

A CRM Application to Manage the Services offered by an Institution

By

Santhosh kumar R

<https://www.salesforce.com/trailblazer/r7tsvx8d5kptxa3929>

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.

Table of Contents

Project Abstract.....	2
Table of Contents	2
Introduction	6
<i>TASK 1: Creating Developer Account</i>	7
<i>TASK 2: Account Activation.....</i>	8
<i>TASK 3: Create Objects From Spreadsheet.....</i>	9
Subtask 1: Create Course Object	9
Subtask 2: Create Relationships Among the Objects	12
Subtask 3: Configure the Case Object.....	14
Subtask 4: Create a Lightning App	16
<i>TASK 4: Create A ScreenFlow For Student Admission Application Process</i>	19
Subtask 1: Add Screen Element.....	19
Subtask 2: Create Student Record Using Create Element.....	21
Subtask 3: Add Course Selection Screen	22
Subtask 4: Add Decision Element.....	23
Subtask 5: Add GET Record Element.....	24
Subtask 6: Create Registration Record Using Create Records Element	25
Subtask 7: Create Email Text Template Variables	26
Subtask 8: Add Action Element	28
Subtask 9: Add Success Screen.....	29
<i>Task 5: Create Users.....</i>	30
Subtask 1: Create a User	30

Subtask 2: Configure the User Settings	31
Task 6: Create an Approval Process for the Property Object	32
Subtask 1: Create Email Templates.....	32
Subtask 2: Create An Approval Process	35
Task 7: Create A Record-Triggered Flow.....	39
Subtask 1: Configure The Start Element.....	39
Subtask 2: Add An Action Element	40
Task 8: Create A ScreenFlow For Existing Students To Book An Appointment	42
Subtask 1: Add Screen Element.....	42
Subtask 2: Get Record	43
Subtask 3: Add a Decision Element.....	44
Subtask 4: Add Screen Element.....	45
Subtask 5: Add GET Record Element.....	46
Subtask 6: Create Appointment Record Using Create Records Element.....	47
Subtask 7: Add Screen Element.....	48
Subtask 8: Add a Subflow Element.....	49
Task 9: Create A ScreenFlow To Combine All The Flows At One Place.....	50
Subtask 1: Add Welcome Screen Element.....	50
Subtask 2: Add Existing or New Student Confirmation Screen	51
Subtask 3: Add Decision Element.....	52
Subtask 4: Add Subflow for Existing Students.....	54
Subtask 5: Add Subflow for New Students.....	54
Task 10: Create A Lightning App Page	56

Subtask 1: Create A Lightning App Page.....56

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.

The project's primary objective was to streamline and automate the institute's core administrative processes. For admission application management, the goal was to enable prospective students to submit their applications through the institute's online portal. This information would be captured directly into the Salesforce CRM, ensuring efficient data handling. Additionally, automated email notifications were set up to acknowledge successful submissions, and tools were provided to admissions staff for generating insightful reports and dashboards, allowing for a deeper analysis of application metrics, acceptance rates, and enrollment trends.

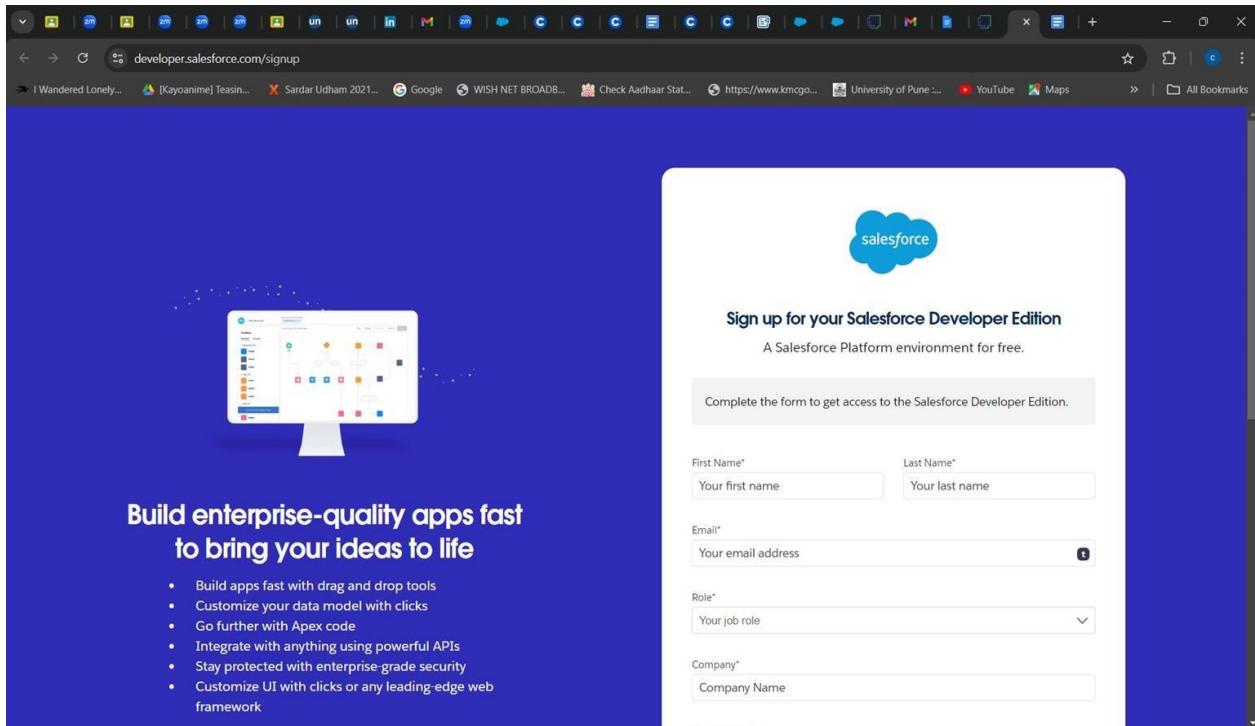
In terms of consulting services management, the project focused on creating a seamless process where students could request consulting services via the institute's website. These requests would be recorded in Salesforce, triggering automated notifications to consultants who could then schedule and manage appointments within the CRM. This ensured that the scheduling process was organized, and the status of each appointment, whether scheduled, completed, or canceled, was tracked effectively.

Another critical component was the implementation of an immigration case management system. Students could initiate immigration cases through various channels such as phone, email, or web. These cases were then logged into Salesforce, with immigration agents receiving immediate notifications to take action. The system also included tools for tracking case statuses and managing related documents, ensuring that every case was processed efficiently and transparently.

This comprehensive Salesforce solution not only addressed the immediate operational needs of EduConsultPro but also set the foundation for future scalability, enabling the institution to provide an enhanced and streamlined experience for both students and staff.

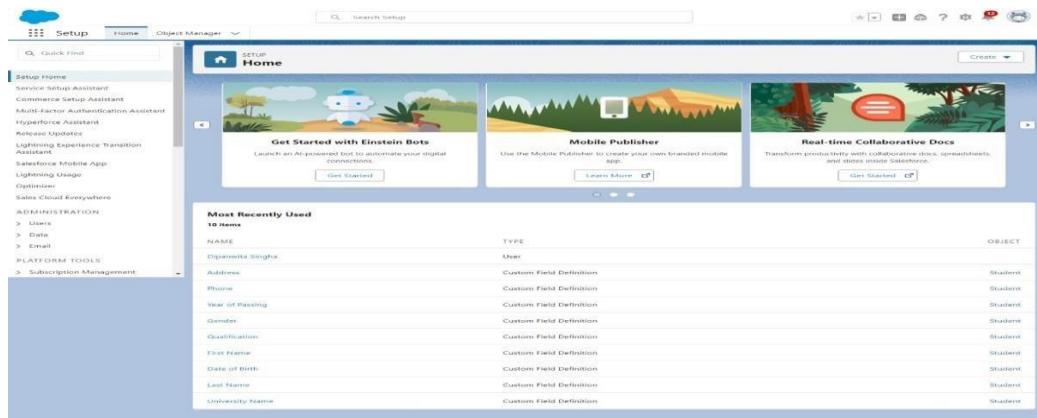
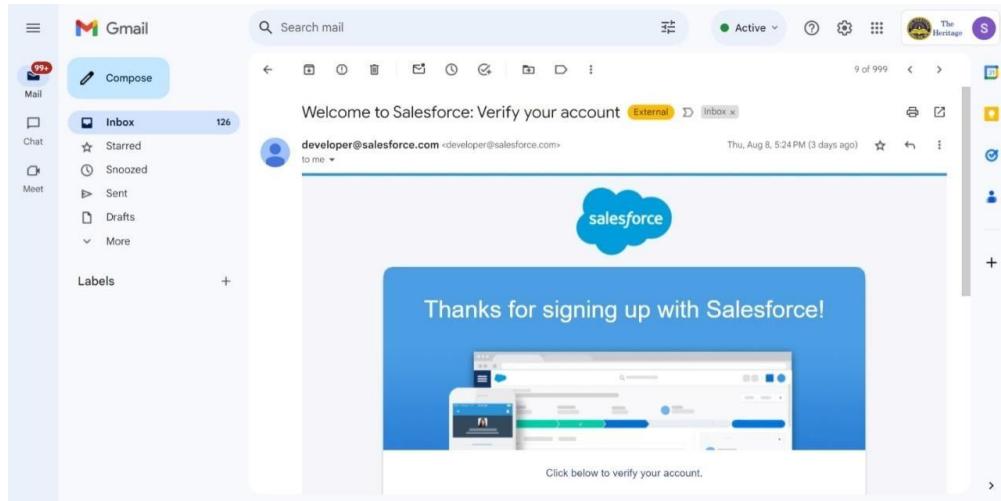
TASK 1: Creating Developer Account

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.

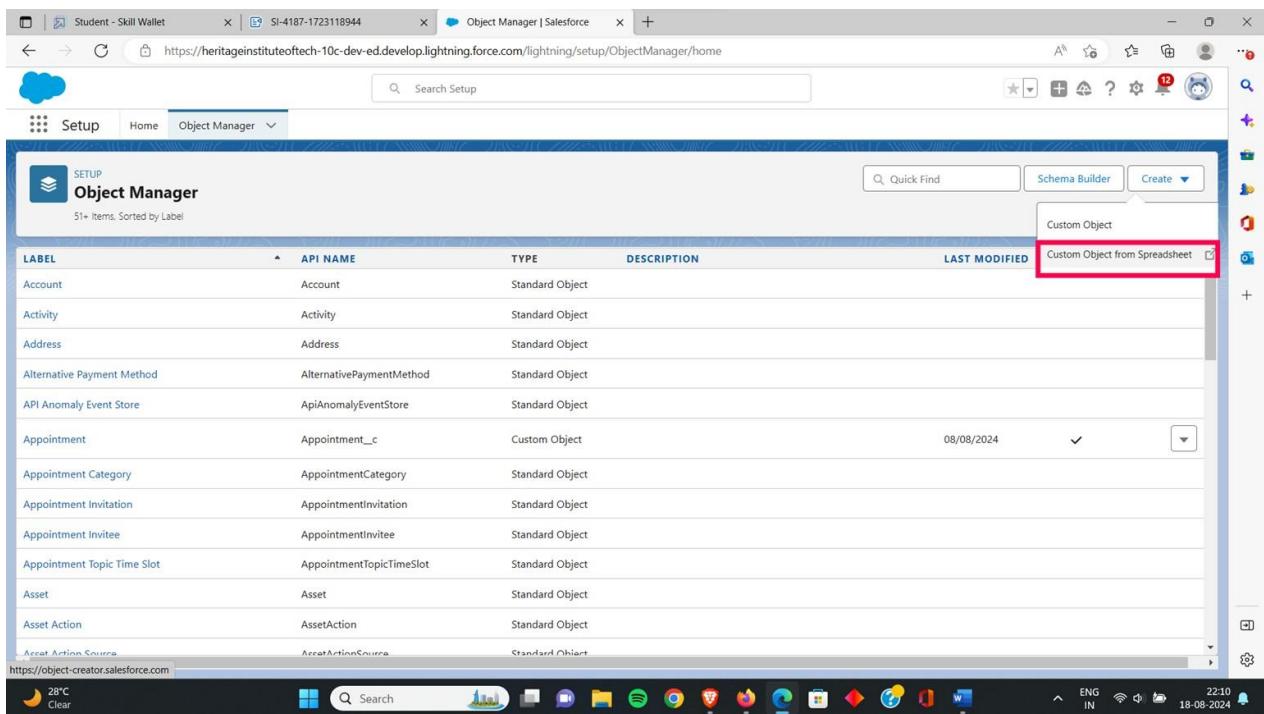


TASK 3: Create Objects from Spreadsheet

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

Subtask 1: Create Course Object

1. Navigate to **Object Manager**.
2. Click on **Create Object from Spreadsheet**.



The screenshot shows the Salesforce Object Manager interface. At the top, there's a navigation bar with tabs like 'Setup', 'Home', and 'Object Manager'. Below it is a search bar and a toolbar with various icons. The main area is titled 'Object Manager' and shows a list of objects. The columns are labeled 'LABEL', 'API NAME', 'TYPE', 'DESCRIPTION', and 'LAST MODIFIED'. A red box highlights the 'Create' button in the top right corner of the table header, which is specifically for creating 'Custom Object from Spreadsheet'.

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		
Appointment	Appointment_c	Custom Object		08/08/2024
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		
Asset Action	AssetAction	Standard Object		
Asset Action Source	AssetActionSource	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 * Field Labels Row
 Enter manually Detect from row 1

Import 2 rows of Data? No, skip import Yes, import data

Record Name Field Let Salesforce Create a Default

Fields 12 of 12 to import Hide mapped fields

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

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

* Plural Label

* API Name

Object Description

> Advanced Settings

Back Finish

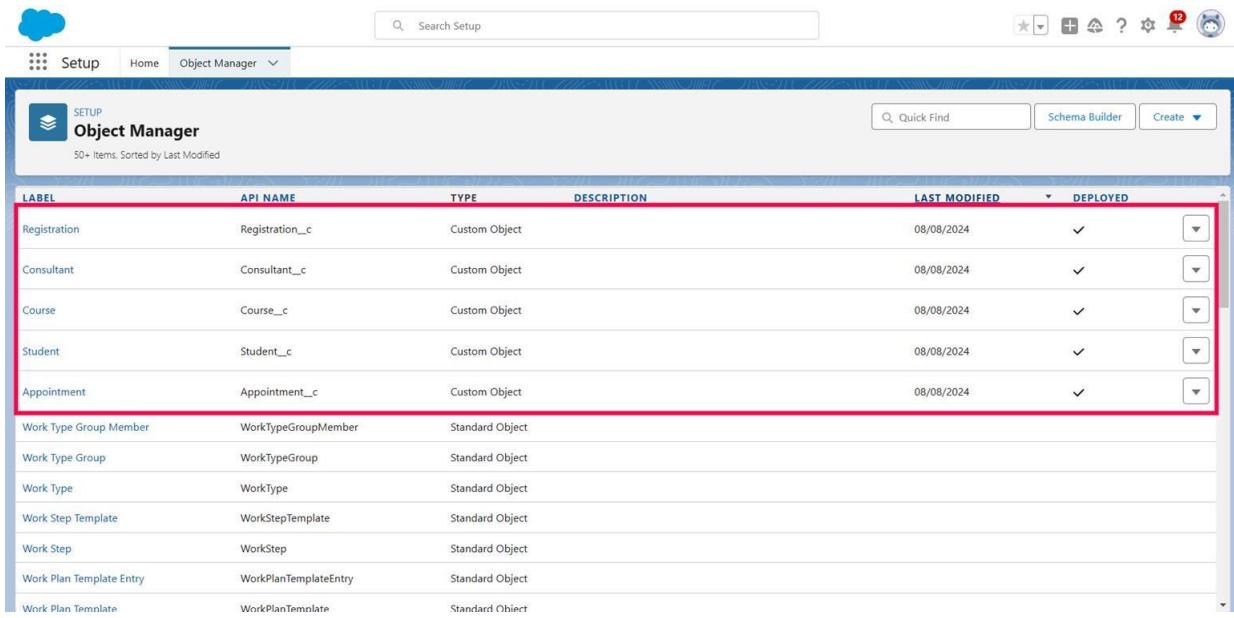
6. Following the same steps create the following objects:

a. Consultant

b. Student

c. Appointment

d. Registration

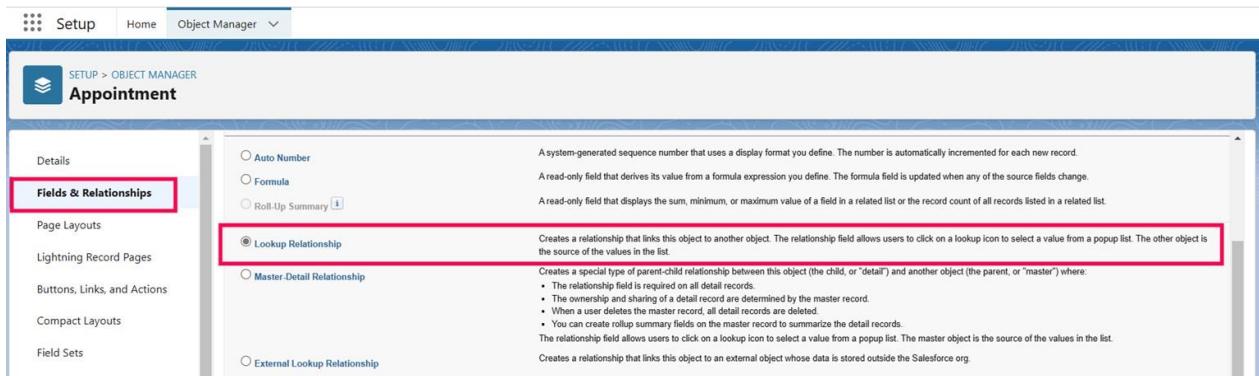


LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Registration	Registration_c	Custom Object		08/08/2024	✓
Consultant	Consultant_c	Custom Object		08/08/2024	✓
Course	Course_c	Custom Object		08/08/2024	✓
Student	Student_c	Custom Object		08/08/2024	✓
Appointment	Appointment_c	Custom Object		08/08/2024	✓
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

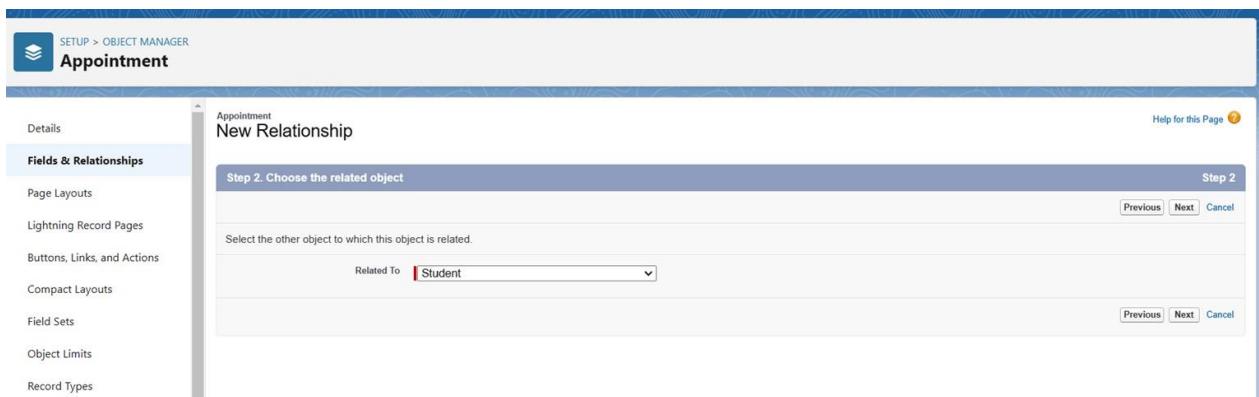
1. Create a Lookup relationship between Appointment and Student Object

- a.** Go to the Appointment object and create a new custom field from **Fields & Relationships** and choose the field type **Lookup relationship**



The screenshot shows the Salesforce Object Manager interface for the 'Appointment' object. The 'Fields & Relationships' tab is active. A red box highlights the 'Lookup Relationship' option under the 'Create New' section. The description for 'Lookup Relationship' states: '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.' Other options shown include 'Auto Number', 'Formula', 'Roll-Up Summary', 'Master-Detail Relationship', and 'External Lookup Relationship'.

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



The screenshot shows the 'New Relationship' wizard for the 'Appointment' object. It is on 'Step 2: Choose the related object'. The 'Related To' dropdown is set to 'Student'. The wizard has 'Previous', 'Next', and 'Cancel' buttons at the bottom.

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

- Appointment and Student Object**
- Student and Case Object**

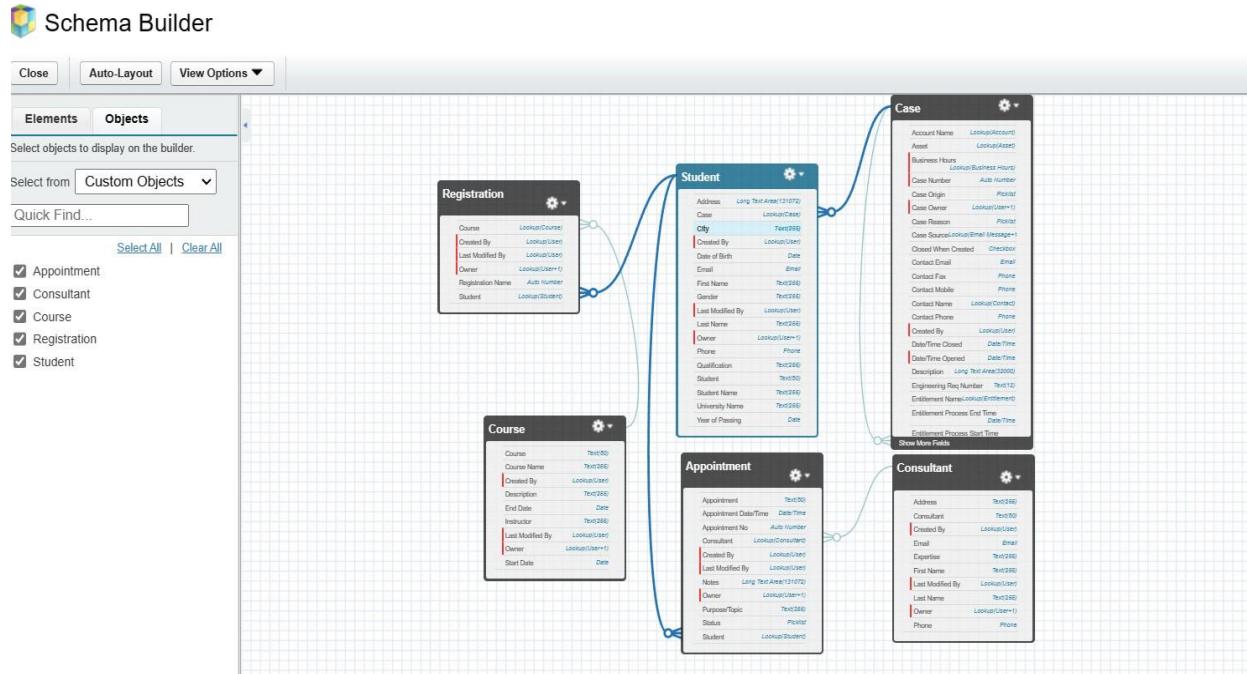
The Lookup relationship **Student** and **Case Object** is to manage student queries related to immigration or visa applications.

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 is a search bar labeled "Search Setup" and various navigation icons. Below the header, the "Object Manager" tab is selected. In the center, there is a table listing various objects with columns for "LABEL", "API NAME", "TYPE", "DESCRIPTION", "LAST MODIFIED", and "DEPLOYED". A red box highlights the "Schema Builder" button located in the top right corner of the table header area.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
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			
Appointment	Appointment_c	Custom Object		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:
 - **Immigration**
 - **Visa Application**

The screenshot shows the Salesforce Object Manager interface for the Case object. On the left, there's a sidebar with various configuration options like Case Page Layouts, Case Close Page Layouts, Lightning Record Pages, etc. The main area is titled 'Fields & Relationships' and lists several fields: Product, Service Contract, SLA Violation, Status, Stopped, Stopped Since, Subject, and Type. The 'Type' field is highlighted with a red box. At the bottom of the list, there are additional fields: Web Company, Web Email, Web Name, and Web Phone.

The screenshot shows the 'Case Type Picklist Values' section. It displays a table with columns: Action, Values, API Name, Default, Chart Colors, and Modified By. The table lists values for Action: Edit | Del | Deactivate, Values: Mechanical, Electrical, Electronic, Structural, Other, Immigration, and Visa Application. The 'Immigration' and 'Visa Application' rows are highlighted with a red box.

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	Mechanical	Mechanical	<input type="checkbox"/>	Assigned dynamically	Dipanwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Electrical	Electrical	<input type="checkbox"/>	Assigned dynamically	Dipanwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Electronic	Electronic	<input type="checkbox"/>	Assigned dynamically	Dipanwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Structural	Structural	<input type="checkbox"/>	Assigned dynamically	Dipanwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Other	Other	<input type="checkbox"/>	Assigned dynamically	Dipanwita Singha, 07/08/2024, 10:50 pm
Edit Del Deactivate	Immigration	Immigration	<input type="checkbox"/>	Assigned dynamically	Dipanwita Singha, 08/08/2024, 1:33 am
Edit Del Deactivate	Visa Application	Visa Application	<input type="checkbox"/>	Assigned dynamically	Dipanwita Singha, 08/08/2024, 1:33 am

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

- **Open**
- **In-progress**

SETUP > OBJECT MANAGER
Case

Fields & Relationships
2 Items, Sorted by Field Label

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

SETUP > OBJECT MANAGER
Case

Fields & Relationships

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

Subtask 4: Create a Lightning App

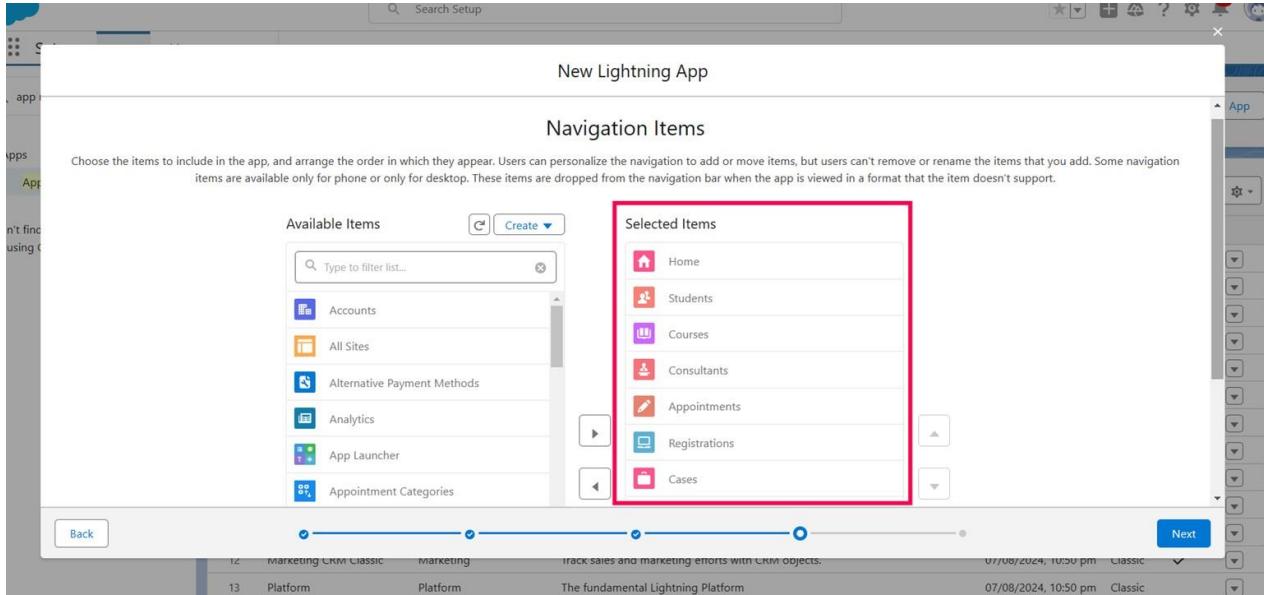
1. In **Setup**, search for **App Manager** in the Quick Find bar.
2. Click on **New Lightning App**.

The screenshot shows the Salesforce Setup interface. In the top left, there's a blue cloud icon. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar says 'Search Setup' with a magnifying glass icon. On the far right of the top bar are several small icons: a star, a plus sign, a question mark, a gear, a bell, and a user profile. Below the top bar, a sidebar on the left has a search field with 'app manager' typed in. It also has a section titled 'Apps' with 'App Manager' highlighted by a red box. A message below says 'Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Lightning Experience App Manager' with a sub-header 'SETUP'. It shows a table of 23 items, sorted by App Name. The columns are 'App Name', 'Developer Name', 'Description', 'Last Modified ...', 'Ap...', and 'Vi...'. The first few rows include 'All Tabs', 'Analytics Studio', 'App Launcher', 'Automation', 'Bolt Solutions', 'Community', 'Content', 'Data Manager', and 'Digital Experiences'. At the top right of this table area, there are two buttons: 'New Lightning App' and 'New Connected App', both of which are highlighted with red boxes. The bottom of the page shows a navigation bar with tabs like 'Marketing', 'Platform', etc., and a 'Next' button.

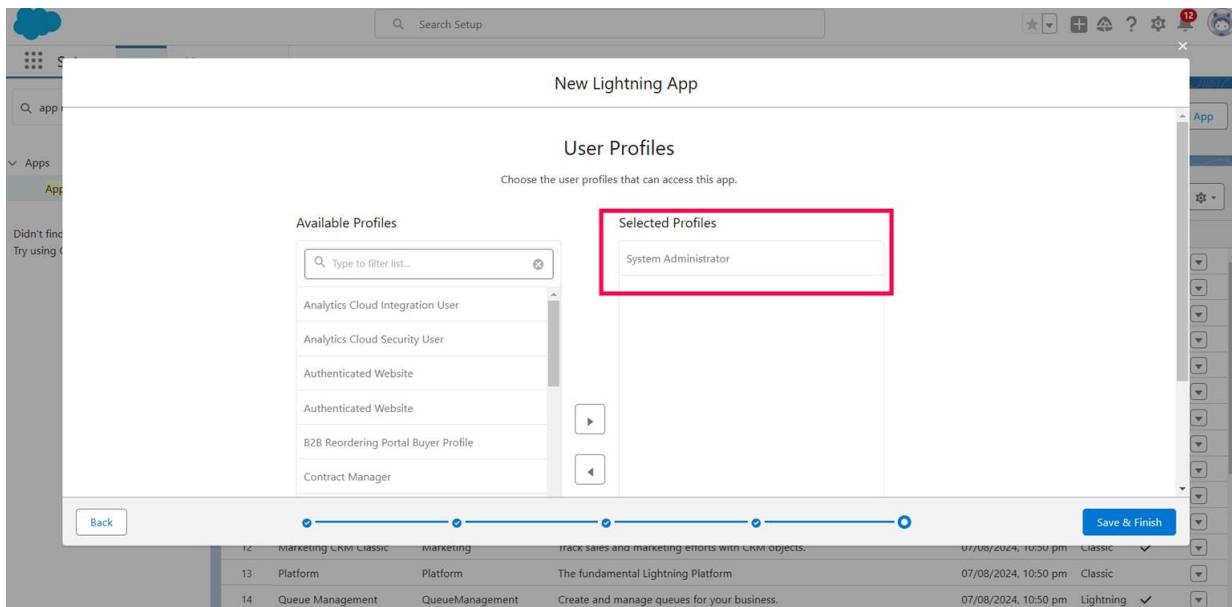
3. Name the app **EduConsultPro**.

This screenshot shows the 'New Lightning App' configuration page. The title at the top is 'New Lightning App'. Below it 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.' There are two main sections: 'App Details' and 'App Branding'. In 'App Details', there are three fields: 'App Name' (containing 'EduConsultPro'), 'Developer Name' (containing 'EduConsultPro'), and 'Description' (with the placeholder 'Enter a description...'). In 'App Branding', there is an 'Image' section with a placeholder 'Upload' button and a color picker set to '#0070D2'. Below this is an 'Org Theme Options' section with a checkbox labeled 'Use the app's image and color instead of the org's custom theme'. At the bottom of the page is a horizontal navigation bar with tabs for 'Marketing', 'Platform', etc., and a 'Next' button.

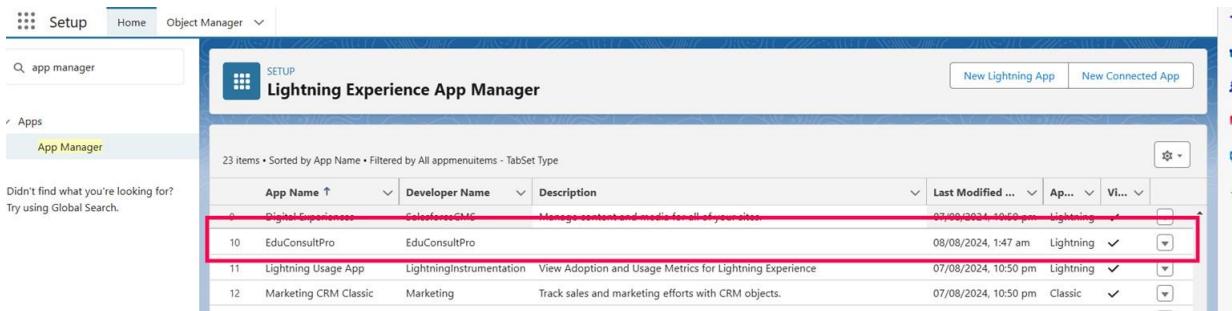
4. Add the following items from the Available Items list to the Selected Items list:
- o **Home, Students, Courses, Consultants, Appointments, Registrations, Cases**



5. Add the **System Administrator** profile from Available Profiles to Selected Profiles and click **Save & Finish**.



6. **EduConsultPro** app is created.



The screenshot shows the Salesforce App Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar on the left contains the text 'app manager'. On the left sidebar, under 'Apps', the 'App Manager' tab is selected. The main content area is titled 'Lightning Experience App Manager' and displays a table of applications. The table has columns for 'App Name', 'Developer Name', 'Description', 'Last Modified ...', 'Ap...', and 'Vi...'. There are 23 items listed, sorted by App Name. The 'EduConsultPro' app, which was created in step 5, is highlighted with a red border. Its details are as follows:

App Name	Developer Name	Description	Last Modified ...	Ap...	Vi...
10 EduConsultPro	EduConsultPro	Manages consults and leads for educational services.	08/08/2024, 1:47 am	Lightning	<input checked="" type="checkbox"/>
11 Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	07/08/2024, 10:50 pm	Lightning	<input checked="" type="checkbox"/>
12 Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	07/08/2024, 10:50 pm	Classic	<input checked="" type="checkbox"/>

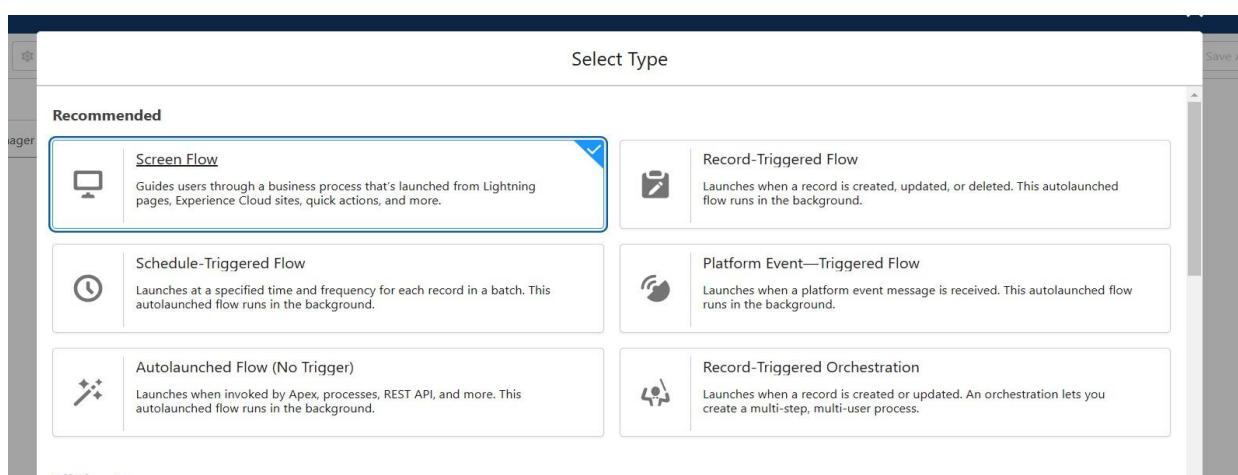
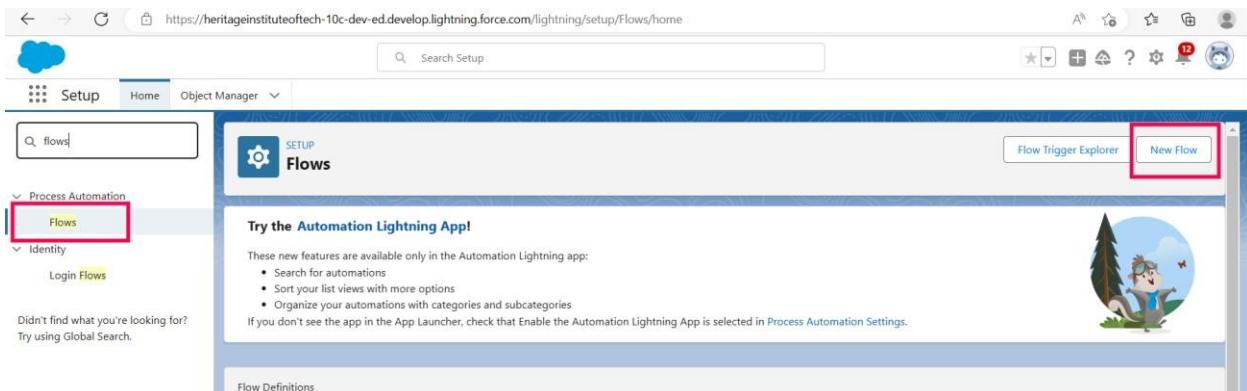
TASK 4: Create A ScreenFlow For Student Admission Application Process

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.

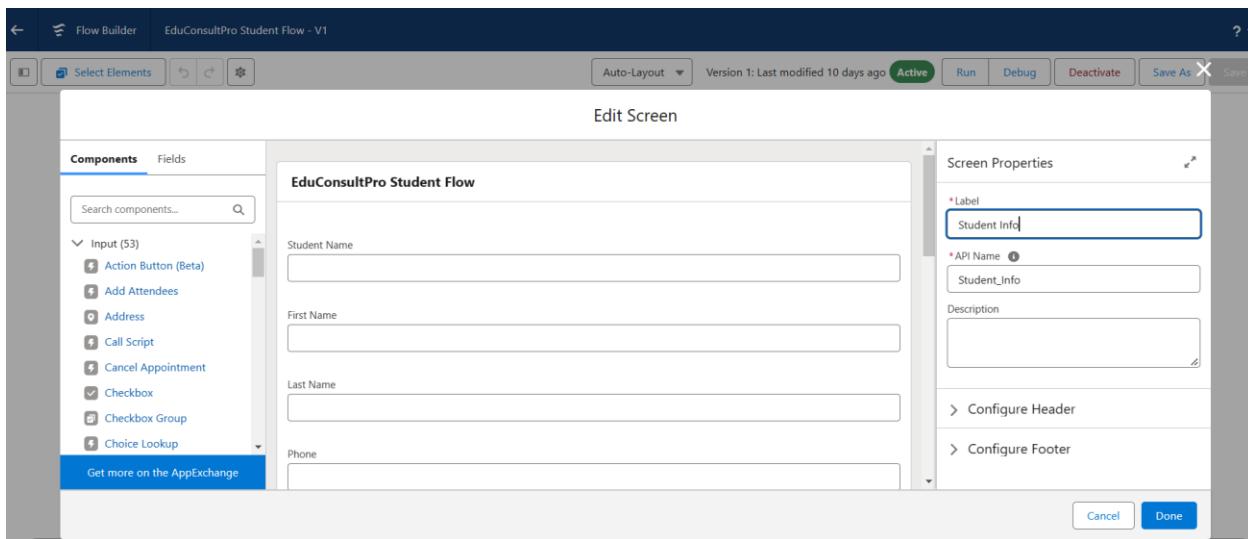
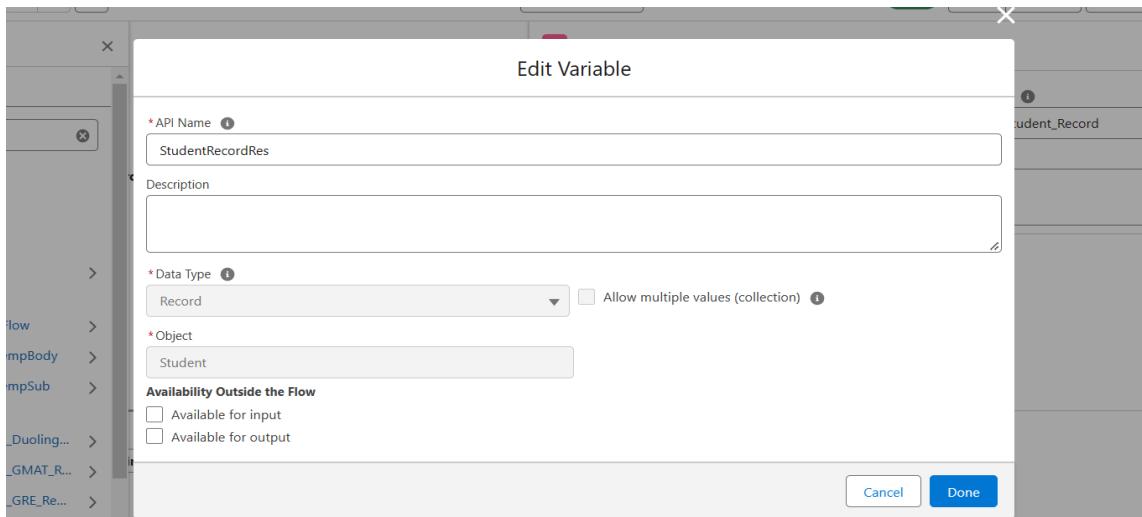
1. Navigate to **Setup**, enter **Flow** in the Quick Find box, and select **New Flow**. Choose **ScreenFlow**.



2. Add a **Screen** element.

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

- **Label:** Enter “Student Info”.
- **Fields:** Create a new **Resource** (StudentRecordRes) to display all fields from the **Student** object.
- **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 Student Record Using Create Element

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 record fields**.
5. Select the **StudentRecordRes** variable from the Student Info screen element under **Create a record from these values**.

The screenshot shows the configuration of a 'Create Records' element. It includes fields for Label (Create Student Record), API Name (Create_Student_Record), Description (empty), and How to set record field values (From a Record Variable). Under 'How Many Records to Create', 'One' is selected. In the 'Create a Record from These Values' section, 'StudentRecordRes' is listed as the record variable. 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.' i

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

Create a Record from These Values:

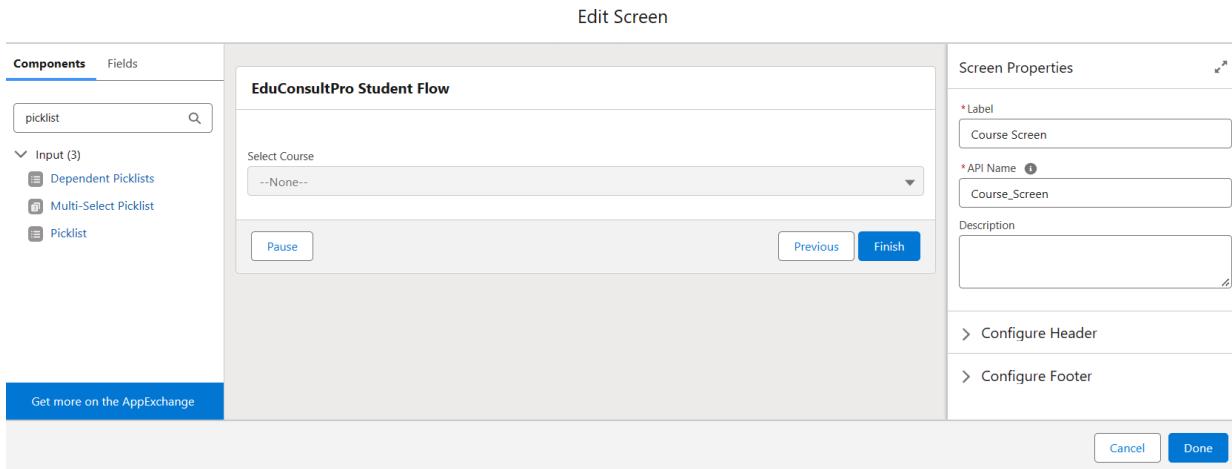
* Record: StudentRecordRes

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

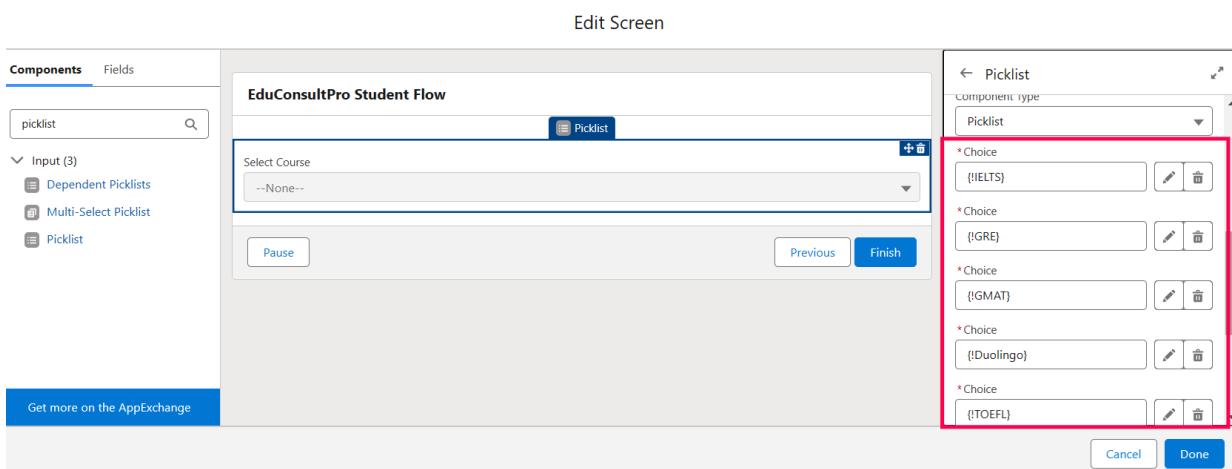
Subtask 3: Add Course Selection Screen

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



Subtask 4: Add Decision Element

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:
 - **Outcome Label:** “Selected IELTS”.
 - **Condition:**
 - **Resource:** Select_Course (Screen Component from Select Course Screen Element)
 - **Operator:** Equals
 - **Value:** IELTS (Choice Variable from Select Course Screen Element)
 - Repeat for GRE, GMAT, Duolingo, TOEFL.

The screenshot shows the configuration of a 'Decision' element. The main panel includes fields for 'Label' (containing 'Selecting Course'), 'API Name' (containing 'Selecting_Course'), and a 'Description' text area. Below this is an 'Outcomes' section with a table for defining outcomes for different courses based on resource, operator, and value conditions.

OUTCOME ORDER	OUTCOME DETAILS	DELETE OUTCOME
Selected IELTS	*Label: Selected IELTS *Outcome API Name: Selected_IELTS	Delete Outcome
Selected GRE		
Selected GMAT		
Selected Duolingo		
Selected TOEFL		
Default Outcome		

Condition Requirements to Execute Outcome: All Conditions Are Met (AND)

Resource: Aa Course Screen > Select_Course X

Operator: Equals

Value: Aa IELTS X

[+ Add Condition](#)

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 of a 'Get Records' element. The 'Label' field is 'Get IELTS Rec', 'API Name' is 'GetIELTS_Rec', and the 'Object' is 'Course'. The 'Condition Requirements' dropdown is set to 'All Conditions Are Met (AND)'. A single condition is defined with 'Name' field, 'Equals' operator, and 'Value' 'Select_Course'.

Field	Operator	Value
Name	Equals	Aa Select_Course X

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:
 - **Course_Name_c:** {!Get_[Course Name]_Rec.Id}
 - **Student_Name_c:** {!StudentRecordRes.Id}

The screenshot shows the 'Create Records' configuration interface. It includes sections for 'Label' (Create IELTS Registration Rec), 'API Name' (Create_IELTS_Registration_Rec), 'Description' (empty), 'How to set record field values' (Manually), 'Create a Record of This Object' (Registration), and 'Set Field Values for the Registration' (two fields: Course_c and Student_c, each with a value from another record). The 'Course_c' field has a value of 'Aa Course from GetIELTS_Rec > Record ID'. The 'Student_c' field has a value of 'Aa StudentRecordRes > Record ID'.

Create Records

* Label: Create IELTS Registration Rec

* API Name: Create_IELTS_Registration_Rec

Description

* How to set record field values: Manually

Create a Record of This Object: Registration

Set Field Values for the Registration

Field	Value
Course_c	Aa Course from GetIELTS_Rec > Record ID
Student_c	Aa StudentRecordRes > Record ID

+ Add Field

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

* API Name ⓘ
StuRegistrationEmailTextTempBody

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.

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 i

StuRegistrationEmailTextTempSub

Description

* Body i

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 Action Element

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:**
 - **Body:** { !StuRegistrationEmailTextTempBody }
 - **Recipient Address List:** { !StudentRecordRes.Email_c }
 - **Subject:** { !StuRegistrationEmailTextTempSub }

The image shows two separate configuration panels for the 'Send Email' action, likely from a low-code development platform like MuleSoft Anypoint Studio.

Left Panel (Initial Configuration):

- Label:** Send Email to Student
- API Name:** Send_Email_to_Student
- Description:** (Empty)
- Selected Action:** Send Email (emailSimple-emailSimple)
- Notes:** 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)

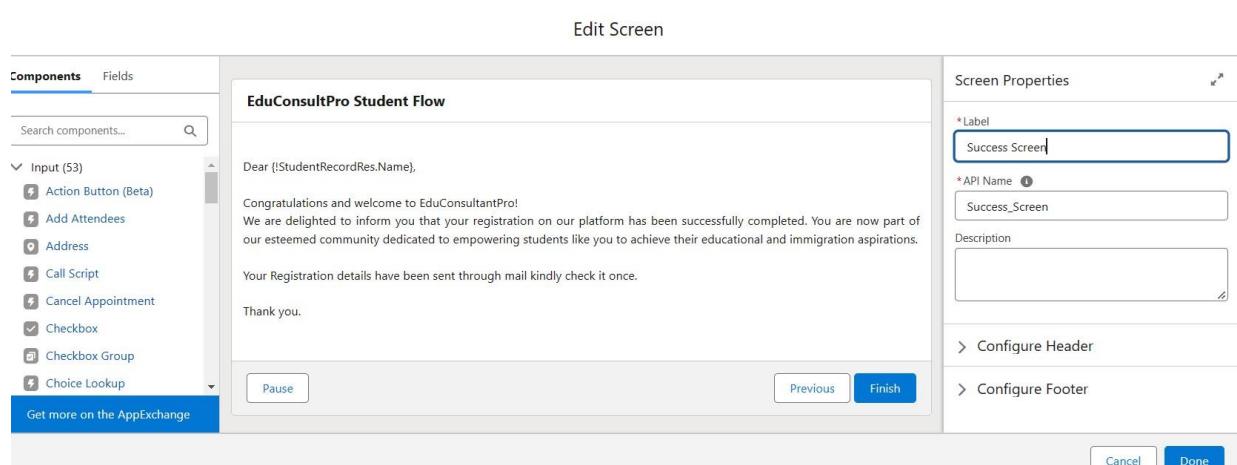
Right Panel (Final Configuration):

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

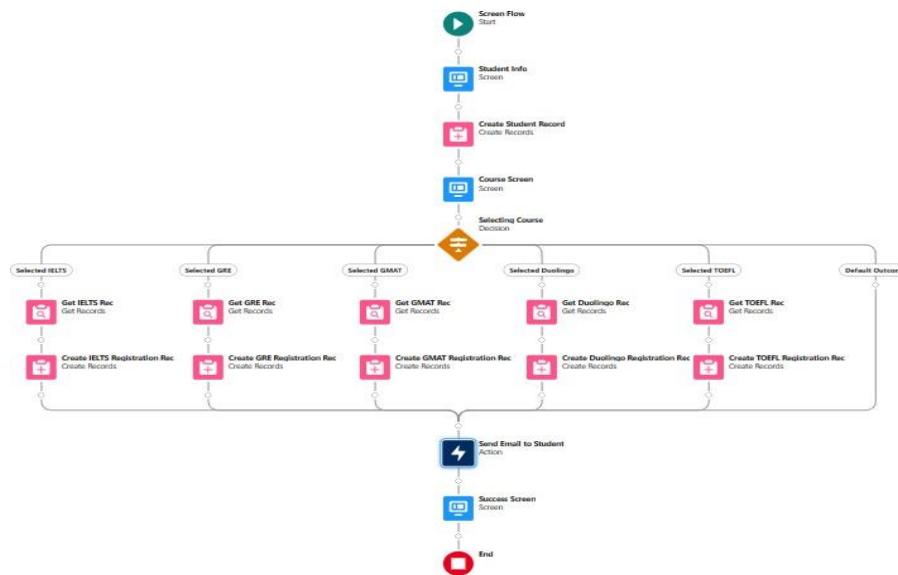
Subtask 9: Add Success Screen

Display a confirmation message to the student after the registration.

1. Add a **Screen** element after the Send Email to Student Action Element.
2. **Label:** “Success Screen”.
3. Add a **Display Text** component:
 - **Label:** “SuccessMessage”.
 - **Text:** Write the success message.



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



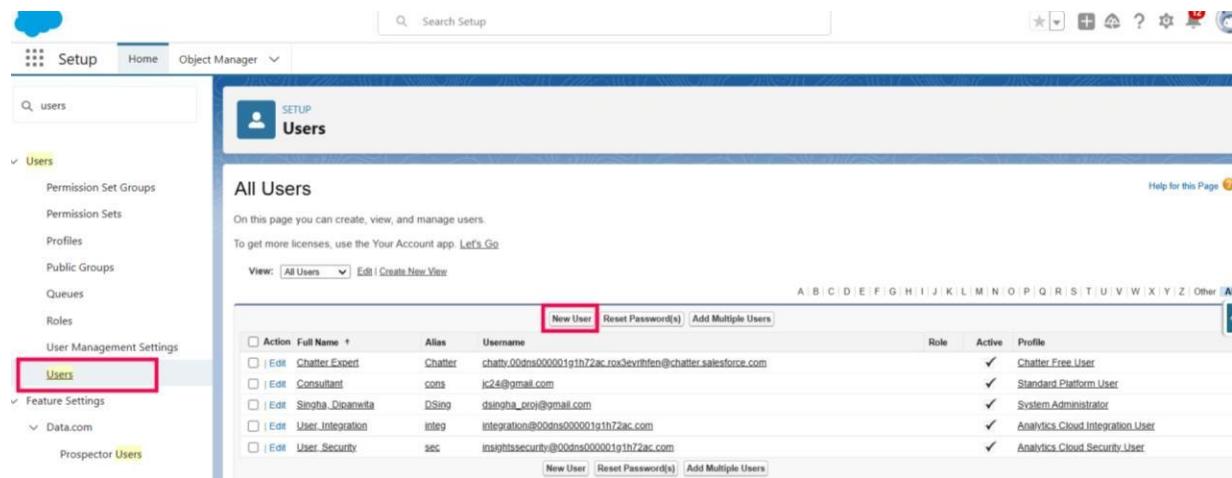
Task 5: Create Users

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 with the 'Users' page selected. On the left, there's a sidebar with 'User Management Settings' expanded, showing 'Users' highlighted with a red box. The main area displays a table of 'All Users' with columns for Action, Full Name, Alias, Username, Role, Active, and Profile. A red box highlights the 'New User' button at the top right of the table. The table lists several users, each with a checkbox and edit links. The 'Profile' column indicates profiles like 'Chatter Free User', 'Standard Platform User', 'System Administrator', 'Analytics Cloud Integration User', and 'Analytics Cloud Security User'. The 'Active' column shows checkmarks for most users.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatty_00dns000001g1h72ac.rox3evriffen@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit	Consultant	cont	lc24@gmail.com		✓	Standard Platform User
<input type="checkbox"/> Edit	Singha_Dipanwita	DSing	dasingha_coy@gmail.com		✓	System Administrator
<input type="checkbox"/> Edit	User_Integration	Integ	integration@00dns000001g1h72ac.com		✓	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User_Security	sec	insightssecurity@00dns000001g1h72ac.com		✓	Analytics Cloud Security User

2. Enter the following details:

- **Last Name:** Consultant
- **License:** Salesforce Platform
- **Profile:** Standard Platform User

3. Complete all mandatory fields as required and save.

The screenshot shows the Salesforce Setup interface under the 'Users' section. A search bar at the top left contains the text 'users'. On the left sidebar, under the 'Users' category, 'Prospector Users' is selected. The main area displays a 'User Edit' screen for a user named 'Consultant'. The 'General Information' section includes fields for First Name, Last Name (highlighted with a red box), Alias, Email, Username, Nickname, Title, Company, Department, and Division. To the right, there are sections for 'Role', 'User License' (set to 'Salesforce Platform'), 'Profile' (set to 'Standard Platform User'), and 'Active' status. Other optional settings like Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, Data.com User Type, and Data.com Monthly Addition Limit are shown below.

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 **Approver Settings** section.
3. In the **Manager Field**, select “Consultant” and save.

The screenshot shows the 'Approver Settings' section of the user edit form. It includes fields for 'Delegated Approver' and 'Manager' (highlighted with a red box). Below these, a dropdown for 'Receive Approval Request Emails' is set to 'Only if I am an approver'. At the bottom of the section are 'Save', 'Save & New', and 'Cancel' buttons.

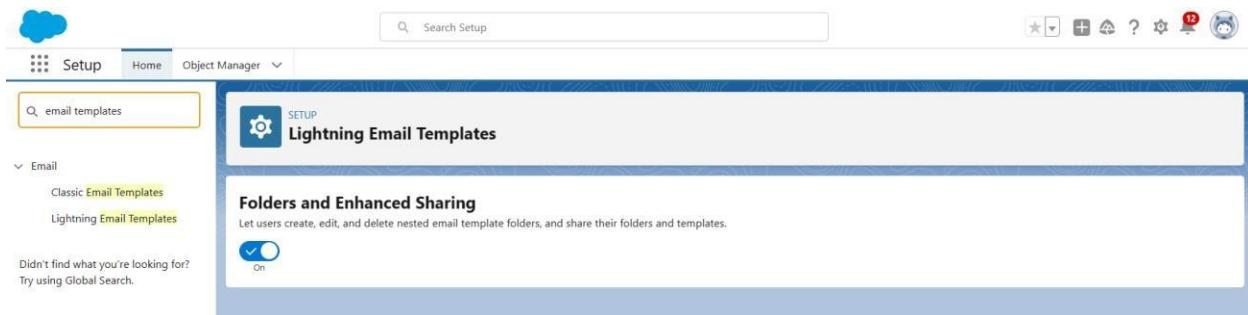
Task 6: Create an Approval Process for the Property Object

Establish an approval process for the Property object, including creating necessary email templates for notifications.

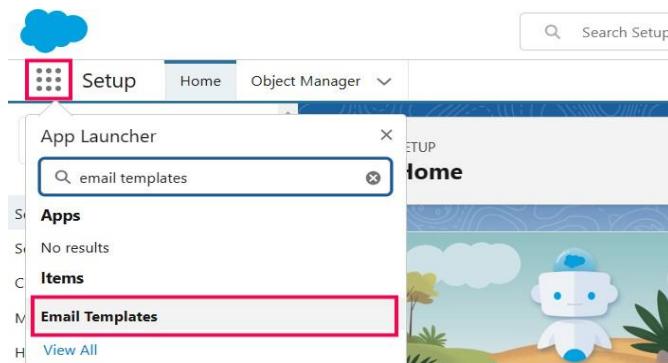
Subtask 1: Create Email Templates

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:

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

Edit Submission Template

* = Required Information

Information

* Email Template Name Submission Template	Related Entity Type -- None --
Description	Folder EduConsultantpro_Email Select Folder

Message Content

Subject Appointment Request with EduConsultantpro Consulta	Enhanced Letterhead Search Enhanced Letterheads...
---	---

HTML Value

HTML Value

Source Font Size **A**

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

[Cancel](#) [Save](#)

4. Create two additional email templates:

- **Approval Template:** Similar to the Submission Template but tailored for approval notifications.

- **Rejection Template:** Similar to the Submission Template but tailored for rejection notifications.

Edit Approval Request Template

* = Required Information

Information

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

Edit Rejection Request Template

* = Required Information

Information

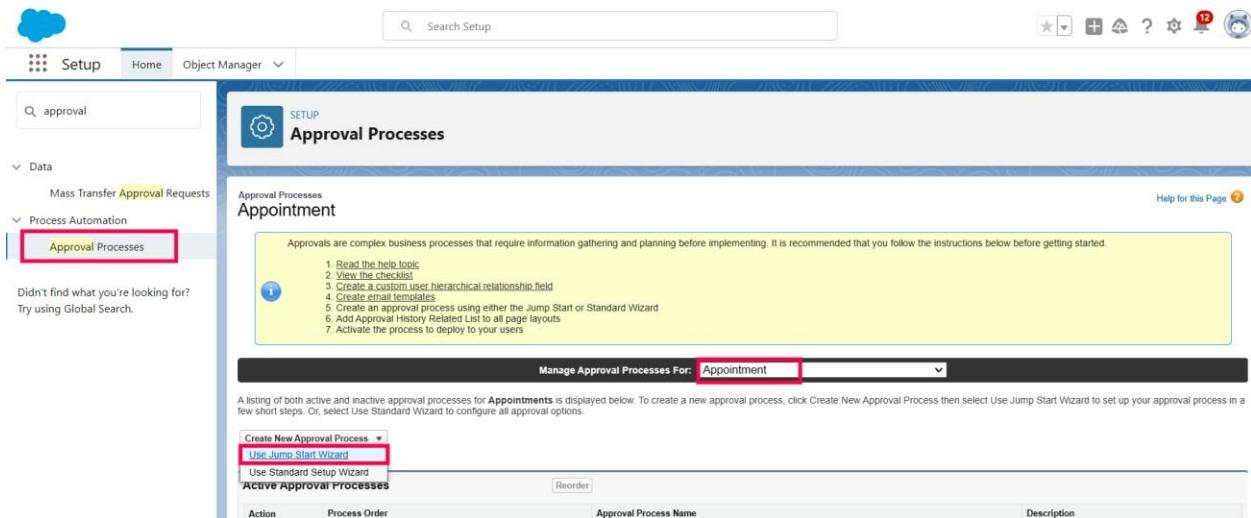
* Email Template Name <input type="text" value="Rejection Request Template"/>	Related Entity Type <input type="text" value="-- None --"/>
Description <input type="text"/>	Folder <input type="text" value="EduConsultantpro_Email"/> Select Folder

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	
Created by Me	Submission Template		EduConsultantpro_Email	Dipanwita Singha	8/8/2024, 2:09 pm	
Private Email Templates	Rejection Request Template		EduConsultantpro_Email	Dipanwita Singha	8/8/2024, 2:21 pm	
Puhlin Email Templates						

Subtask 2: Create An Approval Process

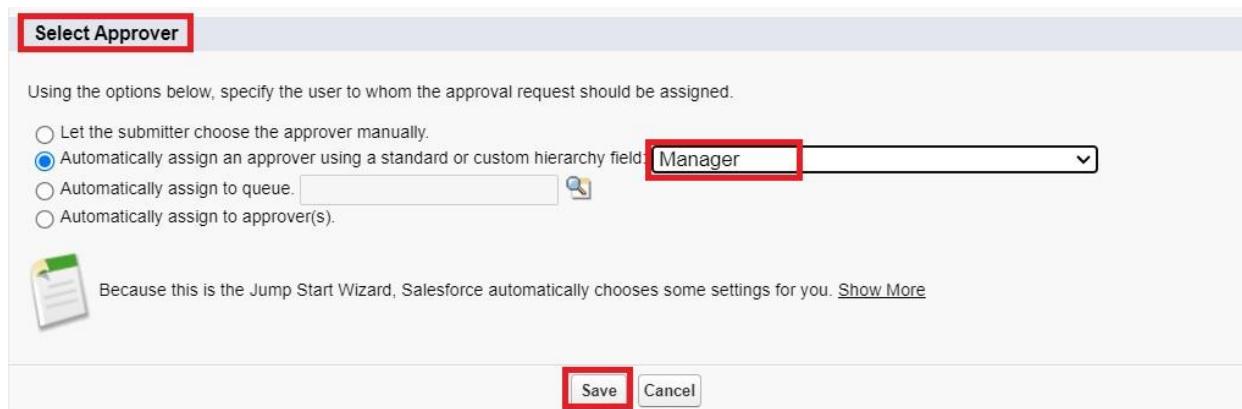
Configure an approval process for the Appointment object to manage request approvals.

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**.



The screenshot shows the Salesforce Setup interface. The left sidebar has sections for Data, Mass Transfer Approval Requests, and Process Automation, with 'Approval Processes' highlighted by a red box. The main content area is titled 'Approval Processes' under 'Appointment'. It contains instructions for creating approval processes and a dropdown menu 'Manage Approval Processes For: Appointment'. A red box highlights the 'Use Jump Start Wizard' option in the dropdown menu.

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



The screenshot shows the 'Select Approver' configuration screen. It includes options for letting the submitter choose the approver manually, automatically assigning to a queue, or automatically assigning to approver(s). A red box highlights the 'Manager' dropdown in the 'Automatically assign an approver using a standard or custom hierarchy field' section. Below the dropdown, there is a note: 'Because this is the Jump Start Wizard, Salesforce automatically chooses some settings for you.' with a 'Show More' link. At the bottom are 'Save' and 'Cancel' buttons, with 'Save' highlighted by a red box.

5. Click **Next** and select **Manager** for the option **Automated Approver 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

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 Approval Process Detail Page**.

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

- Select **Field Update**:
 - **Field to Update:** Appointment: Status
 - **Value:** Pending

Field Update Edit

Identification

Name	<input type="text" value="Submitted"/>
Unique Name	<input type="text" value="Submitted"/> 
Description	<input type="text"/>
Object	Appointment
Field to Update	Appointment: Status
Field Data Type	Picklist
Re-evaluate Workflow Rules after Field Change	<input type="checkbox"/> 

Specify New Field Value

Picklist Options

The value above the current one
 The value below the current one
 A specific value 

10. Click **Add New**:

- Select **Email Alert**:
 - **Description:** Submission Email Alert
 - **Email Template:** Submission Template
 - **Recipient Type:** Select your Name

Email Alert Edit

Edit Email Alert

Description	<input type="text" value="Submission Email Alert"/>
Unique Name	<input type="text" value="Submission_Email_Alert"/> 
Object	Appointment
Email Template	<input type="text" value="Submission Template"/> 
Protected Component	<input type="checkbox"/>

Recipient Type  for

Recipients

Available Recipients	Selected Recipients
<input type="text" value="User: Consultant"/> <input type="text" value="User: Integration User"/> <input type="text" value="User: Security User"/>	<input type="text" value="User: Dipanwita Singha"/>
<input type="button" value="Add"/>  	<input type="button" value="Remove"/>

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

Final Approval Actions i		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 i		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	

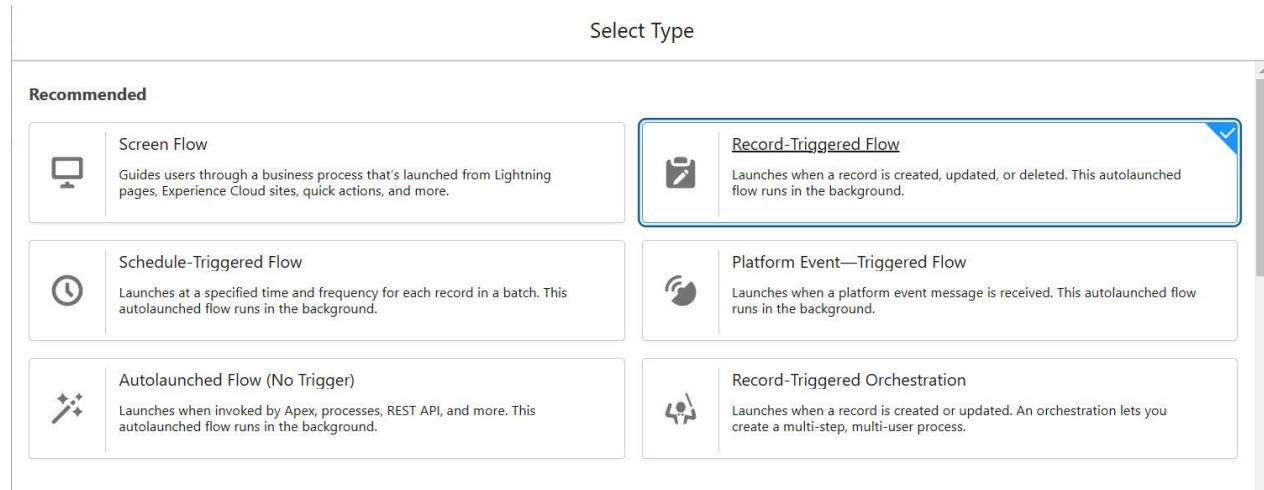
Task 7: Create A Record-Triggered Flow

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 **Configure Start** 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

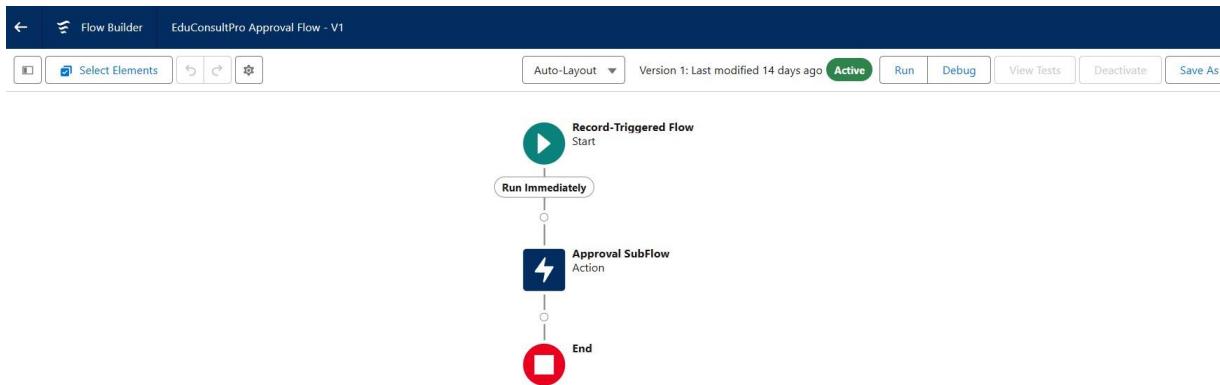
Use values from earlier in the flow to set the inputs for the "Submit for Approval" core action. To use its outputs later in the flow, store them in variables.

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 Approval Flow**, and click **Activate**.

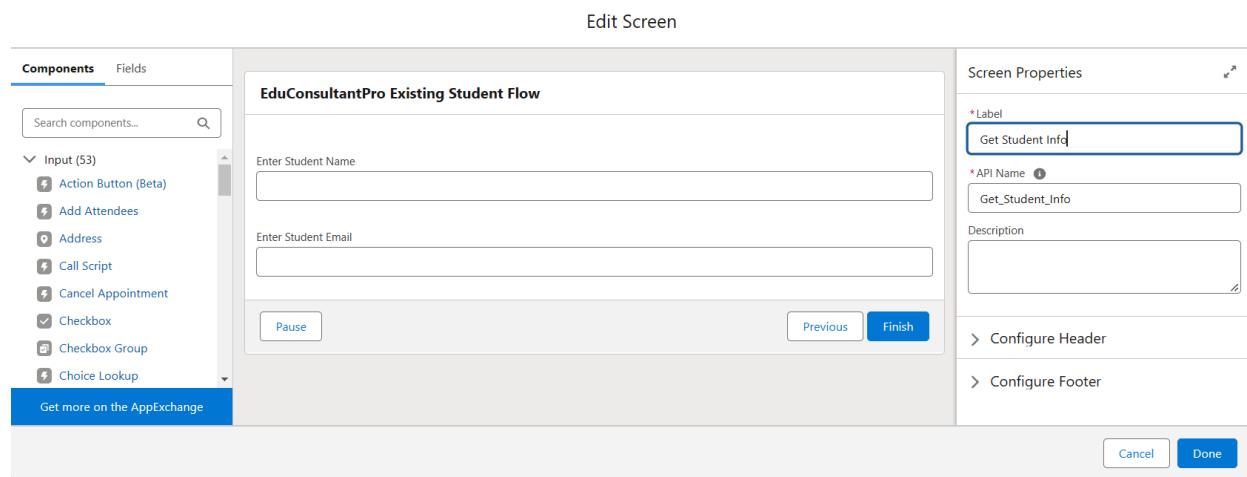


Task 8: Create A ScreenFlow For Existing Students To Book An Appointment

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.
 - 1st Text Component Label: **Enter Student Name**
 - 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:
 - **Select Object:** Student
 - **Condition Requirement:** All Conditions are Met (AND)
 - **Field:** Student Name
 - **Operator:** Equals
 - **Value:** {!Enter_Student_Name}
 - **Field:** Email_c
 - **Operator:** Equals
 - **Value:** {!Enter_Student_Email}

The screenshot shows a configuration interface for a 'Get Records' element. At the top, there are fields for 'Label' (containing 'Get Rec') and 'API Name' (containing 'Get_Rec'). Below these are 'Description' and 'Object' fields (both empty). The main section is titled 'Get Records of This Object' and contains a 'Student' object field. Under 'Filter Student Records', a dropdown menu shows 'All Conditions Are Met (AND)'. Two conditions are listed: one for 'Student_Name__c' with value 'Enter_Student_Name' and another for 'Email_c' with value 'Enter_Student_Email'. A '+ Add Condition' button is at the bottom.

* Label
Get Rec

* API Name
Get_Rec

Description

* Object
Student

Get Records of This Object

Condition Requirements
All Conditions Are Met (AND)

Field	Operator	Value
Student_Name__c	Equals	Aa Enter_Student_Name X
AND Email_c	Equals	Aa Enter_Student_Email X

+ Add Condition

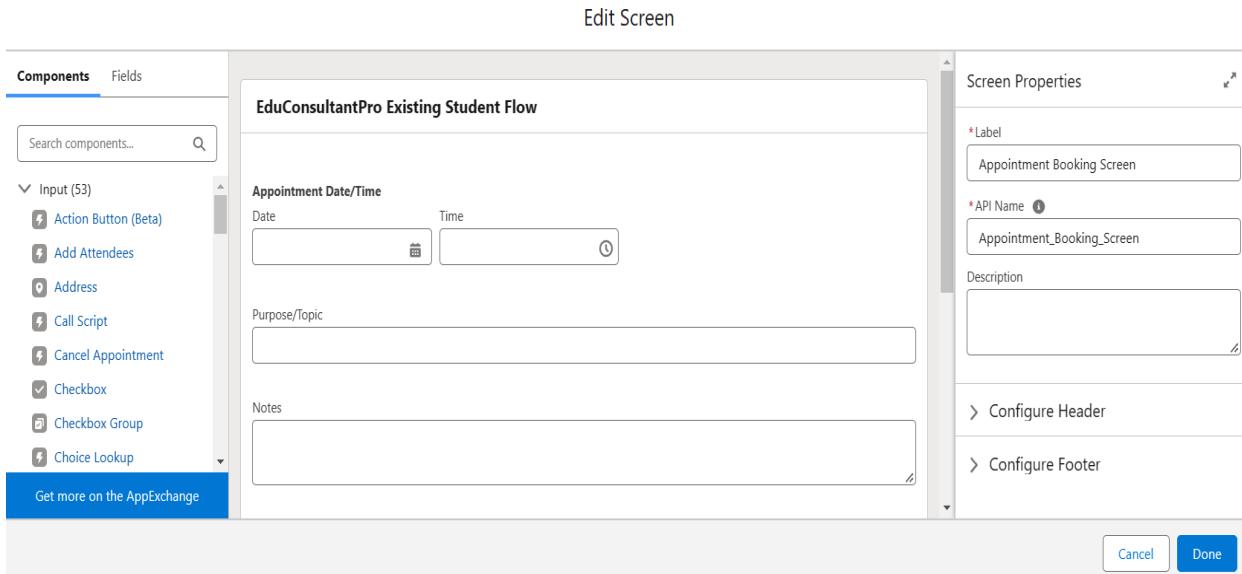
Subtask 3: Add a Decision Element

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

The screenshot shows the configuration of a 'Decision' element. At the top, there are fields for 'Label' (Appointment or Case) and 'API Name' (Appointment_or_Case). Below this is a 'Description' field which is empty. The main section 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.' On the left, there's a sidebar with 'OUTCOME ORDER' and a '+' button. The main area is titled 'OUTCOME DETAILS'. It lists two outcomes: 'Appointment' and 'Case'. For the 'Appointment' outcome, there is a 'Label' field (Appointment), an 'Outcome API Name' field (Appointment), and a 'Condition Requirements to Execute Outcome' dropdown set to 'All Conditions Are Met (AND)'. Below this, there are fields for 'Resource' (Aa How_may_I_Help_you), 'Operator' (Equals), and 'Value' (Aa Book_an_Appointment). A 'Delete Outcome' button is located at the top right of the outcome details area.

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 Decision element, under the Appointment path, and label it as “**Get Consultant Rec**”.
2. Configure the GET Record element:
 - **Select Object:** Consultant
 - **Condition Requirement:** All Conditions are Met (AND)
 - **Field:** Name
 - **Operator:** Equals
 - **Value:** {!AppointmentRecordRes.Consultant_Name_c}

The screenshot shows the configuration interface for a 'Get Records' element. At the top, there are fields for 'Label' (containing 'Get Consultant Rec') and 'API Name' (containing 'Get_Consultant_Rec'). Below these are 'Description' and 'Get Records of This Object' sections. In the 'Get Records of This Object' section, the 'Object' field is set to 'Consultant'. The 'Filter Consultant Records' section contains a 'Condition Requirements' dropdown set to 'All Conditions Are Met (AND)' and a table for defining filter conditions. The table has columns for 'Field' (containing 'Name'), 'Operator' (containing 'Equals'), and 'Value' (containing 'Aa AppointmentRecordRes > Ap...'). A delete icon is visible next to the value field.

* Label

* API Name ⓘ

Get Consultant Rec

Get_Consultant_Rec

Description

Get Records of This Object

* Object

Consultant

Filter Consultant Records

Condition Requirements

All Conditions Are Met (AND) ▾

Field	Operator	Value
Name	Equals	Aa AppointmentRecordRes > Ap... X

+ Add Condition

Subtask 6: Create Appointment Record Using Create Records Element

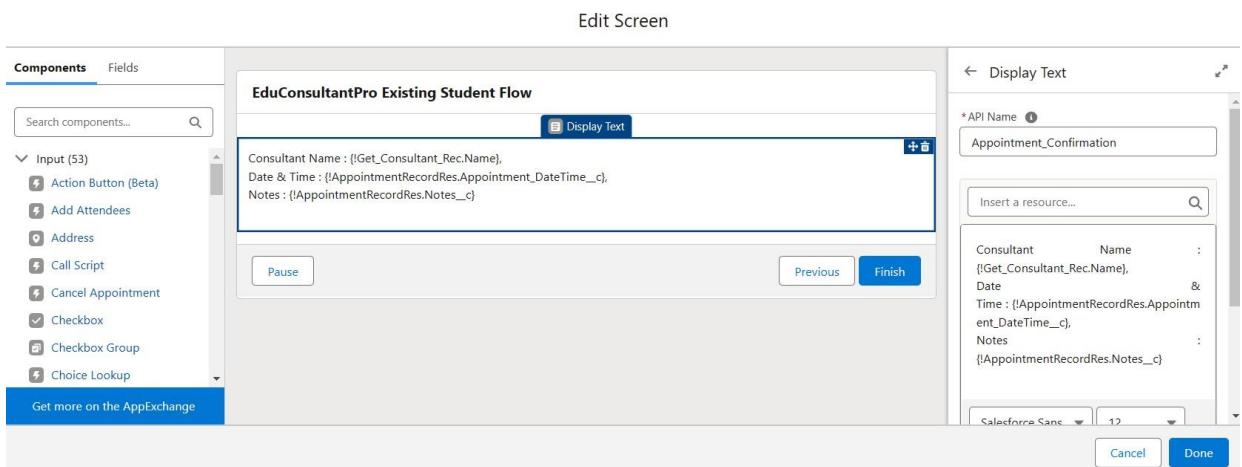
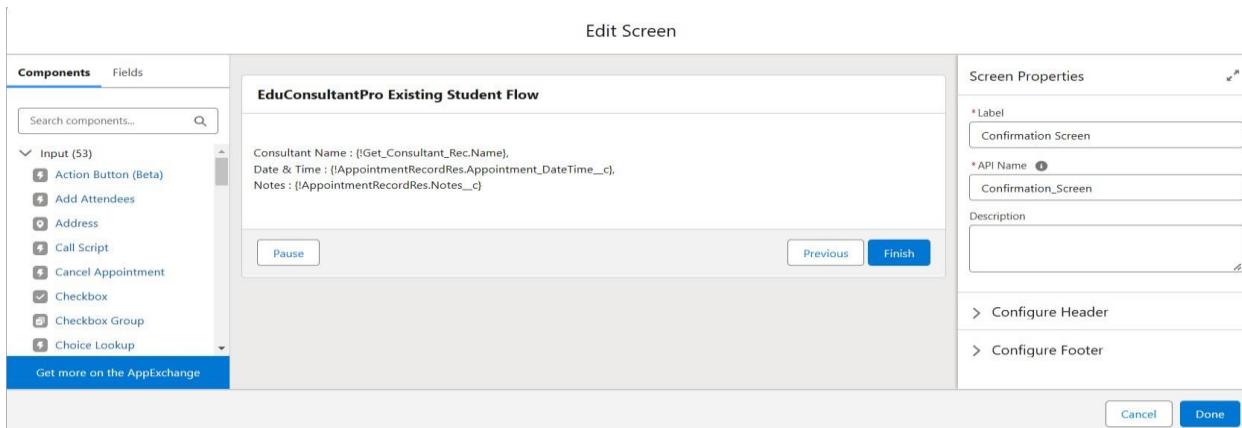
1. Add a **Create** element after the GET Consultant Rec element and label it as “**Create Appointment**”.
2. Configure the Create element:
 - **Select Object:** Appointment
 - **Field Values:**
 - **Appointment_DateTime_c:**
{ !AppointmentRecordRes.Appointment_DateTime_c }
 - **Consultant_c:** { !Get_Consultant_Rec.Id }
 - **Notes_c:** { !AppointmentRecordRes.Notes__c }
 - **PurposeTopic_c:** { !AppointmentRecordRes.PurposeTopic__c }
 - **Student_Name_c:** { !Get_Rec.Id }

The screenshot shows the 'Create Records' configuration screen. At the top, there are fields for 'Label' (containing 'Create Appointment') and 'API Name' (containing 'Create_Appointment'). Below these are sections for 'Description' (empty) and 'How to set record field values' (set to 'Manually'). Under the heading 'Create a Record of This Object', the 'Object' is selected as 'Appointment'. In the 'Set Field Values for the Appointment' section, three fields are configured: 'Appointment_DateTime_c' with a value derived from 'AppointmentRecordRes > Appointment Date/...', 'Consultant_c' with a value derived from 'Consultant from Get_Consultant_Rec > Record ...', and 'Notes__c' with a value derived from '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.

Create a Case

*Label
Create Student Case

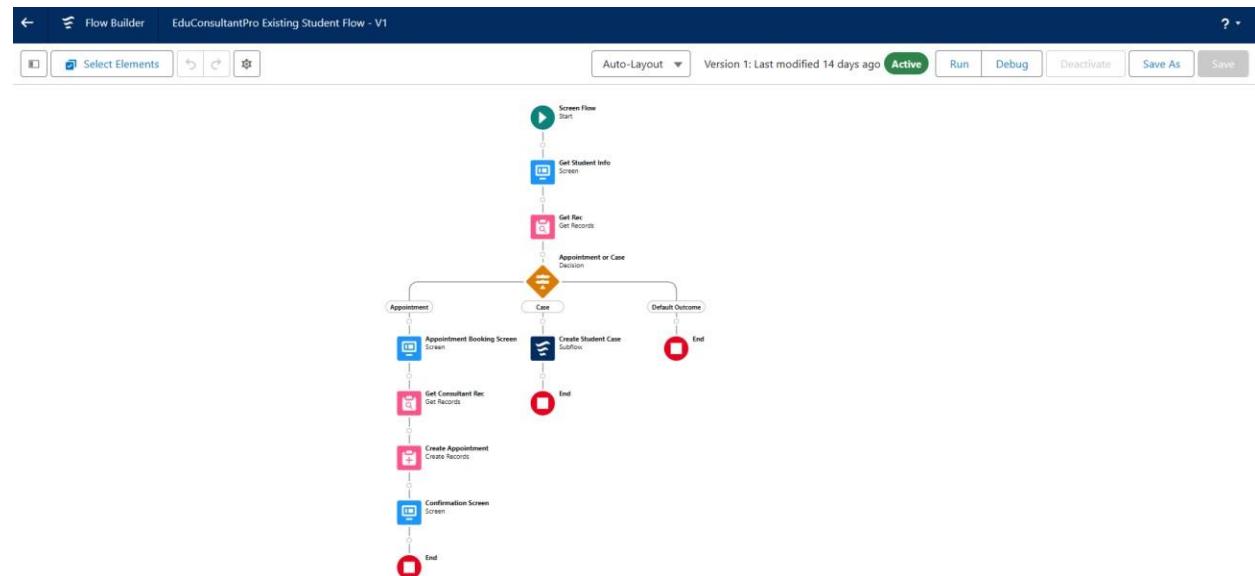
*API Name ⓘ
Create_Student_Case

Description

Referenced Flow

Create a Case
setup_service_experience_Create_Case

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 Welcome Screen 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.

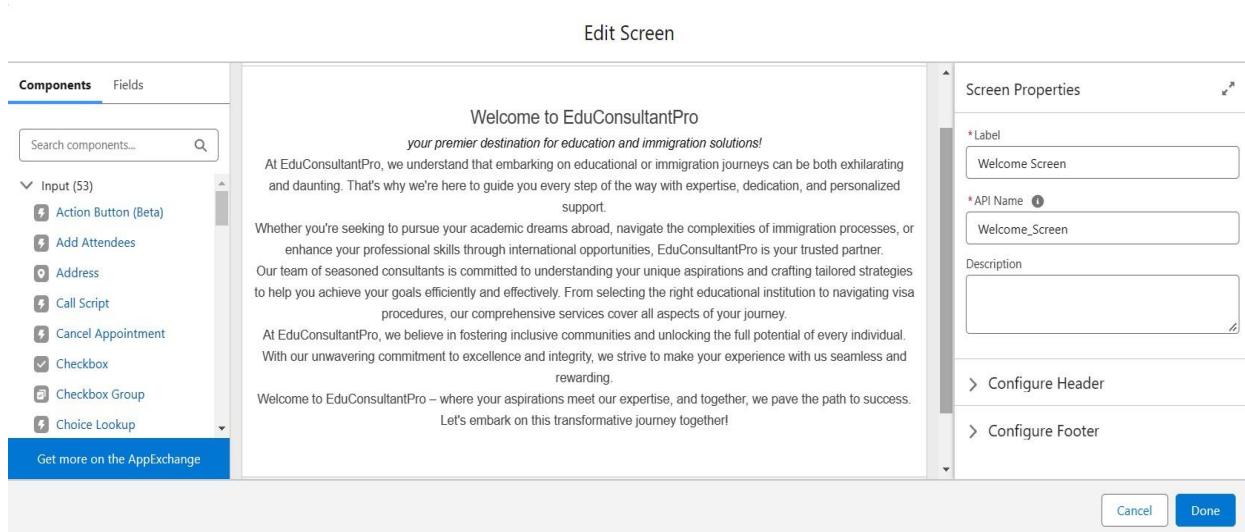
Whether you're seeking to pursue your academic dreams abroad, navigate the complexities of immigration processes, or enhance your professional skills through international opportunities, EduConsultantPro is your trusted partner.

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.

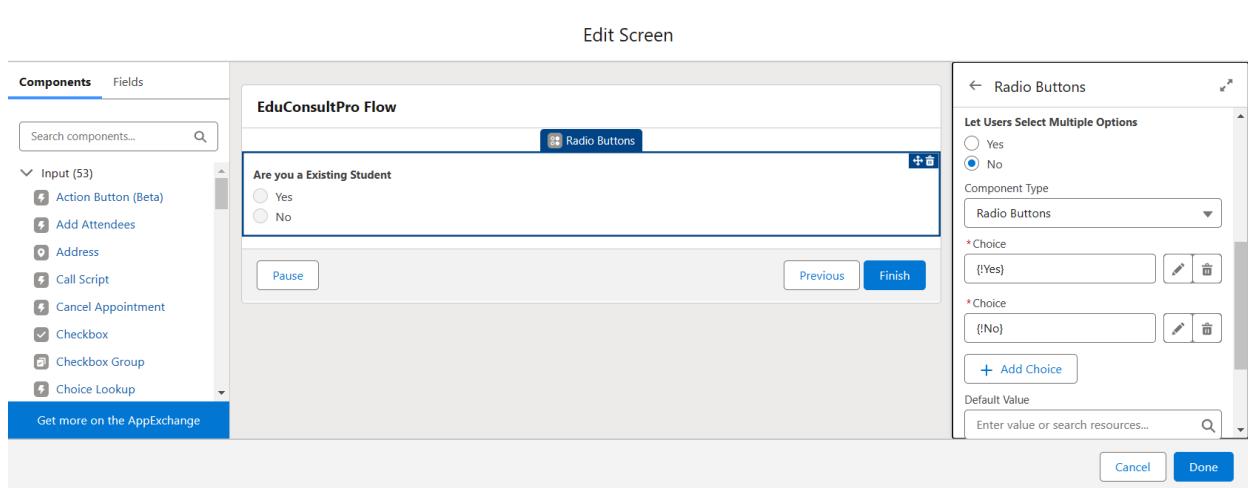
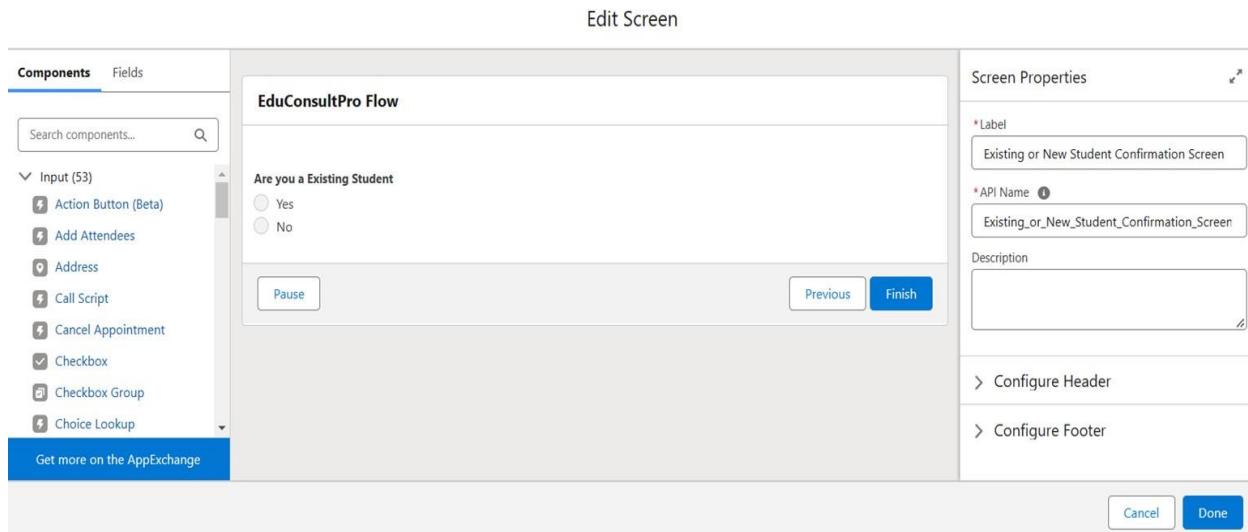
Welcome to EduConsultantPro – where your aspirations meet our expertise, and together, we pave the path to success. Let's embark on this transformative journey together!”

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 Decision Element

1. Add a **Decision** element after the Existing or New Student Confirmation Screen element and label it as “**Decision 1**”.
2. Create an outcome:
 - **Label:** If Existing Student
 - **Resource:** {!Are_you_a_Existing_Student}
 - **Operator:** Equals
 - **Value:** {!Yes}

3. Click the “+” icon to add more outcomes for “No” and other cases as required.

The screenshot shows a "Decision" configuration screen with two outcome entries:

- Outcome 1:**
 - Label:** Decision 1
 - API Name:** Decision_1
 - Description:** (Empty)
- Outcome 2:**
 - Label:** If Existing Student
 - Outcome API Name:** If_Existing_Student
 - Condition Requirements to Execute Outcome:** All Conditions Are Met (AND)
 - Resource:** Aa ... > Are_you_a_Existing_Student
 - Operator:** Equals
 - Value:** Aa Yes
- Outcome 3:**
 - Label:** If not a existing user
 - Outcome API Name:** If_Not_a_Existing_Student
 - Condition Requirements to Execute Outcome:** All Conditions Are Met (AND)
 - Resource:** Aa ... > Are_you_a_Existing_Student
 - Operator:** Equals
 - Value:** Aa No

OUTCOME ORDER: +

OUTCOME DETAILS:

Delete 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 Existing Student Flow**”.
3. Label it as “**Existing Student Flow**”.
4. Click **Done**.

The screenshot shows the configuration interface for a subflow. At the top, there's a title bar with the flow name "EduConsultantPro Existing Student Flow" and a close button. Below the title, there are three input fields: "Label" containing "Existing Student Flow", "API Name" containing "Existing_Student_Flow", and a "Description" field which is empty. Under the "Referenced Flow" section, there's a dropdown menu showing the selected flow: "EduConsultantPro Existing Student Flow" and its API name "EduConsultantPro_Existing_Student_Flow". A note below the dropdown explains how values from the parent flow can be used to set inputs for the subflow, mentioning that by default, the parent flow stores all outputs and providing instructions for referencing outputs via API name or manually assigning variables.

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

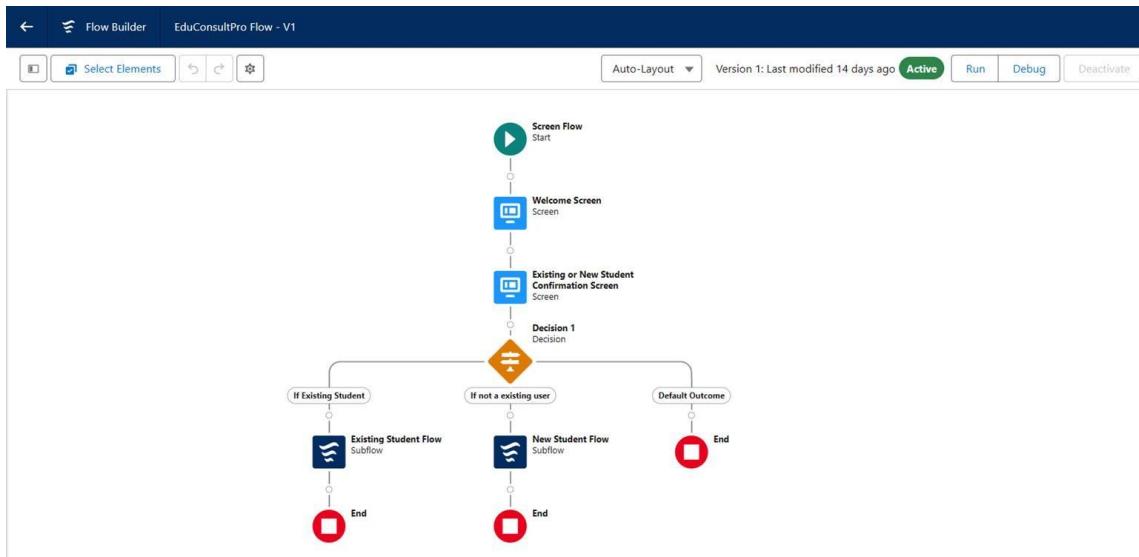
Description

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.



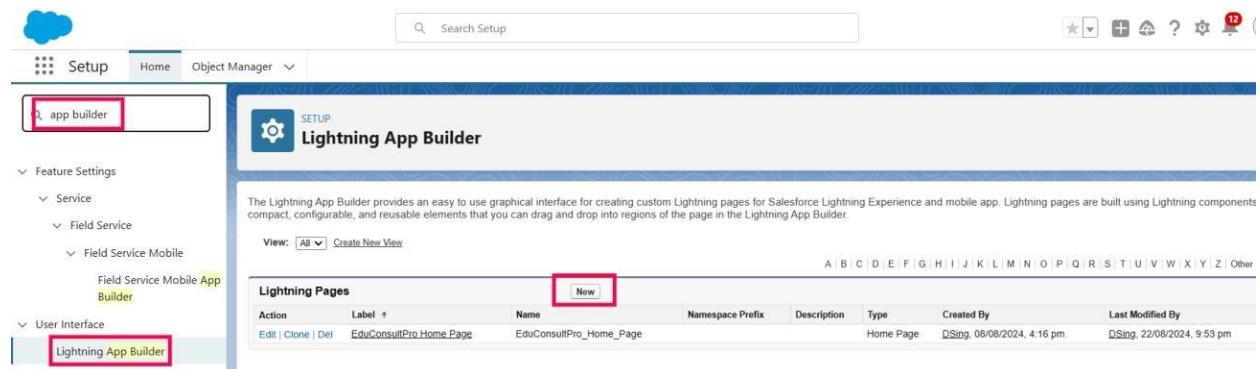
Task 10: Create A Lightning App Page

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:

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



2. Create a New Home Page:

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

Create a new Lightning page

App Page

Home Page

Record Page

Embedded Service Page

Voice Extension

Customize the Lightning Experience Home page.

My Quarterly Performance

Closed USD 235,000 Open (Commit) USD 320,000 Goal USD 420,000

AUG SEPT OCT

Einstein (10+)

7 Contact & Contact

Based on email activity, create contact and contact roles for this account

8 Contacts & Contact

12 Contact-Roles Created

Opportunity Boosting

Next

3. Configure the Home Page:

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

Create a new Lightning page

CHOOSE PAGE TEMPLATE **CLONE SALESFORCE DEFAULT PAGE**

STANDARD (7)	CUSTOM (0)
Header and Three Regions	
Home Template One Region	
Home Template One Region	
Home Template One Region	
Home page header two columns left side bar	
Standard Home Page	
CUSTOM (0)	

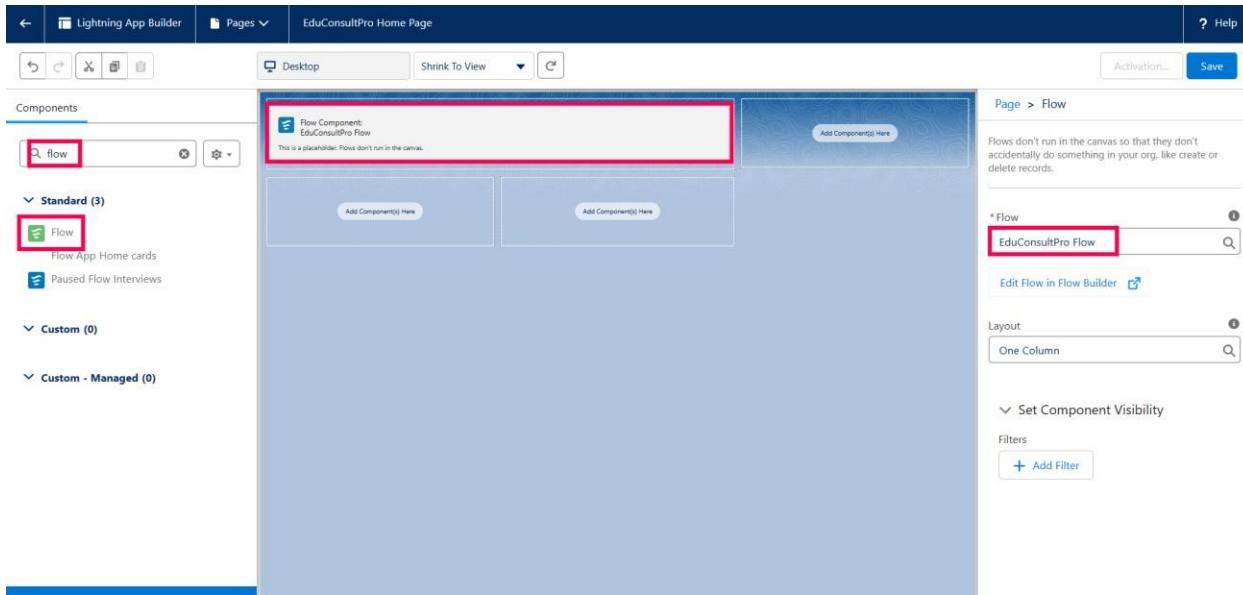
Lightning Experience Home page template with a header above two equal-width regions, and a full-height sidebar.

Supported form factors: desktop.

Back Done

4. Add the Flow Component:

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



5. Save and Activate the Page:

- Click **Save**.
- Click **Activate**.

6. Assign the Page to Apps and Profiles:

- Click **App and Profile**.
- Click **Assign to Apps and Profiles**.
- **Select the Sales App:** Choose **Sales** and click **Next**.
- **Assign to Profiles:** Scroll down and select **System Administrator** profile. Click **Next**.
- **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...	

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](#)

