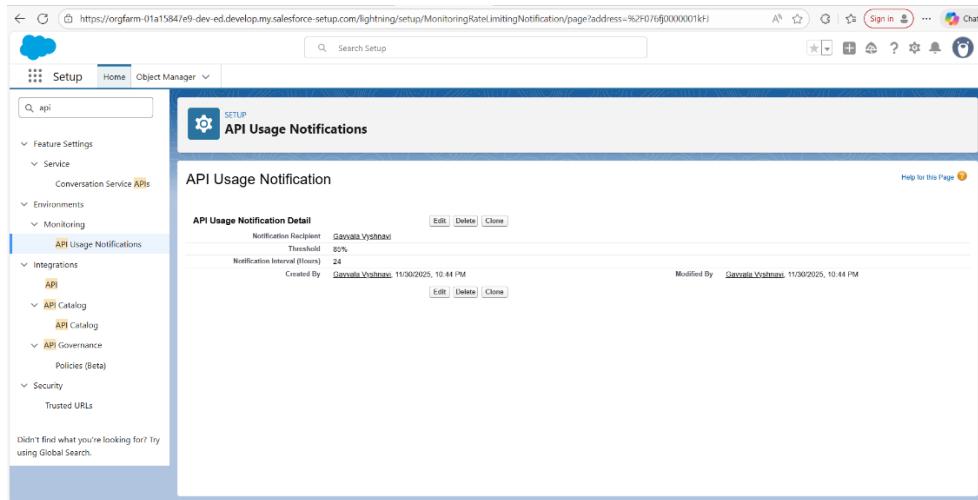
## 7. Salesforce Connect

- Configured External Object (Student\_External\_\_x) to display external student data within Salesforce.
- Verified CRUD operations (read-only or writable depending on requirement).
- Purpose: Provide a unified view of external student records directly in Salesforce UI.



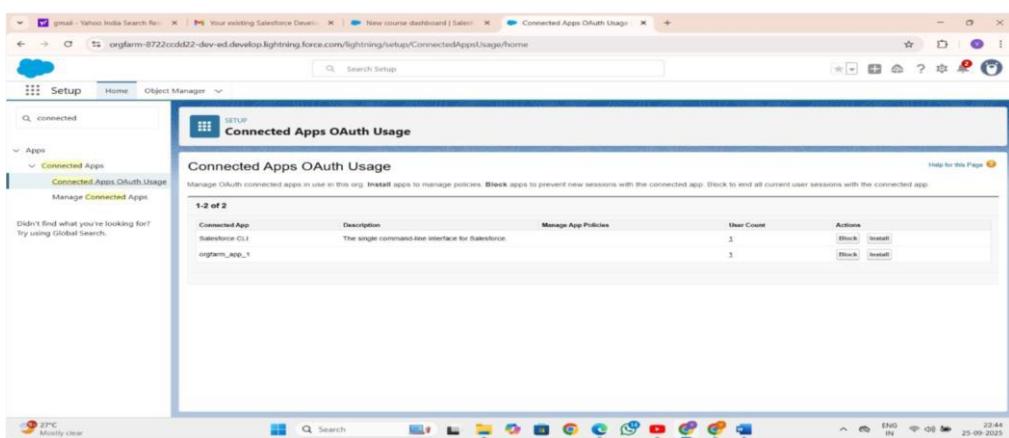
## 8. API Limits

- Enabled **API Usage Notifications** to alert when API usage reaches 75% and 90%.
- Monitored API calls through **System Overview and Reports**.
- Purpose: Prevent API over-consumption and avoid service interruptions.



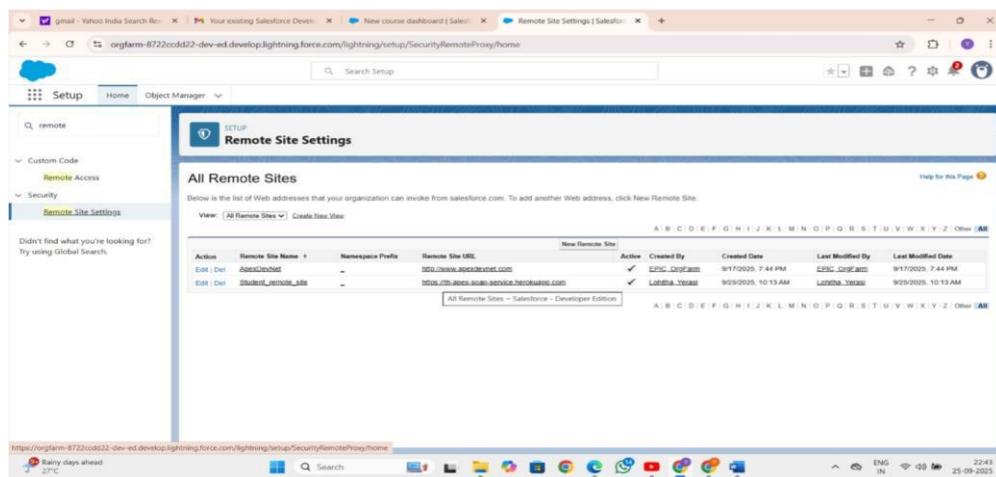
## 9. OAuth & Authentication

- Created Connected App:** InstituteManagementIntegration
- Configured OAuth scopes:
  - Full Access (full)
  - Perform requests on your behalf at any time (refresh\_token, offline\_access)
- Assigned profiles/permission sets.
- Tested authentication using Postman and verified in **Connected Apps OAuth Usage**.
- Purpose: Securely allow external applications to authenticate and integrate.



## 10. Remote Site Settings

- **Created Remote Site:** Student\_API\_Remote\_Site.
- Added base URL of external API to allow callouts.
- Purpose: Ensure Salesforce can make HTTPS callouts to the external Student API.



The screenshot shows the 'Remote Site Settings' page in the Salesforce Setup. The page title is 'Remote Site Settings'. It displays a table of 'All Remote Sites' with two entries:

Action	Remote Site Name	Namespace Prefix	Remote Site URL	Active	Created By	Created Date	Last Modified By	Last Modified Date
Edit   Del	Anonymous	-	00L1a0000000000000	<input checked="" type="checkbox"/>	EEBC_CertClass	9/17/2025, 7:44 PM	EEBC_CertClass	9/17/2025, 7:44 PM
Edit   Del	Student_remote_site	-	00L1a0000000000001	<input checked="" type="checkbox"/>	Lataika_Yadav	9/29/2025, 10:13 AM	Lataika_Yadav	9/29/2025, 10:13 AM

The URL in the browser bar is <https://orgfarm-8722ccdd22-dev-ed.lightning.force.com/lightning/setup/Security/remoteProxy/home>.

# Phase 8: Data Management & Deployment

## Goal :

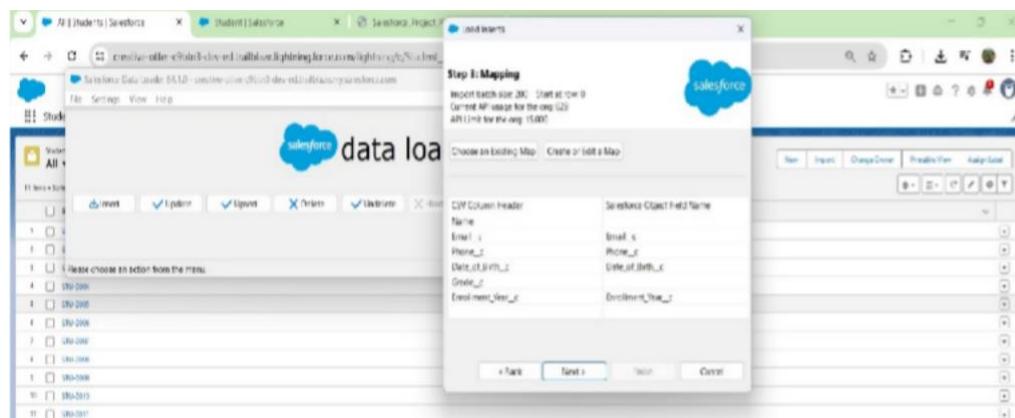
To efficiently manage, migrate, and secure student data and metadata across Salesforce environments using native tools, packages, and modern deployment methods.

### 1. Data Import Wizard

- Imported sample student records and related data through the Salesforce UI.
- Validated successful import with list views and reports.
- Documented mapping fields for reusability.

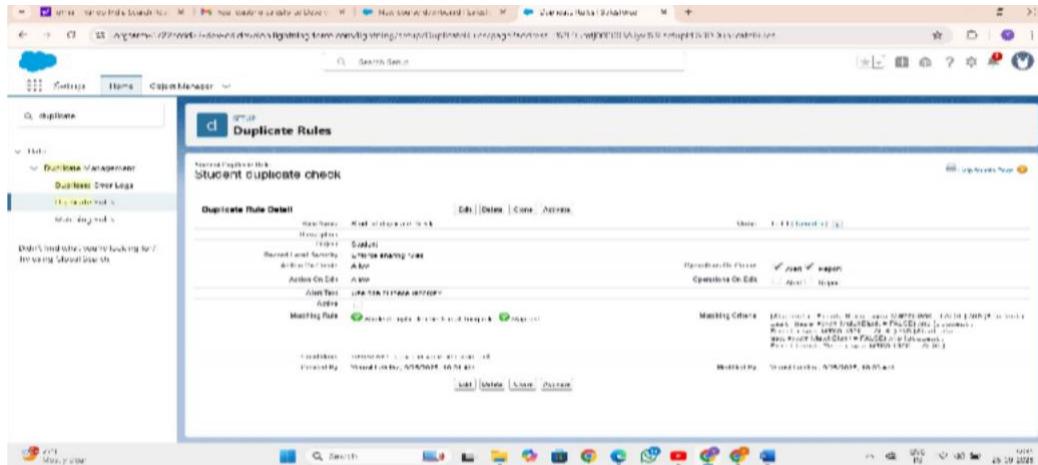
### 2. Data Loader

- Performed bulk inserts, updates, and deletes for large datasets.
- Created CSV templates for Student and Course objects.
- Logged and reviewed success/error CSVs for audit.



### 3. Duplicate Rules

- Defined matching rules to detect duplicate student records.
- Configured duplicate rules to block or allow with alerts.
- Tested with sample data to ensure data quality.



#### 4. Data Export & Backup

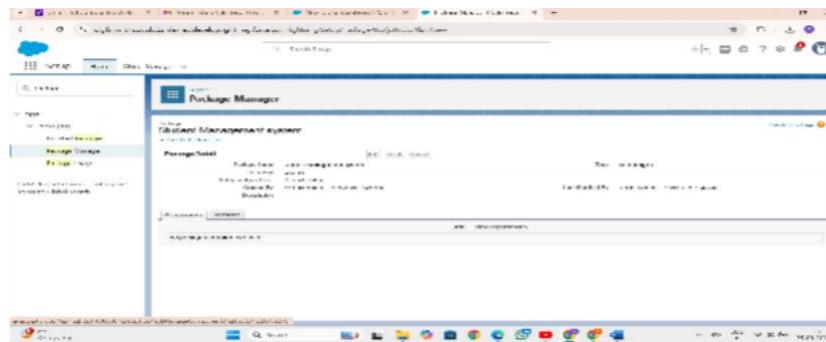
- Scheduled weekly data export from Setup for backup.
- Downloaded CSVs of critical objects (Students, Courses, Enrollments).
- Stored backup securely for rollback scenarios.

#### 5. Change Sets

- Created outbound change sets with custom objects and fields.
- Uploaded change sets to target sandbox and deployed.
- Validated deployment results post-migration.

#### 6. Unmanaged vs Managed Packages

- Built an unmanaged package containing custom objects for sharing.
- Created a managed package to test namespace and code protection.
- Installed both in a new org and documented differences.



#### 7. ANT Migration Tool

- Configured build.xml and package.xml for metadata retrieval.
- Retrieved custom object and Apex class metadata from source org.
- Deployed changes to target org and validated success logs.

## 8. VS Code & SFDX

- Set up VS Code project with Salesforce CLI and authorized orgs.
- Retrieved metadata using sf project retrieve start.
- Modified and deployed changes back using sf project deploy start.

# Phase 9: Reporting, Dashboards & Security Review

## Goal :

Deliver **powerful reporting & analytics** (Reports, Dashboards, Dynamic Dashboards) and implement **robust security controls** (Sharing Settings, Field-Level Security, Session Settings, IP Ranges, Audit Trail) to ensure data is both actionable and protected.

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

- **Tabular Reports:**

- Student List Report (Basic student details)
- Attendance List Report (Daily/Monthly attendance)

- **Summary Reports:**

- Class-wise Student Count
- Attendance Percentage by Class
- Grade Distribution Summary

- **Matrix Reports:**

- Attendance Matrix (Rows = Classes, Columns = Months)
- Grade Performance Matrix (Rows = Subjects, Columns = Terms)

- **Joined Reports:**

- Student + Attendance + Grades combined view for holistic performance analysis
- Course Enrollment + Completion Rate joined report

The screenshot shows a Salesforce Lightning interface with a report titled "Report: Students Student Enrollment Report". The report displays a table of student records with the following data:

Total Records	5		
Status	Student Name	Email	Phone
Applied (5)	gavvalayyashnavi@gmail.com	9602138859	
	gavvalayyashnavi014@gmail.com	9697138859	
	224gt1a2219@sent.ac.in	9014670799	
	john.Dee	9602138859	
	vaishnavi	9602138859	

At the bottom of the report, there are buttons for "Row Counts", "Detail Rows", "Subtotals", and "Grand Total".

Report Name	Description	Folder	Created By	Created On	Subscribed
Exercise Completion by Days to Complete	Analyze an exercise's completion status based on average days to complete.	Unblended Dashboard Reports Summer '24	Automated Process	9/19/2023, 7:05 AM	
Exercise Completion by Days to Complete	Analyze an exercise's completion status based on average days to complete.	Unblended Dashboard Reports Spring '24	Automated Process	9/19/2023, 7:05 AM	
Exercise Completion by User	Analyze how long users are spending on an exercise and the amount of progress they're making.	Unblended Dashboard Reports Spring '24	Automated Process	9/19/2023, 7:05 AM	
Exercise Completion Status	Analyze how long users take to complete an exercise and their completion percentage.	Unblended Dashboard Reports Spring '24	Automated Process	9/19/2023, 7:05 AM	
Exercise Completion Status by Section	Analyze an exercise's completion status based on program sections.	Unblended Dashboard Reports Summer '24	Automated Process	9/19/2023, 7:05 AM	
Exercise Completion Status by Section	Analyze an exercise's completion status based on program sections.	Unblended Dashboard Reports Spring '24	Automated Process	9/19/2023, 7:05 AM	

## 2. Report Types

- Student Report Type:** Includes student personal details, contact info
- Attendance Report Type:** Links Student → Attendance records
- Grades Report Type:** Links Student → Grades with term/subject info
- Courses Report Type:** Links Student → Courses enrolled, status, teacher details

**Details**

Display Label: student attendance report  
 API Name: student\_attendance\_report  
 Description: This is the student attendance report.  
 Created By: Gayvala Vyshnavi, 12/1/25, 12:32 PM  
 Store in Category: other  
 Deployment: Deployed  
 Modified By: Gayvala Vyshnavi, 12/1/25, 12:32 PM

**Fields**

Source Object	Included Fields
Students	20
Attendances	11

**Object Relationships**

Students (A) with at least one related record from Attendances (B)

## 3. Dashboards

- Student Overview Dashboard:** Total Students, New Admissions, Dropouts
- Attendance Dashboard:** Present %, Absent %, Class-wise charts
- Grades Dashboard:** Average Grades per Class, Top Performers, At-Risk Students
- Course Dashboard:** Enrolment Trends, Most Popular Courses

Student: Student Name	Email	Phone
Gayvala Vyshnavi	gayvalavyshnavi0944@gmail.com	9802138869
John Doe	gayvalavyshnavi0944@gmail.com	9802138869
Larkin	224112310@gmail.com.in	9811057099
Vishnavi	gayvalavyshnavi0944@gmail.com	9802138869
Vyshnavi	gayvalavyshnavi0944@gmail.com	9802138869

## 4. Dynamic Dashboards

- Configured dashboards to **run as logged-in user** so each Teacher sees only their class data, Principal sees their school data, and Admin sees all data.
- Applied **dashboard filters** (Class, Term, Subject) for real-time drill-down.

The screenshot shows a dynamic dashboard titled "Institute Overview Dashboard". At the top, there are navigation tabs: Students, Courses, Enrollments, Attendances, Reports, and Dashboards. Below the tabs is a search bar and a toolbar with various icons. A sidebar on the left contains a "Student Enrollment Report" section with a table showing student details like Name, Email, and Phone. The main area is a large grid table with many columns and rows, representing real-time data. At the bottom of the grid, there is a link labeled "View Report (Student Enrollment Report)".

## 5. Sharing Settings:

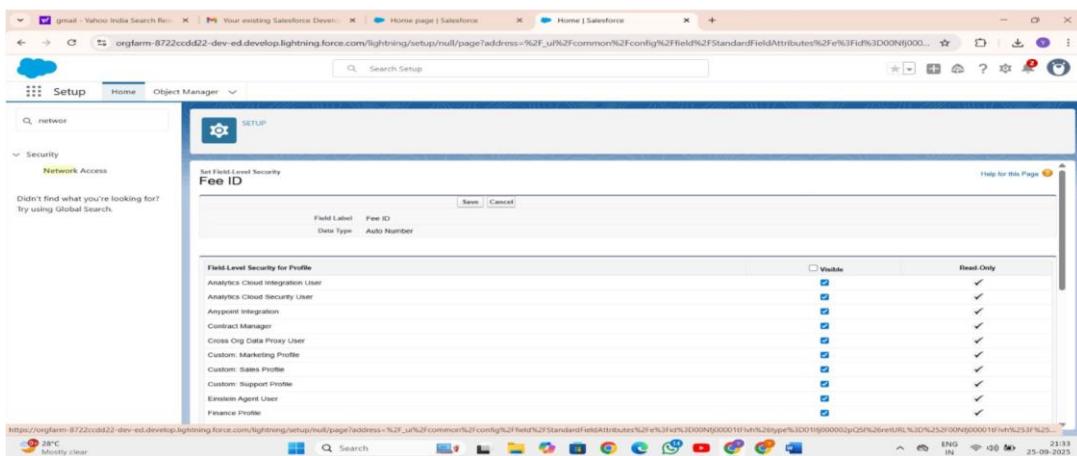
- Organization-Wide Defaults (OWD):**
  - Students, Attendance, Grades = Private
  - Courses=ControlledbyParent

The image contains two screenshots of the Salesforce Sharing Settings page. The top screenshot shows a detailed view of sharing rules for various objects like User, Contact, Account, etc., with columns for Action, Criteria, Shared With, and Access Level. The bottom screenshot shows a more organized view of sharing rules categorized by object type: Attendance, Course, Fee, Grade, and Student, each with its own set of sharing rules.

- **Role Hierarchy:**
  - Admin > Principal > Teacher > Student
- **Sharing Rules:**
  - Criteria-based rules to share records with Teacher Groups based on Class/Section
  - Owner-based sharing for Principals to access all data under their school

## 6. Field-Level Security

- **Sensitive Fields Restricted:**
  - SSN, Parent Contact Info → Visible to Admin & Principal only
  - Fee Details → Visible to Admin only
  - Grades → Read-Only for Teachers, Visible to Principal/Admin
- **Profiles Configured:** Admin, Principal, Teacher, Student with tailored field visibility



## 7. Session Settings

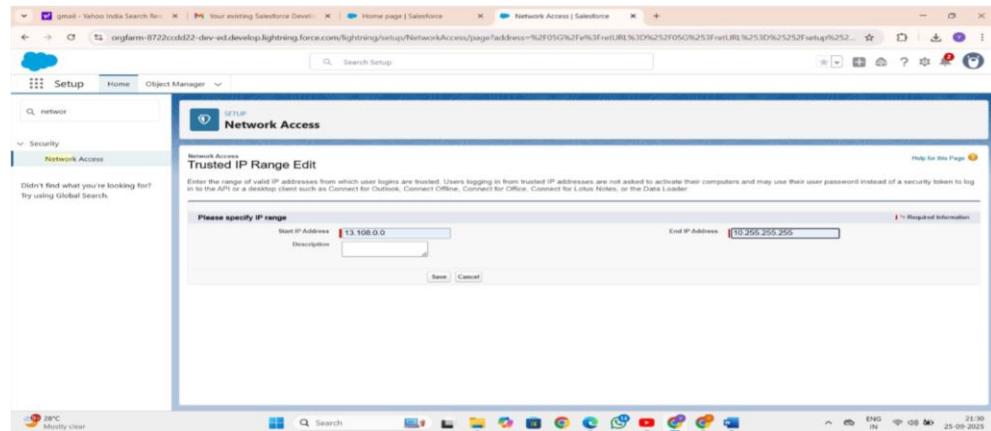
- **Session Timeout:** 30 minutes
- **Logout on Timeout:** Enabled
- **Lock Session to IP/Domain:** Enabled
- **Clickjack & CSRF Protection:** Enabled
- **HTTPS Required:** Enforced for secure communication

## 8. Login IP Ranges

- **Network Access (Org-Wide):** Added school's LAN/Wi-Fi IP range

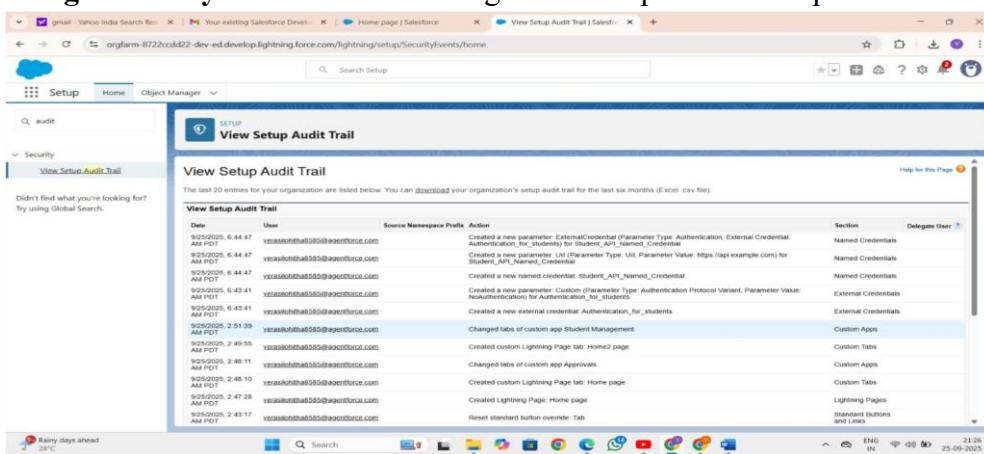
- **Profile-Specific Ranges:**

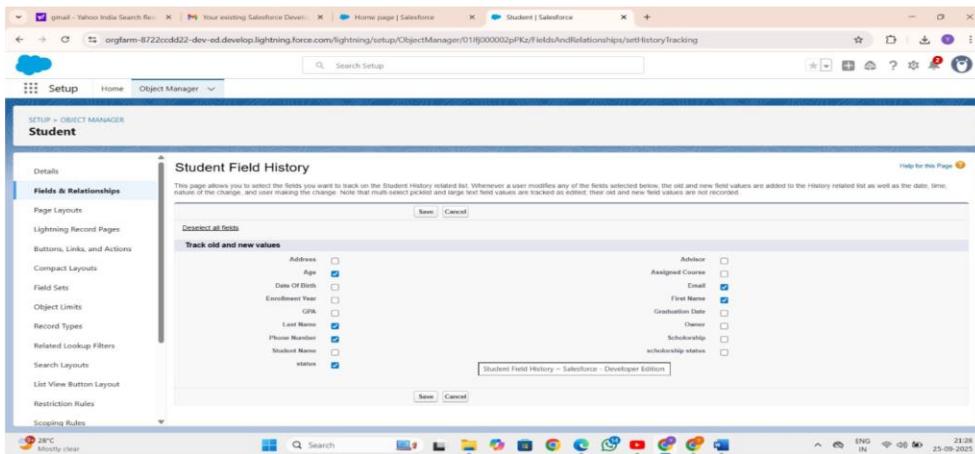
- Teachers: Allowed only from school network
- Admin: Allowed from office network + VPN
- Students: Allowed from broader IP range (school + home if needed)



## 9. Audit Trail

- **Setup Audit Trail:** Enabled and reviewed for changes in roles, profiles, and settings
- **Field History Tracking:**
  - Tracked Fields: Student Status, Assigned Teacher, Grade Changes, Attendance Updates
- **Login History:** Monitored failed logins and suspicious attempts





Identity	
	Setup
Identity	
Login Flows	
<b>Login History</b>	
Security	
Login Access Policies	
Didn't find what you're looking for? Try using Global Search.	

Login History												Help for this Page	
Download Options												Help for this Page	
Username	Login Time	Source IP	Location	Login Type	Status	Browser	Platform	Application	Client Version	API Type	API Version	Login URL	HTTP Method
Z24g1x32h02@salesforce.com	11/10/2025, 10:38:16 PM PST	223.187.22.203	India	Application	Success	Edge 142	Windows	Browser	N/A	N/A	N/A	orgfarm-01a15847d9-dev-ed.develop.my.salesforce.com	POST
Z24g1x32h02@salesforce.com	11/10/2025, 9:09:21 PM PST	27.59.251.233	India	Application	Success	Edge 142	Windows	Browser	N/A	N/A	N/A	login.salesforce.com	POST
spic.kt6kdeSeBaDByphnre.salesforce.com	11/23/2025, 2:20:35 PM PDT	62.1.139.45	United States	Remote Access	Success	Unknown 2.0	Unknown	orgfarm_app_1	N/A	N/A	N/A	login.salesforce.com	POST
Z24g1x32h02@salesforce.com	11/10/2025, 7:47:41 PM PDT	223.187.38.26	India	Application	Success	Edge 141	Windows	Browser	N/A	N/A	N/A	orgfarm-01a15847d9-dev-ed.develop.my.salesforce.com	POST
Z24g1x32h02@salesforce.com	10/19/2025, 4:03:26 AM PDT	223.185.79.210	India	Application	Success	Edge 141	Windows	Browser	N/A	N/A	N/A	orgfarm-01a15847d9-dev-ed.develop.my.salesforce.com	POST
Z24g1x32h02@salesforce.com	10/19/2025, 8:06:29 PM PDT	223.185.79.210	India	Application	Success	Edge 141	Windows	Browser	N/A	N/A	N/A	orgfarm-01a15847d9-dev-ed.develop.my.salesforce.com	POST
Z24g1x32h02@salesforce.com	10/19/2025, 1:14:22 AM PDT	223.185.86.202	India	Application	Success	Edge 141	Windows	Browser	N/A	N/A	N/A	orgfarm-01a15847d9-dev-ed.develop.my.salesforce.com	POST
Z24g1x32h02@salesforce.com	10/17/2025, 8:09:21 PM PDT	223.185.86.202	India	Application	Success	Edge 141	Windows	Browser	N/A	N/A	N/A	orgfarm-01a15847d9-dev-ed.develop.my.salesforce.com	POST
Z24g1x32h02@salesforce.com	9/24/2025, 10:37:17 PM PDT	106.217.237.135	India	Application	Success	Edge 141	Windows	Browser	N/A	N/A	N/A	orgfarm-01a15847d9-dev-ed.develop.my.salesforce.com	POST

## Phase 10: Testing

- Created sample Students, Courses, Enrollments, and Attendance records
- Verified validation rules to prevent duplicate enrollments
- Tested Auto Attendance Flow to ensure records are created automatically
- Checked reports and dashboards for accurate grouping and metrics
- Security tested for correct access permissions and field-level restrictions
- Screenshots captured for all key functionalities

The screenshot shows a Salesforce Lightning application window. At the top, there's a navigation bar with links for Students, Courses, Enrollments, Attendances, Reports, and Dashboards. On the left, there's a sidebar with a cloud icon and the text "Institute Management...". The main area displays a list titled "Recently Viewed" under the "Students" category. The list shows four items, each with a checkbox and a name: 1. Gavvala Vyshnavi, 2. vyshnavi, 3. john Doe, and 4. Larkin. There are also buttons for New, Import, Change Owner, and Assign Label at the top right of the list area.

[https://orgfarm-01a15847e9-dev-ed.develop.lightning.force.com/lightning/r/Student\\_\\_c/a00fj00000U67ycAAJ/view](https://orgfarm-01a15847e9-dev-ed.develop.lightning.force.com/lightning/r/Student__c/a00fj00000U67ycAAJ/view)

The screenshot shows a Salesforce Lightning interface for a student record. The top navigation bar includes links for Students, Courses, Enrollments, Attendances, Reports, and Dashboards. The main content area displays the details of a student named Gavala Vyshnavi. The 'Details' tab is selected, showing fields such as Student Name (Gavala Vyshnavi), Email (gavalavyshnavi264@gmail.com), Phone ((950) 213-8859), DOB (9/5/2005), Enrollment Status (Applied), Enrollment (Gavala Vyshnavi), Course (Java), and Created By (Gavala Vyshnavi). The 'Owner' field shows Gavala Vyshnavi. The 'Last Modified By' field shows Gavala Vyshnavi at 10/17/2025, 8:44 PM. To the right, there is an 'Activity' section with a summary message: 'No activities to show. Get started by sending an email, scheduling a task, and more.' It also indicates 'No past activity. Past meetings and tasks marked as done show up here.'

[https://orgfarm-01a15847e9-dev-ed.develop.lightning.force.com/lightning/o/Course\\_\\_c/list?filterName=\\_Recent](https://orgfarm-01a15847e9-dev-ed.develop.lightning.force.com/lightning/o/Course__c/list?filterName=_Recent)

The screenshot shows a Salesforce Lightning interface for a list of courses. The top navigation bar includes links for Students, Courses, Enrollments, Attendances, Reports, and Dashboards. The main content area displays a list titled 'Recently Viewed' under the 'Courses' category. The list shows 2 items updated a few seconds ago, with columns for Course Name, Number, and Action. The items listed are 1. Java and 2. Python Programming.

The screenshot shows a course detail page for 'Java'. The top navigation bar includes links for Students, Courses (selected), Enrollments, Attendances, Reports, and Dashboards. A search bar and various icons are also present. The main content area displays course details under the 'Details' tab:

Course Name	Owner
Java	Gawala Vyshnavi
Course Name	
Credits	
3	
Department	
CSD	
Duration	
6	
Course Code	
12345	
Start Date	
10/17/2025	
End Date	
4/17/2026	
Created By	Last Modified By
Gawala Vyshnavi, 9/24/2025, 11:49 PM	Gawala Vyshnavi, 10/17/2025, 8:44 PM

The screenshot shows a list of recently viewed enrollments. The top navigation bar is identical to the previous screenshot. The main content area displays a list titled 'Recently Viewed':

Enrollment Name
1 john Doe
2 Gawala Vyshnavi
3 Larkin
4 vyshnavi

[https://orgfarm-01a15847e9-dev-ed.lightning.force.com/lightning/r/Enrollment\\_c/a01fj00000MtmQjAA/view](https://orgfarm-01a15847e9-dev-ed.lightning.force.com/lightning/r/Enrollment_c/a01fj00000MtmQjAA/view)

The screenshot shows the Salesforce Lightning interface for an Enrollment record. The top navigation bar includes links for Students, Courses, Enrollments (selected), Attendances, Reports, and Dashboards. The main content area displays the details for an enrollment named "john Doe". The "Details" tab is selected, showing fields such as Enrollment Name (john Doe), Owner (Gawala Vyshnavi), Student (john Doe), Course (Python Programming), Status (Active), Enrollment Date (10/15/2025), and Created By (Gawala Vyshnavi). The last modified date is also listed as 10/17/2025, 8:41 PM.

[https://orgfarm-01a15847e9-dev-ed.lightning.force.com/lightning/r/Attendance\\_c/a04fj00000YcdxAAC/view](https://orgfarm-01a15847e9-dev-ed.lightning.force.com/lightning/r/Attendance_c/a04fj00000YcdxAAC/view)

The screenshot shows the Salesforce Lightning interface for an Attendance record. The top navigation bar includes links for Students, Courses, Enrollments, Attendances (selected), Reports, and Dashboards. The main content area displays the details for an attendance entry with ID 1. The "Details" tab is selected, showing fields such as Attendance ID (1), Owner (Gawala Vyshnavi), Student (john Doe), Course (Python Programming), Date (10/17/2025), Status (Present), and Enrollment (john Doe). The last modified date is also listed as 10/17/2025, 8:54 PM.

## **11. Conclusion**

The Salesforce Institute Management System is fully functional, tested, and ready for deployment. It streamlines administrative processes, automates attendance tracking, enforces robust security, and provides actionable insights through reports and dashboards. This project demonstrates the power of Salesforce in managing educational operations efficiently and securely.

---