

Salesforce-Based Certificate Request and Student Management System

-Hari Chandana Pedapudi

Project Overview

The Campus Course Enrollment & Student Support CRM is a Salesforce-based application developed to manage student information, course enrollments, and certificate issuance in an educational institution.

The CRM centralizes academic data, automates key processes such as enrollment completion and certificate generation, and provides insights through reports and dashboards.

The system addresses business needs such as reducing manual work, improving data accuracy, streamlining academic operations, and enabling management to monitor enrollments and certifications effectively.

Key Features:

- Custom objects for Students, Courses, Enrollments, and Certificates
- Automated workflows for certificate generation
- Reports and dashboards for academic analysis

Objectives

The primary objective of building the Campus Course Enrollment & Student Support CRM is to automate and streamline academic processes using Salesforce.

The project aims to replace manual enrollment tracking and certificate handling with a centralized, automated CRM solution.

By implementing this CRM, the institution can achieve better student data management, faster certificate issuance, improved operational efficiency, and enhanced decision-making through real-time reports and dashboards.

Phase 1: Problem Understanding & Industry Analysis

1. Requirement Gathering

Educational institutions often face difficulties in managing student enrollments and certifications using manual systems. The following requirements were identified for the CRM:

Functional Requirements

- Maintain detailed student records
- Manage course and department information
- Track student enrollments and course completion status
- Automatically generate certificate requests after course completion
- Issue certificates based on approval status
- Generate analytical reports and dashboards

Non-Functional Requirements

- Secure and scalable CRM architecture
- Role-based access control
- User-friendly Salesforce Lightning interface
- Reliable automation using declarative tools

2. Stakeholder Analysis

Stakeholder	Role	Needs & Expectations
Student	Learner	Easy enrollment and timely certificates
Admin	System manager	Automation and accurate data
Faculty / Academic Office	Course oversight	Enrollment and completion tracking

Management	Decision makers	Reports and dashboards
System Administrator	CRM maintenance	Security and scalability

3. Business Process Mapping

Current Process (Manual)

- Student data maintained in spreadsheets
- Manual enrollment tracking
- Manual certificate issuance
- Limited reporting and visibility

Proposed Process (CRM-Based)

- Centralized student and course database
- Automated enrollment and certificate workflows
- Real-time tracking of enrollment status
- Reports and dashboards for academic insights

4. Industry-Specific Use Case Analysis

Educational institutions require CRM systems to manage the complete student lifecycle efficiently. The following use cases were identified:

Use Case	Description	Salesforce Feature
Student Enrollment	Register students to courses	Custom Objects
Course Completion	Track academic progress	Picklists
Certificate Issuance	Automated certification	Salesforce Flows
Academic Reporting	Analyze enrollment data	Reports & Dashboards

5. AppExchange Exploration

The Salesforce AppExchange provides solutions that can enhance the CRM in the future, such as:

- Salesforce Education Cloud for advanced student lifecycle management
- Data Loader for bulk data import and export
- Email automation tools for student notifications
- Analytics applications for deeper insights

Phase 2: Org Setup & Configuration

Objective

The objective of this phase is to configure the Salesforce environment required to implement the Campus Course Enrollment & Student Support CRM.

This phase ensures that organizational settings, security policies, and user access are properly set up before proceeding with data modeling and automation.

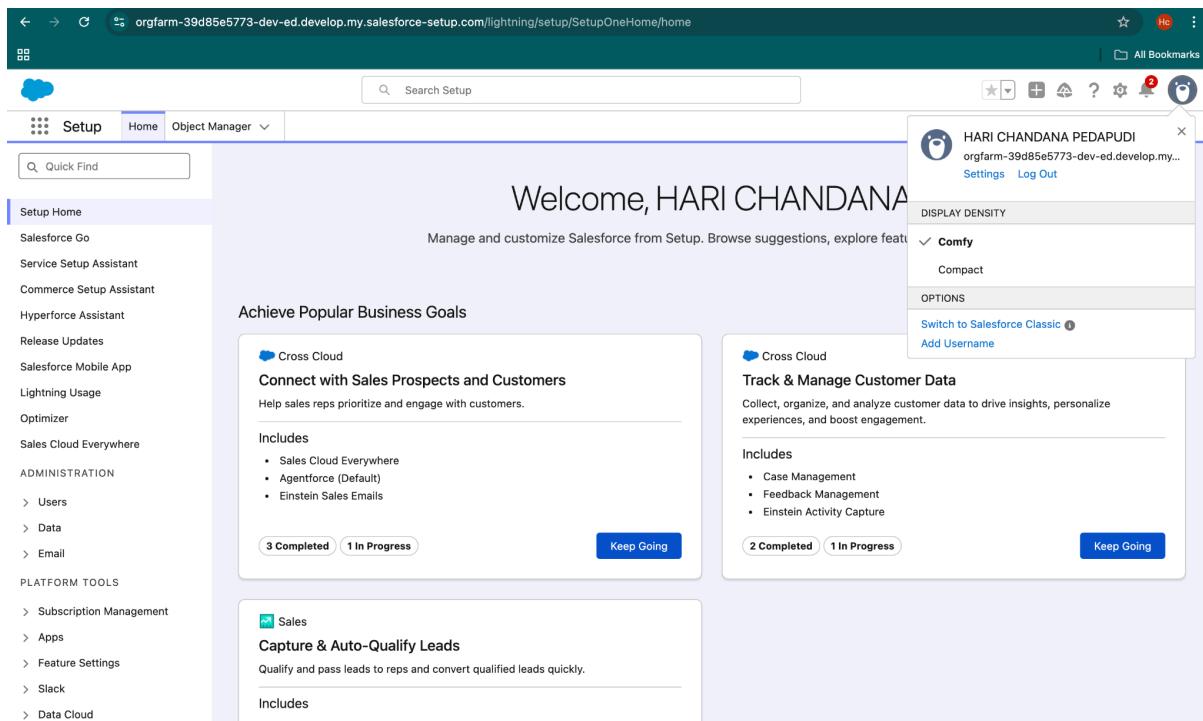
1. Salesforce Editions

Use Case

Salesforce Editions define the features and capabilities available for CRM development.

For this project, Salesforce Developer Edition is used to build and test the CRM functionalities such as custom objects, flows, reports, and dashboards. Developer Edition is suitable for academic and learning purposes.

For real-world institutional deployment, Enterprise Edition is recommended due to its advanced security and scalability features.



2. Company Profile Setup

Company Information is configured to define the organization name, locale, currency, and system preferences required for CRM operations.

Configured Details (as per Developer Org)

- Organization Name: NA
- Organization Edition: Salesforce Developer Edition
- Default Language: English
- Default Locale: English (United States)
- Time Zone: (GMT-08:00) Pacific Standard Time
- Currency Locale: USD

These settings confirm that the project is developed in a Salesforce Developer Org, which is suitable for academic and learning purposes.

The screenshot shows the Salesforce Setup Company Information page. On the left, there's a sidebar with navigation links for Company Settings (Business Hours, Calendar Settings, Public Calendars and Resources), Company Information (Data Protection and Privacy, Fiscal Year, Holidays, Language Settings, My Domain), and a search bar for 'company'. The main content area has a title 'Company Information' with a 'NA' icon. It displays the organization's profile with fields like Organization Name (NA), Primary Contact (OrgFarm EPIC), Division (United States), and Fiscal Year Starts In (January). There are also sections for Activate Multiple Currencies, Enable Data Translation, Newsletter (checked), Admin Newsletter (checked), Hide Notices About System Maintenance, Hide Notices About System Downtime, Locale Formats (ICU), and various system metrics like API Requests, Streaming API Events, and Restricted Logins. At the bottom, it shows 'Created By' as OrgFarm EPIC on 12/1/2025, 3:04 AM and 'Modified By' as OrgFarm EPIC on 12/9/2025, 12:48 AM.

3. Business Hours & Holidays

Use Case

Business Hours define the operational working schedule of the institution and can be used in approvals and future automations.

Configured Values

- Business Hours: Monday – Friday, 9:00 AM to 5:00 PM
- Holidays: National holidays such as Republic Day, Independence Day, Diwali, and Christmas

4. Fiscal Year Settings

Use Case

Fiscal Year settings align reporting and administrative processes with financial and academic standards followed in India.

Configured Value

- Fiscal Year Type: Standard
- Duration: April – March

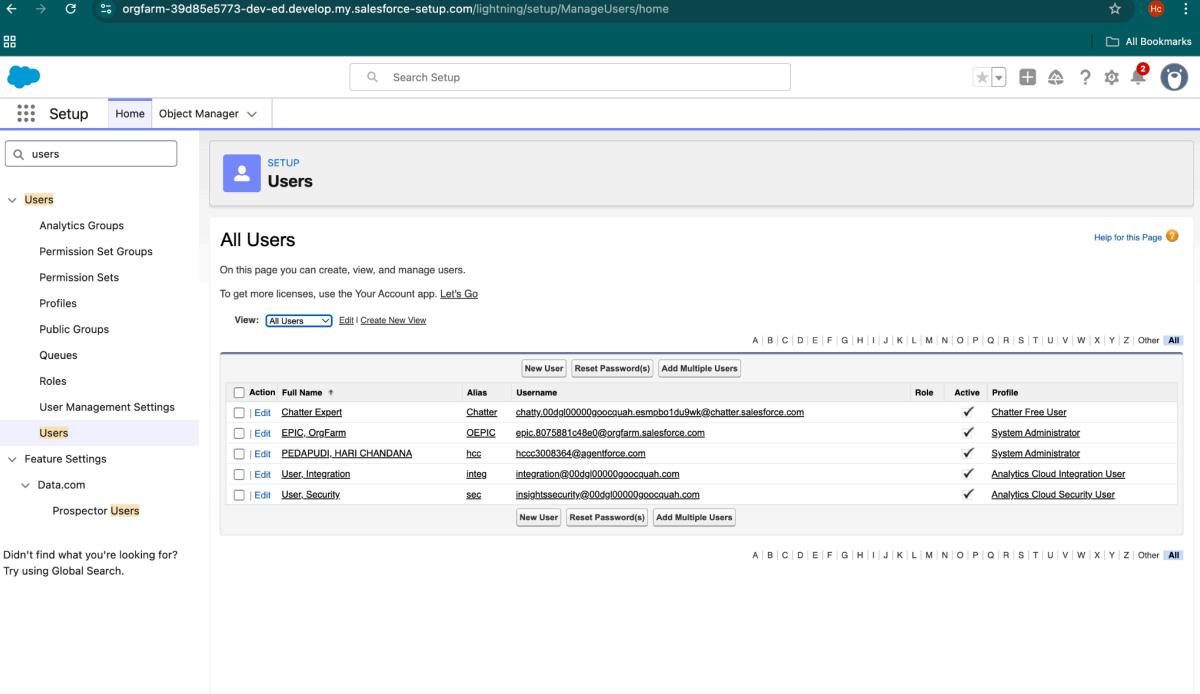
5. User Setup & Licenses

Use Case

User setup defines who can access the CRM and what level of functionality they can use. For this project, users are created to represent administrative and operational roles.

Users Configured

- System Administrator – Full access
- Standard User – Limited access to records



The screenshot shows the Salesforce Setup interface with the 'Users' page selected. The left sidebar includes links for Analytics Groups, Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, and a highlighted 'Users' section. The main content area displays a table of users with columns for Action, Full Name, Alias, Username, Role, Active, and Profile. The table lists six users:

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatty.00dg00000goccauh.esmbo1du9wk@chatter.salesforce.com	✓	✓	Chatter Free User
<input type="checkbox"/> Edit	EPIC_OnFarm	OEPI	oepl.8075881c48e0@orfarm.salesforce.com	✓	✓	System Administrator
<input type="checkbox"/> Edit	PEDAPUDI, HARI CHANDANA	hcc	hcc3008364@agentforce.com	✓	✓	System Administrator
<input type="checkbox"/> Edit	User_Integration	integ	integration@00dg00000goccauh.com	✓	✓	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User_Security	sec	insightssecurity@00dg10000goccauh.com	✓	✓	Analytics Cloud Security User

6. Login Access Policies

Use Case

Login Access Policies help secure the CRM by controlling session behavior and access rules.

Configured Settings

- Session Timeout: 30 minutes
- Login IP Relaxation: Enabled
- Multi-Factor Authentication: Enabled for administrators

7. Developer Org Setup

Use Case

The entire project is developed and tested in a Salesforce Developer Org, which provides a safe environment for learning and demonstration.

8. Sandbox Usage

Use Case

In real-world projects, sandboxes are used to test changes before deployment. For this academic project, sandbox usage is explained conceptually as future practice.

9. Deployment Basics

Use Case

Deployment refers to moving configurations between environments using tools like Change Sets.

For this project, deployment is explained conceptually to demonstrate understanding of Salesforce release management.

Phase 3: Data Modeling & Relationships

Objective

The objective of this phase is to design and implement an efficient data model for the Campus Course Enrollment & Student Support CRM.

This phase focuses on creating custom objects, defining fields, configuring layouts, and establishing appropriate relationships to support academic enrollment and certificate issuance processes.

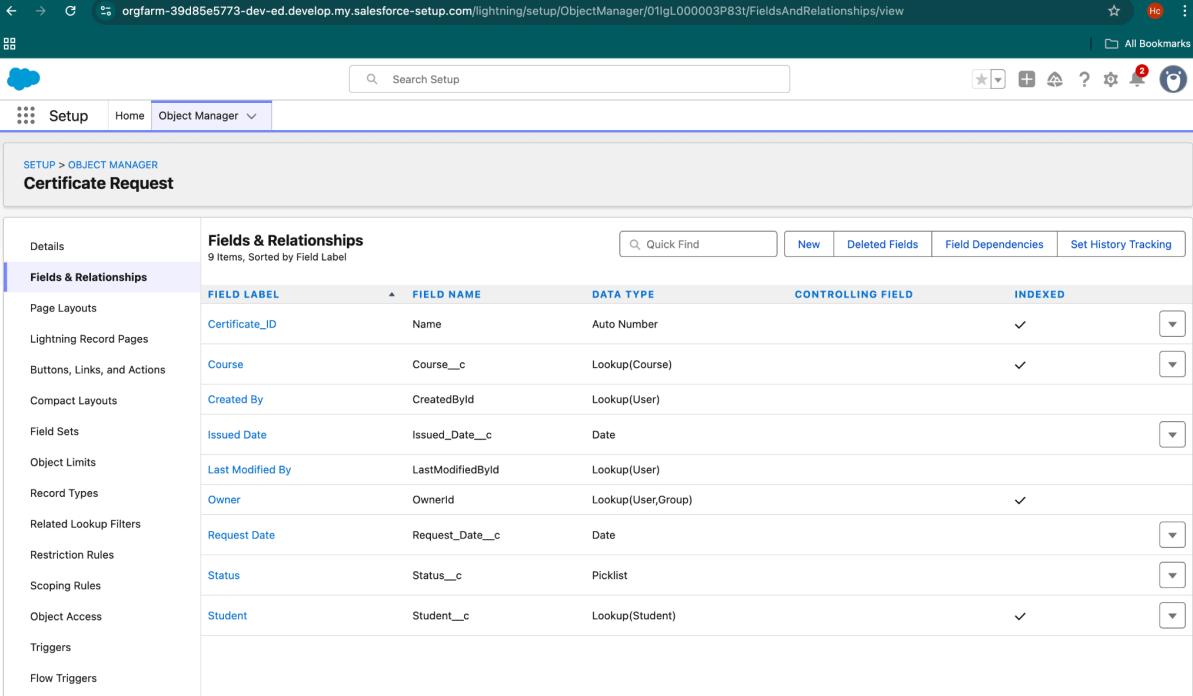
1. Standard & Custom Objects

Use Case

Standard Salesforce objects do not fully support academic enrollment and certification requirements. Therefore, custom objects are created to accurately represent the education domain.

Custom Objects Implemented

- Student_c – Stores student details
- Course_c – Stores course information
- Enrollment_c – Represents student enrollment in courses
- Certificate_Request_c – Manages certificate issuance



The screenshot shows the Salesforce Object Manager Fields & Relationships page for the 'Certificate Request' object. The page title is 'SETUP > OBJECT MANAGER Certificate Request'. On the left, there's a sidebar with various setup categories like Page Layouts, Lightning Record Pages, Buttons, etc. The main area is titled 'Fields & Relationships' and shows 9 items, sorted by Field Label. A table lists the fields with columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Certificate_ID	Name	Auto Number		✓
Course	Course_c	Lookup(Course)		✓
Created By	CreatedById	Lookup(User)		
Issued Date	Issued_Date_c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Request Date	Request_Date_c	Date		
Status	Status_c	Picklist		
Student	Student_c	Lookup(Student)		✓

2. Fields

Use Case

Fields store specific data required for academic tracking, automation, and reporting.

Fields Implemented

Student Object

- Student ID (Auto Number)
- Student Name (Text)
- Email (Email)
- Department (Picklist)

Course Object

- Course ID (Auto Number)
- Course Name (Text)
- Department (Picklist)

Enrollment Object

- Student (Lookup)
- Course (Lookup)
- Status (Picklist: Enrolled, Completed)

Certificate Request Object

- Student (Lookup)
- Course (Lookup)
- Status (Picklist: Requested, Approved, Issued)

3. Record Types

Use Case

Record Types are used when multiple business processes are required within the same object.

4. Page Layouts

Use Case

Page Layouts control how fields are displayed on record pages, improving usability and data entry.

Implementation in Project

Custom page layouts are configured for:

- Student
- Course
- Enrollment
- Certificate Request

5. Compact Layouts

Use Case

Compact layouts define the key fields shown in the highlights panel of a record.

Implementation in Project

Compact layouts are configured to display important information such as name and status.

6. Schema Builder

Use Case

Schema Builder provides a visual view of objects and relationships, helping to understand the data model clearly.

Implementation in Project

Schema Builder is used to visualize Student, Course, Enrollment, and Certificate Request objects and their relationships

7. Lookup vs Master-Detail vs Hierarchical Relationships

Use Case

Relationships define how objects are connected and how data behaves across records.

Implementation in Project

- Lookup Relationships are used for flexibility and independent record management.

Relationships Used

- Enrollment → Student (Lookup)
- Enrollment → Course (Lookup)
- Certificate Request → Student (Lookup)
- Certificate Request → Course (Lookup)

Master-Detail and Hierarchical relationships are not required for this business scenario.

The screenshot shows the Salesforce Setup interface under Object Manager for the 'Certificate Request' object. A custom field named 'Student' is being configured. The 'Fields & Relationships' tab is active. In the 'Field Information' section, the field label is 'Student', field name is 'Student', and API name is 'Student__c'. The object name is 'Certificate Request' and the data type is 'Lookup'. The 'Lookup Options' section indicates the field is related to the 'Student' object and has a child relationship name of 'Certificate_Requests'. The 'Custom Field Definition Detail' section includes buttons for Edit, Set Field-Level Security, View Field Accessibility, and Where is this used?

8. Junction Objects

Use Case

Junction objects are used to manage many-to-many relationships.

Implementation

The Enrollment object acts as a junction object between Student and Course.

9. External Objects

Use Case

External Objects are used to access data from external systems.

Phase 4: Process Automation (Admin)

Objective

The objective of Phase 4 is to automate key academic processes in the Campus Course Enrollment and Student Support CRM using Salesforce administrative tools. Process automation helps reduce manual effort, ensures data accuracy, and improves overall system efficiency.

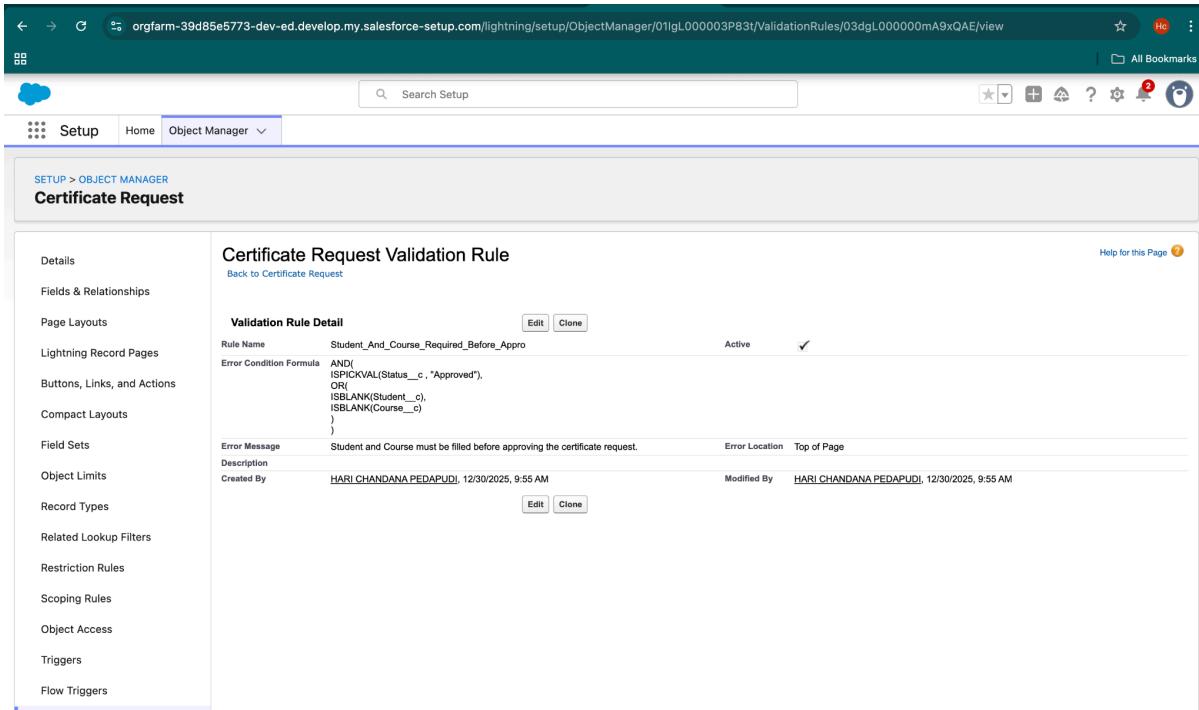
Validation Rules

Purpose:

Validation rules are used to prevent incorrect or incomplete data from being saved in Salesforce.

Implementation in Project:

A validation rule is implemented on the Certificate Request object to ensure that a certificate request cannot be approved unless both Student and Course fields are filled.



The screenshot shows the Salesforce Setup interface for the 'Object Manager' section, specifically for the 'Certificate Request' object. The page title is 'Certificate Request Validation Rule'. The validation rule is named 'Student_And_Course_Required_Before_Approval'. The error condition formula is: AND(ISPICKVAL(Status__c, "Approved"), OR(ISBLANK(Student__c), ISBLANK(Course__c))) . The error message is 'Student and Course must be filled before approving the certificate request.' The rule is active and was created by 'HARI CHANDANA PEDAPUDI' on 12/30/2025, 9:55 AM. The 'Edit' and 'Clone' buttons are visible at the top right of the rule details.

Workflow Rules

Purpose:

Workflow rules are used for simple automation such as field updates and email alerts.

Implementation in Project:

Workflow rules are not implemented because Salesforce Flow Builder is used for automation. Flow Builder is more flexible and is the recommended Salesforce tool.

Process Builder

Purpose:

Process Builder is used to automate business processes based on conditions.

Implementation in Project:

Process Builder is not implemented because all automation requirements are handled using Salesforce Flow Builder.

Approval Process

Purpose:

Approval processes are used when records require formal approvals.

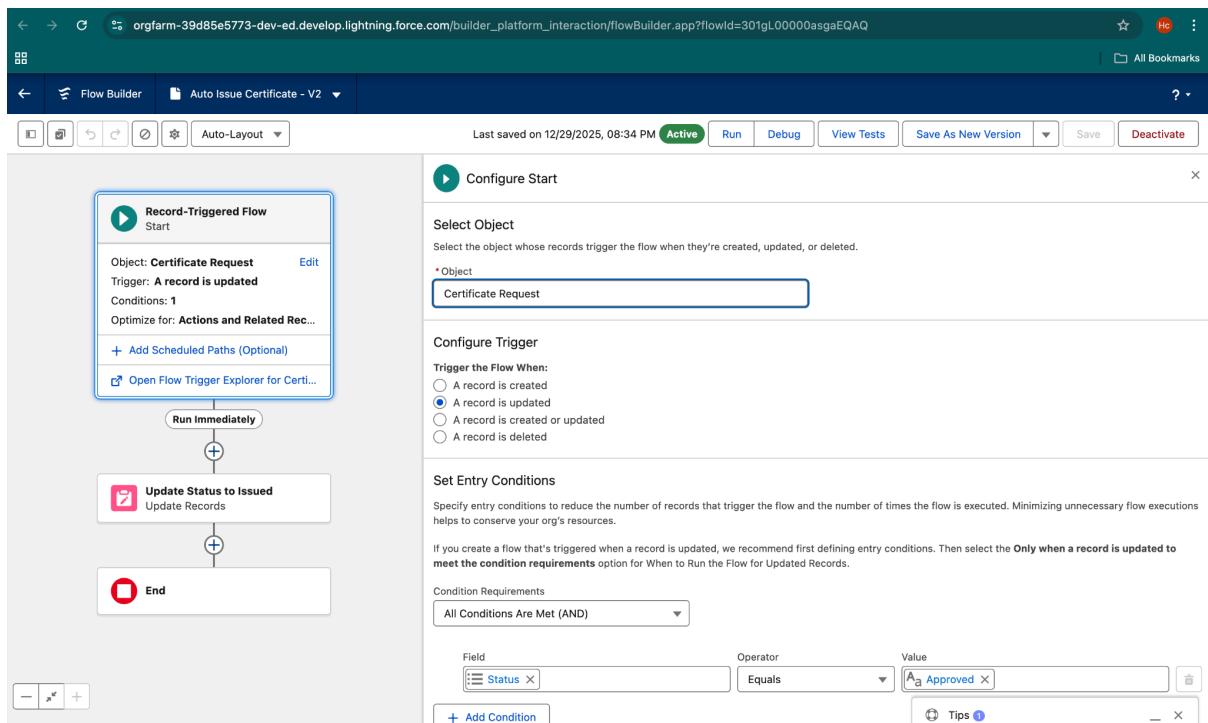
Implementation in Project:

A formal approval process is not configured. Certificate approval is managed using status changes and automation logic to keep the system simple.

Flow Builder (Record-Triggered Flows)

Purpose:

Flow Builder is used to automate business processes based on record changes.



Flow 1: Certificate Request Creation

Use Case:

When a student completes a course, a certificate request should be created automatically.

Automation Logic:

- Object: Enrollment
- Trigger Condition: Status = Completed
- Action: Create Certificate Request record with status = Requested

Flow 2: Certificate Issuance Automation

Use Case:

When a certificate request is approved, the certificate should be issued automatically.

Automation Logic:

- Object: Certificate Request
- Trigger Condition: Status = Approved
- Action: Update status to Issued

Screen Flows

Purpose:

Screen flows guide users through step-by-step interactions.

Implementation in Project:

Screen flows are not implemented because the automation is system-driven and does not require user input.

Scheduled Flows

Purpose:

Scheduled flows run automation at a specific time.

Implementation in Project:

Scheduled flows are not implemented because automation is triggered by record updates.

Auto-Launched Flows

Purpose:

Auto-launched flows run automatically in the background.

Implementation in Project:

The record-triggered flows implemented in this project function as auto-launched flows.

Email Alerts

Purpose:

Email alerts notify users when specific events occur.

Field Updates

Purpose:

Field updates automatically update record values during automation.

Implementation in Project:

Field updates are used within flows to update the Certificate Request status during the issuance process.

Tasks

Purpose:

Tasks are used to assign follow-up actions.

Implementation in Project:

Tasks are not implemented because automation handles all required actions.

Custom Notifications

Purpose:

Custom notifications inform users inside Salesforce.

Phase 5: Apex Programming (Developer)

Objective

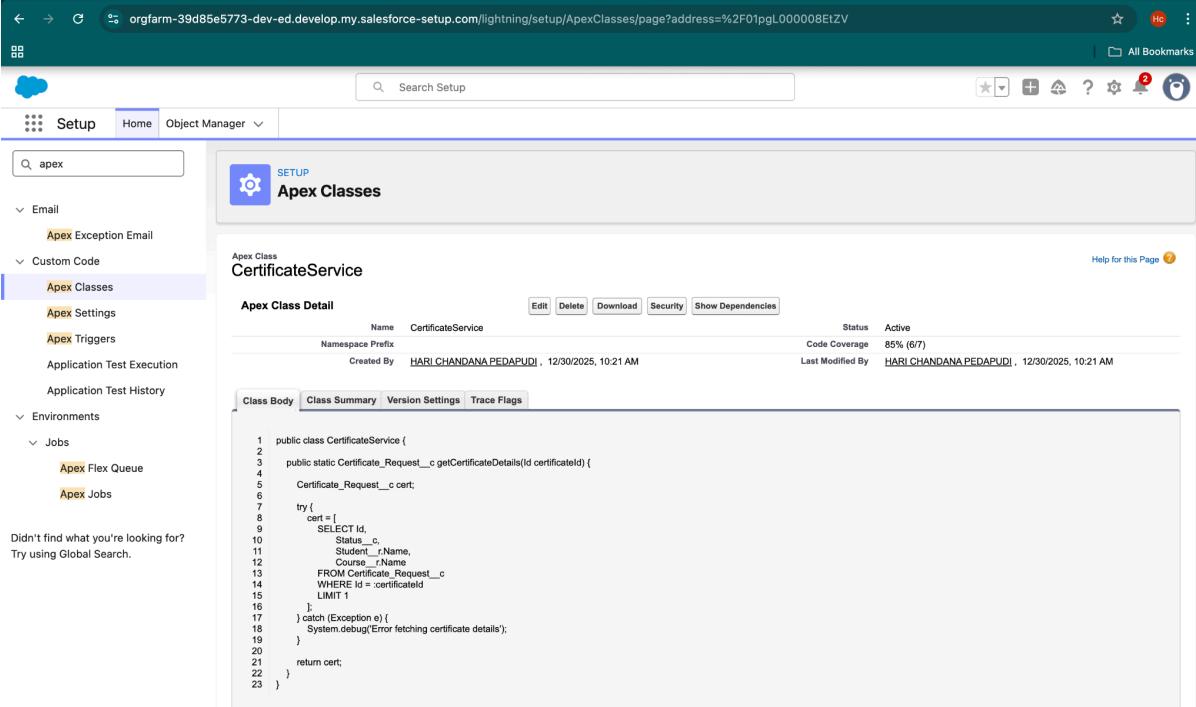
The objective of Phase 5 is to implement server-side business logic using Apex programming where declarative tools are not sufficient. This phase focuses on automation, data processing, and validation through Apex Classes, Apex Triggers, and Test Classes.

Apex Classes

An Apex class named CertificateService is created. The class uses SOQL to fetch certificate request details based on record ID and includes exception handling to manage runtime errors.

Key Concepts Covered

- Apex Class creation
- SOQL queries
- Exception handling



The screenshot shows the Salesforce Setup Apex Classes page. The URL is orgfarm-39d85e5773-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ApexClasses/page?address=%2F01pgL000008EtZV. The page title is "Apex Classes". On the left, there's a sidebar with a search bar and sections for Email, Custom Code (with "Apex Classes" selected), Apex Settings, Apex Triggers, Application Test Execution, Application Test History, Environments, and Jobs. A note says "Didn't find what you're looking for? Try using Global Search." The main content area shows the "Apex Class Detail" for "CertificateService". The class body contains the following Apex code:

```
1 public class CertificateService {
2     public static Certificate_Request__c getCertificateDetails(Id certificateId) {
3         Certificate_Request__c cert;
4         try {
5             cert = [
6                 SELECT Id,
7                     Status__c,
8                     Student__Name,
9                     Course__Name
10                FROM Certificate_Request__c
11                WHERE Id = :certificateId
12                LIMIT 1
13            ];
14        } catch (Exception e) {
15            System.debug('Error fetching certificate details');
16        }
17        return cert;
18    }
19}
```

Apex Triggers

Use Case

To automate certificate issuance when a certificate request is approved, an Apex trigger is implemented. This ensures consistency and removes manual intervention.

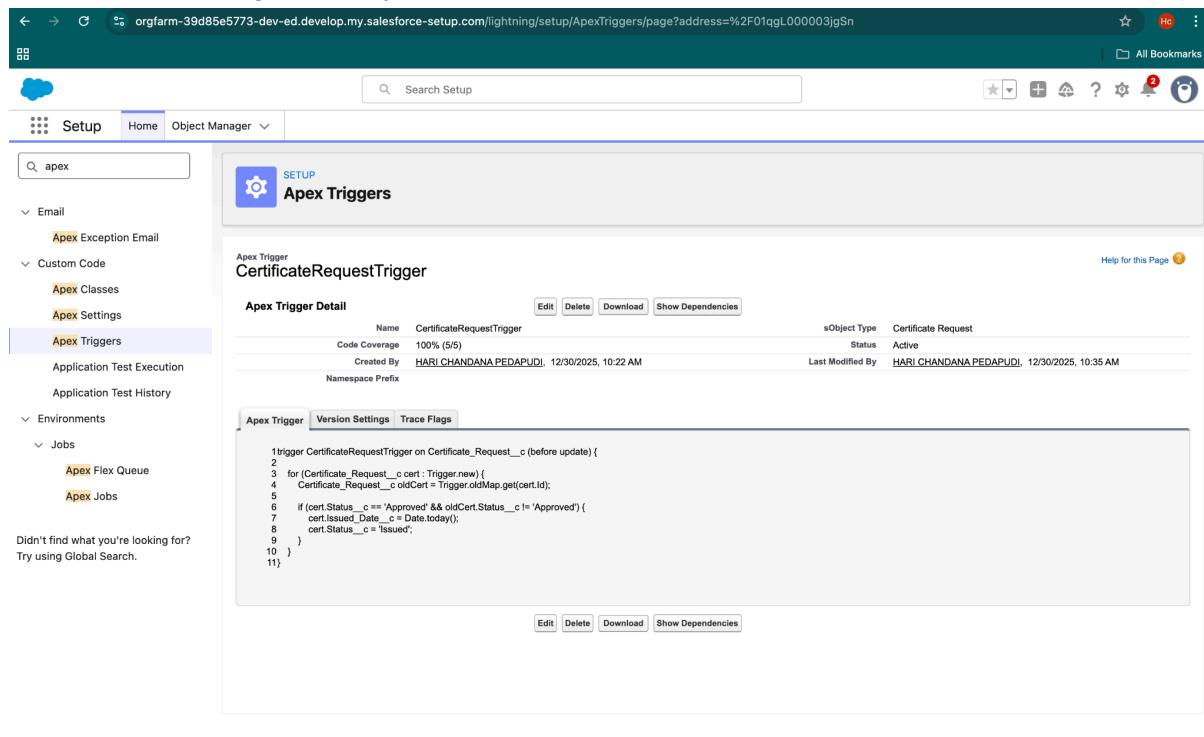
Implementation

A before update Apex trigger is created on the Certificate Request object. When the Status changes from Requested to Approved, the trigger automatically:

- Updates the Issued Date
- Changes the Status to Issued

Why Before Update

Using a before update trigger allows modification of records without additional DML operations, avoiding read-only and recursion issues.



Test Classes

Use Case

Test classes are mandatory in Salesforce to validate Apex code and ensure correct functionality. They also help achieve required code coverage.

Implementation

A test class named CertificateServiceTest is implemented. The test class:

- Creates a Certificate Request record
- Updates the Status to Approved
- Verifies that the trigger changes the Status to Issued
- Confirms the Apex class returns valid data

Key Concepts Covered

- @isTest annotation
- Test data creation
- Assertions
- Trigger and class validation

The screenshot shows the Salesforce Setup interface with the 'Apex Classes' section selected. The search bar at the top left contains 'apex'. The main content area displays the 'CertificateService' class details. The 'Apex Class Detail' section shows the class name 'CertificateService', namespace prefix 'CertificateService', and status 'Active'. Below this, the 'Class Body' tab is selected, showing the Apex code for the class:

```
1 public class CertificateService {  
2     public static Certificate_Request__c getCertificateDetails(Id certificateId) {  
3         Certificate_Request__c cert;  
4         try {  
5             cert = [SELECT Id,  
6                     Student__r.Name,  
7                     Course__r.Name  
8                 FROM Certificate_Request__c  
9                 WHERE Id = :certificateId  
10                LIMIT 1];  
11         } catch (Exception e) {  
12             System.debug('Error fetching certificate details');  
13         }  
14         return cert;  
15     }  
16 }
```

Result & Automation Outcome

- Certificate issuance process is fully automated
- Issued Date and Status are updated automatically
- Business logic is centralized and reusable
- Code coverage requirements are met

The screenshot shows the Salesforce Setup Apex Classes page. The URL is <https://orgfarm-39d85e5773-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ApexClasses/page?address=%2F01pgL0000008E1ZV>. The page title is "Apex Classes". The left sidebar has a search bar for "apex" and categories like Email, Custom Code (with Apex Classes selected), Application Test, Environments, and Jobs. The main content shows the "CertificateService" class detail. The class body contains the following Apex code:

```

1 public class CertificateService {
2     public static Certificate_Request__c getCertificateDetails(Id certificateId) {
3         Certificate_Request__c cert;
4         try {
5             cert = [
6                 SELECT Id,
7                     Status__c,
8                     Student__r.Name,
9                     Course__r.Name
10                FROM Certificate_Request__c
11                WHERE Id = :certificateId
12                LIMIT 1
13            ];
14        } catch (Exception e) {
15            System.debug('Error fetching certificate details');
16        }
17        return cert;
18    }
19 }
20
21
22
23

```

Phase 6: UI Customization (User Experience)

Objective

The objective of Phase 6 is to enhance the user interface and user experience of the Art Gallery / Certificate Management CRM by customizing page layouts, Lightning Record Pages, and compact layouts. This phase ensures that users can easily view, enter, and manage data efficiently.

Page Layout Customization

Use Case

Page Layouts are customized to control the fields, sections, and related lists displayed to users. This helps ensure that users see only relevant information based on business requirements.

Implementation

The Certificate Request Page Layout is customized to:

- Display important fields such as Status, Student, Course, Request Date, and Issued Date
- Organize fields into logical sections

- Remove unnecessary standard fields

Lightning Record Page Customization

Use Case

Lightning Record Pages provide a dynamic and modern UI by using components such as highlights panel, tabs, and related lists. This improves usability and navigation.

Implementation

A custom Lightning Record Page is configured for the Certificate Request object with:

- Highlights Panel showing key fields (Status, Issued Date)
- Tabs for Details and Related Lists
- Activity Timeline for tracking user actions

The screenshot displays a custom Lightning Record Page for a 'Certificate Request' record. At the top, there's a navigation bar with links for 'Campus Enrollment CRM', 'Reports', 'Dashboards', 'Students', 'Courses', 'Enrollments', and 'Certificate Requests'. Below the navigation, the record ID 'CERT-0001' is shown. The main content area has two tabs: 'Related' and 'Details'. Under 'Details', fields are listed: Status (Issued), Owner (HARI CHANDANA PEDAPUDI), Student (STU-0001), Course (CRS-0001), Issued Date (12/30/2025), Created By (HARI CHANDANA PEDAPUDI, 12/9/2025, 8:51 AM), and Last Modified By (HARI CHANDANA PEDAPUDI, 12/30/2025, 11:24 PM). To the right, there's an 'Activity' timeline component with a header 'Activity' and a message 'No activities to show.' It includes filter options and buttons for 'Refresh', 'Expand All', and 'View All'.

Compact Layouts

Use Case

Compact Layouts define which fields appear in the Highlights Panel at the top of a record. This allows users to quickly view critical information without scrolling.

Implementation

A Compact Layout is created for Certificate Request to display:

- Certificate ID
- Status

- Issued Date

Related Lists Customization

Use Case

Related Lists help users view associated records directly from the parent record page.

Implementation

Relevant related lists such as:

- Activities
- History Tracking

are enabled and arranged on the page layout for better visibility.

Phase 7: Integration & External Access

Objective

The objective of Phase 7 is to demonstrate Salesforce's capability to integrate with external systems and services. This phase focuses on secure communication, API access, authentication, and external data connectivity using Salesforce integration features.

Named Credentials

Use Case

Named Credentials are used to securely store endpoint URLs and authentication details for external systems. They simplify callouts by removing hard-coded credentials from Apex code.

Implementation

In this project, a Named Credential is created to represent an external REST service endpoint. Authentication details are managed centrally to ensure security and easy maintenance.

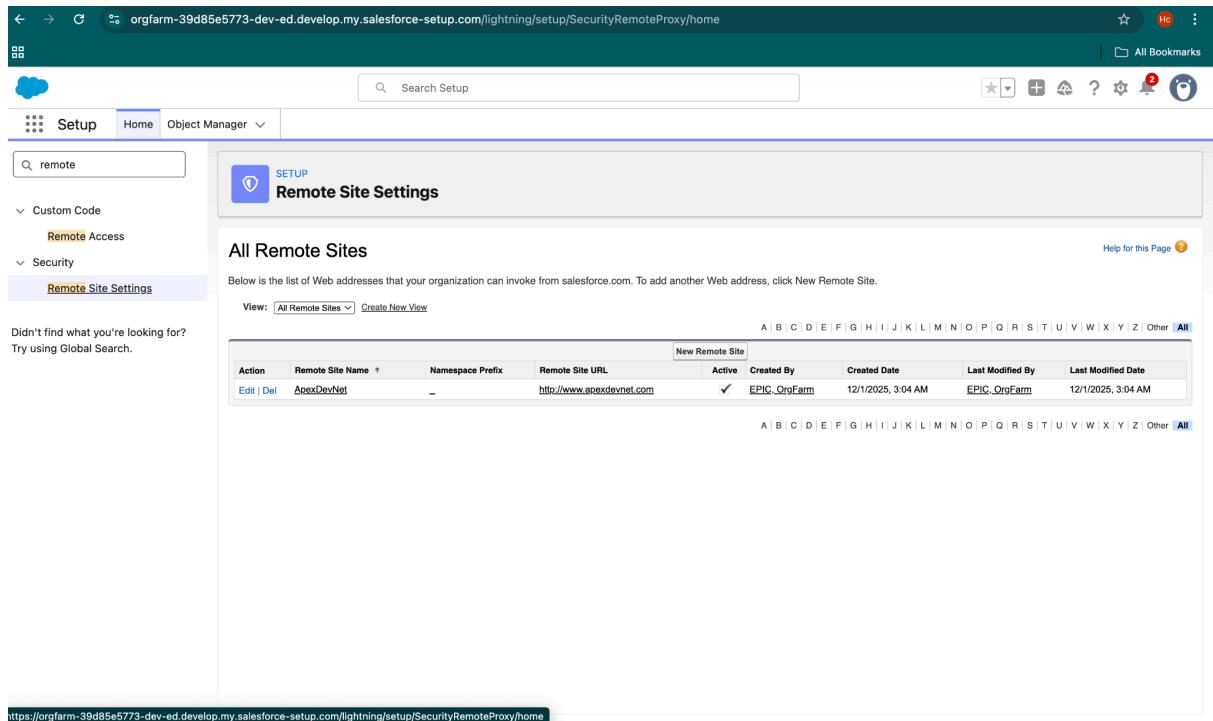
Remote Site Settings

Use Case

Remote Site Settings allow Salesforce to make outbound callouts to external services securely.

Implementation

A Remote Site Setting is configured with an external service URL to enable Apex callouts for integration testing.



The screenshot shows the Salesforce Setup interface. In the left sidebar, under 'Custom Code' and 'Remote Access', the 'Remote Site Settings' option is selected. The main content area is titled 'Remote Site Settings' and displays a table titled 'All Remote Sites'. The table has columns for Action, Remote Site Name, Namespace Prefix, Remote Site URL, Active, Created By, Created Date, Last Modified By, and Last Modified Date. One row is listed: 'Edit | Del' for 'ApexDevNet', Namespace Prefix '-' (empty), Remote Site URL 'http://www.apexdevnet.com', Active checked, Created By 'EPIC_OrgFarm', Created Date '12/1/2025, 3:04 AM', Last Modified By 'EPIC_OrgFarm', and Last Modified Date '12/1/2025, 3:04 AM'. Below the table, there are navigation links for letters A-Z and a link to 'New Remote Site'. The browser address bar at the top shows the URL: https://orgfarm-39d85e5773-dev-ed.develop.my.salesforce-setup.com/lightning/setup/SecurityRemoteProxy/home.

Web Services (REST)

Implementation

A REST-based integration approach is used where Salesforce can consume external REST APIs. This enables data exchange between Salesforce and third-party systems.

Callouts

Implementation

An Apex HTTP callout is implemented to fetch sample data from an external REST service. This demonstrates Salesforce's outbound integration capability.

OAuth & Authentication

Implementation

OAuth authentication is configured through Named Credentials to securely authenticate external API access.

API Limits

Implementation

API usage is monitored using Salesforce System Overview to ensure integrations remain within permitted limits.

External Services

Implementation

External Services are explored to demonstrate low-code integration capabilities. This enables Flow-based integrations for future enhancements.

Salesforce Connect

Implementation

Salesforce Connect is studied and documented as a potential approach for real-time external data access. It is not fully implemented due to project scope.

Phase 8: Data Management & Deployment

Objective

The objective of Phase 8 is to manage data efficiently within Salesforce and demonstrate deployment techniques used to move configurations between environments. This phase focuses on data import, data quality, backup, and deployment best practices.

Data Import Wizard

Use Case

Data Import Wizard is used to import small to medium volumes of data into Salesforce without writing code. It supports standard and custom objects.

Implementation

In this project, Data Import Wizard is used to import sample data for objects such as:

- Student
- Course
- Certificate Request

Data Loader

Implementation

Data Loader is explored to understand bulk data operations. It is used to export Certificate Request records for backup and analysis purposes.

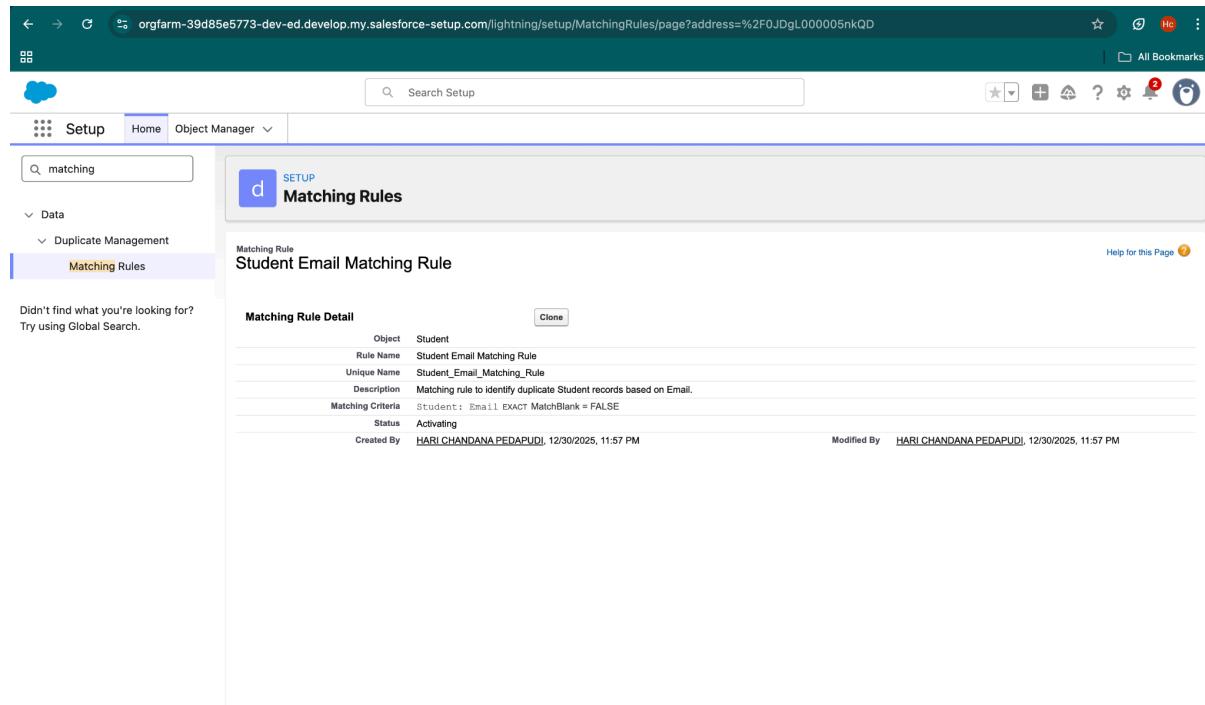
Duplicate Rules

Use Case

Duplicate Rules prevent the creation of duplicate records and maintain data quality.

Implementation

A Duplicate Rule is configured for the Student object to prevent duplicate student records based on key fields such as Email or Student ID.



The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-39d85e5773-dev-ed.develop.my.salesforce-setup.com/lightning/setup/MatchingRules/page?address=%2F0JDgL000005nkQD>. The page title is "Matching Rules". The left sidebar shows "Data" and "Duplicate Management" sections, with "Matching Rules" selected. The main content area displays a "Matching Rule Detail" for "Student Email Matching Rule". The details are as follows:

Object	Student
Rule Name	Student Email Matching Rule
Unique Name	Student_Email_Matching_Rule
Description	Matching rule to identify duplicate Student records based on Email.
Matching Criteria	Student: Email EXACT MatchBlank = FALSE
Status	Activating
Created By	HARI CHANDANA PEDAPUDI, 12/30/2025, 11:57 PM
Modified By	HARI CHANDANA PEDAPUDI, 12/30/2025, 11:57 PM

The screenshot shows the Salesforce Setup interface with the 'Duplicate Rules' page selected. The page title is 'Student Duplicate Rule'. The 'Duplicate Rule Detail' section contains the following information:

Rule Name	Student Duplicate Rule	Order	1 of 1 [Reorder]
Description			
Object	Student		
Record-Level Security	Enforce sharing rules	Operations On Create	<input type="checkbox"/> Alert <input type="checkbox"/> Report
Action On Create	Block	Operations On Edit	<input type="checkbox"/> Alert <input type="checkbox"/> Report
Action On Edit	Block		
Alert Text	Duplicate Student record detected based on Email. Please check existing records before saving.		
Active	<input checked="" type="checkbox"/>		
Matching Rule	<input checked="" type="checkbox"/> Student Email Matching Rule <input checked="" type="checkbox"/> Mapped	Matching Criteria	Student: Email EXACT MatchBlank = FALSE
Conditions			
Created By	HARI CHANDANA PEDAPUDI, 12/31/2025, 12:09 AM	Modified By	HARI CHANDANA PEDAPUDI, 12/31/2025, 12:09 AM

Buttons at the bottom include 'Edit', 'Delete', 'Clone', and 'Deactivate'.

Data Export & Backup

Use Case

Data Export is used to create regular backups of Salesforce data to prevent data loss.

Implementation

Salesforce Data Export is configured to export data periodically. Exported data includes Certificate Requests, Students, and Courses.

Change Sets

Use Case

Outbound and Inbound Change Sets are created to deploy:

- Custom objects
- Fields
- Apex classes
- Apex triggers

Unmanaged vs Managed Packages

Unmanaged packages are studied and documented as they are suitable for internal project deployment and academic use. Managed packages are reviewed conceptually.

ANT Migration Tool

ANT Migration Tool is explored as an alternative deployment option for large-scale projects. It is documented for conceptual understanding.

VS Code & Salesforce DX (SFDX)

VS Code and Salesforce DX are explored to understand project structure, metadata retrieval, and deployment workflows.

Phase 9: Reporting, Dashboards & Security Review

Objective

The objective of Phase 9 is to analyze project data using reports and dashboards and to review the security configuration implemented in the Salesforce CRM.

Reports

Tabular Report – Certificate Requests List

Use Case:

To view all certificate request records in a simple list format.

What We Did:

A tabular report is created on the Certificate Request object displaying fields such as Certificate ID, Student, Course, Status, and Issued Date.

The screenshot shows a Salesforce report interface. At the top, the URL is orgfarm-39d85e5773-dev-ed.lightning.force.com/lightning/r/Report/00OgL0000070YHRUA4/view?queryScope=userFolders. The page title is 'Campus Enrollment CRM'. The navigation bar includes 'Reports', 'Dashboards', 'Students', 'Courses', 'Enrollments', and 'Certificate Requests'. A search bar at the top right contains 'Search...'. Below the navigation, the report title is 'Report: Certificate Requests' and the specific report is 'Certificate Requests by Status'. The report displays a single record with the following details:

Status	Certificate Request: Certificate_ID
Issued (1)	CERT-0001

Below the table, there are summary rows: 'Subtotal' and 'Total (1)'. At the bottom of the report area, there are buttons for 'Row Counts', 'Detail Rows', 'Subtotals', and 'Grand Total'.

Dashboards

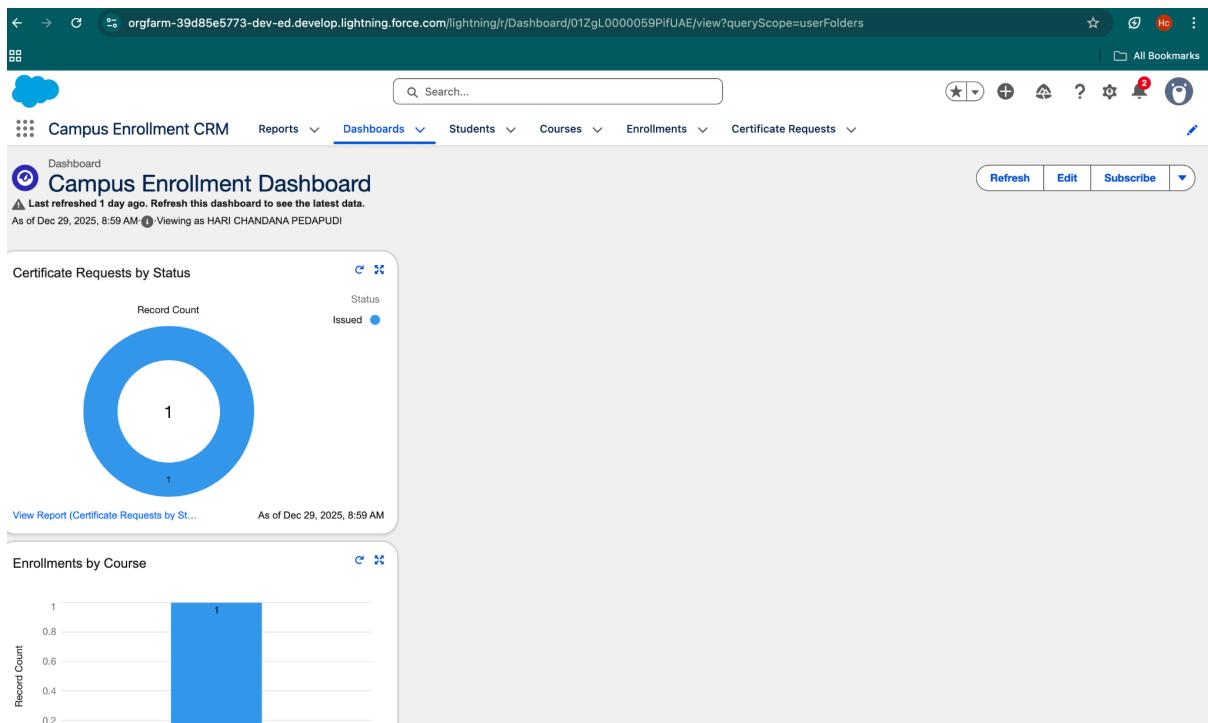
Dashboard – Certificate Management Overview

Use Case:

To get a quick visual overview of certificate processing and request status.

What We Did:

A dashboard is created using the above reports with chart components to visually represent certificate request data.



Profiles

Use Case:

To control what users can access and modify in the system.

What We Did:

Profiles are used to manage permissions for Admin and Standard User roles.

Roles

Use Case:

To manage record visibility using role hierarchy.

What We Did:

Roles are configured to control data visibility across users.

Users

Use Case:

To allow users to access the Salesforce application.

What We Did:

Users are created and assigned appropriate profiles and roles.

Permission Sets

Use Case:

To grant additional permissions without changing user profiles.

What We Did:

Permission sets are created and assigned to users based on requirements.

Organization-Wide Defaults (OWD)

Use Case:

To define default access to records.

What We Did:

OWD settings are configured to control default visibility for Certificate Request records.

The screenshot shows the 'Sharing Settings' page in the Salesforce Setup. The left sidebar has 'Sharing' selected under 'Security'. The main area shows the 'Sharing Settings' section with 'Organization-Wide Defaults' and 'Other Settings' tabs. Under 'Organization-Wide Defaults', the 'Student' object has 'Public Read/Write' as Default Internal Access and 'Private' as Default External Access. The 'Grant Access Using Hierarchies' checkbox is checked. Under 'Other Settings', 'Secure guest user record access' is checked. The 'Sharing Rules' section shows no sharing rules specified. The 'Sharing Overrides' section shows profiles like 'Analytics Cloud Integration User' with specific permissions like 'View All Data' and 'Modify All Records'.

The screenshot shows the 'Sharing Settings' page in the Salesforce Setup, specifically for the 'Certificate Request' object. The left sidebar has 'Sharing' selected under 'Security'. The main area shows the 'Sharing Settings' section with 'Manage sharing settings for: Certificate Request'. The 'Default Sharing Settings' table shows 'Certificate Request' with 'Public Read/Write' as Default Internal Access and 'Private' as Default External Access. The 'Grant Access Using Hierarchies' checkbox is checked. Under 'Other Settings', 'Secure guest user record access' is checked. The 'Sharing Rules' section shows no sharing rules specified. The 'Sharing Overrides' section is not visible in this screenshot.

Sharing Rules

Use Case:

To share records with specific users or roles.

What We Did:

Sharing rules are created for the Certificate Request object to extend record access.

Field-Level Security

Use Case:

To restrict access to sensitive fields.

What We Did:

Field-level security is applied to hide selected fields from standard users.

Session Settings

Use Case:

To improve login and session security.

What We Did:

Session timeout and related security settings are configured.

Audit Trail

Use Case:

To track administrative changes made in the system.

What We Did:

Audit Trail is reviewed to monitor setup and configuration changes.

Phase 10: Quality Assurance Testing

Objective

The objective of Phase 10 is to validate that all implemented Salesforce features work as expected. Testing ensures data accuracy, automation correctness, and system reliability.

Test Case 1: Certificate Request Record Creation

Use Case / Scenario

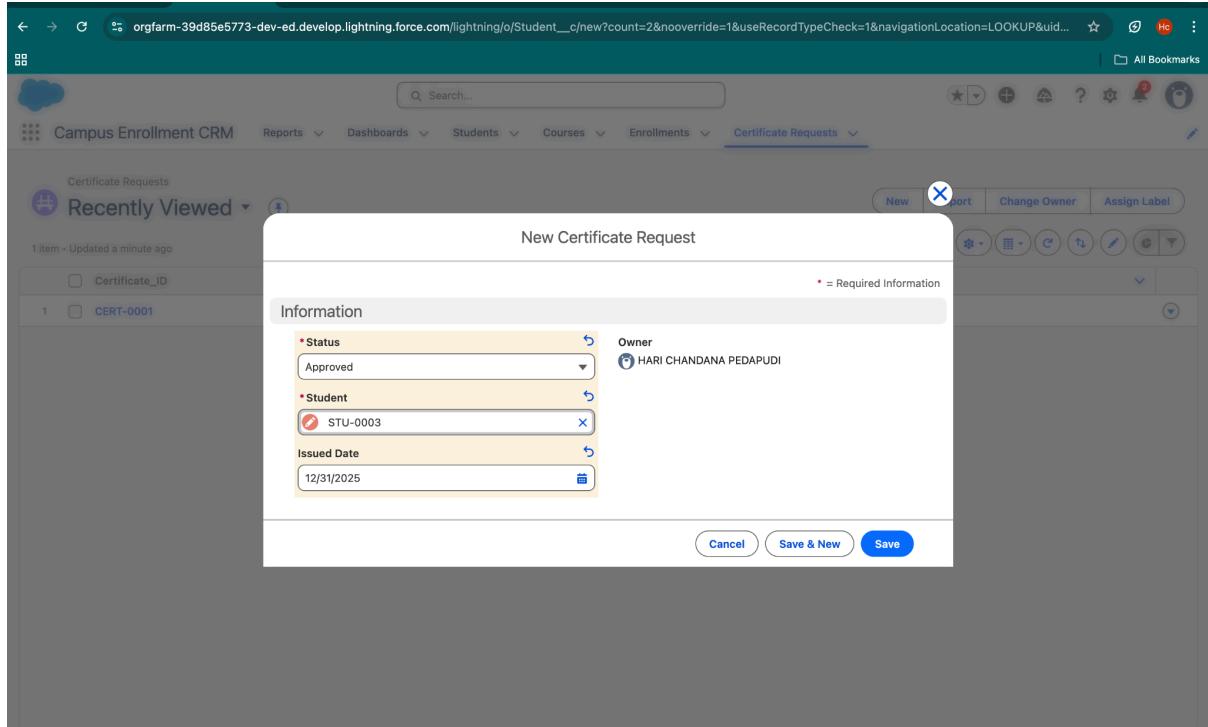
Verify that a Certificate Request record can be created successfully.

Test Steps (Input)

1. Navigate to Certificate Request tab
2. Click New
3. Enter Student, Course, and Status = Requested
4. Click Save

Result

Certificate Request record is created successfully.



Test Case 2: Validation Rule Testing

Use Case / Scenario

Ensure validation rule prevents saving incomplete Certificate Request.

Test Steps (Input)

1. Click New Certificate Request
2. Leave required fields empty
3. Click Save

Result

Validation error message is displayed and record is not saved.

Test Case 3: Apex Trigger – Certificate Issuance Automation

Use Case / Scenario

Verify that certificate is automatically issued when status is Approved.

Test Steps (Input)

1. Open an existing Certificate Request record
2. Change Status from Requested to Approved
3. Click Save

Result

Status changed to Issued and Issued Date is auto-filled.

The screenshot shows a Salesforce Lightning interface for a 'Certificate Request' record. The record ID is 'CERT-0001'. The 'Status' field is set to 'Issued'. The 'Issued Date' field shows '12/30/2025'. The 'Owner' is listed as 'HARI CHANDANA PEDAPUDI'. On the right, the 'Activity' sidebar shows no activities to show.

Test Case 4: Flow Automation Testing

Use Case / Scenario

Verify that record-triggered flow executes correctly.

Test Steps (Input)

1. Update Certificate Request status

2. Save the record

Result

Flow executed successfully and record updated.

Test Case 5: Report Generation Testing

Use Case / Scenario

Verify that reports display correct data.

Test Steps (Input)

1. Open Reports tab
2. Run Certificate Requests by Status report

Result

Report displays correct grouped data.

The screenshot shows a Salesforce Lightning report titled "Report: Certificate Requests" with the specific view "Certificate Requests by Status". The report displays a single record with the following details:

Status ↑	Certificate Request: Certificate_ID
Issued (1)	CERT-0001

Below the table, there are summary rows for "Subtotal" and "Total (1)". At the bottom of the report, there are several checkboxes for "Row Counts", "Detail Rows", "Subtotals", and "Grand Total", with "Subtotals" and "Grand Total" being checked.

Test Case 6: Dashboard Testing

Use Case / Scenario

Verify that dashboard displays report data correctly.

Test Steps (Input)

1. Open Dashboard
2. Refresh dashboard

Result

Dashboard displays updated charts correctly.

