

Student Enrollment & Course Management System (SECMS)

User Story:

The Student Enrollment & Course Management System is a comprehensive Salesforce application designed to streamline and automate the academic enrollment operations of an educational institute. This system enables administrators, enrollment officers, and instructors to efficiently manage student records, course offerings, enrollments, instructor assignments, approval processes, payments, and reporting—all within a unified Salesforce environment.

1. INTRODUCTION

Project Overview :

This project aims to centralize all student, course, and enrollment-related data within a single Salesforce application to ensure easy data access, structured workflows, and accurate reporting. The system allows Enrollment Officers to manage student admissions, course registrations, and communication effectively, while instructors and administrators gain full transparency into course allocations and academic progress. The solution also enables automated instructor assignment, task creation, validation checks, and multi-level approval processes for enrollment management.

With Salesforce Flows, Approval Processes, Validation Rules, Reports, and Dashboards, the organization can ensure that all enrollment operations—from admission to course completion—are handled systematically and with continuous monitoring of key performance indicators such as enrollment volume, revenue, student status distribution, and pending approvals.

Purpose:

The primary purpose of SECMS is to replace manual and fragmented enrollment processes with a scalable, automated, and secure Salesforce solution that ensures:

- Accurate student data management
- Structured enrollment workflows
- Automated approvals and validations
- Transparent reporting and monitoring

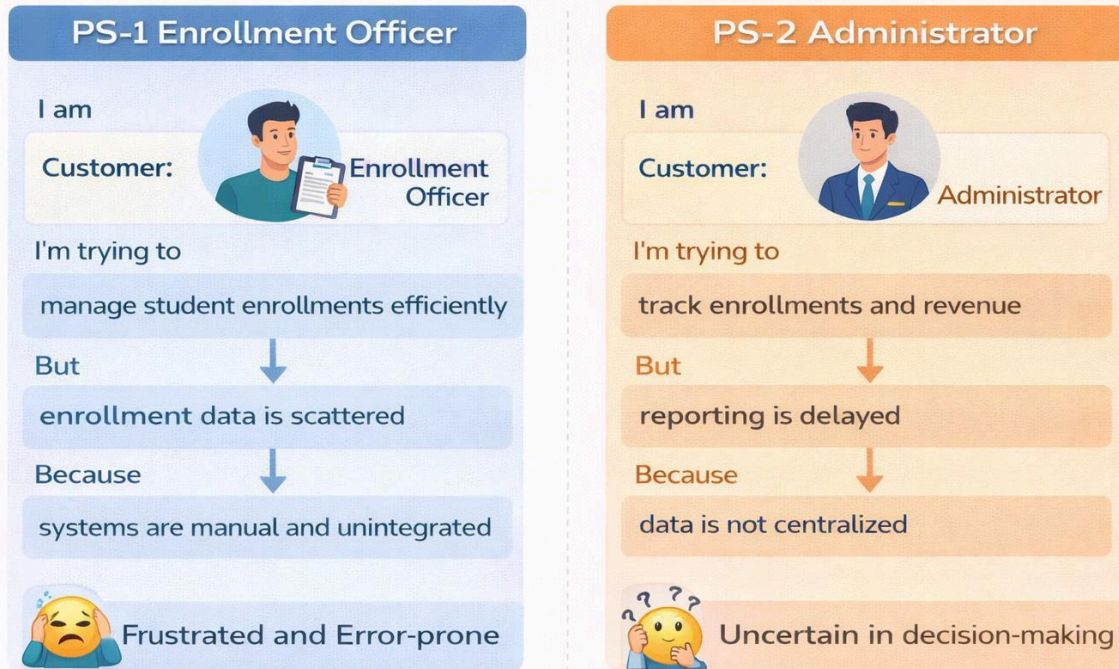
2. IDEATION PHASE

Problem Statement (Customer Problem Statement Template)

PS	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Enrollment Officer	manage student enrollments efficiently	enrollment data is scattered	systems are manual and unintegrated	frustrated and error-prone
PS-2	Administrator	track enrollments and revenue	reporting is delayed	data is not centralized	uncertain in decision-making

2. IDEATION PHASE

2.1 Problem Statement (Customer Problem Statement Template)



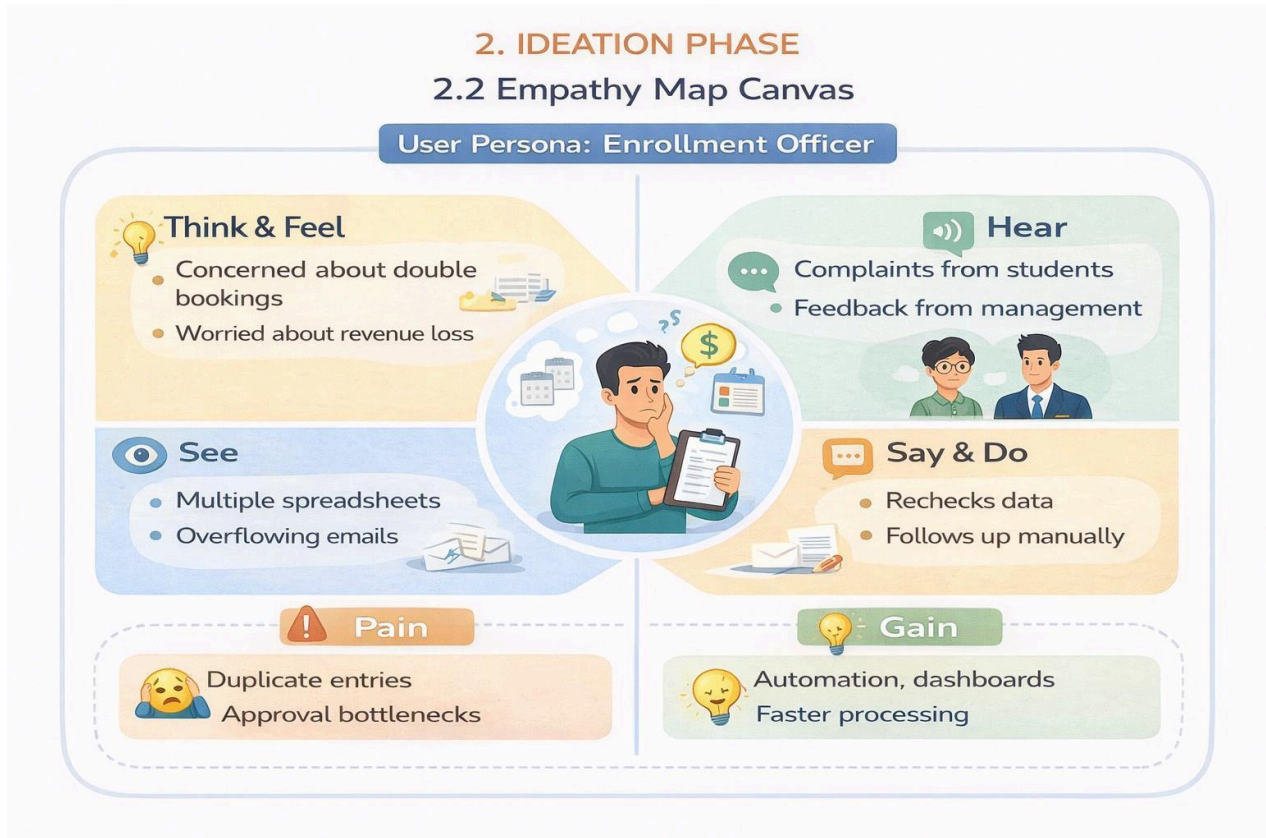
EmpathyMapCanvas

User Persona: Enrollment Officer

- **Think&Feel:** Concerned about double bookings and revenue loss
- **Hear:** Complaints from students and management
- **See:** Multiple spreadsheets and emails
- **Say&Do:** Rechecks data, follows up manually
- **Pain:** Duplicate entries, approval bottlenecks
- **Gain:** Automation, dashboards, faster processing

2. IDEATION PHASE

2.2 Empathy Map Canvas



Brainstorming & Idea Prioritization

Step 1 – Brainstormed Ideas

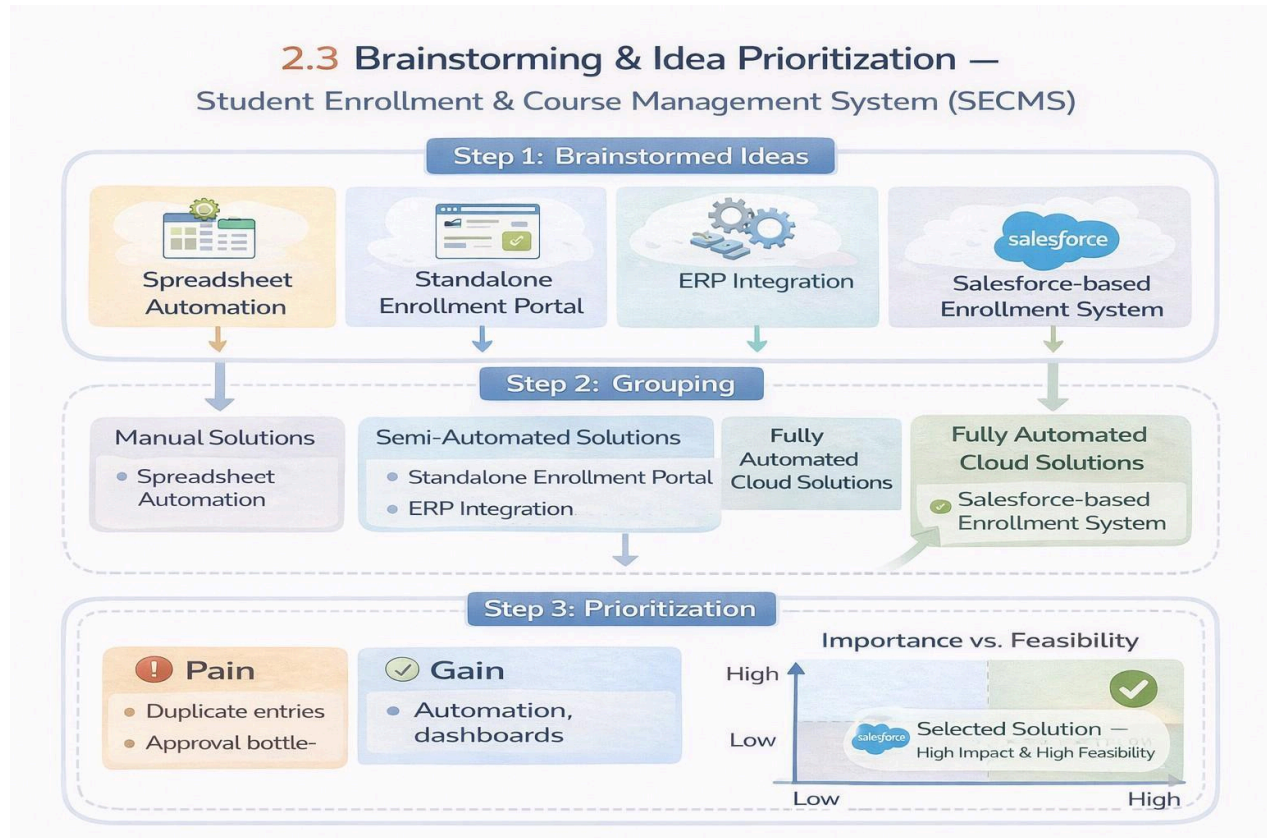
- Spreadsheet automation
- Standalone enrollment portal
- ERP integration
- Salesforce-based enrollment system

Step 2 – Grouping

- Manual solutions
- Semi-automated solutions
- Fully automated cloud solutions

Step3– Prioritization

Salesforce-based SECMS was selected due to **high feasibility and high impact**.



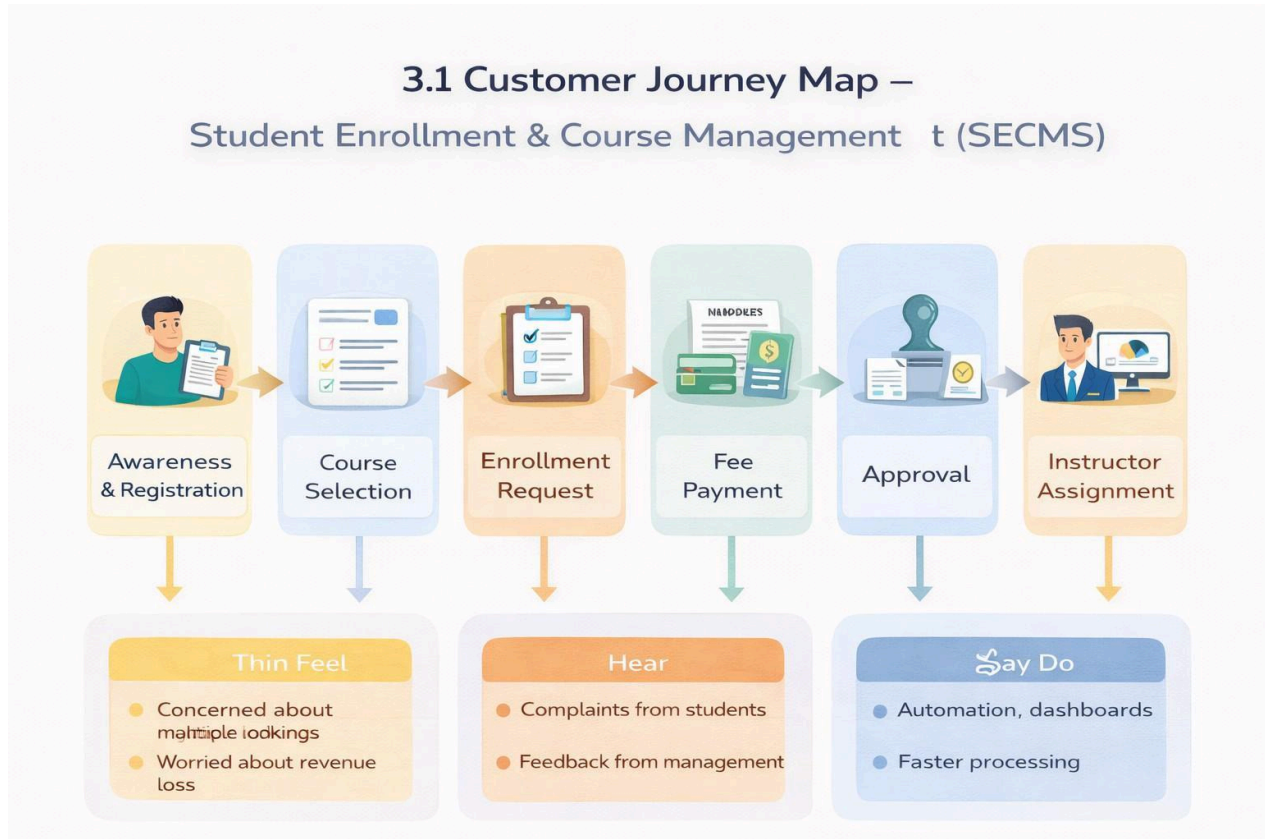
3. REQUIREMENT ANALYSIS PHASE

Customer Journey Map

Stages covered:

- Awareness & Registration
- Course Selection
- Enrollment Request
- Fee Payment
- Approval
- Instructor Assignment

- Reporting & Completion



Solution Requirements

Functional Requirements

FRNo	Epic	Description
FR-1	StudentManagement	Manage student records
FR-2	CourseManagement	Create and manage courses
FR-3	EnrollmentProcessing	Request, approve, reject enrollments

FR-4	InstructorAssignment	Auto-assigninstructors
FR-5	Reporting	Dashboardsandreports

Non-FunctionalRequirements

NFR-1: Usability

The system shall provide a simple, intuitive, and user-friendly interface that enables enrollment officers, administrators, instructors, and students to perform operations efficientlywithminimaltraining.Navigationshallbeorganizedthroughclearlylabeled tabs, dashboards, and forms to reduce user effort and errors.

Examples:

- Quickstudentregistrationform
- One-clickenrollmentapproval
- Role-baseddashboards

NFR-2:Security

The system shall ensure the confidentiality, integrity, and protection of student and institutional data through secure authentication, authorization, and access controls. Sensitiveinformationsuchaspersonaldata,enrollmenthistory,andacademicrecords shall only be accessible to authorized users.

Examples:

- Onlyadminsmodifycoursedetails
- Instructorsviewonlyassignedstudents
- Studentscannotaccessotherstudentrecords

NFR-3:Reliability

The systemshalloperateconsistentlyandaccuratelywithoutfailuresduringenrollment processing, approvals, and reporting. Automation logic shall execute correctly to avoid duplicate or incorrect enrollments.

Examples:

- Noduplicatecourseregistration
- Accuratestatusupdatesafterapprovals
- Consistentreportgeneration

NFR-4:Performance

The system shall provide fast response times for all operations to ensure smooth and uninterrupted user experience, especially during peak admission periods.

Examples:

- Quick dashboard refresh
- Fast search for student records
- No lag during bulk uploads

NFR-5:Scalability

The system shall support future growth in the number of students, courses, departments, and users without degradation in performance or functionality. It shall allow easy addition of new modules and features.

Examples:

- Add new departments without redesign
- Handle increased admissions annually
- Extend to mobile or portal integration

NFR-6:Availability

The system shall be continuously accessible to users during institutional working hours and must minimize downtime. The platform shall support high availability for academic operations.

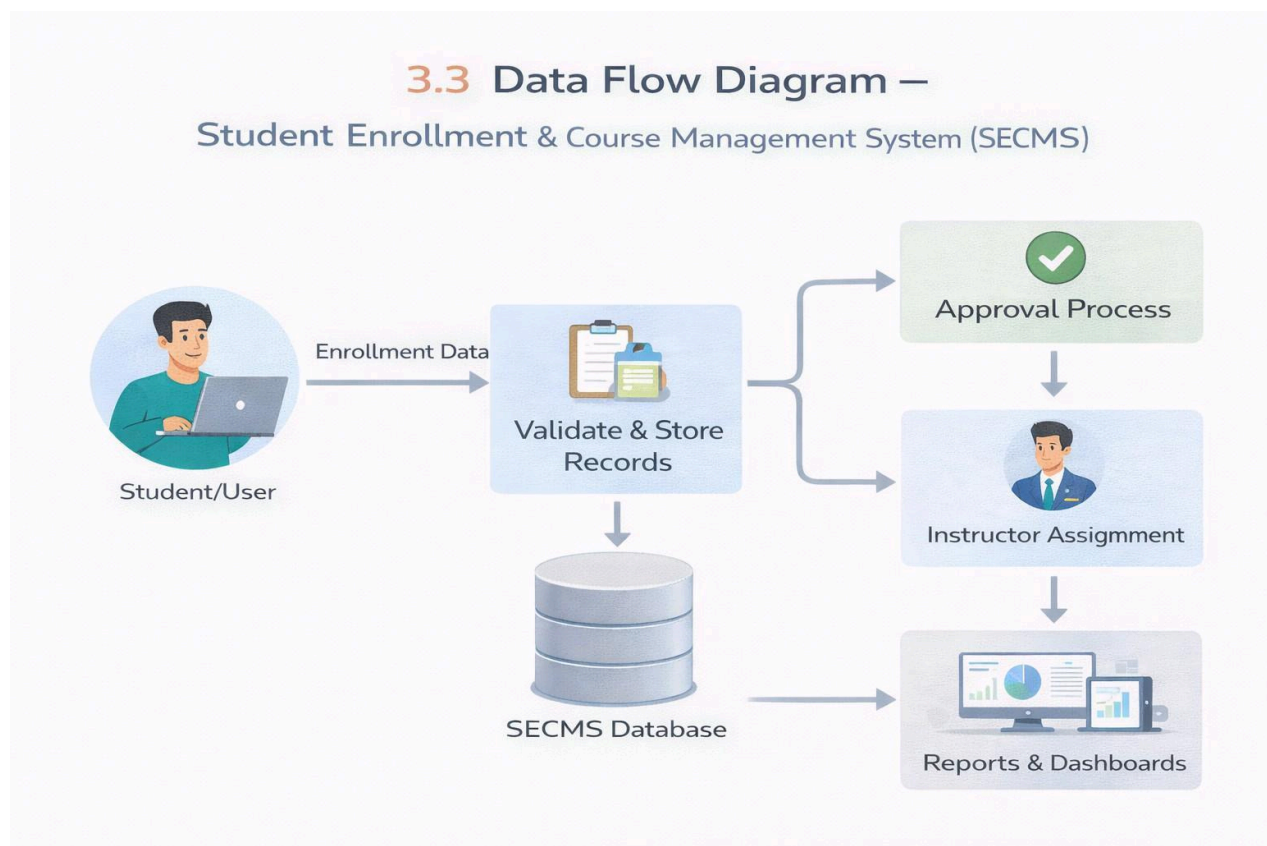
Examples:

- 24/7 cloud access
- Reliable during peak admission season
- Automatic backup of student records

Data Flow Diagram

- User inputs student and enrollment data
- System validates and stores records
- Approval process routes data
- Reports and dashboards consume processed data.

The system processes enrollment requests, validates data, routes approvals, assigns instructors, and generates reports.



Technology Stack

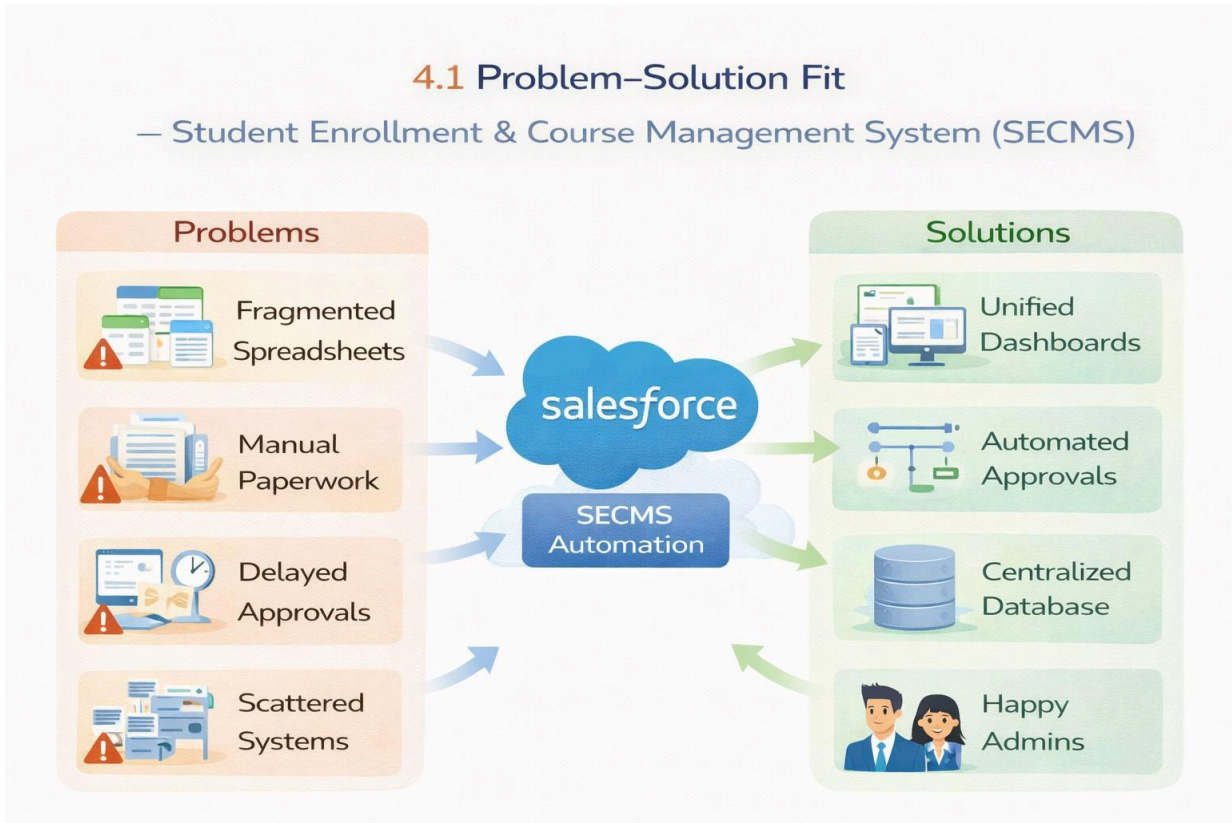
Layer	Technology
UI	Salesforce Lightning

Logic	SalesforceFlows
Database	SalesforceObjects
Automation	Validation Rules,ApprovalProcess
Reporting	Reports&Dashboards

4. PROJECT DESIGN PHASE

Problem–Solution Fit

SECMS directly addresses fragmented enrollment workflows by aligning user behavior, constraints, and automation using Salesforce-native capabilities.



ProposedSolution

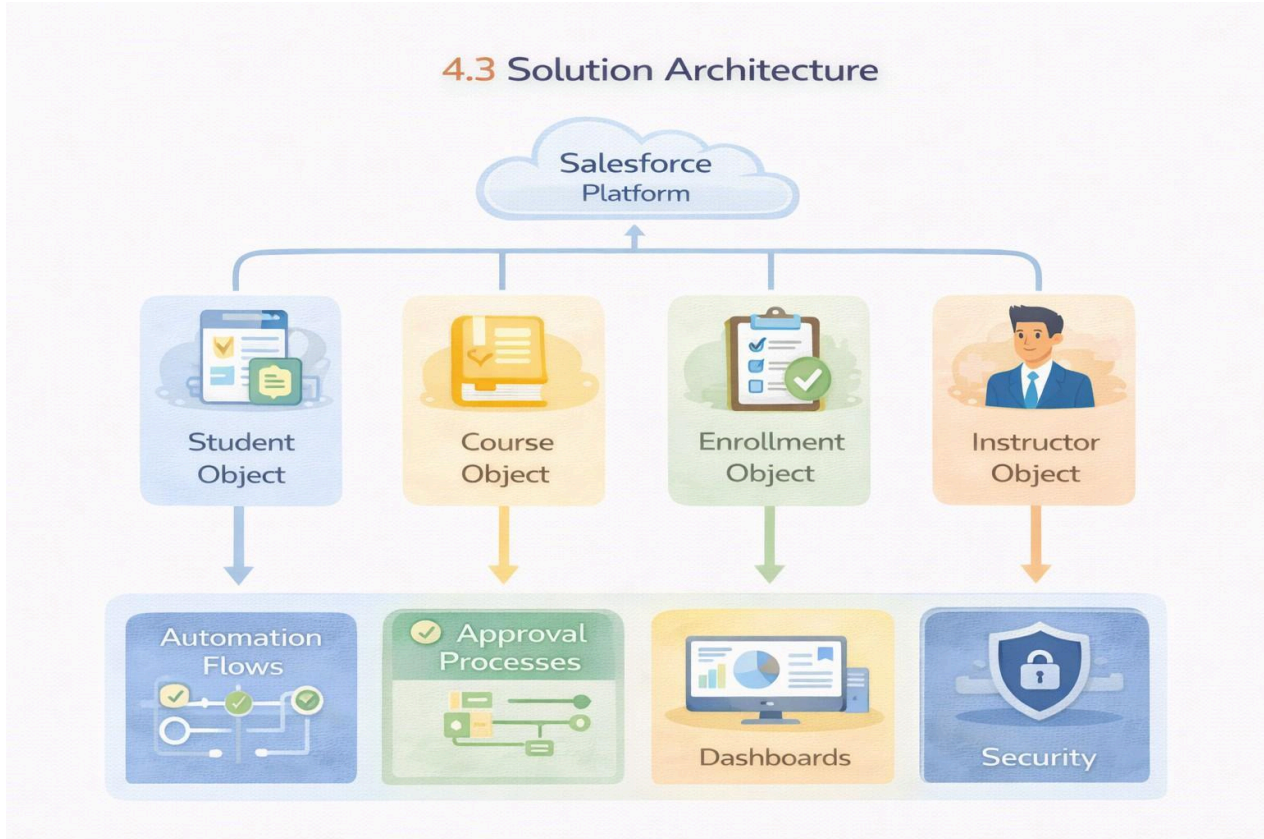
FunctionalRequirements

Parameter	Description
Problem	Manualenrollmenthandling
Solution	Salesforce-basedSECMS
Novelty	End-to-endautomation
SocialImpact	Improvedstudentexperience
RevenueModel	Institutionaldeployment
Scalability	Multi-departmentssupport

SolutionArchitecture

ThearchitectureintegratesSalesforceobjects,automationflows,approvalprocesses, dashboards, and security layers.

4.3 Solution Architecture



5. PROJECT PLANNING & SCHEDULING

Project Planning

- Agile methodology
- Epics → Stories → Story Points
- Sprint-based execution
- Velocity calculation used for estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Developer Setup	USN-1	As a system administrator, I want to create a Salesforce developer accounts that I can configure and deploy the Student Enrollment system..	1	High	Kalaiyarasi j
Sprint-1	Data Modeling	USN-2	As an admin, I want to create custom objects for Student,	3	High	Parameswari B

Sprint	Functional Requirement (Epic)	User Story Number	UserStory/Task	Story Points	Priority	Team Members
			Course, Enrollment, and Instructor so that academic data can be stored and managed efficiently.			
Sprint-2	Data Modeling	USN-3	As a user, I want dedicated tabs for each object so that I can easily navigate between students, courses, enrollments, and instructors.	3	High	Nivetha S
Sprint-2	Data Modeling	USN-4	As an admin, I want to create relevant fields such as student details, course information, and enrollment status so that complete records can be captured accurately.	5	High	Jayasri S
Sprint-2	Data Modeling	USN-5	As an enrollment officer, I want a centralized Lightning App so that I can access all academic operations from one interface.	3	High	Kalaiyarasi j
Sprint-2	Data Modeling	USN-6	As an admin, I want to define relationships between students, courses, and enrollments so that the system can logically link and track academic activities.	5	Medium	Parameswari B
Sprint-2	Data Modeling	USN-7	As a user, I want customized page layouts so that only relevant information is displayed, improving clarity and ease of data entry.	5	High	Parameswari B
Sprint-3	Automation	USN-8	As a system, I want to validate input data such as mandatory fields and enrollment limits so that incorrect or incomplete records are prevented.	3	High	Kalaiyarasi j
Sprint-3	Automation	USN-9	As a system, I want automated flows to handle enrollment processing and notifications so that manual effort is reduced and processes become faster.	3	High	Jayasri S
Sprint-3	Automation	USN-10	As an administrator, I want an approval process for enrollment requests so that registrations are verified before confirmation.	5	High	Nivetha S
Sprint-4	Security	USN-11	As an admin, I want to configure role-based security so that users can only access data relevant to their responsibilities.	5	High	Parameswari B
Sprint-5	Reports	USN-12	As a manager, I want reports on student enrollments, course registrations, and instructor assignments so that I can monitor academic operations.	4	High	Parameswari B
Sprint-6	Dashboards	USN-13	As a manager, I want visual dashboards showing enrollment	4	Medium	Kalaiyarasi j

Sprint	Functional Requirement (Epic)	User Story Number	UserStory/Task	Story Points	Priority	Team Members
			statisticsandtrendssothatIcan quickly analyze system performance.			

ProjectTracker,Velocity&BurndownChart:(4Marks)

Sprint	TotalStory Points	Duration	SprintStart Date	SprintEndDate (Planned)	StoryPoints Completed (as on PlannedEnd Date)	SprintRelease Date (Actual)
Sprint-1	20	6Days	03mar 2026	04mar 2026	20	09mar 2026
Sprint-2	20	6Days	04mar 2026	05 mar 2026		
Sprint-3	20	6Days	06mar 2026	07 mar 2026		
Sprint-4	20	6Days	08 mar 2026	09 mar 2026		

6. ProjectDevelopmentPhase:

ProjectFlow:

Milestone1:SalesforceDeveloperAccountCreation

Milestone2:CustomObjectCreation(Student,Course,Enrollment,Instructor) Milestone 3:

Tabs Creation

Milestone4:LightningAppCreation

Milestone 5: Fields Creation

Milestone 6: Create Relationships

Milestone 7: Page Layout Customization Milestone

8: Validation Rules

Milestone 9: Flows (Automation)

Milestone 10: Approval Process

Milestone 11: Reports

Milestone 12: Dashboards

Milestone 13: Security Setup

Milestone 14: Conclusion

What you'll learn

1. Real-time Salesforce project implementation
2. Designing custom objects and relationships
3. Creating page layouts & record types
4. Writing validation rules
5. Building automation with flows
6. Creating approval workflows
7. Developing reports & dashboards
8. Setting up security with profiles, permission sets, and sharing
9. End-to-end application testing

Milestone 1 - Salesforce :

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?"

What is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Salesforce has everything you need to run your business from anywhere.

Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud. So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:

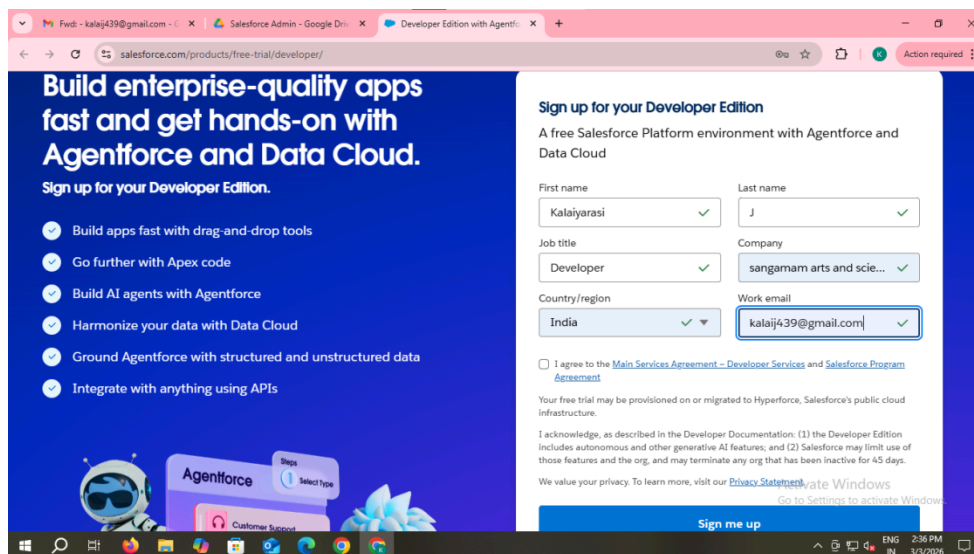
<https://youtu.be/r9EX3IGde5k>

Activity 1: Creating Developer Account:

Creating a developer org in Salesforce.

Go to

<https://developer.salesforce.com/signup> On this sign up form, enter the following details:



The screenshot shows a web browser window with the URL [salesforce.com/products/free-trial/developer/](https://developer.salesforce.com/signup). The page is titled "Sign up for your Developer Edition" and features a blue background with white text. On the left, there is a promotional banner for "Agentforce and Data Cloud" with a list of benefits: "Build apps fast with drag-and-drop tools", "Go further with Apex code", "Build AI agents with Agentforce", "Harmonize your data with Data Cloud", "Ground Agentforce with structured and unstructured data", and "Integrate with anything using APIs". On the right, there is a sign-up form with the following fields: "First name" (Kalaiyarasi), "Last name" (J), "Job title" (Developer), "Company" (sangamam arts and scie...), "Country/region" (India), and "Work email" (kalaij439@gmail.com). Below the form, there is a checkbox for "I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement" and a "Sign me up" button. The browser's taskbar at the bottom shows the time as 2:36 PM on 3/9/2025.

Firstname & Lastname

Email

Role: Developer

Company:CollegeName County

: India

PostalCode:pincode

Username:shouldbeacombinationofyournameandcompany

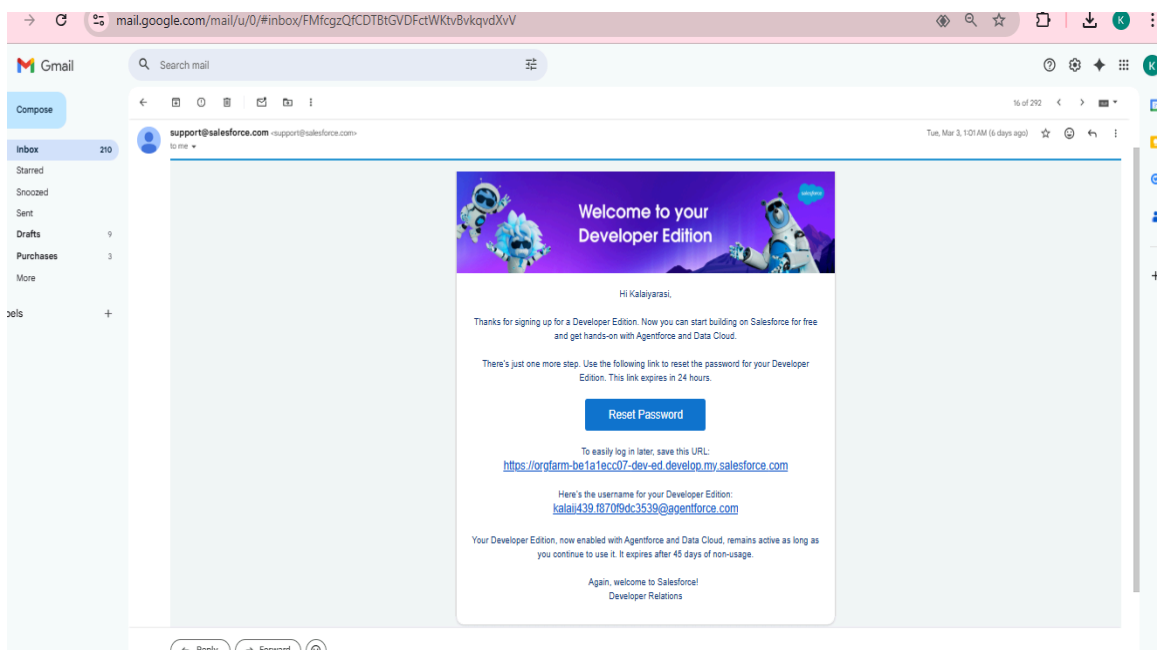
Thisneednotbeanactualemailid,youcangiveanythingintheformat:

username@organization.com

Clickonsignmeupafterfillingthese.

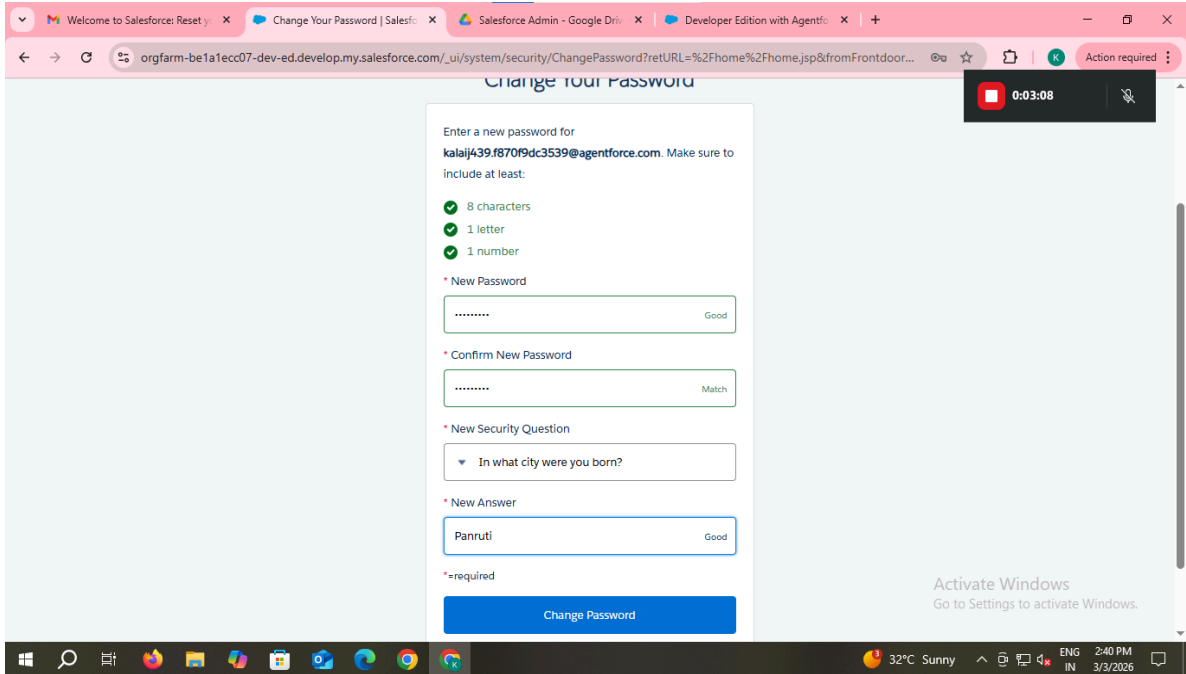
Activity2:AccountActivation:

1. Gototheinboxoftheemailthatyouusedwhilesigningup.Clickonthe verify account to activate your account. The email may take 5-10mins.

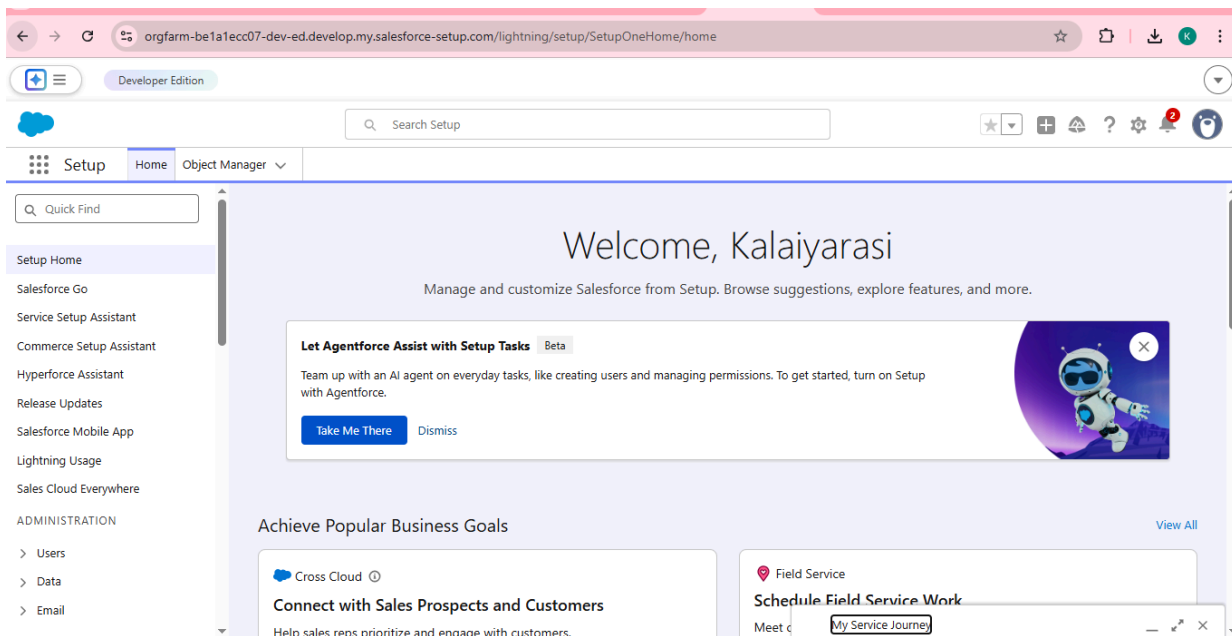


2. ClickonVerifyAccount

3. Giveapasswordandanswersecurityquestionandclickonchange password.



4. Then you will redirect to your salesforce setup page.



Milestone2–Objects:

Activity1:CreatingStudentObject:

The purpose of creating the Student custom object is to store and manage information about students such as their personal details, contact information, and enrollment status.

To create the Student object:

Go to the Setup page.

Click on Object Manager.

Click on Create → Custom Object.

Enter the details as below:

1. Label: Student
2. Plural Label: Students
3. Object Name: Student
4. Record Name: Student Name
5. Data Type: Text
6. Under Optional Features, select:
7. Allow Reports
8. Allow Activities (optional but recommended)
9. Track Field History (optional, if you want)
10. Under Search Status, select:
11. Allow Search
12. Click Save.

orgfarm-be1a1ecc07-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/011fj000005tAAL/edit?address=%2F011fj000005tAAL%2Fe...

Developer Edition

Setup Home Object Manager

SETUP > OBJECT MANAGER

Student

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
- Scoping Rules

Edit Custom Object Student

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports. Be careful when changing the name or label as it may affect existing integrations and merge templates. ! Required Information

Label: Example: Account

Plural Label: Example: Accounts

starts with vowel sound

The Object Name is used when referencing the object via the API.

Object Name: Example: Account

Description:

Context-sensitive Help Setting: Open the standard Salesforce.com Help & Training window
 Open a window using a Visualforce page

Content Name:

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number".

My Service Journey

orgfarm-be1a1ecc07-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/011fj000005tAAL/edit?address=%2F011fj000005tAAL%2Fe...

Developer Edition

Setup Home Object Manager

SETUP > OBJECT MANAGER

Student

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
- Scoping Rules

Context-sensitive Help Setting: Open the standard Salesforce.com Help & Training window
 Open a window using a Visualforce page

Content Name:

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: Example: Account Name

Data Type: Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

- Allow Reports
- Allow Activities
- Track Field History
- Allow in Chatter Groups
- Enable Licensing

Object Classification

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more](#)

- Allow Sharing
- Allow Bulk API Access
- Allow Streaming API Access

Deployment Status

In Development

What is this?

My Service Journey

SETUP > OBJECT MANAGER

Student

- Details**
- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters

Allow in Chatter Groups
 Enable Licensing [i](#)

Object Classification
When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more.](#)

Allow Sharing
 Allow Bulk API Access
 Allow Streaming API Access

Deployment Status [What is this?](#)
 In Development
 Deployed

Search Status
When this setting is enabled, your users can find records of this object type when they search. [Learn more.](#)

Allow Search

My Service Journey — ↗ ✕

InthesamewayCreate**Courses,Instructors,andEnrolmentsobjects**

Milestone3-Fields:

Table1:StudentObjectFields:

Table2:CourseObjectFields:

Table3:InstructorObjectFields:

FieldLabel	DataType
Instructor Name	Text (Standard)
Instructor Code	Text
Expertise	Picklist
Phone	Phone
Email	Email

Table4:EnrollmentObjectFields:

Field Label	Data Type
Student	Lookup (Student)
Course	Lookup(Course)
Instructor	Lookup (Instructor)
Enrollment Date	Date
Enrollment Status	Picklist

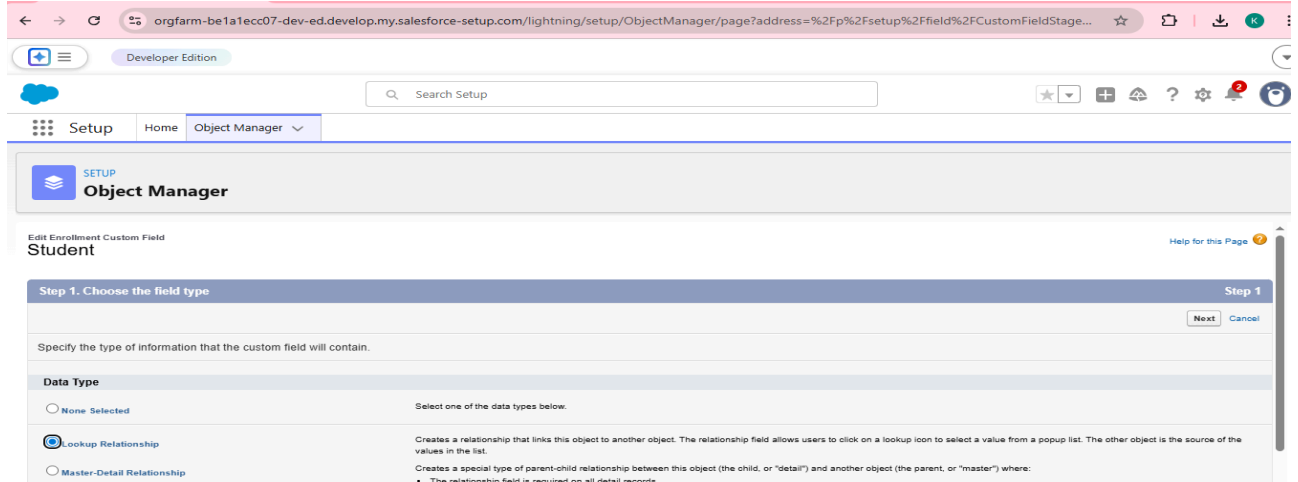
Fees Paid	Checkbox
Total Amount	Currency
Comments	TextArea

Milestone4–CreateRelationships:

1. Activity1:CreateLookupfromEnrollmenttoStudent:

1. GotoSetup→clickObjectManager.
2. TypeEnrollmentinQuickFind→clickEnrollment_____cobject.
3. ClickFields&Relationships→clickNew.
4. SelectLookupRelationship→clickNext.
5. ForRelatedTo,selectStudent_____c→click Next.

6. EnterFieldLabel:Student(APIName:Student____c).
7. ClickNextthroughField-LevelSecurityandLayoutoptions(ensure it's visible and added to layout).
8. ClickSave.

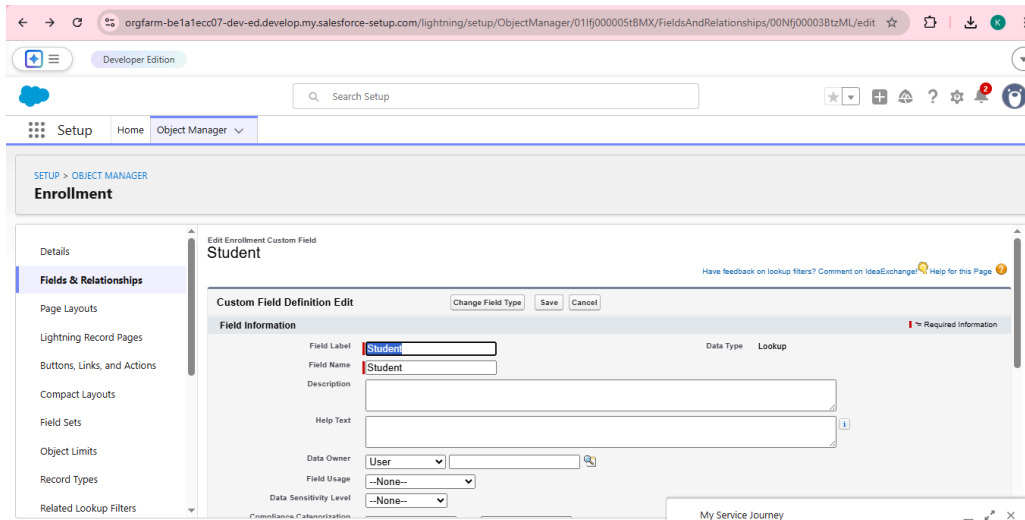


FieldLabel	Data Type
StudentName	Text (Standard)
Email	Email
Phone	Phone
DateofBirth	Date
StudentStatus	Picklist
Address	TextArea

Field Label	Data Type
Course Name	Text (Standard)
Category	Picklist
Duration	Number/Text
CourseFee	Currency
Description	TextArea

Activity2:CreateLookupfromEnrollmenttoCourse

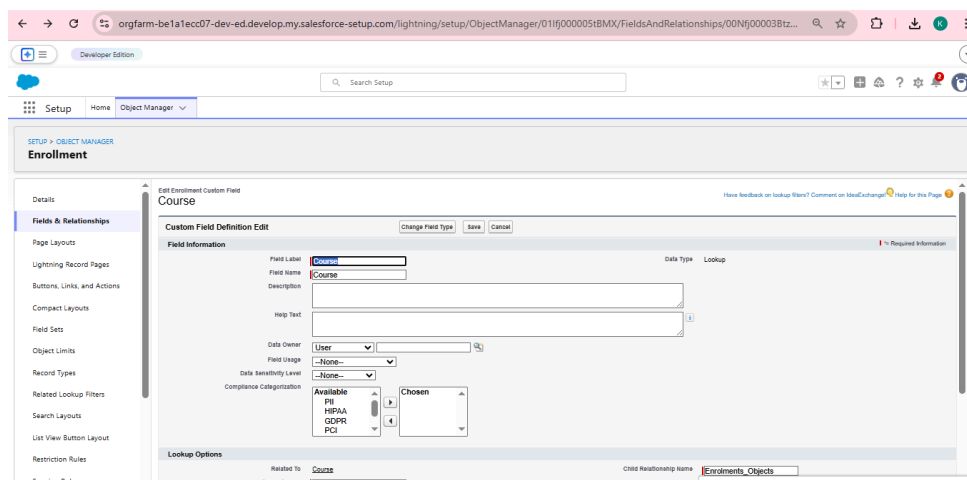
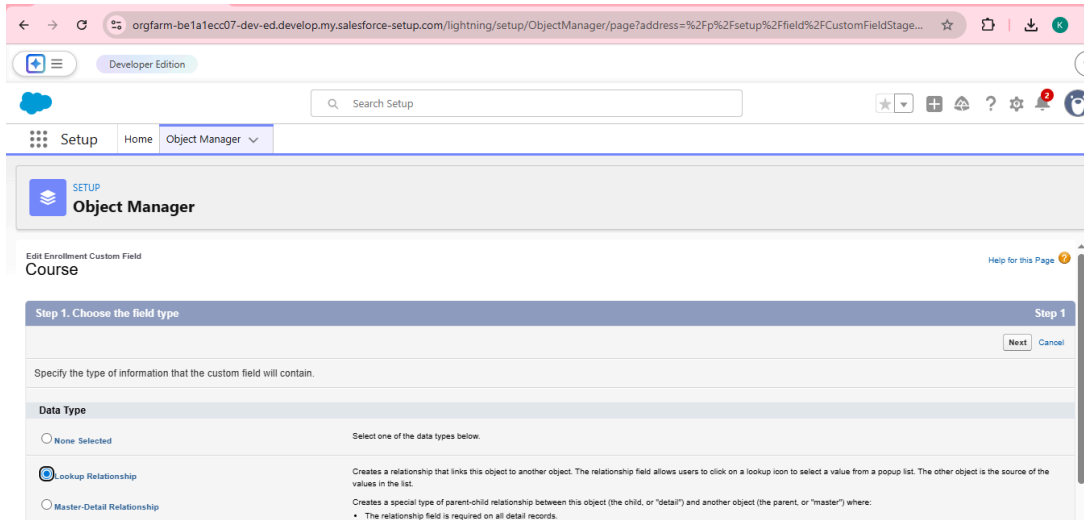
Purpose:



To link each Enrollment to a Course.

Steps:

1. Goto Setup → Object Manager → Enrollment _____ c.
2. Click Fields & Relationships → New.
3. Select Lookup Relationship → Next.
4. Related To: select Course _____ c → click Next.
5. Field Label: Course (API Name: Course _____ c).
6. Click Next (set visibility and layout) → Save.



Activity3:CreateLookupfromEnrollmenttoInstructor:

Purpose:

To store which Instructor is assigned to a particular Enrollment.

Steps:

1. Goto Setup → Object Manager → Enrollment _____ c.
2. Click Fields & Relationships → New.
3. Choose Lookup Relationship → Next.
4. Related To: select Instructor ____ c → Next.
5. Field Label: Instructor (API Name: Instructor _____ c).
6. Complete Field-Level Security and Page Layout options → Save

orgfarm-be1a1ecc07-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/011fj0000051BMX/FieldsAndRelationships/new

Developer Edition

Setup Home Object Manager

SETUP > OBJECT MANAGER

Enrollment

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

New Custom Field

Step 1. Choose the field type

Specify the type of information that the custom field will contain.

Data Type

None Selected Select one of the data types below.

Auto Number A system generated sequence number that uses a display format you define. The number is automatically incremented for each new record.

Formula A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

Roll-Up Summary A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.

Lookup Relationship Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.

Master-Detail Relationship Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:

- The relationship field is required on all detail records.
- The relationship and status of a detail record are determined by the master record.

Next Cancel

orgfarm-be1a1ecc07-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/011fj0000051BMX/FieldsAndRelationships/new

Developer Edition

Setup Home Object Manager

SETUP > OBJECT MANAGER

Enrollment

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

New Relationship

Step 2. Choose the related object

Select the other object to which this object is related.

related to:

Previous Next Cancel

Setup Home Object Manager

SETUP

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more!](#) [Don't show this message again](#)

Custom Object Definition Edit

Save Save & New Cancel

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label Example: Account

Plural Label Example: Accounts

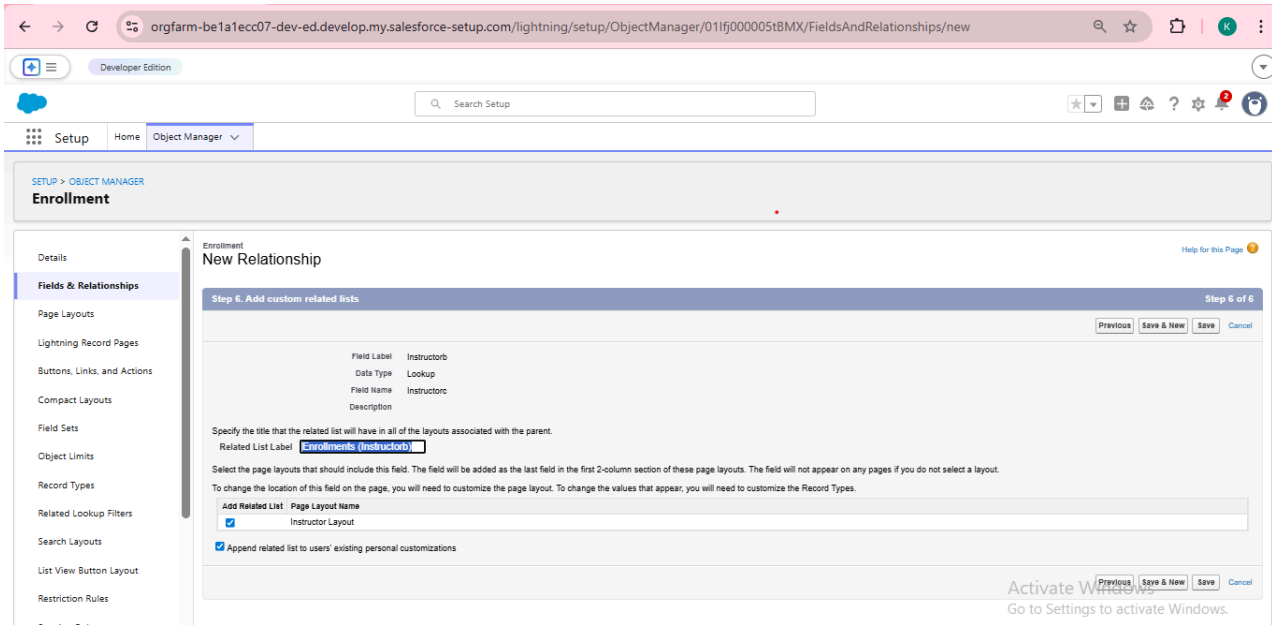
Starts with vowel sound

The Object Name is used when referencing the object via the API.

Object Name Example: Account

Description

Activate Windows
Go to Settings to activate Windows.



Milestone5–ValidationRules:

Activity1:StudentEmailDomainValidation

Purpose:

To ensure students use only allowed institutional email addresses.

Steps:

Go to Setup → Object Manager → Student__c.

Click Validation Rules → New.

Rule Name: Student_Email_Domain_Validation.

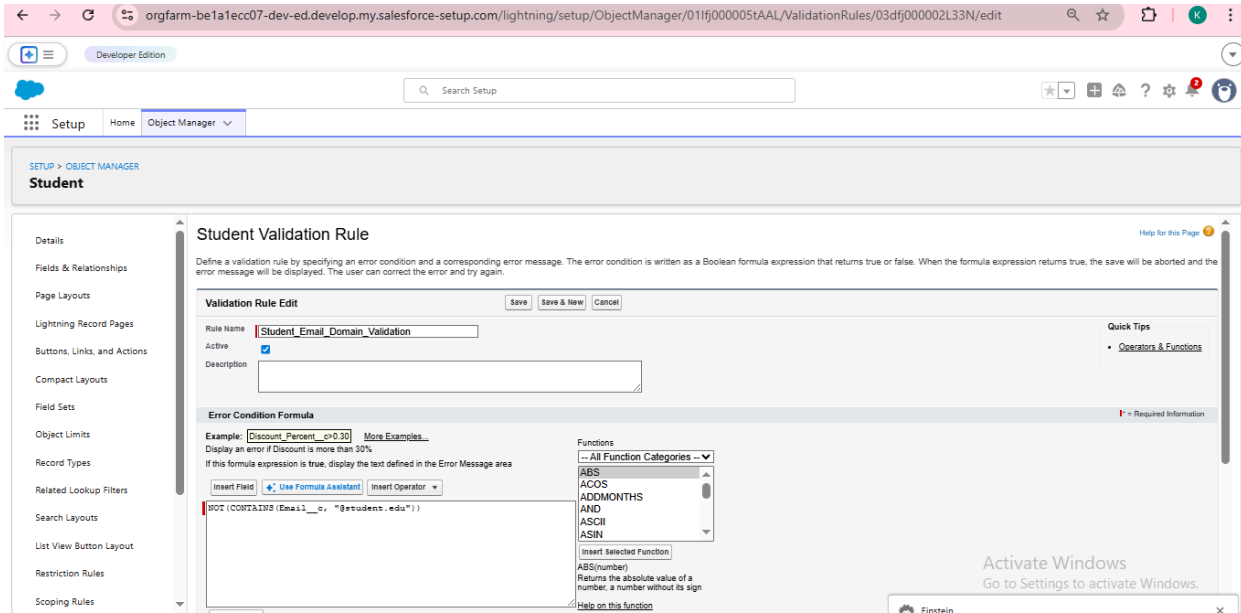
ErrorConditionFormula:(examplefor@student.edu)

- NOT(CONTAINS(Email__c,"@student.edu"))

ErrorMessage:Emailmustbevalidstudentinstitutionalemail. Error

Location: Field → Email_____c.

ClickSave.



Activity2:FeesMustBePaidBeforeApproval

Purpose:

Prevent Enrollment from being approved until fees are repaid. Steps:

1. Goto Setup → Object Manager → Enrollment ____ c.
 2. Click Validation Rules → New.
 3. Rule Name: Fees_Must_Be_Paid_Before_Approval.
 4. Error Condition Formula:


```
AND(
  ISPICKVAL(Enrollment_Status__c, "Approved"),
  Fees_Paid__c = FALSE
)
```
 5. ErrorMessage: Fees must be paid before approving an enrollment.
 6. Error Location: Top of Page (or field as needed).
- Click Save.

The screenshot displays the Salesforce Setup interface for configuring an Enrollment Validation Rule. The rule name is "Fees_Must_Be_Paid_Before_Approval". The error condition formula is: `AND(ISBLANK(Enrollment_Status__c, "Approved"), Fees_Paid__c = FALSE)`. The error message is: "Fees must be paid before approving an enrollment." The error location is set to "Top of Page".

Milestone6–ApprovalProcess

Activity1:CreateApprovalProcessforEnrollmentRequests

Purpose:

RouteEnrollmentrequeststoTrainingManagerforapprovalorrejection.

Steps:

1. GotoSetup→ApprovalProcesses→selectEnrollment_____c.
2. ClickCreateNewApprovalProcess→UseStandardSetup.
3. ProcessName:Approval_Enrollment_Request.
4. EntryCriteria:Enrollment_Status__c="Requested".
5. Saveandconfiguresteps.

The screenshot shows the Salesforce Setup interface. At the top, there are navigation tabs for Setup, Home, and Object Manager. A search bar contains the word "Approval". On the left sidebar, the navigation menu is expanded to "Approval Processes". The main content area is titled "Approval Processes" and includes a "SETUP" icon. Below the title, there is a banner with the text "Get started with Flow Approval Processes in the Approval app where you can manage approval submissions, approval work items, and flow approval processes in one location." and a button labeled "Open Approvals App". A yellow information box contains a list of 7 steps for getting started with approval processes. At the bottom, there is a dropdown menu labeled "Manage Approval Processes For:" with "Enrollment" selected.

Setup Home Object Manager

Approval

Approval Processes

Get started with Flow Approval Processes in the Approval app where you can manage approval submissions, approval work items, and flow approval processes in one location.

Open Approvals App

Approvals are complex business processes that require information gathering and planning before implementing. It is recommended that you follow the instructions below before getting started.

1. [Read the help topic](#)
2. [View the checklist](#)
3. [Create a custom user hierarchical relationship field](#)
4. [Create email templates](#)
5. Create an approval process using either the Jump Start or Standard Wizard
6. Add Approval History Related List to all page layouts
7. Activate the process to deploy to your users

Manage Approval Processes For: Enrollment

The screenshot shows the "Step 1. Enter Name and Description" form in the Salesforce Setup interface. The form is titled "Step 1 of 6" and includes "Next" and "Cancel" buttons. The instructions state "Enter a name and description for your new approval process." The form fields are: "Process Name" with the value "Approval_Enrollment_Request", "Unique Name" with the value "Approval_Enrollment_Request", and "Description" with the value "Approval Process for Enrollment requests== sends to Training Manager". A red asterisk indicates required information.

Approval Processes

Step 1. Enter Name and Description Step 1 of 6

Next Cancel

Enter a name and description for your new approval process.

Enter Name and Description *

Process Name Approval_Enrollment_Request

Unique Name Approval_Enrollment_Request

Description Approval Process for Enrollment requests== sends to Training Manager

Next Cancel

Previous Next Cancel

If only certain types of records should enter this approval process, enter that criteria below. For example, only expense reports from employees at headquarters should use this approval process.

Specify Entry Criteria

Use this approval process if the following :

Field	Operator	Value	
Enrollment: Enrollment Status	equals	Requested	AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		AND

Add Filter Logic...

6. AddApprovalStep:

7. Approver:specificTrainingManageruser.

org-farm-be1a1ecc07-dev-ed.develop.my.salesforce--setup.com/lightning/setup/ApprovalProcesses/page?address=%2Fp%2Fprocess%2FProcessNodeWiz...

Developer Edition

Setup Home Object Manager

Search Setup

Approval Processes

Approval Step Edit
specific Training Manager user.

Step 1 of 3: Enter Name and Description

Save Next Cancel

Enter a name, description, and step number for your new approval step.

Enter Name and Description

Approval Process Name: Approval_Enrollment_Request

Name: specific Training Manager user

Unique Name: specific_Training_Manager_user

Description:

Save Next Cancel

Activate Windows

8. UnderFinalApprovalActions:

9. AddFieldUpdatetosetEnrollment_Status_____c="Approved".

SETUP

Field Updates

Define the field update, including the object associated with the workflow rule, approval process, or entitlement process, the field to update, and the value to apply. Note that the field to update may be on a related object. Fields are shown only for the type that you select.

Field Update Edit

Save Save & New Cancel

Identification

Name: Status to Approved

Unique Name: Status_to_Approved

Description:

Object: Enrollment

Field to Update: Enrollment Status



Field Data Type: Picklist

Re-evaluate Workflow Rules after Field Change:

Required Information

10. UnderFinalRejectionActions:
11. AddFieldUpdatetosetEnrollment_Status_____c="Rejected".

Identification

Name	<input type="text" value="Set Status Rejected"/>
Unique Name	<input type="text" value="Set_Status_Rejected"/> 
Description	<input type="text"/>
Object	Enrollment
Field to Update	<input type="text" value="Enrollment Status"/> ▼
Field Data Type	Picklist
Re-evaluate Workflow Rules after Field Change	<input type="checkbox"/> 

12. AddEmailAlerttosendrejectionemailtostudent.

13. Activatetheapprovalprocess.

Milestone7-Tabs

Activity1:CreatingaTabfortheStudentObject:

Goto Setup.

1. IntheQuickFindbox,typeTabs.
2. UnderCustomObjectTabs,clickNew.
3. SelectStudentfromtheobjectlist.

4. Choose a tab style (icon).
5. Click Next, assign it to all profiles, and then click Save.

The screenshot shows the Salesforce Setup interface for creating a new custom object tab. The navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar contains the text 'tabs'. The left sidebar shows 'User Interface' with 'Rename Tabs and Labels' and 'Tabs' options. The main content area is titled 'SETUP Tabs' and contains the following fields:

- Choose the custom object for this new custom tab. Fill in other details.
- New Custom Object Tab** (Required Information)
- Select an existing custom object or [create a new custom object now](#).
- Object: Student
- Tab Style: Airplane
- (Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.
- Splash Page Custom Link: --None--
- Enter a short description.
- Description: [Empty text box]

Activity 2: Creating Remaining Tabs

Now create the tabs for the remaining objects, they are "Courses, Instructors, Enrollment".

Follow the same steps as mentioned in Activity-1

Milestone 8 - The Lightning App

Activity 1: Creating the Lightning Application for the Project

Navigate to Setup.

Enter App Manager in the Quick Find search box. Click New Lightning App.

Provide the app details:

AppName: Student Enrollment & Course Management Developer

Name auto-fills

Description: Optional, but can describe the purpose of the app. Click Next and choose the following settings:

App Branding (optional): Add logo or select color theme Navigation

Style: Choose Standard Navigation

Click Next and assign the app to specific profiles (e.g., **Admin, Enrollment Officer, Instructor**).

Click Next to select the items (tabs) to be included in the app. Add:

1. Student
2. Course
3. Instructor
4. Enrollment
5. Reports
6. Dashboards
7. Home
8. Tasks

Click Next and review the app settings. Finally, click Save & Finish.

The screenshot shows the Salesforce Lightning Experience App Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main header area displays 'Lightning Experience App Manager' with buttons for 'New Lightning App' and 'New External Client App'. Below the header, there is a search bar and a list of apps. The 'App Manager' app is selected in the left sidebar. The main content area shows a table with 31 items, sorted by App Name. The table has columns for App Name, Developer Name, Description, Last Modified, App Type, and Visibility. The following table represents the data shown in the screenshot:

App Name	Developer N...	Description	Last Modified ...	App ...	Visi...
1 Agentforce Grid	AIWorkbench	Access Agentforce Grid	2/24/2026, 9:47 AM	Lightning	✓
2 Agentforce Studio	AgentforceStudio	Agentforce Studio	2/24/2026, 9:47 AM	Lightning	✓
3 All Tabs	AllTabSet		2/24/2026, 9:38 AM	Classic	

New Lightning App

App Details & Branding

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

App Details

*App Name ⓘ

Student Enrollment & Course Management

*Developer Name ⓘ

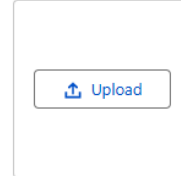
Student_Enrollment_Course_Management

Description ⓘ

Enter a description...

App Branding

Image ⓘ



Primary Color Hex

Value ⓘ

#0070D2

Org Theme Options



Next

New Lightning App

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Create ▼

Type to filter list...

- Accounts
- Action Hub
- Activation Targets
- Activations

Selected Items

Students

Back



N

New Lightning App

User Profiles

Choose the user profiles that can access this app.

Available Profiles

Selected Profiles

System Administrator

ack



Save & Finish

Milestone9–PageLayoutsandRecord:

Activity1:CreateRecordTypesforEnrollment:

Purpose:

To separate New Enrollment vs Re-Enrollment processes.

Steps:

1. Goto Setup → Object Manager → Enrollment__c.

2. Click Record Types → New.
3. For the first record type:
 - Existing Record Type: Master
 - Record Type Label: New Enrollment
 - Enable for required profiles (Admin, Enrollment Officer).
 - Select the default Page Layout → Save & New.
4. For the second record type:
 - Record Type Label: Re-Enrollment
 - Enable for same profiles.
 - Select the default Page Layout → Save.

Activity 2: Separate Page Layouts Create users for Each Enrollment Record Type

The screenshot shows the Salesforce Setup interface in Developer Edition. The browser address bar displays the URL: `orgfarm-be1a1ecc07-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/011fj000005t8MX/RecordTypes/012fj000004VGMLAA4/edit`. The page title is "Enrollment" under "OBJECT MANAGER". The left sidebar contains a navigation menu with "Record Types" selected. The main content area is titled "Edit Record Type" and "Re-Enrollment". Below the title, there is a prompt: "Enter a new name for the selected record type and click Save." The "Record Type" form includes the following fields: "Record Type Label" (Re-Enrollment), "Record Type Name" (Re_Enrollment), "Namespace Prefix" (empty), "Description" (empty), and "Active" (checked). "Save" and "Cancel" buttons are at the bottom of the form. A "Required Information" icon is visible in the top right of the form area. At the bottom right of the page, there is a message: "Activate Windows Go to Settings to activate Windows."

Enrollment

- 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
- Scoping Rules

Edit Record Type
New Enrollment [Help for this Page](#)

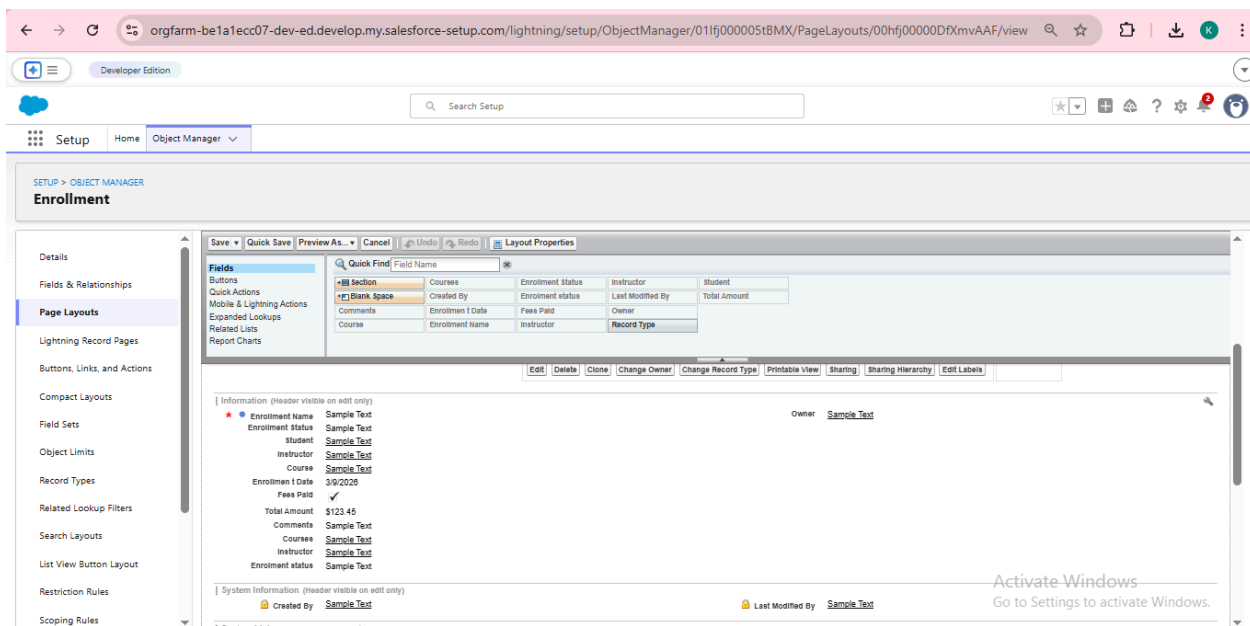
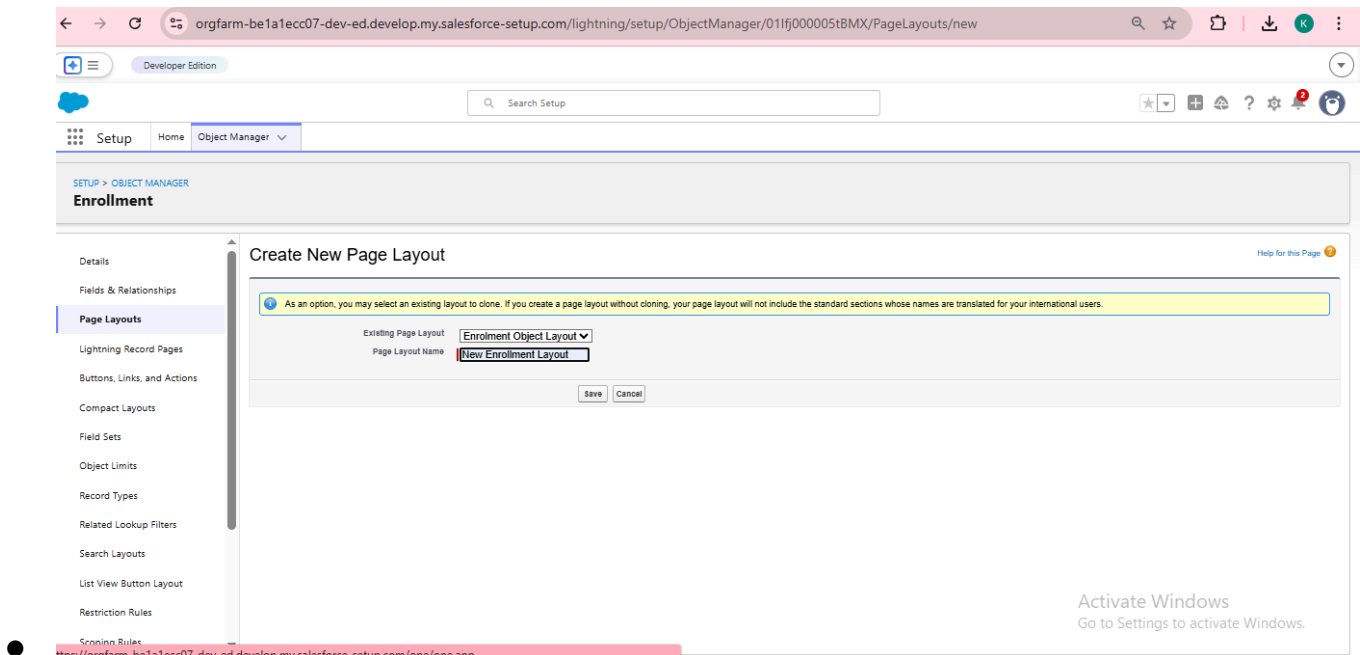
Enter a new name for the selected record type and click Save.

Record Type	
Record Type Label	<input type="text" value="New Enrollment"/>
Record Type Name	<input type="text" value="New_Enrollment"/>
Namespace Prefix	<input type="text"/>
Description	<input type="text"/>
Active	<input checked="" type="checkbox"/>

Purpose:

To show different fields for New Enrollment vs Re-Enrollment. Steps:

1. Enrollment__object → click Page Layouts.
2. Click New to create a layout for New Enrollment:
 - Clone from existing Enrollment Layout → name it New Enrollment Layout → Save.
 - Keep fields relevant for first-time enrollment: Student, Course, Instructor, Enrollment Date, Status, Fees, etc.
 - Remove unnecessary re-enrollment fields if any.
 - Click Save.



3. Again, click New to create Re-Enrollment Layout:

- Clone existing layout → name it Re-Enrollment Layout → Save.
- Include fields like previous Enrollment reference (if any), Re-Enrollment Reason, Status, Fees, etc.
- Click Save.

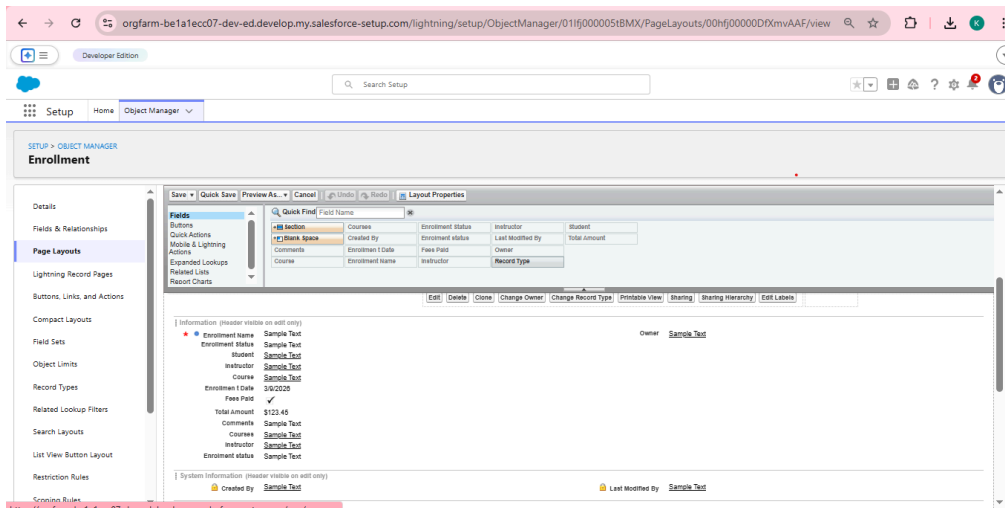
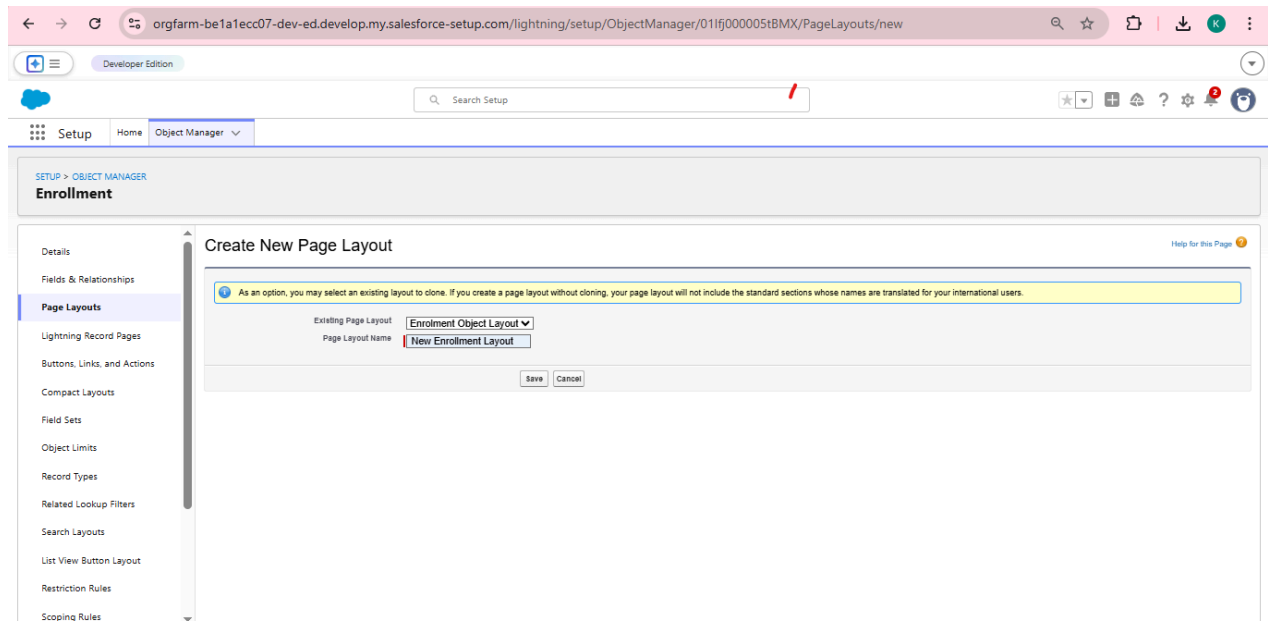
4. Click Page Layout Assignment → Edit Assignment.

5. For each profile and record type:

- New Enrollment → use New Enrollment Layout.

6. Re-Enrollment → use Re-Enrollment Layout.

Click Save.



Activity3:EnrollmentRAddelatedListstoParentObjects:

Purpose:

To see all Enrollments from Student, Course, and Instructor records.

a) Student → Enrollment Related List

Go to Setup → Object Manager → Student__c.

Click Page Layouts → open the main layout.

In the palette, select Related Lists.

Drag Enrollments related list to the layout.

Customize columns if needed → Save.

b) Course → Enrollment Related List

Go to Object Manager → Course__c → Page Layouts.

Open layout → add Enrollments related list.

Save.

c) Instructor → Enrollment Related List

Go to Object Manager → Instructor__c → Page Layouts.

Open layout → add Enrollments related list.

Save.

Milestone10–AutomationwithFlow

Activity1:Auto-PopulateEnrollmentDate(Record-TriggeredFlow)

Purpose:

Set Enrollment__Date__c to today when Enrollment is created.

Steps:

1. GotoSetup→Flows→New Flow.
2. SelectRecord-TriggeredFlow→Create.
3. Object:Enrollment__c.
4. Trigger:Arecordiscreated.
5. OptimizeforActionsandRelatedRecords→Done.

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

* Object

Enrollment

Configure Trigger

Trigger the Flow When:

- A record is created
- A record is updated
- A record is created or updated
- A record is deleted

Set Entry Conditions


Specify entry conditions to reduce the number of records that trigger the flow and the number of records that trigger the flow to conserve your org's resources.


Condition Requirements

None

6. AddUpdateRecordselement:
 - Label:SetEnrollmentDate.
 - Userrecordthattriggeredtheflow.
 - Field:Enrollment_Date__c→Value:\$Flow.CurrentDate.

7. SaveasSetEnrollmentDate→Activate.

 Update Records

* Label * API Name 

Description

*** How to Find Records to Update and Set Their Values**

- Use the enrollment record that triggered the flow
- Update records related to the enrollment record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record

Set Field Values for the Enrollment Record

Field	Value
<input type="text" value="Enrollment_Date"/>	<input type="text" value="Running Flow Interview > CurrentDate"/>

Activity2:SendEmailWhenEnrollmentApproved


Purpose:

When Enrollment_Status__c=Approved,sendemail,alertadmin,andset Student to Active.

Steps(highlevelfordocument):

1. CreateEmailTemplateandEmailAlertforstudentnotification.

2. GotoFlows→New→Record-TriggeredFlow.
3. Object:Enrollment__c,Triggeronupdate.
4. Condition:Enrollment_Status__cEquals"Approved".

 Configure Start

Enrollment

Configure Trigger

Trigger the Flow When:

- A record is created
- A record is updated
- A record is created or updated
- A record is deleted



Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

All Conditions Are Met (AND) ▼

Field	Operator	Value	
 Enrollment Status x	Equals ▼	Aa Approved x	

5. AddGetRecordstofetchrelatedStudent_____c.

Get Records

* Label:

* API Name:

Description:

Get Records of This Object

* Data Source

Data Cloud Object

Salesforce Object

* Object:

Filter Student Records

Condition Requirements:

Field	Operator	Value
<input type="text" value="Status"/>	<input type="text" value="Equals"/>	<input type="text" value="Active"/>

[+ Add Condition](#)

6. AddAction(EmailAlert)tosendemailtostudent.




Enrollment Approved EmailAlert



* Label

Send Student Email


* API Name 


Send_Student_Email

Description

Use values from earlier in the flow to set the inputs for the "Enrollment Approved EmailAlert" email alert. To use its outputs later in the flow, store them in variables.

Set Input Values

A_a * Record ID 

A_a Triggering Enrolement__c > Record ID 

7. AddActiontonotifyAdminifneeded.

8. AddUpdateRecordstosetStudent____c.Status__c="Active".

Update Records
✕

*** Label**

*** API Name** ⓘ

Description

*** How to Find Records to Update and Set Their Values**

- Use the enrollment record that triggered the flow
- Update records related to the enrollment record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record

None—Always Update Record
▼

Set Field Values for the Enrollment Record

Field	Value	
Aa <input style="width: 90%; border: none; border-bottom: 1px solid #ccc; padding: 2px 5px;" type="text" value="Comments"/> ✕	<input style="width: 90%; border: none; border-bottom: 1px solid #ccc; padding: 2px 5px;" type="text" value="True"/> ✕	←
		<div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">✕</div>

9. SaveasEnrollmentApprovedAction→Activate.

Activity3:Auto-AssignInstructorBasedonCourseCategory

Purpose:

AssignInstructorautomaticallybasedonCoursecategory.

Rules:

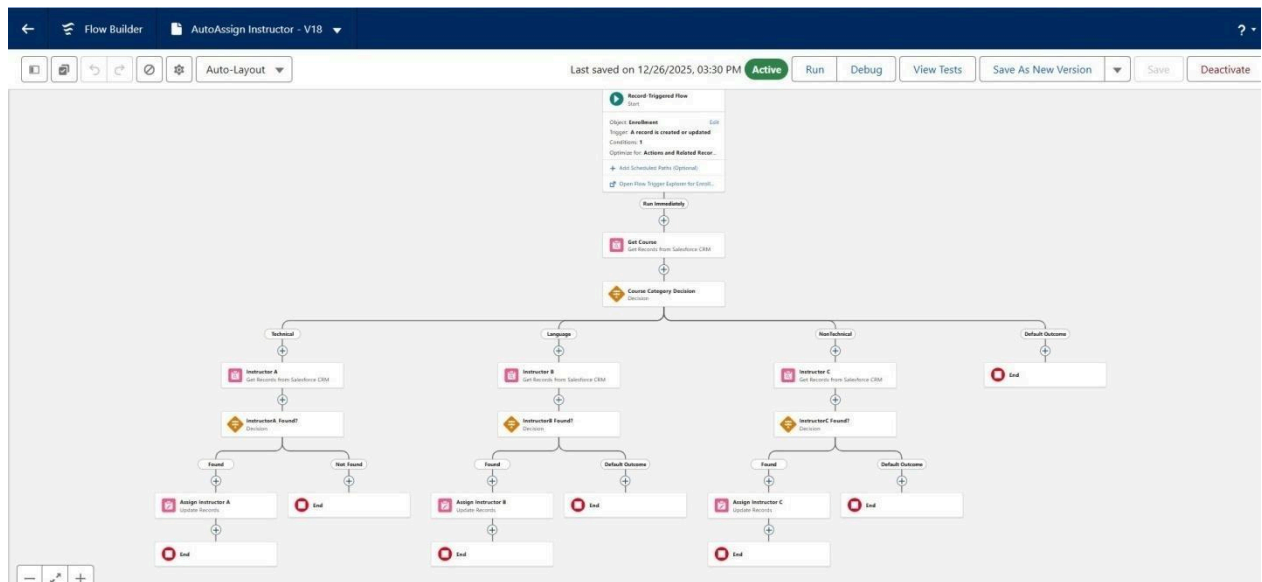
Technical → Instructor A

Language → Instructor B

Non-Technical→InstructorC

Steps:

1. Prepare Instructor__crecordswithcodes(e.g.,INSTR_A,INSTR_B, INSTR_C).
2. CreateRecord-TriggeredFlowonEnrollment____c(created/updated).
3. AddGetRecordstofetchrelatedCourse____c.
4. AddDecisionelementonCourse____r.Category__c:
5. Technical/Language/Non-Technicaloutcomes.
6. Foreachoutcome:
7. GetInstructorrecord(A/B/C)basedoncodeorname.
8. UpdateRecords:setInstructor____c=Get_Instructor_X.Id.
9. SaveasAutoAssign_Instructor→Activate.




Activity4:Auto-CreateTaskforFollow-Up

Purpose:

WhenEnrollmentis“Requested”,createafollow-uptaskforAdmin.

Steps:

1. CreateRecord-TriggeredFlowonEnrollment_____c.
2. Trigger:Recordcreatedorupdated;conditionEnrollment_Status_____c
="Requested".

 Configure Start ✕

* Object
Enrollment

Configure Trigger

Trigger the Flow When:

A record is created

A record is updated

A record is created or updated


A record is deleted

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements
All Conditions Are Met (AND) ▼

Field	Operator	Value	
<input type="text" value="Enrollment Status"/> ✕	<input type="text" value="Equals"/> ▼	<input type="text" value="Requested"/> ✕	

3. AddGetRecordstofetchAdminuser(byUsername/Profile).
4. AddFormula(Date):TODAY()+2forduedate.



Get Records



Get Admin User

Get_Admin_User

Description

Get Records of This Object

*** Data Source**

Data Cloud Object

Salesforce Object

*** Object**

User

Filter User Records

Condition Requirements

All Conditions Are Met (AND) ▼

Field

A Username x

Operator

Equals ▼

Value

geespam198430@agentforce.com 🔍



+ Add Condition

5. AddCreateRecordsfor Task:
6. Subject:Followuponenrollmentrequest.
7. OwnerId:Adminuser Id.
8. WhatId:EnrollmentId.
9. ActivityDate:formulaToday+2.

Create a Record of This Object

* Object

Task

Set Field Values for the Task

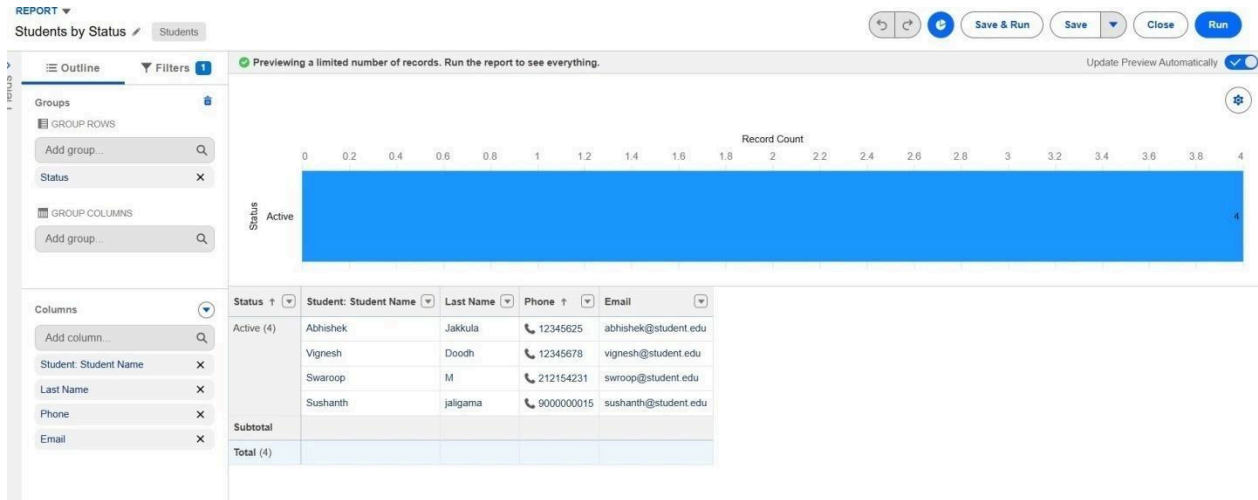
Field	←	Value	
Due Date Only ×		DueDateTodayPlus2 ×	
Field	←	Value	
Description ×		Auto-created follow-up for enrollment request	
Field	←	Value	
Priority ×		Normal ×	
Field	←	Value	
Status ×		Not Started ×	
Field	←	Value	
Subject ×		Follow up on enrollment request	
Field	←	Value	
Related To ID ×		Triggering Enrolement_c > Record ID ×	

10. SaveasCreate_Followup_Task→Activate.

Milestone11–Reports:

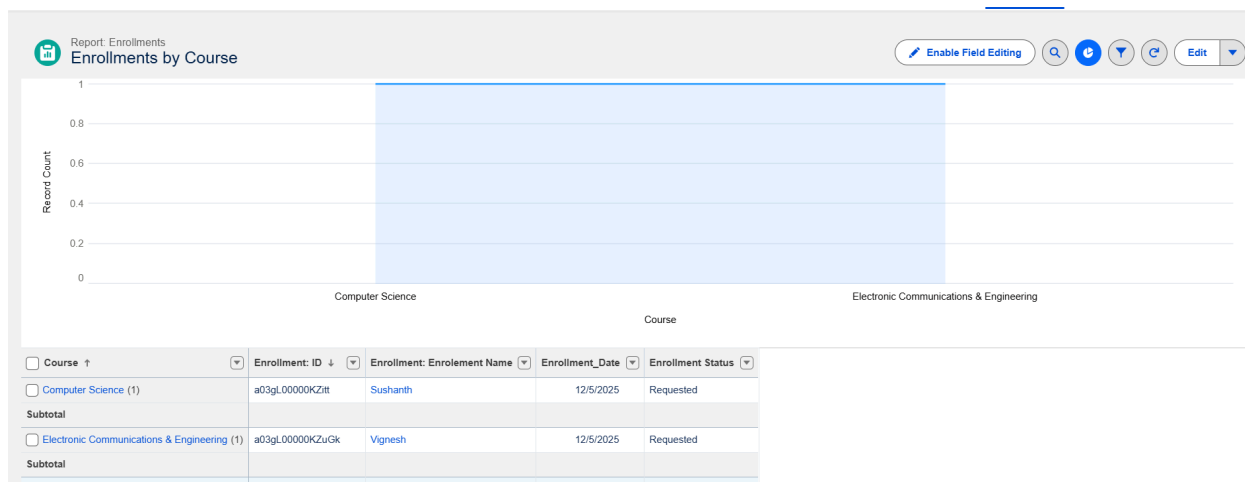
Activity1:Create“StudentsbyStatus”Report

1. Goto**Reports**→**NewReport**.
2. ReportType:*Students*.
3. Grouprowsby*Status_c*.
4. Addcolumnsasneeded(Name,Email,Phone).
5. Saveas*StudentsbyStatus*.



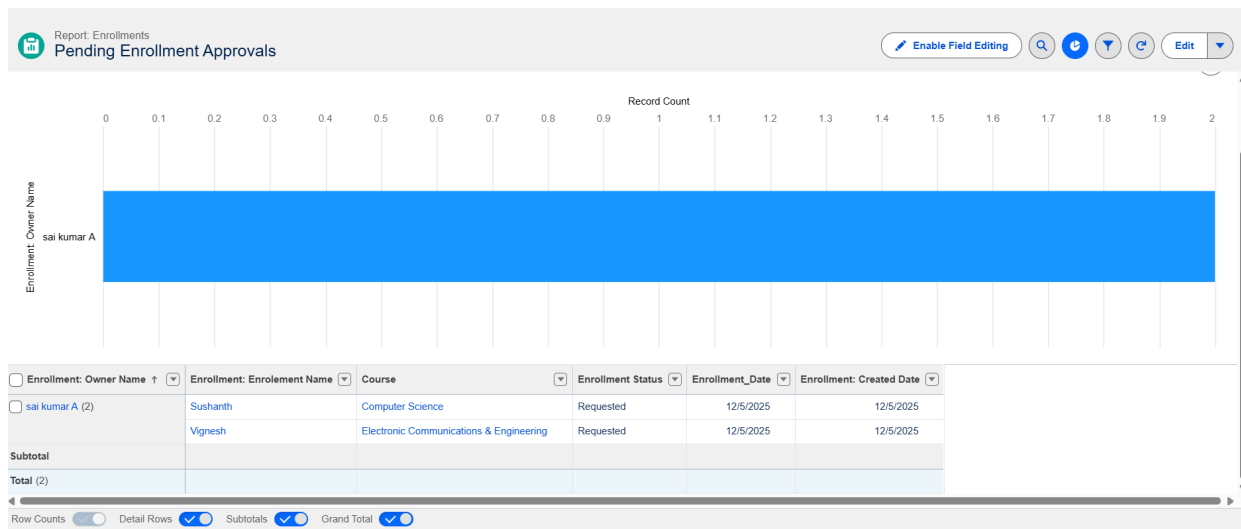
Activity2:Create“EnrollmentsbyCourse”Report

1. NewReport→ReportType:Enrollment
2. Grouprows by Course__r.Name.
3. ShowRecordCountpercouse.
4. SaveasEnrollmentsbyCourse



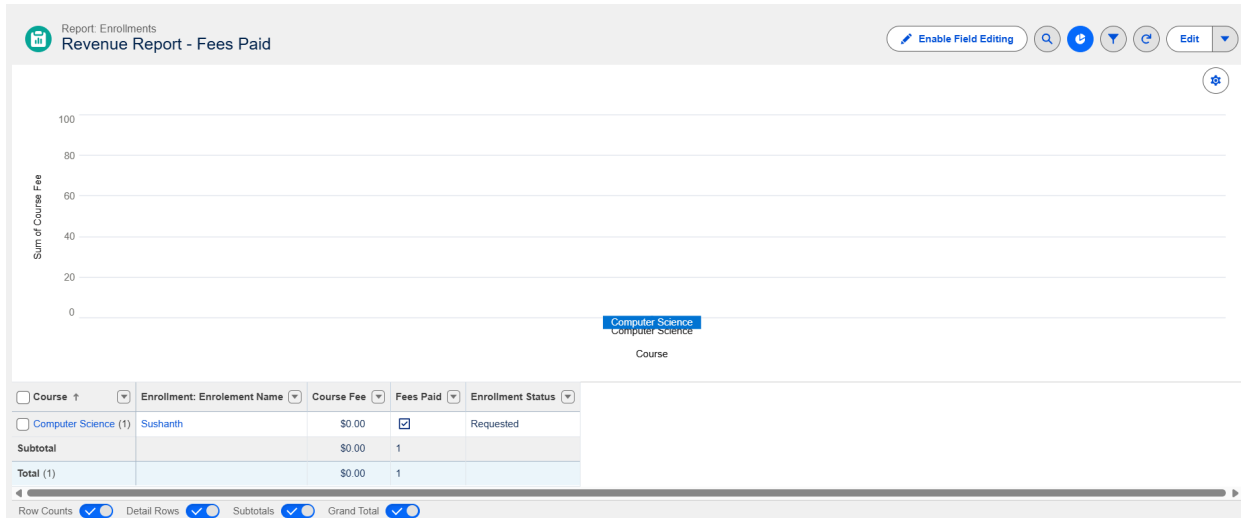
Activity3:Create“PendingEnrollmentApprovals”Report

1. NewReport→Enrollmentsreporttype.
2. Filter:Enrollment_Status__c="Requested".
3. Addcolumns (Student, Course, Owner, CreatedDate).
4. SaveasPendingEnrollmentApprovals.



Activity4:Create“RevenueReport(FeesPaid)”Report

1. NewReport→Enrollmentsreporttype.
2. Filter:Fees_Paid_c=TRUE.
3. GroupbyCourse_r.Name.
4. SummarizeCourse_Fee_corsimilarfeefield.
5. SaveasRevenueReport-FeesPaid.

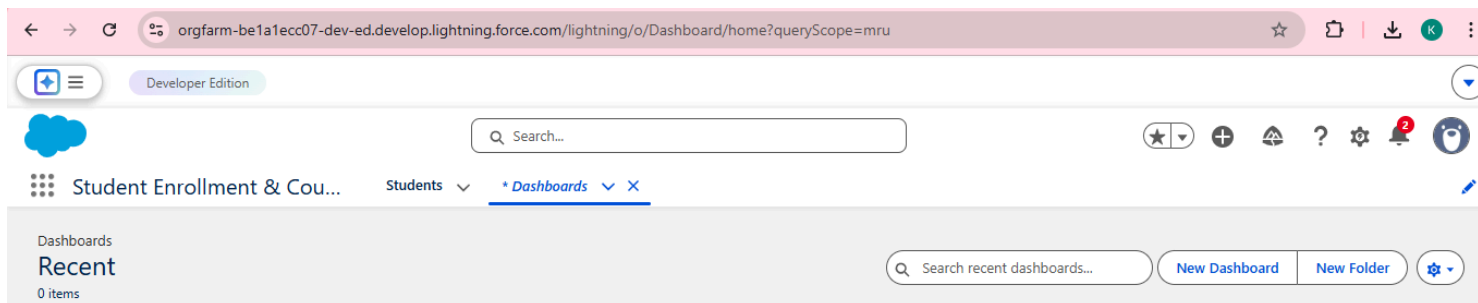


Milestone12–Dashboards:

Activity1:Create“StudentManagemetDnashboard”

Steps :

1. Goto**Dashboards**→**NewDashboard**.
2. Name :**StudentManagementDashboard**.



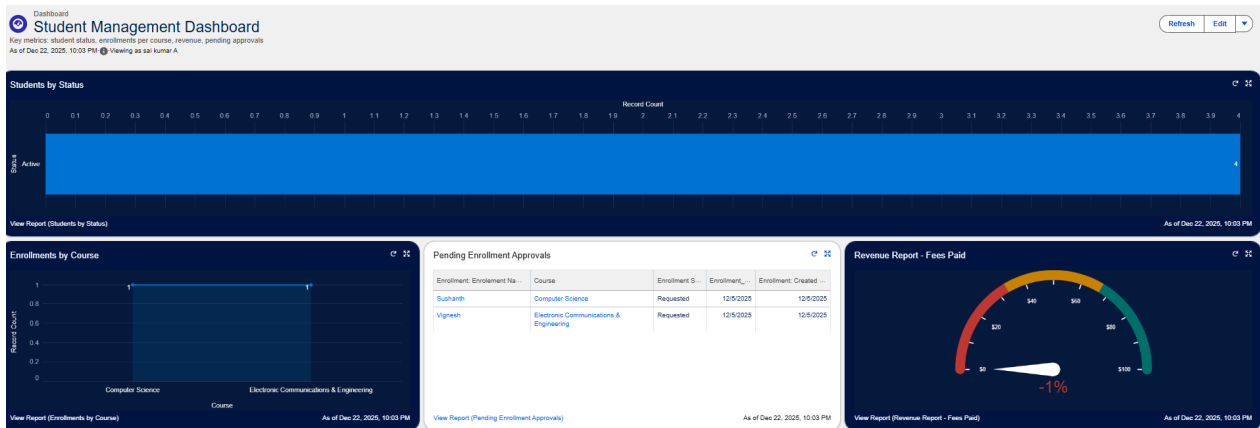
3. Selectfolder→**Create**.
4. SelectnewDashboardfolderandSave.

Addcomponents :

- **DonutChart**→using**StudentsbyStatus**report.
- **BarChart**→using**EnrollmentsbyCourse**report.

- **Gauge**→ using `Revenue Report-FeesPaid` (sum of revenue field).
- **Table**→ using `Pending Enrollment Approvals` report.

4. Arrange components on canvas.



Click **Save** → **Done**.

Milestone 13 – Security Setup

Activity 1: Configure Profiles

):

1. Go to **Setup** → **Profiles** → **New** (or **Clone Profiles**

Required:

- **Admin** – Full access.
- **Enrollment Officer** – Create/Update Enrollments.
- **Instructor** – Read-only access to assigned Enrollments.

2. Steps (example for Enrollment Officer fName :

Enrollment Officer.

3. In **ObjectSettings** :

- o Enrollment__c: Read, Create, Edit.

- o Student__c / Course__c: Read (and Edit if required).

4. Save.

Repeats similarly for **Instructor** profile with **Read-Only** on Enrollments

Activity 2: Create Permission Sets

1. Goto **Setup** → **Permission Sets** → **New**.

2. Create **Custom Object Access** (additional access to project objects if needed).

3. Create **Dashboard Editor** (Create/Customize Reports & Dashboards permissions).

4. Assign to relevant users.

Activity 3: Configure Sharing & OWD

Steps:

1. Goto **Setup** → **Sharing Settings**.

2. Set **Organization-Wide Defaults**:

- **Students (Student__c)** :PublicRead/Write.
- **Courses (Course__c)** :PublicRead/Write.
- **Enrollments (Enrollment__c)** :Private.

3. Save.

For Instructor access to assigned Enrollments, optionally:

- Add a **Userlookup** on Enrollment (Instructor User).
- Use a simple **Record-Triggered Flow** to set **OwnerId= Instructor User** so they can see only their own records.

7. FUNCTIONAL & PERFORMANCE TESTING

During the testing phase, screenshots were captured from the Salesforce user interface to verify the proper functioning of all implemented features of the Student Enrollment & Course Management System (SECMS). These include student, course, instructor, and enrollment record creation; validation rule checks for mandatory fields, eligibility, and enrollment constraints; verification of record type behavior for different course categories; flow execution for automated enrollment approvals and seat availability checks; and Apex trigger functionality to ensure automatic confirmation and record updates. Additionally, screenshots of generated reports and dashboards were documented to validate accurate data visualization, enrollment tracking, and performance monitoring. Submit the Screenshots while testing.

8. ADVANTAGES & DISADVANTAGES

Advantages

- Centralized data
- Automated workflows
- Real-time reporting

Disadvantages

- Salesforce dependency
- Requires initial configuration effort

9. Conclusion:

The Salesforce Student Enrollment & Course Management System successfully streamlines the entire student academic lifecycle—from student registration and course creation to enrollment management, instructor assignment, approvals, revenue tracking, and reporting. By leveraging Salesforce’s automation tools such as validation rules, flows, approval processes, and dashboards, the system improves data accuracy, operational efficiency, and decision-making. This project demonstrates how Salesforce can transform manual academic processes into a scalable, reliable, and user-friendly digital solution.

10. FUTURE SCOPE

- Payment gateway integration
- Student self-service portal
- Analytics-driven insights

- Mobileaccesssupport

11. APPENDIX

- SourceCode:SalesforceConfiguration
- Dataset:SalesforceObjects
- GitHub/Demo:SalesforceDeveloperOrg

Thank You

