

A CRM Application to Manage the Services offered by an Institution

By

MEKAPOTHULA LAKSHMI BHAVANA

Project Abstract

The project involved developing a Salesforce CRM application for EduConsultPro Institute, aimed at improving the management of student admissions, consulting requests, and immigration cases. The system enabled prospective students to submit admission applications online, with data automatically captured in Salesforce and automated email notifications sent to applicants. The project also included managing consulting services, where students could request consultations, and consultants were notified to schedule and manage appointments within the CRM. Additionally, an immigration case management system was implemented to efficiently log, process, and track cases, with automated notifications for agents and robust document management features. This solution significantly streamlined operations and enhanced the overall experience for both students and staff at EduConsultPro.

The proposed CRM application integrates various features to facilitate comprehensive service management, including:

1. **Service Catalog Management:** Users can catalog and categorize different services offered by the institution, including detailed descriptions, pricing, and availability.
2. **Client Interaction Tracking:** The CRM system tracks interactions with clients, recording inquiries, feedback, and service requests to ensure a personalized and responsive approach.
3. **Appointment Scheduling and Management:** Clients can schedule appointments or service sessions, and institutions can manage these schedules, send reminders, and handle cancellations.
4. **Performance Analytics:** The application provides analytical tools to monitor service performance, client satisfaction, and operational metrics, enabling data-driven decision-making.
5. **Automated Workflow:** It automates routine tasks such as follow-ups, notifications, and reporting, improving efficiency and reducing manual effort.
6. **Customizable Dashboards:** Institutions can customize dashboards to display relevant metrics, track service performance, and visualize data for better insight and management.
7. **Integration Capabilities:** The CRM can integrate with other systems such as accounting software, communication tools, and marketing platforms, ensuring a seamless flow of information.

INDEX

S.NO	TASK	PAGE NO
1	PROJECT ABSTRACT	2
2	TABLE OF CONTENTS	3-5
3	INTRODUCTION	6
4	TASK 1: Creating Developer Account	7
5	TASK 2: Account Activation	8
6	TASK 3: Create Objects From Spreadsheet Subtask 1: Create Course Object Subtask 2: Create Relationships Among the Objects Subtask 3: Configure the Case Object Subtask 4: Create a Lightning App	9-18
7	TASK 4: Create A ScreenFlow For Student Admission Application Process Subtask 1: Add Screen Element Subtask 2: Create Student Record Using Create Element Subtask 3: Add Course Selection Screen Subtask 4: Add Decision Element Subtask 5: Add GET Record Element Subtask 6: Create Registration Record Using Create	19-29

	Records Element Subtask 7: Create Email Text Template Variables Subtask 8: Add Action Element Subtask 9: Add Success Screen	
8	Task 5: Create Users Subtask 1: Create a User Subtask 2: Configure the User Settings	30-31
9	Task 6: Create an Approval Process for the Property Object Subtask 1: Create Email Templates Subtask 2: Create An Approval Process	32-38
10	Task 7: Create A Record-Triggered Flow Subtask 1: Configure The Start Element Subtask 2: Add An Action Element	39-41
11	Task 8: Create A ScreenFlow For Existing Students To Book An Appointment Subtask 1: Add Screen Element Subtask 2: Get Record Subtask 3: Add a Decision Element Subtask 4: Add Screen Element Subtask 5: Add GET Record Element Subtask 6: Create Appointment Record Using Create Records Element	42-49

	Subtask 7: Add Screen Element Subtask 8: Add a Subflow Element	
12	Task 9: Create A ScreenFlow To Combine All The Flows At One Place Subtask 1: Add Welcome Screen Element Subtask 2: Add Existing or New Student Confirmation Screen Subtask 3: Add Decision Element Subtask 4: Add Subflow for Existing Students Subtask 5: Add Subflow for New Students	50-55
13	Task 10: Create A Lightning App Page	56-60
	Subtask 1: Create A Lightning App Page	

INTRODUCTION

EduConsultPro Institute, a prominent educational institution, offers a diverse array of courses and programs to students from various backgrounds. With the institute's increasing popularity, managing the growing number of student admissions, consulting requests, and immigration cases became a significant challenge. To overcome these operational inefficiencies, EduConsultPro embarked on a project to implement a robust Customer Relationship Management (CRM) system using Salesforce.

Here are the key points of the Salesforce CRM implementation project at EduConsultPro Institute:

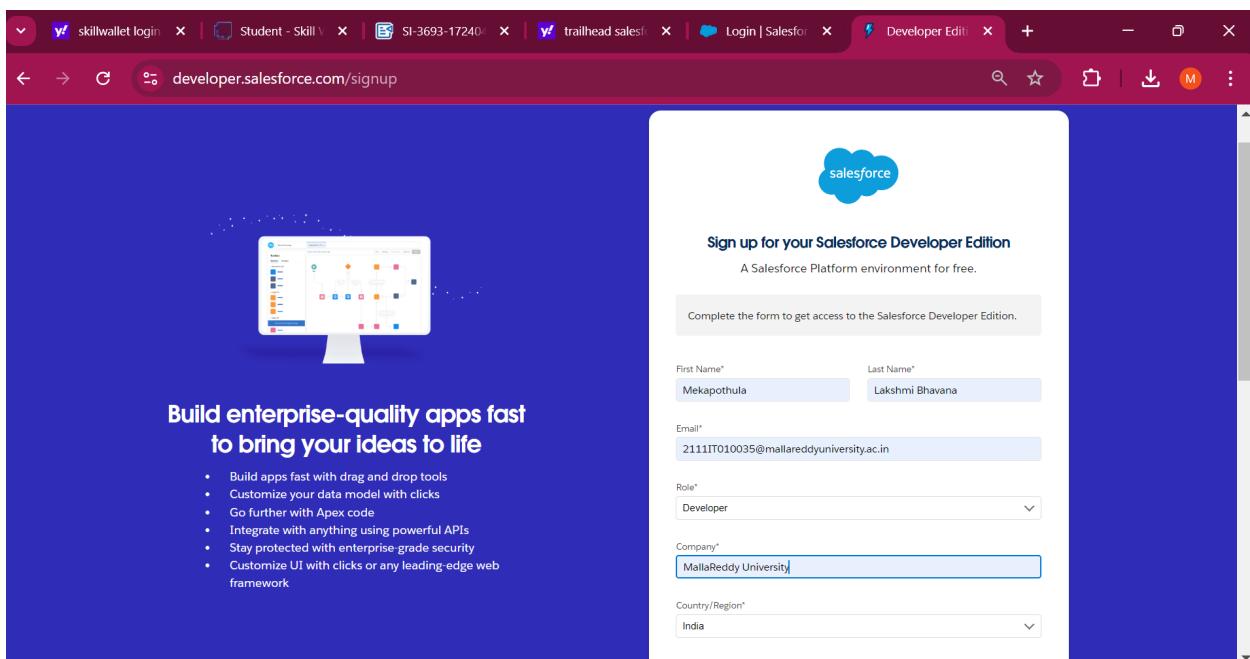
1. **Objective:** Streamline and automate core administrative processes for admissions, consulting, and immigration case management.
2. **Admission Application Management:**
 - Online application submissions captured directly in Salesforce.
 - Automated email notifications for successful submissions.
 - Reporting tools for application metrics, acceptance rates, and enrollment trends.
3. **Consulting Services Management:**
 - Requests for consulting services logged into Salesforce.
 - Automated notifications sent to consultants for appointment scheduling.
 - Tracking of appointment statuses (scheduled, completed, canceled).
4. **Immigration Case Management:**
 - Cases initiated through phone, email, or web captured in Salesforce.
 - Immediate notifications to immigration agents for case management.
 - Tools for tracking case statuses and managing documents.
5. **Outcome:** Enhanced efficiency and transparency in managing student admissions, consulting requests, and immigration cases, with a scalable solution for future growth.

A CRM Application to Manage the Services offered by an Institution

TASK 1: Creating Developer Account

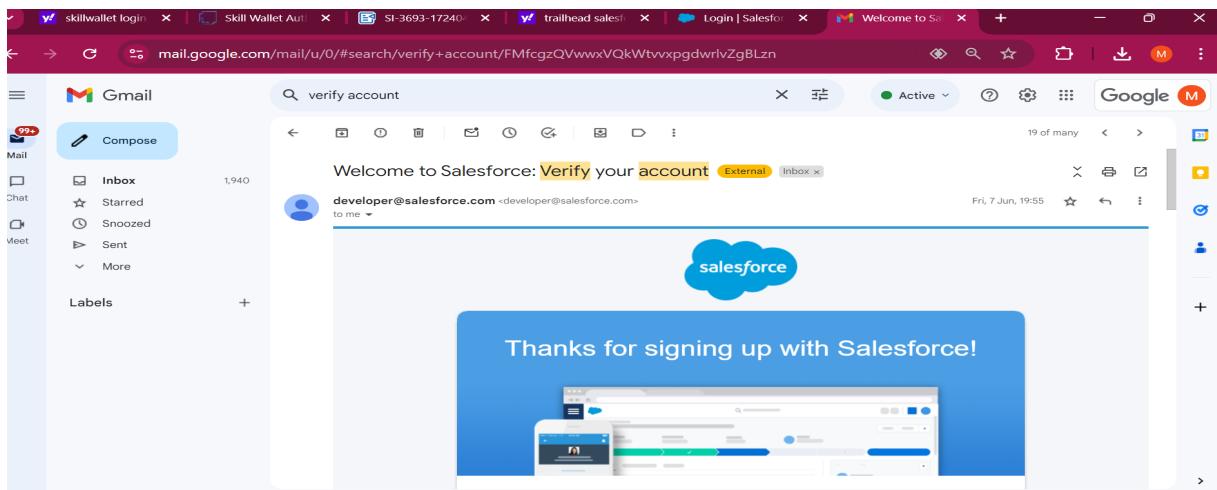
1. Go to <https://developer.salesforce.com/signup>

2. Fill up form with personal information



TASK 2: Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account.
2. Give a password and answer a security question and click on change password.
3. Then you will redirect to your salesforce setup page.



A screenshot of the Salesforce Setup Home page. The left sidebar is titled "Setup" and includes links for "Service Setup Assistant", "Commerce Setup Assistant", "Multi-Factor Authentication Assistant", "Hyperforce Assistant", "Release Updates", "Lightning Experience Transition Assistant", "Salesforce Mobile App", "Lightning Usage", "Optimizer", and "Sales Cloud Everywhere". Below this is a section for "ADMINISTRATION" with links for "Users", "Data", "Email", and "PLATFORM TOOLS". The main content area features three cards: "Get Started with Einstein Bots", "Mobile Publisher", and "Real-time Collaborative Docs". At the bottom, there is a "Most Recently Used" section showing items like "Mekapothula Lakshmi Bhavana" (User), "Course Name" (Custom Field Definition), and "Start Date" (Custom Field Definition). A search bar at the top right says "Search Setup".

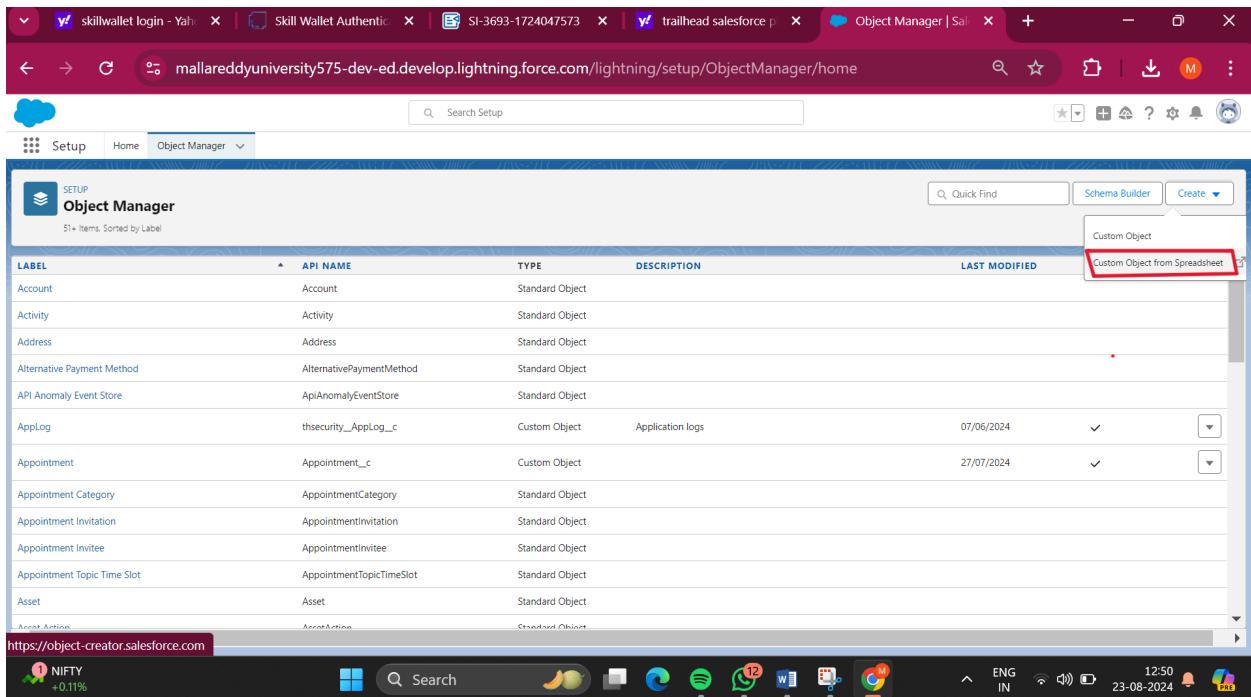
TASK 3: Create Objects From Spreadsheet

This task involves directly creating objects in Salesforce by importing data from spreadsheets and establishing relationships between them.

Subtask 1: Create CourseObject

1. Navigate to **Object Manager**.

2. Click on **Create Object from Spreadsheet**.



The screenshot shows the Salesforce Object Manager interface. At the top, there are several tabs: SETUP, Object Manager, Home, and Setup. The main area is titled "Object Manager" and displays a list of standard and custom objects. A red box highlights the "Create" button in the top right corner of the object list, which is labeled "Custom Object from Spreadsheet". The table below lists various objects with their API names, types, descriptions, and last modified dates.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED
Account	Account	Standard Object		
Activity	Activity	Standard Object		
Address	Address	Standard Object		
Alternative Payment Method	AlternativePaymentMethod	Standard Object		
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object		
AppLog	thsecurity_AppLog_c	Custom Object	Application logs	07/06/2024
Appointment	Appointment_c	Custom Object		27/07/2024
Appointment Category	AppointmentCategory	Standard Object		
Appointment Invitation	AppointmentInvitation	Standard Object		
Appointment Invitee	AppointmentInvitee	Standard Object		
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object		
Asset	Asset	Standard Object		

3. Upload the spreadsheet to Salesforce.

4.Map the fields from the spreadsheet to the object fields.

Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

Worksheet Details

Field Label Source: Detect from row

Field Labels Row: 1

Import 2 rows of Data? Yes, import data

Record Name Field: Let Salesforce Create a Default

IMPORT FILE FIELD NAME	SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
✓ Address	Address	Text	<input checked="" type="checkbox"/>	Hyderabad
✓ City	City	Text	<input checked="" type="checkbox"/>	Hyderabad
✓ Phone	Phone	Phone	<input checked="" type="checkbox"/>	1234567890
✓ Qualification	Qualification	Text	<input checked="" type="checkbox"/>	M.Sc
✓ University Name	University Name	Text	<input type="checkbox"/>	BHIT
✓ Year of Passing	Year of Passing	Date	<input type="checkbox"/>	2020

Fields 12 of 12 to import Hide mapped fields

Back Next

5.Define the Object properties

Create a custom object from a spreadsheet

Object properties

Almost finished! Time to define your object's attributes.

* Label: Student

* Plural Label: Student

* API Name: Student

Object Description:

> Advanced Settings

Back Finish

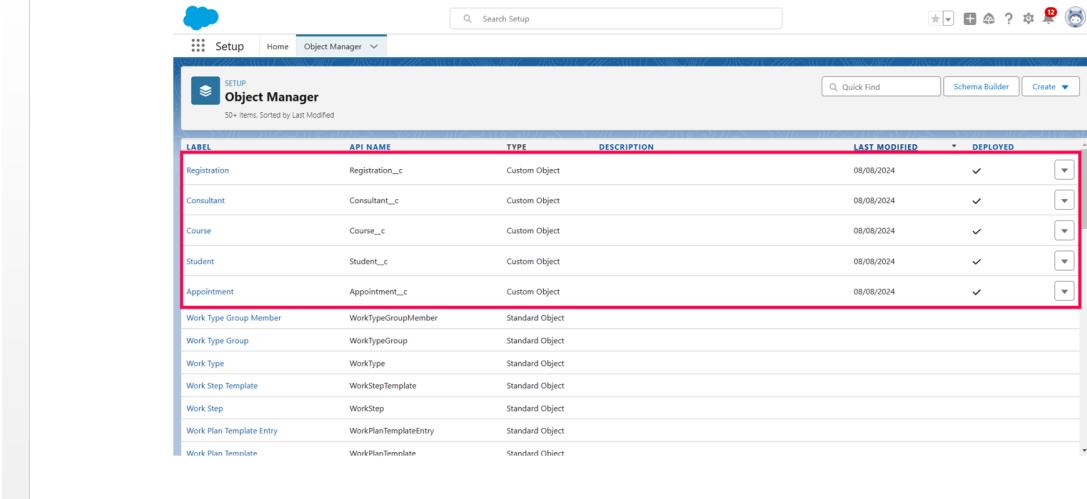
6.Following the same steps create the following objects:

a. Consultant

b. Student

c. Appointment

d. Registration

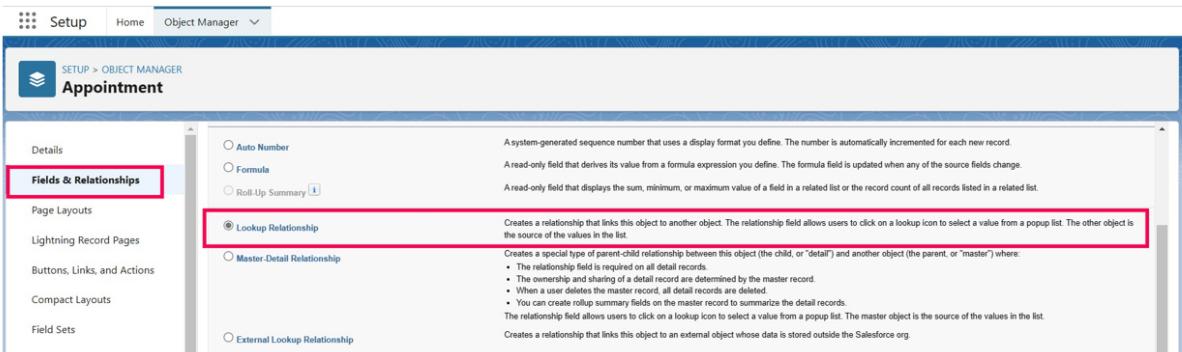


LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Registration	Registration__c	Custom Object		08/08/2024	✓
Consultant	Consultant__c	Custom Object		08/08/2024	✓
Course	Course__c	Custom Object		08/08/2024	✓
Student	Student__c	Custom Object		08/08/2024	✓
Appointment	Appointment__c	Custom Object		08/08/2024	✓
Work Type Group Member	WorkTypeGroupMember	Standard Object			
Work Type Group	WorkTypeGroup	Standard Object			
Work Type	WorkType	Standard Object			
Work Step Template	WorkStepTemplate	Standard Object			
Work Step	WorkStep	Standard Object			
Work Plan Template Entry	WorkPlanTemplateEntry	Standard Object			
Work Plan Template	WorkPlanTemplate	Standard Object			

Subtask 2: Create Relationships Among the Objects

1. Create a Lookup relationship between Appointment and Student Object

- a. Goto the Appointment object and create a new customfield from **Fields & Relationships** and choose the field type **Lookup relationship**



The screenshot shows the Salesforce Object Manager for the 'Appointment' object. The left sidebar lists various setup options like Details, Page Layouts, Lightning Record Pages, etc. The 'Fields & Relationships' option is selected and highlighted with a red box. In the main pane, there are four radio button options: 'Auto Number', 'Formula', 'Roll-Up Summary', and 'Lookup Relationship'. The 'Lookup Relationship' option is also highlighted with a red box. A detailed description of the 'Lookup Relationship' is provided below it.

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

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

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

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

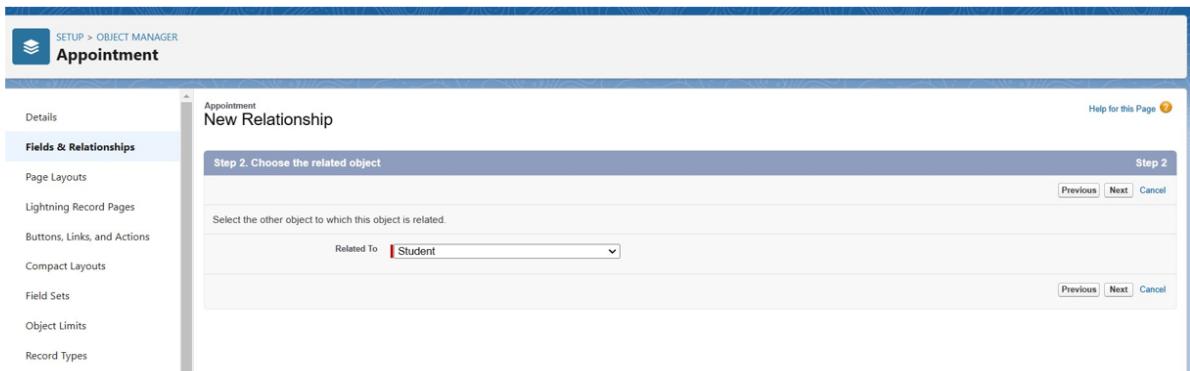
Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:

- The relationship field is required on all detail records.
- The ownership and sharing of a detail record are determined by the master record.
- When a user deletes the master record, all detail records are deleted.
- You can create rollup summary fields on the master record to summarize the detail records.

The relationship field allows users to click on a lookup icon to select a value from a popup list. The master object is the source of the values in the list.

Creates a relationship that links this object to an external object whose data is stored outside the Salesforce org.

- b. Choose the related object for the relationship as **Student**



The screenshot shows the 'New Relationship' wizard for the 'Appointment' object. The left sidebar includes 'Details', 'Fields & Relationships' (which is currently selected), 'Page Layouts', 'Lightning Record Pages', 'Buttons, Links, and Actions', 'Compact Layouts', 'Field Sets', 'Object Limits', and 'Record Types'. The main area is titled 'Step 2. Choose the related object' with a sub-instruction 'Select the other object to which this object is related.' Below this is a 'Related To' dropdown menu with 'Student' selected. Navigation buttons 'Previous', 'Next', and 'Cancel' are at the bottom right.

2. In the same way create **Lookup relationship** between the following:

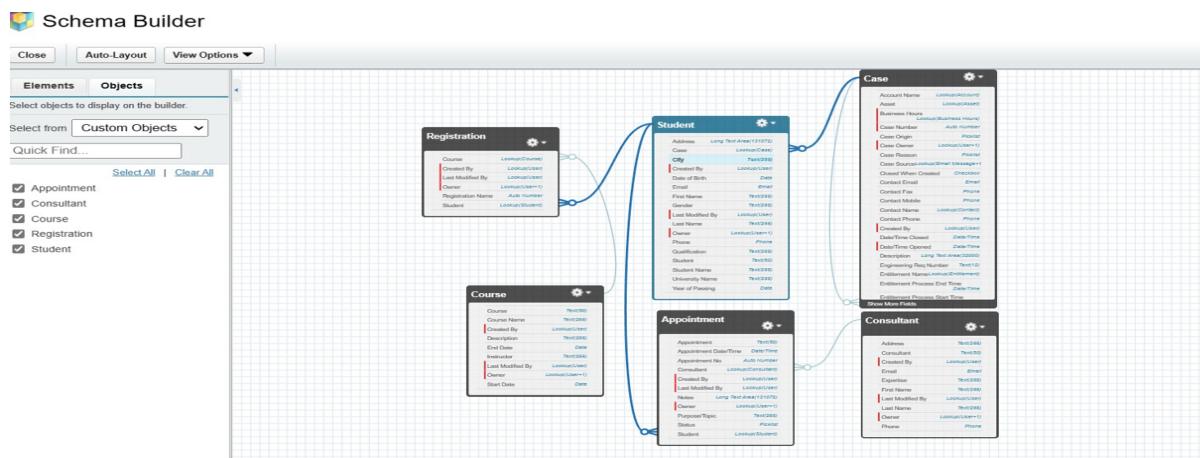
1. **Appointment and Student Object**
2. **Student and Case Object**

The Lookup relationship **Student** and **Case Object** is to manage student queries related to immigration or visa applications.

3. Go to the **Schema Builder** to see the relationship between the objects.

The screenshot shows the Salesforce Object Manager interface. At the top, there are tabs for Setup, Home, and Object Manager. A search bar says "Search Setup". On the right, there are various icons. Below the tabs, it says "SETUP Object Manager 51+ items. Sorted by Label". A "Schema Builder" button is highlighted with a red box. The main area is a table with columns: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The table lists standard objects like Account, Activity, Address, Alternative Payment Method, API Anomaly Event Store, and custom objects like Appointment. The "Appointment" object is selected, showing its details: API Name is Appointment_c, Type is Custom Object, and Last Modified is 08/08/2024.

4. The data model will look like this:



Subtask 3: Configure the Case Object

1. Navigate to **Object Manager**, then select the **Case** object.
2. Edit the **Type** field and add the following values:
 - a. **Immigration**
 - b. **Visa Application**

The screenshot shows the Salesforce Object Manager interface for the Case object. The left sidebar lists various configuration options like Details, Fields & Relationships, Case Page Layouts, etc. The main area displays the 'Fields & Relationships' section, which lists fields such as Product, Service Contract, SLA Violation, Status, Stopped, Stopped Since, Subject, and Type. The 'Type' field is highlighted with a red box. The Type field is defined as a Picklist with values: Web Company, Web Email, Web Name, and Web Phone. The Type field is also listed under the 'Fields & Relationships' section with a red box around it.

The screenshot shows the Salesforce Object Manager interface for the Case object. The left sidebar lists various configuration options like Details, Fields & Relationships, Case Page Layouts, etc. The main area displays the 'Field Dependencies' and 'Validation Rules' sections, both of which are empty. Below these, the 'Case Type Picklist Values' section is shown. This section lists four values: Mechanical, Electrical, Electronic, and Structural. A new value, 'Other', has been added and is highlighted with a red box. The 'Other' value is defined with the API name 'Other'. The 'Case Type Picklist Values' section is also highlighted with a red box.

3.Edit the **Status** field and add the following values:

c.Open

d.In-progress

The screenshot shows the Salesforce Setup interface for the Case object. In the left sidebar, under 'Fields & Relationships', 'Status' is selected. The main area displays the 'Fields & Relationships' table with the following data:

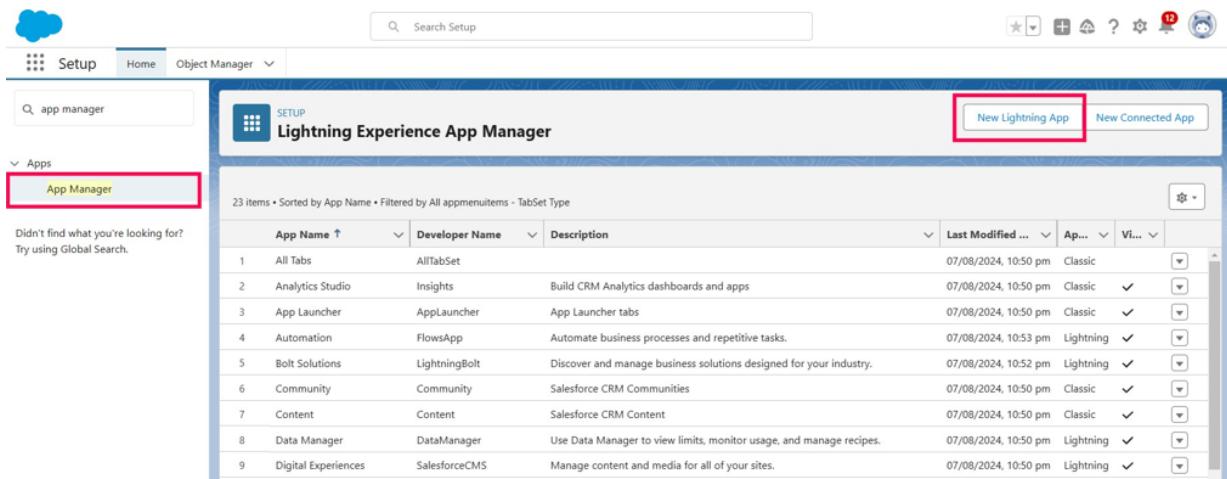
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Milestone Status	MilestoneStatus	Text(30)		
Status	Status	Picklist		

The screenshot shows the Salesforce Setup interface for the Case object. In the left sidebar, under 'Fields & Relationships', 'Status' is selected. The main area displays the 'Picklist Values Used' section with the following data:

Action	Values	API Name	Closed	Default	Chart Colors	Modified By
Edit Deactivate	New	New	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Assigned dynamically	Dianwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Working	Working	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Dianwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Escalated	Escalated	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Dianwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Closed	Closed	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Dianwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Open	Open	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Dianwita Singha, 08/08/2024, 1:32 am
Edit Del Deactivate	In-progress	In-progress	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Dianwita Singha, 08/08/2024, 1:32 am

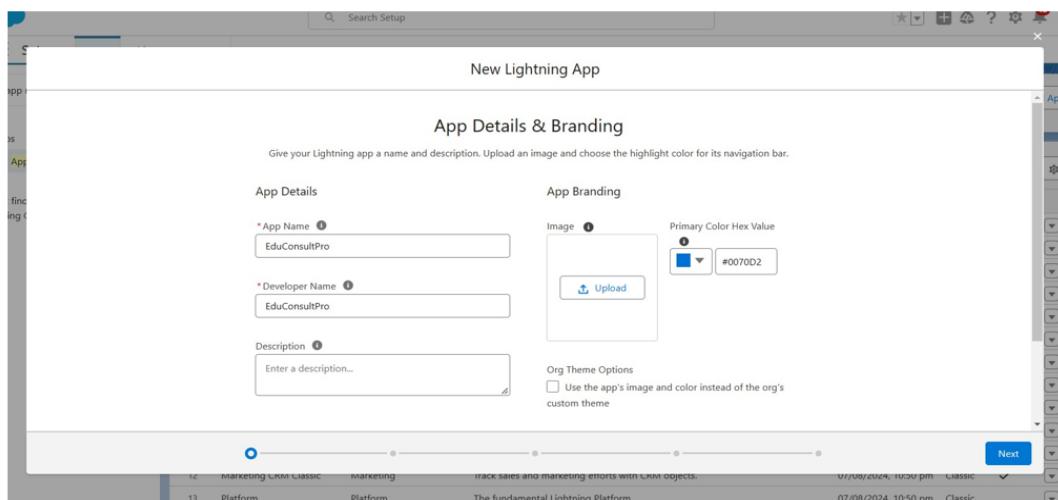
Subtask 4: Create a Lightning App

- 1.**In Setup, search for **App Manager** in the Quick Find bar.
- 2.**Click on **New Lightning App**.



The screenshot shows the Salesforce Setup interface. In the top left, there's a cloud icon followed by 'Setup', 'Home', and 'Object Manager'. A search bar says 'Search Setup'. On the left, a sidebar has a 'Q app manager' search field and a 'v Apps' section with 'App Manager' highlighted with a red box. Below it, a message says 'Didn't find what you're looking for? Try using Global Search.' The main area is titled 'Lightning Experience App Manager' and shows a table of 23 items. The table has columns for 'App Name', 'Developer Name', 'Description', 'Last Modified ...', 'Ap...', and 'Vi...'. The first item is 'All Tabs' with developer 'AllTabSet'. The last item is 'Digital Experiences' with developer 'SalesforceCMS'. At the bottom right of the table is a 'New Lightning App' button, which is also highlighted with a red box.

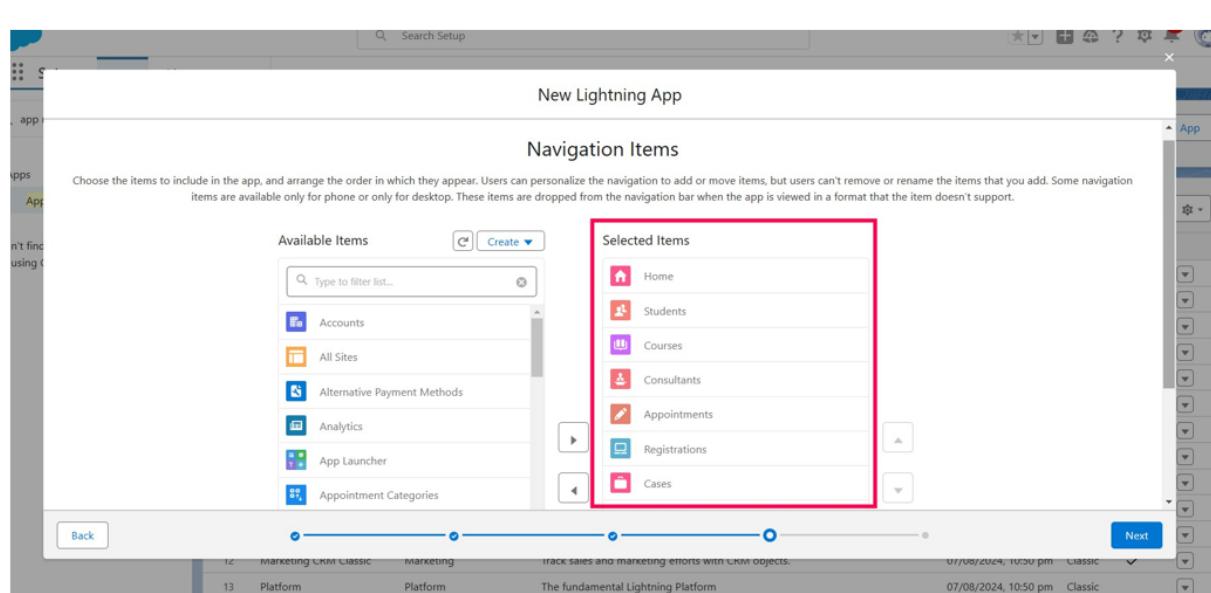
- 3.**Name the app **EduConsultPro**.



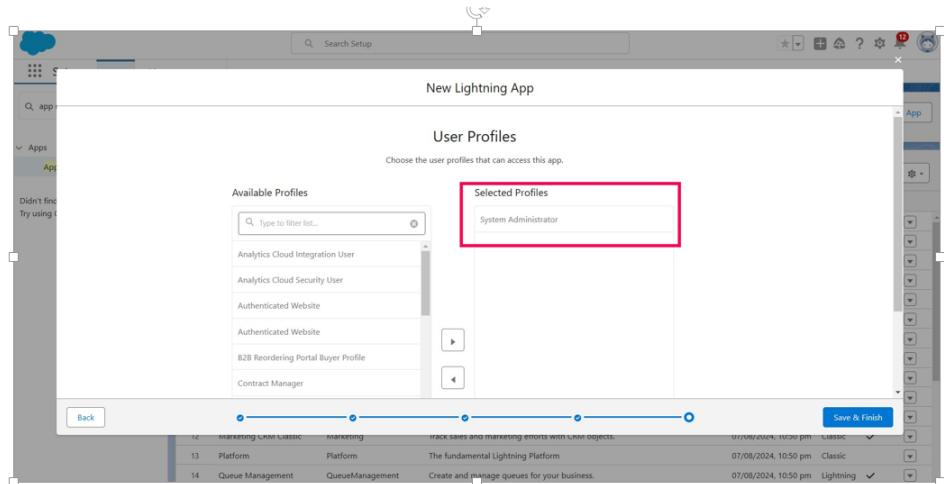
The screenshot shows the 'New Lightning App' configuration screen. At the top, it says 'New Lightning App'. Below that is a section titled 'App Details & Branding' with the sub-instruction 'Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.' Under 'App Details', there are three fields: 'App Name' (set to 'EduConsultPro'), 'Developer Name' (set to 'EduConsultPro'), and 'Description' (with placeholder 'Enter a description...'). Under 'App Branding', there are two sections: 'Image' (with a placeholder 'Upload') and 'Primary Color Hex Value' (set to '#0070D2'). There's also an option 'Org Theme Options' with a checkbox 'Use the app's image and color instead of the org's custom theme'. At the bottom, there's a progress bar with steps 12 through 15, and a 'Next' button.

4.Add the following items from the Available Items list to the SelectedItems list:

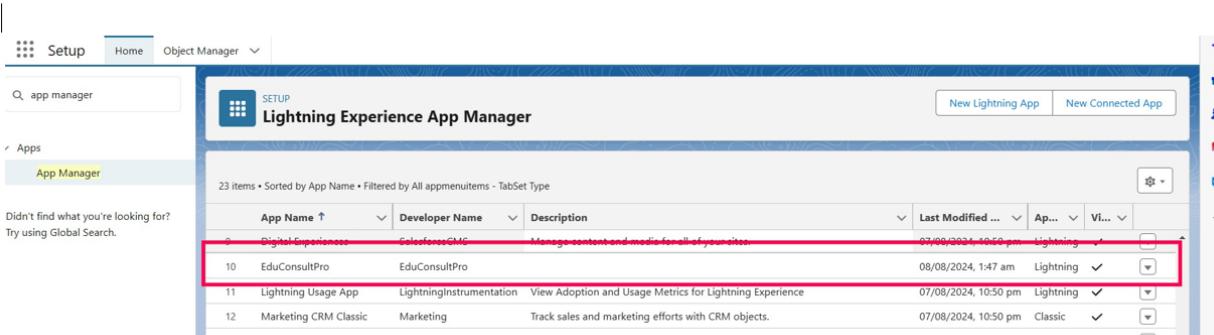
a. . **Home, Students, Courses, Consultants, Appointments, Registrations, Cases**



5.Add the **System Administrator** profile from Available Profilesto Selected Profilesand click **Save & Finish**.



6. EduConsultPro app is created.



The screenshot shows the Salesforce Setup interface with the "App Manager" selected. The main title is "Lightning Experience App Manager". A search bar at the top left contains "app manager". Below it, a sidebar lists "Apps" and "App Manager". A message says " Didn't find what you're looking for? Try using Global Search." The main content area displays a table with the following data:

App Name	Developer Name	Description	Last Modified	Type
Digital Experience	SalesforceCMC	Manage components for the digital experience platform.	07/08/2024, 10:50 pm	Lightning
10 EduConsultPro	EduConsultPro		08/08/2024, 1:47 am	Lightning
11 Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	07/08/2024, 10:50 pm	Lightning
12 Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	07/08/2024, 10:50 pm	Classic

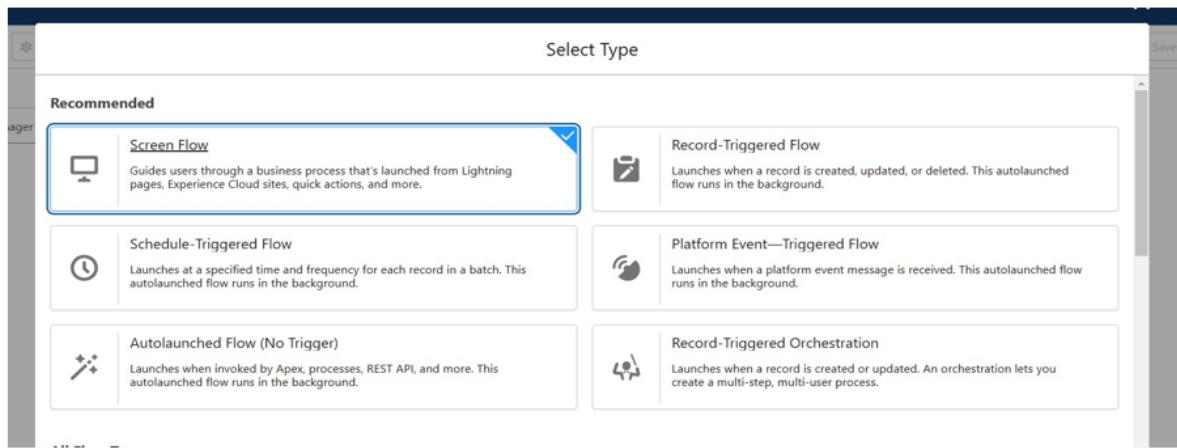
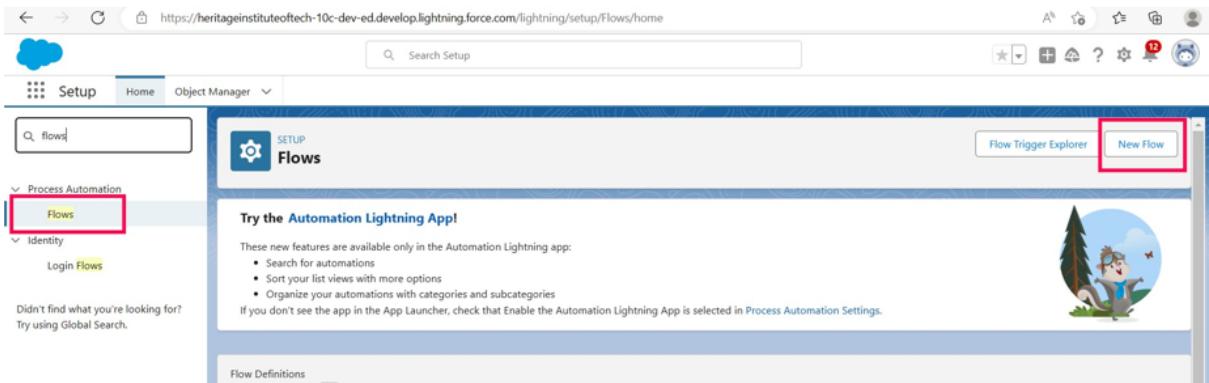
TASK 4: CreateA ScreenFlow For Student Admission Application Process

Design a ScreenFlow to automate the student admission process, including data capture, record creation, course selection, and email notifications.

Subtask 1: Add Screen Element

Set up a screen to capture student information.

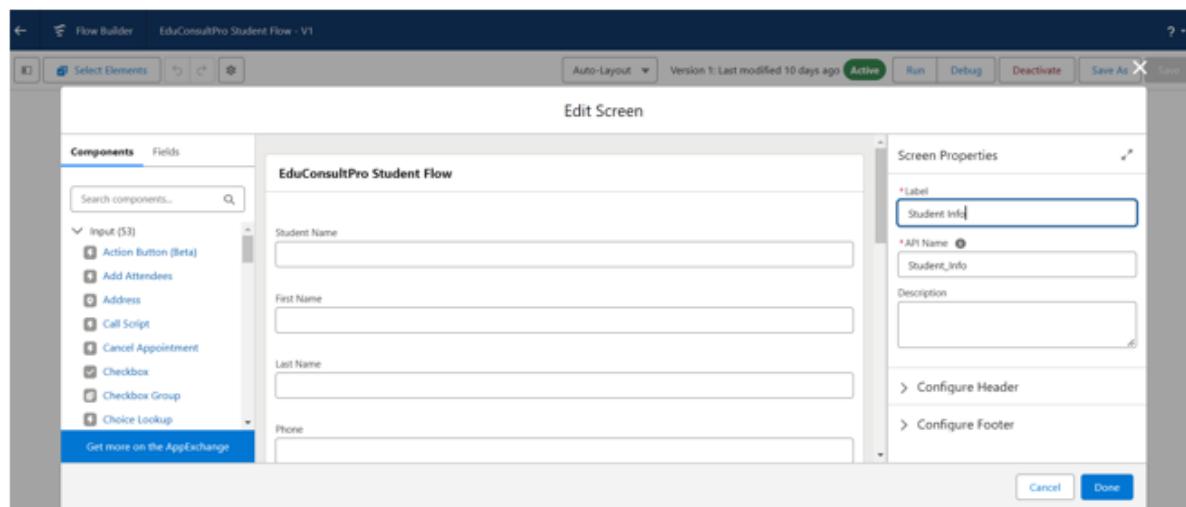
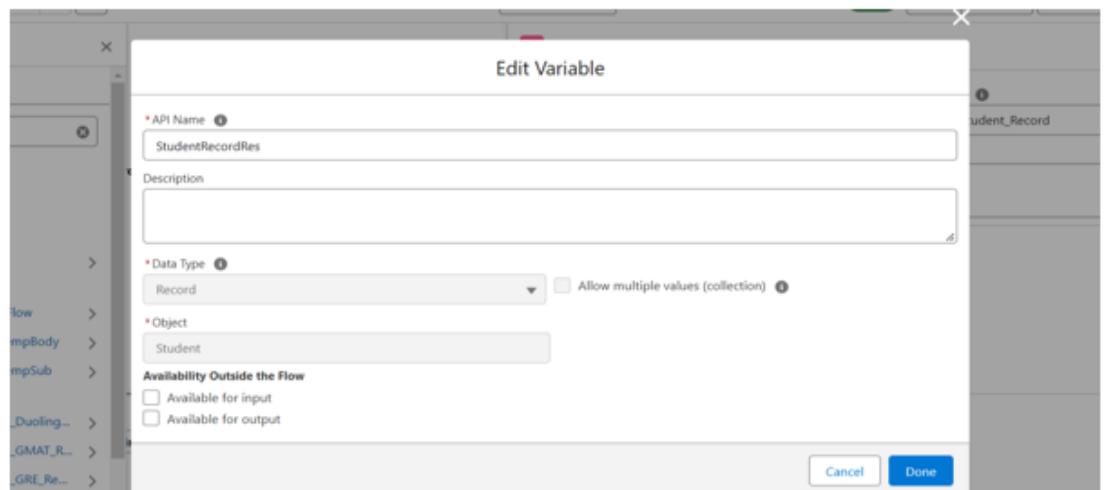
1. Navigate to **Setup**, enter **Flow** in the Quick Find box, and select **New Flow**. Choose **ScreenFlow**.



2.Add a **Screen** element.

3.In the **Screen Properties** pane:

- a. **Label:** Enter “Student Info”.
- b. **Fields:** Create a new **Resource**(StudentRecordRes) to display all fields from the **Student** object.
- c. **Action:** Drag and drop the necessary fields to collect student information on the screen like the Student Name, First Name, Last Name, Phone, Email, DoB, Address, City, Qualification.



Subtask 2: Create StudentRecord Using CreateElement

Set up a process to create a student record from the captured information.

1. Add a **Create** element after the Student Info screen element.
2. **Label:** “Create Student Record”.
3. Select “**One**” under **How many records to Create**.
4. Choose “**Use all values from a record**” under **How to Set the recordfields**.
5. Select the **StudentRecordRes** variable from the Student Info screen element under **Create a record from these values**.

The screenshot shows the configuration for a 'Create Records' element. It includes fields for Label (Create Student Record), API Name (Create_Student_Record), Description (empty), How to set record field values (From a Record Variable), How Many Records to Create (One selected), and Create a Record from These Values (StudentRecordRes selected). A note at the bottom states: 'Make sure that ID is blank. After the flow creates the records, ID is set to match the record that was created.' A small icon with a question mark is present next to the note.

Create Records

*Label: Create Student Record

*API Name: Create_Student_Record

Description:

*How to set record field values: From a Record Variable

How Many Records to Create:
 One
 Multiple

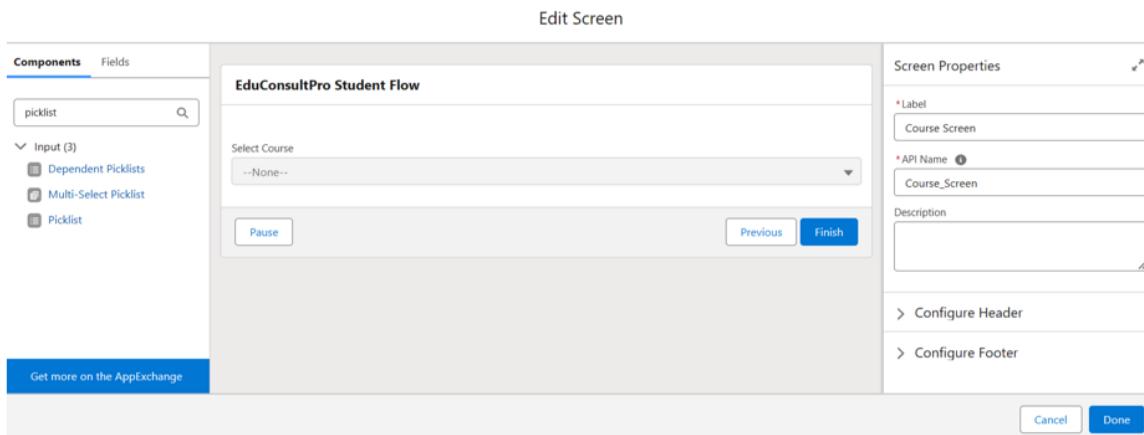
Create a Record from These Values:
* Record: StudentRecordRes

Make sure that ID is blank. After the flow creates the records, ID is set to match the record that was created. ⓘ

Subtask 3: Add CourseSelection Screen

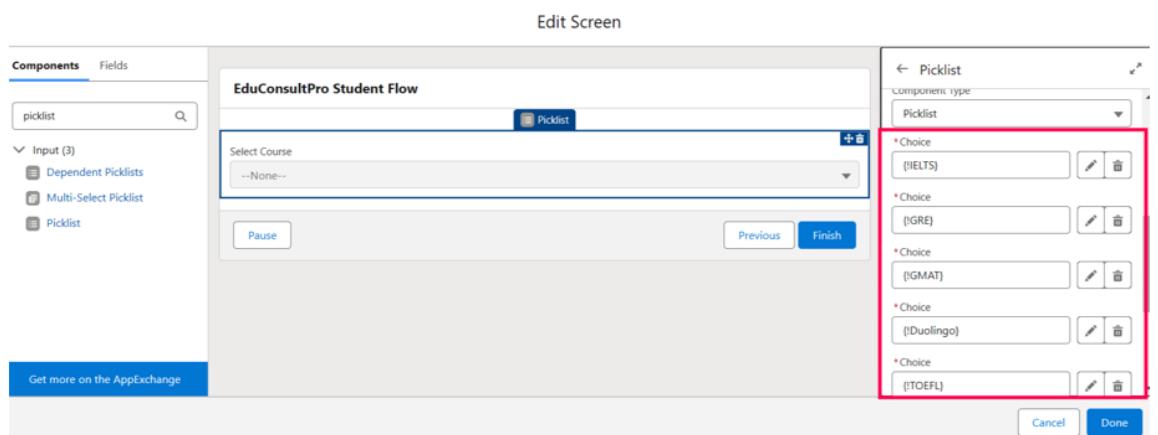
Provide a screen for students to select a course.

1. Add a **Screen** element after the Create Student Record element.
2. **Label:** “Course Screen”.



3.Add a **Picklist** component:

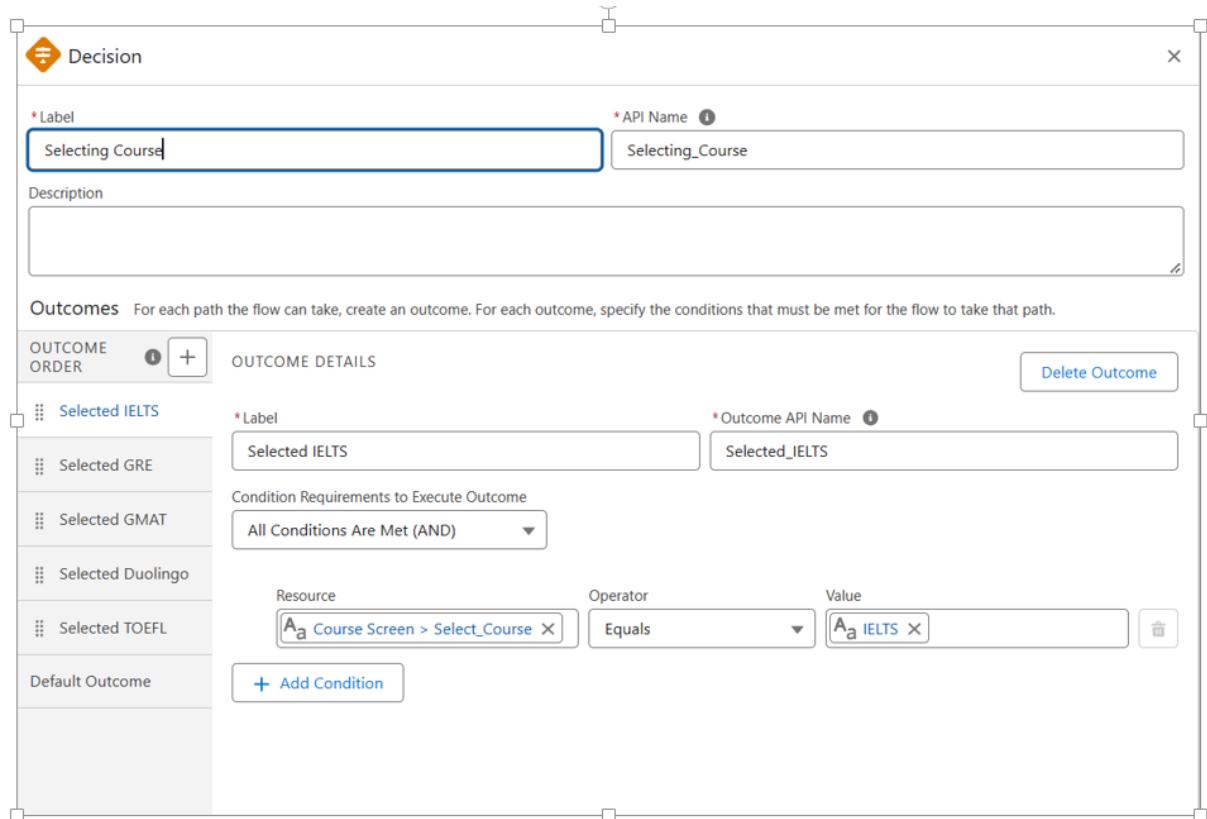
- a. **Label:** “Select Course”.
- b. **Choices:** Enter “IELTS”, “GRE”, “GMAT”, “Duolingo”, and “TOEFL”.



Subtask 4: Add DecisionElement

Define logic to determine which course has been selected.

1. Add a **Decision** element after the Course Screen.
2. **Label:** “Selecting Course”.
3. Define outcomes for each course:
 - a. **Outcome Label:** “Selected IELTS”.
 - b. **Condition:**
 - i. **Resource:** Select_Course (Screen Component from Select Course Screen Element)
 - ii. **Operator:** Equals
 - iii. **Value:** IELTS (Choice Variable from SelectCourse Screen Element)
 - c. Repeat for GRE, GMAT, Duolingo, TOEFL.



Subtask 5: Add GET Record Element

Retrieve course details based on the selected course.

1. Add a **GET Record** element after each Decision path (IELTS, GRE, GMAT, etc.).
2. **Label:** “Get [Course Name] Rec”.
3. **Object:** Course
4. **Condition Requirement:** All Conditions are Met (AND)
5. **Field:** Course Name
6. **Operator:** Equals
7. **Value:** {!Select_Course}

The screenshot shows the configuration interface for a 'Get Records' element. The top section includes fields for 'Label' (containing 'Get IELTS Rec'), 'API Name' (containing 'GetIELTS_Rec'), and a 'Description' area. Below this, under 'Get Records of This Object', the 'Object' is set to 'Course'. In the 'Filter Course Records' section, the 'Condition Requirements' dropdown is set to 'All Conditions Are Met (AND)'. A single condition is defined with 'Field' set to 'Name', 'Operator' set to 'Equals', and 'Value' set to '{!Select_Course}'. There is also a '+ Add Condition' button. The bottom section, 'Sort Course Records', has a 'Sort Order' field which is currently empty.

Subtask 6: Create Registration Record Using Create Records Element

Create a registration record for the selected course.

1. Add a **Create** element after each GET Record element.
2. **Label:** “Create [Course Name] Registration Rec”.
3. **How many records to Create:** “One”.
4. **How to Set the record fields:** Use separate resources, and literal values.
5. **Object:** Registration
6. Set fields:
 - a. **Course_Name__c:** {!Get_[Course Name]_Rec.Id}
 - b. **Student_Name__c:** {!StudentRecordRes.Id}

The screenshot shows the 'Create Records' configuration screen. At the top, there are fields for 'Label' (Create IELTS Registration Req) and 'API Name' (CreateIELTSRegistrationRec). Below these are sections for 'Description' (empty), 'How to set record field values' (Manually), and 'Create a Record of This Object' (Object: Registration). The final section, 'Set Field Values for the Registration', contains two rows of fields and values. The first row has 'Field' (Course_c) and 'Value' (Aa Course from GetIELTSRec > Record ID). The second row has 'Field' (Student_c) and 'Value' (Aa StudentRecordRes > Record ID). A '+ Add Field' button is at the bottom left of this section.

Subtask 7: Create Email Text Template Variables

Set up email templates for student registration confirmation.

1. Click the **toggle toolbox** on the left corner, select **New Resource**, and choose **Text Template**.
2. **API Name:** “StuRegistrationEmailTextTempBody”.
3. **Type:** View as plain text.
4. **Body:** Write a registration confirmation text.

Edit Text Template

The screenshot shows the 'Edit Text Template' dialog box. At the top, there's a field labeled 'API Name' with the value 'StuRegistrationEmailTextTempBody'. Below it is a 'Description' field which is empty. The main area is labeled 'Body' and contains a rich text editor. Inside the editor, the text reads:
Dear {!StudentRecordRes.Name},

Congratulations and welcome to EduConsultantPro!

We are delighted to inform you that your registration on our platform has been successfully completed. You are now part of our esteemed community dedicated to empowering students like you to achieve their educational and immigration aspirations.
At EduConsultantPro, we understand the importance of your academic and career goals, and we are committed to providing you with the highest level of support and guidance throughout your journey.

Cancel Done

5. Repeat to create an email text template for the subject,
API Name: "StuRegistrationEmailTextTempSub".

Edit Text Template

* API Name !
StuRegistrationEmailTextTempSub

Description

* Body !

Insert a resource... View as Plain Text ▾

Dear {!StudentRecordRes.Name},

Congratulations and welcome to EduConsultantPro!

We are delighted to inform you that your registration on our platform has been successfully completed. You are now part of our esteemed community dedicated to empowering students like you to achieve their educational and immigration aspirations.

Cancel Done

Subtask 8: Add ActionElement

Configure the action to send an email to the student.

1. Add an **Action** element after all Decision paths.
2. **Label:** “Send Email to Student”.
3. **Input Values:**
 - a. **Body:** {!StuRegistrationEmailTextTempBody}
 - b. **Recipient Address List:** {!StudentRecordRes.Email__c}
 - c. **Subject:** {!StuRegistrationEmailTextTempSub}

The screenshot shows two side-by-side configurations for a 'Send Email' action. The left panel is the main configuration screen, and the right panel is a detailed view of the 'Send Email' settings.

Main Configuration (Left Panel):

- Label:** Send Email to Student
- API Name:** Send_Email_to_Student
- Description:** (empty)
- Selected Action:** Send Email (emailSimple-emailSimple)
- Note:** Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variables.
- Set Input Values for the Selected Action:**
 - Add Threading Token to Body: Not Included
 - Add Threading Token to Subject: Not Included
 - Body: StuRegistrationEmailTextTempBody (Included)

Settings (Right Panel):

- A_a Recipient Address Collection: Not Included
- A_a Recipient Address List:
 - A_a StudentRecordRes > Email (Included)
- A_a Recipient ID: Not Included
- A_a Related Record ID: Not Included
- Rich-Text-Formatted Body: Not Included
- A_a Sender Email Address: Not Included
- A_a Sender Type: Not Included
- A_a Subject:
 - StuRegistrationEmailTextTempSub (Included)

Subtask 9: Add Success Screen

Display a confirmation message to the student after the registration.

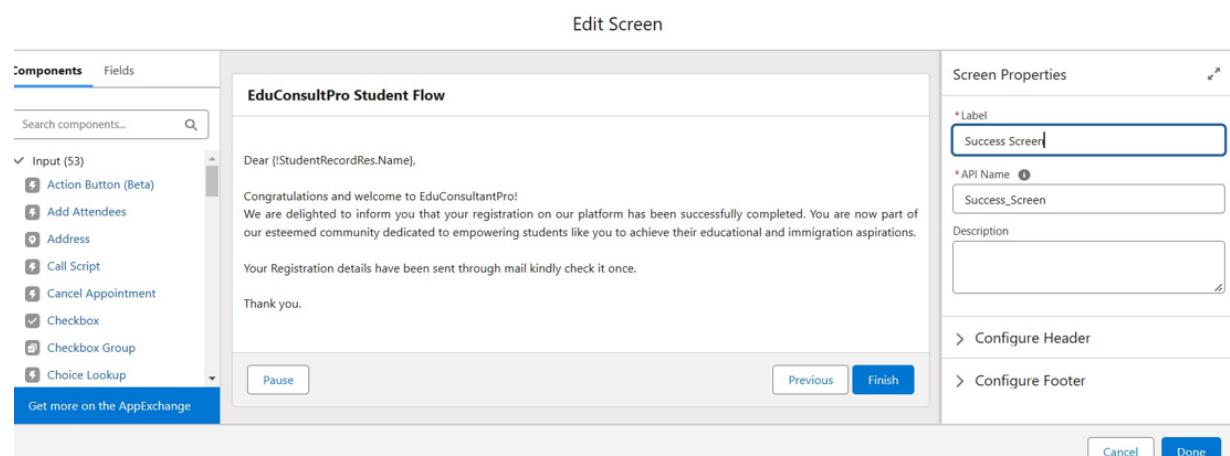
1.Add a **Screen** element after the Send Email toStudent Action Element.

2.Label: “Success Screen”.

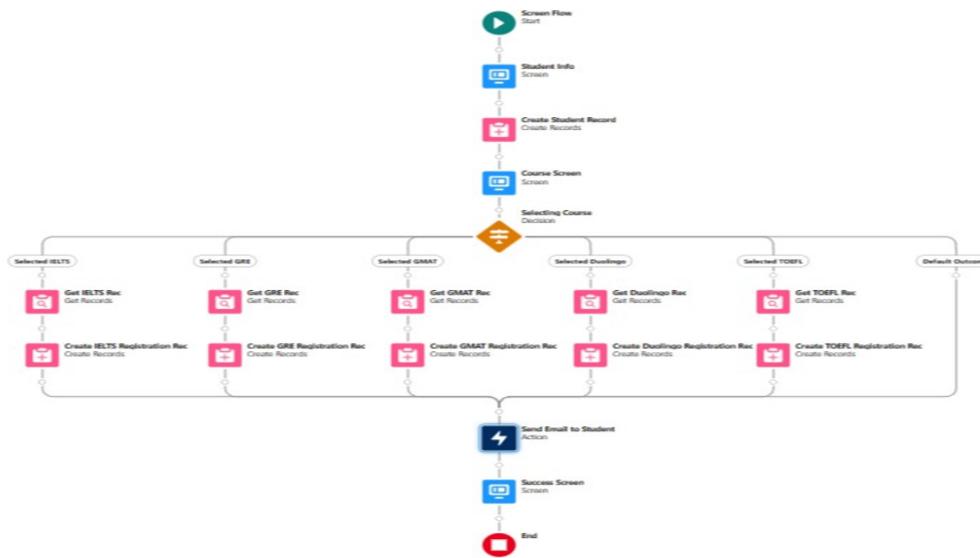
3.Add a **Display Text** component:

a. **Label:** “SuccessMessage”.

b. **Text:** Write the success message.



4.Save the flow andname it “**EduConsultPro Student Flow**”.



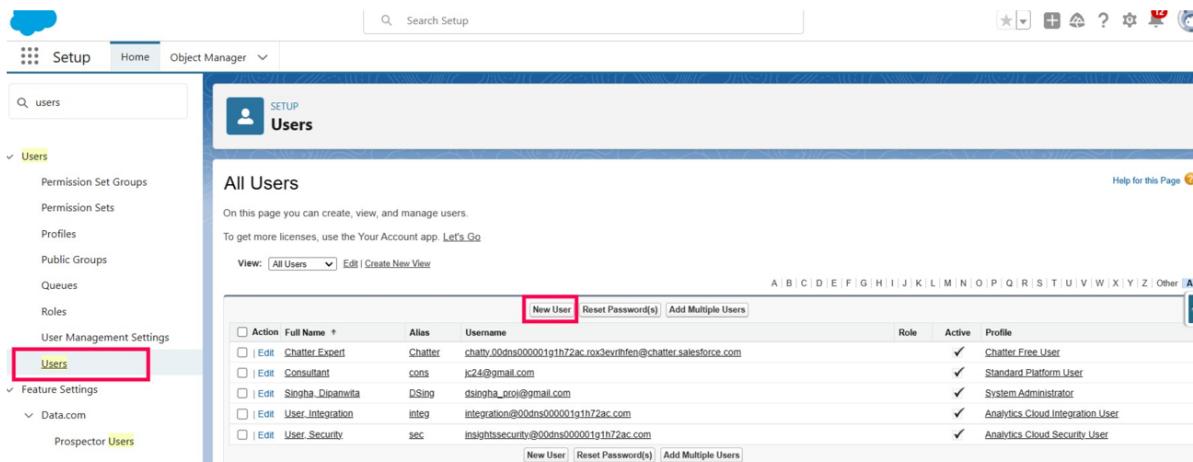
Task 5: Create Users

Set up a new user with the Standard Platform User profile and configure the user settings.

Subtask 1: Create a User

Create a new user with the Standard Platform User profile.

1. Navigate to **Setup > Users > New User**.



The screenshot shows the Salesforce Setup interface. The left sidebar is expanded to show the 'Users' section, which is highlighted with a red box. At the top of the main content area, there is a 'New User' button, also highlighted with a red box. The main area displays a list of existing users with columns for Action, Full Name, Alias, Username, Role, Active status, and Profile. The 'Profile' column shows entries like 'Chatter Free User', 'Standard Platform User', 'System Administrator', 'Analytics Cloud Integration User', and 'Analytics Cloud Security User'. The 'Active' column contains checkmarks for most users.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chatter Expert	Chatter	chatty_00ns000001g1h72ac_rox3evrlfen@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/>	Consultant	cons	k24@gmail.com		✓	Standard Platform User
<input type="checkbox"/>	Singha_Deanwita	DSing	dsingha_croi@gmail.com		✓	System Administrator
<input type="checkbox"/>	User_Integration	Integ	integration@00ns000001g1h72ac.com		✓	Analytics Cloud Integration User
<input type="checkbox"/>	User_Security	sec	Insightssecurity@00dns000001g1h72ac.com		✓	Analytics Cloud Security User

2. Enter the following details:

- a. **Last Name:** Consultant
- b. **License:** Salesforce Platform
- c. **Profile:** Standard PlatformUser

3. Complete all mandatory fields as required and save.

The screenshot shows the Salesforce Setup interface under the 'Users' section. A user named 'Consultant' is being edited. The 'Last Name' field is highlighted with a red box. The 'Role' dropdown, 'User License' dropdown, and 'Profile' dropdown are also highlighted with red boxes. The 'Active' checkbox is checked.

Subtask 2: Configure the User Settings

Update the user settings to include the appropriate approver settings.

1. Navigate to **Setup > Administration > Users** > Locate the newly created user and click **Edit** next to their name.
2. Scroll down to the **ApproverSettings** section.
3. In the **Manager Field**, select “Consultant” and save.

The screenshot shows the 'Approver Settings' section of the user edit screen. The 'Manager' field is highlighted with a red box and contains the value 'Consultant'.

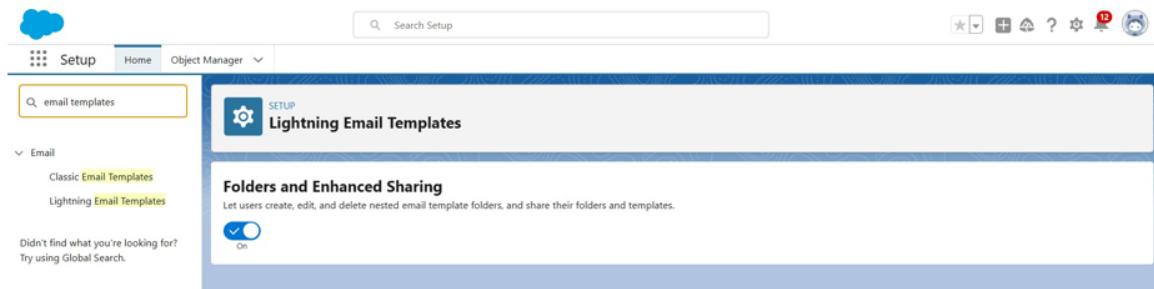
Task 6: Create an ApprovalProcess for the Property Object

Establish an approvalprocess for the Property object,including creating necessaryemail templates for notifications.

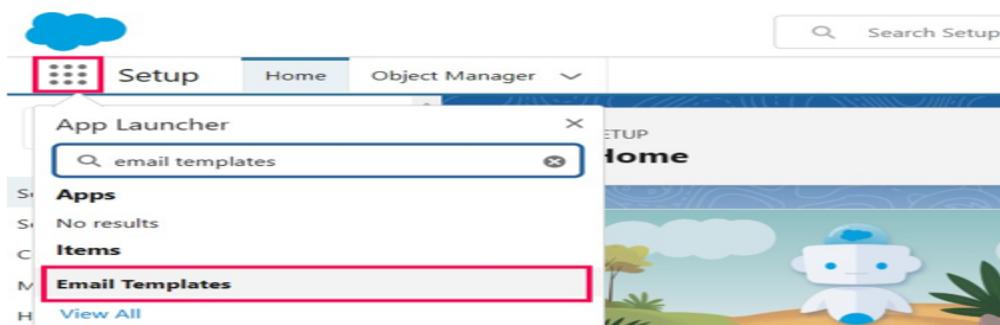
Subtask 1: Create Email Templates

Create and configure email templates to be used in the approval process.

- 1.Navigate to **Setup**,enter **Templates** in the Quick Find box, and select **Lightning Email Templates**. Toggle on.



- 2.Go to the **App Launcher**, search for **Email Templates**, and create a new folder with the desired name.



3.Create a new email template within the created folder:

- a. **Template Name:** Submission Template
- b. Write a message in the HTML Value

Edit Submission Template

* = Required Information

Information

Email Template Name <input type="text" value="Submission Template"/>	Related Entity Type <input type="text" value="-- None --"/>
Description <input type="text"/>	Folder <input type="text" value="EduConsultantpro_Email"/> <input type="button" value="Select Folder"/>

Message Content

Subject <input type="text" value="Appointment Request with EduConsultantpro Consulta"/>	Enhanced Letterhead <input type="text" value="Search Enhanced Letterheads..."/> <input type="button" value="Q"/>
---	--

HTML Value

Dear {{Appointment_c.Student_Name_c}},
I hope this email finds you well. I am writing to confirm the details of our upcoming appointment scheduled for {{Appointment_c.Appointment_DateTime_c}} regarding {{Appointment_c.PurposeTopic_c}}.
Appointment Details:
Appointment No : {{Appointment_c.Name}},
Student Name : {{Appointment_c.Student_Name_c}},
Consultant Name : {{Appointment_c.Consultant_c}},
Date & Time : {{Appointment_c.Appointment_DateTime_c}},
Purpose : {{Appointment_c.PurposeTopic_c}}

I want to assure you that I am looking forward to our meeting and am fully prepared to address any questions or concerns you may have regarding {{Appointment_c.PurposeTopic_c}}. Your success and satisfaction are my top priorities, and I am committed to providing you with the guidance and support you need.

If you have any specific topics or questions you would like to discuss during our appointment, please feel free to share them with me in advance. This will help ensure that our time together is as productive and beneficial as possible.

If for any reason you need to reschedule or cancel our appointment, please notify me at your earliest convenience so that we can make alternative arrangements.

Once again, thank you for choosing to work with me on this matter. I am confident that our collaboration will

4.Create two additional email templates:

- a.**Approval Template:**Similar to the Submission Templatebut tailored for approval notifications.
- b.**Rejection Template:**Similar to the Submission Templatebut tailored for rejection notifications

Edit Approval Request Template

* = Required Information

Information

* Email Template Name

Approval Request Template

Related Entity Type

-- None --

Description

Folder

EduConsultantpro_Email

Select Folder

Edit Rejection Request Template

* = Required Information

Information

* Email Template Name

Rejection Request Template

Related Entity Type

-- None --

Description

Folder

EduConsultantpro_Email

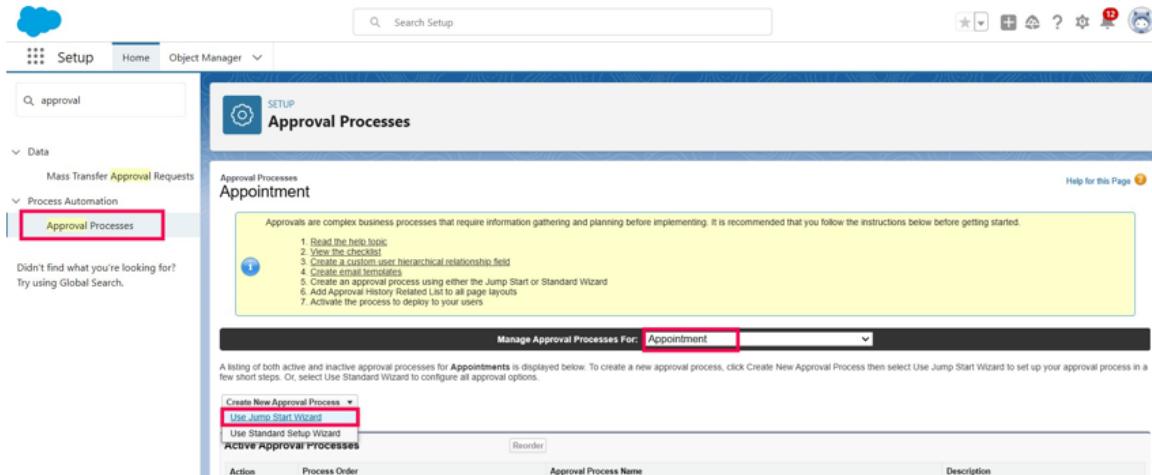
Select Folder

Email Templates						
Recent						
3 items						
EMAIL TEMPLATES	Email Template Name	Description	Folder	Last Modified By	Last Modified Date	
Recent	Approval Request Template		EduConsultantpro_Email	Dipanwita Singha	8/8/2024, 2:22 pm	<input type="button" value=""/>
Created by Me	Submission Template		EduConsultantpro_Email	Dipanwita Singha	8/8/2024, 2:09 pm	<input type="button" value=""/>
Private Email Templates	Rejection Request Template		EduConsultantpro_Email	Dipanwita Singha	8/8/2024, 2:21 pm	<input type="button" value=""/>
Public Email Templates						

Subtask 2: Create An Approval Process

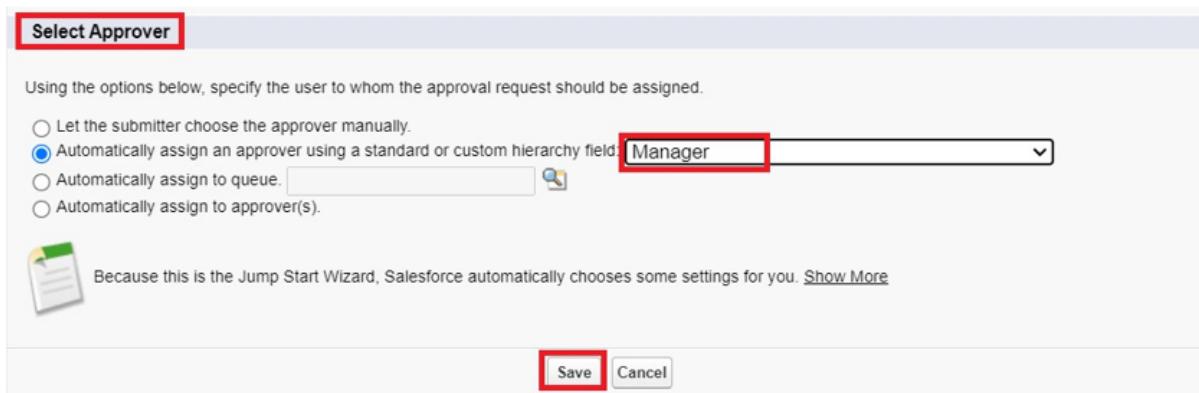
Configure an approval process for the Appointment object to manage request approvals.

1. Navigate to **Setup**, enter **Approval** in the Quick Find box, and select **Approval Processes**.
2. In **Manage Approval Processes For**, select **Appointment**.
3. Click **Create New Approval Process** and choose **Use Jump Start Wizard**.



4. Configure the approval process:

- a. **Process Name:** Appointment Approval
- b. **Approver Selection:** Select **Manager** for “Automatically assign an approver using a standard or custom hierarchy field.”



Select Approver

Using the options below, specify the user to whom the approval request should be assigned.

Let the submitter choose the approver manually.
 Automatically assign an approver using a standard or custom hierarchy field Manager
 Automatically assign to queue.
 Automatically assign to approver(s).

Because this is the Jump Start Wizard, Salesforce automatically chooses some settings for you. [Show More](#)

Save Cancel

- 5.Click **Next** and select **Manager**for the option **AutomatedApprover Determined By**.
- 6.Under **Record Editability Properties**, select **Administrators OR**the currently assigned approver can edit records during the approval process.
- 7.Click **Save**.

Approval Process Edit
Appointment Approval

Step 3. Specify Approver Field and Record Editability Properties

When you define approval steps, you can assign approval requests to different users. One of your options is to use a user approval steps, select a field from the picklist below. Also, when a record is in the approval process, it will always be locked by the currently assigned approver to edit the record.

Select Field Used for Automated Approval Routing

Next Automated Approver Determined By ▼ i

Use Approver Field of Appointment Owner

Record Editability Properties

Administrators **ONLY** can edit records during the approval process.
 Administrators **OR** the currently assigned approver can edit records during the approval process.

- 8.Click **View ApprovalProcess Detail Page**.

9.Under **Initial Submission Actions**, click **Add New**:

c. Select **Field Update**:

- i. **Field to Update:** Appointment: Status
- ii. **Value:** Pending

The screenshot shows the 'Field Update Edit' screen. In the 'Identification' section, 'Name' is set to 'Submitted' and 'Unique Name' is also 'Submitted'. Under 'Object', it is 'Appointment'. 'Field to Update' is 'Appointment: Status', 'Field Data Type' is 'Picklist', and 'Re-evaluate Workflow Rules after Field Change' is checked. In the 'Specify New Field Value' section, 'Picklist Options' show 'A specific value' selected as 'Pending'. At the bottom, there are 'Save', 'Save & New', and 'Cancel' buttons.

10.Click **Add New**:

d. Select **Email Alert**:

- i. **Description:** Submission Email Alert
- ii. **Email Template:** Submission Template
- iii. **Recipient Type:** Select your Name

The screenshot shows the 'Email Alert Edit' screen. In the 'Edit Email Alert' section, 'Description' is 'Submission Email Alert', 'Unique Name' is 'Submission_Email_Alert', 'Object' is 'Appointment', and 'Email Template' is 'Submission Template'. Under 'Recipients', 'Recipient Type' is 'User', and the search results show 'User: Consultant', 'User: Integration User', and 'User: Security User'. A user named 'Dipanwita Singha' is selected and listed under 'Selected Recipients'. At the bottom, there are 'Save', 'Save & New', and 'Cancel' buttons.

11. Repeat steps 9 and 10 for **Final Approval** and **Final Rejection** actions using the appropriate templates and values.

Final Approval Actions 		Add Existing Add New ▾
Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit Remove	Email Alert	Approval Email Alert
Edit Remove	Field Update	Approved

Final Rejection Actions 		Add Existing Add New ▾
Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Field Update	Rejected
Edit Remove	Email Alert	Rejection Email Alert

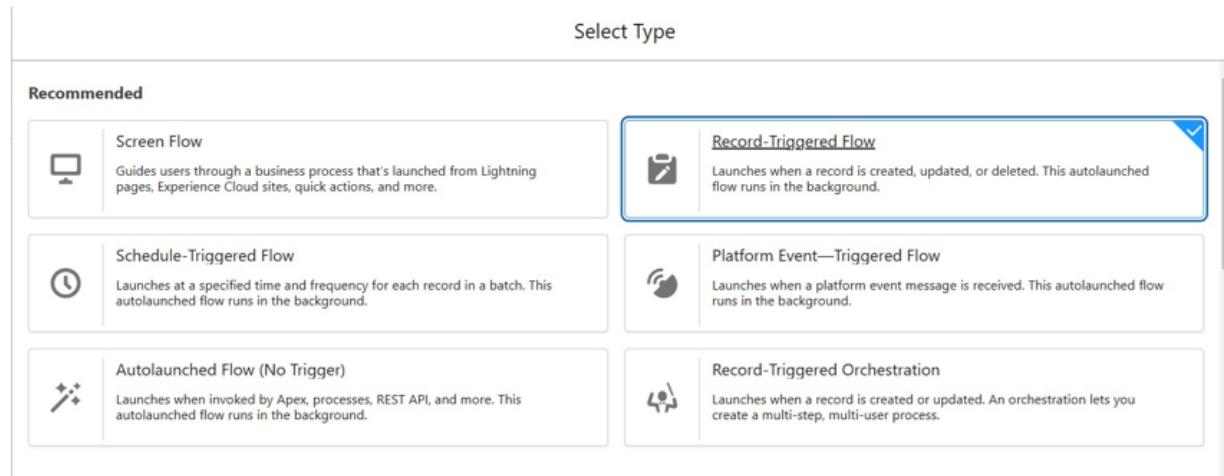
Task 7: Create A Record-Triggered Flow

Set up a Record-Triggered Flow to automate the submission of Appointment records for approval.

Subtask 1: Configure The Start Element

Initiate a flow that triggers upon the creation of an Appointment record.

- 1.From **Setup**, enter **Flows** in the Quick Find box, then select **Flows**.
- 2.Click **New Flow**.
- 3.Select **Record-Triggered Flow**.



- 4.Click **Create**. The **ConfigureStart** window opens.
- 5.For **Object**, select **Appointment**.
- 6.For **Trigger the Flow When**, select **A record is created**.

Select Object

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

* Object

Appointment

Configure Trigger

* Trigger the Flow When:

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

Set Entry Conditions

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

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

Condition Requirements

None

Subtask 2: Add An Action Element

Add an Action element to submit the Appointment record for approval automatically.

1. Add an **Action** element after the **Start** element.
2. Select the **Submit for approval** action and label it as **Approval SubFlow**.
3. Set the **RecordId** to **{!\$Record.Id}**.

Submit for Approval

* Label
Approval SubFlow

* API Name ⓘ
Approval_SubFlow

Description

Submit for Approval ⓘ
submit-submit

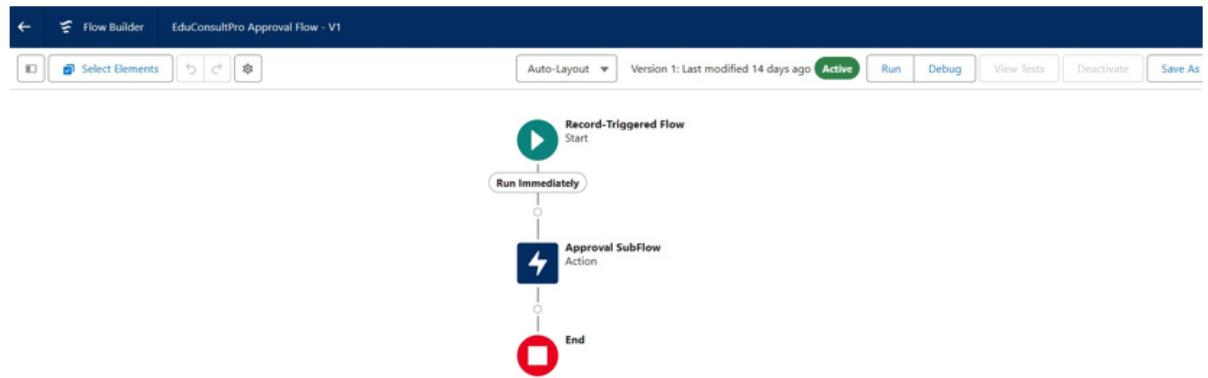
Use values from earlier in the flow to set the inputs for the "Submit for Approval" core action. To use its outputs later in the flow, store them in variables.

Set Input Values for the Selected Action

Aa * Record ID ⓘ
Aa Triggering Appointment__c > Record ID X

Aa Approval Process Name Or ID Not Included

4. Save the Flow, label it as **EduConsultPro ApprovalFlow**, and click **Activate**.

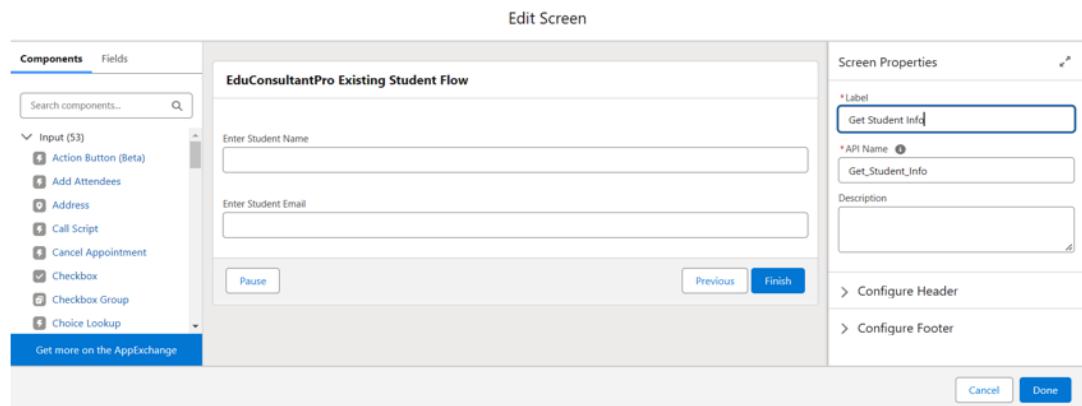


Task 8: Create A ScreenFlow For Existing Students To Book An Appointment

Subtask 1: Add Screen Element

Start the flow by gathering student information using a Screen element.

- 1.From **Setup**, enter **Flow Builder** in Quick Find and select **New Flow** → **ScreenFlow**.
- 2.Add a **Screen** element.
- 3.In the **Screen Properties** pane, for **Label**,enter “**Get Student Info**”.
- 4.Add two **Text** components from the left-side panel.
 - a. 1st Text Component Label:**Enter Student Name**
 - b. 2nd Text Component Label:**Enter Student Email**



- 5.Click **Done**.

Subtask 2: Get Record

1. Add a **GET Record** element after the Screen element, label it as “**Get Rec**”.
2. Configure the GET Record element:
 - a. **Select Object:** Student
 - b. **Condition Requirement:** All Conditions are Met (AND)
 - c. **Field:** Student Name
 - i. **Operator:** Equals
 - ii. **Value:** {!Enter_Student_Name}
 - d. **Field:** Email__c
 - i. **Operator:** Equals
 - ii. **Value:** {!Enter_Student_Email}

Get Records

* Label: Get Rec * API Name: Get_Rec

Description

Get Records of This Object

* Object: Student

Filter Student Records

Condition Requirements: All Conditions Are Met (AND)

Field	Operator	Value	
Student_Name__c	Equals	Aa Enter_Student_Name	
AND	Email__c	Equals	Aa Enter_Student_Email

+ Add Condition

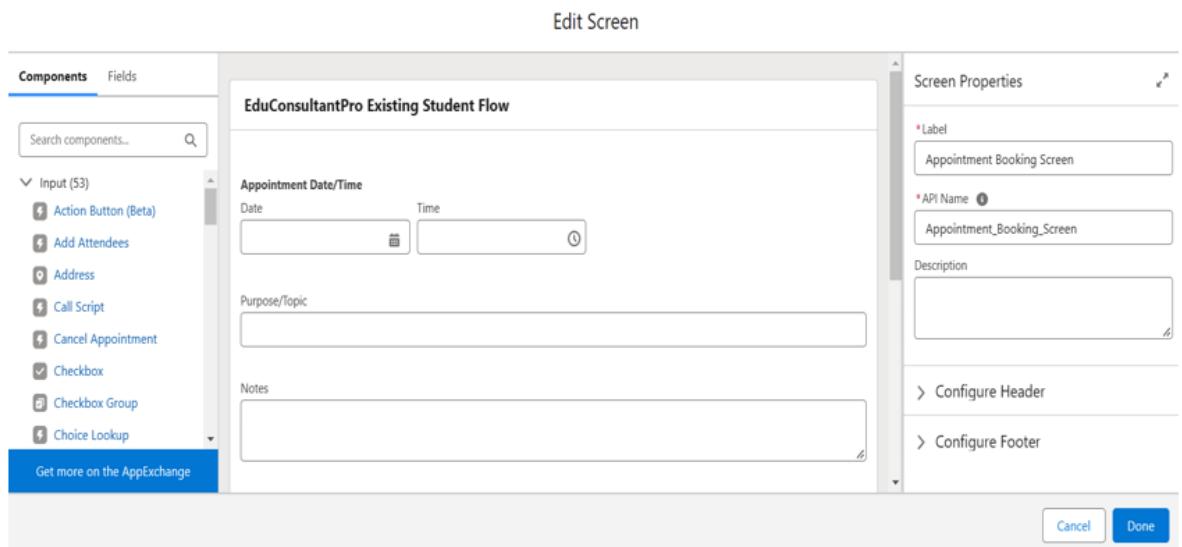
Subtask 3: Add a Decision Element

1. Add a **Decision** element after the GET Record element, label it as “Appointment or Case”.
2. Configure the Decision element:
 - a. **Outcome Label:** Appointment
 - i. **Resource:** {!How_may_I_Help_you}
 - ii. **Operator:** Equals
 - iii. **Value:** {!Book_an_Appointment}
 - b. Click on the “+” icon to add paths for other options such as Case, Default.

The screenshot shows the 'Decision' configuration screen. At the top, there are fields for 'Label' (Appointment or Case) and 'API Name' (Appointment_or_Case). Below this is a 'Description' section with a large text input field. The main area is titled 'Outcomes' with the sub-instruction: 'For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.' There are two outcomes listed: 'Appointment' and 'Case'. Each outcome has a 'Label' field (Appointment and Case respectively) and an 'Outcome API Name' field (Appointment and Appointment). Under 'Default Outcome', the condition requirement is set to 'All Conditions Are Met (AND)'. Below this, there is a row for defining a condition: 'Resource' (A_a How_may_I_Help_you), 'Operator' (Equals), and 'Value' (A_a Book_an_Appointment). A 'Delete Outcome' button is located at the top right of the outcome list.

Subtask 4: Add Screen Element

1. Add a **Screen** element after the Decision element, on the Appointment path, and label it as “**Appointment Booking Screen**”.
2. Click on **Fields**, then **Record Variable Input**, and create a new Resource (AppointmentRecordRes) to display all the fields in the Appointment object.
3. Drag the necessary fields onto the screen to collect student information.
4. Click **Done**.



Subtask 5: Add GET Record Element

1. Add a **GET Record** element after the Decisionelement, under the Appointment path, and label it as “**Get Consultant Rec**”.

2. Configure the GET Record element:

- a. **Select Object:** Consultant
- b. **Condition Requirement:** All Conditions are Met (AND)
- c. **Field:** Name
 - i. **Operator:** Equals
 - ii. **Value:** {!AppointmentRecordRes.Consultant_Name__c}

The screenshot shows the configuration of a 'Get Records' element. At the top, there's a header with a 'Get Records' icon and the title 'Get Records'. Below the header, there are two main sections: 'Get Records of This Object' and 'Filter Consultant Records'.

Get Records of This Object:

- * Label: Get Consultant Rec
- * API Name: Get_Consultant_Rec
- Description: (empty)
- * Object: Consultant

Filter Consultant Records:

- Condition Requirements: All Conditions Are Met (AND)
- Filter Criteria:
 - Field: Name
 - Operator: Equals
 - Value: Aa AppointmentRecordRes > Ap... (with a delete icon)
- + Add Condition

Subtask 6: Create Appointment Record Using CreateRecords Element

1. Add a **Create** element after the GET Consultant Rec element and label it as “**Create Appointment**”.

2. Configure the Create element:

a. **Select Object:** Appointment

b. **Field Values:**

i. **Appointment_DateTime__c:**

{!AppointmentRecordRes.Appointment_DateTime__c}

ii. **Consultant__c:** {!Get_Consultant_Rec.Id}

iii. **Notes__c:** {!AppointmentRecordRes.Notes__c}

iv. **PurposeTopic__c:** {!AppointmentRecordRes.PurposeTopic__c}

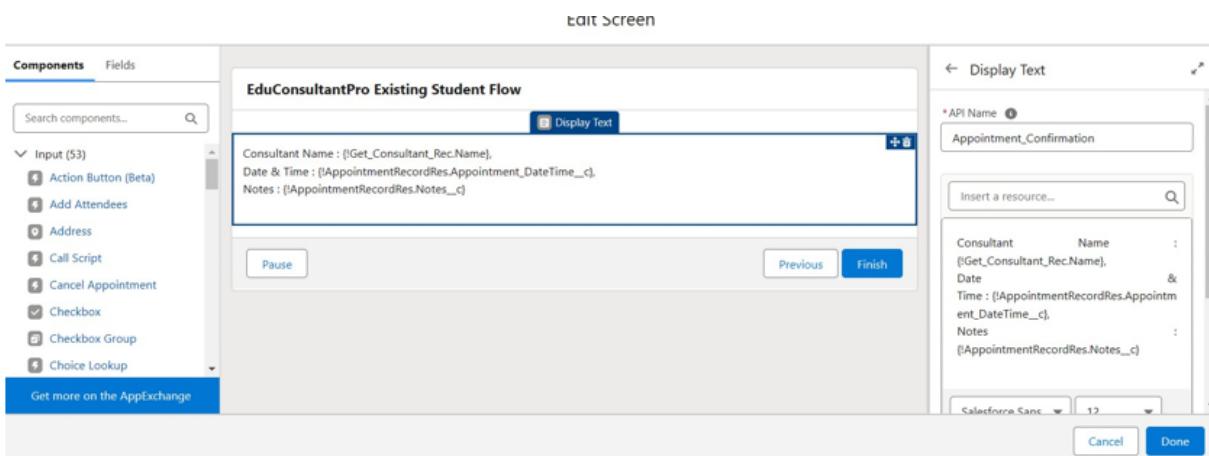
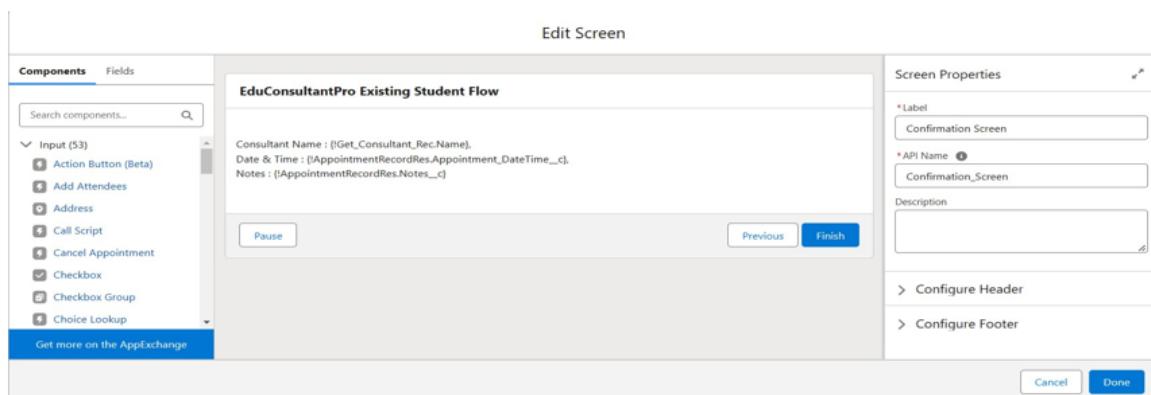
v. **Student_Name__c:** {!Get_Rec.Id}

The screenshot shows the configuration of a 'Create Records' element. It includes fields for Label ('Create Appointment'), API Name ('Create_Appointment'), Description, and How to set record field values ('Manually'). Below this, there's a section to 'Create a Record of This Object' with 'Object' set to 'Appointment'. The final section is 'Set Field Values for the Appointment', where three fields are mapped to their respective values: 'Appointment_DateTime__c' maps to 'AppointmentRecordRes > Appointment Date...', 'Consultant__c' maps to 'Consultant from Get_Consultant_Rec > Record ...', and 'Notes__c' maps to 'AppointmentRecordRes > Notes'.

Subtask 7: Add Screen Element

1. Add a **Screen** element after the Create Appointment element, and label it as “**Confirmation Screen**”.
2. From the left side panel, search for the **Display Text** component and drag it to the main panel, label it as “**Appointment_Confirmation**”.
3. Paste the following text in the Resource picker box: Consultant Name :
 {!Get_Consultant_Rec.Name},

 Date & Time : {!AppointmentRecordRes.Appointment_DateTime_c}, Notes : {!AppointmentRecordRes.Notes_c}
4. Click Done.

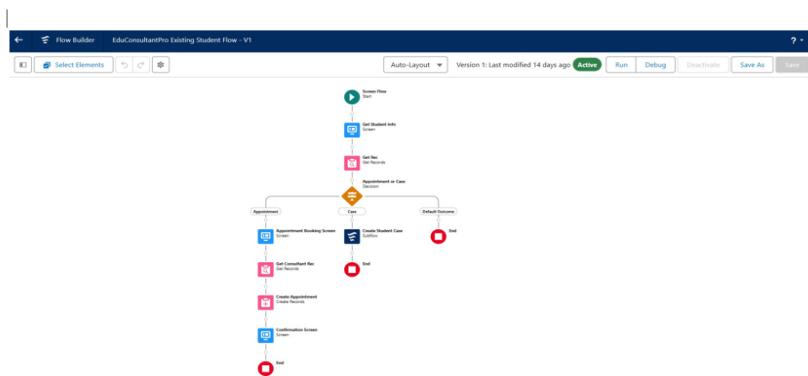


Subtask 8: Add a Subflow Element

1. Add a **Subflow** element after the Decision element, on the Case path, and search and select “**Create a Case**”, label it as “**Create Student Case**”.
2. Configure the Subflow element as needed.



3. Save the flow, label it as “**EduConsultantPro Existing Student Flow**”, and activate it.



Task 9: Create A ScreenFlow To Combine All The Flows At One Place

This task involves creating a central ScreenFlow that integrates various existing flows, allowing users to choose and access different functionalities from a single interface.

Subtask 1: Add WelcomeScreen Element

- 1.From Setup, enter **Flow Builder** in the Quick Find box, select **New Flow** → **Screen Flow**.
- 2.Add a **Screen** element and label it as “**Welcome Screen**”.
- 3.From the left side panel, drag the **Display Text** component to the main panel.
- 4.Label the Display Text component as “**SuccessMessage**”.
- 5.Paste the following text into the Resource picker box:

“Welcome to EduConsultantPro

your premier destination for education and immigration solutions!

At EduConsultantPro, we understand that embarking on educational or immigration journeys can be both exhilarating and daunting. That's why we're here to guide you every step of the way with expertise, dedication, and personalized support.

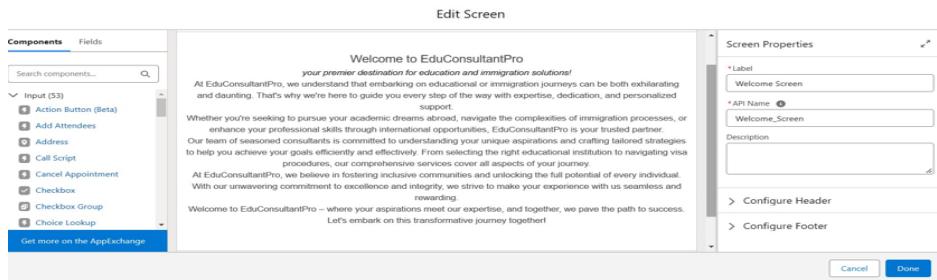
Whether you're seeking to pursue your academic dreams abroad, navigate the complexities of immigration processes, or enhance your professional skills through international opportunities, EduConsultantPro is your trusted partner.

Our team of seasoned consultants is committed to understanding your unique aspirations and crafting tailored strategies to help you achieve your goals efficiently and effectively. From selecting the right educational institution to navigating visa procedures, our comprehensive services cover all aspects of your journey.

At EduConsultantPro, we believe in fostering inclusive communities and unlocking the full potential of every individual. With our unwavering commitment to excellence and integrity, we strive to make your experience with us seamless and rewarding.

Welcome to EduConsultantPro – where your aspirations meet our expertise, and together, we pave the path to success. Let's embark on this transformative journey together!”

6.Click Done.



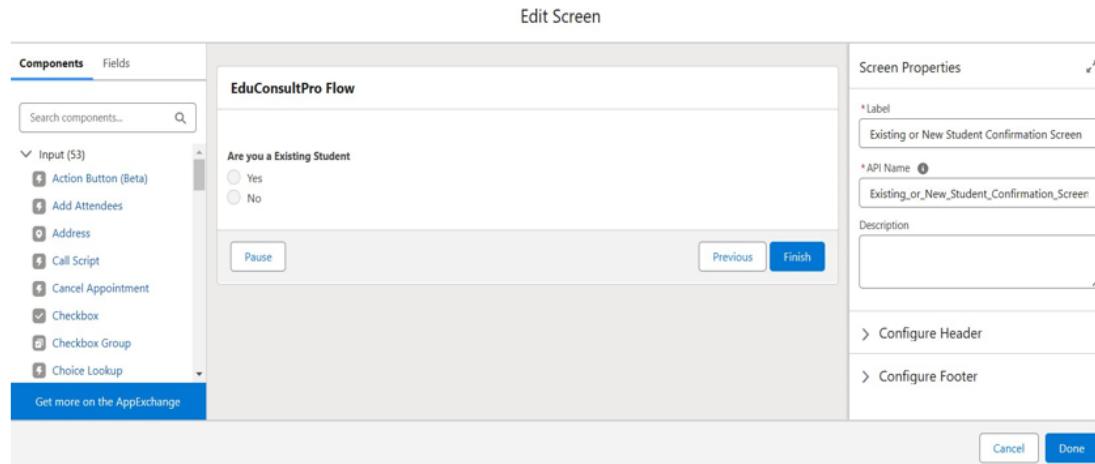
Subtask 2: Add Existing or New Student Confirmation Screen

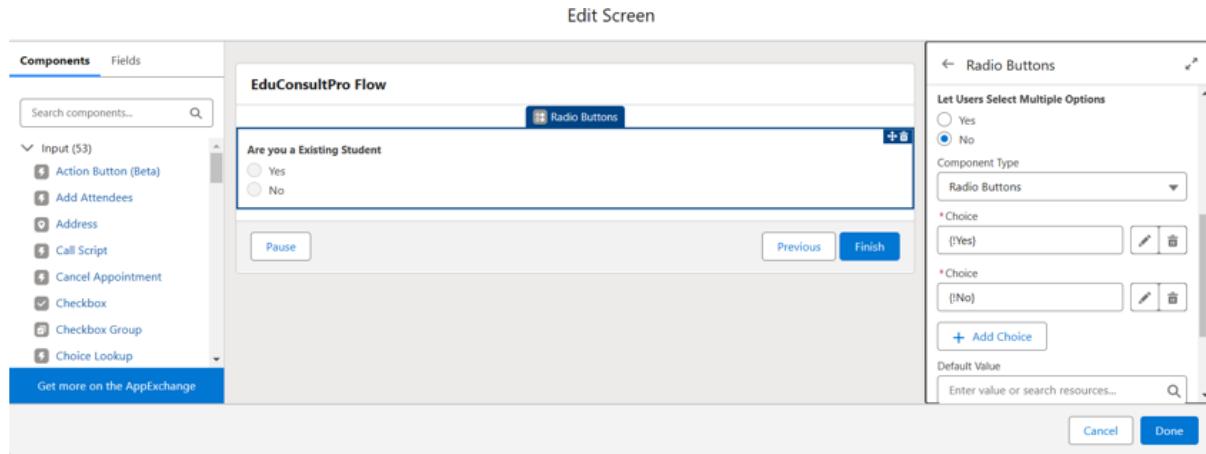
1.Add a **Screen** element after the Welcome Screen element and label it as “**Existing or New Student Confirmation Screen**”.

2.Add a **Radio Button** component from the left side panel.

- Label:** Are you an Existing Student?
- Choice 1:** “Yes”
- Choice 2:** “No”

3.Click **Done**.





Subtask 3: Add DecisionElement

1. Add a **Decision** element after the Existing or New Student Confirmation Screen element and label it as “**Decision 1**”.
2. Create an outcome:
 - a. **Label:** If Existing Student
 - b. **Resource:** {!Are_you_a_Existing_Student}
 - c. **Operator:** Equals
 - d. **Value:** {!Yes}
3. Click the “+” icon to add more outcomes for “No” and other cases as required.

Decision

* Label * API Name

Description

Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome
<ul style="list-style-type: none"> <input checked="" type="checkbox"/> If Existing Student <input type="checkbox"/> If not a existing user 	<p>* Label <input type="text" value="If Existing Student"/> * Outcome API Name <input type="text" value="If_Existing_Student"/></p> <p>Condition Requirements to Execute Outcome <input type="button" value="All Conditions Are Met (AND)"/></p> <p>Resource <input type="text" value="Aa ... > Are_you_a_Existing_Student X"/> Operator <input type="text" value="Equals"/> Value <input type="text" value="Aa Yes X"/> <input type="button" value="Delete"/></p> <p><input type="button" value="+ Add Condition"/></p>	<input type="button" value="Delete Outcome"/>
Default Outcome		

OUTCOME ORDER

OUTCOME DETAILS

OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome
<ul style="list-style-type: none"> <input checked="" type="checkbox"/> If Existing Student <input type="checkbox"/> If not a existing user 	<p>* Label <input type="text" value="If not a existing user"/> * Outcome API Name <input type="text" value="If_Not_a_Existing_Student"/></p> <p>Condition Requirements to Execute Outcome <input type="button" value="All Conditions Are Met (AND)"/></p> <p>Resource <input type="text" value="Aa ... > Are_you_a_Existing_Student X"/> Operator <input type="text" value="Equals"/> Value <input type="text" value="Aa No X"/> <input type="button" value="Delete"/></p> <p><input type="button" value="+ Add Condition"/></p>	<input type="button" value="Delete Outcome"/>
Default Outcome		

Subtask 4: Add Subflow for Existing Students

1. Add a **Subflow** element after the **Decision 1** element on the **If Existing Student** path.
2. Search for and select “**EduConsultantPro ExistingStudent Flow**”.
3. Label it as “**Existing Student Flow**”.
4. Click **Done**.

The screenshot shows the configuration dialog for a subflow named "EduConsultantPro Existing Student Flow". The dialog includes fields for "Label" (set to "Existing Student Flow"), "API Name" (set to "Existing_Student_Flow"), and a "Description" field which is empty. Below the dialog, a section titled "Referenced Flow" displays the subflow's configuration with its name and API name. A descriptive note explains how values from the parent flow can be used to set inputs for the subflow, mentioning both automatic storage of all outputs and manual assignment of individual outputs via API names.

EduConsultantPro Existing Student Flow

Existing Student Flow

Existing_Student_Flow

Use values from the parent flow to set the inputs for the "EduConsultantPro Existing Student Flow" flow. By default, the parent flow stores all outputs. You can either reference outputs via the API name of the Subflow element or manually assign variables in the parent flow to store individual outputs from the "EduConsultantPro Existing Student Flow" flow.

Subtask 5: Add Subflow for New Students

1. Add a **Subflow** element after the **Decision 1** element on the **If Not an Existing Student** path.
2. Search for and select “**EduConsultantPro Student Flow**”.
3. Label it as “**New Student Flow**”.
4. Click **Done**.

EduConsultPro Student Flow

* Label
New Student Flow

* API Name i
New_Student_Flow

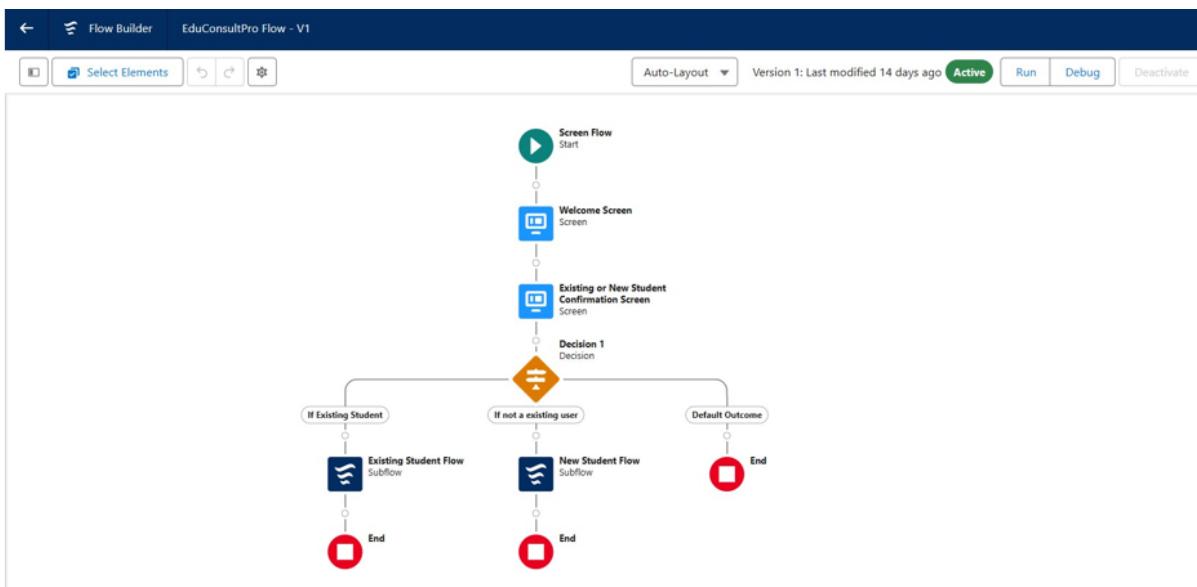
Description

Referenced Flow

EduConsultPro Student Flow
EduConsultPro_Student_Flow

Use values from the parent flow to set the inputs for the "EduConsultPro Student Flow" flow. By default, the parent flow stores all outputs. You can either reference outputs via the API name of the Subflow element or manually assign variables in the parent flow to store individual outputs from the "EduConsultPro Student Flow" flow.

5. Save the flow, label it as “**EduConsultantPro Flow**”, and activate it.



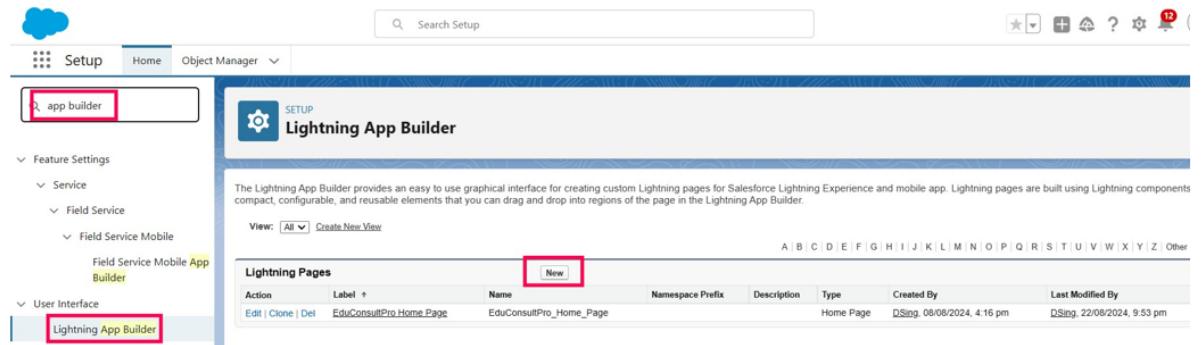
Task 10: Create A Lightning App Page

This task involves creating a Lightning App Page and making it available in the application. The new page will feature the "EduConsultantPro Flow" component.

Subtask 1: Create A Lightning App Page

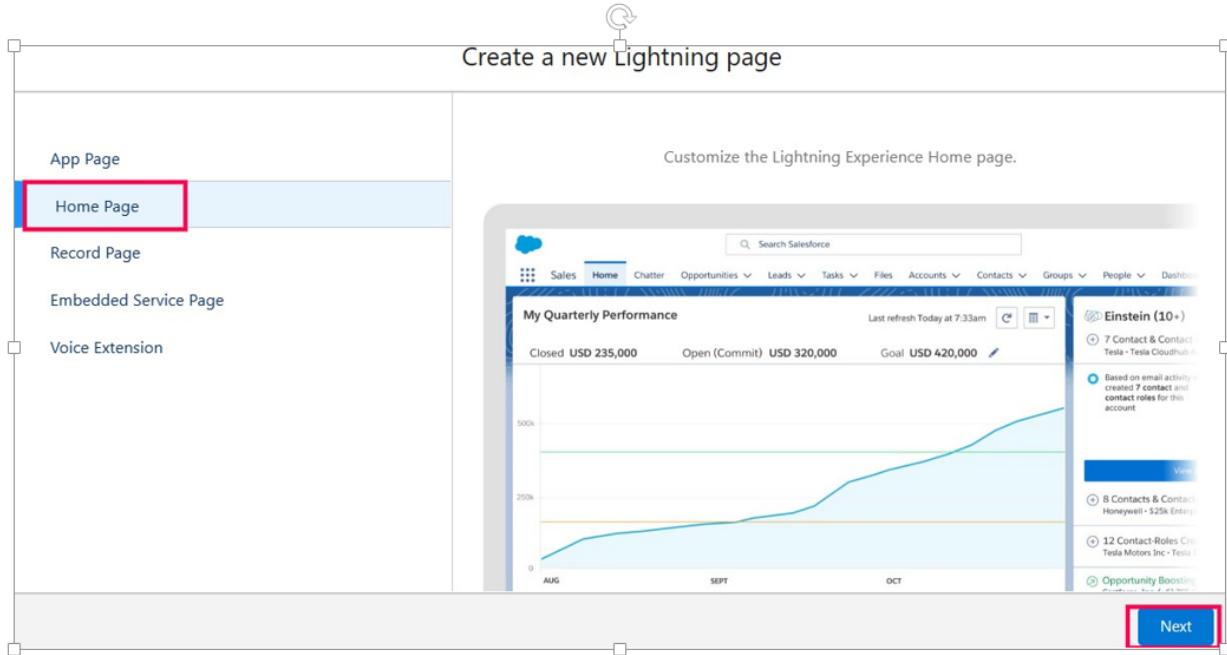
1. Access Lightning App Builder:

- a. From Setup, enter **App Builder** in the Quick Find box.
- b. Click **Lightning App Builder**.



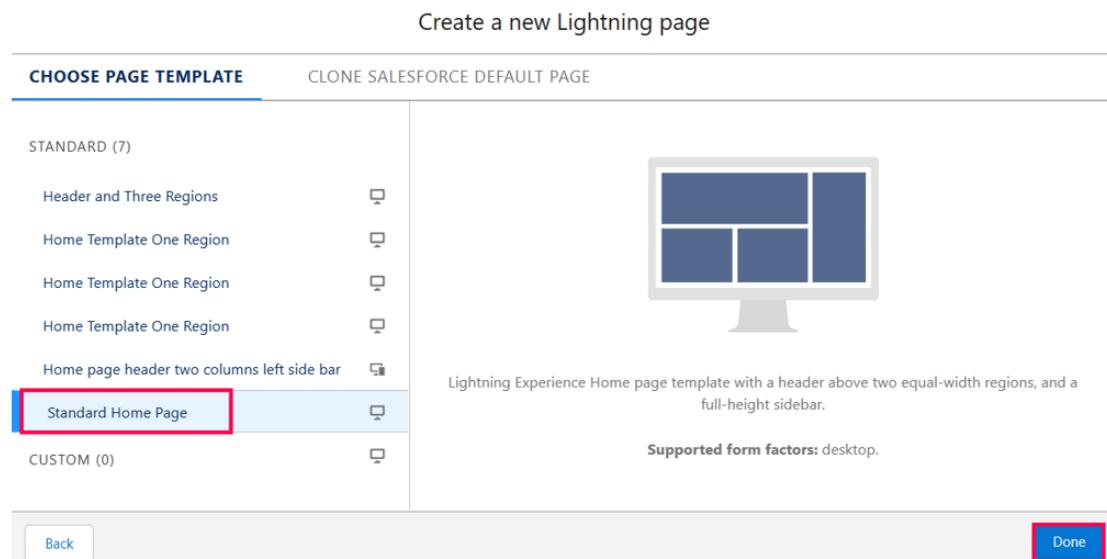
2. Create a New Home Page:

- c. Click **New**.
- d. Select **Home Page** and click **Next**.



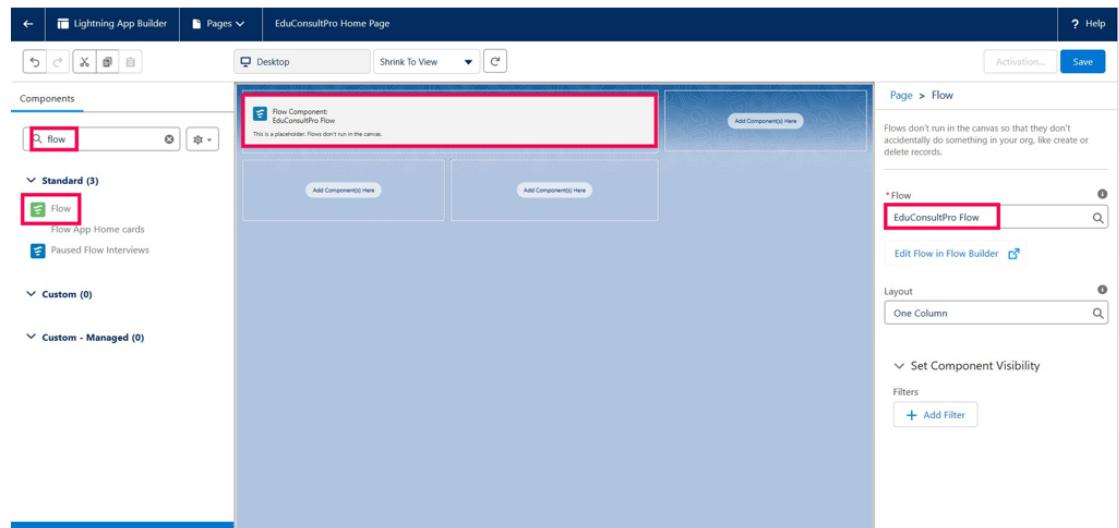
3. Configure the Home Page:

- e. Name the Page: Enter “EduConsultPro Home Page”.
- f. Select a Template: Choose the Standard Home Page template.
- g. Click Done.



4.Add the Flow Component:

- h. Drag the **Flow** component to the **top-right** region of the page layout.
- i. In the component properties pane, search for and select “**EduConsultantPro Flow**”.



5.Save and Activate the Page:

- j. Click **Save**.
- k. Click **Activate**.

6.Assign the Page to Apps and Profiles:

- l. Click **App and Profile**.
- m. Click **Assign to Apps and Profiles**.
- n. **Select the Sales App:** Choose **Sales** and click **Next**.
- o. **Assign to Profiles:** Scroll down and select **System Administrator** profile. Click **Next**.
- p. **Review and Save:** Review the assignment details and click **Save**.

Select Apps

First, select the Lightning apps to display "EduConsultPro Home Page" as the home page. You'll select the related profiles next.

Lightning Apps (12)		1 Selected
App Name	Description	
<input checked="" type="checkbox"/> Sales	Manage your sales process with accounts, leads, opportunities, and m...	
<input type="checkbox"/> Sales Console	(Lightning Experience) Lets sales reps work with multiple records on o...	
<input type="checkbox"/>		

Select Profiles

Select the profiles to display "EduConsultPro Home Page" as the home page.

Profiles (41)		1 Selected
Profile	Description	
<input type="checkbox"/> Silver Partner User		
<input type="checkbox"/> Solution Manager		
<input type="checkbox"/> Standard Platform User		
<input type="checkbox"/> Standard User		
<input checked="" type="checkbox"/> System Administrator		

Activation: EduConsultPro Home Page

The org default home page is displayed unless more specific assignments are made.

The app default home page is displayed for specified apps, and overrides the org default.

Any app and profile assignments are displayed for specified app and profile combinations, and they override all other assignments.

[Learn more about forecast page assignment in Salesforce Help.](#)

Org Default App Default App and Profile

Set the home page for different user profiles when they're using certain apps. These assignments are the most specific, and override all other home page assignments.

Assignments (1)		Add Assignments	Remove Assignments
App	Profile		
Sales	System Administrator		

[Close](#)