

Project Title: A CRM Application to Manage the Services offered by an Institution

1. Project Overview

This project focuses on developing and implementing *ServiceConnect*, a comprehensive Customer Relationship Management (CRM) solution built on the Salesforce platform. It addresses the challenges faced by *EduConsultPro Institute* in managing its service lifecycle—from initial inquiry to service enrollment, ongoing support, and post-service follow-up. The goal is to create a centralized, efficient, and scalable system to enhance operational efficiency, improve communication, and provide data-driven insights. This will lead to an improved experience for customers and staff, fostering service excellence and supporting the long-term growth of *EduConsultPro Institute*.

2. Objectives

Business Goals:

- Enhance Student Onboarding: Streamline the application process for a smooth experience.
- Optimize Appointment Scheduling: User-friendly system minimizing scheduling conflicts.
- Improve Communication & Collaboration: Seamless information sharing between stakeholders.
- Boost Operational Efficiency: Automate tasks, reduce manual entry, and minimize errors.
- Data-Driven Decision Making: Gather data on trends and performance for informed decisions.
- Increase Student Enrollment and Retention: Positive student experience fostering engagement.
- Scalability and Future Growth: Adaptable solution for evolving institutional needs.

Specific Outcomes:

- Functional Salesforce CRM application named "EduConsultPro."
- Custom objects (Course, Consultant, Student, Appointment, Registration, etc.) with

defined fields.

- Multi-stage screen flows for student application, appointment scheduling, and case management.
- Apex triggers for welcome case creation, appointment confirmations, and data validation.
- User-friendly Lightning App interface with custom dashboards for easy navigation.
- Intuitive appointment approval process with notifications.
- Comprehensive testing (unit and user acceptance) and deployment.
- Ongoing maintenance and support.
- Detailed documentation (data models, UI, logic, testing, training).

3. Salesforce Key Features and Concepts Utilized

- **Custom Objects & Fields:** Representing the institution's data model.
- **Relationships (Lookup & Master-Detail):** Linking related data between objects.
- **Lightning App Builder:** Creating a tailored and intuitive user interface.
- **Screen Flows:** Automating complex processes with user input screens and conditional logic.
- **Apex Triggers & Classes:** Enforcing business logic, data validation, and bulkification.
- **Approval Processes:** Managing authorization workflows with notifications.
- **Standard Objects (e.g., Case):** Handling student inquiries and support.
- **Email Templates & Alerts:** Standardized communication for various events.

4. Detailed Steps to Solution Design

Create Objects From Spreadsheet

- **Create Course Object:** A custom object named "Course" was created using the data provided in the 'Course' spreadsheet. This involved mapping the spreadsheet columns to the Salesforce fields in the Course object.

SETUP > OBJECT MANAGER
Course - Course

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

Details

Edit Delete

Description

API Name
Course_Course__c

Custom
✓

Singular Label
Course - Course

Plural Label
Course - Course

Enable Reports
✓

Track Activities
✓

Track Field History

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

- **Create Remaining Objects:** Similar to the Course object, custom objects were created for Consultant, Student, Appointment, and Registration using the provided spreadsheets. Field mapping ensured data integrity during the import process.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Consultant - Consultant	Consultant_Consultant__c	Custom Object		04/01/2025	✓
LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Appointment - Appointment	Appointment_Appointment__c	Custom Object		04/01/2025	✓
LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Student - Student	Student_Student__c	Custom Object		04/01/2025	✓

- **Create Relationship Among the Objects:** Lookup relationships were established between the objects to connect related data:
 - Appointment to Student (Student lookup on Appointment object)
 - Appointment to Consultant (Consultant lookup on Appointment object)
 - A custom object, "Registration," was created to store student and course details. Lookup relationships were added:
 - Registration to Student
 - Registration to Course
 - A lookup relationship was also established between Student and Case objects.

SETUP > OBJECT MANAGER

Appointment - Appointment

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

Object Access

Triggers

Fields & Relationships

11 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment - Appointment	Name	Text(80)		✓
Appointment Date/Time	Appointment_DateTime__c	Date/Time		
Appointment No	Appointment_No__c	Number(18, 0)		
Consultant - Consultant	Consultant_Consultant__c	Lookup(Consultant - Consultant)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Notes	Notes__c	Long Text Area(131072)		
Owner	OwnerId	Lookup(User,Group)		✓
Purpose/Topic	PurposeTopic__c	Text(255)		
Status	Status__c	Picklist		
Student - Student	Student_Student__c	Lookup(Student - Student)		✓

Configure The Case Object

The standard Case object was configured to include custom picklist values:

- **Type Field:** 'Immigration' and 'Visa Application' values added.
- **Status Field:** 'Open' and 'In-Progress' values added. (Consider adding 'Closed' or similar for completeness).

SETUP > OBJECT MANAGER

Case

Details

Fields & Relationships

Case Page Layouts

Case Close 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

Fields & Relationships

35+ Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

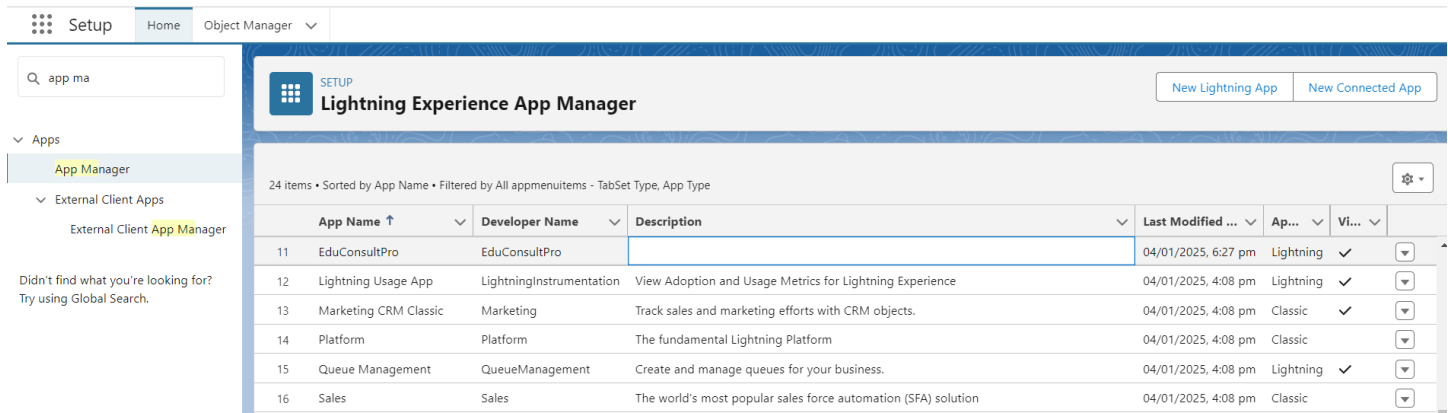
Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Account Name	AccountId	Lookup(Account)		✓
Asset	AssetId	Lookup(Asset)		✓
Business Hours	BusinessHoursId	Lookup(Business Hours)		
Case Number	CaseNumber	Auto Number		✓
Case Origin	Origin	Picklist		
Case Owner	OwnerId	Lookup(User,Group)		✓
Case Reason	Reason	Picklist		
Case Source	SourceId	Lookup(Email Message,Messaging Session)		✓
Closed When Created	IsClosedOnCreate	Checkbox		
Contact Email	ContactEmail	Email		
Contact Fax	ContactFax	Phone		

Create A Lightning App

A Lightning App named “EduConsultPro” was created. This app included tabs for Home, Students, Courses, Consultants, Appointments, Registrations, and Cases, providing users with a centralized location to access all functionalities. The app was made available to the System Administrator profile.



The screenshot shows the Salesforce Lightning Experience App Manager interface. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area displays a list of apps under the heading "Lightning Experience App Manager". The list includes columns for App Name, Developer Name, Description, Last Modified, App Type, and Visibility. The 'EduConsultPro' app is listed at the top of the results.

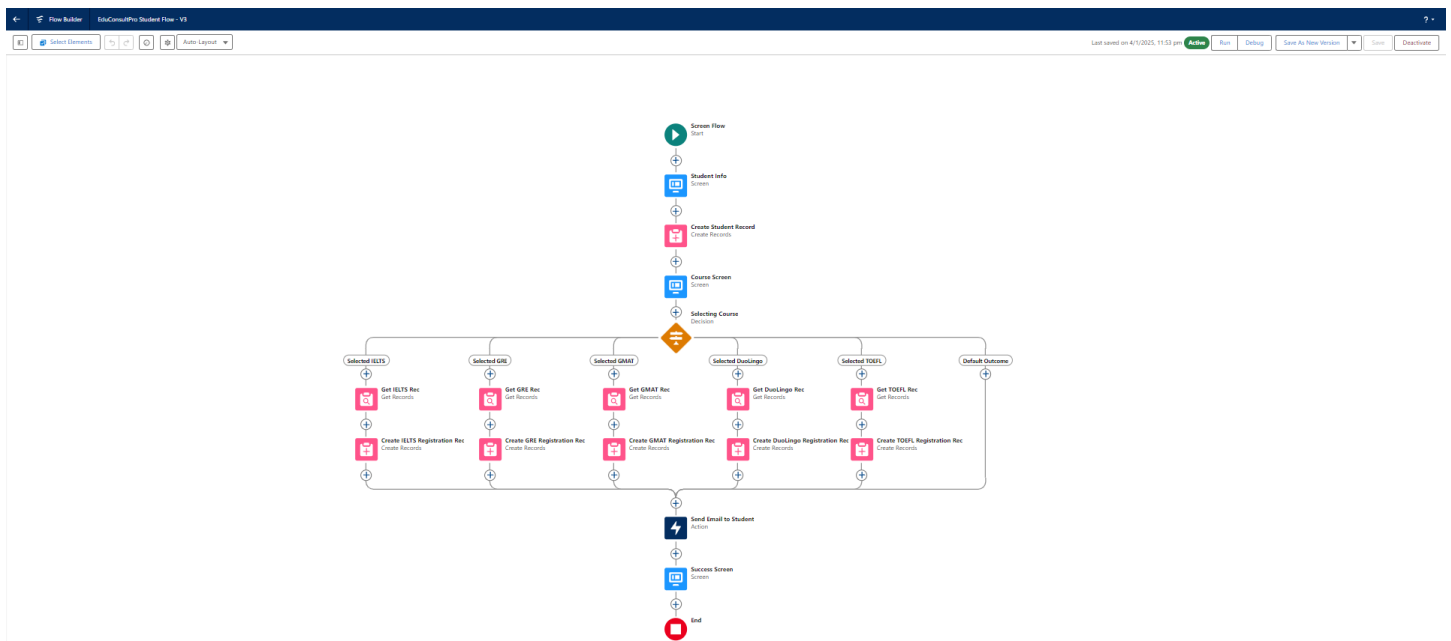
App Name	Developer Name	Description	Last Modified	App Type	Visibility
EduConsultPro	EduConsultPro		04/01/2025, 6:27 pm	Lightning	✓
Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	04/01/2025, 4:08 pm	Lightning	✓
Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	04/01/2025, 4:08 pm	Classic	✓
Platform	Platform	The fundamental Lightning Platform	04/01/2025, 4:08 pm	Classic	✓
Queue Management	QueueManagement	Create and manage queues for your business.	04/01/2025, 4:08 pm	Lightning	✓
Sales	Sales	The world's most popular sales force automation (SFA) solution	04/01/2025, 4:08 pm	Classic	✓

Create A ScreenFlow For Student Admission Application Process

- **Add Screen Element (Student Info):** A screen element was added to collect student information. This element displayed fields from the Student object. A record variable resource StudentRecordRes was created.
- **Create Student Record Using Create Element:** A create element, “Create Student Record,” was added to create a new Student record using the data collected from the “Student Info” screen.
- **Add Screen Element (Course Screen):** This screen allows students to select a course (IELTS, GRE, GMAT, Duolingo, TOEFL) from a picklist. Choice variables were created for each course option.
- **Add Decision Element (Selecting Course):** A decision element checks the selected course from the "Course Screen" and routes the flow based on the selection. Outcomes were defined for each course option (e.g., "Selected IELTS").
- **Add GET Record Element:** For each course outcome, a Get Record element retrieves

the corresponding Course record based on the selected course name.

- **Create Registration Record Using Create Records Element:** A create element creates a Registration record, linking the newly created Student record and the retrieved Course record. This was done for each course outcome path.
- **Create Email Text Template Variables For Email Body And Subject:** Two text template resources were created:
 - StuRegistrationEmailTextTempBody for the email body content.
 - StuRegistrationEmailTextTempSub for the email subject.
- **Add An Action Element (Send Email to Student):** An email alert action was added to send a registration confirmation email to the student. It utilizes the email templates created in the previous step.
- **Add Screen Element (Success Screen):** A final screen displays a success message to the student.



Create Users

- **User:** A new user with the Standard Platform User profile was created.
- **Configure The User Settings:** The new user was assigned a manager in their user

settings, essential for the approval process.

New User Reset Password(s) Add Multiple Users						
Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatty.00dvw00000fgyrpe2at.12xalt9f5dhw@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit	Consultant	cons	322103310141@gypce.ac.in		✓	Standard Platform User
<input type="checkbox"/> Edit	Medavarapu, Ganesh Sri Vardhan	GMeda	ganesh@gypce.sandbox		✓	System Administrator
<input type="checkbox"/> Edit	User, Integration	integ	integration@00dvw00000fgyrpe2at.com		✓	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User, Security	sec	insightssecurity@00dvw00000fgyrpe2at.com		✓	Analytics Cloud Security User
New User Reset Password(s) Add Multiple Users						

Create An Approval Process For Property Object (Should be Appointment Object)

- **Create An Email Template:** Email templates were created for Submission, Approval, and Rejection stages of the Appointment approval process.

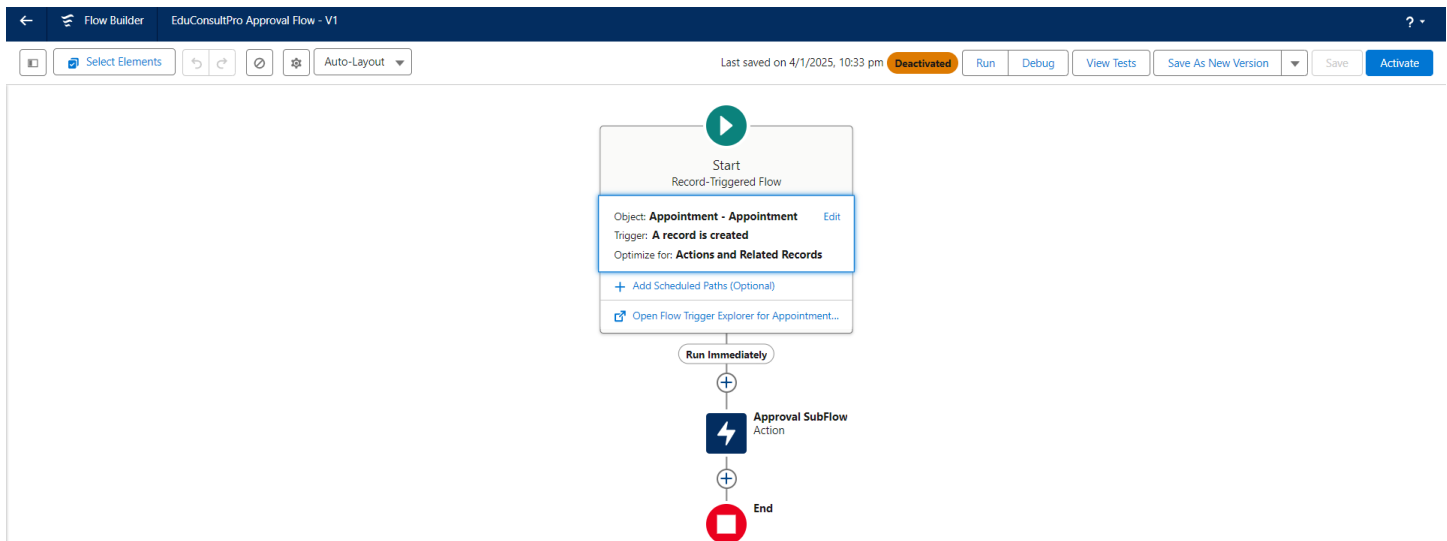
Email Templates					
Recent					
3 items					
EMAIL TEMPLATES	Email Template Name	Description	Folder	Last Modified By	Last Modified Date
Recent	Rejection Template		Project Work	Ganesh Sri Vardhan Medavarapu	4/1/2025, 7:46 pm
Created by Me	Approval Template		Project Work	Ganesh Sri Vardhan Medavarapu	4/1/2025, 7:45 pm
Private Email Templates	Submission Template		Project Work	Ganesh Sri Vardhan Medavarapu	4/1/2025, 7:42 pm
Public Email Templates					

- **Create An Approval Process:** An approval process for the Appointment object was configured. The manager of the submitting user was designated as the approver. Record editability properties were set to allow administrators or the assigned approver to edit records during the approval process. Initial Submission, Final Approval, and Final Rejection actions were configured to update the Appointment record's status and send email notifications using the respective templates.

Action	Approval Process Name	Description
Edit Activate Del	Appointment Approval	

Create A Record Triggered Flow

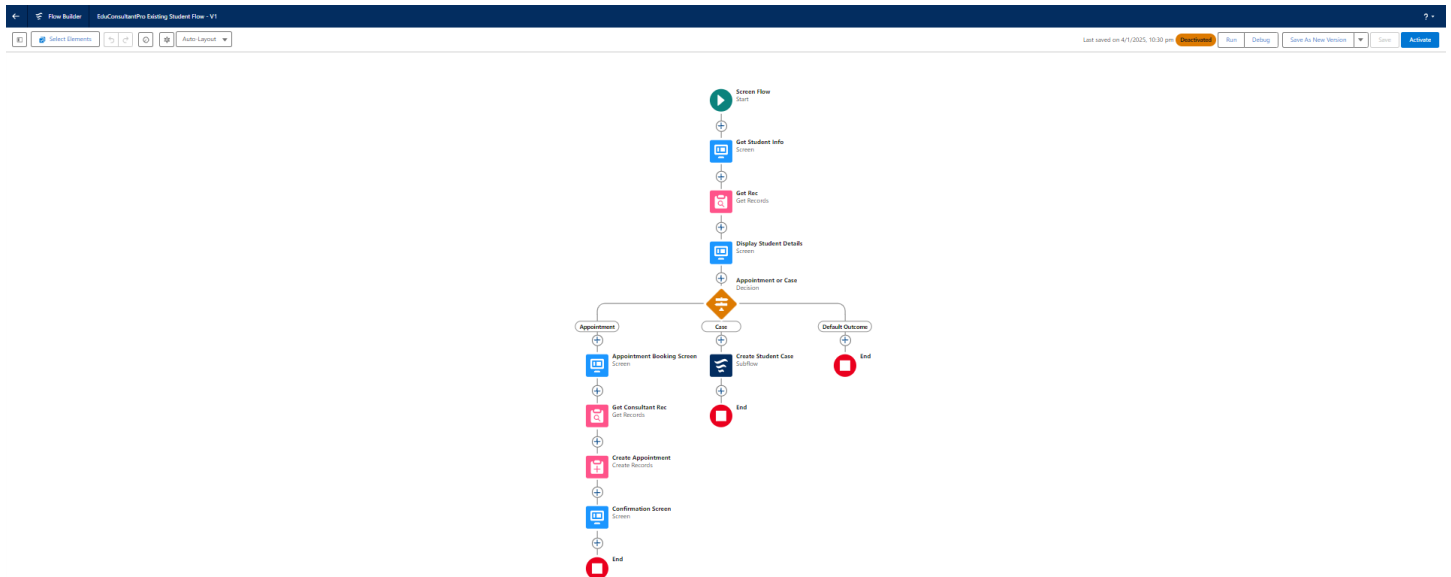
- **Configure The Start Element:** A record-triggered flow was initiated, triggering when an Appointment record is created.
- **Add An Action Element:** An action element was added to submit the newly created Appointment record for approval using the Submit for Approval action.



Create A ScreenFlow For Existing Student To Book An Appointment

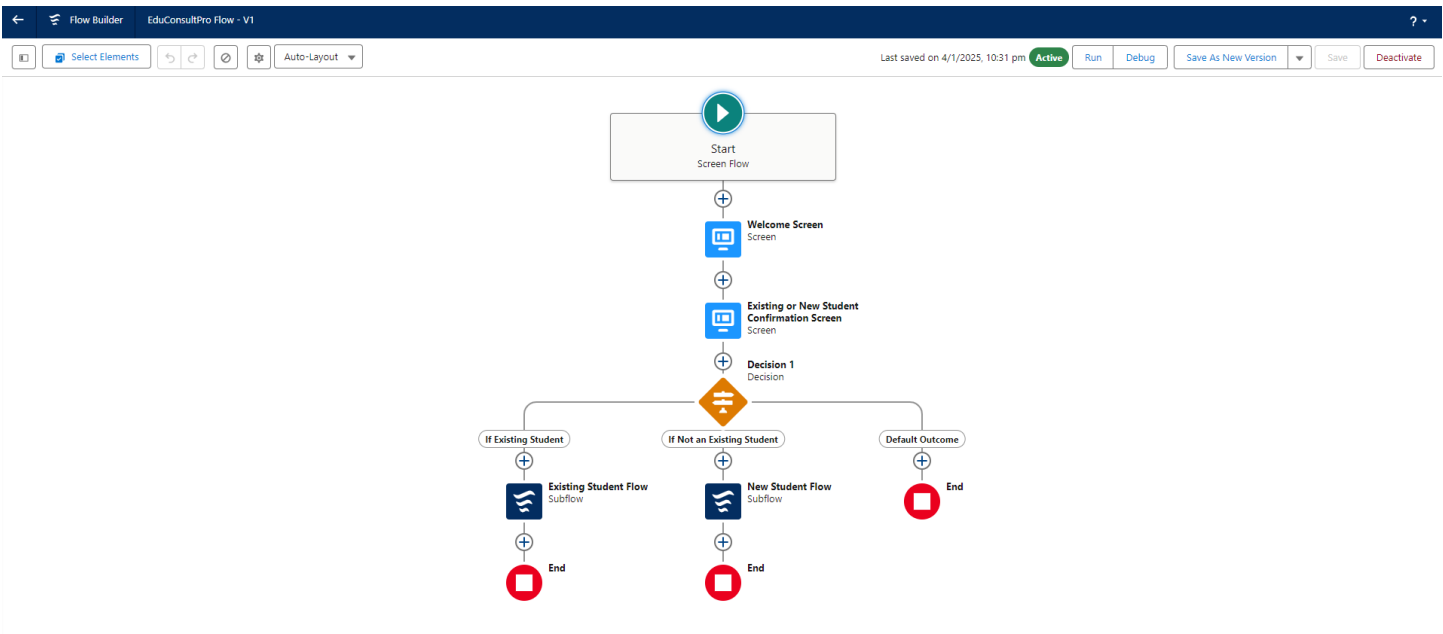
- **Add Screen Element (Get Student Info):** Collects existing student's name and email.
- **Add GET Record Element (Get Rec):** Retrieves the Student record matching the entered name and email.
- **Add Decision Element (Appointment or Case):** Determines whether the student wants to book an appointment or create a case (this branch of the flow wasn't fully described in the original instructions, so requires further definition). Presumably, a screen would be needed before this decision element to offer those choices.
- **Add Screen Element (Appointment Booking Screen):** Displays fields from the Appointment object for the student to fill in. AppointmentRecordRes resource is used.
- **Add GET Record Element (Get Consultant Rec):** Retrieves the Consultant record based on the selected consultant name.
- **Create Appointment Record Using Create Records Element (Create Appointment):** Creates an Appointment record with details provided.
- **Add Screen Element (Confirmation Screen):** Displays confirmation message with appointment details.
- **Add An SubFlow Element (Create Student Case):** (This was under the "Case" path of

the Decision element and requires further definition, assuming it involves creating a Case record if the student chose that option).



Create A ScreenFlow To Combine All The Flows At One Place

- **Add Screen Element (Welcome Screen):** Displays a welcome message.
- **Add Screen Element (Existing or New Student Confirmation Screen):** Asks the user if they are an existing student.
- **Add Decision Element (Decision 1):** Routes the flow based on user input (Existing/New Student).
- **Add An SubFlow Element (Existing Student Flow / New Student Flow):** Calls the respective Subflow based on the decision element outcome (Student Admission flow or Existing Student Appointment booking flow).



Create A Lightning App Page

- **Create a Lightning App Page:** A Lightning app page "EduConsultPro Home Page" was created and configured as the default Home page for the "Sales" app and assigned to the System Administrator profile. This page displayed the Combined Flow.

Lightning App Builder: Pages EduConsultPro Home Page

Desktop Shrink To View

Activation... Save

Components

Search...

Standard (24)

- Accordion
- Chatter Feed
- Chatter Publisher
- CRM Analytics Collection
- CRM Analytics Dashboard
- Dashboard
- Einstein Next Best Action
- Flow
- Launchpad
- List View
- Lobby Management
- LWC CRM Analytics Dashboard
- Quip Associated Documents
- Quip Document

Get more on the AppExchange

Flow Component: Add or Modify Service Appointment Attendees

This is a placeholder. Flows don't run in the canvas.

Add Component(s) Here

Add Component(s) Here

Page

* Label
EduConsultPro Home Page

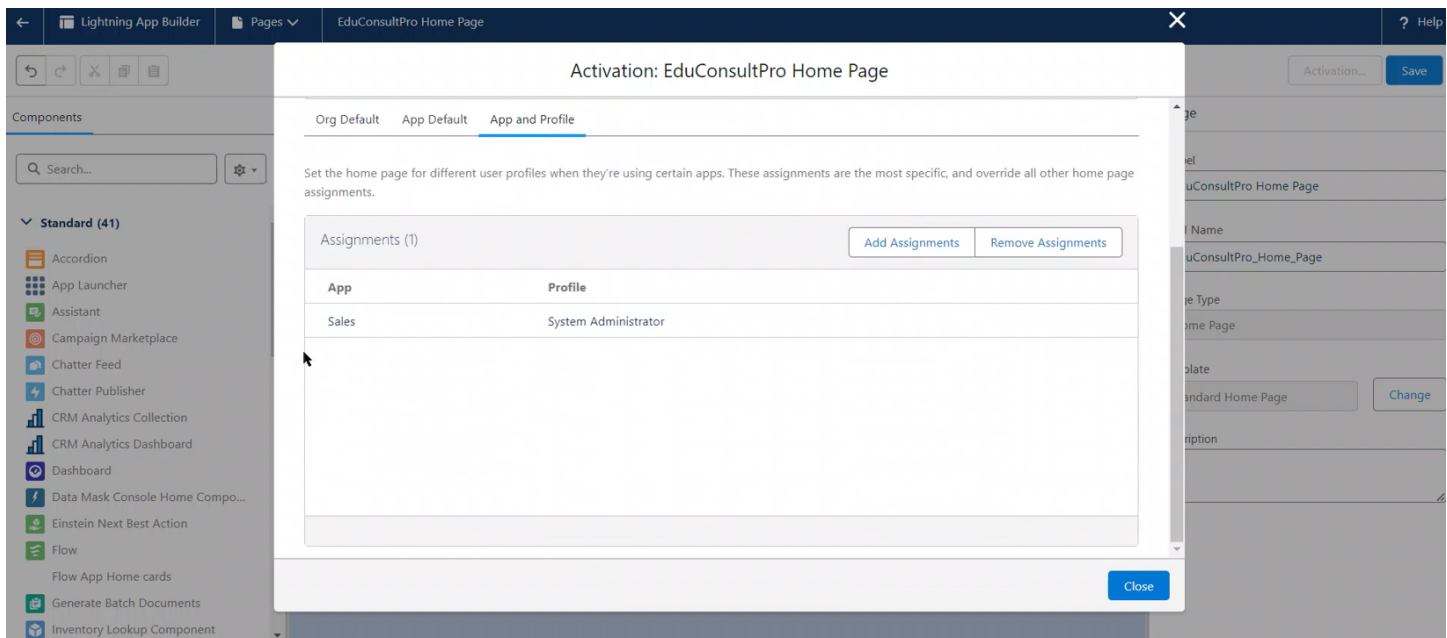
* API Name
EduConsultPro_Home_Page

* Page Type
App Page

Template
Header and Left Sidebar [Change](#)

Description

Actions
[Select...](#)



5. Testing and Validation

- **Unit Testing:** Apex unit tests with high code coverage for all triggers and classes.
- **Integration Testing:** Verification of interactions between objects and standard functionality.
- **User Acceptance Testing (UAT):** Real-world testing by users for usability and effectiveness.
- **Performance Testing:** Ensuring system responsiveness and stability under various loads.

```

Trigger1.apxt | Trigger2.apxt
Code Coverage: None | API Version: 62
1 // Trigger 1: Automatically create a welcome case when a new student is created
2 trigger StudentTrigger on Student__c (after insert) {
3     List<Case> casesToInsert = new List<Case>();
4
5     for(Student__c student : Trigger.new) {
6         // Create a welcome case for each new student
7         Case welcomeCase = new Case(
8             Subject = 'Welcome to EduConsultPro',
9             Description = 'Welcome to EduConsultPro! This case has been automatically created to track your initial onboarding.',
10            Status = 'Open',
11            Origin = 'New Student Registration',
12            Priority = 'Normal',
13            ContactEmail = student.Email__c,
14            Student__c = student.Id // Assuming there's a lookup field on Case to Student__c
15        );
16        casesToInsert.add(welcomeCase);
17    }
18
19    if(!casesToInsert.isEmpty()) {
20        try {
21            insert casesToInsert;
22        } catch(DMLException e) {
23            // Add error handling
24            for(Student__c student : Trigger.new) {
25                student.addError('Unable to create welcome case: ' + e.getMessage());
26            }
27        }
28    }
29 }
    
```

```

trigger1.apxt * Trigger2.apxt *
Code Coverage: None API Version: 62

1 // Trigger 2: Validate appointment scheduling and send notification
2 trigger AppointmentTrigger on Appointment__c (before insert, before update, after insert) {
3     if(trigger.isBefore) {
4         // Validate appointment scheduling
5         for(Appointment__c appt : Trigger.new) {
6             // Check if appointment is during business hours (9 AM to 5 PM)
7             DateTime apptDateTime = DateTime.newInstance(
8                 appt.Appointment_Date__c,
9                 appt.Appointment_Time__c
10            );
11
12            Integer hourOfDay = apptDateTime.hour();
13
14            if(hourOfDay < 9 || hourOfDay >= 17) {
15                appt.addError('Appointments can only be scheduled between 9 AM and 5 PM');
16            }
17
18            // Check if consultant is available at the requested time
19            if(trigger.isInsert || (trigger.isUpdate &&
20                appt.Consultant__c != trigger.oldMap.get(appt.Id).Consultant__c ||
21                appt.Appointment_Date__c != trigger.oldMap.get(appt.Id).Appointment_Date__c ||
22                appt.Appointment_Time__c != trigger.oldMap.get(appt.Id).Appointment_Time__c)) {
23
24                // Query for existing appointments
25                List<Appointment__c> existingAppointments = [
26                    SELECT Id
27                    FROM Appointment__c
28
29                    WHERE Consultant__c = :appt.Consultant__c
30                    AND Appointment_Date__c = :appt.Appointment_Date__c
31                    AND Appointment_Time__c = :appt.Appointment_Time__c
32                    AND Id != :appt.Id
33                ];
34
35                if(!existingAppointments.isEmpty()) {
36                    appt.addError('Consultant is already booked for this time slot');
37                }
38            }
39        }
40    }
41
42    if(trigger.isAfter && trigger.isInsert) {
43        // Send email notifications
44        List<Messaging.SingleEmailMessage> emailsToSend = new List<Messaging.SingleEmailMessage>();
45
46        for(Appointment__c appt : Trigger.new) {
47            // Query for related records to get email addresses
48            Appointment__c apptWithRelations = [
49                SELECT Id, Student__r.Email__c, Consultant__r.Email__c,
50                    Appointment_Date__c, Appointment_Time__c
51                FROM Appointment__c
52                WHERE Id = :appt.Id
53            ];
54
55            if(!emailsToSend.isEmpty()) {
56                try {
57                    Messaging.sendEmail(emailsToSend);
58                } catch(Exception e) {
59                    // Handle email sending errors
60                    System.debug('Error sending appointment notification emails: ' + e.getMessage());
61                }
62            }
63        }
64    }
65 }

```

6. Key Scenarios Addressed by Salesforce in the Implementation Project

- **Multi-Channel Communication:** Personalized communication through email and the

platform.

- **Role-Based Access Control:** Maintaining data security and confidentiality through access levels.
- **Reporting and Analytics:** Real-time monitoring of key metrics and KPIs.
- **Integration with Existing Systems (Future Consideration):** Flexibility for future expansion.

7. Conclusion

The "CRM Application to Manage the Services Offered by an Institution" project successfully developed a tailored Salesforce CRM for EduConsultPro Institute, streamlining operations, enhancing communication, and centralizing key information.

With automated processes for admissions and appointment scheduling, the CRM reduces manual work, minimizes errors, and boosts staff efficiency. Automated email notifications keep students informed throughout their engagement, enriching the overall student experience.

By consolidating all relevant data, the CRM empowers EduConsultPro with comprehensive reporting and analytics, enabling data-driven decisions and continuous improvement. The flexible Salesforce platform supports future scalability, positioning EduConsultPro Institute to grow and adapt to evolving needs while delivering efficient, student-centered service management.