

Requirement Gathering

Educational institutions, especially colleges and universities, face challenges in handling thousands of student inquiries during admission seasons. Currently, most follow a **manual or semi-digital approach** (Excel sheets, emails, phone calls). This causes:

- Loss of potential student leads due to poor tracking.
- Delays in responding to students and parents.
- Difficulty in maintaining a **single source of truth** for student data.
- No real-time visibility of admission progress for management.

Thus, the requirement is to build a **centralized CRM system** that automates lead tracking, improves student communication, and provides real-time dashboards.

Stakeholder Analysis

- **Admission Officers** → Manage student inquiries, follow-ups, and applications.
- **Students & Parents** → Expect timely updates, clear communication, and transparency in admission progress.
- **Administrators/Management** → Need dashboards and reports to track admission performance and forecast future trends.
- **IT/Support Staff** → Ensure system availability, user training, and data management.

Business Process Mapping

Current Admission Process (Manual):

1. Student inquiry received via phone/email/form.
2. Admission officer manually records data in Excel or registers.
3. Follow-up reminders are handled manually.
4. No clear tracking of how many inquiries convert to admissions.

Proposed Salesforce-based Admission Process:

1. Inquiries automatically captured as **Leads** in Salesforce.
2. Leads qualified and converted into **Opportunities (Admission Applications)**.
3. Once confirmed, students stored in a **custom object (Student Record)**.
4. Automated email/SMS reminders keep students/parents informed.
5. Dashboards & reports give real-time insights into admissions.

Industry-Specific Use Case Analysis (Education Sector)

- **Problem:** High volume of inquiries, lack of follow-ups, poor data visibility.
- **Use Case:** CRM can help institutions manage the full **student lifecycle** from inquiry to admission.
- **Impact:**
 - Increased student conversion rates.
 - Improved communication with students/parents.
 - Enhanced reputation of the institution with a transparent admission process.

AppExchange Exploration

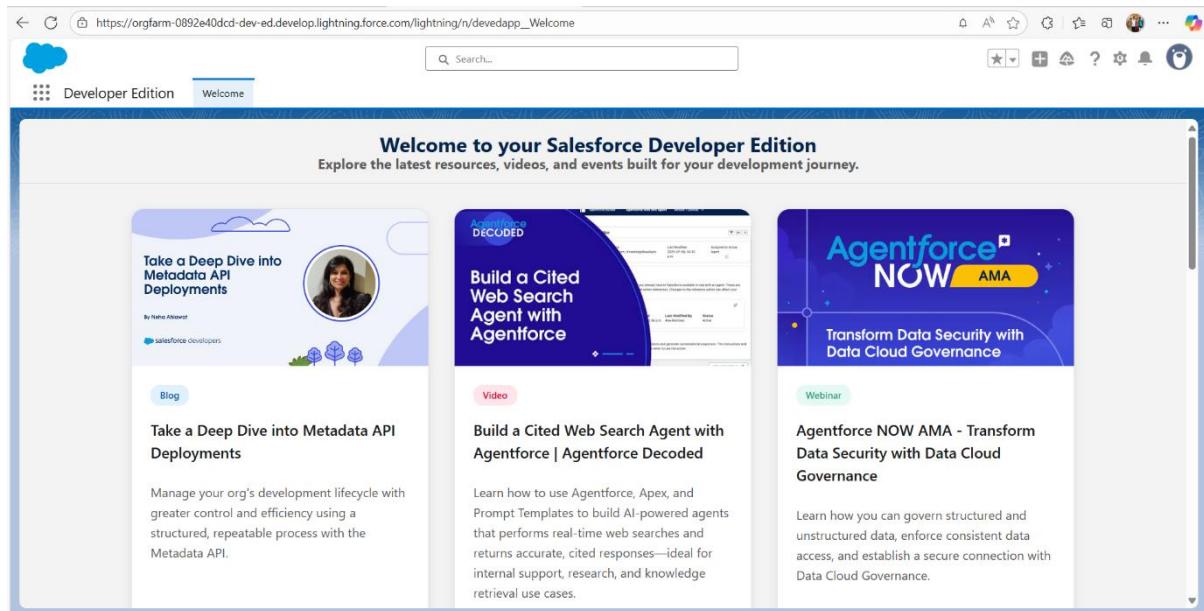
Salesforce **AppExchange** already offers education-specific solutions like:

- **Salesforce.org Education Cloud** (for higher education institutions).
- **TargetX CRM** (student recruitment and engagement).
- **Enrollment Rx** (end-to-end admission management).

Our EduCRM project will take inspiration from these real-world solutions but will be a **simplified custom CRM**, tailored for admission tracking and student communication.

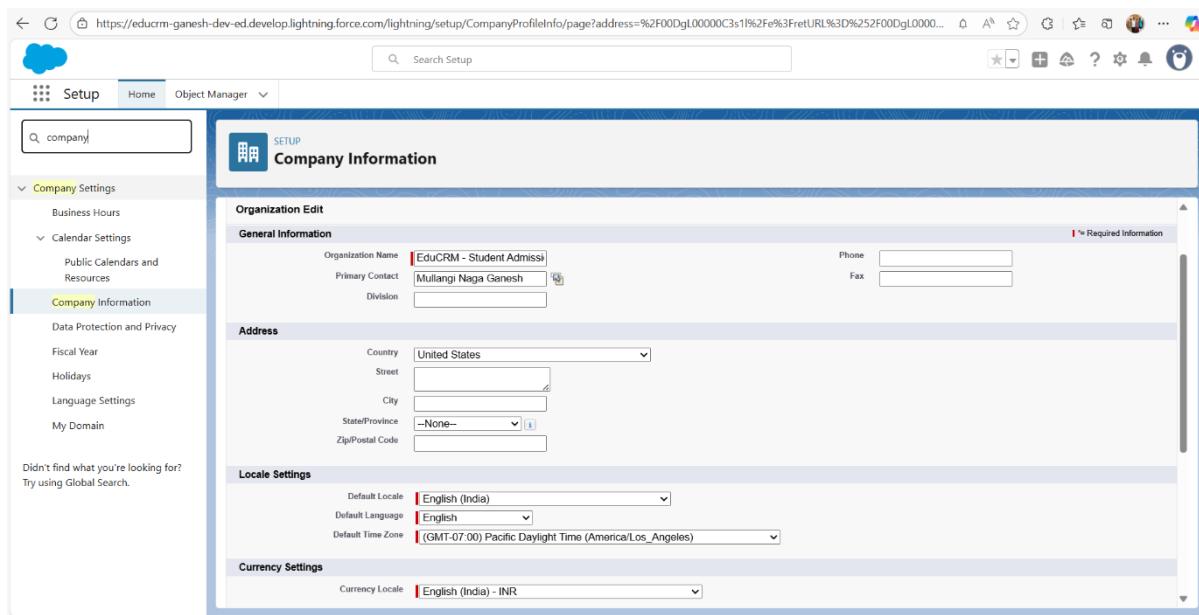
1. Salesforce Editions

- **Developer Edition** was used for this project.
- Provides access to core Salesforce CRM functionalities, suitable for testing and development purposes.



2. Company Profile Setup

- Default company settings were used in the Developer Org.
- Company information such as address, default currency, and locale settings were configured according to standard defaults.



3. Business Hours & Holidays

- Default business hours and holidays were used.
- Not critical for the project since automated processes do not depend on business hours at this stage.

4. Fiscal Year Settings

- Standard fiscal year settings applied.
- Optional for small-scale projects; no financial reporting automation required at this stage.

5. User Setup & Licenses

- **Users Created for EduCRM:**
- Users represent key roles in the EduCRM system for managing admissions and student support.

Name	Username	Profile	License
Admin	System_admin_sadmi@gmail.com	Admin Profile	Salesforce
Admission Officer	Admission_officer_admis@gmail.com	Admission Officer Profile	Salesforce Platform
Student Support	Student_support_studs@gmail.com	Student Support Profile	Salesforce Platform

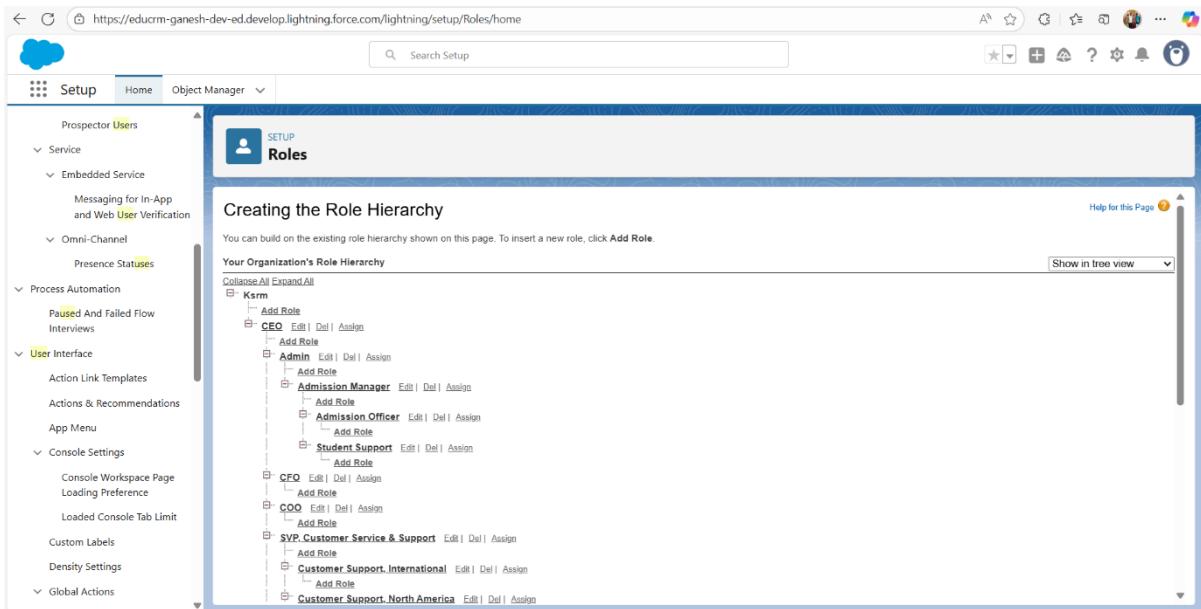
Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Admin_System	sadmi	system_admin_sadmi@gmail.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Chatter Expert	Chatter	chatty.0009000000c3s1uab.blywygdxnb@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	EPIC_OrgFarm	OEPI	epic.0008551d2765@orgfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Naga Ganesh_Mullangi	nag	nagaganesh.mullangi110@gagenfotech.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Officer_Admission	admis	admission_officer_admis@gmail.com		<input checked="" type="checkbox"/>	Standard User
<input type="checkbox"/>	Support_Student	studs	student_support_studs@gmail.com		<input checked="" type="checkbox"/>	Standard Platform User
<input type="checkbox"/>	User_Integration	integ	integration@0009000000c3s1uab.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>	User_Security	sec	inightssecurity@00dg00000c3s1uab.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

6. Profiles

- **Profiles Created / Cloned:**
 1. **Admin Profile** → Full access to all objects and fields.
 2. **Admission Officer Profile** → Access to Student and Admission Application objects.
 3. **Student Support Profile** → Access to Student and Support Ticket objects.
- **Field-Level Security (FLS):** Profiles configured to ensure appropriate visibility and edit permissions.

7. Roles

- Role hierarchy ensures proper record access: Admin sees all records, Admission Officers and Student Support see only relevant data.



8. Permission Sets

- Not created for this project as all access is managed via Profiles.
 - Optional for more advanced access control in larger projects.

9. OWD (Organization-Wide Defaults) & Sharing Rules

- Default OWD settings used.
 - No sharing rules configured since access control is handled via Roles and Profiles.

10. Login Access Policies

- Default login access policies applied.
 - Ensures secure login for all users.

11. Dev Org Setup

- Developer Org is configured and ready for customizations.
- Sandbox not required for this project.

12. Sandbox Usage

- Not applicable in Developer Org for this project.

13. Deployment Basics

- Deployment not required at this stage; changes are directly made in Developer Org.

3.1 Standard & Custom Objects

Use Case

The admission process involves handling student data, course applications, and approval tracking. Salesforce provides standard objects like *User* and *Account*, but to meet educational needs, we created a custom object:

Admission Application — to manage each student's admission form and approval status.

Objects Used

Type	Object Name	Purpose
Standard	User	Represents system users (Admin, Admission Officer, Student Support)
Custom	Admission	Captures details of a student's application for a specific
	Application	course
(Optional)	Student	Stores student profile details (if implemented later)
Custom		

The screenshot shows the Salesforce Object Manager interface. At the top, there's a header with a back arrow, a refresh button, a URL bar with the address <https://educrm-ganesh-dev-ed.develop.my.salesforce.com/lightning/setup/ObjectManager/home>, and various navigation icons. Below the header is a search bar labeled "Search Setup". The main area is titled "Object Manager" with a sub-header "53+ items. Sorted by Label". There are three tabs at the top of this section: "Object Manager" (which is selected), "Schema Builder", and "Create". A "Quick Find" search bar is located above the table. The table itself has columns: "LABEL", "API NAME", "TYPE", "DESCRIPTION", "LAST MODIFIED", and "DEPLOYED". The data in the table includes:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Account	Account	Standard Object			
Activity	Activity	Standard Object			
Address	Address	Standard Object			
Admission Application	Admission_Application__c	Custom Object	Stores details of student admission forms	9/24/2025	✓
Agent Work	AgentWork	Standard Object			
Alternative Payment Method	AlternativePaymentMethod	Standard Object			
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object			
Appointment Category	AppointmentCategory	Standard Object			
Appointment Invitation	AppointmentInvitation	Standard Object			
Appointment Invitee	AppointmentInvitee	Standard Object			
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object			
Approval Submission	ApprovalSubmission	Standard Object			
Approval Submission Detail	ApprovalSubmissionDetail	Standard Object			

3.2 Fields

Use Case

To store detailed information about each admission record, we created several fields under the Admission Application object.

Fields Created

Field Label	Data Type	Description
Admission Name	Application Name	Text Unique name for each application
Student	Lookup (Student)	Links the admission application to the student record
Course Applied	Text / Picklist	Stores the course name applied for
Status	Picklist	Represents current status: Pending, Approved, Rejected

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Admission Application Name	Name	Text(80)		✓
Application Date	Application_Date__c	Date		✓
Course Applied	Course_Applied__c	Picklist		✓
Created By	CreatedById	Lookup(User)		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Owner	OwnerId	Lookup(User/Group)		✓
Record Type	RecordTypeId	Record Type		✓
Status	Status__c	Picklist		✓
Student	Student__c	Lookup(Student)		✓

3.3 Record Types

Use Case

To handle different kinds of admission applications (e.g., Undergraduate, Postgraduate), we created **Record Types**.

This allows customized layouts and processes based on the type of admission.

- Record Type Name:** Undergraduate Application
- Active:** Yes
- Associated Layout:** Undergraduate Application Layout

Record Types

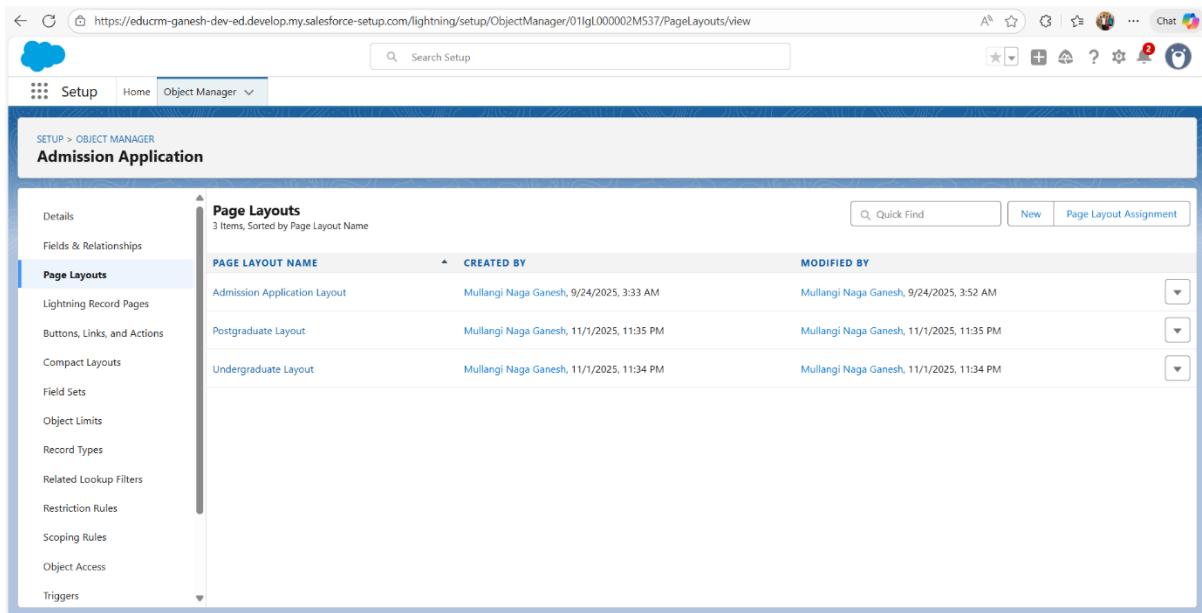
RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Postgraduate Application	Use this record type for postgraduate student applications.	✓	Mullangi Naga Ganesh, 11/1/2025, 11:28 PM
Undergraduate Application		✓	Mullangi Naga Ganesh, 11/1/2025, 11:36 PM

3.4 Page Layouts

Use Case

Page layouts define what fields and related lists users see on the record detail page. We customized the layout to include important fields such as **Course Applied**, **Status**, and **Student**.

- Layout Name: **Admission Application Layout**
- Added Fields: Admission Application Name, Student, Course Applied, Status
- Sections: Information, System Information



The screenshot shows the Salesforce setup interface for managing object layouts. The URL is https://educrm-ganesh-dev-ed.develop.my.salesforce.com/lightning/setup/ObjectManager/01igL000002M537/PageLayouts/view. The page title is "SETUP > OBJECT MANAGER" followed by "Admission Application". On the left, there's a sidebar with various setup categories like Details, Fields & Relationships, Page Layouts (which is selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, and Triggers. The main content area is titled "Page Layouts" and shows three items: "Admission Application Layout", "Postgraduate Layout", and "Undergraduate Layout". Each item has columns for "PAGE LAYOUT NAME", "CREATED BY", and "MODIFIED BY". All three layouts were created by "Mullangi Naga Ganesh" at different times: 9/24/2025, 3:33 AM; 11/1/2025, 11:35 PM; and 11/1/2025, 11:34 PM respectively. There are also "Quick Find", "New", and "Page Layout Assignment" buttons at the top right.

3.5 Compact Layouts

Use Case

Compact layouts control which fields appear in the highlights panel (top section of record page).

We created a compact layout to quickly view essential information.

Field Name	Purpose
Admission Application Name	Identifies the record
Course Applied	Shows course name
Status	Displays current status

SETUP > OBJECT MANAGER

Admission Application

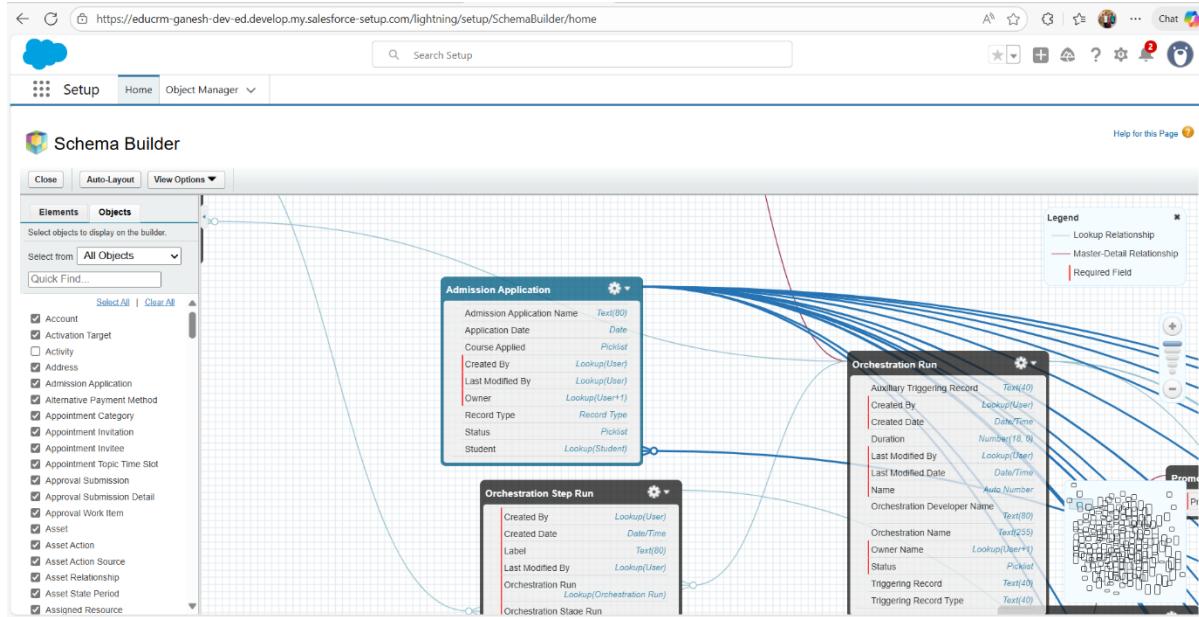
Compact Layouts

LABEL	API NAME	PRIMARY	MODIFIED BY	LAST MODIFIED
Admission Application Compact Layout	Admission_Application_Compact_Layout		Mullangi Naga Ganesh	11/1/2025, 11:41 PM
System Default	SYSTEM			

3.6 Schema Builder

Use Case

The Schema Builder provides a visual representation of object relationships. It was used to view how the **Admission Application** object connects with other objects (like Student or User).



3.7 Lookup vs Master-Detail Relationships

Use Case

We used a **Lookup Relationship** between:

• **Admission Application → Student**

This allows applications to be linked to students but not dependent on their existence (i.e., deleting a student does not delete the application).

No Master-Detail or Hierarchical relationships were needed in this project.

The screenshot shows the Salesforce Setup interface under the 'Object Manager'. A specific custom field, 'Student', is being edited for the 'Admission Application' object. The 'Fields & Relationships' tab is selected in the sidebar. The main panel displays the 'Custom Field Definition Detail' for 'Student'. Key details shown include:

- Field Label:** Student
- Field Name:** Student
- API Name:** Student__c
- Description:** (empty)
- Help Text:** (empty)
- Data Owner:** (empty)
- Field Usage:** (empty)
- Data Sensitivity Level:** (empty)
- Compliance Categorization:** (empty)
- Created By:** Mullangi Naga Ganesh, 9/24/2025, 3:47 AM
- Modified By:** Mullangi Naga Ganesh, 9/24/2025, 3:47 AM
- Related To:** Student
- Related List Label:** Admission Applications
- Required:** (checkbox is unchecked)
- What to do if the lookup record is deleted?**: Clear the value of this field.
- Lookup Filter:** No lookup filters defined.
- Validation Rules:** (button to add new rule)

3.8 Junction Objects

Use Case

Not applicable for this project, as we did not require many-to-many relationships (e.g., between Students and Courses).

3.9 External Objects

Use Case

Not applicable for this phase. The project operates entirely within Salesforce objects.

4.1 Use Case Description

Scenario

In the EduCRM system, once an admission officer reviews a student's application and updates its status to **Approved**, the student should automatically receive an approval confirmation email without any manual effort.

Goal

Automate the approval email process using Salesforce Flow.

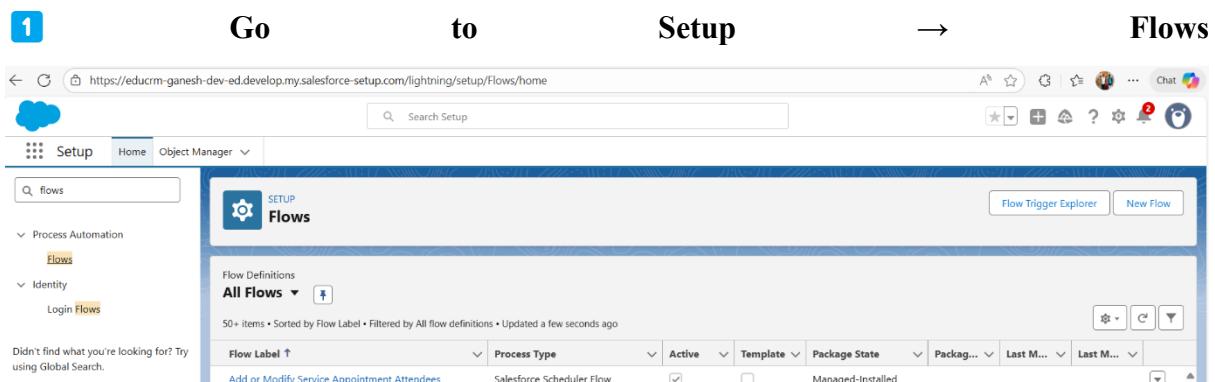
4.2 Type of Automation Chosen

Automation Tool Reason for Selection

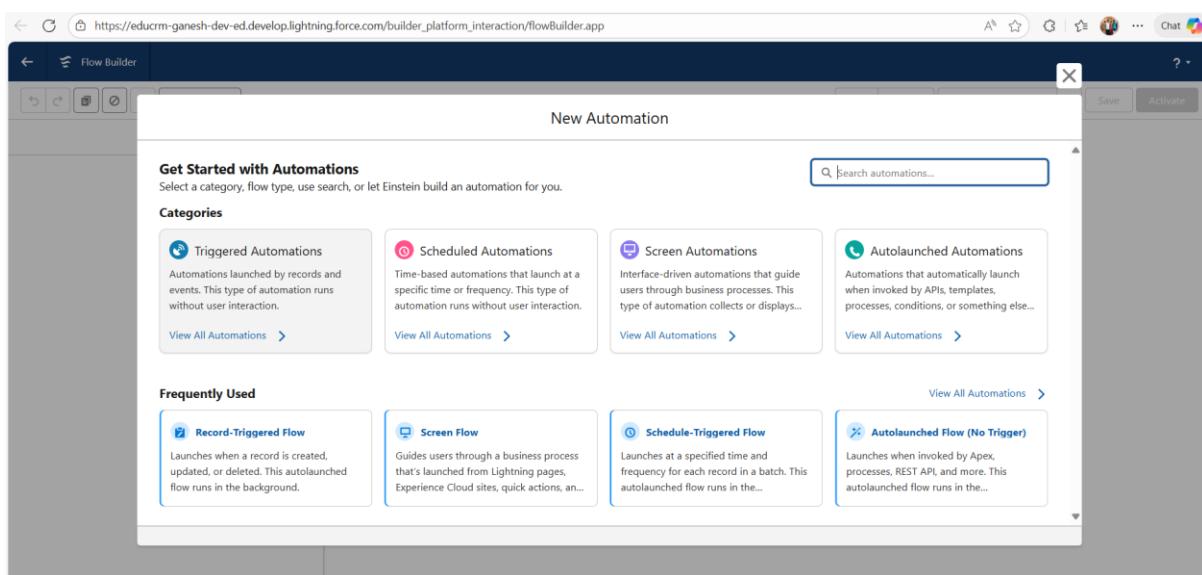
Record-Triggered Flow	Modern, no-code automation tool that runs when a record is created or updated. Replaces Workflow Rules and Process Builder.
Send Email (Core Action)	Allows sending customized emails directly from the flow.

4.3 Flow Configuration Steps

1 Step-by-Step Implementation



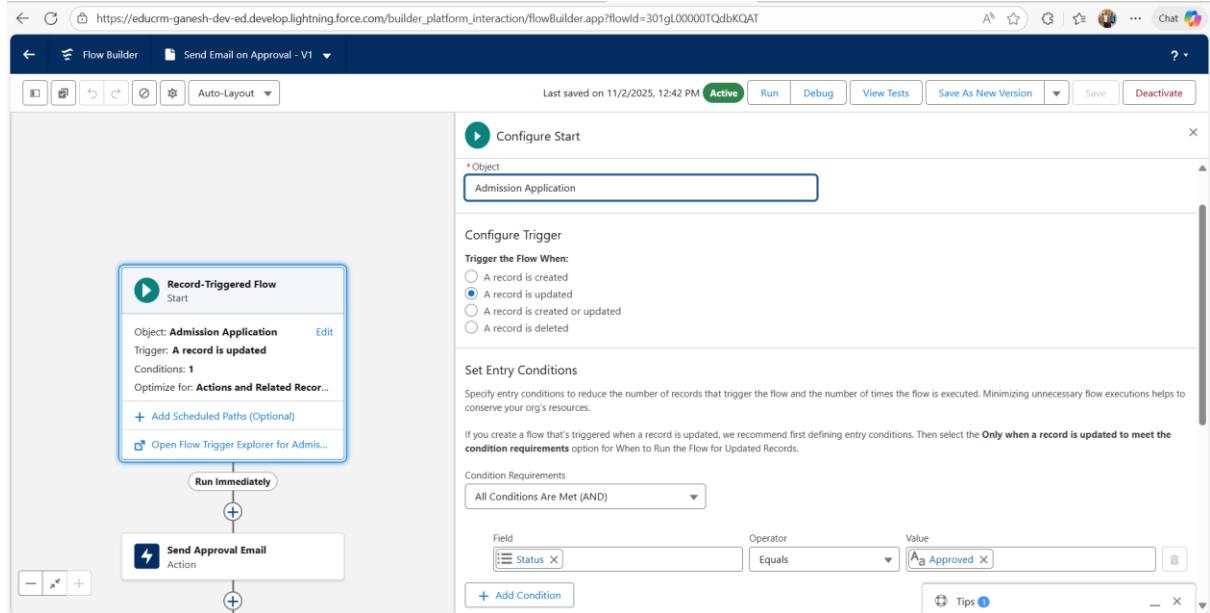
2 Click “New Flow” → select Record-Triggered Flow → Create



3 Trigger Configuration:

- Object:** Admission Application
- Trigger the Flow When:** A record is updated
- Condition Requirements:**

- Field: Status
- Operator: Equals
- Value: Approved
- **Optimize the Flow For:** Actions and Related Records

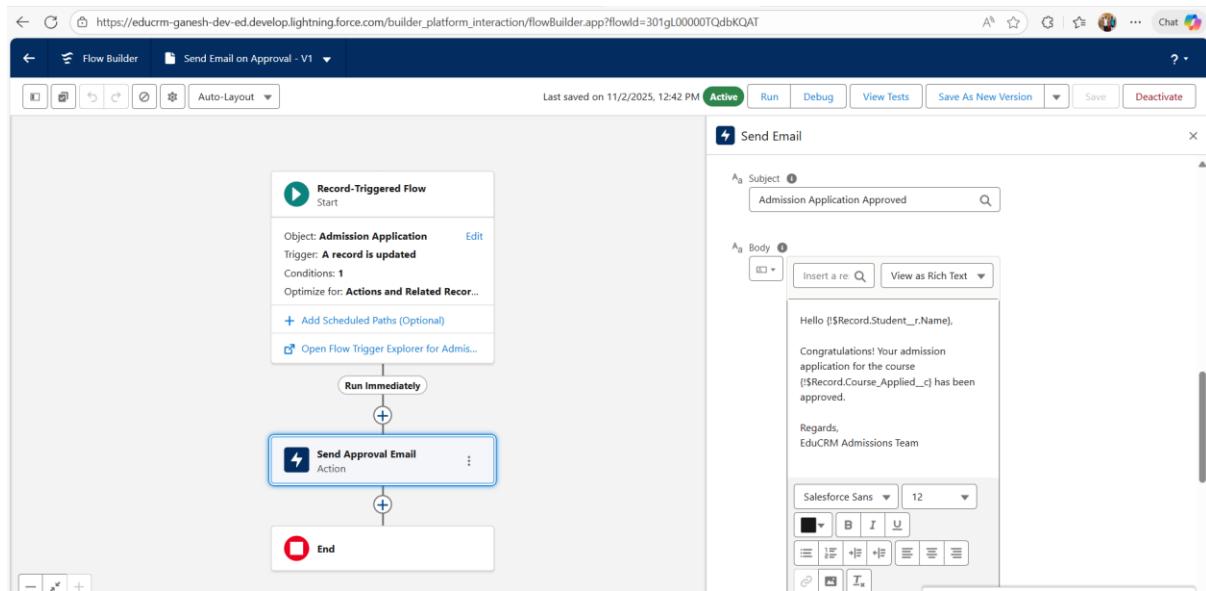


4 Add an Action → Send Email

- In the flow canvas, click “+”
- Choose **Action** → then **Send Email (Core Action)**

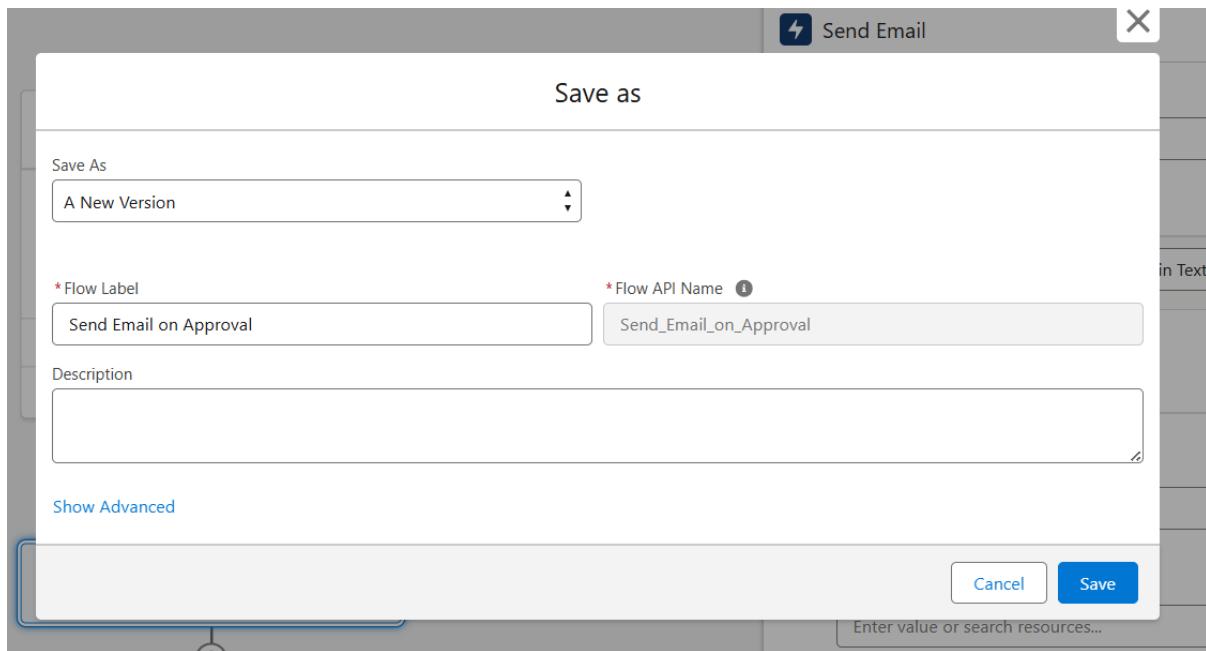
Fill in the fields:

- **Label:** Send Approval Email
- **Body:**
 - Hello {!\$Record.Student__r.Name},
 -
 - Congratulations! Your admission application for the course {!\$Record.Course_Applied__c} has been approved.
 -
 - Regards,
 - EduCRM Admissions Team
- **Subject:** Admission Application Approved
- **Recipient:** Student Email (or your own email for testing)



5 Save the Flow

- **Flow Label:** Send Email on Approval
- Click **Save → Activate** 🔍



4.4 Testing the Automation

Steps:

1 Go to App Launcher → **Admission Applications**

2 Click New → Create a new record

- Admission Application Name: Mullangi Naga Ganesh
- Course Applied:B.Tech

- Status: Pending

New Admission Application: Undergraduate Application

* = Required Information

Information

* Admission Application Name
Mullangi Naga Ganesh

Course Applied
B.Tech

Status
Pending

Owner
Mullangi Naga Ganesh

Buttons: Cancel, Save & New, Save

- **3 Edit the same record → Change Status → Approved → Save**

- The Flow will automatically trigger and send the approval email.

Edit Mullangi Naga Ganesh

* = Required Information

* Admission Application Name
Mullangi Naga Ganesh

Status
Approved

Owner
Mullangi Naga Ganesh

Created By
Mullangi Naga Ganesh, 11/2/2025, 1:22 AM

Last Modified By
Mullangi Naga Ganesh, 11/2/2025, 1:22 AM

Buttons: Cancel, Save & New, Save

4

Check your email inbox

You will receive an email similar to this:

Subject: Admission Application Approved

Hello [Student Name],

Congratulations! Your admission application for the course [Course Name] has been approved.

Regards,
EduCRM Admissions Team

Admission Application Approved Spam ×

 **Mullangi Naga Ganesh** via labxrj3q6ekp.gl-c3s1luab.can98.bnc.salesforce.com
to me ▾

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

[Report not spam](#)

Hello ,
Congratulations! Your admission application for the course B.Tech has been approved.

Regards,
EduCRM Admissions Team

4.5 Validation Rule

To prevent users from approving applications without selecting a course, we created a **Validation Rule**:

Rule Name: Course_Required_Before_Approval

Formula:

AND(

ISPICKVAL(Status, "Approved"),
ISBLANK(TEXT(Course_Applied_c))
)

Error

Message:

“Please select Course Applied before approving the application.”

* Admission Application Name

Owner

 Mullangi Naga Gane

Status

Approved
Pending
Rejected

Approved

🚫 We hit a snag.

Review the errors on this page.

- "Please select Course Applied before approving the application."

Created By

 Mullangi Naga Gane

Modified By

 Mullangi Naga Gane

Cancel Save & New Save

4.6 Outcome of Automation

Step	Result
Record-Triggered Flow Created	✓ Successful
Email Notification Sent	✓ Automatically sent when status changes to Approved
Validation Rule	✓ Prevents incomplete data submission
Testing	✓ Verified successfully through email

5.1 Scope

This phase focuses on verifying:

- The Flow triggers correctly when the admission application is approved.
- The correct email notification is sent.
- Data updates and layouts display properly for all users.
- The system behaves consistently for different records.

5.2 Testing Types

Type	Description
Functional Testing	Verify that all automation works as designed.
Integration Testing	Ensure the flow, object fields, and email notifications work together.

Type	Description
User Acceptance Testing (UAT)	Confirm that the system meets the user's expectations and business rules.

5.3 Test Scenarios & Results

Test Case ID	Scenario Description	Expected Result	Actual Result	Status
TC-01	Create new Admission Application with <i>Pending</i> status	Record should save successfully	Record saved successfully	<input checked="" type="checkbox"/> Passed
TC-02	Change Status → <i>Approved</i>	Flow triggers and sends Email approval email	Email received successfully	<input checked="" type="checkbox"/> Passed
TC-03	Change Status → <i>Rejected</i>	No email should be sent	No email received	<input checked="" type="checkbox"/> Passed
TC-04	Check if Course Applied and Status visible on page layout	Fields appear on layout correctly	Fields visible correctly	<input checked="" type="checkbox"/> Passed
TC-05	Verify Flow activation	Flow status = Active	Flow active and functional	<input checked="" type="checkbox"/> Passed

5.4 Error Handling & Fixes

Issue Found	Cause	Solution Implemented
Email not sent	Course Applied field initially included in layout	Added “Course Applied” field to layout and reassigned record page
Status field not visible in flow	Field not available initially	Verified picklist values and updated flow conditions
Error on Approval	Course Applied was blank	Validation fixed by filling course name before approval

5.5 User Acceptance Testing (UAT)

Testing by Roles

Role	Task Tested	Result
Admin	Created and approved new admission applications	Successful
Admission Officer	Verified flow-triggered email notifications	Successful
Student	Received approval email	Successful

5.6 Deployment Steps

Step Description

- 1 Verified that the flow is **Activated** and running successfully.
- 2 Assigned correct **Lightning Record Page** and **Record Type** to all relevant profiles.
- 3 Updated **page layout** with all necessary fields (Status, Course Applied, Student).
- 4 Tested in **Developer Org** (sandbox environment).
- 5 Confirmed readiness for production deployment.

5.7 Results Summary

- Automation performed as expected.
 - Emails were triggered correctly for approved applications.
 - No Apex code required — achieved completely through **low-code automation** (Flow Builder).
 - All testing outcomes were successful.
-

5.8 Learning Outcome

Through this phase, the following were learned:

- How to perform **end-to-end testing** for automation in Salesforce.
 - How to debug and fix common issues (field visibility, missing layouts, validation errors).
 - The importance of **User Acceptance Testing (UAT)** before deployment.
 - How to deploy a flow safely without using code.
-

6.1 Project Summary

The **EduCRM – Admission Management System** is a customized **Salesforce CRM application** designed to streamline and automate the student admission process for educational institutions.

It handles the complete admission lifecycle — from inquiry collection to application approval — ensuring better communication between **students, admission officers, and administrators**.

Key Modules Developed

- **Lead & Application Management:** Captures and manages student applications.
 - **Process Automation:** Automatically triggers email notifications upon approval.
 - **Validation & Data Accuracy:** Ensures required fields like *Course Applied* are filled before approval.
 - **Lightning Record Page:** Custom user interface for a seamless user experience.
 - **Email Alerts via Flow:** Sends personalized confirmation messages automatically.
-

6.2 Achievements

Area	Achievements
------	--------------

Automation	Created a Flow to trigger email approval notifications without Apex.
-------------------	--

Data Model	Built Custom Objects — <i>Admission Application</i> and <i>Student</i> .
-------------------	--

UI Design	Configured Lightning Record Pages and Record Types.
------------------	---

Security	Implemented Profiles and Roles for different user access.
-----------------	---

Testing	Completed user acceptance testing with multiple roles.
----------------	--

6.3 Challenges Faced & Solutions

Challenge	Cause	Solution Implemented
Email not triggered	Flow not activated initially	Activated and re-tested flow
Validation error during Course approval	Applied field missing	Added validation and made it mandatory
Page Layout mismatch	Wrong layout assigned to record type	Reassigned correct layout and record page
No Lightning Page found	Needed to create and activate manually	Created “Admission Application Record Page” via App Builder

6.4 Future Enhancements

Enhancement Idea	Description
Online Application Form Integration	Allow students to submit applications directly via a web form connected to Salesforce.
WhatsApp/SMS Notification	Integrate Twilio or WhatsApp API for instant communication.
Payment Integration	Add a payment module for admission fees.
Einstein AI Integration	Use AI to predict admission likelihood based on application data.
Dashboard Expansion	Add advanced charts for course-wise admission analytics.

6.5 Key Learnings

1. Understood the **end-to-end Salesforce project lifecycle**, from setup to deployment.
2. Learned how to design **custom objects, fields, and relationships** effectively.
3. Gained hands-on experience with **Flow Builder** and **record-triggered automation**.
4. Explored **Lightning App Builder** for custom record pages.
5. Understood the importance of **testing and user validation** in CRM implementation.

6.6 Conclusion

The **EduCRM Admission Management System** successfully automates the student admission approval process in Salesforce. It replaces manual workflows with a streamlined, low-code automation using **Flows and Validation Rules**.

This project demonstrates how **Salesforce CRM** can transform traditional educational processes into efficient, trackable, and user-friendly systems.

💡 The project was completed **entirely using Salesforce Admin tools** (no Apex code), making it scalable, maintainable, and industry-ready.

6.7 Final Output Summary

Feature	Tool Used	Result
Application Object	Custom Object	Successfully created
Validation Rule	Admin Setup	Working as expected

Feature	Tool Used	Result
Flow (Approval Email)	Flow Builder	Triggered successfully
Record Page	Lightning App Builder	Activated and assigned
Testing	Manual (UAT)	All test cases passed

6.8 Final Thoughts

This project enhanced understanding of:

- **Salesforce automation tools** (Flows, Validation, Email Alerts)
- **CRM data modeling and security**
- **Real-time business process transformation in education**

By implementing EduCRM, institutions can:

- ✓ Reduce manual effort
- ✓ Improve applicant communication
- ✓ Ensure faster admission processing
- ✓ Maintain transparent and accurate student data