

A CRM Application to Manage the Services offered by an Institution

Scope of the Project:

The project aims to implement Salesforce CRM to revolutionize the admission and enquiry processes at EduConsultPro Institute. It will involve automating the entire admission workflow, from managing prospective student inquiries to finalizing enrollments. The CRM will serve as a centralized system, enabling admissions staff to efficiently handle applications, track enquiries, and manage case interactions. Students will benefit from a seamless application experience, real-time updates, and personalized support. The project scope also includes integrating Salesforce's analytics features to provide insights into admission trends and decision-making. Additionally, the system will facilitate smooth communication between students and the admissions team, ensuring transparency throughout the process.

Objective of the Project

The primary objective is to streamline EduConsultPro Institute's admission and enquiry management by leveraging Salesforce CRM. It aims to enhance the student experience by simplifying application processes, ensuring prompt query resolution, and delivering transparent updates. The project seeks to empower admissions staff with tools for efficient application review, data tracking, and case management, reducing manual tasks and errors. Furthermore, the objective is to increase the institute's operational efficiency and student satisfaction through improved process clarity and faster turnaround times. By implementing this solution, EduConsultPro will establish itself as a technologically advanced institution, meeting the growing demands of prospective students and stakeholders.

Admission Application Management

Students can submit detailed admission applications via the institute's portal, with all data stored in Salesforce CRM. Automated email notifications confirm submissions, and staff gain insights through dashboards analyzing application metrics and trends.

Approval Process Requirements

An automated approval process reviews consulting requests, with email alerts for approvals or rejections. Requests are automatically submitted upon creation for seamless processing.

Consulting Services Management

Students can request consulting services through the portal, with details stored in Salesforce CRM. Consultants receive notifications, manage requests, and schedule appointments directly within Salesforce, with status tracking for all interactions.

Immigration Case Management Students can submit immigration cases through various channels, with details stored in Salesforce CRM. Agents track and process cases with integrated tools for document management and collaboration, ensuring streamlined updates on case status.

What is Salesforce?

Salesforce is a leading cloud-based customer relationship management (CRM) platform that

helps businesses manage and optimize interactions with customers, clients, and stakeholders. It provides a suite of tools and services to streamline sales, marketing, customer service, and business operations. Salesforce enables organizations to collect, store, and analyze customer data, automate workflows, and enhance collaboration across teams.

With customizable applications, Salesforce caters to a wide range of industries, offering solutions for everything from lead management and marketing automation to analytics and AI-driven insights. It is known for its flexibility, scalability, and integration capabilities, making it a preferred choice for businesses of all sizes seeking to improve efficiency and customer engagement.

Creating Developer Account:

Before the start of the project development, we need to create an account in Salesforce developer edition. This involves the following steps:

Creating a developer org in Salesforce.

1. Go to <https://developer.salesforce.com/signup>
2. On the sign up form, enter the following details :
3. First name & Last name
4. Email
5. Role : Developer
6. Company : College Name
7. Country : India
8. Postal Code : pin code
9. Username : should be a combination of your name and company This need not be an actual email id, you can give anything in the format : username@organization.com

Click on sign me up after filling these.

developer.salesforce.com/signup

Sign up for your Salesforce Developer Edition
A Salesforce Platform environment for free.

Complete the form to get access to the Salesforce Developer Edition.

First name*
Your first name

Last name*
Your last name

Email*
Your email address

Role*
Your job role

Company*
Company Name

Country/Region*
Country/Region

Postal Code*
Your postal code

Username*
[email@company.sandbox]
Your username must be in the form of an email address. It does not have to be real. It must be unique and cannot be associated with another Salesforce login credential. Read more about username recommendations.

☐ I agree to the Main Services Agreement - Developer Services and Salesforce Program Agreement.

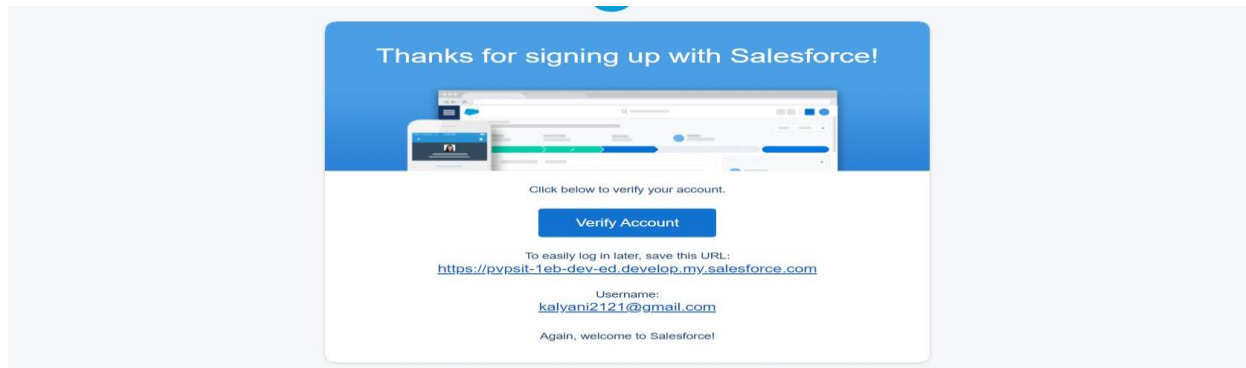
Sign me up

Build enterprise-quality apps fast to bring your ideas to life

- Build apps fast with drag and drop tools
- Customize your data model with clicks
- Go further with Apex code
- Integrate with anything using powerful APIs
- Stay protected with enterprise-grade security
- Customize UI with clicks or any leading-edge web framework

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.



MileStone 1: Create Objects from Spreadsheet

What Is an Object?

Salesforce objects are database tables that permit you to store data that is specific to organization. What are the types of Salesforce objects Salesforce objects are of two types:

1. **Standard Objects:** Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. **Custom Objects:** Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

Create Course object

1. Go to your object manager and and click on create object from spreadsheet
2. Click on the link to get the spreadsheet, Course.

A	B	C	D	E
Course Name	Description	Start Date	End Date	Instructor
IELTS	Let's Learn IELTS	03-01-2024	05/31/2024	Sandeep
GRE	Let's Learn GRE	03-01-2024	07-11-2024	Shivam
TOEFL	Let's Learn TOEFL	04-01-2024	07/27/2024	Prajwal
DuoLingo	Let's Learn DuoLingo	04-01-2024	09/14/2024	Sameer
GMAT	Let's Learn GMAT	04-01-2024	01-09-2024	Sanjay

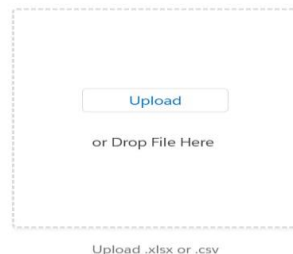
Download this spreadsheet from the link in .csv format.

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

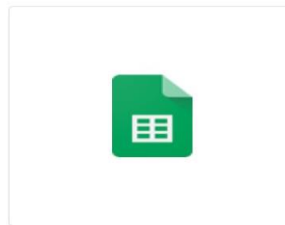
Create a custom object from a spreadsheet

Select a spreadsheet

Select a source for your new object data.



Upload .xlsx or .csv



Google Sheet



Office 365 or Drive

Click next and next , finally save the data.



Create a custom object from a spreadsheet

Define object and fields

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

CSV File Details

Encoding Format: Values Separated By: Field Label Source: ☐ Enter manually ☒ Detect from row * Field Labels Row: Import 5 rows of Data?: ☐ No, skip import ☒ Yes, import data Record Name Field:

Fields 5 of 5 to import ☐ Hide mapped fields

IMPORT FILE FIELD NAME	Salesforce FIELD NAME	Salesforce FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
✓ Course Name	Course Name	Text	<input checked="" type="checkbox"/>	IELTS
✓ Description	Description	Text	<input checked="" type="checkbox"/>	Let's Learn IELTS
✓ Start Date	Start Date	Phone	<input checked="" type="checkbox"/>	03-01-2024
✓ End Date	End Date	Date	<input checked="" type="checkbox"/>	05/31/2024
✓ Instructor	Instructor	Text	<input checked="" type="checkbox"/>	Sandeep

Back

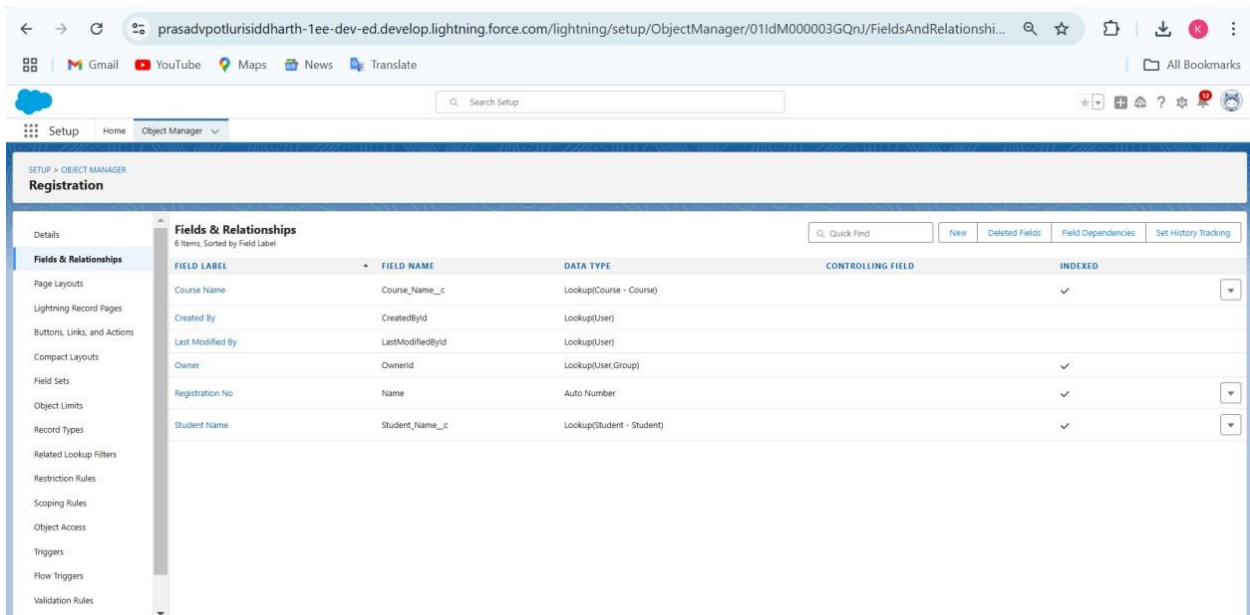
Follow the steps which we have followed for course object creation to create the remaining objects .Use the excel sheet data given in the module for following objects.

- Consultant
- Student
- Appointment

Create Relationship among the objects

- Create lookup between appointment and student, appointment and consultant.For that open object manager and search for appointment object and select it. Open fields and relationships and select New and then create the loop up relationship.
- Create an object to store the information student and course details with the name Registration.Refer the data model for more information about the fields and relationships of

object Registration.

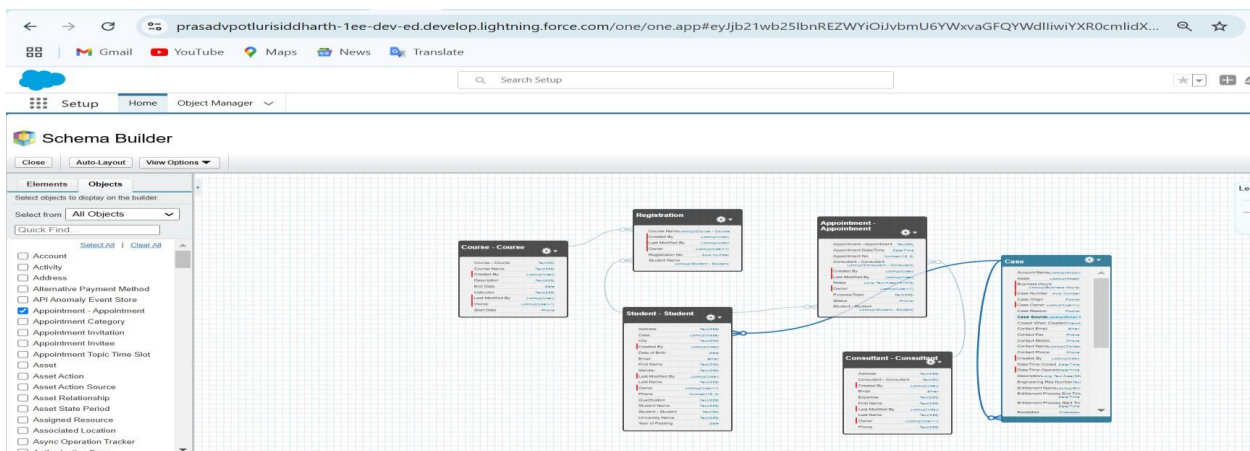


The screenshot shows the Salesforce Object Manager interface for the 'Registration' object. The 'Fields & Relationships' tab is active, displaying a table of fields. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are: Course Name (Lookup(Course - Course)), Created By (Lookup(User)), Last Modified By (Lookup(User)), Owner (Lookup(User, Group)), Registration No (Auto Number), and Student Name (Lookup(Student - Student)).

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course Name	Course_Name__c	Lookup(Course - Course)		✓
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User, Group)		✓
Registration No	Name	Auto Number		✓
Student Name	Student_Name__c	Lookup(Student - Student)		✓

3. Also create a lookup between student and case to store the student queries for immigration or visa application.

4. The data model should be similar to the below Data Model with fields & relationships:



Configure the Case Object

1. Go to object manager, edit case object.
2. Select the "Type" field and add the values in it.
 - Immigration
 - Visa Application

Setup > OBJECT MANAGER

Case

Details

Fields & Relationships

Case Page Layouts

Case Close Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Hierarchy Columns

Inactive picklist values: 0

Field Dependencies

No dependencies defined.

Validation Rules

No validation rules defined.

Case Type Picklist Values

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	Mechanical	Mechanical	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Electrical	Electrical	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Electronic	Electronic	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Structural	Structural	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Other	Other	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Immigration	Immigration	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 19/11/2024, 10:21 pm
Edit Del Deactivate	Visa Application	Visa Application	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 19/11/2024, 10:22 pm

Inactive Values

No inactive Values values defined.

1. Now Select the “Status” field and add the values in it.
2. Open
3. In-progress.

Setup > OBJECT MANAGER

Case

Details

Fields & Relationships

Case Page Layouts

Case Close Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

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	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Working	Working	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Escalated	Escalated	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Closed	Closed	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 18/11/2024, 1:13 pm
Edit Del Deactivate	Open	Open	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 19/11/2024, 10:23 pm
Edit Del Deactivate	In-progress	In-progress	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	kalvani.maddala, 19/11/2024, 10:23 pm

Inactive Values

No inactive Values values defined.

Before stepping into creation of a lightning app it is important to create the tabs for the objects that we created earlier.

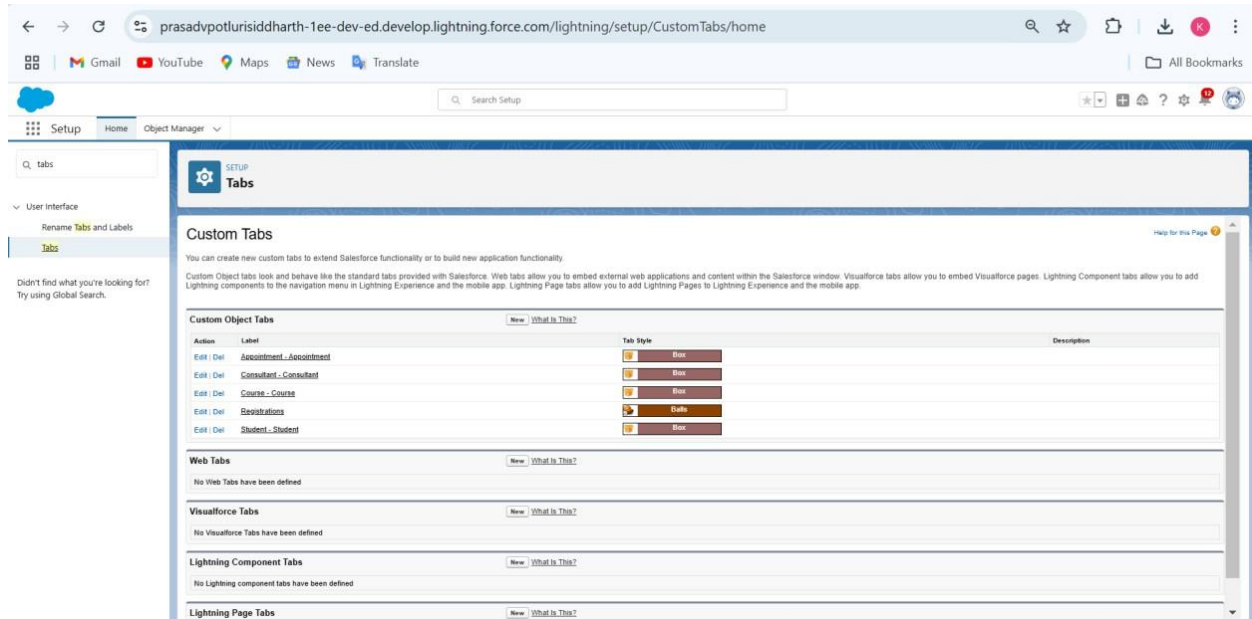
Tabs

What is Tab : A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Creating a Custom Tab

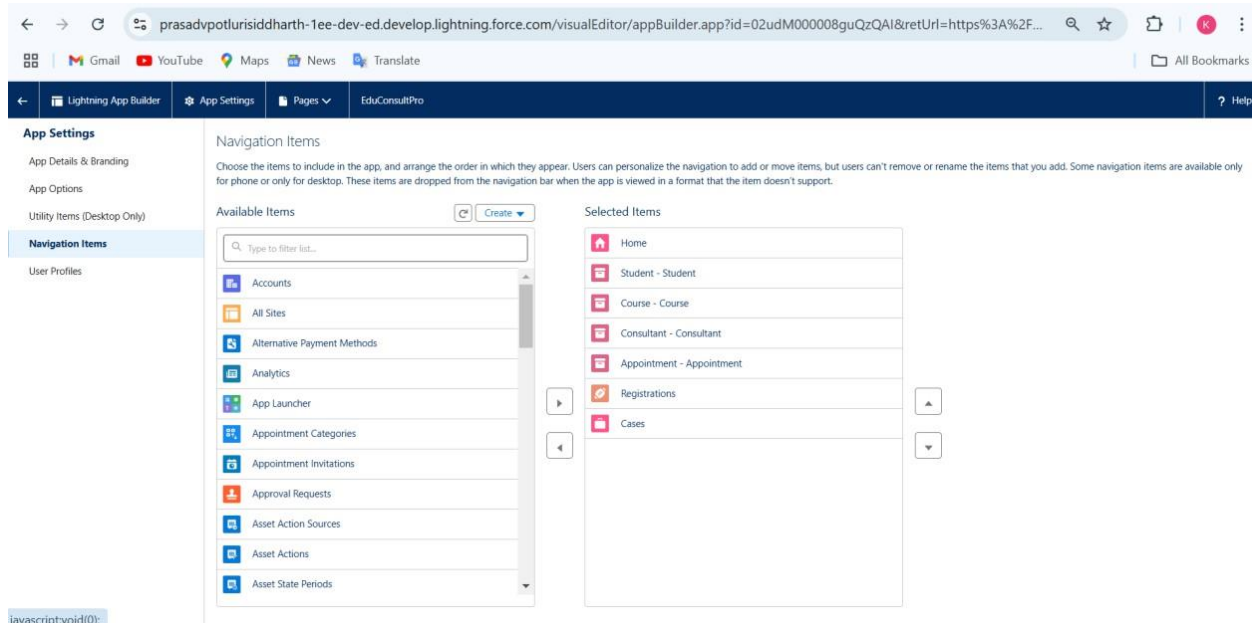
1. Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)

2. Select Object>> Select the tab style >> Next (Add to profile page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
3. Make sure that the Append tab to users' existing personal customizations is checked.
4. Click save.



Create a Lightning App

1. Go to Setup, search for the App Manager in quick find
2. Click on New Lightning App
3. Give app name as “EduConsultPro”, click Next, Next, Next
4. Add Home, Students, Courses, Consultants, Appointments, Registrations, and Cases from the Available Items to Selected Items.

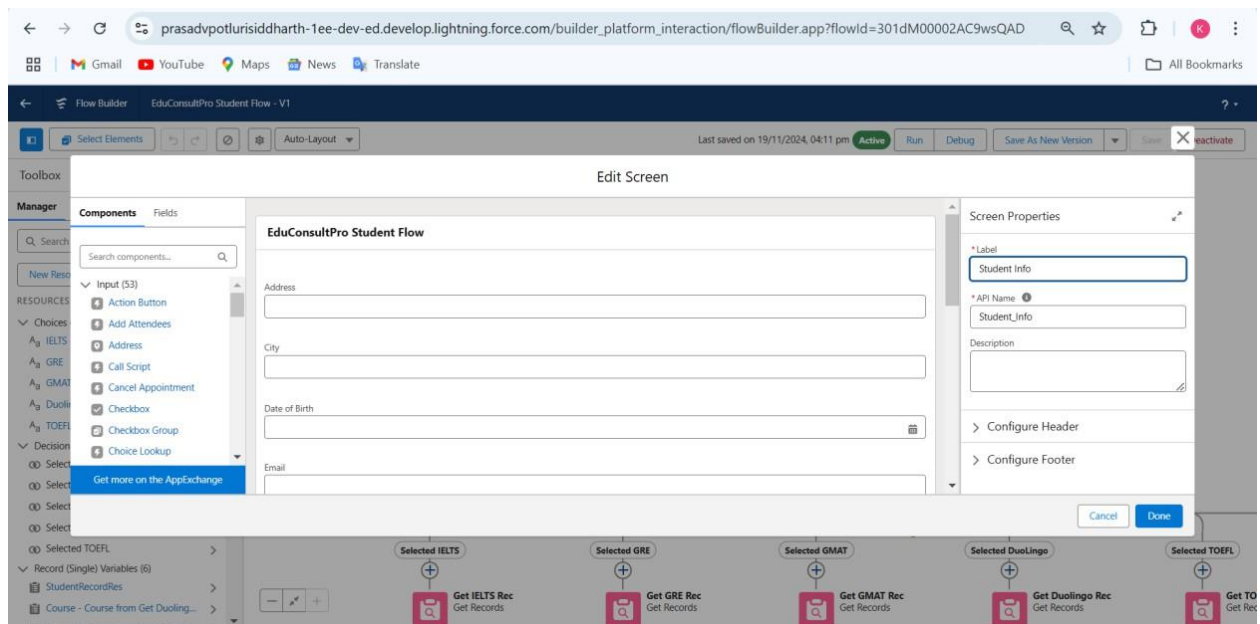


5. Add “System Administrator” profile from Available Profiles to Selected Profiles, click Save & Finish. A Lightning app with name EduConsultPro is created successfully.

MileStone 2 :Create a ScreenFlow for Student Admission Application process.

Add Screen Element

1. From Setup, enter Flow Builder in quick find, select new flow --> ScreenFlow.
2. Add a Screen element.
3. In the Screen Properties pane, for Label, enter “Student Info”.
4. Click on Fields, click on the record variable input and create a new Resource(StudentRecordRes) to display all the fields which are in the student object. Drag all the fields which are needed to add on the screen in order to collect the student information.




Create Student Record using Create Element

1. Add a Create element after Student Info Screen Element, Label it as “Create Student Record.”


2. Select “one” under How many records to Create, and select “use all values from a record” under How to Set the record fields.
3. Select the record variable resource(StudentRecordRes) which we have created in the Student Info screen element, under Create a record from these values.

Last saved on 19/11/2024, 04:11 pm **Active** [Run](#) [Debug](#) [Save As New Version](#) [Save](#) [Deactivate](#)

 Create Records ×

* Label


Create Student Record

* API Name 

Create_Student_Record

Description

* How to set record field values

From a Record Variable 


How Many Records to Create


☒ One


☐ Multiple


Create a Record from These Values

* Record

 StudentRecordRes ×

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

Update Existing Record 

 Disabled

Add Screen Element

1. Add a Screen Element after Create Student Record Element and label it as Course Screen.
2. Add a picklist component from the left side panel label it as “Select Course”, under choices type “IELTS” and enter. This creates a variable with the name IELTS.
3. Repeat the same for GRE, GMAT, Duolingo, TOEFL.

Last saved on 19/11/2024, 04:11 pm

Active

Run

Debug

Save As New Version



Save

Deactivate



Create Records



* Label

Create Student Record

* API Name

Create_Student_Record

Description

* How to set record field values

From a Record Variable



How Many Records to Create



One



Multiple

Create a Record from These Values

* Record

StudentRecordRes

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

Update Existing Record



Disabled

Add Decision Element

1. Add a Decision Element after Select Course Screen Element, label it as Selecting Course.
2. Under outcome label it as "Selected IELTS" and write the condition such as below: Resource : Select_Course (Screen Component from Select Course Screen Element) Operator : Equals Value : IELTS (Choice Variable from Select Course Screen Element)
3. Click on the "+" icon and Repeat step 2 for other options mentioned as below: a) GRE b) GMAT c) DuoLingo d) TOEFL
4. Click Done.

Last saved on 19/11/2024, 04:11 pm
Active
Run
Debug
Save As New Version
Save
Deactivate

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
Selected GRE
Selected GMAT
Selected Duolingo
Selected TOEFL
Default Outcome

* Label
Selected IELTS
* Outcome API Name
Selected_IELTS
Condition Requirements to Execute Outcome
All Conditions Are Met (AND)
Resource
Course Screen > Select Course
Operator
Equals
Value
IELTS
+ Add Condition

Add GET Record Element

1. Add a GET Record Element after Decision Element, under the IELTS path and label it as “Get IELTS Rec”.
2. Select Object : Course Condition Requirement : All Conditions are Met(AND) Field : Course Name Operator : Equals Value : {!Select_Course}

Last saved on 19/11/2024, 04:11 pm

Active

Run

Debug

Save As New Version



Save

Deactivate



Get Records



* Label

Get IELTS Rec

* API Name

Get_IELTS_Rec

Description

Get Records of This Object

* Object

Course - Course

Filter Course - Course Records

Condition Requirements

All Conditions Are Met (AND) ▼

Field

Course_Name_c

Operator

Equals ▼

Value

Select_Course x



+ Add Condition

Create Registration Record using Create Records Element

1. Add a Create element after the Get IELTS Rec element and label it as "Create IELTS Registration Rec".
2. Select "one" under How many records to Create, and select "Use separate resources, and literal values" under How to Set the record fields.
3. Select Object : Registration Field : Course_Name_c Value : {!Get_IELTS_Rec.Id} Field : Student_Name_c Value : {!StudentRecordRes.Id}



Create Records



* Label

Create IELTS Registration Rec

* API Name

Create_IELTS_Registration_Rec

Description

* How to set record field values

Manually

Create a Record of This Object

* Object

Registration

Set Field Values for the Registration

Field

Course Name

Value

Course - Course from Get IELTS Rec > Record ID



Field

Student Name

Value

StudentRecordRes > Record ID



Add Field

Create Email Text Template Variables for email body and subject

1. Click on the toggle toolbox on the left corner, click “New Resource”, then select “Text Template” as Resource Type.
2. Give the API name as “StuRegistrationEmailTextTempBody”, select “view as plain text” and paste the message text in body as below and follow same to create for subject.

Add an Action Element

1. Add an Action Element after all the Decision paths, label it as “Send Email to Student”. 2. Under “Set input values for selected action”, include body, Recipient Address List and Subject.
3. For input Body : {!StuRegistrationEmailTextTempBody}, Recipient Address List : {!StudentRecordRes.Email__c}, Subject : {!StuRegistrationEmailTextTempSub}.



Send Email



!8Q 

 Send Email to Studen

..APIName 0

[Send_Email_to_Student

Description:



Send Email 0

emailSimple-emailSimple

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

Set Input Values for the Selected Action

CD Add Threading Token to Body



Nat Excluded

CD Add Threading Token to Subject



Nat Included

Aa BCC Recipient Address List



Nat Included

/2024, 04:11 pm

Active

Run

Debug

Save As New Version

Save

Deactivate



Send Email



Recipient Address Collection	Not Included
Recipient Address List	<input checked="" type="checkbox"/> Included
<div>StudentRecordRes > Email </div>	
Recipient ID	<input type="checkbox"/> Not Included
Related Record ID	<input type="checkbox"/> Not Included
Rich-Text-Formatted Body	<input type="checkbox"/> Not Included
Sender Email Address	<input type="checkbox"/> Not Included
Sender Type	<input type="checkbox"/> Not Included
Subject	<input checked="" type="checkbox"/> Included
<div>StuRegistrationEmailTextTempSub </div>	
Use Line Breaks	<input type="checkbox"/> Not Included

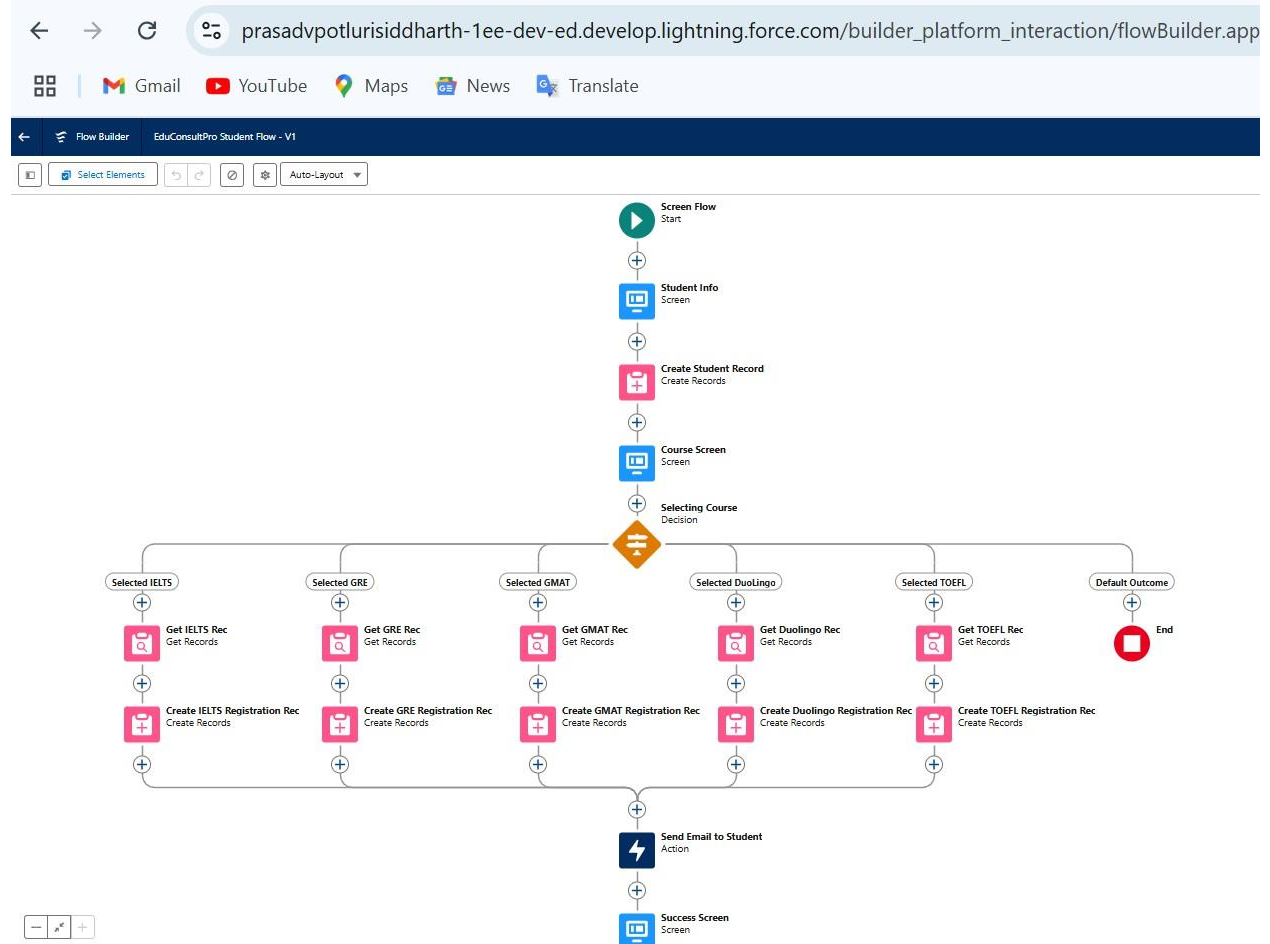
Add Screen Element

1. Add a Screen Element after the Send Email to Student Action Element, label it as Success Screen.
2. From the left side panel search for the Display text component and drag it to the main panel, label it as "SuccessMessage".
3. Paste the below in the Resource picker box. "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.”

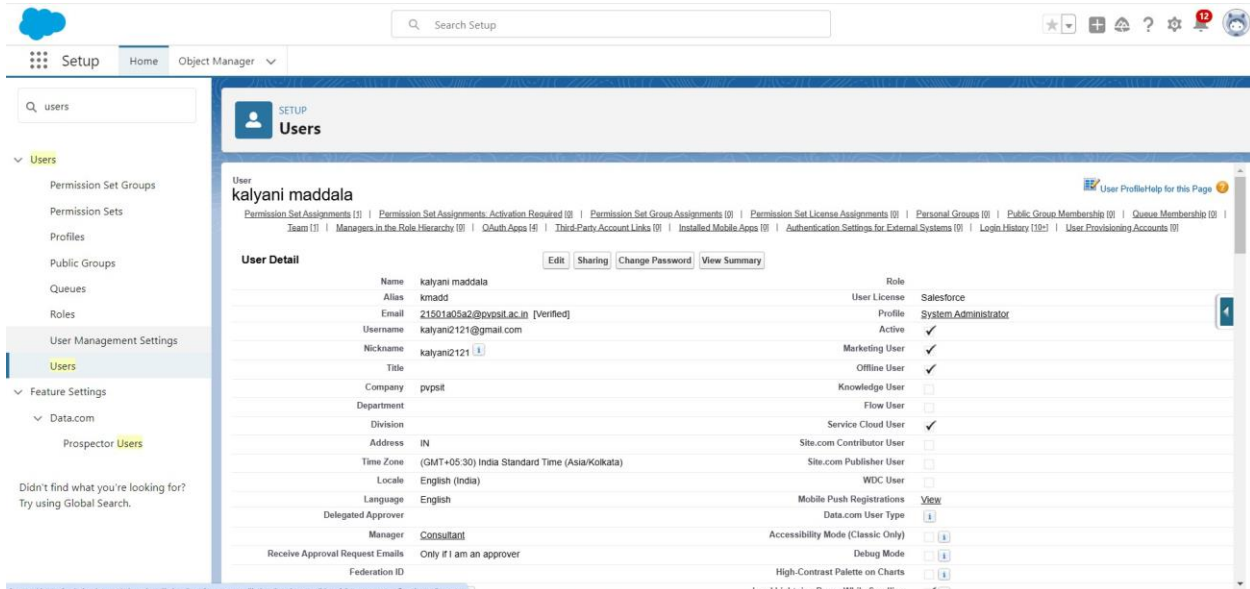
4. Click Done.

5. Save the flow and name it as “EduConsultPro Student Flow”. Your flow will look as shown below:



MileStone 3:Create Users User

1. Go to Setup --> Administration --> Users --> New User
2. LastName : Consultant
3. License : Salesforce Platform
4. Profile : Standard Platform User
5. Fill all the mandatory fields & Save.

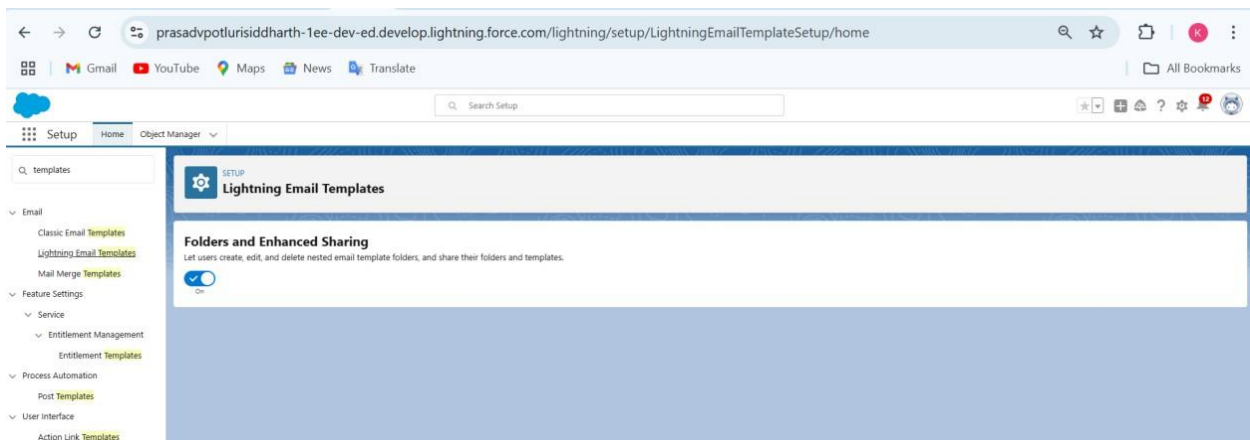


Configure the User Settings

1. Go to Setup --> Administration --> Users --> click Edit next to your name
2. Scroll down to bottom, under Approver Settings, Select "Consultant" the Manager Field.
3. Click Save.

MileStone 4: Create an Approval Process for Property Object

From Setup, enter Templates in the Quick Find box, and then select Lightning Email Templates, toggle on.



1. go to app launcher, search for "Email Templates", Create a new folder with the desired name. Here I named it as emails.
2. Then create a new email template, select the folder which we have created in the previous steps, enter the below text in the HTML Value and Save it as "Submission Template".
 "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}}},

ConsultantName:{{{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 lead to positive outcomes and progress toward your goals.

If you have any questions or require further information before our scheduled appointment, please don't hesitate to reach out to me.

Looking forward to our meeting.

Best regards,

{{{Recipient.Name}}},

EduConsultantPro"

Recently Viewed Email Templates

Name	Description	Template Type
Approval Template		Custom
Rejection Template		Custom
Submission Template		Custom

© 2000-2014 iMentorPro.com. All rights reserved.

Create an Approval Process

1. From Setup, enter Approval in the Quick Find box, and then select Approval Processes.
2. In Manage Approval Processes For, select Appointment.
3. Click Create New Approval Process --> Use Jump Start Wizard.
4. Configure the approval process.
5. Process Name - Appointment Approval, Under Select Approver, Select Manager for the option : "Automatically assign an approver using a standard or custom hierarchy field."
6. Click next and "Next Automated Approver Determined By" --> Select Manager.
7. From Record Editability Properties --> Click on Administrators OR the currently assigned approver can edit records during the approval process.
8. Save the approval process.
9. Click View Approval Process Detail Page.
10. Under Initial Submission Actions, click Add New --> Field Update, and configure it with these

values.

Field Value

Name Submitted

Field to Update Appointment: Status

11. click Add New --> Email Alert, and configure it with these values. Description : Submission
Email Alert Unique Name : Auto Populates Email Template : Submission Template Recipient
Type : Select your Name
12. Repeat the Steps 10 - 11 for Final Approval and Final Rejection actions.

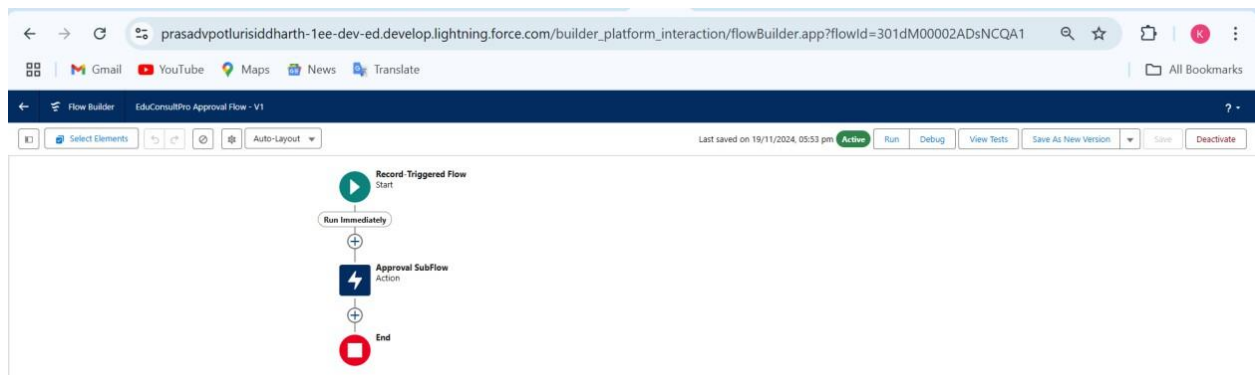
MileStone 5: Create a Record Triggered Flow

Configure the Start Element

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. The flow will look like this:

Add an Action Element

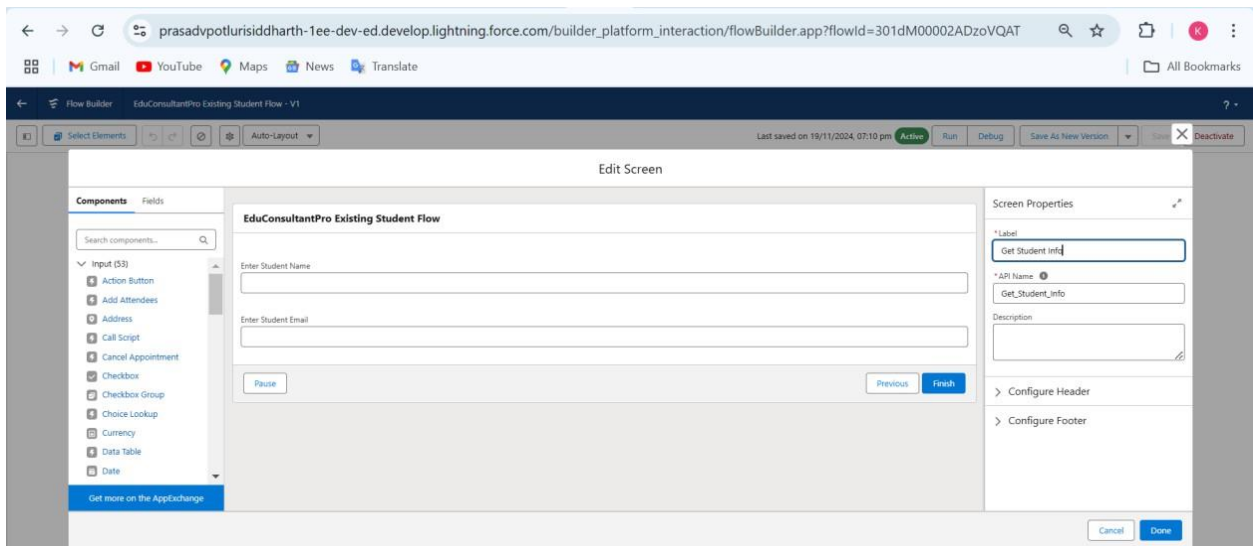
1. Add an Action element after the Start Element and Select the Submit for approval action, label it as “Approval SubFlow”.
2. Set the RecordId to “\${Record.Id}”.



MileStone 6: Create a ScreenFlow for Existing Student to Book an Appointment


Add Screen Element

1. From Setup, enter Flow Builder in quick find, 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. Give the Label's as follows:
1st Text Component Label : Enter Student Name
2nd Text Component Label : Enter Student Email



Add GET Record Element

1. Add a GET Record Element after Decision Element, under the IELTS path and label it as “Get Rec”.
2. Select Object : Student Condition Requirement : All Conditions are Met(AND) Field : Student Name Operator : Equals Value : {!Enter_Student_Name} Field : Email_c Operator : Equals Value : {!Enter_Student_Email}


Get Records
×

* Label

Get Rec

* API Name ⓘ

Get_Rec

Description

Get Records of This Object





* Object

Student - Student

Filter Student - Student Records

Condition Requirements

All Conditions Are Met (AND) ▼

Field	Operator	Value	
Student_Name__c	Equals ▼	 Enter_Student_Name ×	
AND Email__c	Equals ▼	 Enter_Student_Email ×	

+ Add Condition

Add Decision Element

1. Add a Decision Element after Select Display Student Details Element, label it as “Appointment or Case”.
2. Under outcome label it as “Appointment” and write the condition such as below: Resource : {!How_may_I_Help_you} Operator : Equals Value : {!Book_an_Appointment}
3. Click on the “+” icon and Repeat step 2 for Case options mentioned.

Decision

* Label

Appointment or Case

* API Name ⓘ

Appointment_or_Case

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 ⓘ +

Appointment

Case

Default Outcome

OUTCOME DETAILS

Delete Outcome

* Label

Appointment

* Outcome API Name ⓘ

Appointment

Condition Requirements to Execute Outcome

All Conditions Are Met (AND) ▼

Resource

Operator

Value

...

...t Details > How may I help you

×

Equals

▼

...

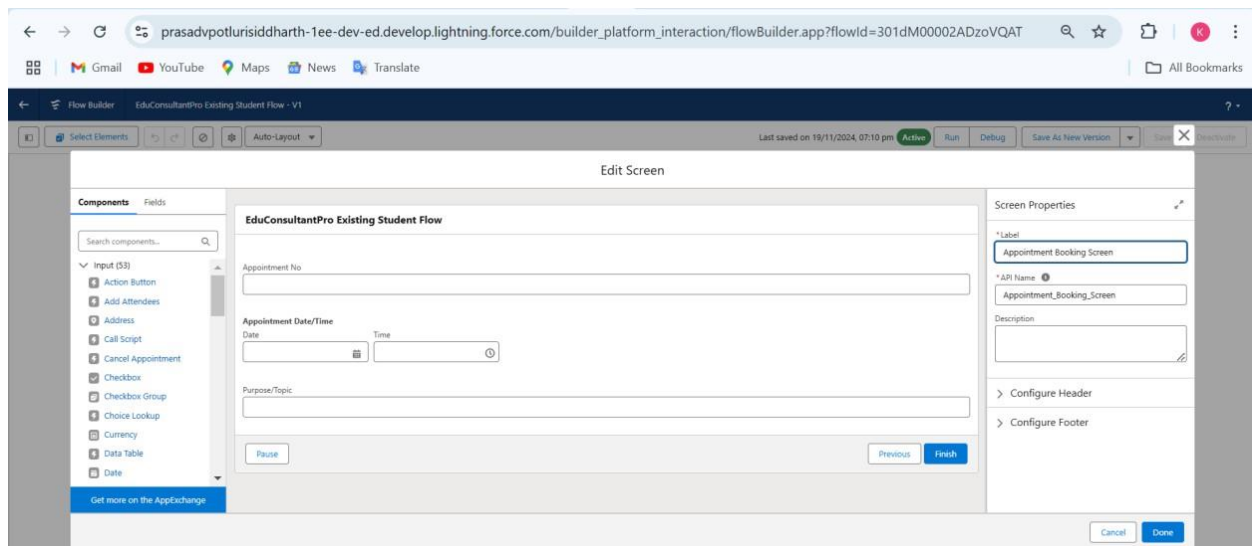
Book_an_Appointment

×

+ Add Condition

Add Screen Element

1. Add a Screen element after the Decision Element, on the Appointment path and label it as "Appointment Booking Screen".
2. Click on Fields, click on the record variable input and create a new Resource (AppointmentRecordRes) to display all the fields which are in the Appointment object.
3. Drag all the fields which are needed to add on the screen inorder to collect the student information.



Add GET Record Element

1. Add a GET Record Element after Decision Element, under the Appointment path and label it as "Get Consultant Rec".
2. Select Object : Consultant Condition Requirement : All Conditions are Met(AND) Field : Name Operator : Equals Value : {!AppointmentRecordRes.Consultant_Name__c}

Get Records
✕

* Label

* API Name ⓘ

Description

Get Records of This Object

* Object

Filter Consultant - Consultant Records

Condition Requirements

All Conditions Are Met (AND) ▼

Field	Operator	Value
<input type="text" value="First_Name__c"/>	<input type="text" value="Equals"/>	<input type="text" value="Aa AppointmentRecordRes > Ap..."/> ✕

+ Add Condition

Create Appointment Record using Create Records Element

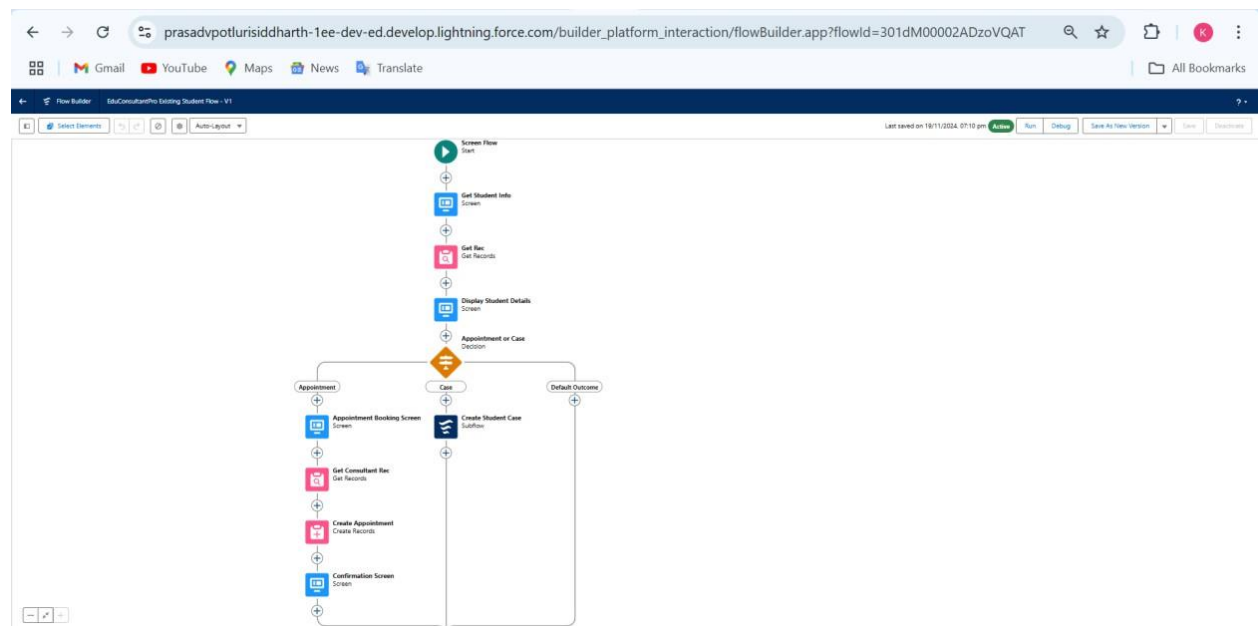
1. Add a Create element after the Get Consultant Rec element and label it as “Create Appointment”.
2. Select “one” under How many records to Create, and select “Use separate resources, and literal values” under How to Set the record fields.
3. Select Object : Appointment Field : Appointment_DateTime_c Value : {!AppointmentRecordRes.Appointment_DateTime_c} Field : Consultant_c Value : {!Get_Consultant_Rec.Id} Field : Notes_c Value : {!AppointmentRecordRes.Notes_c} Field : PurposeTopic_c Value : {!AppointmentRecordRes.PurposeTopic_c} Field : Student_Name_c Value : {!Get_Rec.Id}

Add Screen Element

1. Add a Screen Element after the Send Email to Student Action Element, label it as “Confirmation Screen”.
2. From the left side panel search for the Display text component and drag it to the main panel, label it as “Appointment_Confirmation”. 3. Paste the below in the Resource picker box. Consultant Name : {!Get_Consultant_Rec.Name}, Date & Time : {!AppointmentRecordRes.Appointment_DateTime_c}, Notes : {!AppointmentRecordRes.Notes_c},

Add an SubFlow Element

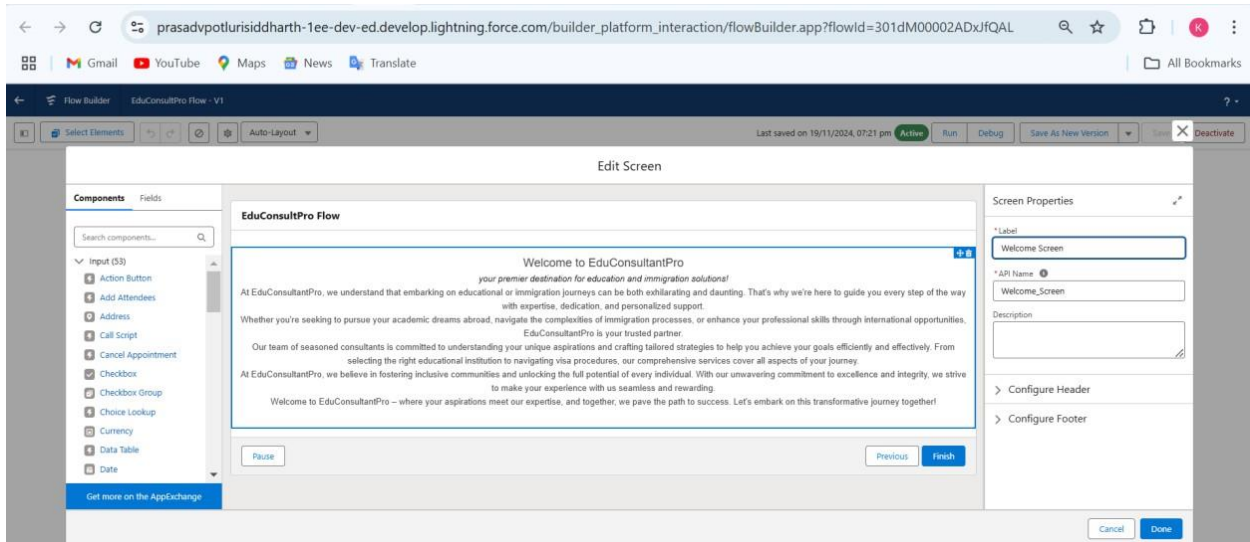
1. Add a subflow element after the Decision Element, on the Case path and search and Select for “Create a Case”, label it as “Create Student Case”.
2. Save the flow and label it as “EduConsultantPro Existing Student Flow”, you can use the below image for reference.



Milestone 7: Create a ScreenFlow to Combine all the flows at one place

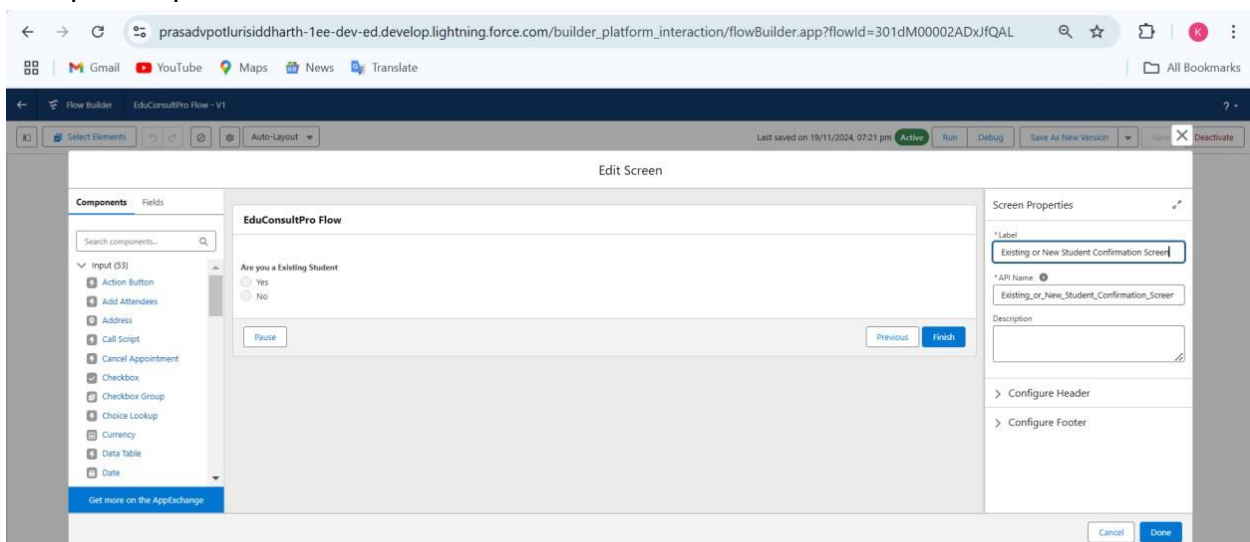
Add Screen Element

1. Add a Screen Element and label it as Welcome Screen.
2. From the left side panel search for the Display text component and drag it to the main panel, label it as “SuccessMessage”.
3. Paste the below in the Resource picker box.



Add Screen Element

1. Add a Screen Element after the Welcome Screen Element, label it as “Existing or New Student Confirmation Screen”.
2. Add a radio button component from the left side panel, label : Are you a Existing Student
3. Click on Add Choice --> type “Yes” in the input field --> click Create Yes choice.
4. Repeat step 6 and create an “No” choice resource.



Add Decision Element

1. Add a Decision Element after Existing or New Student Confirmation Screen Element, label it as “Decision 1”.
2. Under outcome label it as “If Existing Student” and write the condition such as below:
Resource : {!Are_you_a_Existing_Student} Operator : Equals Value : {!Yes}
3. Click on the “+” icon and Repeat step 2 for No options mentioned.

Decision

* Label

Decision 1

* API Name

Decision_1

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

If Existing Student

If not Existing Student

Default Outcome

* Label

If Existing Student

* Outcome API Name

If_Existing_Student

Condition Requirements to Execute Outcome

All Conditions Are Met (AND)

Resource

Operator

Value

Aa ...n > Are you a Existing Student

Equals

Aa Yes

+ Add Condition

Add an SubFlow Element

1. Add a subflow element after the Decision 1 Element on the if Existing Student path and search and Select for “EduConsultantPro Existing Student Flow ”, label it as “Existing Student Flow”.
2. Save the flow and label it as “EduConsultantPro Existing Student Flow”.



EduConsultantPro Existing Student Flow



* Label

Existing Student Flow

* API Name 

Existing_Student_Flow

Description

Referenced Flow



EduConsultantPro Existing Student Flow

EduConsultantPro_Existing_Student_Flow



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

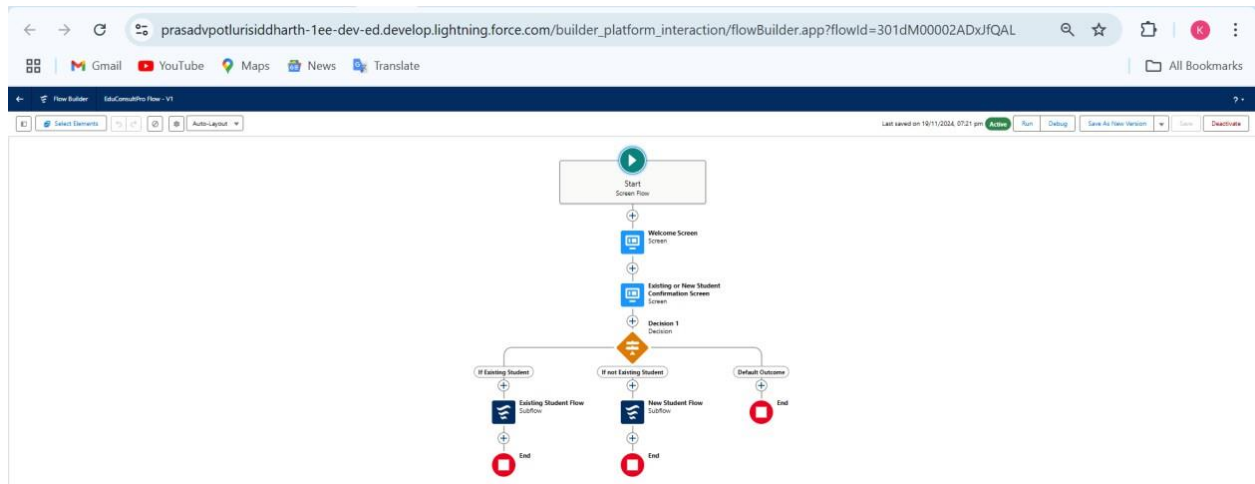


Nothing to set or store here

This flow has no inputs or outputs.

Add an SubFlow Element

1. Add a subflow element after the Decision 1 Element on the if Not an Existing Student path and search and Select for "EduConsultantPro Student Flow", label it as "New Student Flow".
2. Save the flow and label it as "EduConsultantPro Existing Student Flow".



APEX AND TRIGGERS

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions. A trigger is Apex code that executes before or after the following types of operations:

- insert
- update
- delete
- merge
- upsert
- undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin. You can define triggers for top-level standard objects that support triggers, such as Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To

define a trigger, from the object management settings for the object who triggers you want to access, go to Triggers. There are primarily two types of Apex Triggers: Before Trigger: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database. After Trigger: This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

Trigger Handler :

◆ The ValidateAppointmentTime Trigger validates that appointment times are within working hours (9:00 AM to 6:00 PM) and ensures there are no overlapping appointments for the same consultant.

◆ If a conflict is detected, it prevents the appointment from being saved and triggers the Apex Class to reschedule the appointment to the next available slot.

How to create a new trigger :

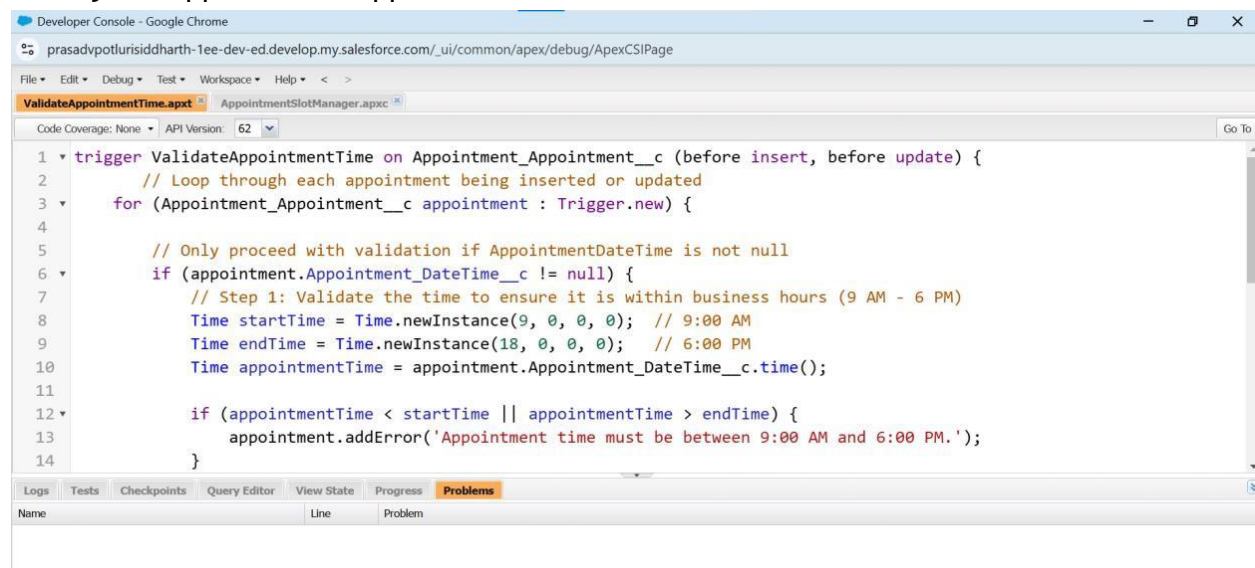
1. Click on developer console and you will be navigated to a new console window

Click on File menu in the tool bar, and click on new Trigger.

3. Enter the trigger name and the object to be triggered.

4. Name : ValidateAppointmentTime

5. sObject : Appointment__Appointment__c

The screenshot shows the Salesforce Developer Console interface. The top bar indicates the user is logged in as 'prasadpotlurisiddharth-1ee-dev-ed' and the current page is 'ApexCSIPage'. The main editor displays the code for a trigger named 'ValidateAppointmentTime.apxt' on the 'Appointment__Appointment__c' object. The code is as follows:

```
1 trigger ValidateAppointmentTime on Appointment__Appointment__c (before insert, before update) {
2     // Loop through each appointment being inserted or updated
3     for (Appointment__Appointment__c appointment : Trigger.new) {
4
5         // Only proceed with validation if AppointmentDateTime is not null
6         if (appointment.Appointment_DateTime__c != null) {
7             // Step 1: Validate the time to ensure it is within business hours (9 AM - 6 PM)
8             Time startTime = Time.newInstance(9, 0, 0, 0); // 9:00 AM
9             Time endTime = Time.newInstance(18, 0, 0, 0); // 6:00 PM
10            Time appointmentTime = appointment.Appointment_DateTime__c.time();
11
12            if (appointmentTime < startTime || appointmentTime > endTime) {
13                appointment.addError('Appointment time must be between 9:00 AM and 6:00 PM.');The bottom of the console shows tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Problems' tab is currently active, showing a table with columns 'Name', 'Line', and 'Problem'.
```

Code:

```
trigger ValidateAppointmentTime on Appointment__Appointment__c (before insert,
before update)
{ for (Appointment__Appointment__c appointment : Trigger.new) {
if (appointment.Appointment_DateTime__c != null) {
Time startTime = Time.newInstance(9, 0, 0, 0); // 9:00 AM
```



```

Time endTime = Time.newInstance(18, 0, 0, 0); // 6:00 PM
Time appointmentTime = appointment.Appointment_DateTime__c.time();
if (appointmentTime < startTime || appointmentTime > endTime) {
    appointment.addError('Appointment time must be between 9:00 AM and 6:00 PM.');
```

Apex Handler

The AppointmentSlotManager Apex Class dynamically identifies the next available appointment slot for a consultant if there is a conflict and reschedules the conflicting appointment. It updates the appointment record with the new time and ensures the consultant's schedule remains within capacity limits.

Developer Console - Google Chrome

Address bar: prasadvptotlurididharth-1ee-dev-ed.developer.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

ValidateAppointmentTime.apxt AppointmentSlotManager.apxc

Code Coverage: None API Version: 62

```

1 public class AppointmentSlotManager {
2     // Method to assign an appointment slot to a student
3     public static String assignAppointmentSlot(Appointment_Appointment__c appointment) {
4         // Step 1: Try to find the next available slot
5         DateTime nextAvailableSlot = findNextAvailableSlot(appointment.Appointment_DateTime__c, appointment.Consultant_Co
6
7         if (nextAvailableSlot != null) {
8             // Step 2: Update the appointment with the next available time
9             appointment.Appointment_DateTime__c = nextAvailableSlot;
10
11         try {
12             // Step 3: Update the appointment with the new time
13             update appointment;
14             return 'Appointment has been rescheduled to ' + nextAvailableSlot.Format('dd-MM-yyyy HH:mm:ss');

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem
------	------	---------

Code:

```

public class AppointmentSlotManager
{ public static String assignAppointmentSlot(Appointment_Appointment__c appointment) {
    DateTime nextAvailableSlot=findNextAvailableSlot(appointment.Appointment_DateTime__c,
    appointment.Consultant__Consultant__c);
    if (nextAvailableSlot != null) {
    appointment.Appointment_DateTime__c = nextAvailableSlot;
    try {
    update appointment;
    return 'Appointment has been rescheduled to ' + nextAvailableSlot.format();
    }
    catch (Exception e) {
    return 'Error occurred while rescheduling the appointment: ' + e.getMessage();
    }
    }
    else { return 'No available slots for rescheduling.';
    }
    }
    public static DateTime findNextAvailableSlot(DateTime appointmentDateTime, Id consultantId) {
        List<Appointment_Appointment__c> nextAvailableAppointments = [ SELECT
        Appointment_DateTime__c
        FROM Appointment_Appointment__c
        WHERE Consultant__Consultant__c = :consultantId
        AND Appointment_DateTime__c > :appointmentDateTime
        ORDER BY Appointment_DateTime__c ASC LIMIT 1 ]; if (!nextAvailableAppointments.isEmpty()){
        return nextAvailableAppointments[0].Appointment_DateTime__c;
        }
        else {
        return null;
        }
        }
    }
}

```

Milestone 8:Create a lightning app page

1. From Setup, enter App Builder in the Quick Find box, then click Lightning App Builder. 2. Click New, select Home Page, then click Next.
3. Step through the wizard and name the page “EduConsultPro Home Page”, select the Standard Home Page template, and then click Done.
4. Drag the Flow component to the top-right region.
5. Search for the “EduConsultantPro Flow” and click Save.



Activation...

Save

Components

Search...

Standard (40)

- !!! App Launcher
- iii Assistant
- Cil Chatter Feed
- gchatterPublisher
- d CRM Analytics Collection
- d CRM Analytics Dashboard
- B Dashboard
- mm Data Mask Console Home Compo...
- D EinsteinNextBestAction
- iii Flow
- Flow App Home cards
- m Generate Batch Documents
- Inventory Lookup Component
- Items to Approve

Get more on the AppExchange

Flow Component
EduConsultPro Flow

Add Component(s) Here

Page

EduConsultPro Home Page

EduConsultPro Home Page

Page Type
Home Page

Template
Standard Home Page

Description