

# **A CRM Application to Manage the Services offered by an Institution**

*By*

**Nithish Kumar M**

<https://www.salesforce.com/trailblazer/fctkh6g5br15fa0iv8>

# **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

<b>Project Abstract.....</b>	<b>2</b>
<b>Table of Contents .....</b>	<b>2</b>
<b>Introduction .....</b>	<b>6</b>
<b><i>TASK 1: Creating Developer Account .....</i></b>	<b>7</b>
<b><i>TASK 2: Account Activation .....</i></b>	<b>8</b>
<b><i>TASK 3: Create Objects From Spreadsheet .....</i></b>	<b>9</b>
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
<b><i>TASK 4: Create A ScreenFlow For Student Admission Application Process</i></b>	<b>19</b>
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

<b>Task 5: Create Users .....</b>	30
Subtask 1: Create a User .....	30 Subtask
2: Configure the User Settings .....	31
 <b>Task 6: Create an Approval Process for the Property Object .....</b>	32
Subtask 1: Create Email Templates .....	32
Subtask 2: Create An Approval Process .....	35
 <b>Task 7: Create A Record-Triggered Flow .....</b>	39
Subtask 1: Configure The Start Element .....	39
Subtask 2: Add An Action Element .....	40
 <b>Task 8: Create A ScreenFlow For Existing Students To Book An Appointment</b>	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
 <b>Task 9: Create A ScreenFlow To Combine All The Flows At One Place .....</b>	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
<b>Task 10: Create A Lightning App Page .....</b>	<b>56</b>

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.

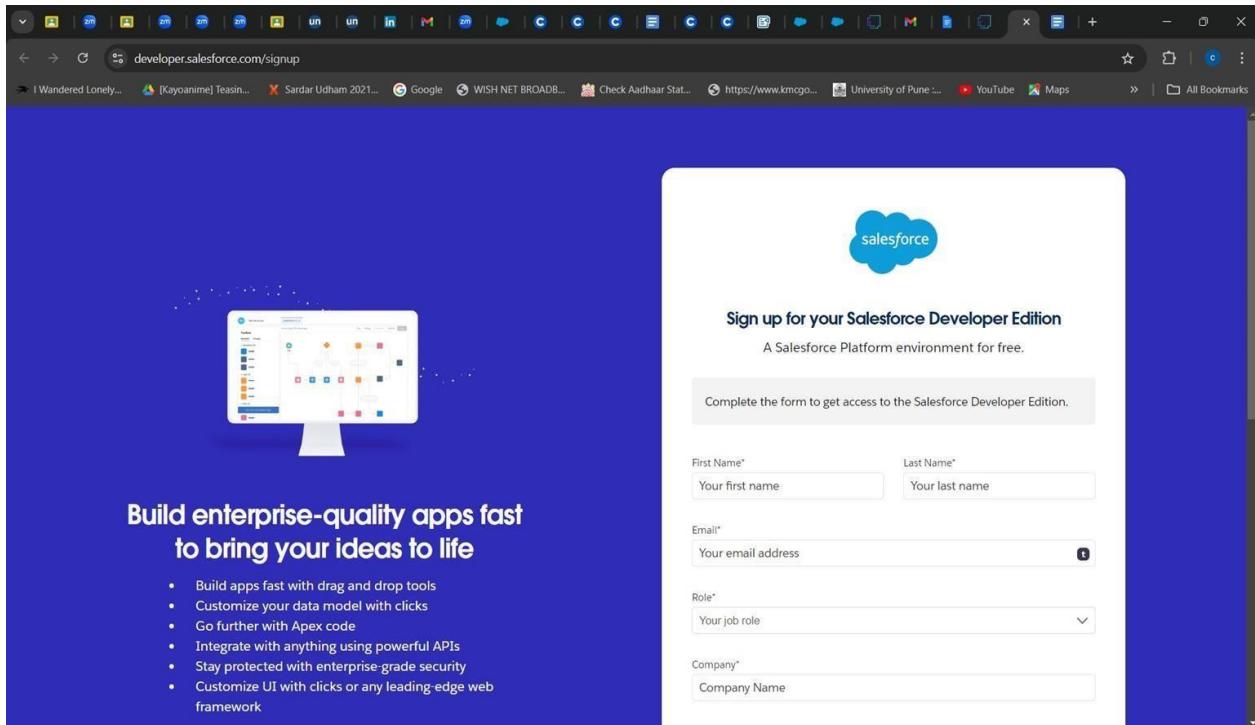
Another critical component was the implementation of an immigration case management system. Students could initiate immigration cases through various channels such as phone, email, or web. These cases were then logged into Salesforce, with immigration agents receiving immediate notifications to take 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

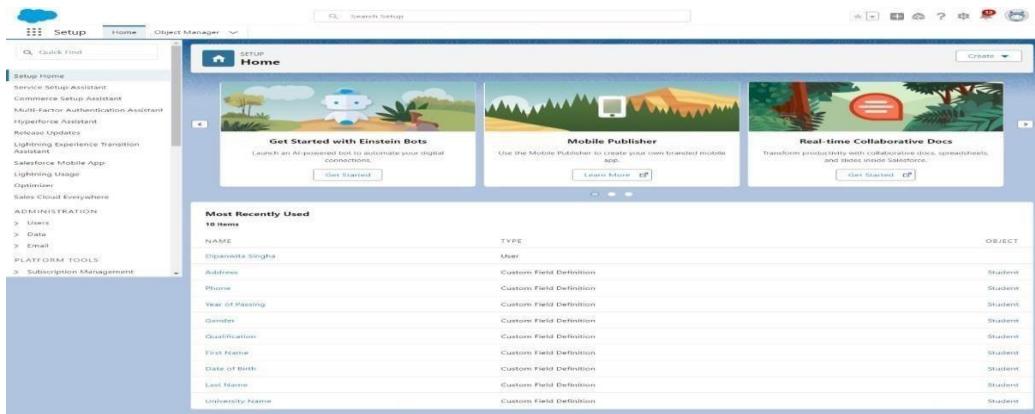
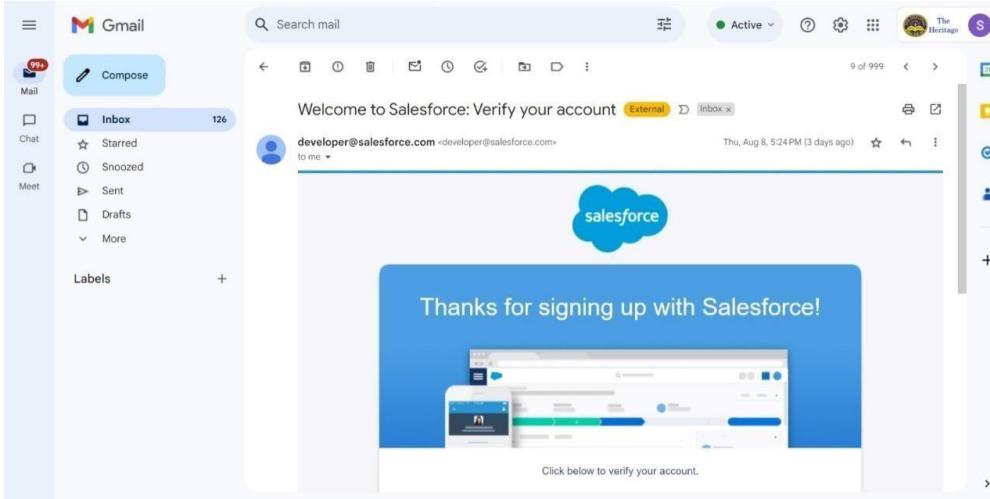
1. Go to <https://developer.salesforce.com/signup>
2. Fill up form with personal information



## 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



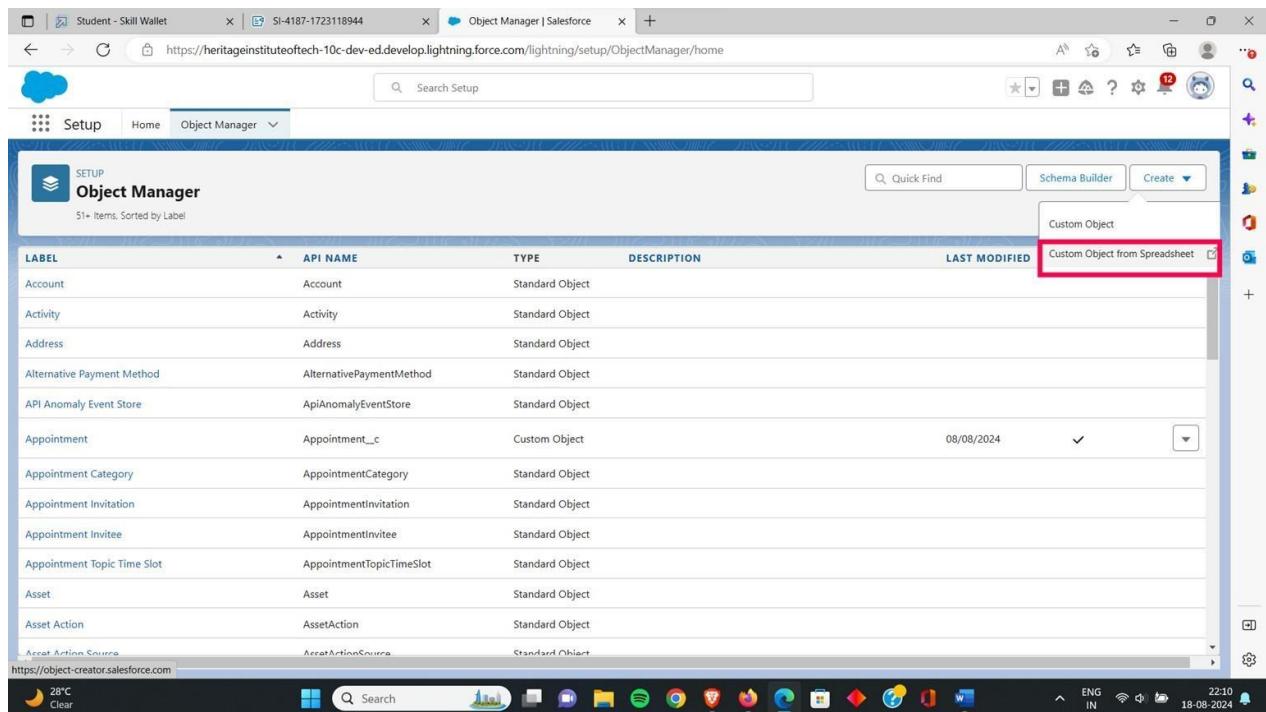
## 3: Create Objects from Spreadsheet

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

### *Subtask 1: Create Course Object*

# TASK

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 tabs for 'SETUP', 'Home', and 'Object Manager'. Below the tabs is a search bar labeled 'Search Setup'. On the right side of the header, there are several icons, including a 'Custom Object' icon which is highlighted with a red box. The main area is titled 'Object Manager' and displays a table of objects. The columns are labeled 'LABEL', 'API NAME', 'TYPE', 'DESCRIPTION', and 'LAST MODIFIED'. A 'Quick Find' bar is located above the table, and there are buttons for 'Schema Builder' and 'Create'. The 'LAST MODIFIED' column shows the last update date for each object.

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		
Accept Action Source	AcceptActionSource	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	Import 2 rows of Data?	Record Name Field
<input type="radio"/> Enter manually	1	<input type="radio"/> No, skip import	<input type="radio"/> Let Salesforce Create a Default
<input checked="" type="radio"/> Detect from row			

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	<input type="text" value="Student"/>
* Plural Label	<input type="text" value="Student"/>
* API Name	<input type="text" value="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			

## 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**

## **Subtask**

The screenshot shows the Salesforce Object Manager interface for the 'Appointment' object. In the left sidebar, 'Fields & Relationships' is selected and highlighted with a red box. On the right, there are four options: 'Auto Number', 'Formula', 'Roll-Up Summary', and 'Lookup Relationship'. The 'Lookup Relationship' option is selected and also highlighted with a red box. A detailed description of 'Lookup Relationship' follows:

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.

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. It is on 'Step 2: Choose the related object'. The title is 'Appointment New Relationship'. The sub-step title is 'Step 2. Choose the related object'. The instruction says 'Select the other object to which this object is related.' Below is a dropdown menu labeled 'Related To' with 'Student' selected. At the bottom right are 'Previous', 'Next', and 'Cancel' buttons.

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

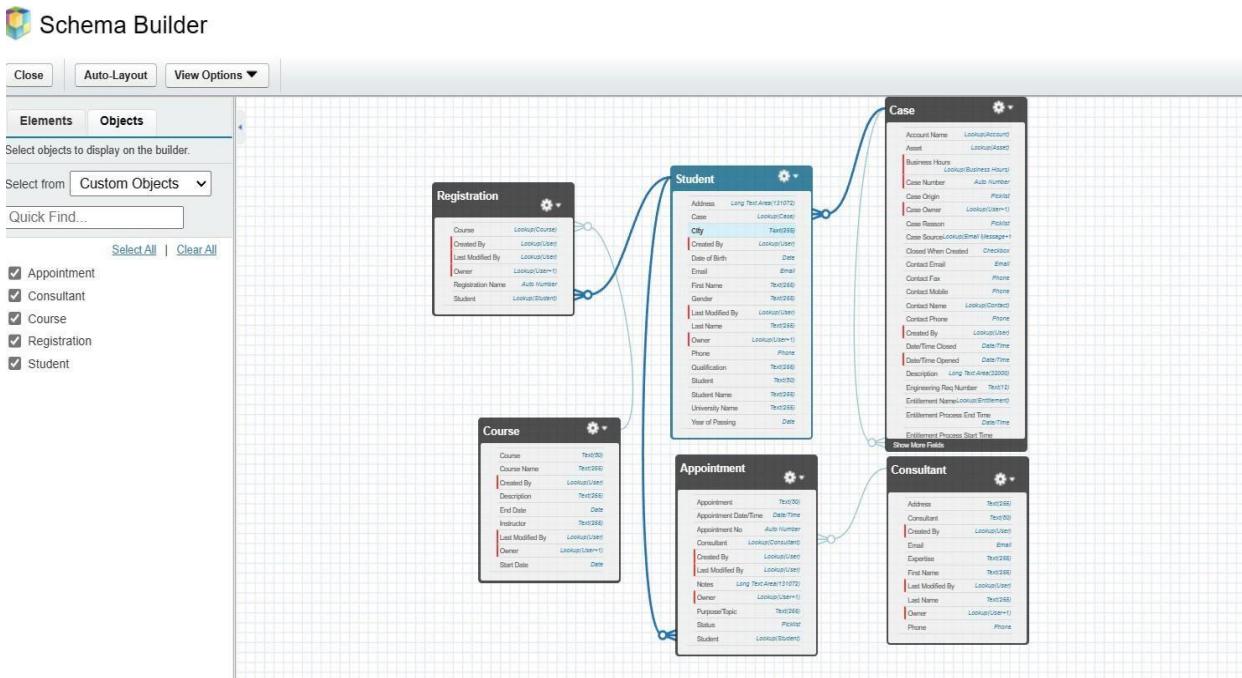
- **Appointment and Student Object**
- **Student and Case Object**

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.

Screenshot of the Salesforce Object Manager page. The page title is "Object Manager". Below it, a table lists various objects with columns for Label, API Name, Type, Description, Last Modified, and Deployed. The "Schema Builder" button is highlighted with a red box in the top right corner.

**4.** The data model will look like this:



### 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**

## Subtask

The screenshot shows the Salesforce Setup interface for the Case object. The left sidebar lists various setup categories like Details, Case Page Layouts, etc. The main area is titled 'Fields & Relationships' and contains a table of fields. The 'Type' field is highlighted with a red box.

Field Label	API Name	Type
Product	ProductId	Lookup(Product)
Service Contract	ServiceContractId	Lookup(Service Contract)
SLA Violation	SLAViolation__c	Picklist
Status	Status	Picklist
Stopped	IsStopped	Checkbox
Stopped Since	StopStartDate	Date/Time
Subject	Subject	Text(255)
<b>Type</b>	<b>Type</b>	<b>Picklist</b>
Web Company	SuppliedCompany	Text(80)
Web Email	SuppliedEmail	Email
Web Name	SuppliedName	Text(80)
Web Phone	SuppliedPhone	Text(40)

The screenshot shows the Salesforce Setup interface for the Case object. The left sidebar lists various setup categories. The main area is titled 'Case Type Picklist Values' and contains a table of picklist values. The 'Immigration' and 'Visa Application' values 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
<b>Edit   Del   Deactivate</b>	<b>Immigration</b>	<b>Immigration</b>	<input type="checkbox"/>	<b>Assigned dynamically</b>	<b>Dipanwita Singha 08/08/2024, 1:33 am</b>
<b>Edit   Del   Deactivate</b>	<b>Visa Application</b>	<b>Visa Application</b>	<input type="checkbox"/>	<b>Assigned dynamically</b>	<b>Dipanwita Singha 08/08/2024, 1:33 am</b>

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

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

SETUP > OBJECT MANAGER

**Case**

Details

**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		

Case Page Layouts

Case Close Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

SETUP > OBJECT MANAGER

**Case**

Details

**Fields & Relationships**

Case Page Layouts

Case Close Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

**Picklist Values Used**

Active picklist values: 6 (100 max)  
Inactive picklist values: 0

**Field Dependencies**

No dependencies defined.

**Validation Rules**

No validation rules defined.

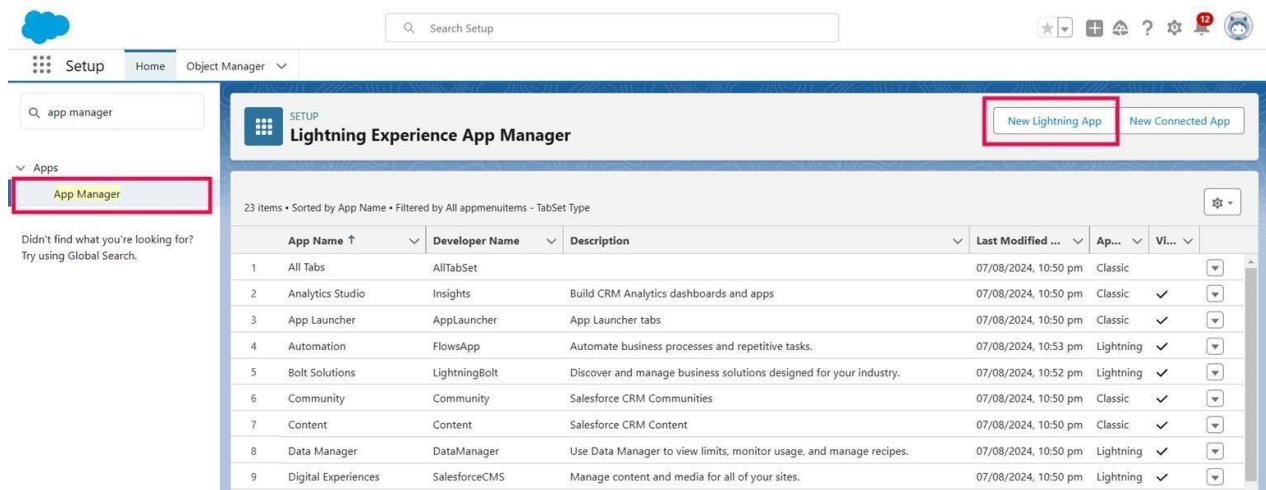
**Case Status Picklist Values**

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

## 4: Create a Lightning App

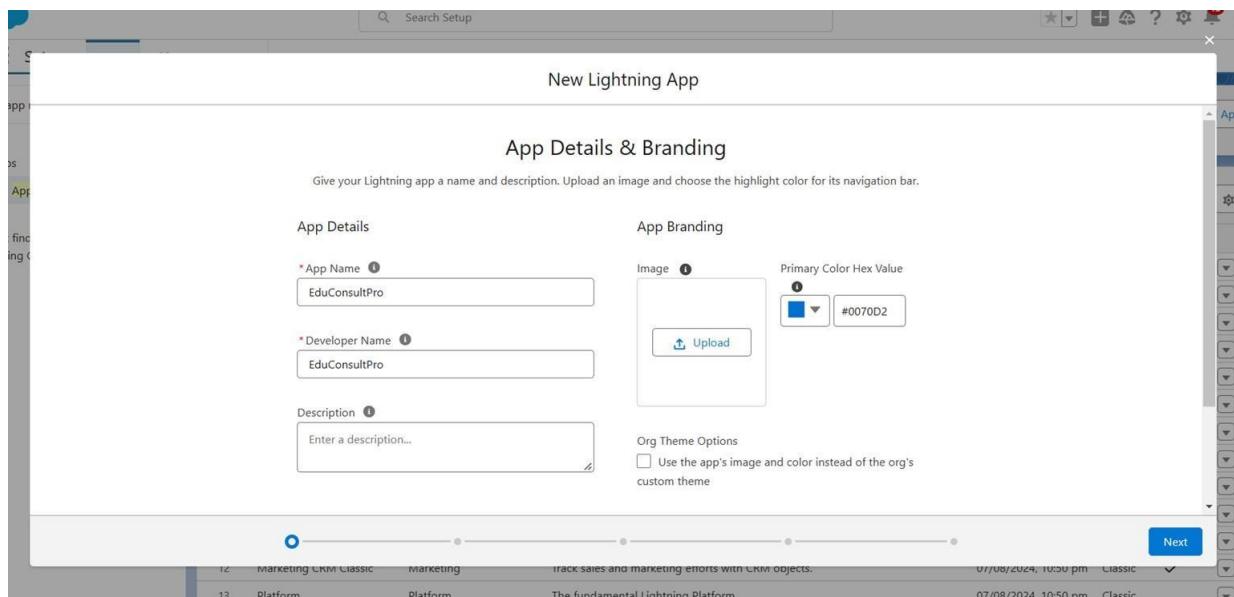
1. In **Setup**, search for **App Manager** in the Quick Find bar.
2. Click on **New Lightning App**.

## **Subtask**



The screenshot shows the 'Lightning Experience App Manager' interface. In the top right corner, there are two buttons: 'New Lightning App' and 'New Connected App'. The 'New Lightning App' button is highlighted with a red box.

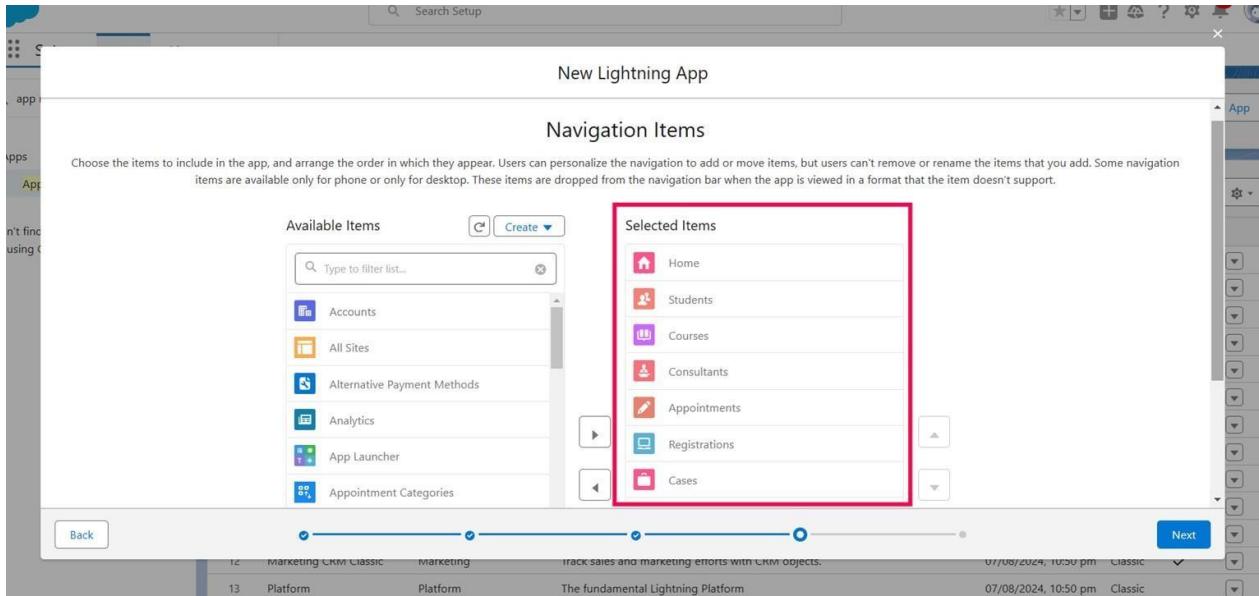
### **3. Name the app EduConsultPro.**



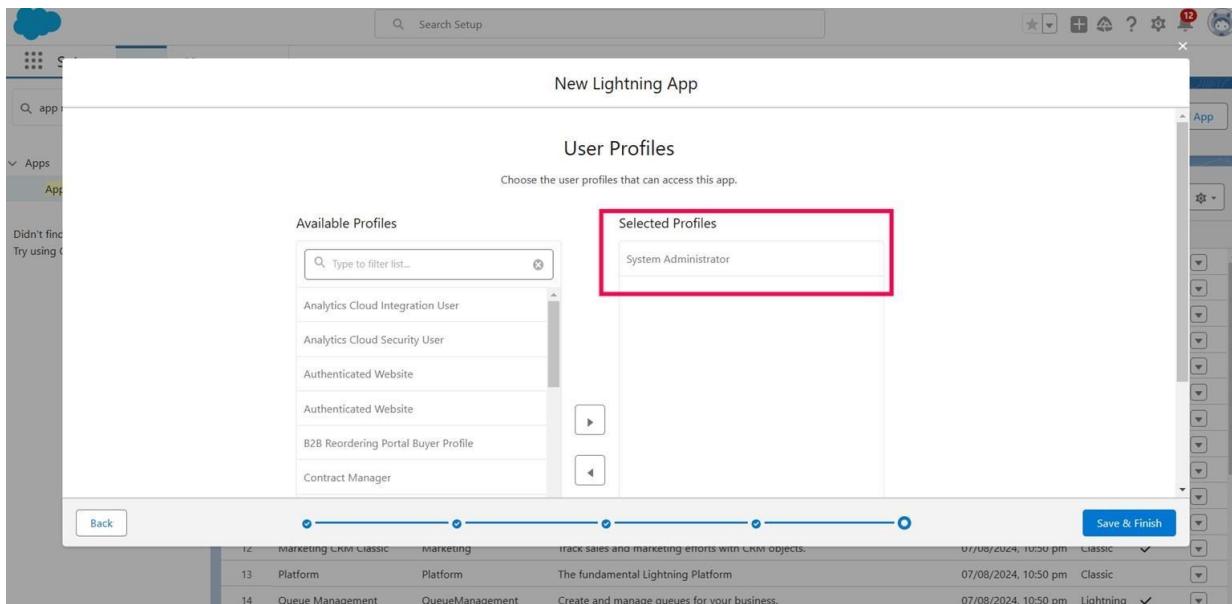
The screenshot shows the 'New Lightning App' configuration screen. In the 'App Details' section, the 'App Name' field is set to 'EduConsultPro'. In the 'App Branding' section, there is a placeholder image with an 'Upload' button and a primary color hex value of '#0070D2'. At the bottom, a progress bar shows steps 12 and 13 completed, and step 14 pending.

### **4. Add the following items from the Available Items list to the Selected Items list:**

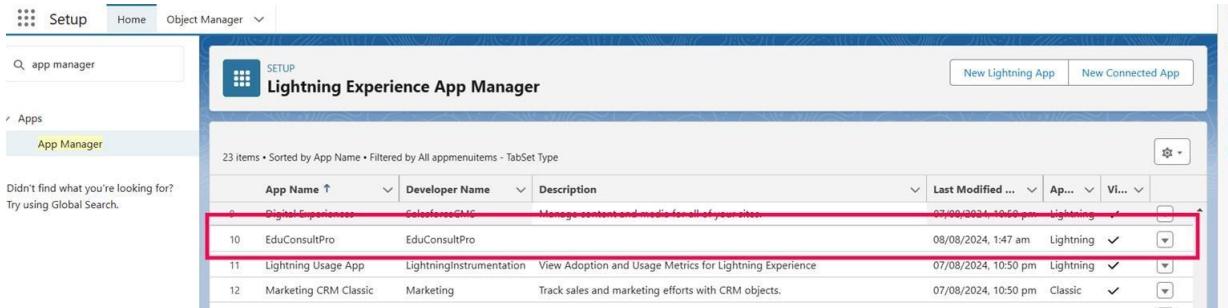
- o Home, Students, Courses, Consultants, Appointments, Registrations, Cases**



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. A message at the top says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Lightning Experience App Manager' and displays a table of apps. The table has columns: 'App Name ↑', 'Developer Name', 'Description', 'Last Modified ...', 'Ap...', and 'Vi...'. There are 23 items listed. The first item is 'Digital Experiences' by 'Salesforce CMS'. The second item, 'EduConsultPro' by 'EduConsultPro', is highlighted with a red border. The third item is 'Lightning Usage App' by 'LightningInstrumentation'. The fourth item is 'Marketing CRM Classic' by 'Marketing'. The 'EduConsultPro' row shows the last modification date as '08/08/2024, 1:47 am' and the type as 'Lightning'.

App Name ↑	Developer Name	Description	Last Modified ...	Ap...	Vi...
0 Digital Experiences	Salesforce CMS	Manage content and media for all of your sites.	07/08/2024, 10:58 pm	Lightning	<input checked="" type="checkbox"/>
10 EduConsultPro	EduConsultPro		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"/>

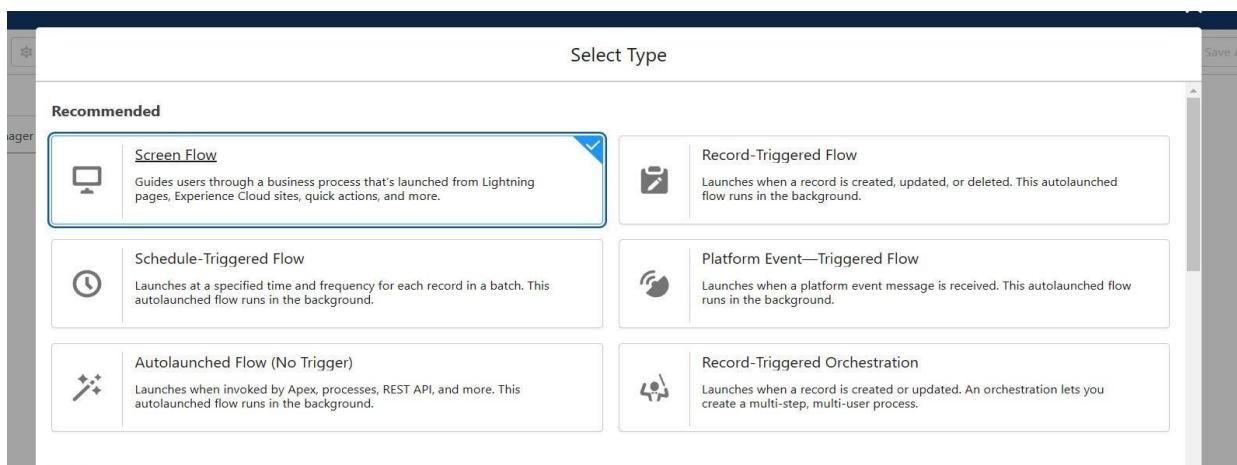
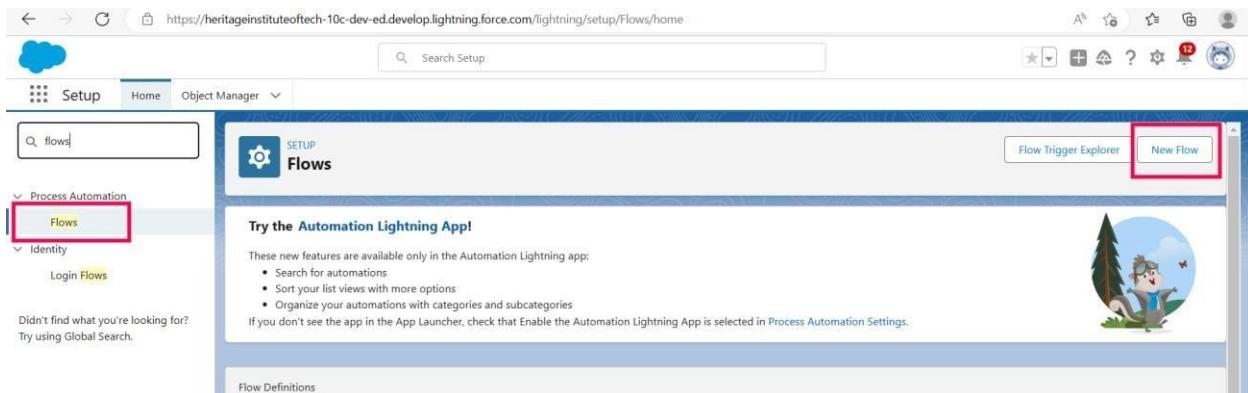
# TASK 4: Create A ScreenFlow For Student Admission Application Process

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

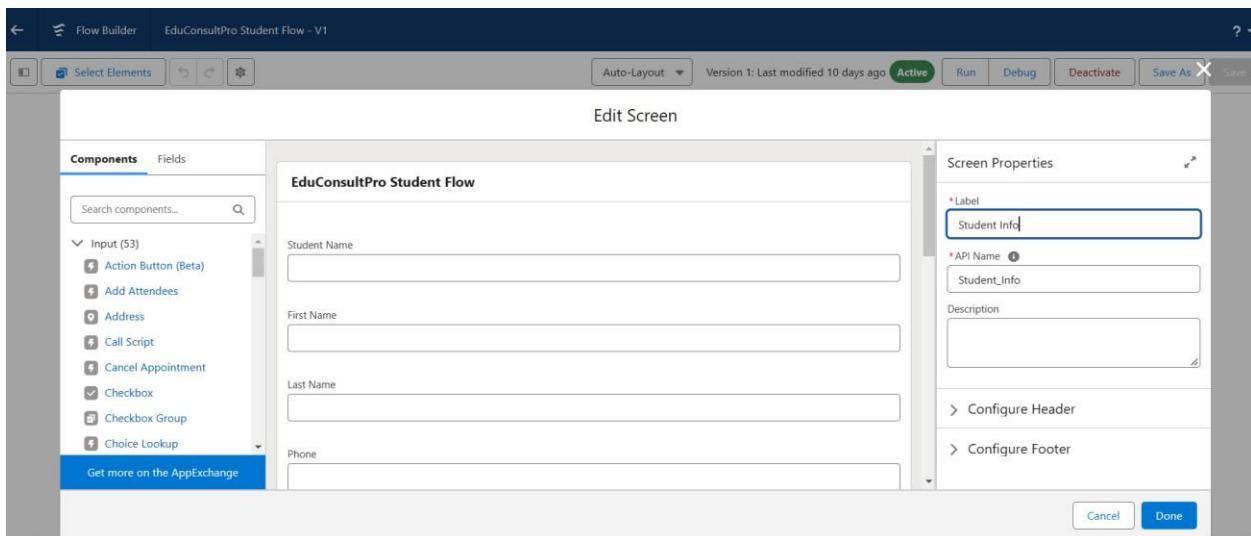
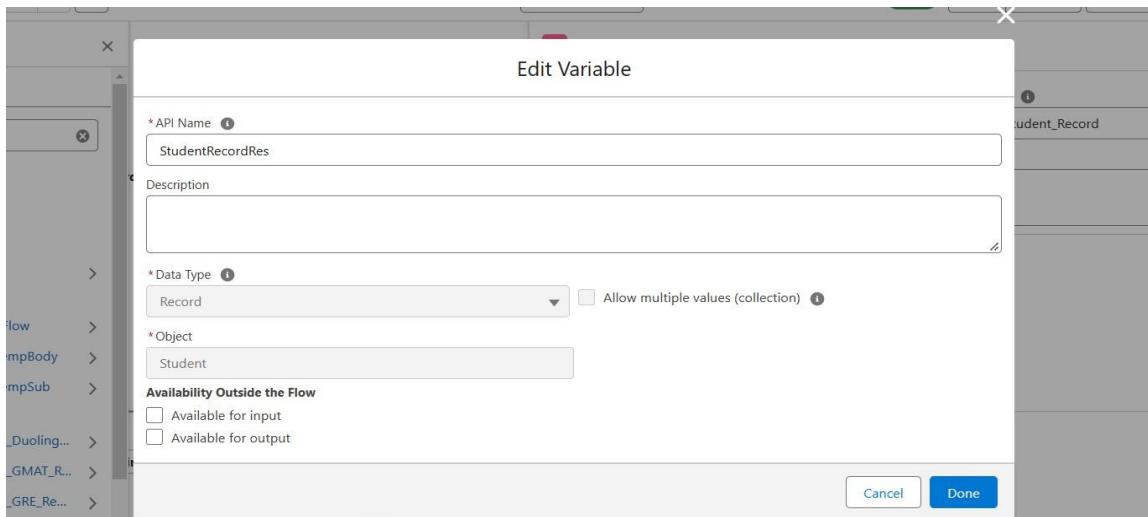
## ***Subtask 1: Add Screen Element***

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 interface for a 'Create Records' element. It includes fields for Label, API Name, Description, How to set record field values, How Many Records to Create, and Create a Record from These Values.

**Fields:**

- \*Label:** Create Student Record
- \*API Name:** Create\_Student\_Record
- Description:** (Empty text area)
- \*How to set record field values:** From a Record Variable
- How Many Records to Create:** One (radio button selected)
- Create a Record from These Values:** StudentRecordRes

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

## **Subtask Add**

### **3: 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      Add***

### ***4:      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.

## **Subtask Add**

Decision

\*Label: Selecting Course

\*API Name: Selecting\_Course

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
Selected IELTS	*Label: Selected IELTS *Outcome API Name: Selected_IELTS	
Selected GRE		
Selected GMAT		
Selected Duolingo		
Selected TOEFL		
Default Outcome		

All Conditions Are Met (AND)

Resource: Aa Course Screen > Select\_Course Operator: Equals Value: Aa IELTS

+ Add Condition

## **5: 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}

## Subtask Add

 Get Records X

* Label	* API Name <span style="color: #808080;">i</span>	
Get IELTS Rec	GetIELTSRec	
Description		
<div style="border: 1px solid #ccc; height: 40px; margin-top: 10px;"></div>		
Get Records of This Object		
* Object		
Course		
Filter Course Records		
Condition Requirements		
All Conditions Are Met (AND) ▼		
Field	Operator	Value
Name	Equals	Aa Select_Course X <span style="float: right;">Delete</span>
<span style="border: 1px solid #ccc; padding: 2px 10px; color: #0070C0;">+ Add Condition</span>		
Sort Course Records		
Sort Order		

## **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 fields for Label (Create IELTS Registration Rec), API Name (Create\_IELTS\_Registration\_Rec), Description, and a dropdown for How to set record field values (Manually). Below this, there's a section for creating a record of the 'Registration' object, and a table for setting field values with examples like 'Course\_c' and 'Student\_c'.

Field	Value
Course_c	← Aa Course from GetIELTSRec > Record ID X
Student_c	← Aa StudentRecordRes > Record ID X

+ 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 ⓘ

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”.

### *Subtask*

## 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!

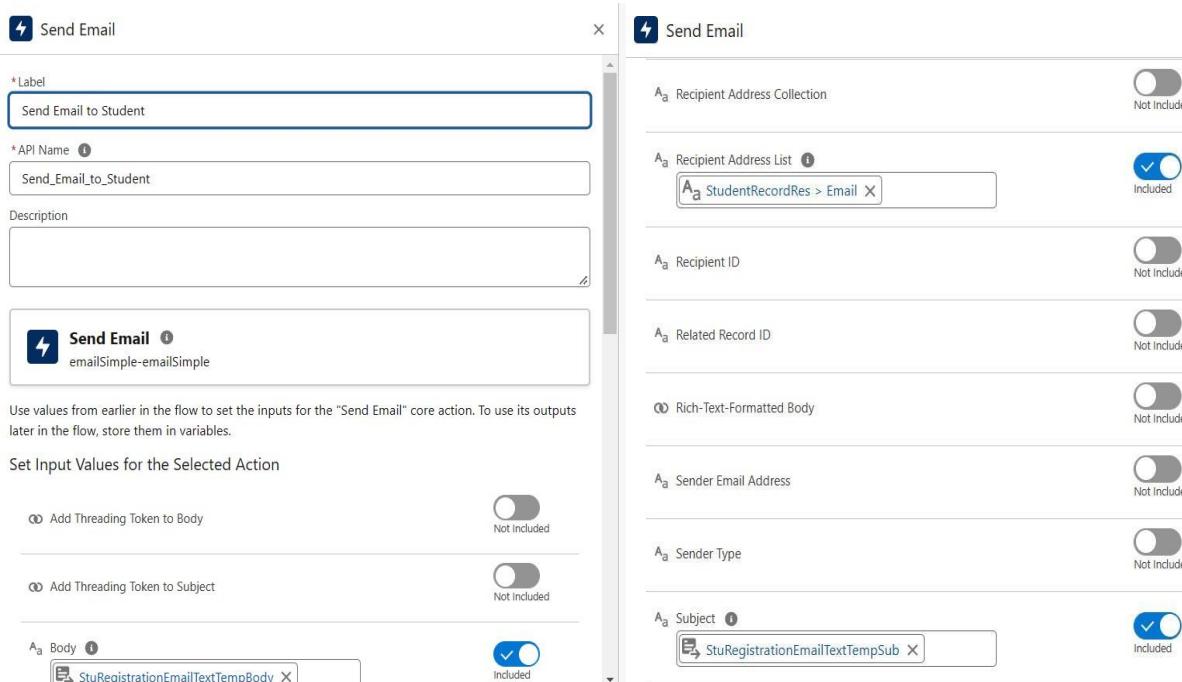
We are delighted to inform you that your registration on our platform has been successfully completed. You are now part of our esteemed community dedicated to empowering students like you to achieve their educational and immigration aspirations.

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}



## **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.

Edit Screen

**Components**

Search components...

- Input (53)
  - Action Button (Beta)
  - Add Attendees
  - Address
  - Call Script
  - Cancel Appointment
  - Checkbox
  - Checkbox Group
  - Choice Lookup

[Get more on the AppExchange](#)

**EduConsultPro Student Flow**

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.  
 Your Registration details have been sent through mail kindly check it once.  
 Thank you.

[Pause](#) [Previous](#) [Finish](#)

**Screen Properties**

\*Label: Success Screen

\*API Name: Success\_Screen

Description:

> Configure Header  
 > Configure Footer

[Cancel](#) [Done](#)

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**.

2. Enter the following details:

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

3. Complete all mandatory fields as required and save.

## ***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 'Users' setup page in Salesforce. At the top, there are fields for Street, City, Zip/Postal Code, State/Province, and Country (set to IN). Below that is a 'Single Sign On Information' section with a Federation ID field. Under 'Locale Settings', Time Zone is set to (GMT+05:30) India Standard Time (Asia/Kolkata), Locale to English (India), and Language to English. In the 'Approver Settings' section, the 'Delegated Approver' field is empty. The 'Manager' field contains 'Consultant' and is highlighted with a red box. Below it, 'Receive Approval Request Emails' is set to 'Only if I am an approver'. At the bottom, there are 'Save', 'Save & New', and 'Cancel' buttons.

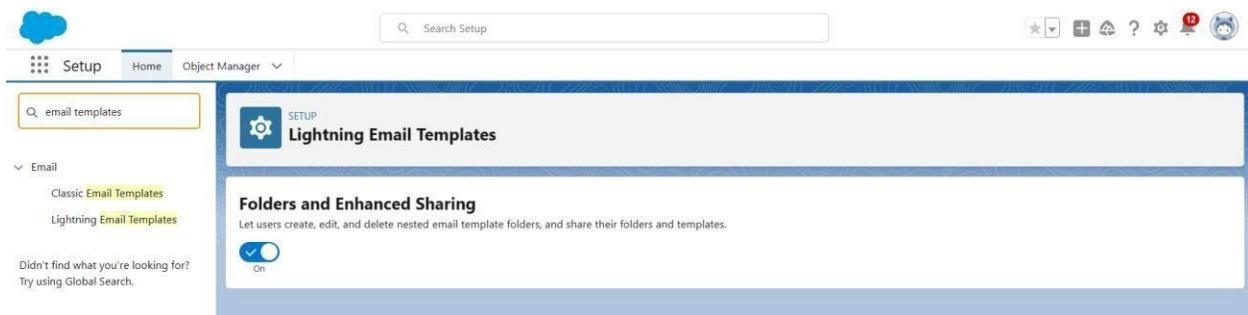
## Task 6: Create an Approval Process for the Property Object

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

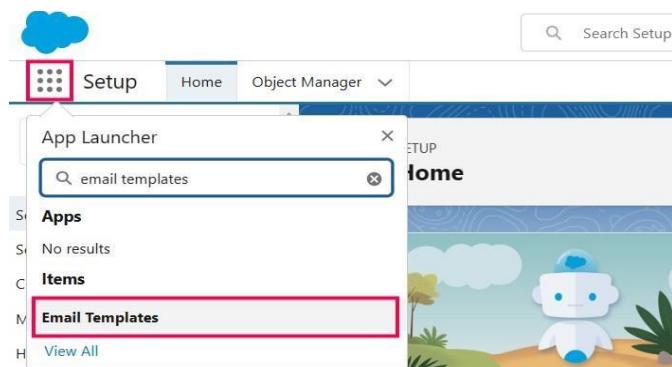
### *Subtask 1: Create Email Templates*

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**

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

**Message Content**

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

**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:

- **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

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

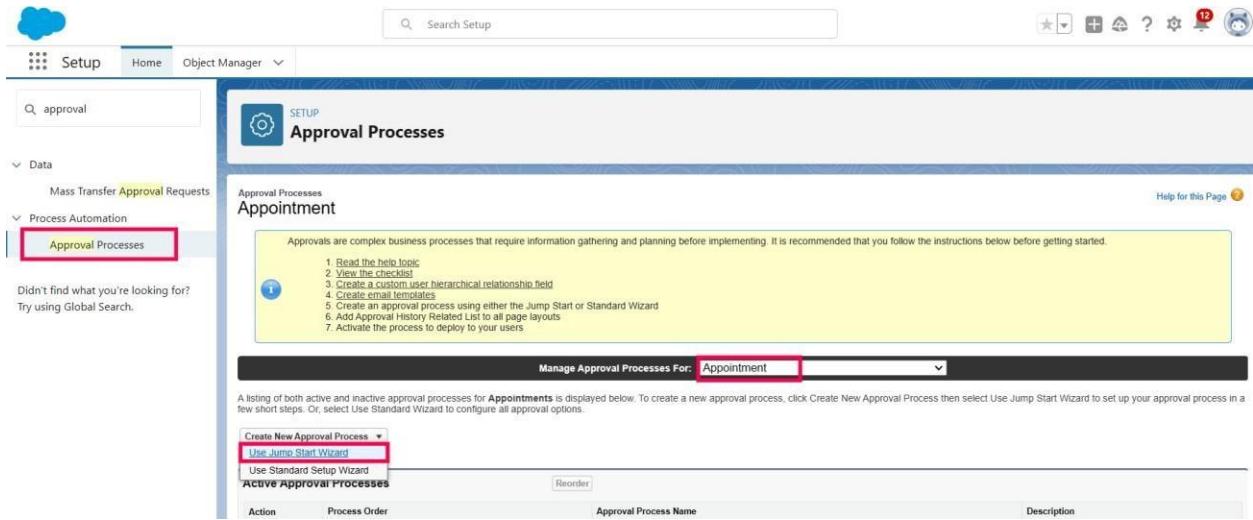
Email Templates						
Recent						
3 items						
EMAIL TEMPLATES	Email Template Name	Description	Folder	Last Modified By	Last Modified Date	
Recent	Approval Request Template		EduConsultantpro_Email	Dipanwita Singha	8/8/2024, 2:22 pm	
Created by Me	Submission Template		EduConsultantpro_Email	Dipanwita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

## 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**.

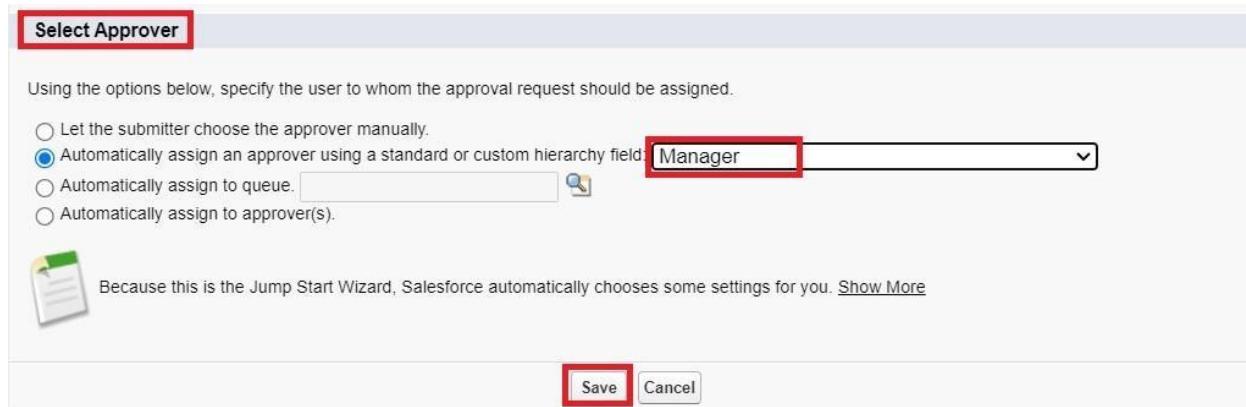
3. Click **Create New Approval Process** and choose **Use Jump Start Wizard**.



The screenshot shows the Salesforce Setup interface. In the top navigation bar, 'Setup' is selected. The main search bar contains 'approval'. Under the 'Data' category, 'Mass Transfer Approval Requests' is listed. Under 'Process Automation', 'Approval Processes' is highlighted with a red box. Below the navigation, a sidebar lists 'Data' and 'Process Automation'. A message says 'Didnt find what you're looking for? Try using Global Search.' The main content area is titled 'Approval Processes' and 'Appointment'. It includes a help topic about approvals and a step-by-step guide for creating them. A dropdown menu 'Manage Approval Processes For: Appointment' is open, with 'Use Jump Start Wizard' highlighted with a red box. Other options in the dropdown are 'Create New Approval Process', 'Use Standard Setup Wizard', and 'Active Approval Processes'. A table below shows columns for Action, Process Order, Approval Process Name, and Description.

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. At the top, a red box highlights the 'Select Approver' button. The main area contains instructions: 'Using the options below, specify the user to whom the approval request should be assigned.' Below this are four radio button options:

- Let the submitter choose the approver manually.
- Automatically assign an approver using a standard or custom hierarchy field. This option is selected, and its value 'Manager' is shown in a dropdown menu with a red box around it.
- Automatically assign to queue. A search input field is next to it.
- Automatically assign to approver(s).

A note below states: 'Because this is the Jump Start Wizard, Salesforce automatically chooses some settings for you. [Show More](#)'. At the bottom, there are 'Save' and 'Cancel' buttons, with 'Save' highlighted with 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**.

## 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	<input type="text" value="Manager"/> 
Use Approver Field of Appointment Owner	<input type="checkbox"/>

#### 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**

Save Save & New Cancel

**Edit Email Alert**

Description	Submission Email Alert								
Unique Name	Submission_Email_Alert								
Object	Appointment								
Email Template	Submission Template								
Protected Component	<input type="checkbox"/>								
Recipient Type	Search: User <input type="button" value="▼"/> for: <input type="text"/> <input type="button" value="Find"/>								
Recipients	<table border="1"> <tr> <th>Available Recipients</th> <th>Selected Recipients</th> </tr> <tr> <td>User: Consultant User: Integration User User: Security User</td> <td>User: Dipanwita Singha</td> </tr> <tr> <td>Add <input type="button" value="▶"/></td> <td></td> </tr> <tr> <td>Remove <input type="button" value="◀"/></td> <td></td> </tr> </table>	Available Recipients	Selected Recipients	User: Consultant User: Integration User User: Security User	User: Dipanwita Singha	Add <input type="button" value="▶"/>		Remove <input type="button" value="◀"/>	
Available Recipients	Selected Recipients								
User: Consultant User: Integration User User: Security User	User: Dipanwita Singha								
Add <input type="button" value="▶"/>									
Remove <input type="button" value="◀"/>									

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	<u>Approval Email Alert</u>
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	<u>Rejected</u>
Edit   Remove	Email Alert	<u>Rejection Email Alert</u>

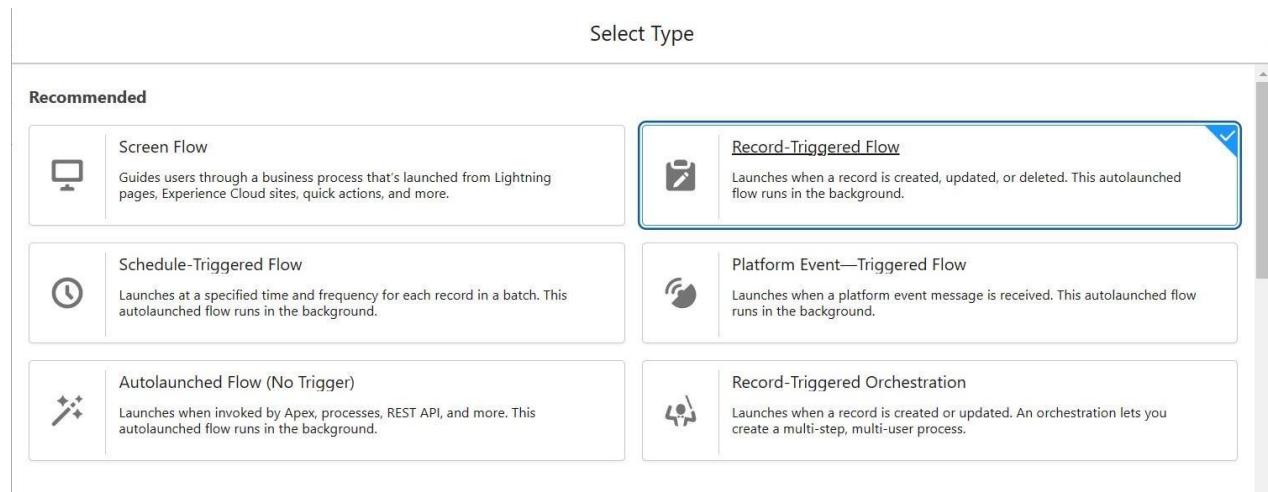
# Task 7: Create A Record-Triggered Flow

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.

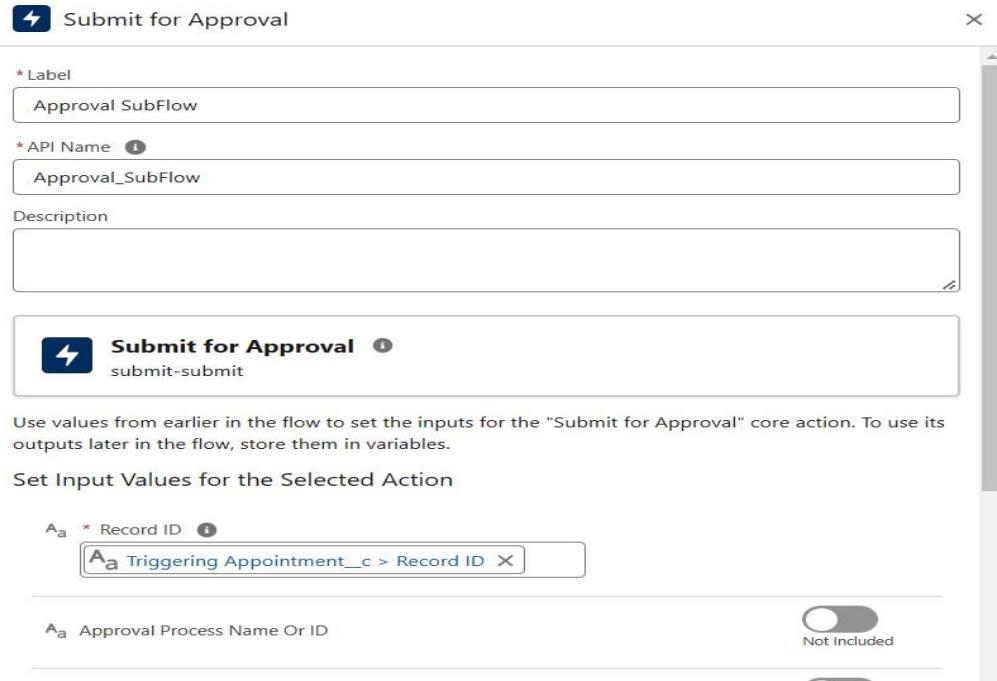
Condition Requirements

None

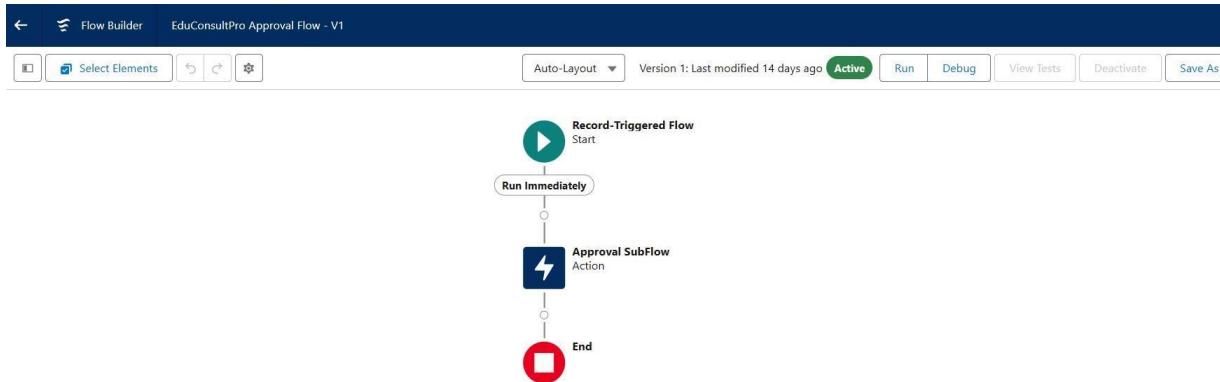
## ***Subtask 2: Add An Action Element***

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}**.



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

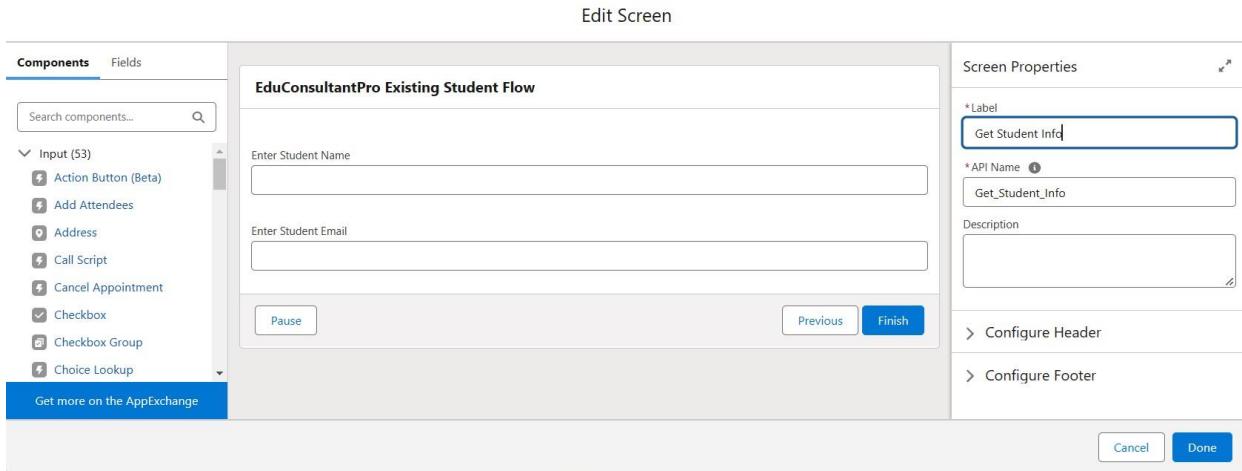


## Task 8: Create A ScreenFlow For Existing Students To Book An Appointment

### *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*

1.

## 2: *Get Record*

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

**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 X
AND Email_c	Equals	Aa Enter_Student_Email X

+ Add Condition

***Subtask      Add***

1.

- **Value:** {!Enter\_Student\_Email}

## ***Subtask      Add***

1.

### ***3:      a Decision Element***

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.

## **Subtask      Add**

1.

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 these is a 'Description' section with a large text input field. A note below says 'Outcomes' with the 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, a sidebar lists 'OUTCOME ORDER' with options for 'Appointment' and 'Case'. The main area is titled 'OUTCOME DETAILS' and contains a table for outcomes. The first row shows 'Appointment' with 'Label' and 'Outcome API Name' both set to 'Appointment'. Below this is a 'Condition Requirements to Execute Outcome' section with a dropdown set to 'All Conditions Are Met (AND)'. Underneath are fields for 'Resource' (A\_a How\_may\_I\_Help\_you), 'Operator' (Equals), and 'Value' (A\_a Book\_an\_Appointment). A 'Delete Outcome' button is also visible.

### **4:      Screen Element**

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.

## **Subtask      Add**

1.

4. Click **Done**.

Edit Screen

EduConsultantPro Existing Student Flow

Appointment Date/Time

Date Time

Purpose/Topic

Notes

Screen Properties

\* Label: Appointment Booking Screen

\* API Name: Appointment\_Booking\_Screen

Description:

Configure Header

Configure Footer

Cancel Done

## ***Subtask      Add***

1.

2.

### ***5:      GET Record Element***

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

- **Select Object:** Consultant
- **Condition Requirement:** All Conditions are Met (AND)
- **Field:** Name
  - **Operator:** Equals
  - **Value:** {!AppointmentRecordRes.Consultant\_Name\_c}

## **Subtask**

1.

2.

The screenshot shows the configuration of 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 sections for 'Description' (empty) and 'Get Records of This Object'. Under 'Object', 'Consultant' is selected. The next section, 'Filter Consultant Records', contains a 'Condition Requirements' dropdown set to 'All Conditions Are Met (AND)'. A single condition is defined with 'Field' 'Name', 'Operator' 'Equals', and 'Value' 'AppointmentRecordRes > Ap...'. A blue '+ Add Condition' button is visible at the bottom left of this section.

## **6: Create Appointment Record Using Create Records Element**

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

Configure the Create element:

- **Select Object:** Appointment

- **Field Values:**

- **Appointment\_DateTime\_c:**

- $\{!AppointmentRecordRes.Appointment\_DateTime\_c\}$

## **Subtask**

- 1.
2.
  - o **Consultant\_c:** {!Get\_Consultant\_Rec.Id}
  - o **Notes\_c:** {!AppointmentRecordRes.Notes\_c}
  - o **PurposeTopic\_c:** {!AppointmentRecordRes.PurposeTopic\_c}
  - o **Student\_Name\_c:** {!Get\_Rec.Id}

**Create Records**

* Label	* API Name <small>(i)</small>				
Create Appointment	Create_Appointment				
Description					
* How to set record field values					
Manually					
Create a Record of This Object					
* Object	Appointment				
Set Field Values for the Appointment					
Field	Appointment_DateTime_c	Value	AppointmentRecordRes > Appointment Date/...	X	trash
Field	Consultant_c	Value	Aa Consultant from Get_Consultant_Rec > Record ...	X	trash
Field	Notes_c	Value	Aa AppointmentRecordRes > Notes	X	trash

## **7: Add Screen Element**

Add a **Screen** element after the Create Appointment element, and label it as

## Subtask

1.

2.

### “Confirmation Screen”.

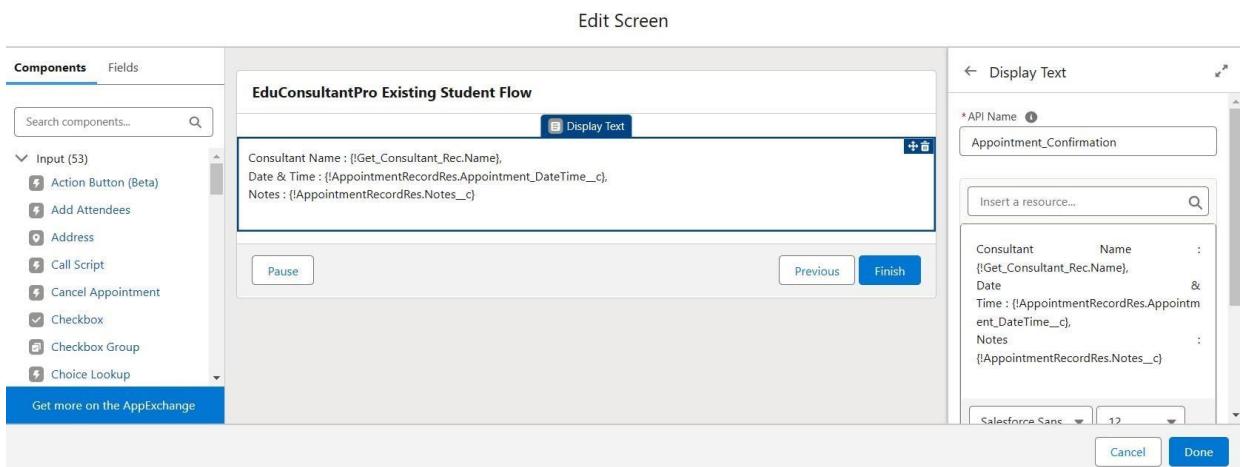
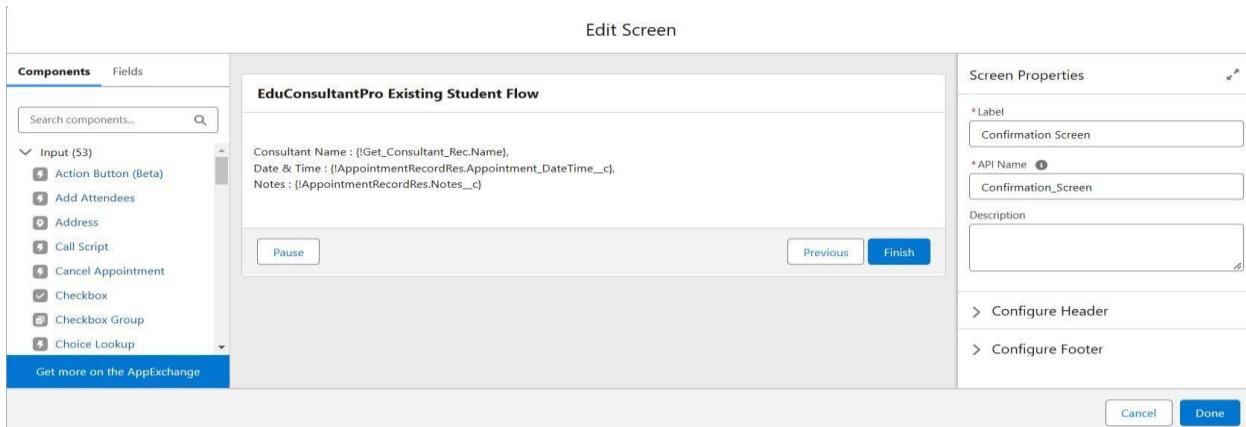
From the left side panel, search for the **Display Text** component and drag it to the main panel, label it as “**Appointment\_Confirmation**”.

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***

1.

2.

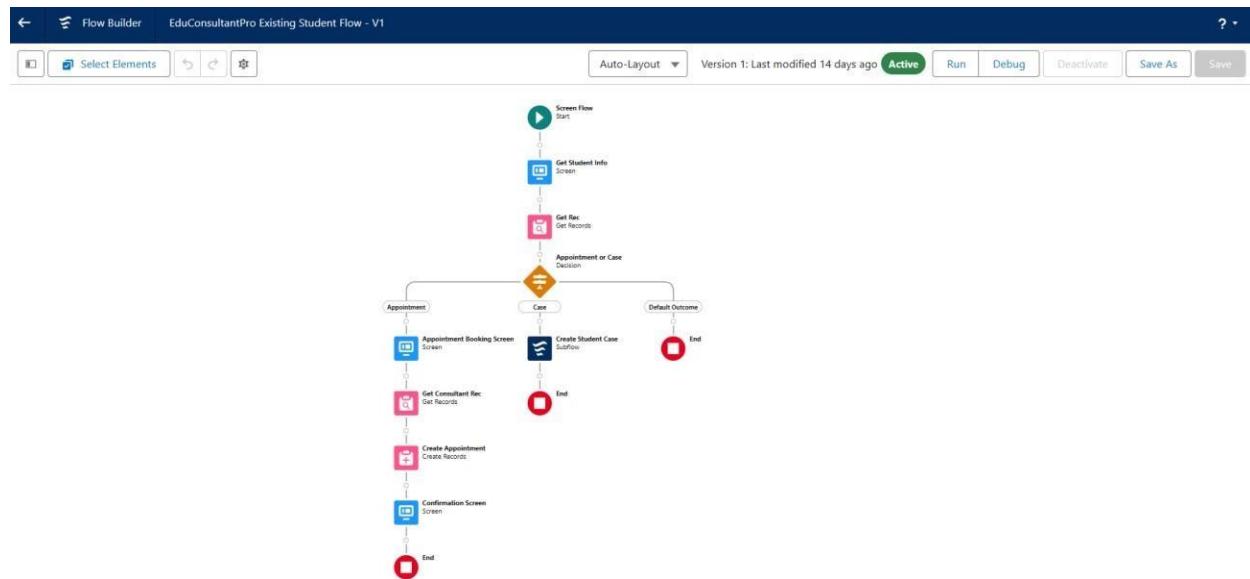
## **Subtask**

### **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 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.

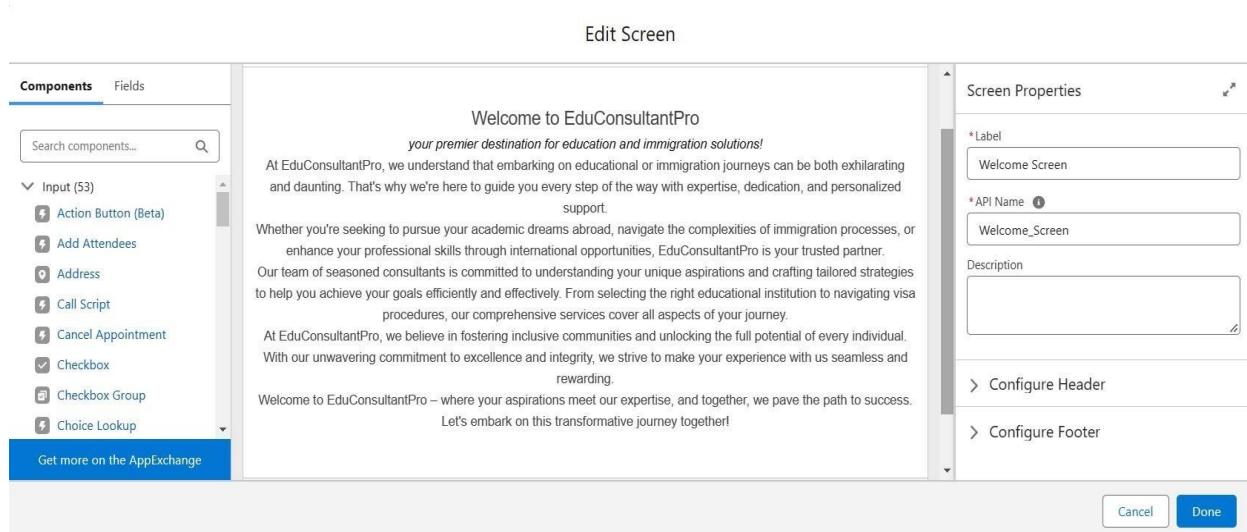
Whether you're seeking to pursue your academic dreams abroad, navigate the complexities of immigration processes, or enhance your professional skills through international opportunities, EduConsultantPro is your trusted partner.

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.

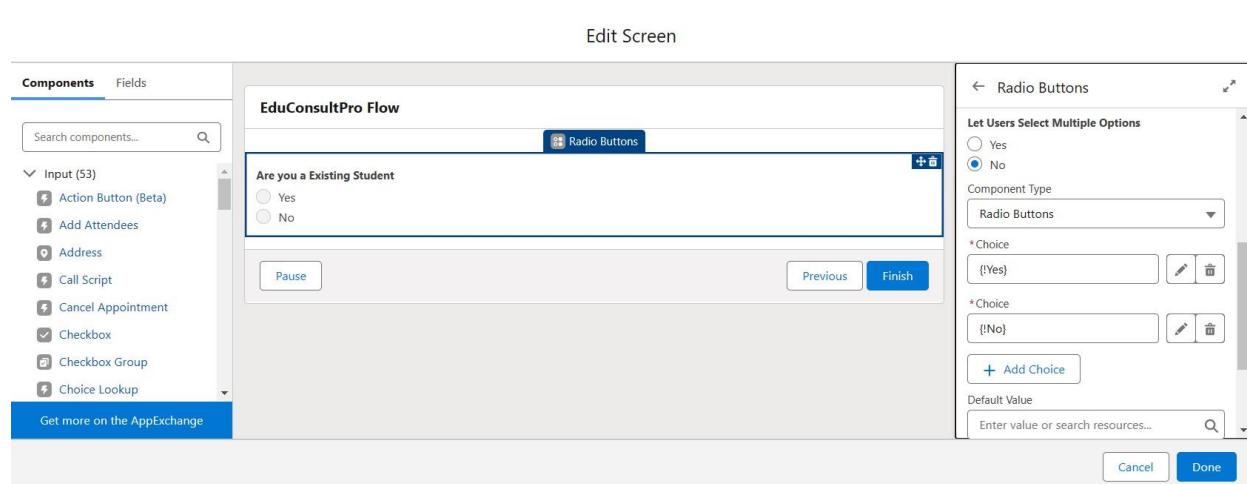
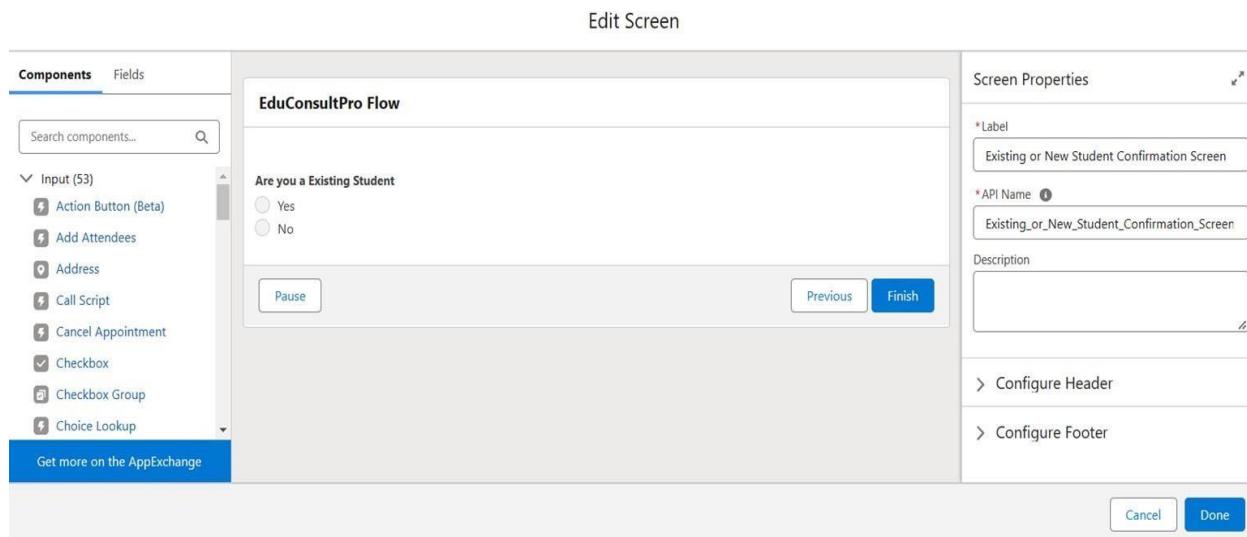
Welcome to EduConsultantPro – where your aspirations meet our expertise, and together, we pave the path to success. Let's embark on this transformative journey together!”

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 distinct outcome sections.

**Top Outcome Configuration:**

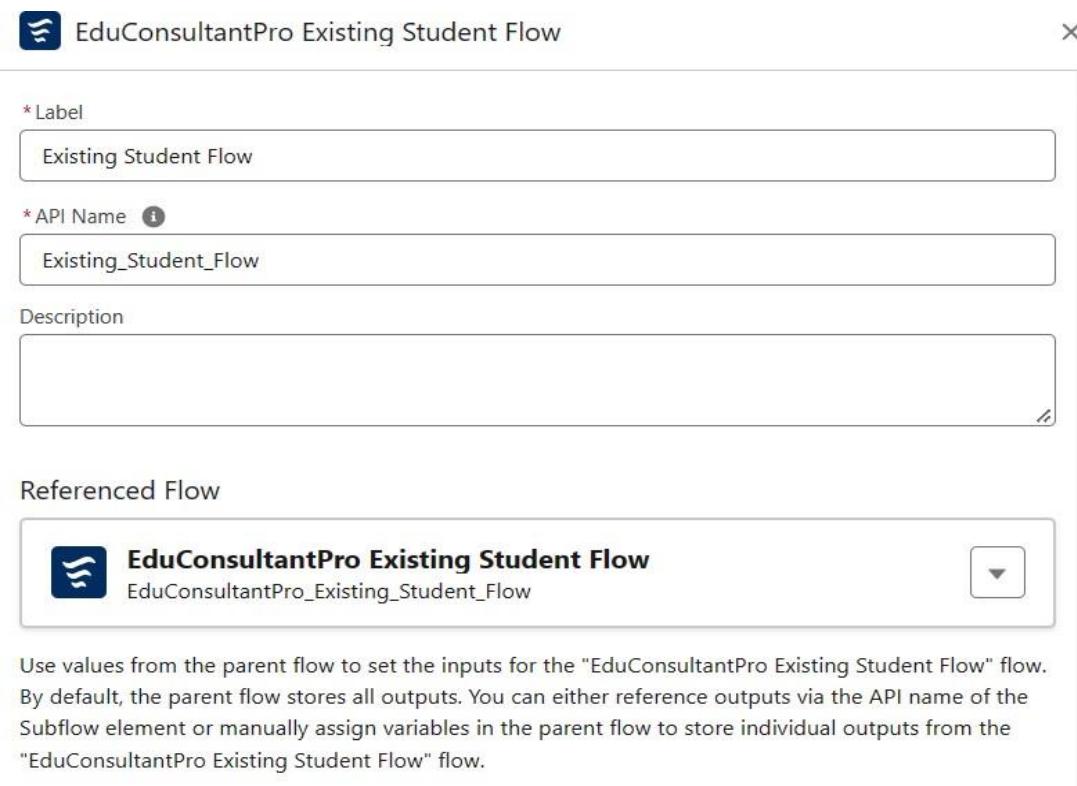
- Label:** Decision 1
- API Name:** Decision\_1
- Description:** (Empty)
- Outcomes:** For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.
- OUTCOME ORDER:** If Existing Student (highlighted in blue)
- OUTCOME DETAILS:**
  - 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 X
  - Operator:** Equals
  - Value:** Aa Yes X
- Add Condition:** + Add Condition

**Bottom Outcome Configuration:**

- OUTCOME ORDER:** If not a existing user (highlighted in blue)
- OUTCOME DETAILS:**
  - 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 X
  - Operator:** Equals
  - Value:** Aa No X
- Add Condition:** + Add Condition

## ***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**.



## ***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 ⓘ  
New\_Student\_Flow

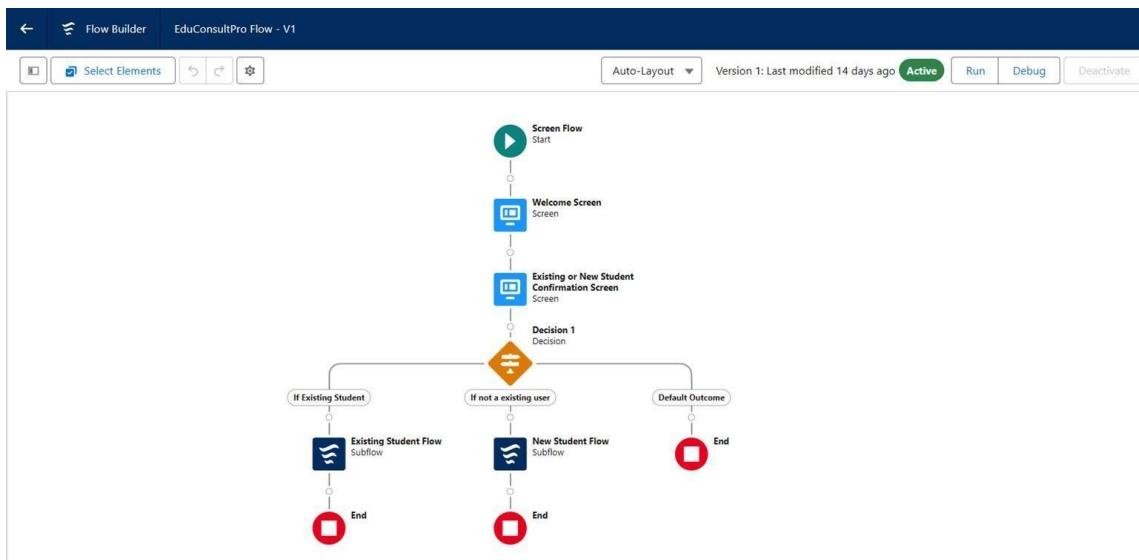
Description

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.



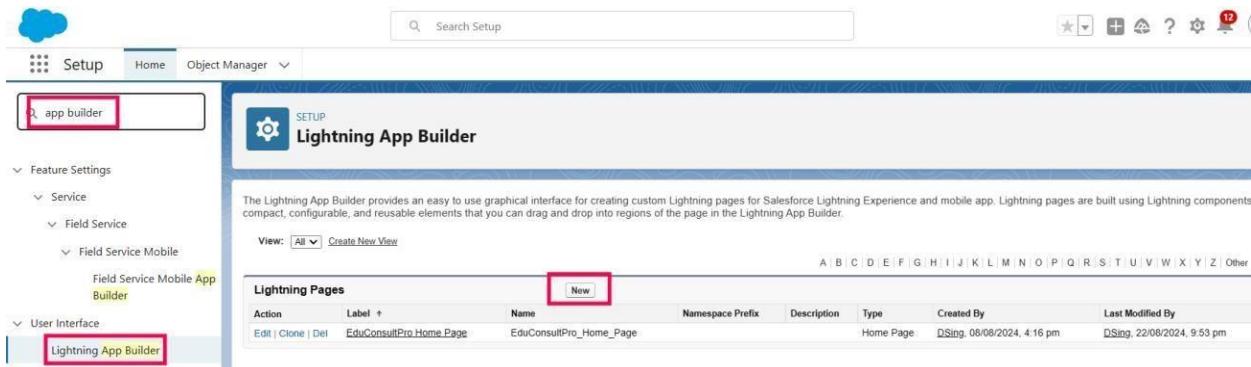
# Task 10: Create A Lightning App Page

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

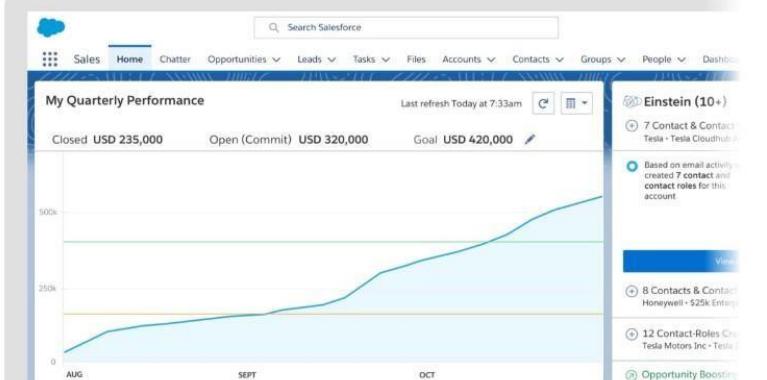
**Home Page**

Record Page

Embedded Service Page

Voice Extension

Customize the Lightning Experience Home page.



**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)	
Header and Three Regions	
Home Template One Region	
Home Template One Region	
Home Template One Region	
Home page header two columns left side bar	
<b>Standard Home Page</b>	

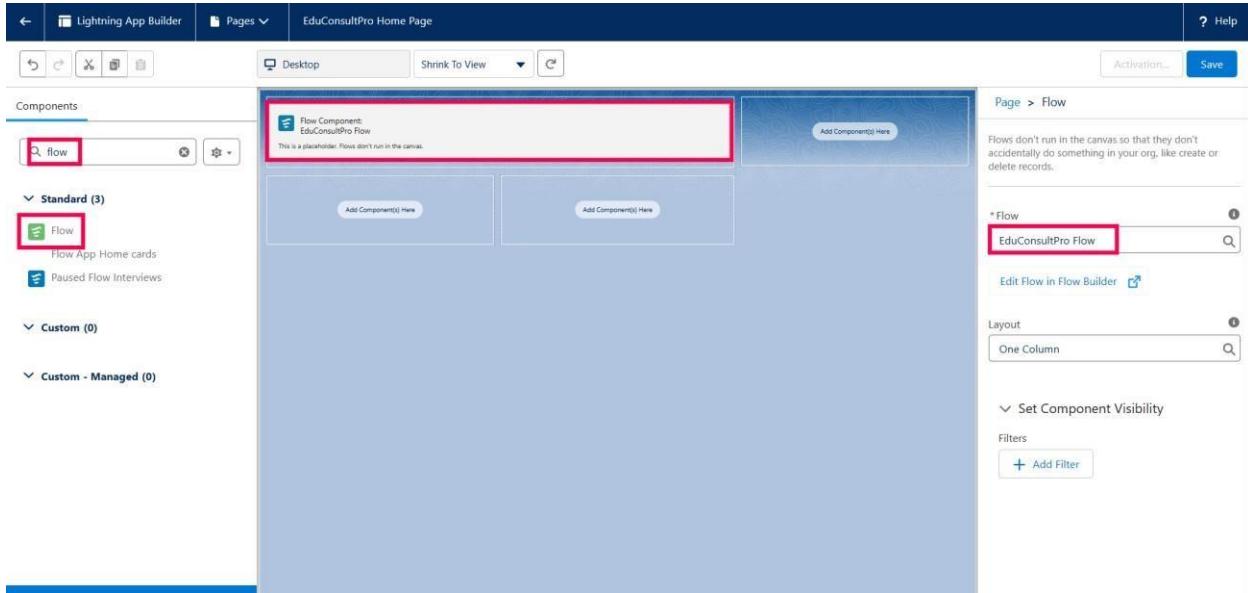
Lightning Experience Home page template with a header above two equal-width regions, and a full-height sidebar.

**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**.

## 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...	
<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)		<a href="#">Add Assignments</a>	<a href="#">Remove Assignments</a>
App	Profile		
Sales	System Administrator		

[Close](#)