

**Project Name:**

**A CRM APPLICATION TO MANAGE THE SERVICES  
OFFERED BY AN INSTITUTION**

**By:**

Lakshmi Prasanna Gottam

**Email id:** [2111it010036@mallareddyuniversity.ac.in](mailto:2111it010036@mallareddyuniversity.ac.in)

# ABSTRACT

The CRM (Customer Relationship Management) application designed to manage the services offered by an institution provides a comprehensive platform to enhance operational efficiency, streamline service delivery, and improve customer engagement. This application is tailored to meet the needs of institutions such as educational establishments, healthcare providers, or service-oriented organizations. The core functionality of the CRM includes robust customer profile management, which enables the institution to store and access detailed information about individuals, track interactions, and manage service requests. The service management module allows for the efficient cataloging of available services, handling of requests, and scheduling of appointments, while ensuring timely updates and notifications.

Communication tools integrated into the CRM facilitate seamless interaction between the institution and its clients, leveraging email, SMS, and in-app messaging to ensure consistent and effective communication. The application also incorporates features for appointment and event management, including calendar synchronization and automated reminders, which aid in minimizing scheduling conflicts and optimizing resource allocation. Additionally, the CRM application includes feedback and survey mechanisms to capture customer insights and measure satisfaction. By analyzing this data, institutions can refine their services and enhance the overall customer experience.

## INDEX

S. NO.	TASK	PAGE NO.
1	Create Objects from Spreadsheet	5-10
2	Create a ScreenFlow for Student Admission Application process	10-29
3	Create Users	30-32
4	Create an Approval Process for Property Object	32-38
5	Create a Record Triggered Flow	38-41
6	Create a ScreenFlow for Existing Student to Book an Appointment	41-58
7	Create a ScreenFlow to Combine all the flows at one place	59-64
8	Create a lightning app page	64-66

# INTRODUCTION

In today's dynamic and competitive landscape, institutions such as educational organizations, healthcare providers, and various service-oriented entities face the challenge of delivering high-quality, efficient, and personalized services to their clients. Managing these services effectively requires not only operational efficiency but also a deep understanding of customer needs and interactions. This is where a Customer Relationship Management (CRM) application becomes crucial. A CRM application designed for managing the services offered by an institution provides a centralized platform to streamline service delivery, enhance customer engagement, and optimize internal operations. Such an application integrates various functionalities that address the specific needs of the institution, enabling a comprehensive approach to service management. The CRM system facilitates the creation and maintenance of detailed customer profiles, allowing institutions to track interactions, service requests, and preferences. By offering tools for efficient service request handling, appointment scheduling, and real-time communication, the CRM application ensures that customer needs are addressed promptly and effectively.

Moreover, the application's communication features, including email integration, SMS notifications, and in-app messaging, support seamless and consistent interactions between the institution and its clients. This fosters improved customer satisfaction and loyalty.

Appointment and event management tools within the CRM help in organizing and optimizing institutional resources, while automated reminders reduce the likelihood of missed appointments and scheduling conflicts. Additionally, feedback and survey modules enable institutions to gather valuable insights, which can be used to refine services and enhance the overall customer experience. In essence, a CRM application tailored to manage the services of an institution not only improves operational efficiency but also builds stronger, more meaningful relationships with clients. By leveraging such a system, institutions can achieve greater organizational effectiveness, meet customer expectations more consistently, and maintain a competitive edge in their respective fields.

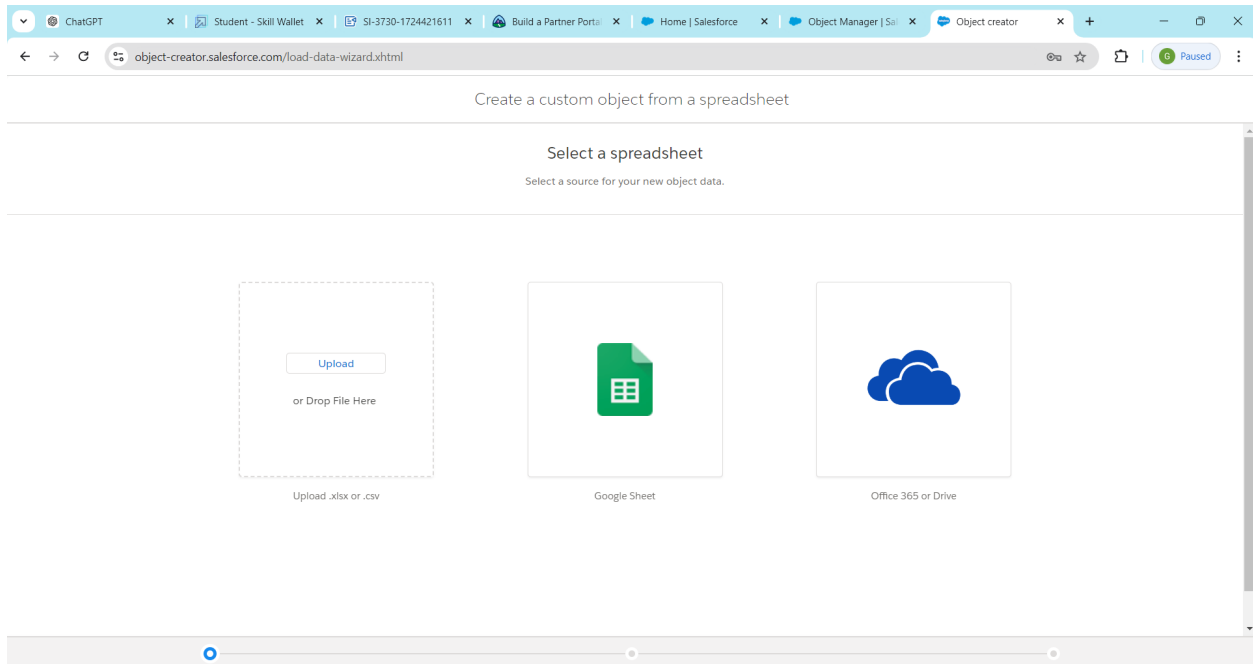
## Task 1: Create Objects from Spreadsheet

### A. Create Course object

In this step 1 we need to create an object using spreadsheet from objectmanager

Click on the link to get the spreadsheet, [Course](#).

After downloading, upload the file, map the fields and upload to create an object.



### A. Create Remaining objects

Repeat the steps which we have followed for course object creation.

Use the following sheets for remaining objects.

i. [Consultnt](#)

ii. [Student](#)

iii. [Appointment](#)

☐ **Create Relationship among the objects**

To Create:

- i. Go to Setup in Salesforce.
- ii. Under Objects and Fields, select Object Manager.
- iii. Click Create and choose Custom Object.
- iv. Enter Appointment as the label and configure the fields as required.
- v. Follow the same steps as creating the Appointment object and then create the remaining fields.
  - i. Appointment ID (Auto Number or Text, depending on format)
  - ii. Date (Date/Time)
  - iii. Time (Date/Time)
  - iv. Location (Text)
  - v. Consultant (Lookup to Consultant object)
  - vi. Student (Lookup to Student object)
- vii. Purpose (Text)

## **Define Relationships**

- Appointment to Student
- Lookup Relationship:
  - Create a field in the Appointment object that is a Lookup Relationship to the Student object.
  - Go to the Appointment object in Object Manager.
  - Click on Fields & Relationships.
  - Click New and choose Lookup Relationship.

- Select Student as the related object.

Follow the same steps as defining the relationships between Appointment to Student and then define the relationships between the following

- a. Appointment to Consultant
- b. Registration to Student
- c. Case to Student

## **Create Tabs for Objects**

After creating the custom objects and relationships, you'll need to create tabs to provide easy access to these objects.

- a. Go to Setup.
- b. Under User Interface, select Tabs.
- c. Click New to create a new tab.
- d. Select Custom Object Tabs and choose the custom object you want to create a tab for.
- e. Configure the tab settings, including the tab label, tab style, and visibility.
- f. Repeat the process for each object: Appointment, Student, Consultant, Registration, and Case.

This setup will allow you to effectively manage and view data related to appointments, students, consultants, registrations, and cases in Salesforce.

## **D)Configure the Case Object:**

Configuration Steps for the Case Object:

1. Navigate to Object Manager:
  - Go to Setup in Salesforce.
  - In the Quick Find box, type Object Manager and select it.
2. Find and Edit the Case Object:

- Scroll down or search for Case in the list of objects.
  - Click on Case to open its details.
  - Click Fields & Relationships to manage fields.
3. Configure the "Type" Field:
- Locate the Type field in the list of fields and click on it.
  - Click the Edit button to modify the field.
  - In the Picklist Values section, add new values:
    - a. Click New to add each value.
      - Enter Immigration and click Save & New.
      - Enter Visa Application and click Save.
4. Configure the "Status" Field:
- Go back to Fields & Relationships.
  - Find the Status field and click on it.
  - Click the Edit button.
  - In the Picklist Values section, add new values:
    - a. Click New to add each value.
      - Enter Open and click Save & New.
      - Enter In-progress and click Save.
5. Verify Changes:
- After saving, ensure that the new values are available for selection.
  - You might want to test this by creating a new case or editing an existing case to see if the new values are present in the picklist.

## **E) Create a Lightning App**

### Step-by-Step Guide to Create a Lightning App

1. Access Setup:
- Log in to Salesforce.
  - Click on the Gear Icon (⚙) in the upper right corner.
  - Select Setup from the dropdown menu.



2. Search for App Manager:

- In the Quick Find box on the left, type App Manager.
- Click on App Manager under Apps.

3. Create a New Lightning App:

- Click the New Lightning App button at the top right of the App Manager page.

4. Enter App Details:

- App Name: Enter EduConsultPro.
- Click Next to proceed.

5. Choose App Options:

- App Options: You can choose to configure or skip some options here. Generally, you can keep the default selections.
- Click Next.

6. Configure Navigation Items:

- In the Navigation Items section, you'll see two columns: Available Items and Selected Items.
- From the Available Items column, find and add the following items to the Selected Items column:
  - Home
  - Students
  - Courses
  - Consultants
  - Appointments
  - Registrations
  - Cases
- Use the Add button (or drag and drop) to move each item from Available to Selected.
- Click Next to proceed.

7. Assign User Profiles:

- In the User Profiles section, you will see a list of profiles.
  - From the Available Profiles column, select System Administrator and move it to the Selected Profiles column.
  - Click Next to proceed.
8. Review and Save:
- Review the details of your new app configuration.
  - Click Save & Finish to complete the process.
9. Verify the App Creation:
- To ensure the app was created successfully, go back to the App Manager.
  - Search for EduConsultPro in the list of apps.
  - You should see your new app listed. You can click the App Launcher (grid icon) to find and open your newly created app.

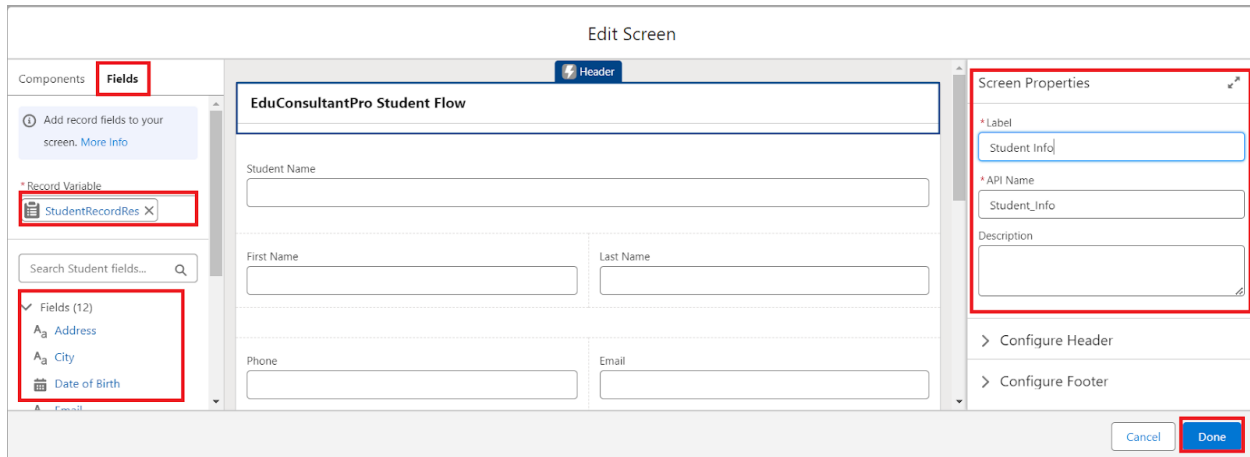
## **TASK:2**

### **Create a ScreenFlow for Student Admission Application process.**

#### **A)Adding a Screen Element in Flow Builder**

1. Access Flow Builder:
  - Log in to Salesforce.
  - Click on the Gear Icon (⚙) in the upper right corner.
  - Select Setup from the dropdown menu.
  - In the Quick Find box, type Flow Builder and select Flows from the results.
2. Create a New Flow:
  - Click the New Flow button.
  - Choose Screen Flow as the flow type.
  - Click Next.
3. Configure the Flow:

- Enter a name for your flow if prompted. You can use a name like Student Information Collection.
  - Click Create to open the Flow Builder.
4. Add a Screen Element:
- In the Flow Builder, drag the Screen element from the Elements panel on the left side of the screen to the canvas.
  - Drop it into the flow canvas.
5. Configure the Screen Element:
- In the Screen Properties pane on the right side, set the following:
    - a. Label: Enter Student Info.
      - You can also add a Description if desired to provide more context for the user.
6. Add Fields to the Screen:
- Click on Fields in the Screen Properties pane.
  - To display fields from the Student object, you need to create a record variable.
    - a. Click on the Record Variable input to open the variable selection pane.
      - Click New Resource to create a new variable.
        - Resource Type: Choose Variable.
        - API Name: Enter a name like StudentRecordRes.
        - Data Type: Select Record.
        - Object: Choose the Student object (or your specific object that holds student information).
        - Allow Multiple Values: Ensure this is unchecked if you are only dealing with a single record at a time.
        - Click Done.



### 1. Drag and Drop Fields:

- With StudentRecordRes now available, you should see a list of fields from the Student object in the Fields section of the Screen Properties pane.
- Drag the fields you need onto the screen. Common fields might include Name, Email, Phone Number, etc., but include any fields that are necessary to collect the student information.

### 2. Save the Screen Element:

- Once you have added all the necessary fields, you can adjust their order and layout if needed.
- Click Done in the Screen Properties pane to save the configuration of the Screen element.

### 3. Save and Activate the Flow:

- After configuring your screen element, make sure to save the flow by clicking the Save button.
- Provide a name and description for the flow if prompted.
- To make the flow available for use, click Activate.

## B)Create Student Record using Create Element

### Steps to Create a Student Record Using Create Element

#### 1. Open Your Flow:

- If you're not already in Flow Builder, log in to Salesforce.

- Click the Gear Icon (⚙) and select Setup.
- In the Quick Find box, type Flows and select Flows.
- Find and open the flow you're working on (e.g., Student Information Collection).

2. Add the Create Element:

- In Flow Builder, locate the Create Records element in the Elements panel on the left side.
- Drag the Create Records element onto the canvas, placing it after the Student Info Screen element.

3. Configure the Create Records Element:

- Label: Enter Create Student Record in the Label field.
- How many records to create: Select One.
- How to set the record fields: Choose Use all values from a record.

4. Set the Record Variable Resource:

- Under Create a record from these values, select the record variable resource you previously created, which is StudentRecordRes.
- Ensure that StudentRecordRes is the resource containing the student information captured from the screen.

### New Create Records

\* Label

Create Student Record

\* API Name

Create\_Student\_Record

Description

How Many Records to Create

☒ One

☐ Multiple


How to Set the Record Fields

☒ Use all values from a record

☐ Use separate resources, and literal values


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.



1. Map the Fields:
  - Since you've selected Use all values from a record, the fields from the StudentRecordRes variable will automatically map to the corresponding fields in the Student object. Ensure that the fields you need to create the record are correctly mapped.
2. Save and Connect the Elements:
  - Click Done to save the configuration for the Create Records element.
  - Connect the Student Info Screen element to the Create Student Record element by dragging an arrow from the Student Info element to the Create Student Record element on the canvas.
3. Save and Activate the Flow:

- Click the Save button to save your flow.
- Provide a name and description if prompted.
- To make the flow available for use, click Activate.

## C)Add Screen Element

### Steps to Add a Screen Element with a Picklist Component

1. Open Your Flow:
  - Log in to Salesforce.
  - Click the Gear Icon (⚙) and select Setup.
  - In the Quick Find box, type Flows and select Flows.
  - Open the flow you're working on (e.g., Student Information Collection).
2. Add a Screen Element:
  - In Flow Builder, drag the Screen element from the Elements panel on the left side to the canvas, placing it after the Create Student Record element. ■ Drop it onto the canvas.
3. Configure the Screen Element:
  - Label: Enter Course Screen.
  - You can also add a Description if needed to provide more context.
4. Add a Picklist Component:
  - In the Components section of the Screen Properties pane, drag the Picklist component onto the screen area.
5. Configure the Picklist Component:
  - Label: Enter Select Course.
  - API Name: This will be automatically populated based on the label but you can adjust if needed.
  - Choice Values: Click on Choices to configure the options.
    - Click New Choice to add each picklist value. ○ For each choice:
      - Label: Enter the name of the course, e.g., IELTS.
      - API Name: It will be automatically populated. For IELTS, it might be something like IELTS.
      - Value: Enter the same value as the label.
    - Click Done after entering each choice.
  - Repeat the above steps to add the following choices:
    - GRE

- GMAT
- Duolingo
- TOEFL

6. Save the Screen Element:

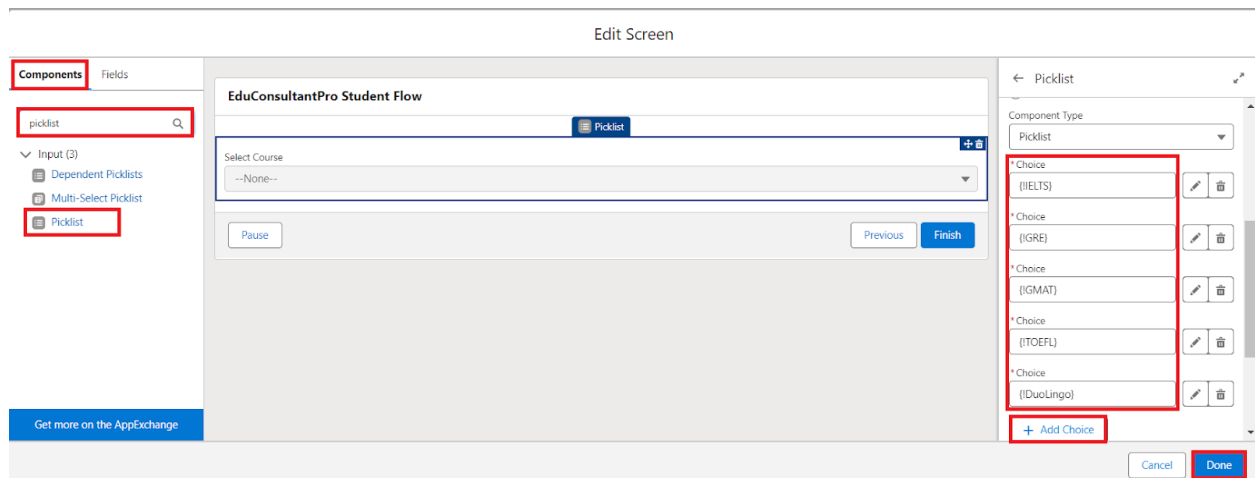
- After adding all the picklist options, click Done in the Screen Properties pane.

7. Connect the Elements:

- Connect the Create Student Record element to the newly created Course Screen element by dragging an arrow from Create Student Record to Course Screen on the canvas.

8. Save and Activate the Flow:

- Click Save to save your changes.
- Provide a name and description for the flow if prompted.
- Click Activate to make the flow available for use.



## D)Add Decision Element

### Steps to Add and Configure a Decision Element

1. Open Your Flow:

- Log in to Salesforce.
- Click the Gear Icon (⚙) and select Setup.
- In the Quick Find box, type Flows and select Flows.
- Open the flow you're working on (e.g., Student Information Collection).

2. Add a Decision Element:



- In Flow Builder, locate the Decision element in the Elements panel on the left side.
  - Drag the Decision element onto the canvas, placing it after the Select Course Screen element.
  - Drop it onto the canvas.
3. Configure the Decision Element:
- Label: Enter Selecting Course.
  - API Name: This will be automatically populated, but you can adjust it if needed.
4. Define Outcomes:
- Default Outcome: By default, there is a Default Outcome that will be used if none of the specified conditions are met. You can rename it or leave it as is.
5. Add Conditions for Each Option:
6. For each course choice, add an outcome with conditions:
7. a. Add Outcome for IELTS:
- Click + Add Outcome.
  - Outcome Label: Enter Selected IELTS.
  - Condition Logic: Set the conditions as follows:
    - Resource: Select Select\_Course (this should be the API name of the picklist component from the Select Course Screen element).
    - Operator: Choose Equals.
    - Value: Enter IELTS (this should match the API name or exact value of the picklist choice for IELTS).
8. b. Add Outcome for GRE:
- Click + Add Outcome.
  - Outcome Label: Enter Selected GRE.
  - Condition Logic: Set the conditions as follows:
    - Resource: Select Select\_Course.

- Operator: Choose Equals.
- Value: Enter GRE.

9. c. Add Outcome for GMAT:

- Click + Add Outcome.
- Outcome Label: Enter Selected GMAT.
- Condition Logic: Set the conditions as follows:
  - Resource: Select Select\_Course.
  - Operator: Choose Equals.
  - Value: Enter GMAT.

10.d. Add Outcome for Duolingo:

- Click + Add Outcome.
- Outcome Label: Enter Selected Duolingo.
- Condition Logic: Set the conditions as follows:
  - Resource: Select Select\_Course.
  - Operator: Choose Equals.
  - Value: Enter Duolingo.

11.e. Add Outcome for TOEFL:

- Click + Add Outcome.
- Outcome Label: Enter Selected TOEFL.
- Condition Logic: Set the conditions as follows:
  - Resource: Select Select\_Course.
  - Operator: Choose Equals.
  - Value: Enter TOEFL.

12. Save the Decision Element:

- After configuring all the outcomes, click Done to save the Decision element.

13. Connect the Elements:

- Connect the Select Course Screen element to the Selecting Course Decision element by dragging an arrow from Select Course to Selecting Course.

#### 14. Save and Activate the Flow:

- Click the Save button to save your changes.
- Provide a name and description for the flow if prompted.
- Click Activate to make the flow available for use.

**Edit Decision**  
Selecting Course (*Selecting\_Course*)

\* Label:  \* API Name:

Description:

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

OUTCOME ORDER	+	OUTCOME DETAILS	Delete Outcome
1		<p>* Label: <input type="text" value="Selected IELTS"/> * Outcome API Name: <input type="text" value="Selected_IELTS"/></p> <p>Condition Requirements to Execute Outcome: <input type="text" value="All Conditions Are Met (AND)"/></p> <p>Resource: <input type="text" value="Select_Course"/> Operator: <input type="text" value="Equals"/> Value: <input type="text" value="IELTS"/></p> <p>+ Add Condition</p>	

## E) Add GET Record Element

### Steps to Add a Get Records Element

#### 1. Open Your Flow:

- Log in to Salesforce.
- Click the Gear Icon (⚙) and select Setup.
- In the Quick Find box, type Flows and select Flows.
- Open the flow you're working on (e.g., Student Information Collection).

#### 2. Add a Get Records Element:

- In Flow Builder, locate the Get Records element in the Elements panel on the left side.

- Drag the Get Records element onto the canvas, placing it under the Selected IELTS path of the Selecting Course Decision element.
  - Drop it onto the canvas.
3. Configure the Get Records Element:
    - Label: Enter Get IELTS Rec.
    - API Name: This will be automatically populated but can be adjusted if needed.
  4. Select Object:
    - Object: Choose Course from the dropdown menu.
  5. Define Conditions:
    - Condition Requirement: Select All Conditions Are Met (AND).
    - Field: Select Course Name (or the field that corresponds to the course name in your Course object).
    - Operator: Choose Equals.
    - Value: Enter {!Select\_Course}. This is the API name of the picklist component from the Select Course Screen element. Ensure that this variable is correctly referenced to pull in the user's selection.
  6. Store the Records:
    - How Many Records to Store: Select Only the first record if you are expecting a single match.
    - Automatically store all fields if you want to use all fields from the Course record in subsequent elements.
  7. Save the Get Records Element:
    - Click Done to save the configuration for the Get Records element.
  8. Connect the Elements:
    - Connect the Selected IELTS path from the Selecting Course Decision element to the Get IELTS Rec element by dragging an arrow from Selected IELTS to Get IELTS Rec.
  9. Save and Activate the Flow:

- Click the Save button to save your changes.
- Provide a name and description for the flow if prompted.
- Click Activate to make the flow available for use.

**Edit Get Records**  
Get IELTS Rec (*Get\_IELTS\_Rec*)

\* Label:  \* API Name:

Description:

**Get Records of This Object**

\* Object:

**Filter Course Records**

Condition Requirements:

Field	Operator	Value
<input type="text" value="Name"/>	<input type="text" value="Equals"/>	<input type="text" value="A Select_Course"/>

+ Add Condition

## G)Create Email Text Template Variables for email body and subject

### Steps to Create Email Text Template Variables

#### 1. Open Your Flow:

- Log in to Salesforce.
- Click the Gear Icon (⚙) and select Setup.
- In the Quick Find box, type Flows and select Flows.
- Open the flow you're working on (e.g., Student Information Collection).

#### 2. Add a New Text Template Resource:

- Click the Toggle Toolbox on the left side of the Flow Builder screen.

- Click New Resource.
- Select Text Template as the Resource Type.

### 3. Configure the Text Template:

- API Name: Enter StuRegistrationEmailTextTempBody.
- Description: (Optional) Enter a description to help identify the template.
- Text Template Type: Select View as Plain Text.
- Body: Paste the following text into the body field. Make sure to include the merge field for dynamic data:

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.

Here are a few key points to help you get started:

**Explore Our Resources:** Take some time to explore the wide range of resources, tools, and services available on the EduConsultantPro platform. From educational insights to immigration advice, we offer comprehensive support tailored to your needs.

**Connect with Our Consultants:** Our team of experienced consultants is here to assist you at every stage of your educational and immigration endeavors. Don't hesitate to reach out to us with any questions, concerns, or inquiries you may have. We're here to help!

**Stay Updated:** Keep an eye on your inbox for important updates, announcements, and exclusive opportunities from EduConsultantPro. We'll ensure that you're informed about the latest developments and relevant information to support your journey.

**Engage with the Community:** Join our vibrant community of students, professionals, and experts who share your interests and aspirations. Participating in discussions, networking, and sharing your experiences can enrich your journey with valuable insights and connections.

Once again, congratulations on your registration, and welcome to EduConsultantPro!

We look forward to supporting you as you embark on this exciting new chapter.

Best regards,

The EduConsultantPro Team

1. Save the Text Template:

- Click Done to save the text template.

Creating the Email Subject Template Variable

Repeat the above steps to create a Text Template for the email subject:

1. Add a New Text Template Resource:

- Click New Resource.
- Select Text Template as the Resource Type.

2. Configure the Text Template:

- API Name: Enter StuRegistrationEmailSubject.
- Description: (Optional) Enter a description to help identify the template.
- Text Template Type: Select View as Plain Text.
- Body: Enter the following text for the subject line:
  - Welcome to EduConsultantPro,

{!StudentRecordRes.Name}! • Save the Text Template:

- Click Done to save the text template.

### Edit Text Template

**\* API Name**

StuRegistrationEmailTextTempBody

**Description**

**\* Body** ⓘ

**Resource Picker**

🔍

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.

Here are a few key points to help you get started:

Cancel

Done

## H)Add an Action Element

### Steps to Add and Configure the Action Element

1. Open Your Flow:
  - Log in to Salesforce.
  - Click the Gear Icon (⚙️) and select Setup.
  - In the Quick Find box, type Flows and select Flows.
  - Open the flow you're working on (e.g., Student Information Collection).
2. Add the Action Element:
  - In Flow Builder, locate the Action element in the Elements panel on the left side.
  - Drag the Action element onto the canvas, placing it after all Decision paths



(e.g., under Selected IELTS, Selected GRE, Selected GMAT, Selected Duolingo, Selected TOEFL).

3. Configure the Action Element:

- Label: Enter Send Email to Student.
- API Name: This will be automatically populated but can be adjusted if needed.

4. Select the Action Type:

- Under Action Type, choose Send Email from the available options.

5. Set Input Values for Selected Action:

- Click on Set Input Values to configure the email details.

6. a. Set Input Body:

- Body: Choose the variable for the email body.
  - a. Click on the Body field.
    - Select the Text Template resource you created for the body of the email, which is `{!StuRegistrationEmailTextTempBody}`.

7. b. Set Input Recipient Address List:

- Recipient Address List: Enter the recipient's email address.
  - a. Click on the Recipient Address List field.
    - Choose the appropriate variable that contains the recipient's email address, which is `{!StudentRecordRes.Email__c}`.

8. c. Set Input Subject:

- Subject: Choose the subject for the email.
  - a. Click on the Subject field.
    - Select the Text Template resource you created for the subject of the email, which is `{!StuRegistrationEmailTextTempSub}`.

9. Save the Action Element:

- Click Done to save the configuration for the Action element.


10. Connect the Elements:

- Connect each Decision path (e.g., Selected IELTS, Selected GRE, Selected

GMAT, Selected Duolingo, Selected TOEFL) to the Send Email to Student Action element by dragging an arrow from each path to the Action element.

## 11. Save and Activate the Flow:

- Click the Save button to save your changes.
- Provide a name and description for the flow if prompted.
- Click Activate to make the flow available for use.

 Edit Send Email

Send Email to Student (*Send\_Email\_to\_Student*)

\* Label







Send Email to Student

\* API Name

Send\_Email\_to\_Student

Description

### Set Input Values for the Selected Action

<b>A<sub>a</sub></b> Body 	<div><div></div><div>{!StuRegistrationEmailTextTempBody}</div></div>	<div><div></div><div>Include</div></div>
<b>A<sub>a</sub></b> Recipient Address List 	<div><div></div><div>{!StudentRecordRes.Email__c}</div></div>	<div><div></div><div>Include</div></div>
<b>A<sub>a</sub></b> Subject 	<div><div></div><div>{!StuRegistrationEmailTextTempSub}</div></div>	<div><div></div><div>Include</div></div>

## I)Add Screen Element

## Steps to Add and Configure the Success Screen Element

- ## 1. Open Your Flow:

- Log in to Salesforce.
  - Click the Gear Icon (⚙) and select Setup.
  - In the Quick Find box, type Flows and select Flows.
  - Open the flow you're working on (e.g., Student Information Collection).
2. Add a Screen Element:
- In Flow Builder, locate the Screen element in the Elements panel on the left side.
  - Drag the Screen element onto the canvas, placing it after the Send Email to Student action element.
  - Drop it onto the canvas.
3. Configure the Screen Element:
- Label: Enter Success Screen.
  - API Name: This will be automatically populated but can be adjusted if needed.
4. Add a Display Text Component:
- In the Components section of the Screen Properties pane, search for Display Text.
  - Drag the Display Text component onto the screen area.
5. Configure the Display Text Component:
- Label: Enter SuccessMessage.
  - Text: In the text area, paste the following content:

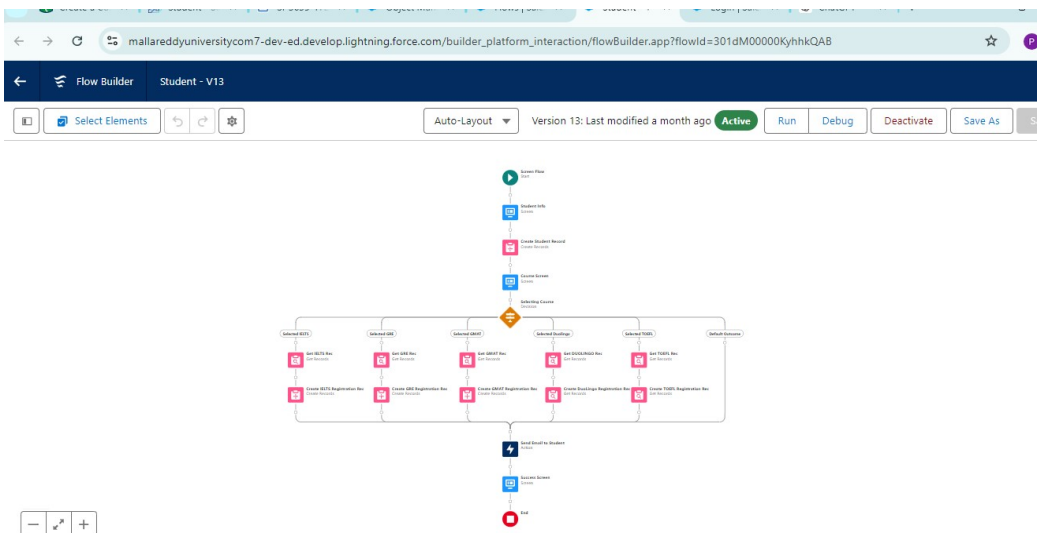
Dear {!StudentRecordRes.Name},

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 your email once. Thank you.

- Ensure that the merge field `{!StudentRecordRes.Name}` is correctly used to personalize the message with the student's name.
1. Save the Screen Element:
    - Click Done to save the configuration for the Screen element.
  2. Connect the Elements:
    - Connect the Send Email to Student action element to the Success Screen element by dragging an arrow from Send Email to Student to Success Screen.
  3. Save and Activate the Flow:
    - Click Save to save your changes.
    - Provide a name for the flow, such as EduConsultPro Student Flow, and enter a description if prompted.

Click Activate to make the flow available for use.



## TASK 3:

## CREATE USERS

### A) User

1. Log in to Salesforce:

- Open your web browser and log in to your Salesforce instance.

2. Navigate to User Management:

- Click on the Gear Icon (⚙) in the top-right corner to access the Setup menu.
- In the Quick Find box on the left side, type Users and select Users under the Administration section.

3. Create a New User:

- Click the New User button at the top of the User list page.

4. Fill Out User Details:

- Last Name: Enter Consultant.
- License: Choose Salesforce Platform from the dropdown menu.  
Profile: Select Standard Platform User from the dropdown menu.

5. Fill in the other mandatory fields as follows:

- First Name: Enter the user's first name (e.g., John).
- Email: Enter the user's email address (e.g., john.consultant@example.com).
- Username: Enter a unique username in the form of an email address (e.g., john.consultant@example.com).
- Alias: Enter a short nickname or alias (e.g., JConsultant).
- Nickname: Enter a nickname if needed (optional).
- Role: Select the appropriate role if roles are used in your organization (optional).
- Locale Settings: Set the locale, time zone, and language preferences if needed (optional).
- Active: Ensure the Active checkbox is selected to enable the user.

1. Save the User Record:

- Click the Save button to create the new user.

2. Verify User Creation:

- After saving, you will be redirected to the User detail page.
- Confirm that all entered details are correct and that the user is listed in the User

list.

## B)Configure the User Settings

### 1. Log in to Salesforce:

- Open your web browser and log in to your Salesforce instance.

### 2. Navigate to User Management:

- Click on the Gear Icon (⚙) in the top-right corner to access the Setup menu.
- In the Quick Find box on the left side, type Users and select Users under the Administration section.

### 3. Edit Your User Record:

- Locate your user record in the list of users.

Click the Edit link next to your name to open the user details for editing.

### 4. Configure Approver Settings:

- Scroll down to the Approver Settings section. This is usually located towards the bottom of the user detail page.
- In the Manager field under Approver Settings, select Consultant from the dropdown menu. This sets the specified user as your manager or approver.

### 5. Save the Changes:

- Click the Save button to apply your changes.

### 6. Verify Settings:

- After saving, review the user details to ensure that the Manager field has been updated correctly to Consultant.

The screenshot shows the 'Approver Settings' section in Salesforce. It includes a 'Delegated Approver' field, a 'Manager' dropdown menu currently set to 'Consultant', and a 'Receive Approval Request Emails' dropdown menu set to 'Only if I am an approver'. The 'Manager' field and the 'Save' button at the bottom are highlighted with red boxes.

## **TASK:4**

### **Create an Approval Process for Property Object**

#### **A)Create an Email Template**

##### Steps to Create a Lightning Email Template

##### 1. Enable Lightning Email Templates:

- Log in to Salesforce.
- Click on the Gear Icon (⚙) in the top-right corner and select Setup.
- In the Quick Find box, type Templates.
- Select Lightning Email Templates from the results.

##### 1. Ensure the feature is toggled on.

##### 2. Access Email Templates in the App Launcher:

- Click on the App Launcher (grid icon) in the top-left corner.
- Search for Email Templates and select the Email Templates app from the results.

##### 3. Create a New Folder:

- In the Email Templates app, click the Folders tab.
- Click the New Folder button.
- Enter a Name for the folder (e.g., Appointment Templates).
- Click Save to create the folder.

##### 4. Create a New Email Template:

- Go to the Email Templates tab within the Email Templates app.
- Click New Email Template.
- Folder: Select the folder you created earlier (e.g., Appointment Templates).
- Name: Enter Submission Template as the name of the email template.

##### 5. Enter Email Content:

- HTML Value: In the HTML Body field, paste the following content:

- html
- Copy code
- 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 support you need.

Best regards,

[Your Name]



### New Email Template

Information

Email Template Name

Student Request Template

Related Entity Type

Appointment

Description

Folder

EduConsultantPro

Select Folder

Message Content

Subject

Appointment Request with EduConsultantPro Consulta

Enhanced Letterhead

Search Enhanced Letterheads...

HTML Value

Source

Font

Size

B

I

U

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}}},

Cancel

Save

- You can replace [Your Name] with your actual name or a placeholder if needed.

## 6. Save the Email Template:

- Click **Save** to save the email template with the specified content.

## B) Create an Approval Process

1. Go to Setup:

- From Setup, enter "Approval" in the Quick Find box.
- Select "Approval Processes."

## 2. Manage Approval Processes For:

- In the "Manage Approval Processes For" dropdown, select "Appointment."

### 3. Create New Approval Process:

- Click on "Create New Approval Process."
- Choose "Use Jump Start Wizard."

## Configure the Approval Process

1. Process Name:
  - Enter "Appointment Approval" for the Process Name.
2. Select Approver:
  - Under "Select Approver," choose "Manager" for the option: “Automatically assign an approver using a standard or custom hierarchy field.”
  - Select "Manager" under "Automated Approver Determined By."
3. Record Editability Properties:
  - Choose "Administrators or the currently assigned approver can edit records during the approval process."
4. Save the Approval Process:
  - Click "Save."

#### View Approval Process Detail Page

1. Initial Submission Actions:
  - Click on "View Approval Process Detail Page."
  - Under "Initial Submission Actions," click "Add New."
2. Field Update:
  - Select "Field Update."
  - Configure with the following values:
    - a. Name: Submitted
      - Field to Update: Appointment: Status
      - A Specific Value: Pending
  - Click "Save."
3. Email Alert:
  - Click "Add New" under Initial Submission Actions.
  - Select "Email Alert."
  - Configure with these values:
    - a. Description: Submission Email Alert

- Unique Name: (Auto-populates)
- Email Template: Submission Template
- Recipient Type: Select your name

■ Click "Save."

Show More'. At the bottom are 'Save' and 'Cancel' buttons."/>

**Select Approver**

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

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

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

**Save** Cancel

## Configure Final Approval and Final Rejection Actions

### 1. Final Approval Actions:

- Under "Final Approval Actions," click "Add New."
- Select "Field Update."
- Configure with these values:
  - a. Name: Approved
    - Field to Update: Appointment: Status
    - A Specific Value: Approved
- Click "Save."
- Click "Add New" under Final Approval Actions.
- Select "Email Alert."
- Configure with these values:
  - Description: Final Approval Email Alert
  - Unique Name: (Auto-populates)
  - Email Template: Approval Template
  - Recipient Type: Select your name

- Click "Save."

## 2. Final Rejection Actions:

- Under "Final Rejection Actions," click "Add New."
- Select "Field Update."
- Configure with these values:
  - a. Name: Rejected
    - Field to Update: Appointment: Status
    - A Specific Value: Rejected
- Click "Save."
- Click "Add New" under Final Rejection Actions.



- Select "Email Alert."
- Configure with these values:
  - Description: Final Rejection Email Alert
  - Unique Name: (Auto-populates)
  - Email Template: Rejection Template
  - Recipient Type: Select your name
- Click "Save."

## Finalize and Activate

1. Review Your Approval Process:
  - Ensure all steps and configurations are correct.
2. Activate the Approval Process:
  - Click "Activate" to make the approval process active.

## **TASK 5:**

### **Create a Record Triggered Flow**

1. Go to Setup:
  - From Setup, enter "Flows" in the Quick Find box.
  - Select "Flows."
2. Create a New Flow:
  - Click on "New Flow."
3. Select Flow Type:
  - Choose "Record-Triggered Flow."
  - Click "Create."

### **A)Configure the Start Element**

1. Configure Start Element:
  - The "Configure Start" window will open.
2. Select Object:
  - For "Object," select "Appointment."
3. Trigger the Flow When:
  - For "Trigger the Flow When," select "A record is created."
4. Optimize the Flow for:
  - (Optional) Choose the optimization settings according to your needs, such as “Fast Field Updates” or “Actions and Related Records,” depending on what the flow is designed to do.
5. Set Entry Conditions (if needed):
  - If you need the flow to trigger only under certain conditions, specify those here. For example, you might want to check if certain fields meet specific criteria.
6. Define Trigger Criteria:
  - If applicable, configure the trigger criteria to specify additional conditions for

when the flow should start. This is useful if you don't want the flow to trigger on every record creation but only on specific conditions.

## 7. Click Done:

- After configuring the start element, click "Done."

### 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

\* Optimize the Flow for:

#### Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs *before* the record is saved to the database.

#### Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs *after* the record is saved to the database.

- ☐ Include a Run Asynchronously path to access an external system after the original transaction for the triggering record is successfully committed


## B) Add an Action Element

### 1. Open Your Flow:

- If you haven't already, navigate to the Flow you created.
- Click on "Edit" to open the Flow Builder.

## 2. Add the Action Element:

- After the Start element, click on the "+" icon to add a new element.
- Select "Action" from the list of element types.

 Edit Submit for Approval  
approval subflow (*approval\_subflow*)

\* Label

approval subflow

\* API Name

approval\_subflow

Description

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

A<sub>a</sub> \* Record ID ⓘ

{!\$Record.Id}

A<sub>a</sub> Approval Process Name Or ID

☐ Don't Include

A<sub>a</sub> Next Approver IDs

☐ Don't Include


## 3. Configure the Action Element:

- Action Type:
  - In the "Action" configuration screen, search for and select "Submit for Approval" from the list of available actions.
- Label:
  - a. Set the Label for the action element to "Approval SubFlow."
- Record ID:
  - Set the "RecordId" field to `{!$Record.Id}`. This ensures that the approval

process is triggered for the record that initiated the flow.

#### 4. Configure the Submit for Approval Action:

- Approval Process:
  - a. Make sure that the correct Approval Process is associated with this action. If you have multiple approval processes, ensure that the one for appointments is selected.

 Edit Submit for Approval  
approval subflow (*approval\_subflow*)

\* Label

approval\_subflow

\* API Name

approval\_subflow

Description

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

A<sub>a</sub> \* Record ID ⓘ

{!\$Record.Id}

A<sub>a</sub> Approval Process Name Or ID

☐ Don't Include

A<sub>a</sub> Next Approver IDs

☐ Don't Include

#### 5. Connect the Elements:

- Ensure that the Action element is properly connected to the Start element in your flow.

#### 6. Save and Test Your Flow:

- Click "Save" to save your changes.
- Click "Debug" to test your flow and ensure that it works as expected. This step helps verify that the record is submitted for approval correctly.

#### 7. Activate the Flow:



- Once you have tested the flow and confirmed it works as expected, click "Activate" to make the flow live.

## **TASK 6:**

### **Create a ScreenFlow for Existing Student to Book an Appointment**

#### **A)Add Screen Element**

Create and Configure a Screen Flow

1. Navigate to Flow Builder:

- From Setup, enter "Flow Builder" in the Quick Find box.
- Select "New Flow."

2. Select Flow Type:

- Choose "Screen Flow."
- Click "Create."

Add and Configure the Screen Element

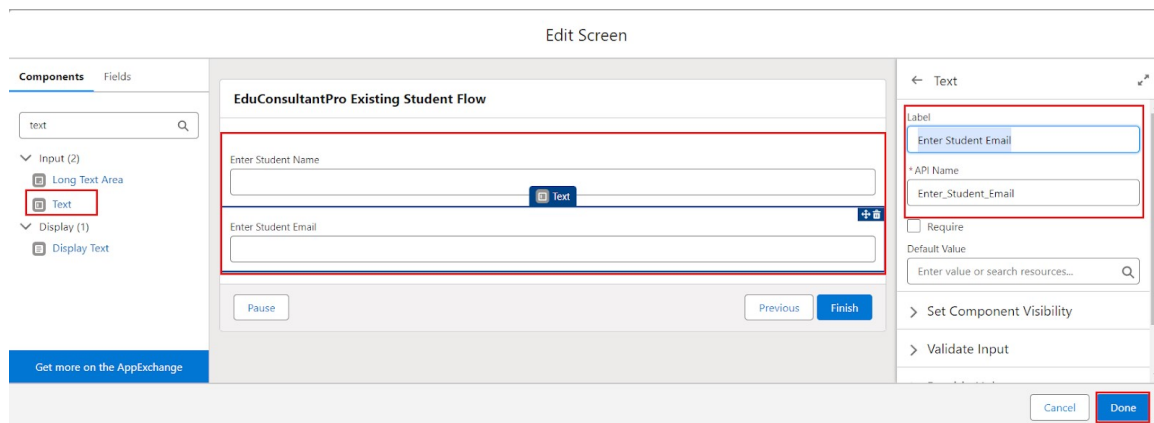
1. Add a Screen Element:

- In the Flow Builder, click on the "+" icon to add a new element. ■ Select "Screen" from the list of elements.

2. Configure Screen Properties:

- In the "Screen Properties" pane that appears:
  - Label: Enter "Get Student Info."
  - API Name: (This auto-generates based on the label, but you can edit it if needed.)
- a. Help Text: (Optional) Provide any help text if you want to guide users on what to

enter in the fields.



### 3. Add Text Components:

- Drag and drop two "Text" components from the left-side panel onto the screen area.

### 4. Configure the First Text Component:

- Click on the first Text component to open its properties.

#### a. Label: Enter "Enter Student Name."

- API Name: (Auto-generates, but you can customize if needed.)
- Data Type: Leave as "Text" (which is the default).
- Required: (Optional) Check this if you want to make the field mandatory.
- Default Value: (Optional) Set a default value if needed.

### 5. Configure the Second Text Component:

- Click on the second Text component to open its properties.

#### a. Label: Enter "Enter Student Email."

- API Name: (Auto-generates, but you can customize if needed.)
- Data Type: Leave as "Text" (which is the default).
- Required: (Optional) Check this if you want to make the field mandatory.
- Default Value: (Optional) Set a default value if needed.

### 6. Add Component Validation:

- For the "Enter Student Email" component, you might want to add validation for email format. This can be done by specifying validation rules in the component settings.

#### 7. Save the Screen Element:

- Click "Done" to save the screen configuration.

### Connect the Screen Element

#### 1. Connect the Elements:

- Ensure that the Screen element is properly connected to other elements in your flow. For instance, if you have a start element, connect it to the "Get Student Info" screen.
- If the flow includes other steps after gathering the student information, ensure to connect the Screen element to these subsequent steps.

### Test and Activate the Flow

#### 1. Save and Test:

- Click "Save" to save your flow.
- Click "Debug" to test your flow and ensure that the screen appears as expected and captures the information correctly.

#### 2. Activate the Flow:

- Once you are satisfied with the configuration and testing, click "Activate" to make the flow live.

## **B)Add GET Record Element**

#### 1. Open Your Flow:

- Navigate to Flow Builder where your flow is being edited.
- Ensure you are in the correct flow where you need to add the GET Records

element.

2. Add GET Records Element:

- Click on the "+" icon in the Flow Builder to add a new element.
- Select "Get Records" from the list of element types.

3. Configure GET Records Element:

- Label:
  - a. Enter "Get Rec" for the label.
- Object:
  - For "Object," select "Student."
- Condition Requirements:
  - Select "All Conditions Are Met (AND)." This ensures that all specified conditions must be true for the record to be retrieved.

4. Set the Filter Conditions:


- Field: Select "Student Name."
  - a. Operator: Equals
    - Value: Enter {!Enter\_Student\_Name} (This refers to the variable or value from the previous Screen element where the student name is input.)
- Click "Add Condition" to add another condition.
- Field: Select "Email\_\_c."
  - Operator: Equals
    - Value: Enter {!Enter\_Student\_Email} (This refers to the variable or value from the previous Screen element where the student email is input.)

5. Store the Record Data:

- How Many Records to Store:
  - a. Choose "Only the first record" if you expect to find only one student matching the criteria.
    - Choose "All records" if you expect multiple records and want to process them all.

■ Store Record Data In:

- Specify how you want to store the record. Typically, you would store it in a variable for later use. For example, create a variable named "StudentRecord" to store the retrieved Student record data.

 Edit Get Records  
Get Rec (*Get\_Rec*)

\* 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

Name

Equals

Enter\_Student\_Name

X

Field

Operator

Value

Email\_c

Equals

Enter\_Student\_Email

X

AND

+ Add Condition

6. Connect the Elements:

- Ensure that the GET Records element is connected to the correct path in your flow. It should follow the Decision element under the IELTS path, as specified.

7. Save the GET Records Element:

- Click "Done" to save your configuration. Test

and Activate the Flow

1. Save and Test:

- Click "Save" to save your changes.
- Click "Debug" to test the flow. Ensure that the GET Records element correctly retrieves the Student record based on the provided criteria.

2. Activate the Flow:

- Once the flow has been tested and is functioning as expected, click "Activate" to make the flow live.

## C)Add Decision Element

### Add and Configure the Decision Element

#### 1. Open Your Flow:

- Navigate to Flow Builder where your flow is being edited.

#### 2. Add Decision Element:

- Click on the "+" icon in the Flow Builder to add a new element.
- Select "Decision" from the list of element types.

#### 3. Configure Decision Element Properties: ■ Label:

- a. Enter "Appointment or Case" for the label. ■ API Name:

- The API Name auto-generates, but you can customize it if needed.

The screenshot shows the 'Edit Decision' configuration window for a decision element named 'Appointment or Case (Appointment\_or\_Case)'. The interface includes the following fields and sections:

- \* Label:** A text field containing 'Appointment or Case'.
- \* API Name:** A text field containing 'Appointment\_or\_Case'.
- Description:** A large empty text area.
- Outcomes:** A section with a header '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.' It includes a 'Delete Outcome' button.
- OUTCOME ORDER:** A list on the left with 'Appointment' and 'Case'. 'Appointment' is selected and highlighted in blue.
- OUTCOME DETAILS:** A section for the selected 'Appointment' outcome.
  - \* Label:** A text field containing 'Appointment'.
  - \* Outcome API Name:** A text field containing 'Appointment'.
  - Condition Requirements to Execute Outcome:** A dropdown menu showing 'All Conditions Are Met (AND)'.
  - Resource:** A text field containing 'How\_may\_I\_Help\_you'.
  - Operator:** A dropdown menu showing 'Equals'.
  - Value:** A text field containing 'Book\_an\_Appointment'.
  - + Add Condition:** A button to add more conditions.

#### 4. Configure Outcomes:

- Outcome Label: Appointment

- a. Click on "Add Outcome" to create the first outcome.
  - Label: Enter "Appointment."
  - API Name: (Auto-generates, but you can customize it if needed.)
  - Condition Requirements: Select "All Conditions Are Met (AND)".
  - Resource: Select {!How\_may\_I\_Help\_you} (the variable that stores the user's input or choice).
  - Operator: Equals
  - Value: Enter {!Book\_an\_Appointment} (the value representing the appointment option).

- Add Another Outcome:
  - Click on the "+" icon again to add another outcome.
  - Label: Enter "Case."
  - API Name: (Auto-generates, but you can customize it if needed.)
  - Condition Requirements: Select "All Conditions Are Met (AND)".
  - Resource: Select {!How\_may\_I\_Help\_you}.
  - Operator: Equals
  - Value: Enter {!Log\_a\_Case} (the value representing the case option).

5. Set Default Outcome (Optional):

- You can also set a default outcome if none of the conditions are met. This outcome will be used if the conditions for "Appointment" and "Case" are not satisfied.
- Label: Enter "Default" or similar.
- API Name: (Auto-generates, but you can customize it if needed.)

6. Connect the Decision Element:

- Connect the "Select Display Student Details" element to the "Appointment or Case" decision element.
- Connect each outcome of the Decision element to the appropriate subsequent elements in your flow (e.g., actions for appointment handling or case handling).

## 7. Save the Decision Element:

- Click "Done" to save your configuration.

## **D)Add Screen Element**

### Add and Configure the Screen Element

#### 1. Open Your Flow:

- Navigate to Flow Builder where your flow is being edited.

#### 2. Add Screen Element:

- Click on the "+" icon in the Flow Builder to add a new element.
- Select "Screen" from the list of element types.

#### 3. Configure Screen Properties:

- Label: Enter "Appointment Booking Screen."
- API Name: (This auto-generates but can be customized if needed.)
- Help Text: (Optional) Provide any help text to guide users on what information to enter.

#### 4. Create and Configure a Record Variable:

- Create a Record Variable:
  - a. If you haven't already created a record variable for the Appointment object, do so now.
    - Go to the "Manager" tab on the left side of the Flow Builder.
    - Click "New Resource."
    - Select "Variable" for Resource Type.
    - Set the following values:
      - API Name: AppointmentRecordRes
      - Data Type: Record
      - Object: Appointment
      - Allow multiple values (Collection): No (unless you need to



handle multiple records)

- Available for Input: Yes
- Available for Output: Yes (if needed for later use)

- Click "Done" to create the variable.

#### 5. Add Fields to the Screen:

- In the Screen Properties pane, click on "Fields."
- Under the "Record Variable Input" section, select the record variable AppointmentRecordRes you just created.
- Drag and drop the fields you need from the Appointment object onto the screen.
  - a. For example, if you need fields like Appointment Date, Appointment Time, and Description, drag those fields to the screen.

#### 6. Configure Each Field:

- Click on each field you've dragged onto the screen to configure its properties:
  - a. Label: Ensure each field has a clear label that indicates what information should be entered.
    - API Name: (Auto-generates but can be customized if needed.)
    - Required: (Set to "Required" if the field must be filled out.)

#### 7. Save the Screen Element:

- Once you have added and configured all necessary fields, click "Done" to save the Screen element.

#### 8. Connect the Elements:

- Ensure that the Screen element is connected properly to the previous Decision element on the "Appointment" path.
- Connect this Screen element to the next elements in your flow after users complete the appointment booking.

## E)Add GET Record Element

## Add and Configure the GET Records Element

### 1. Open Your Flow:

- Navigate to Flow Builder where your flow is being edited.

### 2. Add GET Records Element:

- Click on the "+" icon in the Flow Builder to add a new element.
- Select "Get Records" from the list of element types.

### 3. Configure GET Records Element Properties:

- Label:
  - a. Enter "Get Consultant Rec" for the label.
- API Name:
  - This auto-generates but can be customized if needed.

### 4. Select Object:

- For Object, select "Consultant."

### 5. Set Condition Requirements:

- Condition Requirement: Select "All Conditions Are Met (AND)." This means all specified conditions must be true for the record to be retrieved.

New Get Records

\* Label: Get Consultant Rec

\* API Name: Get\_Constant\_Rec

Description

Filter Consultant Records

Condition Requirements: All Conditions Are Met (AND)

Field	Operator	Value
Name	Equals	AppointmentRecordRes > Con...

+ Add Condition

Sort Consultant Records

Sort Order: Not Sorted

How Many Records to Store: ☒ Only the first record, ☐ All records

## 6. Define Filter Conditions:

- Field: Select "Name."
- a. Operator: Equals
  - Value: Enter `{!AppointmentRecordRes.Consultant_Name__c}`
    - This value references the field in the Appointment record variable that contains the Consultant's name.

## 7. Store the Record Data:

- How Many Records to Store:
  - a. Choose "Only the first record" if you expect to retrieve a single Consultant record.
    - Choose "All records" if you expect multiple records and want to process them all (usually "Only the first record" is sufficient for a single consultant).
- Store Record Data In:
  - Create a new variable to store the retrieved Consultant record.
  - Click on "New Resource."
    - Resource Type: Variable
    - API Name: ConsultantRecordRes
    - Data Type: Record
    - Object: Consultant
    - Allow multiple values (Collection): No (for single record)
    - Available for Input: No (usually no input is required)
    - Available for Output: Yes (if you need to use this variable later in the flow)
  - Click "Done" to create the variable.
  - Select the newly created variable for storing the record data.

## 8. Connect the Elements:

- Ensure that the GET Records element is connected to the appropriate path in

your flow after the Decision element under the Appointment path.

- Connect the GET Records element to subsequent actions or elements in the flow that need to use the retrieved Consultant data.

9. Save the GET Records Element:

- Click "Done" to save the configuration of the GET Records element.
- Label: Appointment Booking Screen
- Fields: Include fields from the Appointment object.

Test and Activate the Flow:

1. Save and Test:

- Click "Save" to save your changes.
- Click "Debug" to test the flow and verify that the GET Records element correctly retrieves the Consultant record based on the provided name.

2. Activate the Flow:

- Once you have tested and validated that the GET Records element functions as expected, click "Activate" to make the flow live.

## **F)Create Appointment Record using Create Records Element**

Add and Configure the Create Records Element

1. Open Your Flow:

- Navigate to Flow Builder where your flow is being edited.

2. Add Create Records Element:

- Click on the "+" icon in the Flow Builder to add a new element.
- Select "Create Records" from the list of element types.

Edit Create Records  
Create Appointment (*Create\_Appointment*)

\* Label

Create Appointment

\* API Name

Create\_Appointment

Description

Field	Value	
Appointment_DateTime__c	←  AppointmentRecordRes > Appointment Date/Ti... X	
Consultant__c	←  Consultant from Get_Consultant_Rec > Record ID X	
Notes__c	←  AppointmentRecordRes > Notes X	
PurposeTopic__c	←  AppointmentRecordRes > Purpose/Topic X	
Student_Name__c	←  Student from Get_Rec > Record ID X	

### 3. Configure Create Records Element Properties:

- Label:
  - Enter "Create Appointment" for the label.
- API Name:
  - a. This auto-generates but can be customized if needed.

### 4. Configure the Record Creation:

- How Many Records to Create:
  - a. Select "One" if you are creating a single record.
- How to Set the Record Fields:
  - Select "Use separate resources, and literal values" to manually specify field values.

### 5. Specify Object:

- Object: Select "Appointment."

### 6. Set Field Values:

- Configure the field values by specifying each field and its corresponding value:
- Field:Appointment\_DateTime\_\_c

- Value: {!AppointmentRecordRes.Appointment\_DateTime\_\_c}
  - Field:Consultant\_\_c
  - Value: {!Get\_Consultant\_Rec.Id}
2. This retrieves the Consultant record ID from the GET Records element "Get Consultant Rec."
    - Field:Notes\_\_c
      - Value: {!AppointmentRecordRes.Notes\_\_c}
      - Field:PurposeTopic\_\_c
      - Value: {!AppointmentRecordRes.PurposeTopic\_\_c}
      - Field:Student\_Name\_\_c
      - Value: {!Get\_Rec.Id}
    1. This retrieves the Student record ID from the GET Records element "Get Rec."
7. Connect the Elements:
    - Ensure that the "Create Appointment" element is connected to the previous element, which should be the "Get Consultant Rec" element.
    - Connect this element to the next steps in your flow where the newly created appointment record will be processed or used.
  8. Save the Create Records Element:
    - Click "Done" to save the configuration of the Create Records element.

## **G)Add Screen Element**

### Add and Configure the Screen Element

1. Open Your Flow:
  - Navigate to Flow Builder where your flow is being edited.
2. Add Screen Element:
  - Click on the "+" icon in the Flow Builder to add a new element.
  - Select "Screen" from the list of element types.

3. Configure Screen Properties:
  - Label: Enter "Confirmation Screen."
  - API Name: (Auto-generates but can be customized if needed.)
  - Help Text: (Optional) Provide any additional instructions if necessary.
4. Add Display Text Component:
  - From the left side panel, search for "Display Text."
  - Drag the "Display Text" component onto the main panel of the screen.
5. Configure Display Text Component:
  - Label: Enter "Appointment\_Confirmation."
  - API Name: (Auto-generates but can be customized if needed.)
6. Enter Display Text Content:
  - In the Text field of the Display Text component, paste the following content:

## **H)Add an SubFlow Element**

### Add and Configure the Subflow Element

1. Open Your Flow:
  - Navigate to Flow Builder where your flow is being edited.
2. Add Subflow Element:
  - Click on the "+" icon in the Flow Builder to add a new element.
  - Select "Subflow" from the list of element types.
3. Configure Subflow Properties:
  - Label: Enter "Create Student Case" for the label.
  - API Name: (This auto-generates but can be customized if needed.)
4. Select the Subflow:
  - Click on the "Subflow" field to search for and select the existing subflow named "Create a Case."
    - Ensure that you have already created the subflow "Create a Case" in

Flow Builder. If not, you will need to create it beforehand.

5. Pass Variables to the Subflow (if needed):

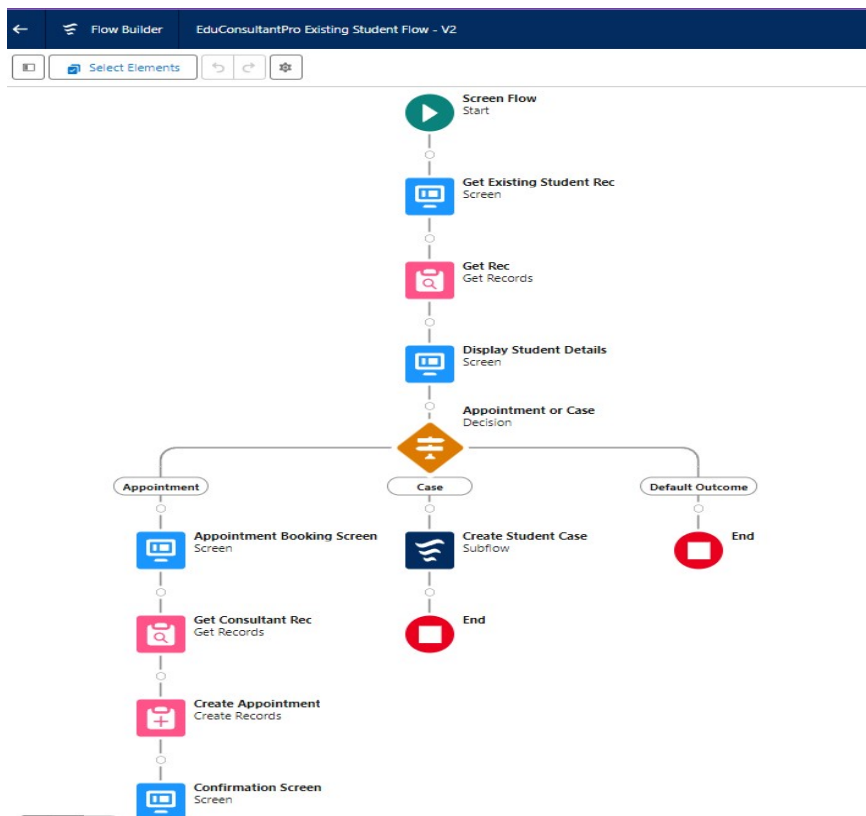
- If the subflow “Create a Case” requires input variables, you will need to pass those variables from the current flow.
- Map the required fields from the current flow to the subflow variables:
  - Input Variables: Map fields from your current flow to the subflow’s input variables.
  - Output Variables: If the subflow returns any output variables, you can map them to resources in the parent flow.

6. Connect the Elements:

- Ensure that the Subflow element is connected to the appropriate path in your flow after the Decision element under the "Case" path.

7. Save the Subflow Element:

- Click "Done" to save the configuration of the Subflow element.





## Save and Label the Flow

- a. Save the Flow:
  - Click on "Save" in the top-right corner of the Flow Builder.
- b. Label the Flow:
  - Flow Label: Enter "EduConsultantPro Existing Student Flow."
  - Flow API Name: This auto-generates but can be customized if needed.
  - Ensure all changes are saved.
- c. Activate the Flow (if ready):
  - Once the flow is fully configured and tested, click "Activate" to make it live.

## TASK 7:

### Create a ScreenFlow to Combine all the flows at one place

#### A)Add Screen Element

#### Add and Configure the Screen Element

1. Open Your Flow:
  - Navigate to Flow Builder where your flow is being edited.
2. Add Screen Element:
  - Click on the "+" icon in the Flow Builder to add a new element.
  - Select "Screen" from the list of element types.
3. Configure Screen Properties:
  - Label: Enter "Welcome Screen" for the label.
  - API Name: (This auto-generates but can be customized if needed.)

- Help Text: (Optional) Provide any additional instructions if necessary.
4. Add Display Text Component:
    - From the left side panel, search for the "Display Text" component.
    - Drag the "Display Text" component onto the main panel of the screen.
  5. Configure Display Text Component:
    - Label: Enter "SuccessMessage."
    - API Name: (This auto-generates but can be customized if needed.)
  6. Enter Display Text Content:
    - In the Text field of the Display Text component, paste the following content:
    - plaintext
    - `Copy code`
    - 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!

1. Save the Screen Element:

- Click "Done" to save the configuration of the Screen element.

2. Connect the Elements:

- Connect this "Welcome Screen" element to the appropriate place in your flow, typically at the beginning or where you want users to see the welcome message.

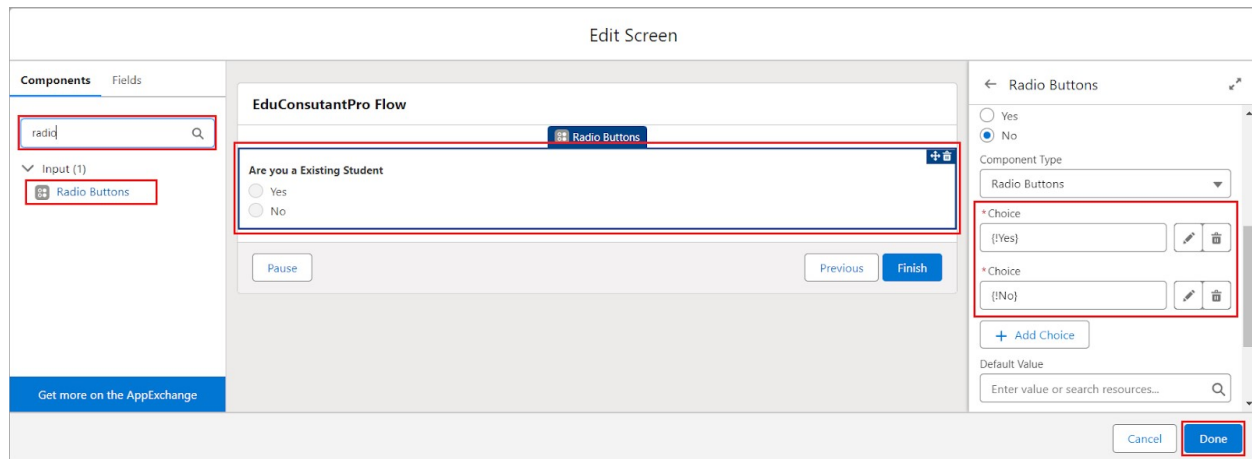
## Test and Activate the Flow

1. Save and Test:

- Click "Save" to save your changes.
- Click "Debug" to test the flow and ensure that the welcome screen displays correctly and the message appears as expected.

2. Activate the Flow:

- Once testing is complete and the flow works as intended, click "Activate" to make the flow live.



## B)Add Decision Element

Add and Configure the Decision Element

1. Open Your Flow:
  - Navigate to Flow Builder where your flow is being edited.
2. Add Decision Element:
  - Click on the "+" icon in the Flow Builder to add a new element.
  - Select "Decision" from the list of element types.
3. Configure Decision Element Properties:
  - Label: Enter "Decision 1" for the label.
  - API Name: (This auto-generates but can be customized if needed.)
  - Help Text: (Optional) Provide any additional instructions if necessary.
4. Define Outcomes:
5. Outcome 1: If Existing Student
  - Outcome Label: Enter "If Existing Student."
  - Conditions:
    - a. Resource: `{!Are_you_a_Existing_Student}`
      - Operator: Equals
      - Value: `{!Yes}`

- Click on the “+” icon to add another outcome.
- 6. Outcome 2: If Not Existing Student
  - Outcome Label: Enter "If Not Existing Student."
  - Conditions:
    - a. Resource: {!Are\_you\_a\_Existing\_Student}
      - Operator: Equals
      - Value: {!No}
- 7. Connect the Elements:
  - Connect the "Decision 1" element to the "Existing or New Student Confirmation Screen" element.
  - Ensure that the outcomes of the Decision element are connected to the appropriate subsequent elements or paths in your flow.
- 8. Save the Decision Element:
  - Click "Done" to save the configuration of the Decision element.


## **D)Add an SubFlow Element**

1. Open Your Flow:
  - Navigate to Flow Builder where your flow is being edited.
2. Add Subflow Element:
  - Click on the "+" icon in the Flow Builder to add a new element.
  - Select "Subflow" from the list of element types.
3. Configure Subflow Properties:
  - Label: Enter "Existing Student Flow" for the label.
  - API Name: (This auto-generates but can be customized if needed.)
4. Select the Subflow:
  - Click on the "Subflow" field to search for and select the existing subflow named "EduConsultantPro Existing Student Flow."
  - Ensure that the subflow “EduConsultantPro Existing Student Flow” is already created in your

environment. If not, create it beforehand.

5. Pass Variables to the Subflow (if needed):

- If the subflow “EduConsultantPro Existing Student Flow” requires input variables, map those variables from the current flow.
- Input Variables: Map fields from your current flow to the subflow’s input variables.
- Output Variables: If the subflow returns any output variables, map them to resources in the parent flow.

 Edit Subflow  
New Student Flow (*New\_Student\_Flow*)

\* Label

New Student Flow


\* API Name


New\_Student\_Flow

Description

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

Referenced Flow

 **EduConsultantPro Student Flow**

Open Referenced Flow 

6. Connect the Elements:


- Ensure that the "Existing Student Flow" Subflow element is connected to the "If Existing Student" path of the "Decision 1" element.

7. Save the Subflow Element:

- Click "Done" to save the configuration of the Subflow element.

## Save and Label the Flow

1. Save the Flow:
  - Click on "Save" in the top-right corner of the Flow Builder.
2. Label the Flow:
  - Flow Label: Enter "EduConsultantPro Existing Student Flow."
  - Flow API Name: This auto-generates but can be customized if needed.
  - Ensure all changes are saved.
3. Activate the Flow (if ready):
  - Once the flow is fully configured and tested, click "Activate" to make it live.

 Edit Subflow  
New Student Flow (*New\_Student\_Flow*) ✕

\* Label

New Student Flow


\* API Name


New\_Student\_Flow

Description

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

Referenced Flow

 **EduConsultantPro Student Flow**

Open Referenced Flow 

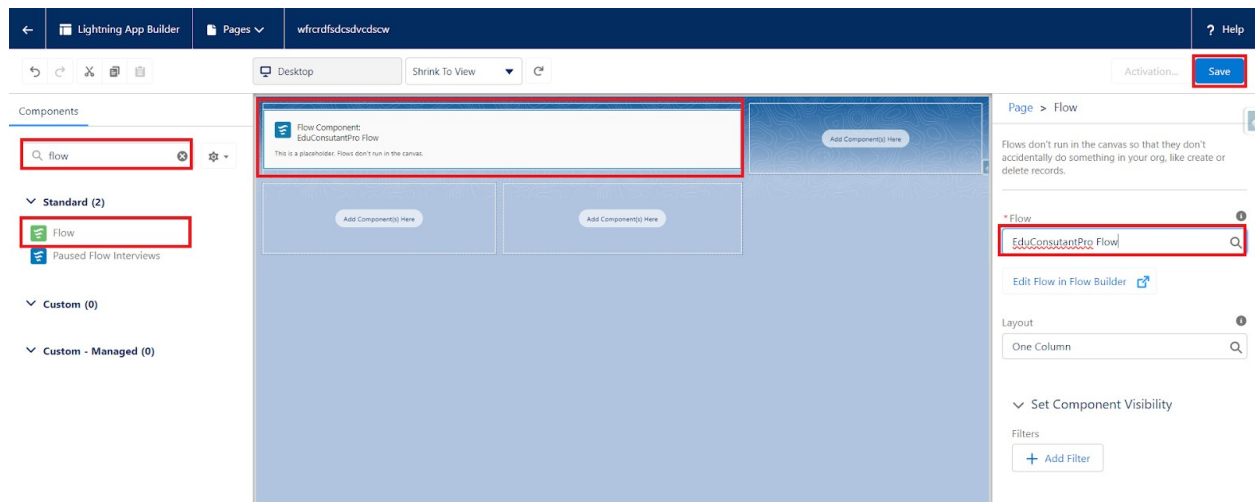
## TASK 8:

### Create a lightning app page

1. Access Lightning App Builder:
  - From Setup, enter "App Builder" in the Quick Find box.
  - Click on "Lightning App Builder" from the search results.
2. Create a New Page:
  - Click the "New" button to start creating a new page.
3. Select Page Type:
  - Select "Home Page" as the type of page you want to create.
  - Click "Next."
4. Configure Page Properties:
  - Page Name: Enter "EduConsultPro Home Page."
  - Template: Select the "Standard Home Page" template.
  - Click "Done" to proceed.
5. Add Flow Component:
  - In the Lightning App Builder, you'll be on the page layout editor.
  - Locate the "Flow" component in the left-hand component panel.
  - Drag the "Flow" component to the top-right region of the page layout.
6. Configure Flow Component:
  - Click on the "Flow" component that you just dragged onto the page.
  - In the properties panel on the right, search for the "EduConsultantPro Flow."
  - Select the "EduConsultantPro Flow" from the list.
7. Save the Page:
  - Click "Save" in the top-right corner of the page layout editor.
8. Activate the Page:
  - Click "Activate" to make the page available to users. You will need to set visibility options for the page:



- App Default: Choose which apps will have access to this page.
- Profile: Set which user profiles will have access.
- App and Profile Assignment: Optionally, configure specific app and profile assignments.



**THANK YOU**