

# **Institution Distributions**

By

Rohith Darapureddy

2111IT010062@mallareddyuniversity.ac.in

## **Project Abstract**

In today's competitive environment, effective customer relationship management is vital for any institution aiming to provide superior service. This project focuses on developing a customized CRM application using Salesforce to help students choose a best college. Salesforce, a leading CRM platform, offers robust tools and capabilities to streamline service management and improve customer satisfaction.

The CRM application will serve as a centralized platform where institutions can track and manage all customer interactions, from meeting to fulfilling the dreams of students. It will allow institutions to store and access detailed student information, including contact details, and preferences. By implementing this CRM application, institutions can expect to see significant improvements in customer satisfaction, and overall operational efficiency. The application will provide a structured approach to handling one to one interactions, reducing the chances of miscommunication and missed opportunities. Additionally, the insights gained from the application's analytics will help institutions identify areas for improvement and optimize their services.

# INDEX PAGE

<b>Topics</b>	<b>page No</b>
Cover Page	1
Project Abstract	2
Introduction	6
Task-1 Create Objects from Spreadsheet	7 - 12
1.1 Create Students object	7
1.2 Create Bookings object	7
1.3 Create signups object	8
1.4 Create Advisors object	8
1.5 Create Reports object	9
1.6 Create studies object	10
1.7 Create Detailss object	11
1.8 Create Infrastructures object	11
1.9 Create contacts object	11
1.10 Create Relationship among the objects	11
1.11 Create Tabs for the respective objects	12
1.12 Create Lightning App Page	12
TASK-2 ScreenFlow for Student Application process	13 - 35
2.1 Create the flow	13
2.2 Add a screen element	15
2.3 create a record	16
2.4 Add a screen element	17
2.5 Add a screen element	17
2.6 Add get record element	18

2.7 Add get record element	19
2.8 Add get record element	20
2.9 Add get record element	21
2.10 Add get record element	22
2.11 Add get record element	<b>23</b>
2.12 create record element	<b>25</b>
2.13 create record element	<b>26</b>
2.14 create record element	<b>27</b>
2.15 create record element	<b>28</b>
2.16 create record element	<b>29</b>
2.17 create record element	<b>30</b>
2.18 create email template	<b>31</b>
2.19 create email template	<b>33</b>
2.20 Add Action element	<b>34</b>
2.21 Add Action element	<b>35</b>
TASK-3 Create a User and make suitable settings	<b>36 - 37</b>
3.1 Create a User	<b>36</b>
3.2 Configure the user settings	<b>37</b>
TASK-4 Creating Email Templates and Approval Processes	<b>38 - 44</b>
4.1 create email template	<b>38</b>
4.2 create email template	<b>42</b>
4.3 create email template	<b>44</b>
4.4 create an approval process	<b>46</b>
TASK-5 Record Triggered Flow Creation	<b>52 - 53</b>
5.1 create email template	<b>52</b>
5.2 Add action element	<b>53</b>

TASK-6 Existing student flow for a Booking	<b>55 - 61</b>
6.1 Add a screen	<b>55</b>
6.2 Add GET Record Element	<b>56</b>
6.3 Add Decision Element	<b>57</b>
6.4 Add screen	<b>58</b>
6.5 Add GET Record Element	<b>59</b>
6.6 Add a Create Record Element	<b>60</b>
6.7 Add screen	<b>61</b>
6.8 Add screen	<b>61</b>
TASK-7 Flow to combine all flows	<b>62 - 66</b>
7.1 Add screen	<b>62</b>
7.2 Add screen	<b>63</b>
7.3 Add Decision Element	<b>64</b>
7.4 Add Subflow	<b>65</b>
7.5 Add Subflow	<b>66</b>
TASK-8 Lightning app page creation	<b>67</b>
8.1 creating a lightning app page	<b>67</b>

## **Introduction**

Educational institution offering a variety of courses and programs in diverse fields. With a growing number of prospective students seeking admission each year, the institute faces challenges in managing the admission process, students enquiry, and study related advices .To address these challenges, Institution decides to leverage Salesforce CRM to streamline the admission process and enhance the overall experience for both students and admissions staff.

The use case focuses on the admission process for prospective students interested in enrolling in courses and programs. The goal is to provide a seamless and transparent experience for students while enabling admissions staff to efficiently review and process admission applications, students enquiry and case management.

In today's competitive business environment, maintaining a strong relationship with customers is crucial for the success of any institution. A Customer Relationship Management (CRM) application is designed to streamline and enhance interactions between an institution and its customers. This project aims to develop a CRM application specifically tailored to manage the services offered by the institution, ensuring efficient service delivery and improving customer satisfaction.

The primary objective of the CRM application is to provide a comprehensive platform that integrates all aspects of customer interaction and service management. This includes managing customer information, interacting with students, and providing insights through data analysis. By centralizing these functions, the CRM application will enable the institution to deliver a more personalized and responsive customer experience.

## **Task-1 Create Objects from Spreadsheet**

- Directly Creating Objects from Spreadsheet in Salesforce
- First we have to create some spreadsheets of respective object with some values in it.

### **1.1 Create Students object**

- Go to your object manager and click on create object from spreadsheet.
- select the option upload.
- select the students spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.

### **1.2 Create Bookings object**

- Go to your object manager and click on create object from spreadsheet.
- select the option upload.
- select the Bookings spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.

Create a custom object from a spreadsheet

## Nice Work!



Now you can add your object to a Lightning app. You might need to refresh the object list to see it.

Import Overview

Object Created	Booking
Fields Detected	9
Rows Detected	13
Fields Created	5
Rows Imported	13

[Import Another Object](#)

### 1.3 Create signups object

- Go to your object manager and click on create object from spreadsheet.
- select the option upload.
- select the signups spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.

## Nice Work!



Now you can add your object to a Lightning app. You might need to refresh the object list to see it.

Import Overview

Object Created	Signup
Fields Detected	7
Rows Detected	13
Fields Created	3
Rows Imported	13

[Import Another Object](#)

### 1.4 Create Advisors object

- Go to your object manager and click on create object from spreadsheet.

- select the option upload.
- select the Advisors spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.

Create a custom object from a spreadsheet

**Nice Work!**

A cartoon dog wearing a superhero cape and mask.

Now you can add your object to a Lightning app. You might need to refresh the object list to see it.

**Import Overview**

Object Created	Fields Detected	Fields Created	Rows Detected	Rows Imported
Advisors	9	5	13	13

[Import Another Object](#)

## 1.5 Create Reports object

- Go to your object manager and click on create object from spreadsheet.
- select the option upload.
- select the Reports spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.

**Object Created**  
Reports

**Fields Detected**  
19

**Fields Created**  
19

**Rows Detected**  
13

**Rows Imported**  
0

[Import Another Object](#)

## 1.6 Create studies object

- Go to your object manager and click on create object from spreadsheet.
- select the option upload.
- select the studies spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.

Create a custom object from a spreadsheet

**Nice Work!**



Now you can add your object to a Lightning app. You might need to refresh the object list to see it.

**Import Overview**

**Object Created**  
studies

**Fields Detected**  
5

**Fields Created**  
5

**Rows Detected**  
6

**Rows Imported**  
6

[Import Another Object](#)

## **1.7 Create Details object**

- Go to your object manager and click on create object from spreadsheet.
- select the option upload.
- select the Details spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.

## **1.8 Create Infrastructures object**

- Go to your object manager and click on create object from spreadsheet.
- select the option upload.
- select the Infrastructures spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.

## **1.9 Create contacts object**

- Go to your object manager and click on create object from spreadsheet.
- select the option upload.
- select the contacts spreadsheet.
- Then set the suitable data types for the fields and click on next.
- Now, in the object properties give the label and click on create.
- Now, in the object properties give the label and click on create.

## **1.10 Create Relationship among the objects**

- Create lookup between Booking and student, Booking and Advisors.
- Also create a lookup between student and reports to store the student queries.

The screenshot shows the Salesforce Object Manager interface with two tabs: 'Booking' and 'student'. Both tabs display a table of fields and relationships.

**Booking Tab Fields & Relationships:**

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Advisors	Advisors__c	Lookup(Advisors)		✓
Booking	Name	Text(80)		✓
Booking Date/Time	Booking_DateTime__c	Date/Time		
Booking No	Booking_No__c	Number(18, 0)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Notes	Notes__c	Text(255)		
Owner	OwnerId	Lookup(User/Group)		✓
SOP/Topic	SOPTopic__c	Text(255)		
Status	Status__c	Picklist		
student	student__c	Lookup(student)		✓
Student Name	Student_Name__c	Text(20)		

**student Tab Fields & Relationships:**

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(255)		
City	City__c	Text(255)		
Created By	CreatedById	Lookup(User)		
Date of Birth	Date_of_Birth__c	Date		
Email	Email__c	Email		
Gender	Gender__c	Text(255)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		✓
Phone	Phone__c	Number(18, 0)		
Qualification	Qualification__c	Text(255)		
Reports	Reports__c	Lookup(Reports)		
student	Name	Text(80)		✓
Student Name	Student_Name__c	Text(255)		
University Name	University_Name__c	Text(255)		

## 1.11 create Tabs for the respective objects

- In the home, go to quick find box and search for the tabs.
- Go to tabs, under custom tabs click on new.
- Give object name as which object you are relating this tab.
- And give the tab style and click on next, save.

## 1.12 create Lightning App Page

- Go to Setup, search for the App Manager in quick find.
- Click on New Lightning App.
- Give app name as "Institution", click Next, Next, Next.

- Add Students, studies, advisors, contacts, Bookings, Infrastructures, signups, reports, contacts, Detailss from the Available Items to Selected Items.
- Add “System Administrator” profile from Available Profiles to Selected Profiles, click Save & Finish.

The screenshot shows the Salesforce Setup interface with the 'App Manager' selected. The search bar at the top contains 'app ma'. On the left, there's a sidebar with 'Setup' selected and a search bar for 'app ma'. Below it, 'Apps' is expanded, showing 'App Manager' which is also selected. Under 'App Manager', there are links for 'External Client Apps' and 'External Client App Manager'. A message says 'Didn't find what you're looking for? Try using Global Search.' The main area is titled 'Lightning Experience App Manager' and displays a table of 24 items. The columns are: App Name ↑, Developer Name, Description, Last Modified D..., App T..., Visi..., and a dropdown menu icon. The table rows are:

	App Name ↑	Developer Name	Description	Last Modified D...	App T...	Visi...	
1	All Tabs	AllTabSet	Build CRM Analytics dashboards and apps	25/10/2024, 9:09 am	Classic	✓	▼
2	Analytics Studio	Insights	Discover and manage business solutions designed for your industry.	25/10/2024, 9:09 am	Classic	✓	▼
3	App Launcher	AppLauncher	Create and maintain business rules that perform complex lookups and calculations.	25/10/2024, 9:09 am	Classic	✓	▼
4	Automation	FlowsApp	Automate business processes and repetitive tasks.	25/10/2024, 9:16 am	Lightning	✓	▼
5	Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	25/10/2024, 9:09 am	Lightning	✓	▼
6	Business Rules Engine	ExpressionSetConsole	Create and maintain business rules that perform complex lookups and calculations.	25/10/2024, 9:16 am	Lightning	✓	▼
7	Community	Community	Salesforce CRM Communities	25/10/2024, 9:09 am	Classic	✓	▼
8	Content	Content	Salesforce CRM Content	25/10/2024, 9:09 am	Classic	✓	▼
9	Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	25/10/2024, 9:09 am	Lightning	✓	▼
10	Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	25/10/2024, 9:09 am	Lightning	✓	▼
11	Institution	Institution	Information about the Institution	25/10/2024, 8:50 pm	Lightning	✓	▼
12	Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	25/10/2024, 9:09 am	Lightning	✓	▼

## TASK-2 ScreenFlow for Student Application process

### 2.1 Create the flow

- Go to setup, search flows in the quick find box.
- Click on flows, click on new flow.
- Click on screen flow and click on create.

**Setup Home**

Search Setup: flows

Process Automation: Flows

Identity: Login Flows

Most Recently Used: Advisors (Custom Object Definition)

**Flows**

Try the Automation Lightning App!

Flow Definitions: All Flows

Flow Label	Process Type	Act...	Te...	Package State	Pa...	Last Modified By	Last Modified D...
Add or Modify Service Appointment Attendees	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-Installed			
Basic Approval Request	Flow Orchestration for CMS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Book Appointment from Invitation	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Cancel Item Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Change Case Owner to Incident Owner	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Charter Conference Eligibility	Auto-launched Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			

New Flow

Select how you'd like to start building your automation.

**Start From Scratch**  
Select your automation type and start building on an empty canvas.

**Use a Template**  
Select a pre-built flow and customize it to fit your needs.

Back Next

Select Type

**Recommended**

- Screen Flow** Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.
- Record-Triggered Flow** Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.
- Schedule-Triggered Flow** Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.
- Platform Event—Triggered Flow** Launches when a platform event message is received. This autolaunched flow runs in the background.
- Autolaunched Flow (No Trigger)** Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.
- Record-Triggered Orchestration** Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

**All Flow Types**

- Autolaunched Flow (No Trigger)
- Autolaunched Orchestration (No Trigger)

[Back](#) [Create](#)

## 2.2 Add a screen element

- Click on the flow and select screen.
- Give the name for the screen as AboutStudent.
- Now, in the left side go to fields and click on new resource.
- create a record variable named infoofstudents for accessing all the student object fields into teh flow.
- After creating, select and drag the necessary fields onto the screen, the click on done.

Edit Screen

Last saved on 25/10/2024, 10:31 pm [Active](#) [Run](#) [Debug](#) [Save As New Version](#) [Deactivate](#)

**Components:** Fields

**Fields**

- Add record fields to your screen. [More info](#)
- + Record Variable **Infoofstudents**
- Search student fields...
- Fields (12)
  - Ay Address
  - Ay City
  - Date of Birth
  - Ay Email
  - Ay Gender
  - Ay student
  - # Phone
  - Ay Qualification
  - Reports
  - Ay Student Name
  - Ay University Name
  - # Year of Passing

**InstitutionstreamBooking**

Student Name

Gender

Email

Date of Birth

Address

Phone

Qualification

Student Name

**Screen Properties**

- Label **AboutStudent**
- Alt Name **AboutStudent**
- Description
- > Configure Header
- > Configure Footer

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

## 2.3 create a record

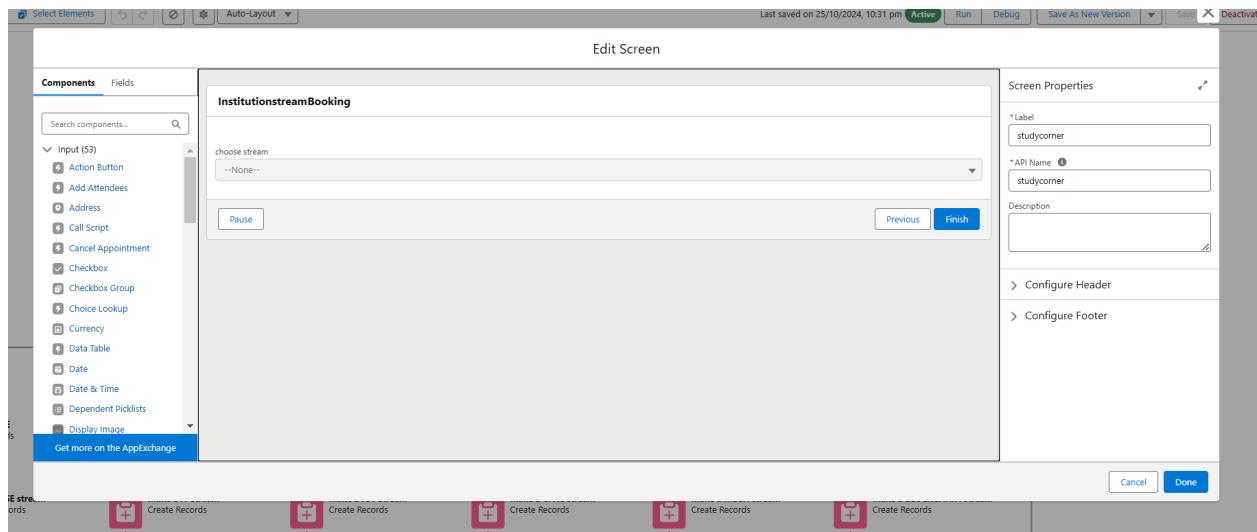
- Under AboutStudent, click on + , click on create record.
- Give the name as Studentcorner.
- Give the record value as infoofstudents.
- click on save.

The screenshot shows the 'Create Records' configuration screen within a flow builder. The top bar has a green progress bar and a close button (X). The main area contains the following fields:

- \* Label:** studentcorner
- \* API Name:** studentcorner
- Description:** (empty text area)
- \* How to set record field values:** From a Record Variable
- How Many Records to Create:** One (radio button selected)
- Create a Record from These Values:**
  - \* Record:** Infoofstudents (with a delete icon)
- Note:** Make sure that ID is blank. After the flow creates the records, ID is set to match the record that was created. (with a help icon)
- Update Existing Record:** (disabled, indicated by a greyed-out icon and the word 'Disabled' next to it)

## 2.4 Add a screen element

- Under studentcorner, click on + , click on screen.
- Give the name as studycorner.
- In the components, search for picklist, give name as choosestream.
- In the choice place, type CSE and click create CSE choice, then CSE choice is created.
- In the choice place, type IT and click create IT choice, then IT choice is created.
- In the choice place, type MECH and click create MECH0 choice, then MECH choice is created.
- In the choice place, type IOT and click create IOT choice, then IOT choice is created.
- In the choice place, type CIVIL and click create CIVIL choice, then CIVIL choice is created.
- In the choice place, type BLOCKCHAIN and click create BLOCKCHAIN choice, then BLOCKCHAIN choice is created.
- click on done.



## 2.5 Add a screen element

- Under AboutStudent, click on + , click on decision element.
- Give the name as Making First Step.

- Under outcome label it as “Took CSE” and write the condition such as below:

Resource : choosestream

Operator : Equals

Value : CSE

- Click on + beside outcome label and perform the above step for the IT,IOT,MECH,CIVIL,BLOCKCHAIN.
- click on save.

OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome
Took CSE	*Label Took CSE *Outcome API Name Took_CSE Condition Requirements to Execute Outcome All Conditions Are Met (AND) Resource studycorner > choose stream Operator Equals Value CSE	<a href="#">Delete Outcome</a>
Took IT		
Took IOT		
Took CIVIL		
Took MECH		
Took BLOCKCHAIN		
Default Outcome		

## 2.6 Add get record element

- Under the decision element , in the took CSE path, add a get record element.
- label it as access CSE.
- Give the object as studies and place the following in the field and

value :

field: Name ; value: choose\_stream

- Click on save.

The screenshot shows the configuration interface for a 'Get Records' element. At the top, there are fields for 'Label' (access CSE) and 'API Name' (access\_CSE). Below these are sections for 'Description' (empty), 'Get Records of This Object' (studies), 'Object' (studies), and 'Condition Requirements'. A dropdown menu shows 'All Conditions Are Met (AND)'. Below this, a condition row is displayed with 'Field' set to 'Name', 'Operator' set to 'Equals', and 'Value' set to 'Aa choose\_stream'. A 'Delete' button is shown next to the value field. A 'Add Condition' button is also present. Further down, there is a 'Sort studies Records' section with a 'Sort Order' dropdown set to 'Not Sorted'. A warning message states: '⚠ If you store only the first record, filter by a unique field, such as ID.' In the bottom left, there is a 'How Many Records to Store' section with two radio buttons: 'Only the first record' (selected) and 'All records'.

## 2.7 Add get record element

- Under the decision element , in the took IT path, add a get record element.
- label it as access IT.
- Give the object as studies and place the following in the field and

value :

field: Name ; value: choose\_stream

- Click on save.

The screenshot shows the 'Get Records' configuration page in Salesforce. The 'Label' field is set to 'access IT'. The 'API Name' field is set to 'access\_IT'. The 'Object' field is set to 'studies'. Under 'Filter studies Records', there is a condition requirement 'All Conditions Are Met (AND)' with one condition: 'Name Equals A\_a choose\_stream'. Under 'Sort studies Records', the 'Sort Order' is set to 'Not Sorted'. A warning message states: '⚠ If you store only the first record, filter by a unique field, such as ID.' In the 'How Many Records to Store' section, 'Only the first record' is selected. In the 'How to Store Record Data' section, 'Automatically store all fields' is selected.

Get Records

\* Label access IT

\* API Name access\_IT

Description

Get Records of This Object

\* Object studies

Filter studies Records

Condition Requirements All Conditions Are Met (AND)

Field	Operator	Value
Name	Equals	A_a choose_stream

+ Add Condition

Sort studies Records

Sort Order Not Sorted

⚠ If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store

Only the first record

All records

How to Store Record Data

Automatically store all fields

Choose fields and let Salesforce do the rest

Choose fields and let Salesforce do the rest

## 2.8 Add get record element

- Under the decision element , in the took CIVIL path, add a get record element.
- label it as access CIVIL.

- Give the object as studies and place the following in the field and value : field: Name ; value: choose\_stream
- Click on save.

**Get Records**

**\*Label**: access CIVIL    **\*API Name**: access\_CIVIL

**Description**: (empty)

**Get Records of This Object**

**\*Object**: studies

**Filter studies Records**

**Condition Requirements**: All Conditions Are Met (AND)

Field	Operator	Value
Name	Equals	A_a choose_stream <span style="border: 1px solid black; padding: 2px;">X</span>

**+ Add Condition**

**Sort studies Records**

**Sort Order**: Not Sorted ⚠ If you store only the first record, filter by a unique field, such as ID.

**How Many Records to Store**:  Only the first record  All records

**How to Store Record Data**:  Automatically store all fields  Choose fields and let Salesforce do the rest  Choose fields and assign variables (advanced)

## 2.9 Add get record element

- Under the decision element , in the took MECH path, add a get record element.

- label it as access MECH.
- Give the object as studies and place the following in the field and value :
  - field: Name ; value: choose\_stream
- Click on save.

Last Saved On 25/10/2024, 10:51 pm **Active** Run Debug Save As New Version Save Unpublish

### Get Records

\* Label  \* API Name

Description

Get Records of This Object

\* Object

Filter studies Records

Condition Requirements

Field	Operator	Value
Name	Equals	A_a <a href="#">choose_stream</a> <input type="button" value="X"/>

[+ Add Condition](#)

Sort studies Records

Sort Order  ⚠ If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store  
 Only the first record  
 All records

How to Store Record Data  
 Automatically store all fields  
 Choose fields and let Salesforce do the rest  
 Choose fields and assign variables (advanced)

## 2.10 Add get record element

- Under the decision element , in the took BLOCKCHAIN path, add a get

record element.

- label it as access BLOCKCHAIN.
- Give the object as studies and place the following in the field and value :  
field: Name ; value: choose\_stream
- Click on save.

The screenshot shows the configuration of a 'Get Records' element. The 'Label' is set to 'access BLOCKCHAIN' and the 'API Name' is 'access\_BLOCKCHAIN'. The 'Object' is set to 'studies'. Under 'Filter studies Records', the condition 'All Conditions Are Met (AND)' is selected. A single filter is applied with the field 'Name', operator 'Equals', and value 'choose\_stream'. The 'Sort studies Records' section shows 'Not Sorted' as the sort order. In the 'How Many Records to Store' section, 'Only the first record' is selected. The 'How to Store Record Data' section has 'Automatically store all fields' selected. A warning message states: '⚠ If you store only the first record, filter by a unique field, such as ID.'

## 2.11 Add get record element

- Under the decision element , in the took IOT path, add a get record element.
- label it as access IOT.

- Give the object as studies and place the following in the field and value :
  - field: Name ; value: choose\_stream
- Click on save.

The screenshot shows the configuration interface for a 'Get Records' step in a Salesforce flow named 'access IOT'. The configuration includes:

- Label:** access IOT
- API Name:** access\_IOT
- Description:** (empty)
- Object:** studies
- Condition Requirements:** All Conditions Are Met (AND)
  - Field:** Name
  - Operator:** Equals
  - Value:** A\_a choose\_stream
- Add Condition:** + Add Condition
- Sort Order:** Not Sorted
  - A warning message: If you store only the first record, filter by a unique field, such as ID.
- How Many Records to Store:**
  - Only the first record
  - All records
- How to Store Record Data:**
  - Automatically store all fields
  - Choose fields and let Salesforce do the rest
  - Choose fields and assign variables (advanced)

## **2.12 create record element**

- Under the took CSE path, create a record element.
- label it as make a CSE stream.
- Give the object as signup and place the following field values:  
field1: Course Name ; value1: studies from access CSE.Record ID  
field1: Student Name ; value1: infoofstudents.Record ID
- Click on save.

Last saved on 25/10/2024, 10:31 pm Active Run Debug Save As New Version ▾ Save Deactivate

 Create Records ×

\* Label \* API Name ⓘ  
Make a CSE stream Make\_a\_CSE\_stream

Description

\* How to set record field values  
Manually

Create a Record of This Object  
\* Object  
Signup

Set Field Values for the Signup

Field <span style="border: 1px solid #ccc; padding: 2px;">A_a Course Name X</span>	Value <span style="border: 1px solid #ccc; padding: 2px;">A_a studies from access CSE &gt; Record ID X</span> <span style="margin-left: 10px;">←</span> <span style="border: 1px solid #ccc; padding: 2px;">A_a</span> <span style="border: 1px solid #ccc; padding: 2px;">Infoofstudents &gt; Record ID X</span> <span style="margin-left: 10px;">←</span> <span style="border: 1px solid #ccc; padding: 2px;">A_a</span>
Field <span style="border: 1px solid #ccc; padding: 2px;">A_a Student Name X</span>	Value <span style="border: 1px solid #ccc; padding: 2px;">A_a Infoofstudents &gt; Record ID X</span> <span style="margin-left: 10px;">←</span> <span style="border: 1px solid #ccc; padding: 2px;">A_a</span>

+ Add Field  Manually assign variables

Check for Matching Records Disabled

## 2.13 create record element

- Under the took IT path, create a record element.
- label it as make a IT stream.
- Give the object as signup and place the following field values:
  - field1: Course Name ; value1: studies from access IT.Record ID
  - field1: Student Name ; value1: infoofstudents.Record ID
- Click on save.

**Create Records**

\* Label: Make a IT stream

\* API Name: Make\_a\_IT\_stream

Description:

\* How to set record field values: Manually

Create a Record of This Object: Signup

Set Field Values for the Signup

Field	Value
A_a Course Name	A_a studies from access IT > Record ID
A_a Student Name	A_a Infoofstudents > Record ID

+ Add Field

Manually assign variables

Check for Matching Records:  Disabled

## 2.14 create record element

- Under the took IOT path, create a record element.
- label it as make a IOT stream.
- Give the object as signup and place the following field values:  
field1: Course Name ; value1: studies from access IOT.Record ID  
field1: Student Name ; value1: infoofstudents.Record ID
- Click on save.

Last saved on 25/10/2024, 10:31 pm Active Run Debug Save As New Version Save Deactivate

---

Create Records
X

---

\*Label
\*API Name i

Make\_a\_IOT\_stream

---

Description

---

Tools
\* How to set record field values

Manually
▼

---

Create a Record of This Object

\*Object

---

Set Field Values for the Signup

Field

Value

X

Field

Value

X

+ Add Field
 Manually assign variables

---

Check for Matching Records
Disabled

## 2.15 create record element

- Under the took CIVIL path, create a record element.
- label it as make a CIVIL stream.
- Give the object as signup and place the following field values:  
field1: Course Name ; value1: studies from access CIVIL.Record ID  
field1: Student Name ; value1: infoofstudents.Record ID
- Click on save.

Last saved on 25/10/2024, 10:31 pm

Active

Run

Debug

Save As New Version

Save

Deactivate

**Create Records**

**Label**: Make a CIVIL stream | **API Name**: MAke\_a\_CIVIL\_stream

**Description**:

**How to set record field values**: Manually

**Create a Record of This Object**: Signup

**Set Field Values for the Signup**

Field	Value
A <sub>a</sub> Course Name	A <sub>a</sub> studies from access CIVIL > Record ID
A <sub>a</sub> Student Name	A <sub>a</sub> Infoofstudents > Record ID

**Add Field**

Manually assign variables

**Check for Matching Records**: Disabled

## 2.16 create record element

- Under the took MECH path, create a record element.
- label it as make a MECH stream.
- Give the object as signup and place the following field values:  
field1: Course Name ; value1: studies from access MECH.Record ID  
field1: Student Name ; value1: infoofstudents.Record ID

- Click on save.

Last saved on 25/10/2024, 10:31 pm Active

Run Debug Save As New Version Save Deactivate

**Create Records**

**\*Label**: Make a MECH stream

**\*API Name**: Make\_a\_MECH\_stream

**Description**: (empty)

**\*How to set record field values**: Manually

**Create a Record of This Object**: Signup

**Set Field Values for the Signup**

Field	Value
A_a Course Name	A_a studies from access MECH > Record ID
A_a Student Name	A_a Infoofstudents > Record ID

**+ Add Field**

Manually assign variables

**Check for Matching Records** Disabled

## 2.17 create record element

- Under the took BLOCKCHAIN path, create a record element.
- label it as make a BLOCKCHAIN stream.
- Give the object as signup and place the following field values:  
field1: Course Name ; value1: studies from access  
BLOCKCHAIN.Record ID  
field1: Student Name ; value1: infoofstudents.Record ID
- Click on save.

Last saved on 25/10/2024, 10:31 pm Active

**Create Records**

**\* Label** Make a BLOCKCHAIN stream      **\* API Name** Make\_a\_BLOCKCHAIN\_stream

Description

**\* How to set record field values** Manually

**Create a Record of This Object**

**\* Object** Signup

**Set Field Values for the Signup**

Field	Value
A_a Course Name	A_a studies from access BLOCKCHAIN > Record ID
A_a Student Name	A_a Infoofstudents > Record ID

+ Add Field     Manually assign variables

**Check for Matching Records** Disabled

## 2.18 create email template

- Go to the top left side corner , go to toggle box.
- click on new resource.
- select the text template.
- Give the API name as Successfulstreambookingmailtemp.
- In the Body place , write the below text
 

Dear {!Infoofstudents.Name},  
 Congratulations and welcome to InstitutionstreamBooking!  
 We are delighted to inform you that your booking 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 advisarial things.  
 At InstitutionstreamBooking, we understand the importance of your

academic and career goals, and we are committed to provide our support to you and perfect guidance throughout your journey. Here are a few key points to help you get started:

**Explore the Institution :** Take some time to explore our institution and services we are providing to students .

**Connect with Our Advisors:** Our team of experienced advisors is here to assist you at each and every stage of your educational aspects including admissions, abroad study, subject regarding ,etc... 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 InstitutionstreamBooking. We'll ensure that you're informed about the latest developments and relevant information to support your journey.

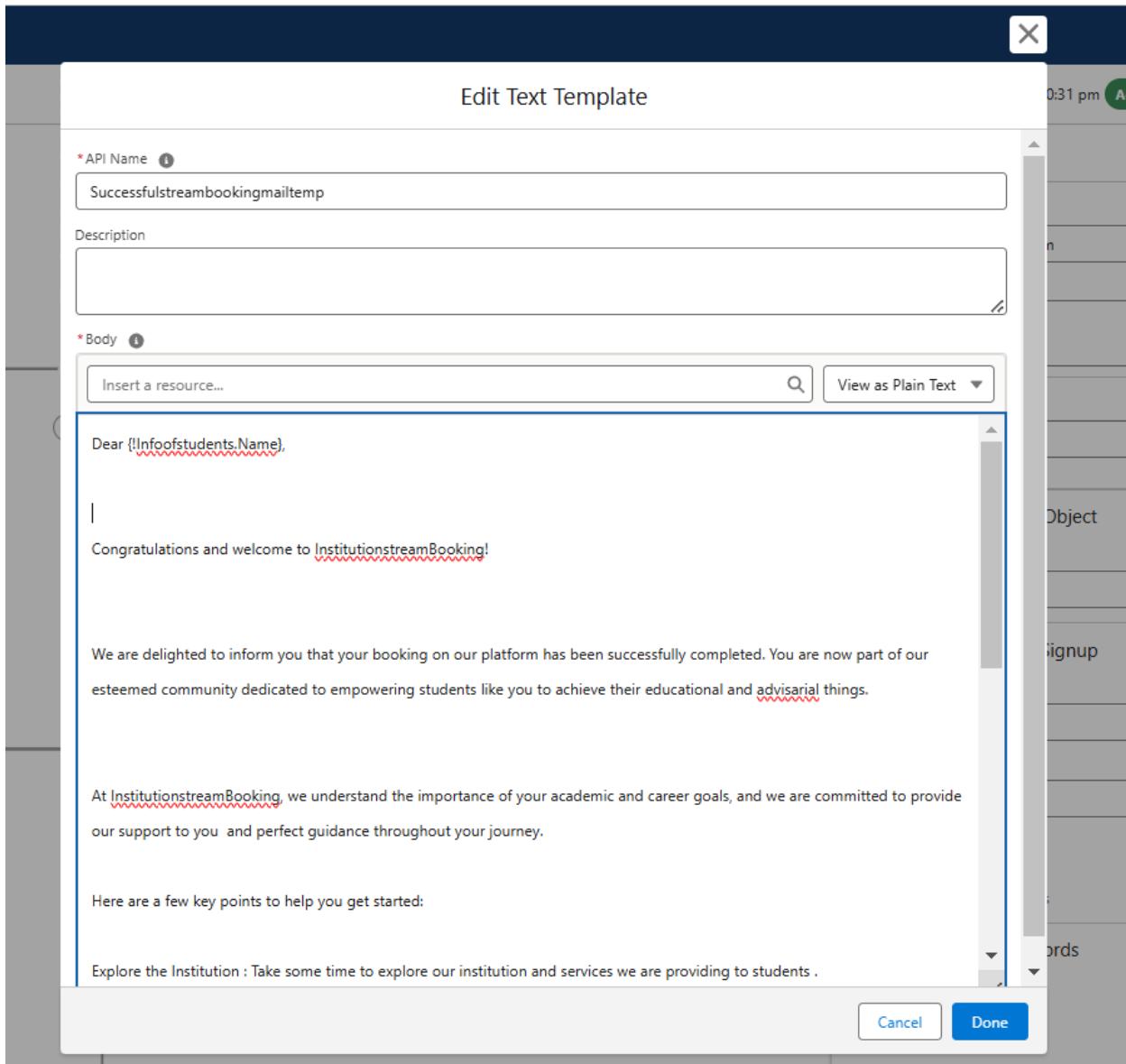
**Connect to the expertise :** Join our vibrant community of students, professionals, and experts who share your passion for education and global opportunities. Connect with like-minded individuals, participate in discussions, and expand your network.

Once again, congratulations on taking this important step towards realizing your academic and career aspirations. We are thrilled to have you as part of the InstitutionstreamBooking family and look forward to supporting you on your journey to success.

If you have any questions or need assistance, please don't hesitate to contact us.

Thank you.

► click on done.



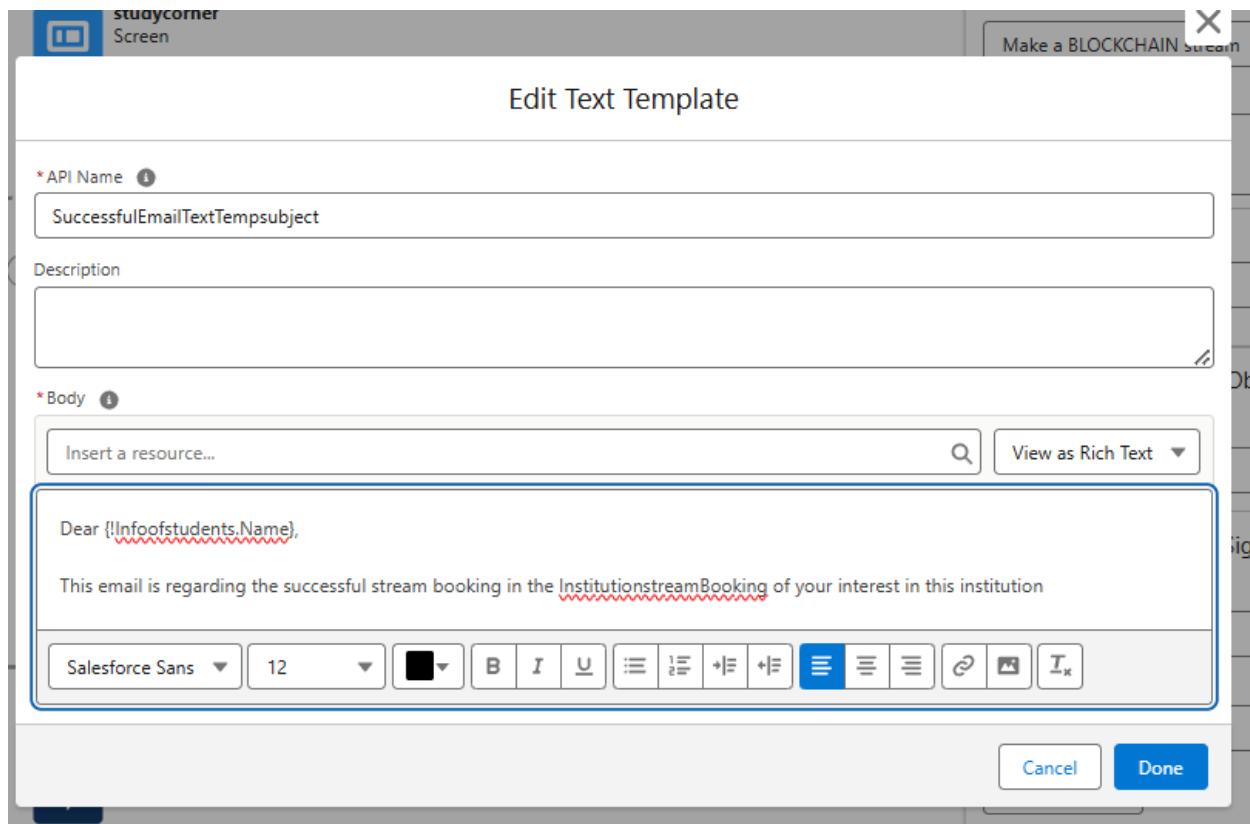
## 2.19 create email template

- Go to the top left side corner , go to toggle box.
- click on new resource.
- select the text template.
- Give the API name as SuccessfulEmailTextTempsubject.

In the Body place , write the below text

Dear {Infoofstudents.Name},

This email is regarding the successful stream booking in the InstitutionstreamBooking of your interest in this institution



## 2.20 Add Action element

- After all the paths under the decision element, add an action element.
- give name as Confirmation mail.
- give the following input values for the select action:
  - Body: Successfullstreambookingmailtemp
  - Recipient Address List: infoofstudents.Email
  - Subject: SuccessfulEmailTextTempsubject
- Click on save.

**Send Email**

\* Label: Confirmation mail

\* API Name: Confirmation\_mail

Description:

**Send Email** 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**

- Add Threading Token to Body: Not Included
- Add Threading Token to Subject: Not Included
- BCC Recipient Address List: Not Included
- Body: Included  
Successfulstreambookingmailtemp
- CC Recipient Address List: Not Included
- Email Template ID: Not Included

## 2.21 Add Action element

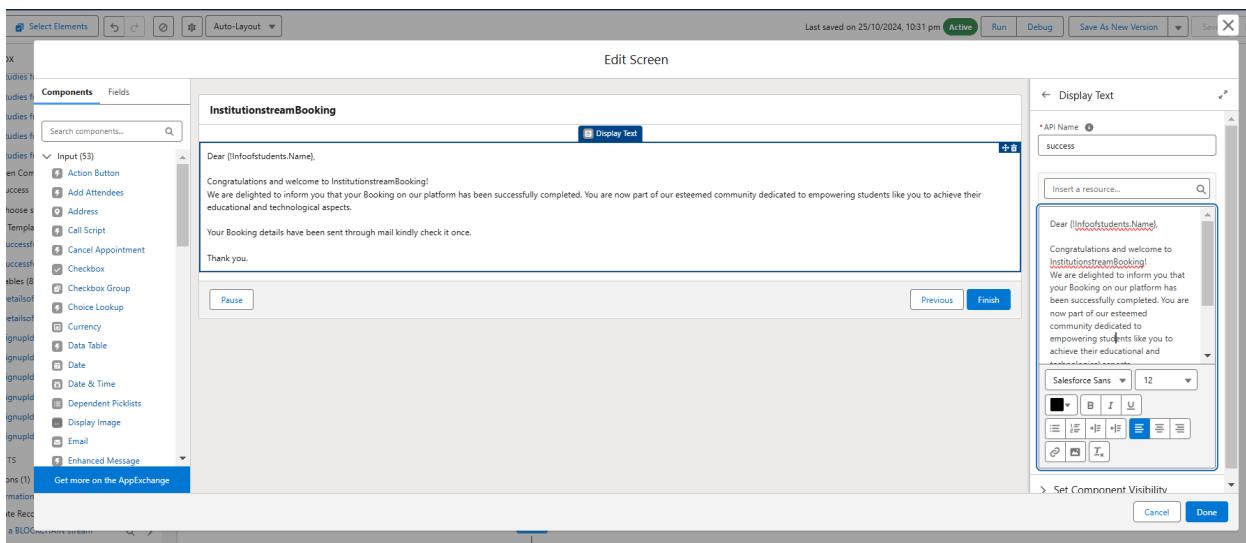
- After the confirmation action element, add the screen element.
- give name as perfectjob.
- In the left corner, in the components, search display text.
- grab it on to the main screen.
- give name as success.
- In the body, give the below text:

Dear {!Infoofstudents.Name},  
 Congratulations and welcome to InstitutionstreamBooking!  
 We are delighted to inform you that your Booking 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 technological aspects.

Your Booking details have been sent through mail kindly check it once.

Thank you.

► click on done.



► Now click on save at the top right corner and give name as InstitutionstreamBooking.

### **TASK-3 Create a User and make suitable settings**

#### **3.1 Create a User**

► In the setup , go to quick find box, search users.

► click on new user.

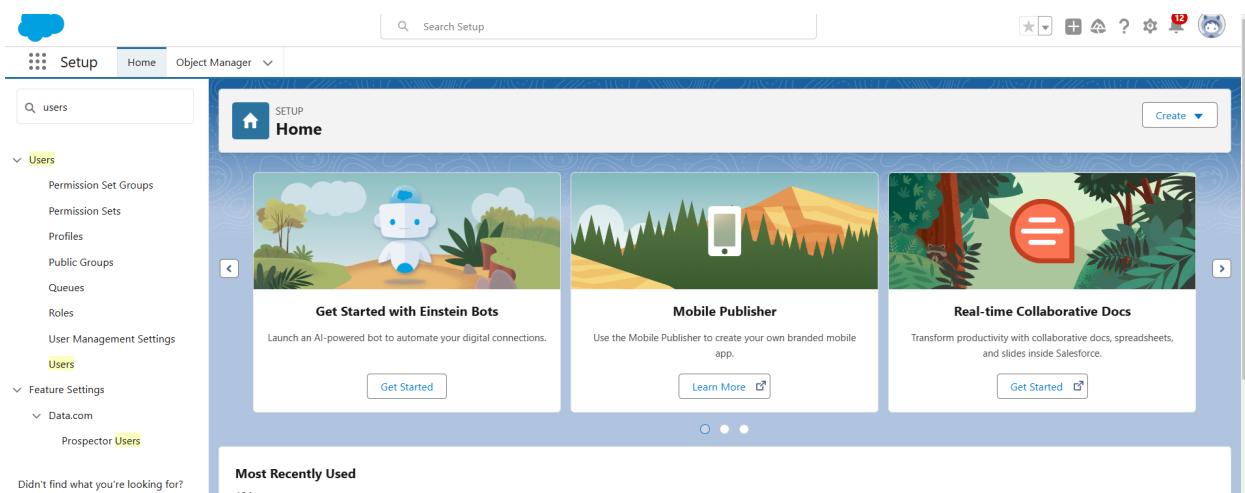
► give the following values:

name : Advisor

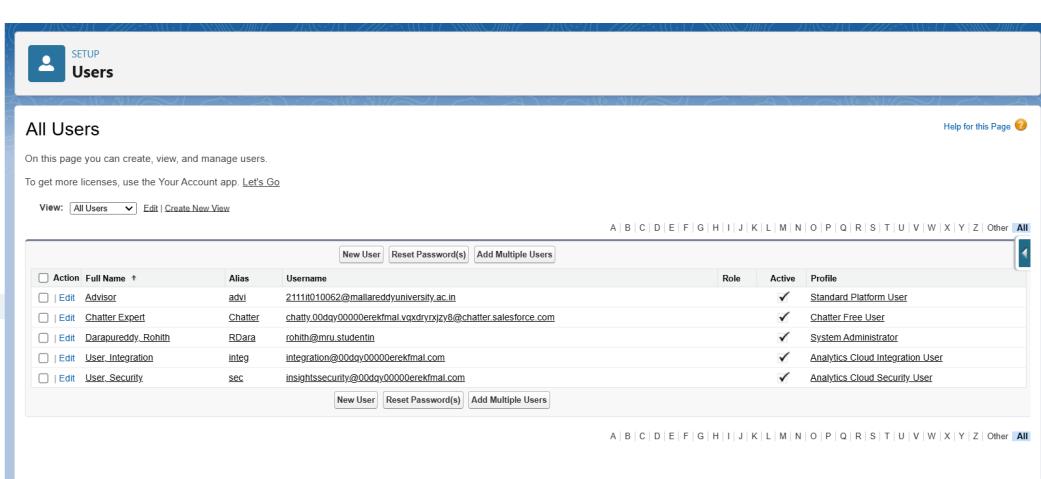
nickname: Advi

mail : 2111IT010062@mallareddyuni.ac.in

username: Advi@blue  
 License : Salesforce Platform  
 Profile : Standard Platform User  
 ➤ click on save.



The screenshot shows the Salesforce Setup Home page. On the left, there's a sidebar with a search bar and navigation links for Setup, Home, Object Manager, and various system settings. The main area features three promotional cards: "Get Started with Einstein Bots" (Launch an AI-powered bot to automate your digital connections), "Mobile Publisher" (Use the Mobile Publisher to create your own branded mobile app), and "Real-time Collaborative Docs" (Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce). Below these cards is a section titled "Most Recently Used" which includes a link to "Users".

The screenshot shows the "All Users" page under the "Users" section of the Setup menu. It displays a list of users with columns for Action, Full Name, Alias, Username, Role, Active status, and Profile. The users listed are:

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> <a href="#">Edit</a>	Advisor	advi	2111lt010062@mailareddyuniiversity.ac.in		<input checked="" type="checkbox"/>	Standard Platform User
<input type="checkbox"/> <a href="#">Edit</a>	Chatter Expert	Chatter	chaty_00day00000erekfm@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/> <a href="#">Edit</a>	Darapureddy_Rohith	RDara	rohit@mru.studentin		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/> <a href="#">Edit</a>	User_Integration	Integ	integration@00day00000erekfm@com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/> <a href="#">Edit</a>	User_Security	sec	insightsecurity@00day00000erekfm@com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

### 3.2 Configure the user settings

- In the setup , go to quick find box, search users.
- click on edit next to your name.
- scroll down to last , under approver settings,select advisor at manager.
- click on save.

The screenshot shows the Salesforce Setup interface with the following sections:

- Single Sign On Information:** Federation ID
- Locale Settings:**
  - Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)
  - Locale: English (India)
  - Language: English
- Approver Settings:**
  - Delegated Approver
  - Manager: Advisor
  - Receive Approval Request Emails: Only if I am an approver

At the bottom are Save, Save & New, and Cancel buttons.

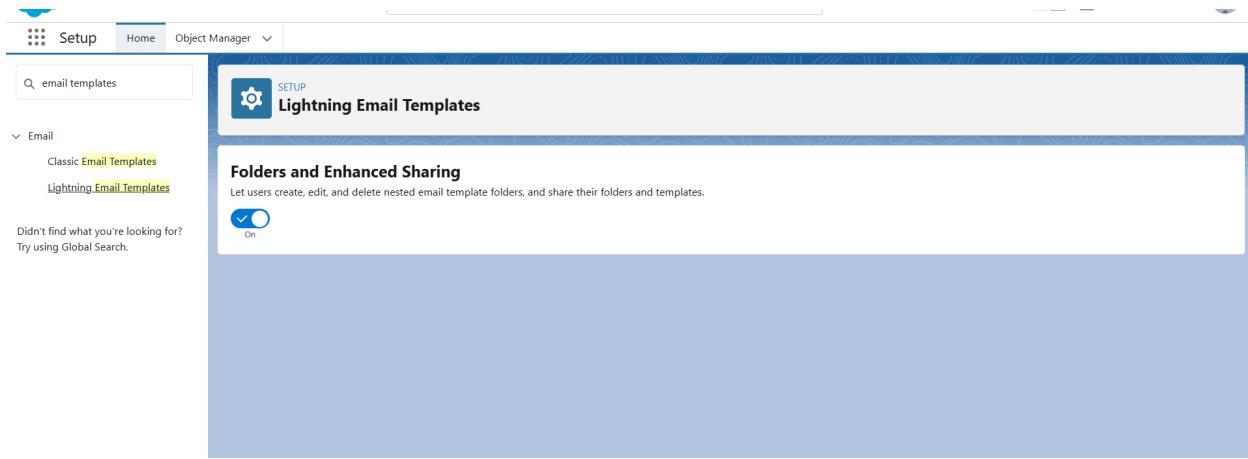
## TASK-4 Creating Email Templates and Approval Processes

### 4.1 create email template

- In the setup , go to quick find box, search Email Templates.
- click on Lightning Email Templates.
- Under the folders and Enhanced sharing, turn on the toggle.

The screenshot shows the Salesforce Home page with the following elements:

- Search Bar:** Search Setup
- Setup Tab:** Setup
- Home Tab:** Home
- Object Manager:** Object Manager
- Search Bar:** Q Email templates
- Categories:**
  - Email
  - Classic Email Templates
  - Lightning Email Templates
- Message:** Didn't find what you're looking for? Try using Global Search.
- Home Cards:**
  - Get Started with Einstein Bots:** Launch an AI-powered bot to automate your digital connections. Get Started
  - Mobile Publisher:** Use the Mobile Publisher to create your own branded mobile app. Learn More
  - Real-time Collaborative Docs:** Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce. Get Started
- Most Recently Used:** 10 items (Rohith Darapureddy, User)
- System Navigation:** Cloud, Home, Object Manager, Help, Notifications
- Desktop Taskbar:** Icons for File, Find, Home, Object Manager, Help, Notifications, and more.
- System Status:** Weather (78°F), Haze, ENG IN, 19:17, 26-10-2024, Bell icon



- Go to app launcher, search for Email templates.
- click on new folder, give email as a name to the folder.
- click on new email template.
- give confirmation template as the name for the email template
- select the folder as email.
- Write the below text in the message content:

Dear {{Booking\_\_c.Student\_Name\_\_c}},

I hope this email finds you well. I am writing to confirm the details of our upcoming booking scheduled for {{Booking\_\_c.Booking\_Date\_Time\_\_c}} regarding {{Booking\_\_c.SOP\_Topic\_\_c}}.

Booking Details:

Booking No: {{Booking\_\_c.Name}}

Student Name: {{Booking\_\_c.Student\_Name\_\_c}}

Advisor's Name: {{Booking\_\_c.Advisors\_\_c}}

Date & Time: {{Booking\_\_c.Booking\_Date\_Time\_\_c}}

Purpose: {{Booking\_\_c.SOP\_Topic\_\_c}}

I want to assure you that I am looking forward to our meeting and am prepared to address any questions or concerns regarding {{Booking\_\_c.SOP\_Topic\_\_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 specific topics or questions to discuss during our appointment, please feel free to share them in advance to ensure our time together is as productive as possible.

If you need to reschedule or cancel, please notify me at your earliest convenience to make alternative arrangements.

Thank you for choosing to work with me. I am confident our collaboration will lead to positive outcomes toward your goals.

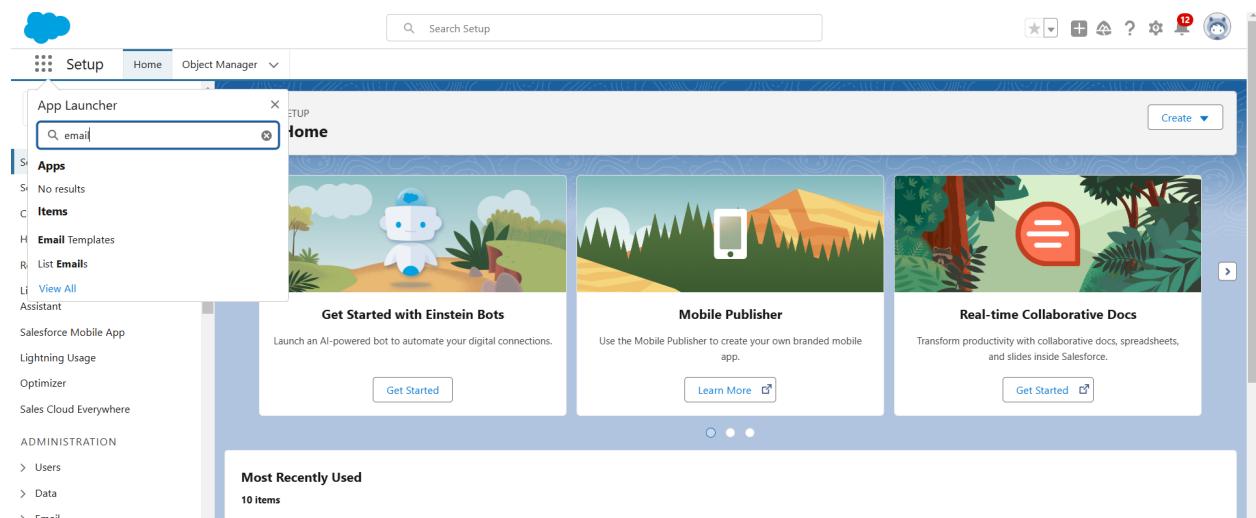
If you have any questions or require further information, please don't hesitate to reach out.

Looking forward to our meeting.

Best regards,

**{{Recipient.Name}}**

Institutionstream Booking



The screenshot shows the institution's dashboard with a sidebar on the left containing navigation links like Bookings, students, studies, Advisors, Infrastructures, Details, Reports, Signups, Rankings, contacts, and Dashboards. The main area displays 'Email Templates' under 'All Folders'. A table lists one item: 'email' by Rohith Darapureddy, last modified on 25/10/2024, 10:44 pm. There are also sections for 'EMAIL TEMPLATES' (Recent, Created by Me, Private Email Templates, Public Email Templates, All Email Templates) and 'FOLDERS' (All Folders, Created by Me, Shared with Me).

The screenshot shows the 'New Email Template' dialog. At the top right, it says '\* = Required Information'. The 'Information' section includes fields for 'Email Template Name' (with a required asterisk), 'Related Entity Type' (set to 'None'), 'Description' (empty), and 'Folder' (set to 'Private Email Templates' with a 'Select Folder' button). The 'Message Content' section includes fields for 'Subject' (empty) and 'Enhanced Letterhead' (with a search bar 'Search Enhanced Letterheads...'). At the bottom right are 'Cancel' and 'Save' buttons.

Search... X

\* = Required Information

**Information**

\* Email Template Name: confirmation template

Related Entity Type: Booking

Description:

Folder: email [Select Folder](#)

**Message Content**

Subject:

Enhanced Letterhead:  [Search Enhanced Letterheads...](#) Q

HTML Value:

Dear {{Booking\_c.Student\_Name\_c}},  
I hope this email finds you well. I am writing to confirm the details of our upcoming booking scheduled for {{Booking\_c.Booking\_Date\_Time\_c}} regarding {{Booking\_c.SOP\_Topic\_c}}.

**Booking Details:**  
Booking No: {{Booking\_c.Name}}  
Student Name: {{Booking\_c.Student\_Name\_c}}  
Advisor's Name: {{Booking\_c.Advisors\_c}}  
Date & Time: {{Booking\_c.Booking\_Date\_Time\_c}}  
Purpose: {{Booking\_c.SOP\_Topic\_c}}

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

► click on save.

## 4.2 create email template

- click on new email template.
- give approval template as the name for the email template
- select the folder as email.
- Write the below text in the message content:

Dear {{Booking\_c.Student\_Name\_c}},

I hope this email finds you well. I am writing to give you the message regarding the approval of your booking at {{Booking\_c.Booking\_Date\_Time\_c}} regarding

**{{Booking\_\_c.SOP\_Topic\_\_c}}.**

Booking Details:

Booking No: {{Booking\_\_c.Name}}

Student Name: {{Booking\_\_c.Student\_Name\_\_c}}

Advisor's Name: {{Booking\_\_c.Advisors\_\_c}}

Date & Time: {{Booking\_\_c.Booking\_Date\_Time\_\_c}}

Purpose: {{Booking\_\_c.SOP\_Topic\_\_c}}

I am very glad to have you on board with us and experiencing our prestigious institution and I am eagerly waiting for our meeting and am prepared to address any questions or concerns regarding {{Booking\_\_c.SOP\_Topic\_\_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 specific topics or questions to discuss during our interaction, please feel free to ask and enquire them so that you can have a better understanding about the institution and the services provided by the institution to the students.

Thank you for choosing us. I am confident our collaboration will lead to positive outcomes toward your goals.

If you have any questions or require further information, please don't hesitate to reach out.

Looking forward to our meeting.

Best regards,

{{Recipient.Name}}

Institutionstream Booking

\* = Required Information

**Information**

Email Template Name: Approval Template

Related Entity Type: Booking

Description:

Folder: email [Select Folder](#)

**Message Content**

Subject:

Enhanced Letterhead: Search Enhanced Letterheads... [Q](#)

HTML Value:

Dear {{Booking\_c.Student\_Name\_c}},  
I hope this email finds you well. I am writing to give you the message regarding the approval of your booking at {{Booking\_c.Booking\_Date\_Time\_c}} regarding {{Booking\_c.SOP\_Topic\_c}}.

**Booking Details:**  
Booking No: {{Booking\_c.Name}}  
Student Name: {{Booking\_c.Student\_Name\_c}}  
Advisor's Name: {{Booking\_c.Advisors\_c}}  
Date & Time: {{Booking\_c.Booking\_Date\_Time\_c}}  
Purpose: {{Booking\_c.SOP\_Topic\_c}}

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

### 4.3 create email template

- click on new email template.
- give rejection template as the name for the email template
- select the folder as email.

Write the below text in the message content:

Dear {{Booking\_c.Student\_Name\_c}},

I hope this email finds you well. I am writing this email to convey you that your request has been cancelled due to some issues.

**Booking Details:**

Booking No: {{Booking\_c.Name}}

Student Name: {{Booking\_c.Student\_Name\_c}}

Advisor's Name: {{Booking\_c.Advisors\_c}}

Date & Time: {{Booking\_\_c.Booking\_Date\_Time\_\_c}}

Purpose: {{Booking\_\_c.SOP\_Topic\_\_c}}

A small issue has been occurred so that there is a rejection of your booking regarding {{Booking\_\_c.SOP\_Topic\_\_c}}.

I suggest you to check all the necessary things and make a book again.

Thank you for choosing us. I am confident our collaboration will lead to positive outcomes toward your goals.

If you have any questions or require further information, please don't hesitate to reach out.

Looking forward to our meeting.

Best regards,

{{Recipient.Name}}

Institutionstream Booking

\* = Required Information

### Information

<p>* Email Template Name  <input type="text" value="Rejection Template"/></p>	<p>Related Entity Type  <input type="text" value="Booking"/></p>
<p>Description  <input type="text"/></p>	
<p>Folder  <input type="text" value="email"/> <a href="#">Select Folder</a></p>	

### Message Content

<p>Subject  <input type="text"/></p>	<p>Enhanced Letterhead  <input type="text"/> <a href="#">Search Enhanced Letterheads...</a> </p>
<p>HTML Value  <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;">Source</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;">Font</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;">Size</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"><b>B</b></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"><i>I</i></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"><u>U</u></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"><a href="#">A</a></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 5px;"></span> </p>	

Dear {{Booking\_c.Student\_Name\_c}},

I hope this email finds you well. I am writing this email to convey you that your request has been cancelled due to some issues.

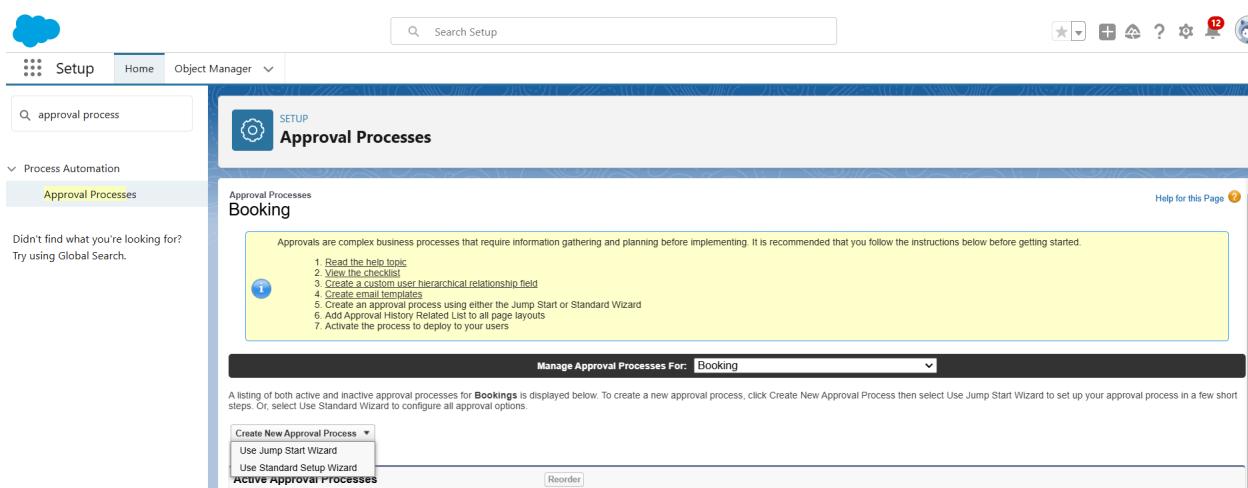
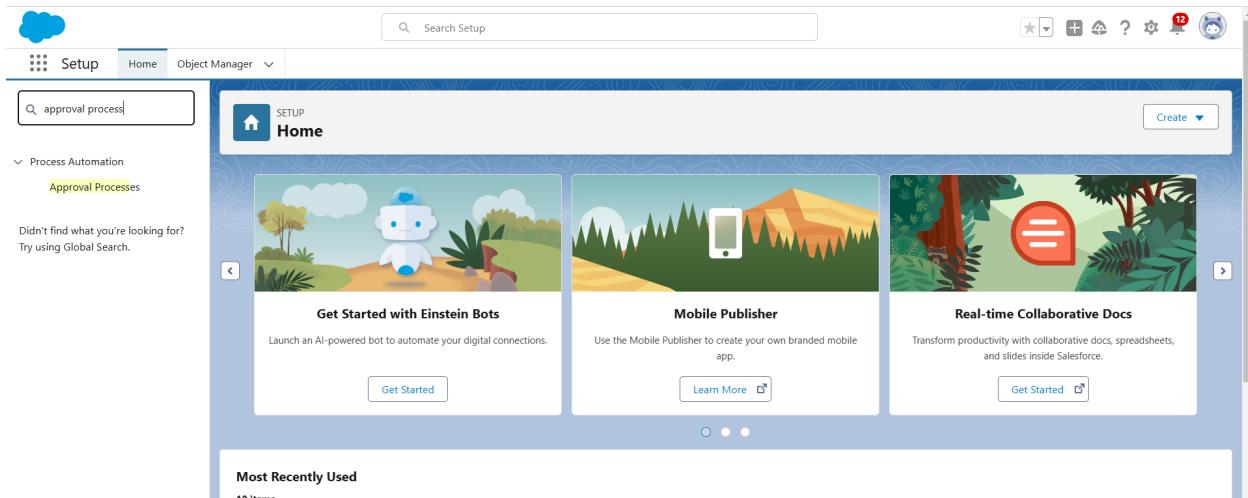
**Booking Details:**

Booking No: {{Booking\_c.Name}}  
Student Name: {{Booking\_c.Student\_Name\_c}}  
Advisor's Name: {{Booking\_c.Advisors\_c}}  
Date & Time: {{Booking\_c.Booking\_Date\_Time\_c}}  
Purpose: {{Booking\_c.SOP\_Topic\_c}}

[Cancel](#) Save

#### 4.4 create an approval process

- In the quick find box, search for approval processes.
- click on the approval processes,In manage approval process for select Booking.
- click on create new approval processes with the option using jump start wizard.



- Give name as Booking approval and scroll down.
- for the option Automatically assign the approver using a standard or custom hierarchy field, select manager.
- click on save.
- Now click on edit which is beside the created approval process.
- now click on next,next.
- Then give Manager for Next Automated Approval determined by.
- Then in the Record Editability Settings, select Administrators OR the currently assigned approver can edit records during the approval process.
- click on next,next and save.

The screenshots illustrate the configuration of an Approval Process in Salesforce:

- Top Screenshot:** Approval Process Jump Start Wizard for 'Booking'. It shows the 'Approval Process Information' section with fields for Name ('Booking approval'), Unique Name ('Booking\_approval'), and Approval Assignment Email Template. It also includes a checkbox for 'Use Approver Field of Booking Owner'.
- Middle Screenshot:** Approval Process Jump Start Wizard for 'Booking'. It shows the 'Specify Entry Criteria' section, which allows defining conditions for the approval process to run. A table is provided for specifying fields, operators, and values.
- Bottom Screenshot:** Step 3 of 6 for 'Booking approval'. It shows the 'Select Approver' section, where the approver is automatically assigned using a standard or custom hierarchy field ('Manager'). It also shows the 'Step 3. Specify Approver Field and Record Editability Properties' section, which includes a note about routing requests to different users and a table for defining record editability properties.

- click on the created approval process, go to intial submission actions.
- click on add new, select field update.
- give name submitted, field to update as status, piclist value as

pending and then click on save.

- click on add new under intial submission actions, select email alert.
- give name submission email alert.
- give email template as submission template under the classic email template of email folder.
- select your name from available requirements and add it to teh selected requirements.
- click on save.

The screenshot shows the Salesforce Setup interface with the following details:

**Approval Processes Page:**

- Process Name: Booking approval
- Unique Name: Booking\_approval
- Description:
- Entry Criteria:
- Record Editability: Administrator OR Current Approver
- Next Automated Approver Determined By: Manager of Record Submitter
- Approval Assignment Email Template:
- Initial Submitters: Booking Owner
- Created By: Rohith Darapureddy, 25/10/2024, 11:43 pm
- Modified By: Rohith Darapureddy, 26/10/2024, 8:16 pm
- Initial Submission Actions:**
  - Action: Record Lock
  - Type: Task
  - Description: Lock the record
  - Action: Email Alert
  - Type: Email Alert
  - Description: Send an email alert
  - Action: Field Update
  - Type: Field Update
  - Description: Update field value

**Field Update Edit Dialog:**

**Identification:**

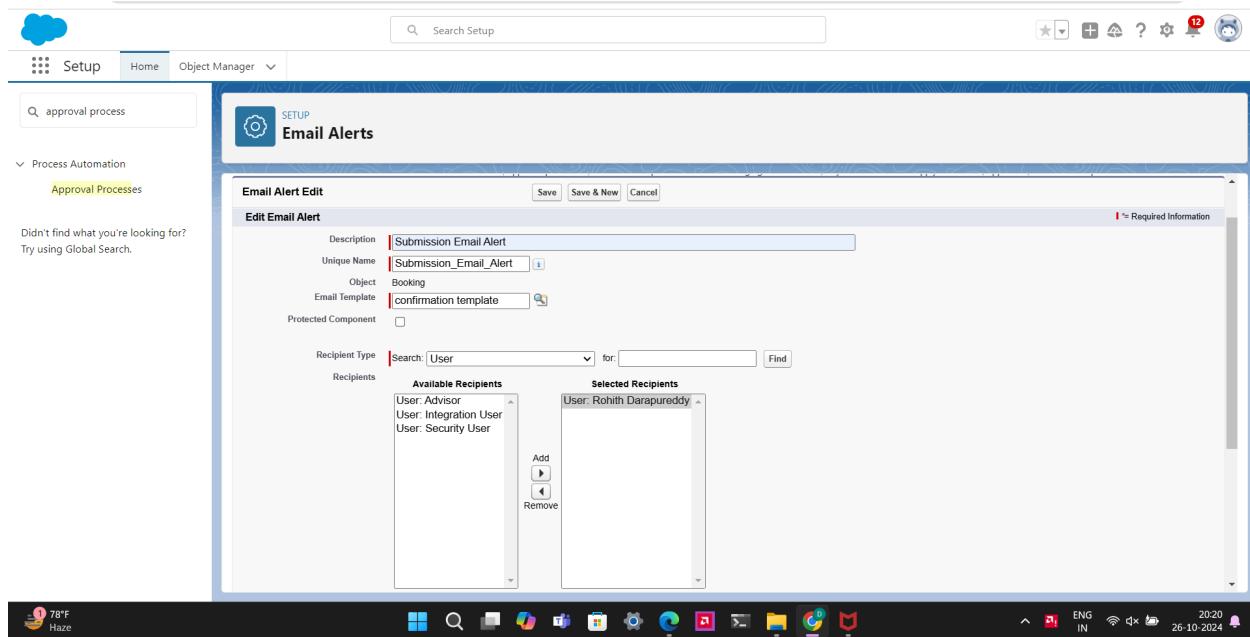
- Name: Submitted
- Unique Name: Submitted
- Description:
- Object: Booking
- Field to Update: Status
- Field Data Type: Picklist
- Re-evaluate Workflow Rules after Field Change:

**Specify New Field Value:**

**Picklist Options:**

- The value above the current one
- The value below the current one
- A specific value: pending

Buttons at the bottom: Save, Save & New, Cancel.



- click on the created approval process, go toFinal Approval actions.
- click on add new, select field update.
- give name Approved, field to update as status, piclist value as pending and then click on save.
- click on add new under Final Approval actions, select email alert.
- give name Approval email alert.
- give email template as Approval template under the classic email template of email folder.
- select your name from available requirements and add it to teh selected requirements.
- click on save.

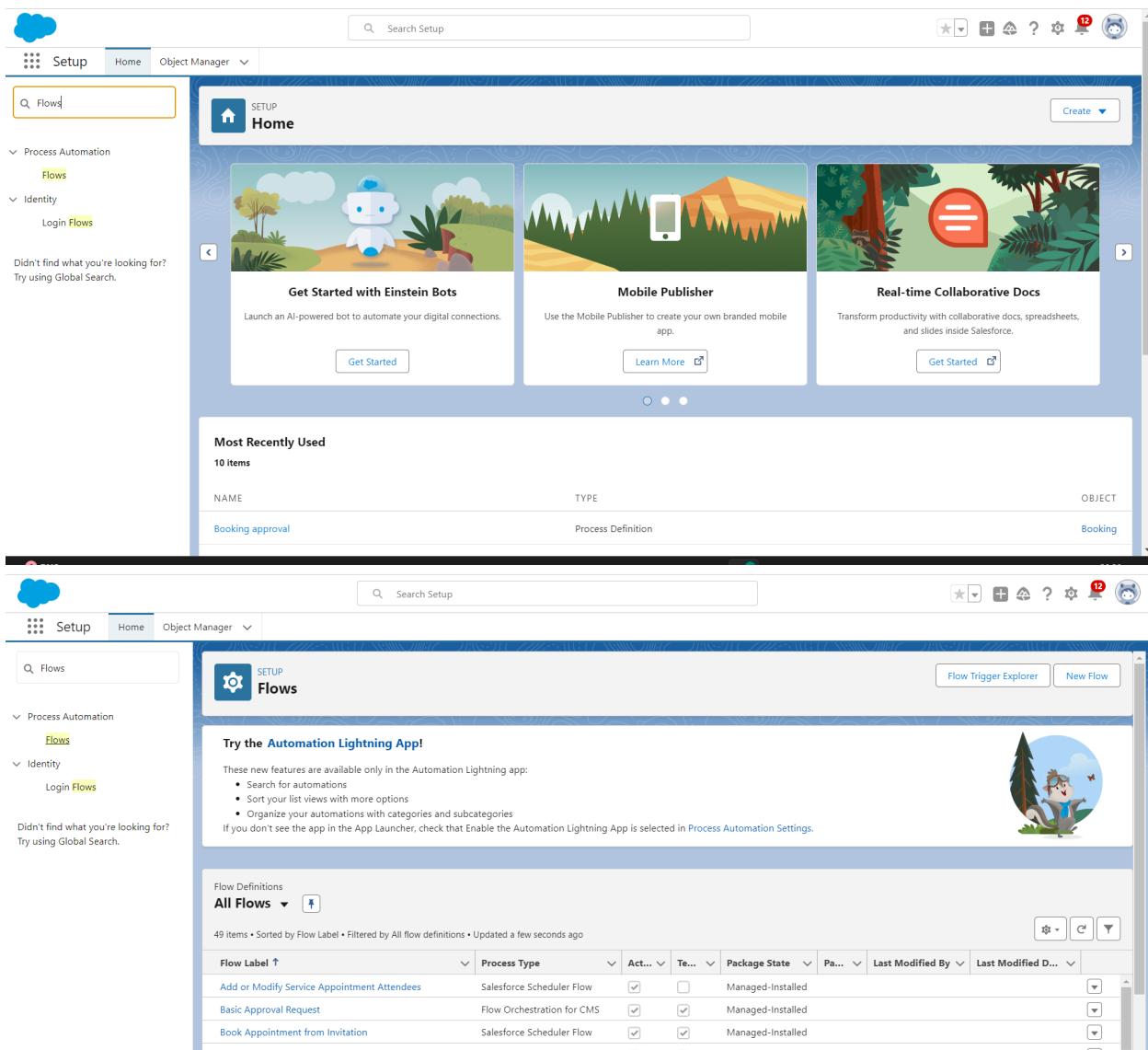
- click on the created approval process, go to Final Rejection actions.
- click on add new, select field update.
- give name Rejected, field to update as status, piclist value as pending and then click on save.
- click on add new under Final Rejection actions, select email alert.
- give name Rejection email alert.
- give email template as Rejection template under the classic email template of email folder.
- select your name from available requirements and add it to teh selected requirements.
- click on save.

The top screenshot shows the 'Field Updates' page in the Salesforce Setup. A new field update is being created for the 'Rejected' status on the 'Booking' object. The 'Field Data Type' is set to 'Picklist'. In the 'Specify New Field Value' section, the 'Picklist Options' are set to 'A specific value: pending'. The bottom screenshot shows the 'Email Alerts' page in the Salesforce Setup. A new email alert is being created for 'Rejection Email Alert'. The 'Email Template' is set to 'Rejection Template'. Under 'Recipients', 'User: Rohith Darapureddy' is selected from the 'Available Recipients' list.

## TASK-5 Record Triggered Flow Creation

### 5.1 create email template

- Go to the quick find box, search for flows.
- click on flows, click on new flow, click on Record-Triggered Flow and click on create.
- The configure window opens.
- give Booking as object.



## 5.2 Add action element

- Add an action element after the start, give name as subflow.
- set Record.createdbyid value for the set inout values for the selected action field.
- click on save.
- click on save and give name as InstitutonstreamBookingadvancedflow.

**Select Type**

**Recommended**

- Screen Flow**  
Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.
- Record-Triggered Flow**  
Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.
- Schedule-Triggered Flow**  
Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.
- Platform Event—Triggered Flow**  
Launches when a platform event message is received. This autolaunched flow runs in the background.
- Autolaunched Flow (No Trigger)**  
Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.
- Record-Triggered Orchestration**  
Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

**All Flow Types**

- Autolaunched Flow (No Trigger)
- Autolaunched Orchestration (No Trigger)

**Flow Builder**

**Start** Record-Triggered Flow

Object: Booking  
Trigger: A record is created  
Optimize for: Actions and Related Records

+ Add Scheduled Paths (Optional)  
Open Flow Trigger Explorer for Booking

**End**

**Configure Start**

**Select Object**  
Select the object whose records trigger the flow when they're created, updated, or deleted.  
\* Object: Booking

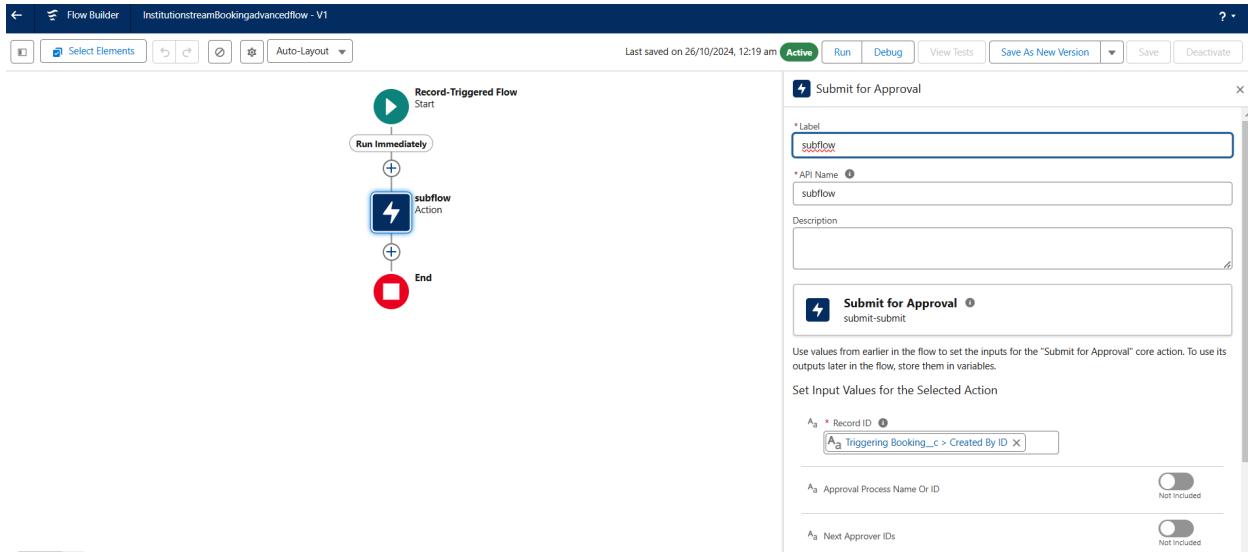
**Configure Trigger**

\* 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

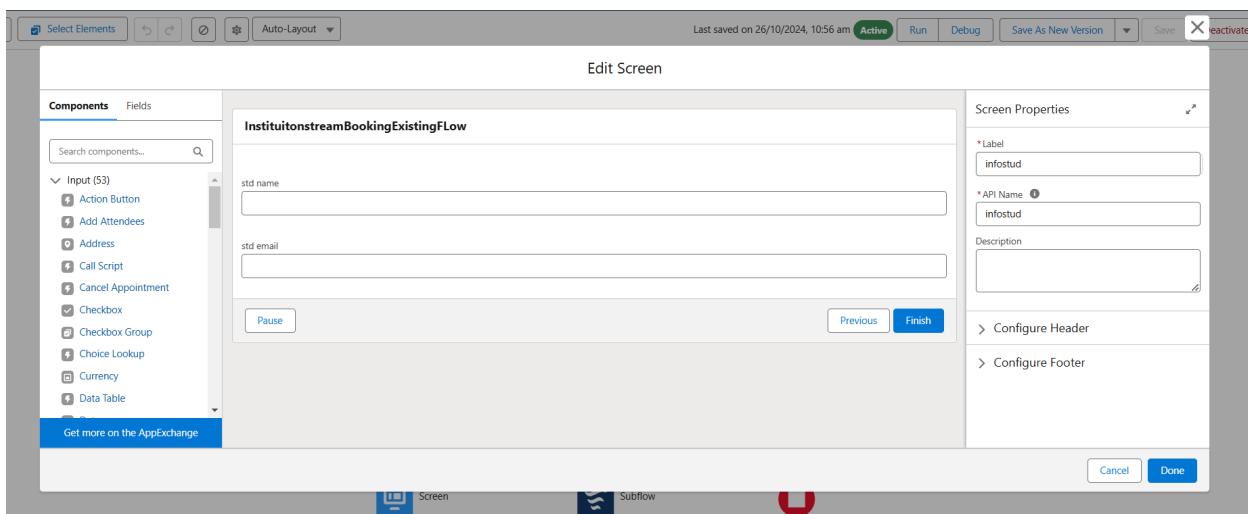
\* Optimize the Flow for:



## TASK-6 Existing student flow for a Booking

### 6.1 Add a screen

- Go to the quick find box, search for flows.
- click on flows, click on new flow, click on screen Flow and click on create.
- Click on the + after the start, click on screen,give infostud as the screen name.
- Now in the left side corner, go to components, search for text.
- Add two text components to the screen.
- name the two text components as std name & std email,
- click on Done.



## 6.2 Add GET Record Element

- After the infostud element, add a GET record element.
- name the element as access data.
- set object name as student.
- place the following fields in the student records:  
field : Student\_Name\_\_c ; value : std\_name,  
field : Email\_\_c ; value : std\_email
- Click on save.

The screenshot shows the configuration interface for a 'Get Records' element. At the top, there are buttons for 'Run', 'Debug', 'Save As New Version', 'Save', and 'Deactivate'. The 'Active' button is highlighted. Below this, the 'Get Records' configuration screen is displayed.

**Get Records**

**Label:** access data    **API Name:** access\_data

**Description:** (empty)

**Get Records of This Object**

**Object:** student

**Filter student Records**

**Condition Requirements:** All Conditions Are Met (AND)

**Filter Criteria:**

- Field: Student\_Name\_\_c, Operator: Equals, Value: A\_a std\_name
- AND  
Field: Email\_\_c, Operator: Equals, Value: A\_a std\_email

**Add Condition:** + Add Condition

**Sort student Records**

**Sort Order:** (empty)

## 6.3 Add Decision Element

- After the access data element, add a decision element.
- name the decision element as Booking or Report.
- Under the outcome details, give label as Booking details.
- place the following values:  
Resource : How\_may\_I\_help\_you ; value : Book\_an\_Appointment.
- click on + beside the outcome order.
- label as report details.
- place the following values:  
Resource : How\_may\_I\_help\_you ; value : Book\_an\_Appointment.
- click on save.

The screenshot shows the configuration of a 'Decision' element. At the top, there are buttons for 'Run', 'Debug', 'Save As New Version', 'Save', and 'Deactivate'. The main area has tabs for 'Decision' and 'X'. It includes fields for 'Label' (Booking or Report) and 'API Name' (Booking\_or\_Report). A 'Description' text area is present. Below this, under 'Outcomes', it says 'For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.' There is a table for outcomes:

OUTCOME ORDER	OUTCOME DETAILS	Actions
Booking details	* Label: Booking details * Outcome API Name: Booking_details	Delete Outcome
report details		
Default Outcome		

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

Resource: A\_a How\_may\_I\_Help\_You, Operator: Equals, Value: A\_a Book\_An\_Appointment

+ Add Condition

Last saved on 26/10/2024, 10:56 am Active Run Debug Save As New Version Save Deactivate

**Decision**

\* Label: Booking or Report

\* API Name: Booking\_or\_Report

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
Booking details	* Label: report details * Outcome API Name: report_details	<span>Delete Outcome</span>
report details		
Default Outcome	All Conditions Are Met (AND)	
	Resource: A_a How_may_I_Help_You Operator: Equals Value: A_a Book_An_Appointment	<span>Add Condition</span>

## 6.4 Add screen

- Add a screen in the booking details path.
- give name as schedule Booking screen.
- Go to the fields in the top left side corner, click on record variable, click on new resource.
- label it as Booking,give object as Booking,click on done.
- Drag the necessary Booking object fields into the screen,click on save.

Last saved on 26/10/2024, 10:56 am Active Run Debug Save As New Version Save Deactivate

**Edit Screen**

**Components**

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

**InstitutionstreamBookingExistingFlow**

**Advisors**

No preview is available for this component.

**Fields**

Booking No.

SOP/Topic

Student Name

**Screen Properties**

\* Label: Schedule Booking Screen

\* API Name: Schedule\_Booking\_Screen

Description:

> Configure Header

> Configure Footer

Cancel Done

## 6.5 Add GET Record Element

- After the Schedule Booking Screen element, add a GET record element.
- name the element as Get Booking, set object name as Advisors.
- place the following fields in the student records:  
field : Name ; value : {!Booking.Advisors\_\_c}.
- click on save.

The screenshot shows the configuration interface for a 'Get Records' element. At the top, there are buttons for 'Run', 'Debug', 'Save As New Version', 'Save', and 'Deactivate'. Below this, the 'Label' is set to 'Get Booking' and the 'API Name' is 'Get\_Booking'. The 'Object' is set to 'Advisors'. Under 'Condition Requirements', it says 'All Conditions Are Met (AND)'. A single condition is defined: 'Name' field equals 'Booking > Advisors'. The 'Sort Order' is set to 'Not Sorted'. A warning message states: '⚠ If you store only the first record, filter by a unique field, such as ID.'

## 6.6 Add a Create Record Element

- After the Get Booking element, add a create record element.
- label it as Make a Book, set object as Booking.
- set the following field values :

```
field1 :Advisors__c ; value1 :{!Get_Booking.Id},  
field1 :Booking_DateTime__c; value1 :{!Booking.Booking_DateTime__c} ,  
field1 :Notes__c; value1 :{!Booking.Notes__c} ,  
field1 :SOPTopic__c ; value1 :{!Booking.SOPTopic__c},  
field1 :Student_Name__c ; value1 :{!access_data.Id}
```

- Click on save.

The screenshot shows the configuration of a 'Create Records' element. At the top, there are buttons for 'Run', 'Debug', 'Save As New Version', 'Save', and 'Deactivate'. The main area is titled 'Create Records' and contains the following fields:

- Label:** Make a Book
- API Name:** Make\_a\_Book
- Description:** (empty)
- How to set record field values:** Manually
- Create a Record of This Object:** Object: Booking
- Set Field Values for the Booking:** A table with five rows, each showing a field from the 'Get Booking' element and its corresponding value from the 'access\_data' object.

Field	Value
Aa Advisors	Aa Advisors from Get Booking > Record ID
Booking Date/Time	Booking > Booking Date/Time
Aa Notes	Aa Booking > Notes
Aa SOP/Topic	Aa Booking > SOP/Topic
Aa Student Name	Aa student from access data > Record ID
- Add Field:** + Add Field
- Manually assign variables:** (checkbox)

## 6.7 Add screen

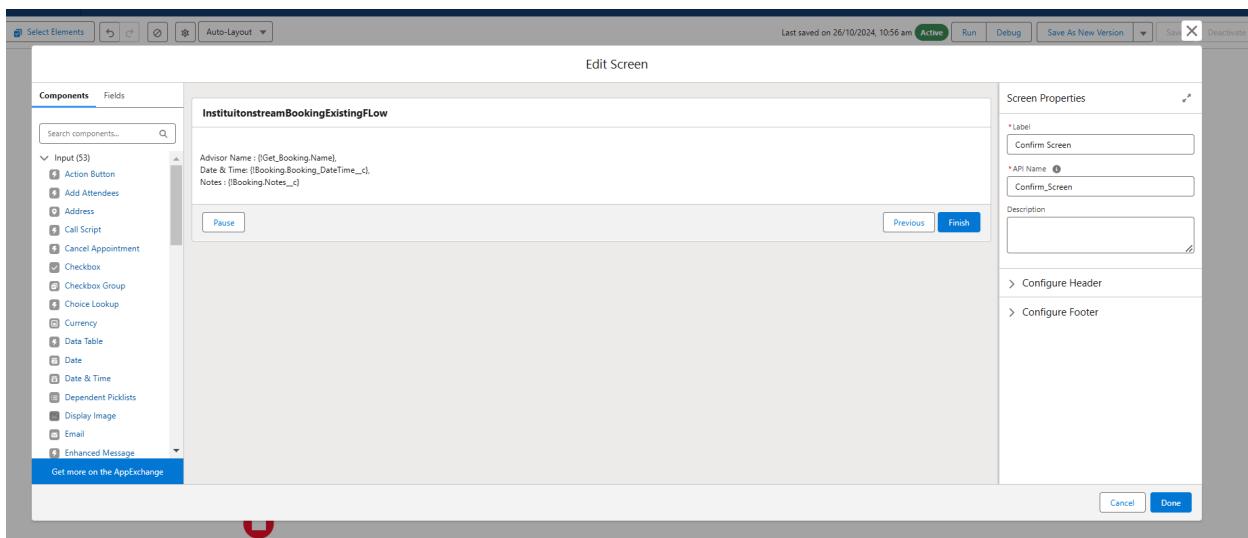
- Add a screen after the make a book element,give name as confirm screen.
- Go to the components in the left side corner, search for display text.
- drag the display text component onto the main screen,give BookingSuccess as the name.
- place the following text in the display text body:

Advisor Name : {!Get.Booking.Name},

Date & Time: {!Booking.Booking\_DateTime\_c},

Notes : {!Booking.Notes\_c}

- click on done.



## 6.8 Add screen

- Add a subflow element in the report details path.
- select create a case and label teh subflow as make a stud report.
- save the flow with the name InstitutionstramBookingExistingFlow.

**Create a Case**

---

\* Label  
make a stud report

\* API Name ⓘ  
make\_a\_stud\_report

Description

Referenced Flow

**Create a Case**  
setup\_service\_experience\_Create\_Case

Use values from the parent flow to set the inputs for the "Create a Case" 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 "Create a Case" flow.

## TASK-7 Flow to combine all flows

### 7.1 Add screen

- Add a screen after the start element,give name as welcome.
- Go to the components in the left side , search for display text.
- drag the display text component onto the main screen,give confirmsms as the name.
- place the followimng text:

Welcome to InstitutionstreamBooking

*your dream place to achieve your academic goals*

At InstitutionstreamBooking, we understand that embarking on educational aspects can be more interesting. 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, or

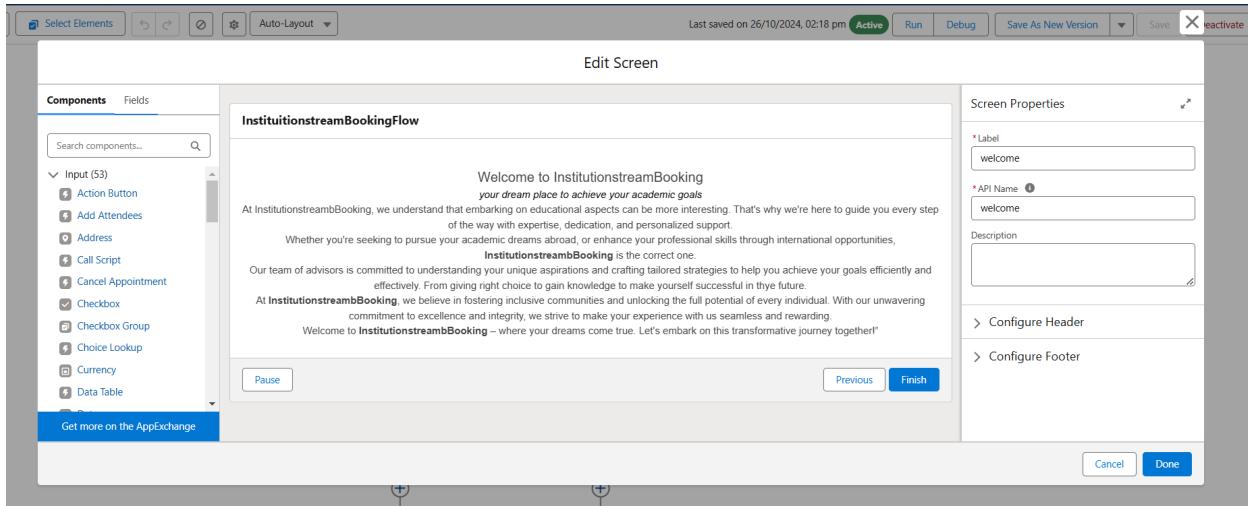
enhance your professional skills through international opportunities,  
**InstitutionstreamBooking** is the correct one.

Our team of advisors is committed to understanding your unique aspirations and crafting tailored strategies to help you achieve your goals efficiently and effectively. From giving right choice to gain knowledge to make yourself successful in thy future.

At **InstitutionstreamBooking**, 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 **InstitutionstreamBooking** – where your dreams come true.  
Let's embark on this transformative journey together!"

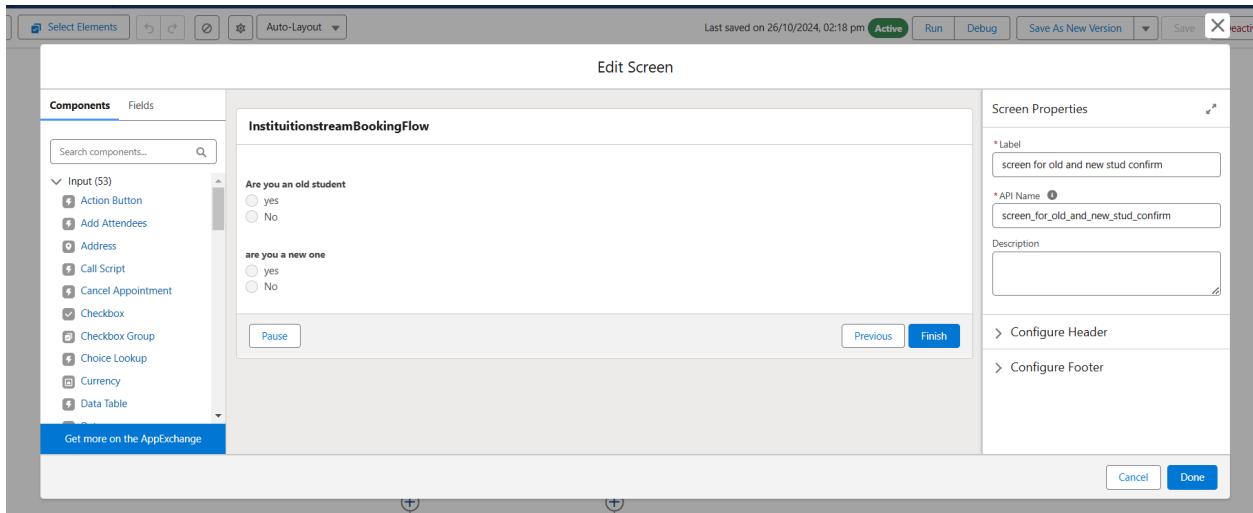
► click on done.



## 7.2 Add screen

- Add a screen after the welcome screen element,give name as screen for old and new stud confirm.
- Go to the components in the left side , search for radio buttons.
- drag the radio button component onto the main screen,give are you an old student as the name.
- create new choices yes and no.

- drag the radio button component onto the main screen,give are you an new one as the name.
- create new choices yes and no.
- click on done.



### 7.3 Add Decision Element

- After the screen for old and new stud confirm element, add a decision element,name the decision element as making a right choice.
- Under the outcome details, give label as if old student.
- place the following values:  
Resource :{!Are\_you\_an\_old\_student} ; value :{!yes}.
- click on + beside the outcome order,label as if new studuent.
- place the following values:  
Resource :{!are\_you\_a\_new\_one} ; value :{!No}.
- click on save.

Last saved on 26/10/2024, 02:18 pm Active Run Debug Save As New Version ▼ Save Deactivate

### Decision

* Label	* API Name
making a right choice	making_a_right_choice
Description	
<pre>graph TD     Start(( )) --&gt; Decision{Decision}     Decision --&gt; IfOld[if old student]     Decision --&gt; IfNew[if new student]     IfOld --&gt; OldStudentFlow[InstitutionStreamBookingExistingFFlow]     IfOld --&gt; Default[Default Outcome]     IfNew --&gt; NewStudentFlow[InstitutionStreamBookingExistingFFlow]     Default --&gt; AllConditions[All Conditions Are Met (AND)]     AllConditions --&gt; Condition[Condition Requirements to Execute Outcome]     Condition --&gt; Resource[Resource: Aa ...nfirm &gt; Are you an old student]     Condition --&gt; Operator[Operator: Equals]     Condition --&gt; Value[Value: Aa yes]</pre>	

**OUTCOME ORDER** + i **OUTCOME DETAILS** Delete Outcome

if old student if old student if old student

if new student if new student if new student

**Default Outcome** All Conditions Are Met (AND)

**Condition Requirements to Execute Outcome**

Resource: Aa ...nfirm > Are you an old student Operator: Equals Value: Aa yes X

+ Add Condition

## 7.4 Add Subflow

- In the if old student path, add a sub flow, select InstitutionStreamBookingExistingFFlow and label the subflow as old flow.
- click on save.

**InstituitonstreamBookingExistingFLow**

\* Label: old flow

\* API Name: old\_flow

Description:

Referenced Flow:

**InstituitonstreamBookingExistingFLow**  
InstituitonstreamBookingExistingFLow

Use values from the parent flow to set the inputs for the "InstituitonstreamBookingExistingFLow" 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 "InstituitonstreamBookingExistingFLow" flow.

Set Input Values

A_a Book_An_Appointment	<input type="checkbox"/> Not Included
A_a How_may_I_Help_You	<input type="checkbox"/> Not Included

## 7.5 Add Subflow

- In the if new student path, add a sub flow, select InstituitonstreamBooking and label the subflow as the new flow.
- click on save.

InstitutionstreamBooking

\* Label  
new flow

\* API Name ⓘ  
new\_flow

Description

Referenced Flow

**InstitutionstreamBooking**  
InstitutionstreamBooking

Use values from the parent flow to set the inputs for the "InstitutionstreamBooking" 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 "InstitutionstreamBooking" flow.

- save the flow with the name InstitutionstreamBookingFlow.

## TASK-8 Lightning app page creation

### 8.1 creating a lightning app page

- Go to setup, in the quick find box, search for lightning app builder.
- click on new, click on home page , click next.
- give the label as InstitutionstreamBookingHomepage for the lightning app page.
- select the standard home page,then click done.
- Drag the Flow component to the top-right region.
- Search for the InstitutionstreamBookingFlow and click Save.

- Click Activate, Click App and Profile, then click Assign to Apps and Profiles.
- Select the Sales app, then click Next.
- Scroll down the list of profiles and select System Administrator, then click Next.
- Review the assignment, and then click Save.

The screenshots illustrate the Salesforce Lightning App Builder interface:

**Top Screenshot (Home Page):**

- Header:** Setup, Home, Object Manager, Search bar, and various icons.
- Left Sidebar:** User Interface, Lightning App Builder, and a search bar.
- Content Area:**
  - Get Started with Einstein Bots:** AI-powered bot to automate digital connections. Includes a "Get Started" button.
  - Mobile Publisher:** Create your own branded mobile app. Includes a "Learn More" button.
  - Real-time Collaborative Docs:** Transform productivity with collaborative docs, spreadsheets, and slides. Includes a "Get Started" button.
- Bottom Section:** Most Recently Used (10 items) table with columns: NAME, TYPE, and OBJECT.

**Bottom Screenshot (Lightning Pages List View):**

- Header:** Setup, Home, Object Manager, Search bar, and various icons.
- Left Sidebar:** Quick Find, Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Sales Cloud Everywhere, Administration (Users, Data).
- Content Area:**
  - Lightning App Builder:** Provides an easy-to-use graphical interface for creating custom Lightning pages for Salesforce Lightning Experience and mobile app.
  - Lightning Pages List:**

Action	Label	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Edit   Clone   Del	InstitutionstreamBookingHomepage	InstitutionstreamBookingHomepage			Home Page	RData, 26/10/2024, 2:26 pm	RData, 26/10/2024, 2:31 pm

Create a new Lightning page

App Page

Home Page

Record Page

Embedded Service Page

Voice Extension

Customize the Lightning Experience Home page.

My Quarterly Performance

Last refresh Today at 7:33am

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

AUG SEPT OCT

Einstein (10+)

- 7 Contact & Contacts
- Based on email activity
- Created 7 contact and contact roles for this account
- View

8 Contacts & Contact

122 Contact-Roles Created

Opportunity Boosting

Next

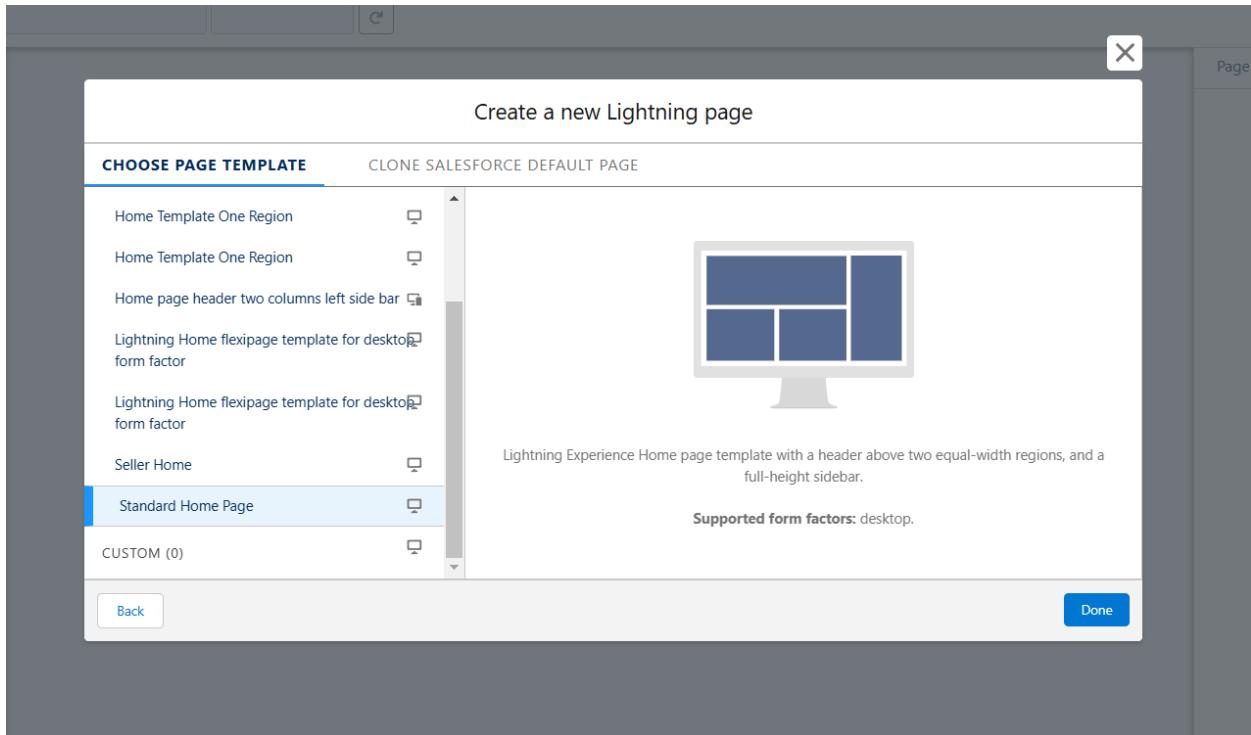
Create a new Lightning page

\*Label

InstitutionstreamBookingHomepage

Back

Next



The screenshot shows the Lightning App Builder interface. At the top, the navigation bar includes 'Lightning App Builder', 'Pages', and 'InstitutionstreamBookingHomepage'. The main area is titled 'Desktop' and shows a canvas with two columns. A placeholder component 'Flow Component: InstitutionstreamBookingFlow' is placed in the first column. The second column has a placeholder 'Add Component(s) Here'. To the left of the canvas is a 'Components' sidebar with a search bar and a list of components under 'Standard (40)', including Accordion, App Launcher, Assistant, Chatter Feed, Chatter Publisher, CRM Analytics Collection, CRM Analytics Dashboard, Dashboard, Data Mask Console Home Component, Einstein Next Best Action, Flow, Flow App Home cards, Generate Batch Documents, Inventory Lookup Component, Items to Approve, and Key Deals. To the right of the canvas is a 'Page' configuration panel with fields for 'Label' (set to 'InstitutionstreamBookingHomepage'), 'API Name' (set to 'InstitutionstreamBookingHomepage'), 'Page Type' (set to 'Home Page'), 'Template' (set to 'Standard Home Page'), and a 'Description' field.

