

Project Report On

Airlines Management System – Developer

Milestone - 01: Creating Salesforce Org

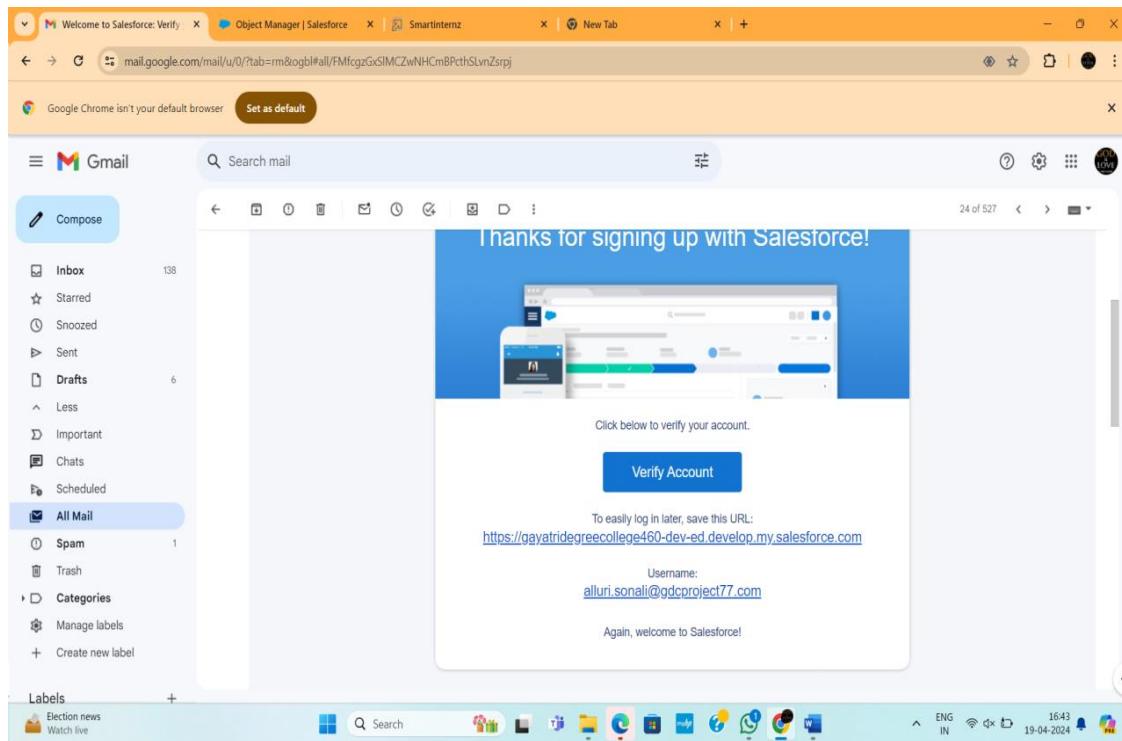
Go to <https://developer.salesforce.com/signup>

On the sign up form, enter the following details :

1. **First name & Last name** –ALLURI SONALI
2. **Email** – alluri.sonali@gmail.com
3. **Role : Developer**
4. **Company** : Gayatri Degree College
5. **Country** : India
6. **Postal Code** : 517502
7. **Username** :alluri.sonali@gdcproject77.com

Account Activation

Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.

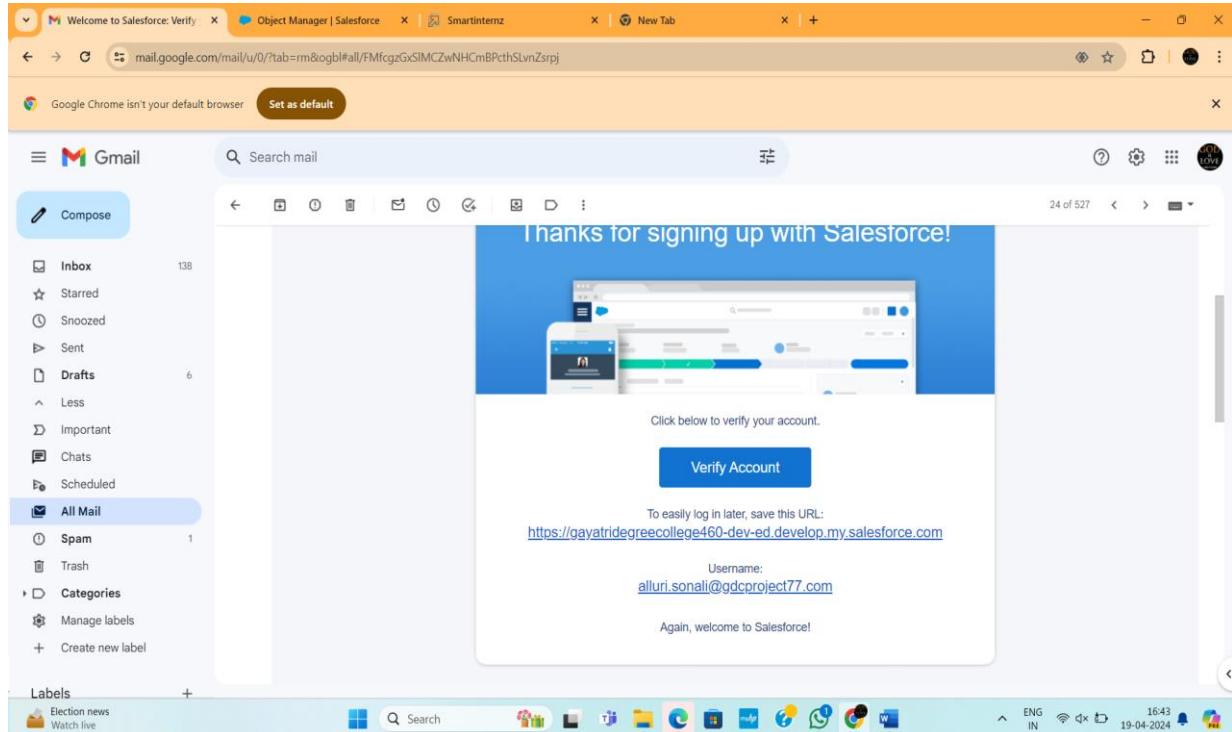


Click on Verify Account

Give a password and answer a security question and click on change password.

Login To Salesforce Account

- 1.Go to salesforce.com and click on login.
- 2.Enter the username and password that you just created.
- 3.After login this is the home page which you will see.



Milestone – 02 :object creation

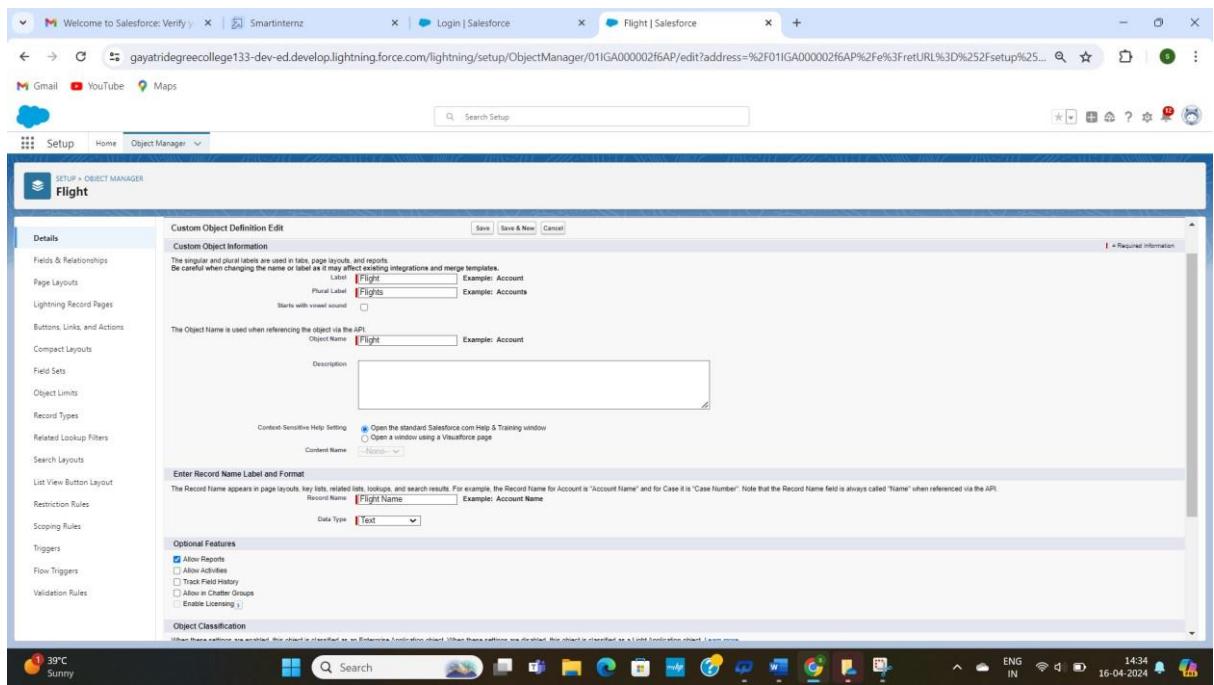
Create Flight Object

The purpose of creating a Flight custom object is to have a clear picture of the flight details.

To create an object:

From the setup page --> Click on Object Manager -->Click on Create --> Click on Custom Object.

1. **Enter the label name-->** Flight
2. **Plural label name-->** Flights
3. **Enter Record Name Label and Format**
 - **Record Name :** Flight Name
 - **Data Type :** Text
- 4.Click on Allow reports,
- 5.Allow search --> Save.



Create Booking Object

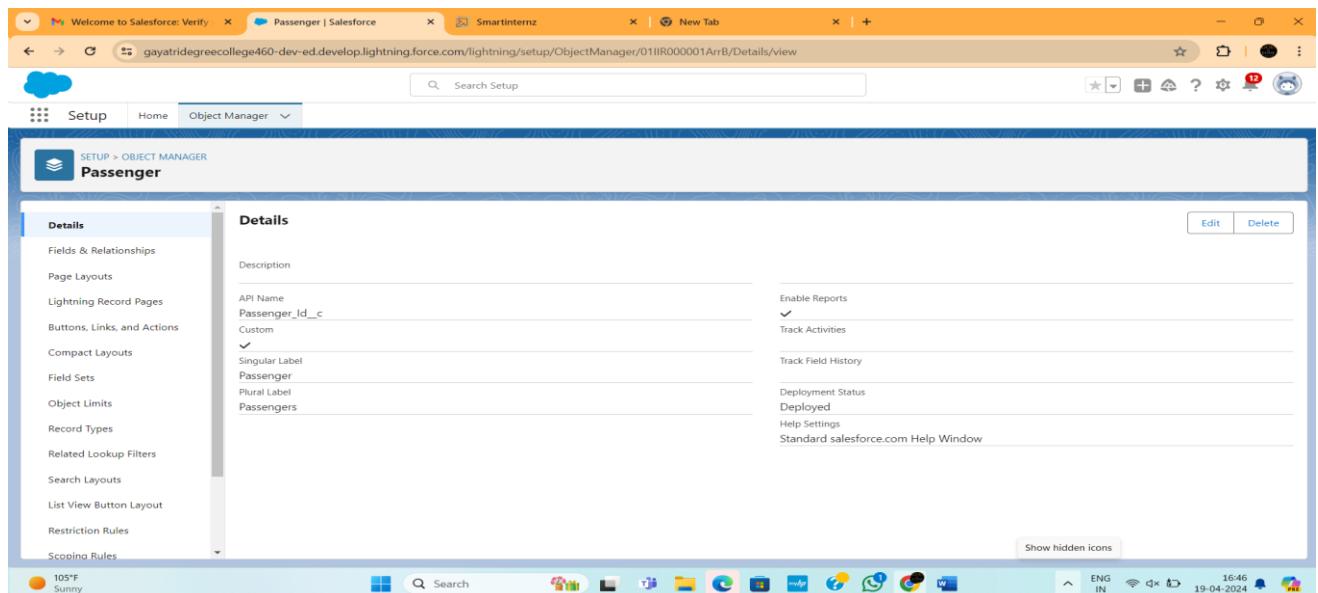
From the setup page --> Click on Object Manager --> Click on Create --> Click on Custom Object.

1. Enter the label name--> Booking
2. Plural label name--> Bookings
3. Enter Record Name Label and Format
 - Record Name : Booking Id
 - Data Type : Auto Number
 - Display Format : Bk-{0000}
 - Starting Number : 1
- 4.Click on Allow reports,
- 5.Allow search --> Save

Object Manager					
1 Items, Sorted by Label					
LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Booking	Booking_c	Custom Object		15/04/2024	✓

Create passenger and Crew

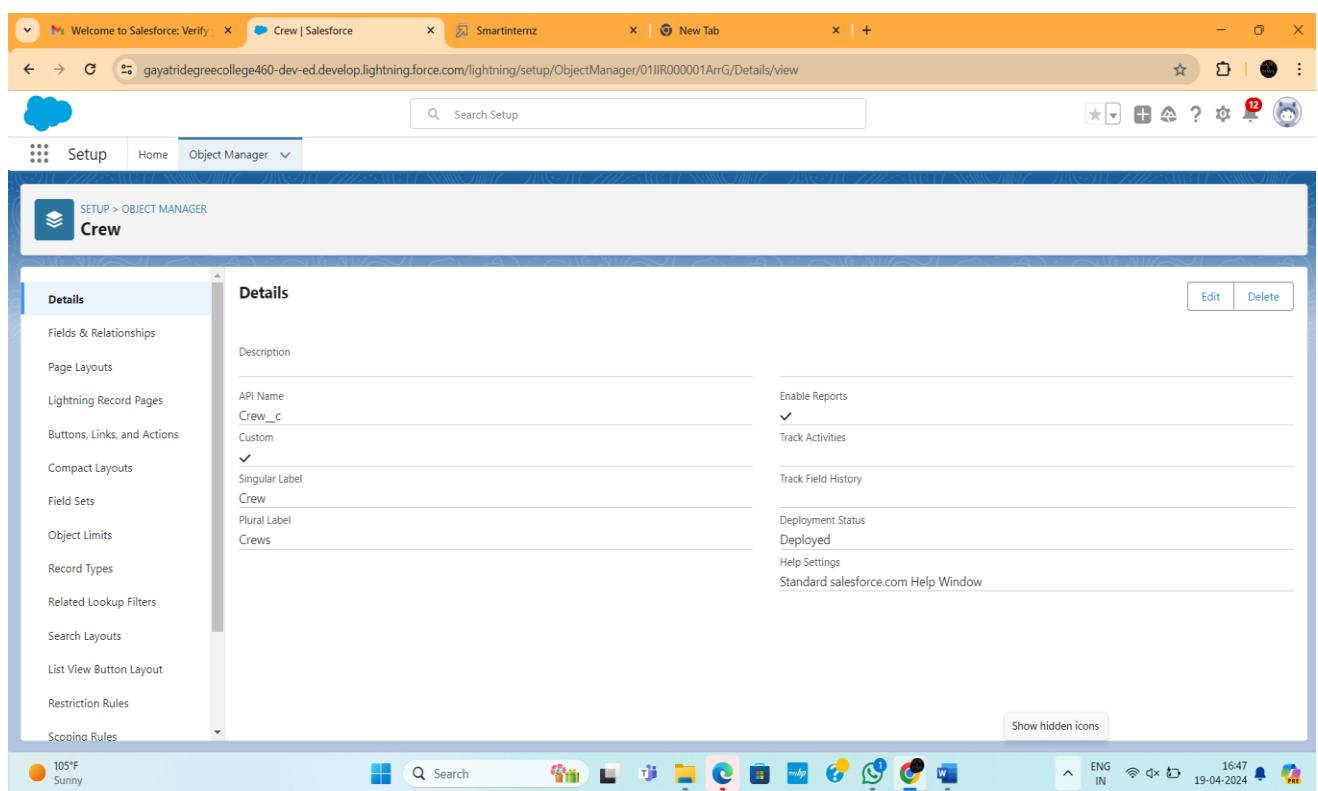
Create 2 more objects with record names as Passenger Id and Crew Id



The screenshot shows the Salesforce Object Manager interface for the 'Passenger' object. The left sidebar lists various configuration options: Fields & Relationships, 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, Restriction Rules, and Scoping Rules. The main 'Details' tab displays the following information:

Setting	Value
Description	
API Name	Passenger_Id_c
Custom	✓
Singular Label	Passenger
Plural Label	Passengers
Enable Reports	✓
Track Activities	
Track Field History	
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

The status bar at the bottom indicates it's 105°F and sunny.



The screenshot shows the Salesforce Object Manager interface for the 'Crew' object. The left sidebar lists the same configuration options as the Passenger object. The main 'Details' tab displays the following information:

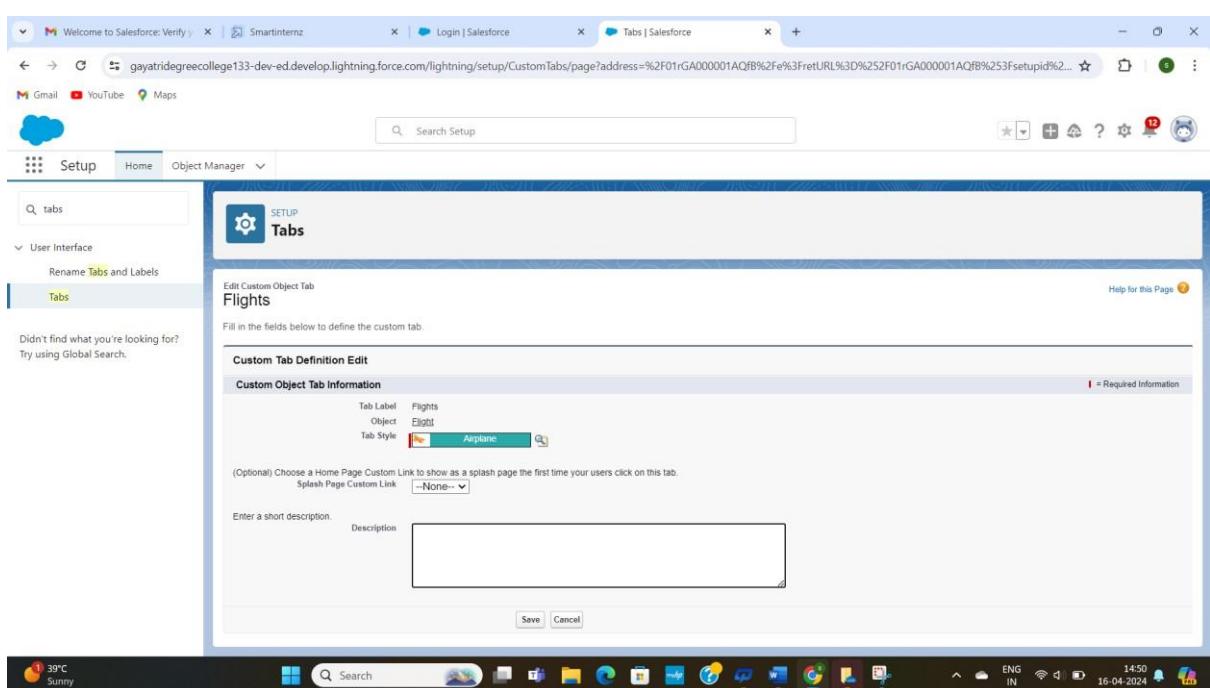
Setting	Value
Description	
API Name	Crew_c
Custom	✓
Singular Label	Crew
Plural Label	Crews
Enable Reports	✓
Track Activities	
Track Field History	
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

The status bar at the bottom indicates it's 105°F and sunny.

Milestone - 03 : Tabs

Creating A Custom Tab(Flight)

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

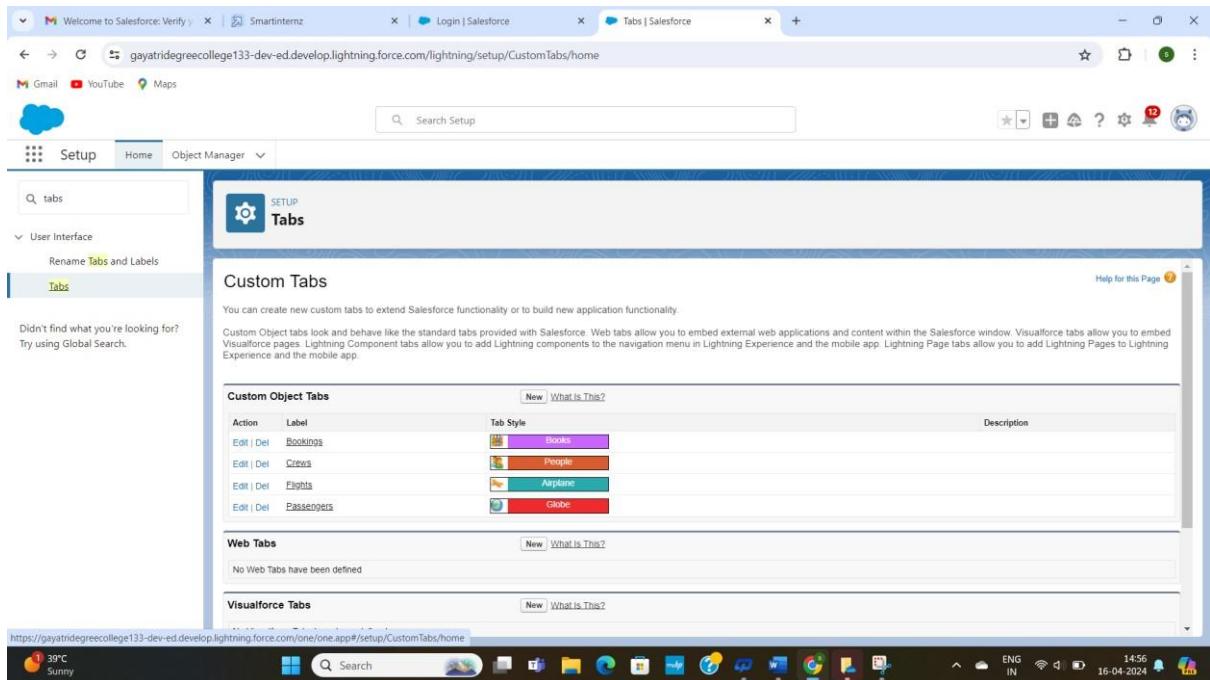


Creating A Custom Tab(Booking)

1. Go to setup page --> type Tabs in Quick Find bar --> click on tabs --> New (under custom object tab)
2. Select Object(Booking) Select the tab style --> Next (Add to profiles page) keep it as default --> Next (Add to Custom App) keep it as default --> Save.

Creating Custom Tabs(Crew and Passengers)

Now create tabs for Crew and Passengers objects.



Milestone-04: The Lightning App

Create A Lightning App

Go to setup page --> search “app manager” in quick find -->select “app manager” -->click on New lightning App.

1. Fill the app name in app details and branding as follow

App Name : AirLine Management System

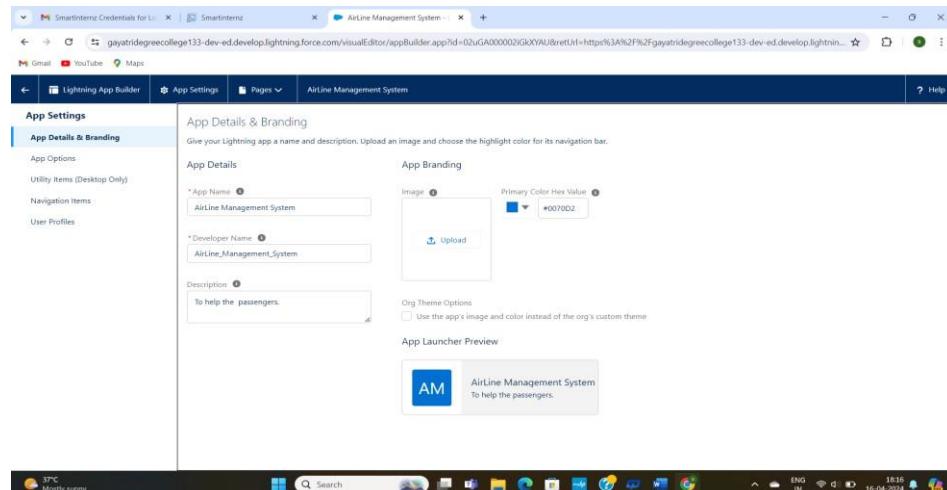
Developer Name : this will auto populated

Description : Give a meaningful description

Image : optional (if you want to give any image you can otherwise not mandatory)

Primary color hex value : keep this default

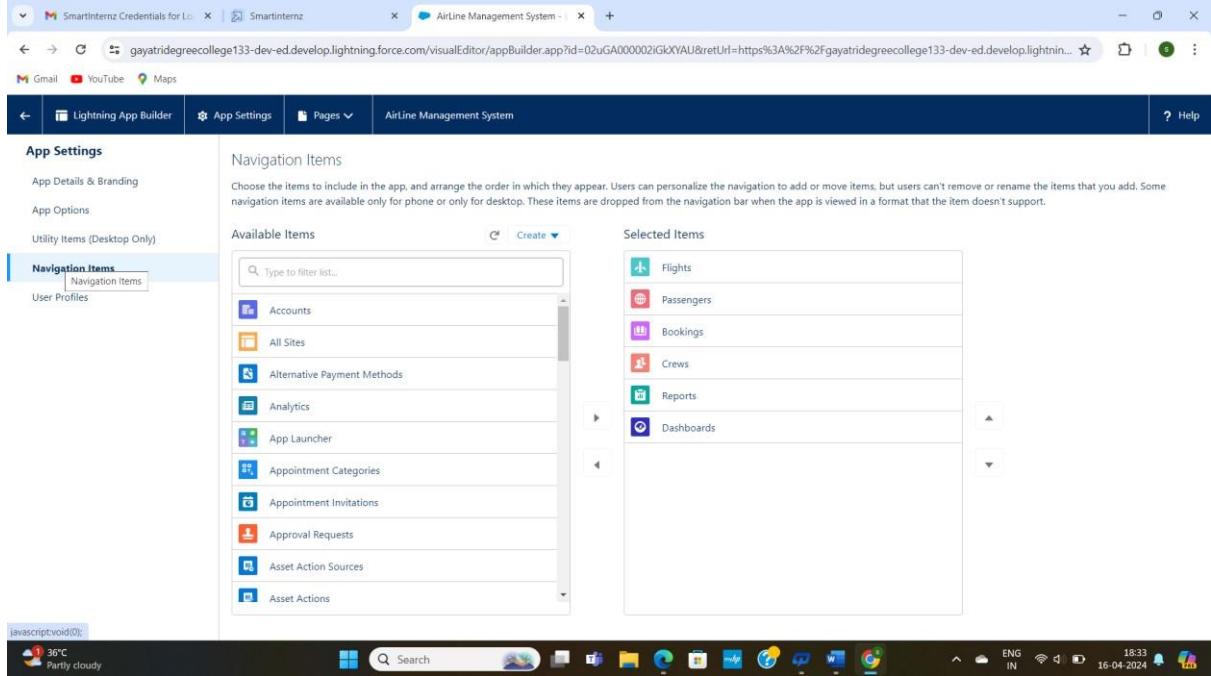
2. Then click Next --> (App option page) keep it as default --> Next --> (Utility Items) keep it as keep it as default --> Next.



3.To Add Navigation Items:

Search the items in the search bar(Flight, Passengers, Booking, Crew, Reports, Dashboard) from the search bar and move it using the arrow button --> Next.

Note: select the custom object which we have created in the previous activity.



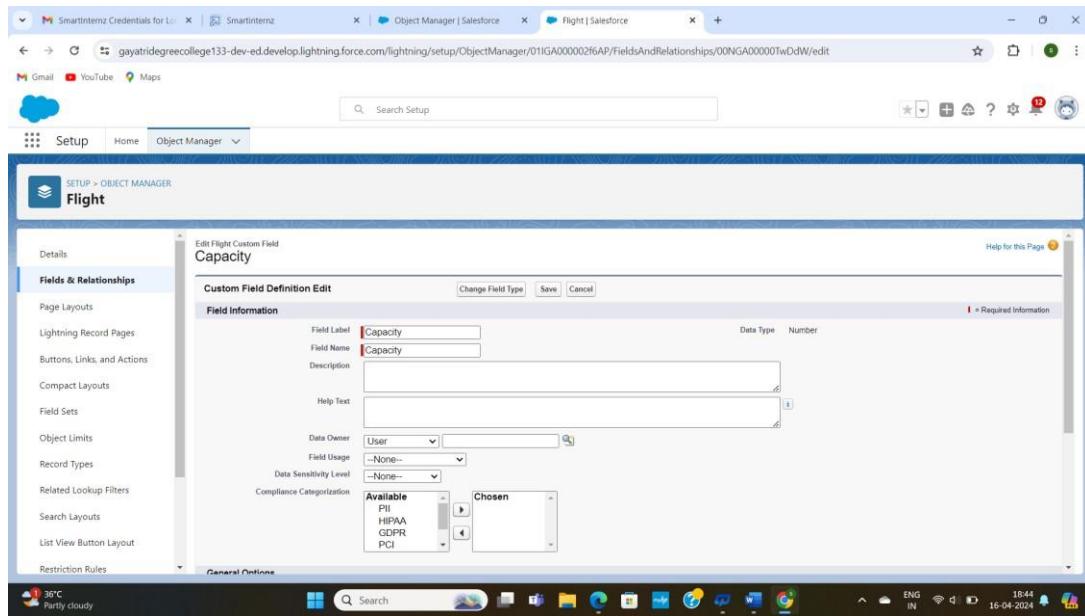
4.To Add User Profiles:

Search profiles (System administrator) in the search bar --> click on the arrow button --> save & finish.

Milestone 05: Fields

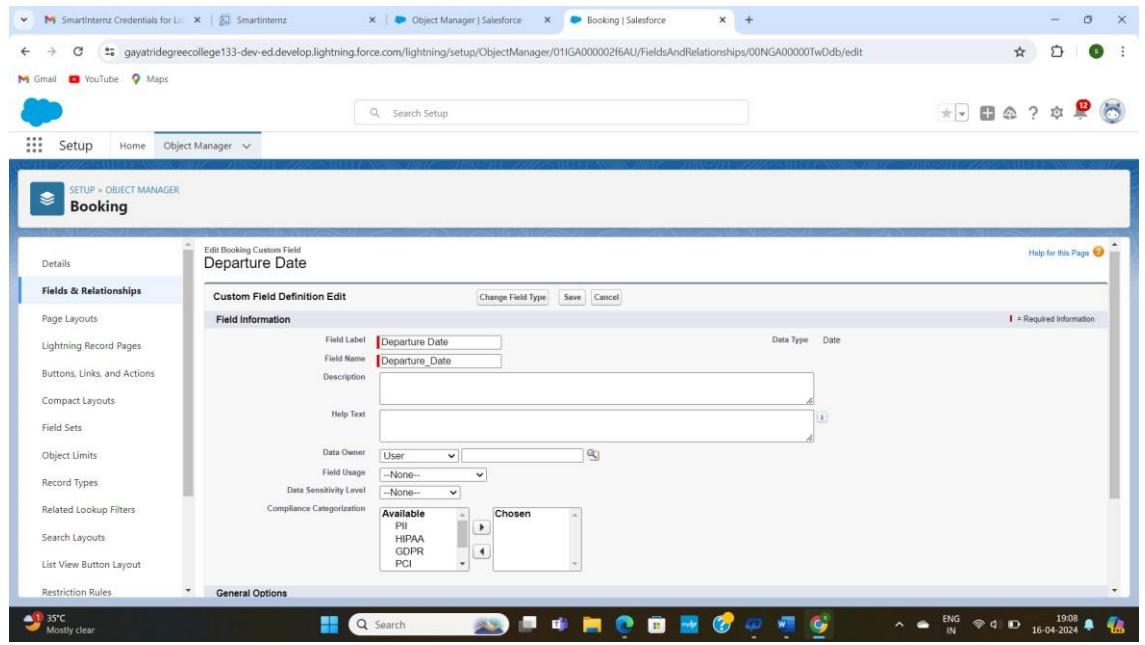
Creating Field In Flight Object

1. Go to setup --> click on Object Manager --> type object name(Flight) in quick find bar--> click on the object.
2. Now click on “Fields & Relationships” --> New
3. Select Data type as “Number”.
4. Click on Next
5. Fill the above as following:
 - **Field Label: Capacity**
 - **Length : 18**
 - **Field Name :** gets auto generated
 - Click on Next ? Next --> Save and new



Creating Departure Date On Booking Object

1. Go to setup ? click on Object Manager ? type object name(Booking) in quick find bar? click on the object.
2. Now click on “Fields & Relationships” ? New.
3. Select Data type as “Date” and click Next.
4. Click on Next.
5. Fill the above as following:
 - o **Field Label:**Departure Date.
 - o **Field Name :** gets auto generated.
 - o **Click on Next --> Next --> Save and new**



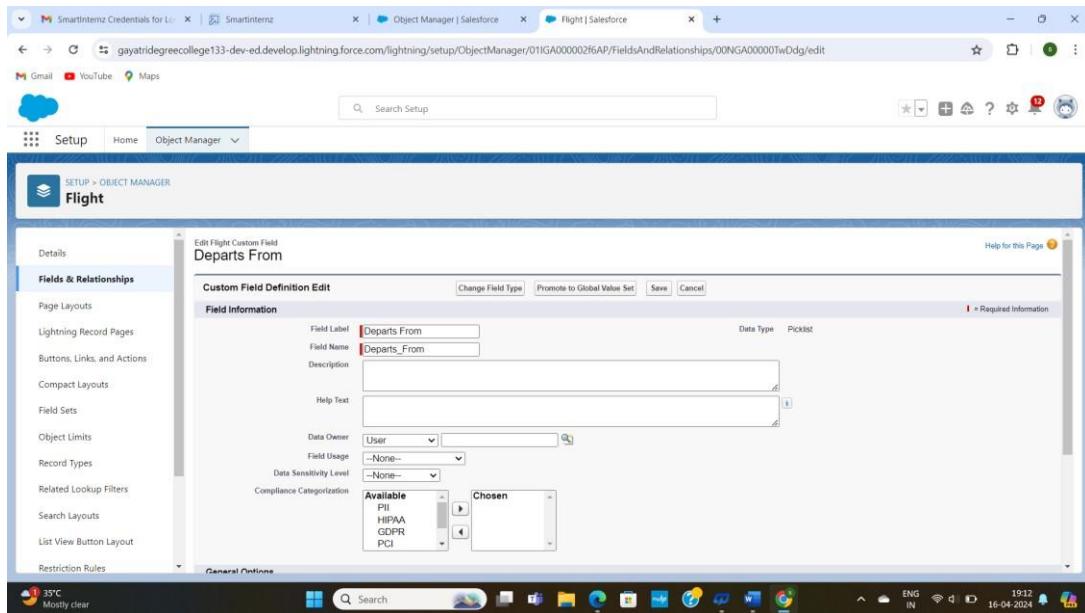
W.

Creating Picklist Field On Flight Object.

1. Go to setup --> click on Object Manager --> type object name(Flight) in quick find bar--> click on the object.
2. Now click on “Fields & Relationships” -->New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Departs From”, under values select “Enter values, with each value separated by a new line” and enter values as shown below:

Delhi
Mumbai
Bengaluru
Chennai
Kolkata

5.Click Next--> Next --> Next --> Save & New.



Creating Lookup Relationship Between Flight And Booking

Creating Lookup Relationship between Flight & Booking Object

To Create a lookup relationship

1. Go to the setup page --> click on object manager -->type object name(Booking) in the quick find bar--> click on the object.
2. Click on fields & relationship --> click on New.
3. Select “lookup relationship” as data type and click Next.
4. For field label related to: select “Flight” object and click Next.
5. Give Field Label as “Flight” and click Next.
6. Next --> Next --> Save.

Creating Remaining Fields

Now create the remaining fields using the data types mentioned in the table.

SI No	Object	Field	Datatype

1	Passenger	1. 2. Name 3. 4. Phone 5. 6. Passport Number 7. Passport Issue Date Passport Expiry Date Date of Birth Nationality	Text Phone Number Date Date Date Picklist : Values as below
			India China USA Canada Mexico Australia South Korea North Korea Germany

The screenshot shows the Salesforce Object Manager interface for the 'Passenger' object. The left sidebar lists various setup options: Details, Fields & Relationships (selected), 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, and Restriction Rules. The main content area displays the 'Fields & Relationships' section, which lists 11 items sorted by field label. The fields include:

Field Label	Type	Description
Last Modified By	Lookup(User)	LastModifiedById
Name	Text(50)	Name_c
Nationality	Picklist	Nationality_c
Owner	Lookup(User/Group)	OwnerId
Passenger Id	Auto Number	Name
Passport Expiry Date	Date	Passport_Expiry_Date_c
Passport Issue Date	Date	Passport_Issue_Date_c
Passport Number	Number(18, 0)	Passport_Number_c
Phone	Phone	Phone_c

SI No	Object	Field	Datatype

2	Booking	Passenger Name Departs From Departs To Travellers	Lookup relationship with Passenger Picklist : Values are same as Activity 3 Picklist : Values are same as Activity 3 Picklist : Values as below																
		Class	<table border="1"> <tr> <td>Adult</td> <td></td> </tr> <tr> <td>Child</td> <td></td> </tr> <tr> <td>Infant</td> <td></td> </tr> <tr> <td colspan="2">Picklist : Values as below</td> </tr> <tr> <td>Economy</td> <td></td> </tr> <tr> <td>Business</td> <td></td> </tr> <tr> <td>First</td> <td></td> </tr> <tr> <td>Class</td> <td></td> </tr> </table>	Adult		Child		Infant		Picklist : Values as below		Economy		Business		First		Class	
Adult																			
Child																			
Infant																			
Picklist : Values as below																			
Economy																			
Business																			
First																			
Class																			

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

- Page Header:** Smartinternz Credentials for L... | Smartinternz | Object Manager | Salesforce | Booking | Salesforce
- Search Bar:** Search Setup
- Navigation:** Setup | Home | Object Manager
- Section:** SETUP > OBJECT MANAGER | Booking
- Fields & Relationships:**
 - Fields & Relationships:** 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, Restriction Rules.
 - Table:** Displays 11 items, Sorted by Field Label.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Booking Id	Name	Auto Number		✓
Class	Class_c	Picklist		✓
Created By	CreatedBy	Lookup(User)		
Departs From	Departs_From__c	Picklist		✓
Departs To	Departs_To__c	Picklist	Departs From	✓
Departure Date	Departure_Date__c	Date		✓
Flight	Flight_c	Lookup(Flight)		✓
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
- System Status:** 3PC Mostly clear
- System Icons:** Search, Home, Recent, Help, Notifications, User Profile
- System Date/Time:** 16-04-2024 19:24

Sl No	Object	Field	Datatype
3	Flight	1. Departs to 2. PNR Number 3. Departure Date 4. Departure Time 5. Arrival Time	Picklist : Values are same as Activity 3 Number Date Time Time

SETUP > OBJECT MANAGER

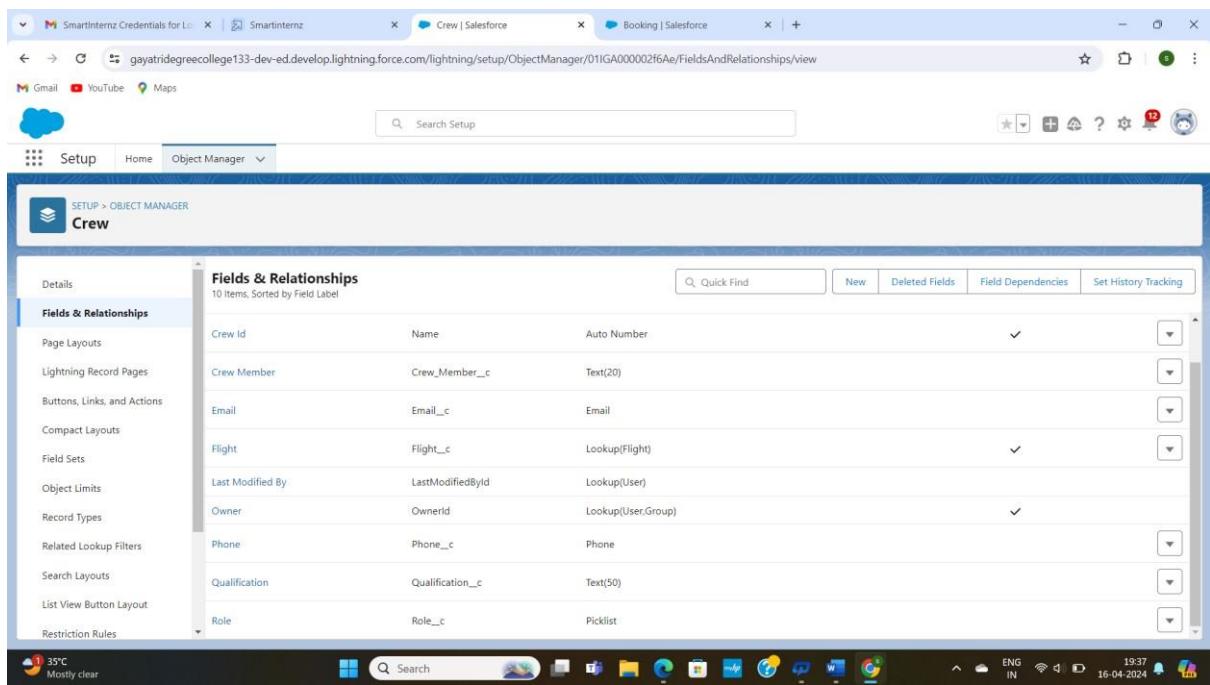
Flight

Fields & Relationships
12 items, Sorted by Field Label

Crew Member	Crew_Member__c	Text(20)			
Departs From	Departs_From__c	Picklist			
Departs to	Departs_to__c	Picklist	Departs From		
Departure Date	Departure_Date__c	Date			
Departure Time	Departure_Time__c	Time			
Flight Name	Name	Text(80)			
Last Modified By	LastModifiedBy	Lookup(User)			
Owner	OwnerId	Lookup(User,Group)			
PNR Number	PNR_Number__c	Number(18, 0)			

Sl No	Object	Field	Datatype
4	Crew	1. Flight 2. Crew Member 3. Phone 4. Email 5. Qualification 6. Role	Lookup relationship with Flight Text(20) Phone Email Text Picklist: Values as below

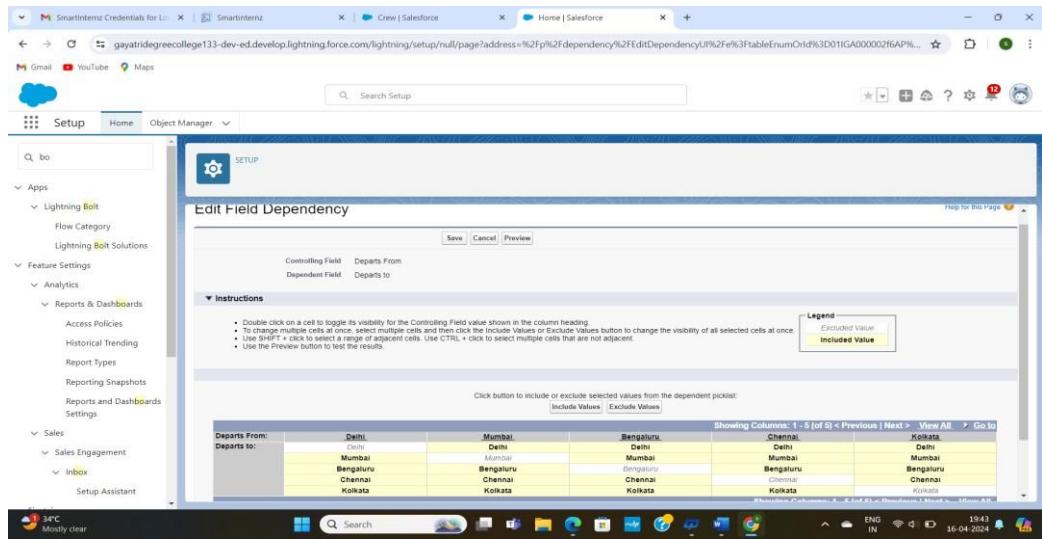
Pilot
 Co-Pilot
 Flight Attendants
 Air Hosts
 Air Hostesses
 Air Stewards
 Air Stewardesses
 On-Board Chefs



Creating Field Dependencies

To create a field dependency:

1. Go to object manager -->select the object (flight) --> click fields & relationships --> click "Field Dependencies".
2. Click "New" under Field Dependencies.
3. Select "Departs From" for Controlling Field.
4. Select "Departs to" for Dependent Field and click Continue.
5. While pressing CTRL button select the values as shown below and click on include values. 6. Click on save



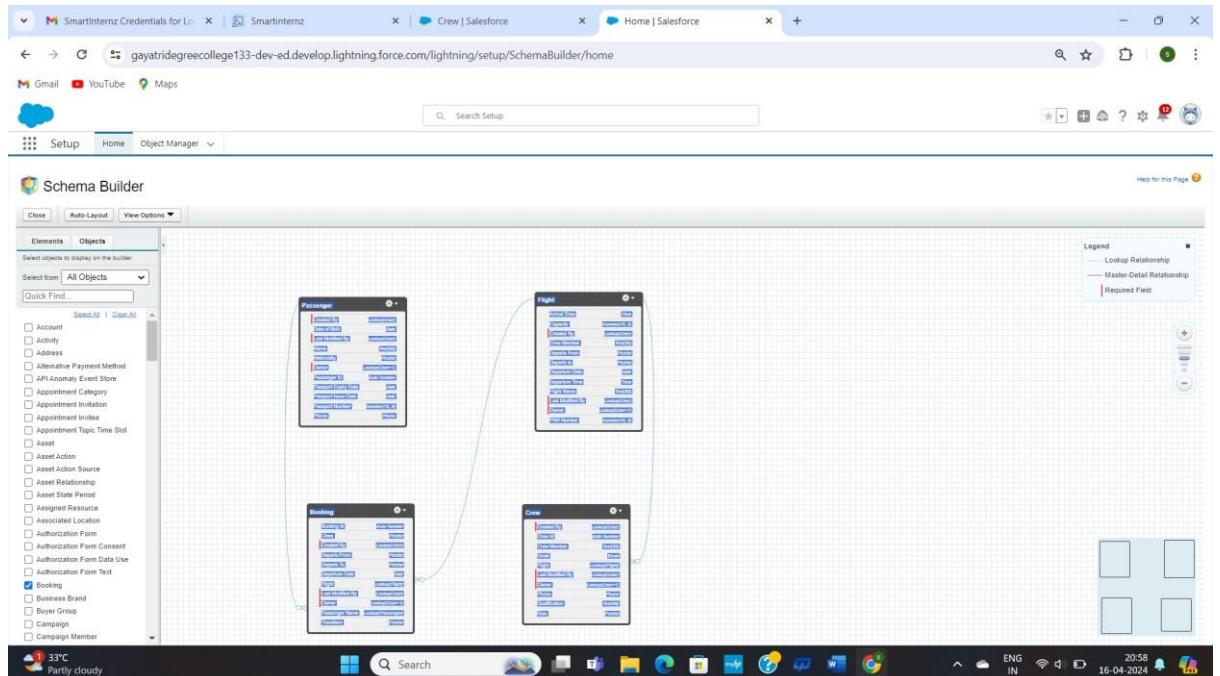
Create similar Field dependencies for booking object as we have done in activity 6.

Schema Builder

Schema Builder is a powerful tool within Salesforce that allows you to visualise, explore, and design the relationships between objects in your Salesforce organisation. It provides a graphical representation of the data model, making it easier to understand the structure and connections between different objects.

Creating Schema Builder

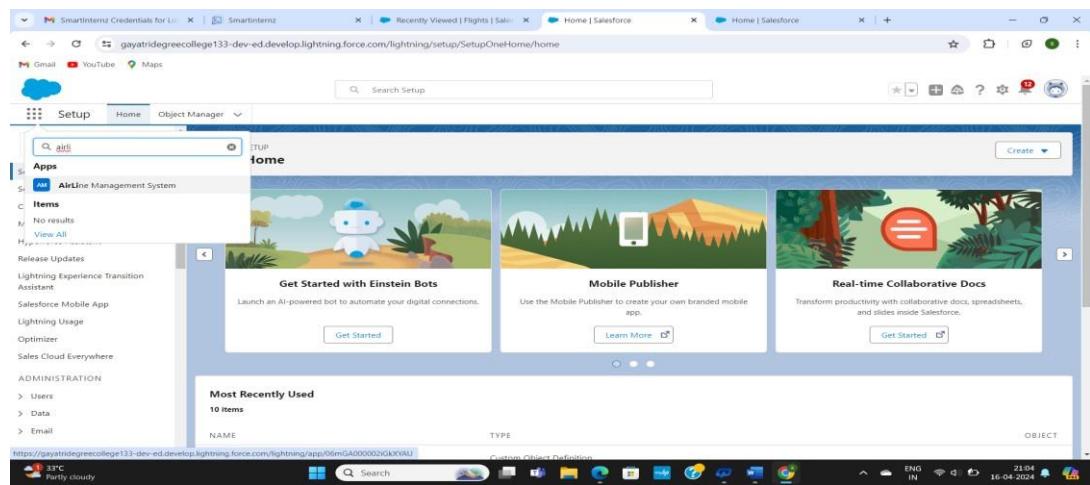
1. Go to setup --> click on Object Manager --> Schema Builder.
2. Click on Clear All
3. Select objects--> Enter Objects which we have created in the above Milestone 2 (Passenger, Booking, Flight & Crew) in a quick box and select them.



Milestone 06: User Adoption

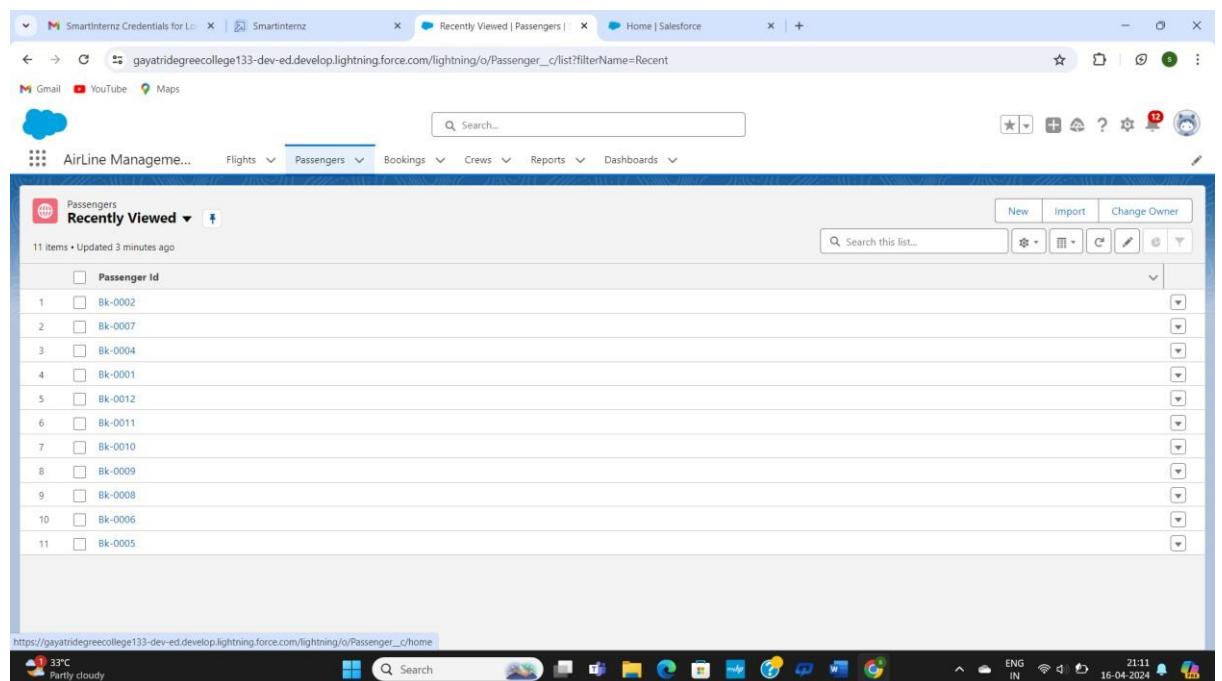
Create A Record (Passenger)

1. Click on App Launcher on the left side of the screen.
2. Search AirLine Management System & click on it.
3. Click on the Passenger tab.
4. Click New.



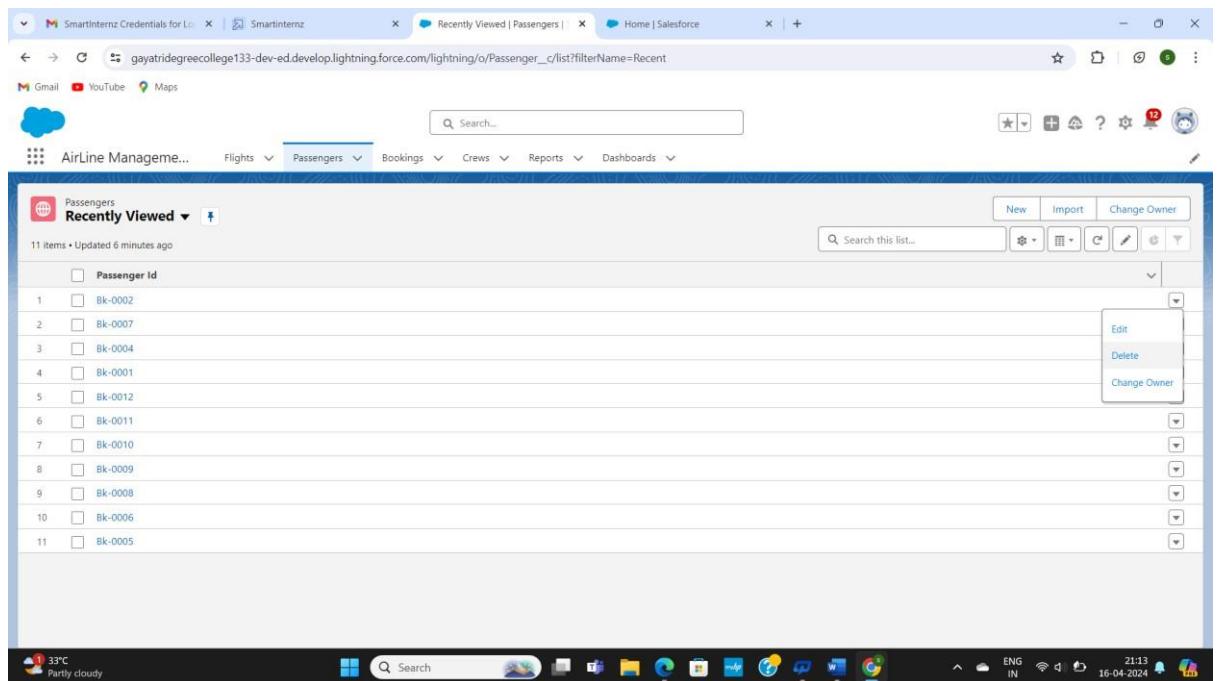
View A Record(Passenger)

1. Click on App Launcher on the left side of the screen.
2. Search AirLine Management System & click on it.
3. Click on the Passenger Tab.
4. Click on any record name. you can see the details of the Passenger.



Delete A Record(Passenger)

1. Click on App Launcher on the left side of the screen.
2. Search AirLine Management System & click on it.
3. Click on the Passenger Tab.
4. Click on Arrow at right hand side on that Particular record.
5. Click delete.



Create at least 10 records for each of the objects: Passenger, Booking, Flight, Crew.

Milestone-07: Profiles

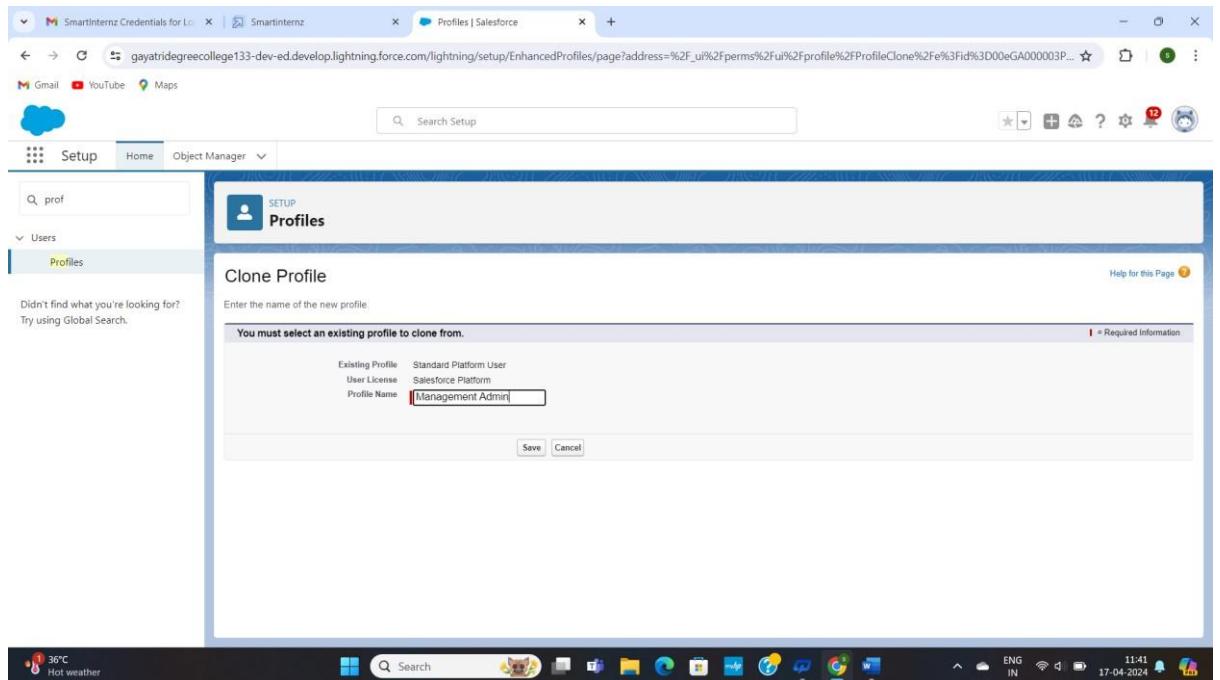
General Admin Profile

To create a new profile:

1. Go to setup --> type profiles in quick find box --> click on profiles --> clone the desired profile (Standard Platform User)-->enter profile name (General Admin)--> Save.
2. While still on the profile page, then click Edit.
3. Scroll down to Custom Object Permissions and Give access permissions for read, create and edit on Passenger, and booking objects.
4. Scroll down and Click on Save.

Management Admin Profile

1. Go to setup --> type profiles in quick find box --> click on profiles --> clone the desired profile (Standard Platform User) --> enter profile name (Management Admin) -->Save.
2. While still on the profile page, then click Edit.
3. Scroll down to Custom Object Permissions and Give access permissions for read, edit and create Flight and Crew.
4. Scroll down and Click on Save.



Create Senior Admin & Crew Member Profiles

Create Crew Member Profile for Crew Members and give them access permission to read only as we have done in the previous activity.

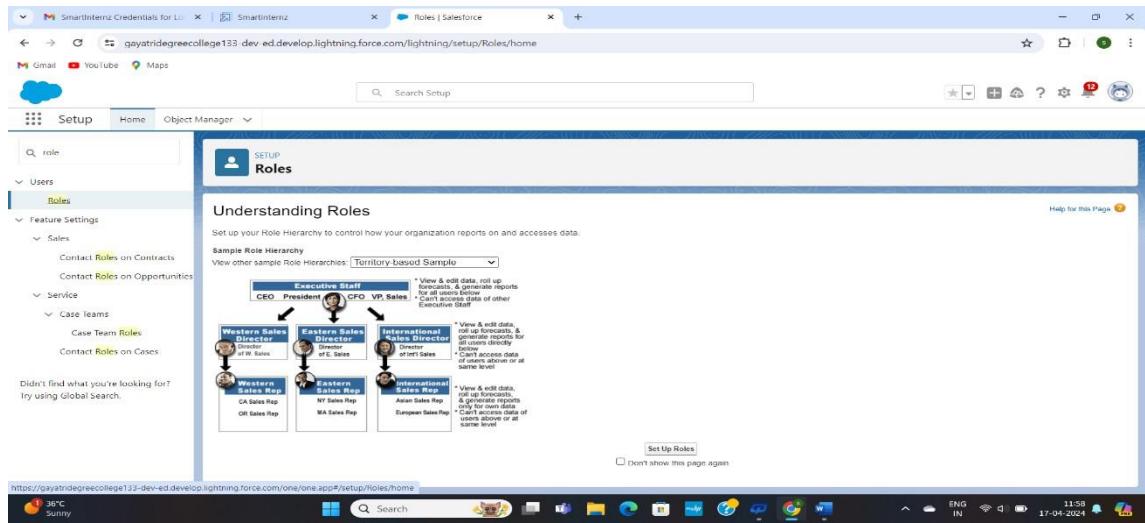
Create Senior Admin Profiles for Senior Admin in the Organization and give him access permission to read, create, edit and delete for all 4 objects.

Note: for Senior Admin clone System Administrator profile.

Milestone-08 : Role

Creating Admin Manager Role

Go to quick find --> Search for Roles --> click on set up roles.



1. Click on Expand All and click on add role under whom this role works.
2. Give Label as "Senior Admin" and Role name gets auto populated. Check to whom this role (Senior Admin) reports. Then click on Save.
3. Create three more roles for General Admin, Management Admin and Crew Member. Note: Crew Members report to Management Admin, Management Admin & General Admin Reports to Senior Admin and only Senior Admin reports to CEO.

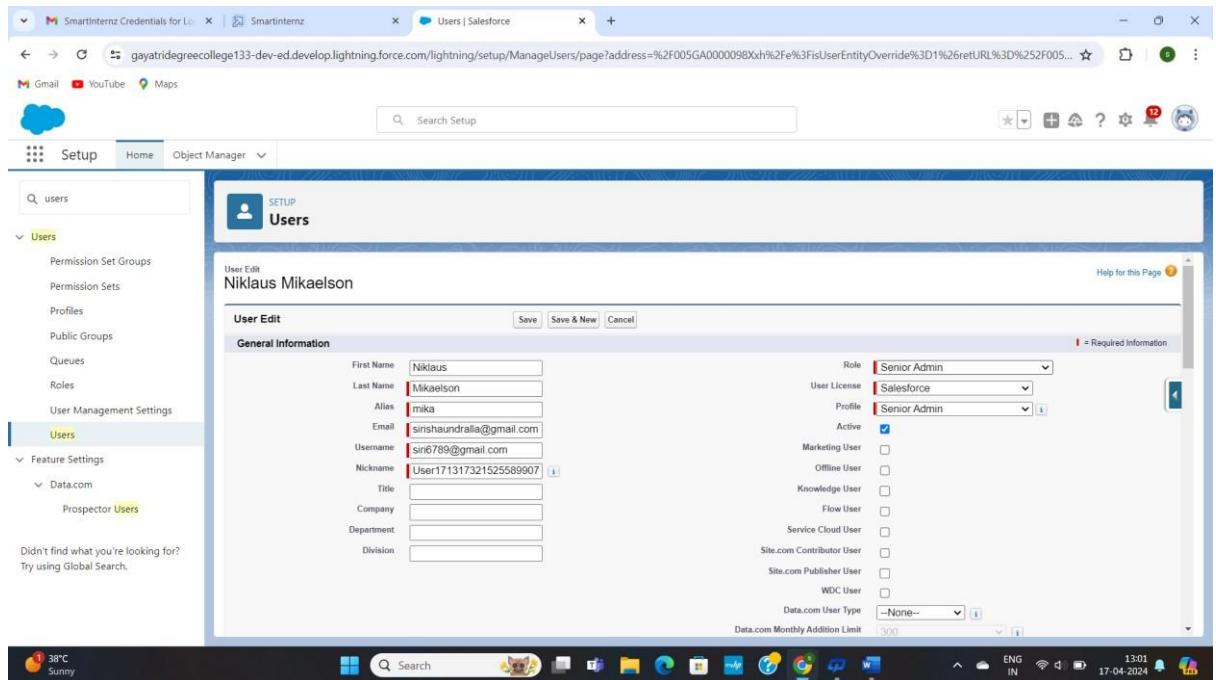
Milestone 9 : Users

Create User

Go to setup --> type users in quick find box --> select users --> click New user.

Fill in the fields

1. **First Name :** Niklaus
2. **Last Name :** Mikaelson
3. **Alias :** Give a Alias Name
4. **Email id :** Give your Personal Email id
5. **Username :** Username should be in this form: text@text.text
6. **Nick Name :** Give a Nickname
7. **Role :** Senior Admin
8. **User licence :** Salesforce
9. **Profiles :** Senior Admin



Create user for General Admin

1. Go to setup --> type users in quick find box --> select users --> click New user.

Fill in the fields

- First Name : Kol
- Last Name : Mikaelson
- Alias : Give a Alias Name
- Email id : Give your Personal Email id
- Username : Username should be in this form: text@text.text
- Nick Name : Give a Nickname
- Role : General Admin
- User license : Standard Platform
- Profiles : General Admin

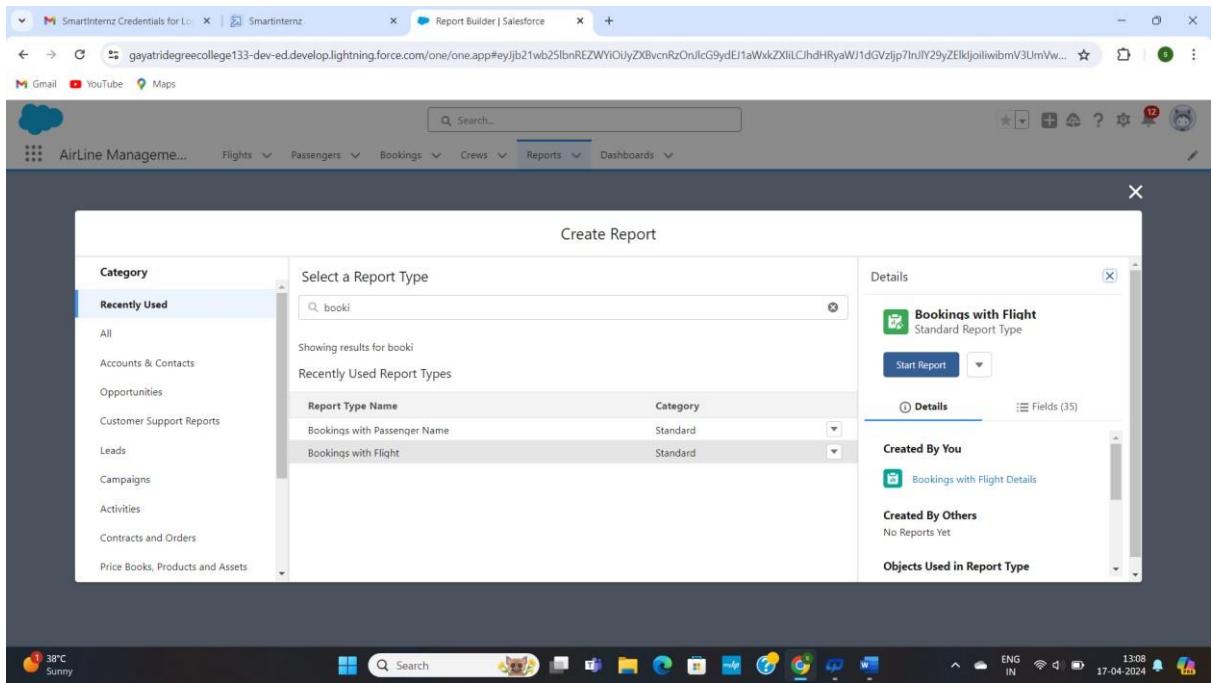
3.Save

Create two more users for Management Admin and Crew Members as mentioned in activity 2.

Milestone 10 : Reports

Create Report

1. Go to the app --> click on the reports tab
2. Click New Report.
3. Select report type from category or from report type panel or from search panel --> click on start report.



4. Customize your report

- Check the toggle on the update preview automatically button.
 - Add fields from the left pane as shown below.
 - Add the “Traveller” field in Group Rows.
 - Turn the toggle off for the subtotal button.
- 5.Click on save.
6.Save the report as “Bookings with Flight Details”.
7.Click on Save.

Note: Reports may get varied from the above pictures as the data might be different.

1. Create a report with a report type: “Booking with Passenger Name”.
2. Create a report with report type: “Crew with Flight”

Milestone 11 : Dashboards

1. Create Dashboard
2. Go to the app --> click on the Dashboards tabs.
3. Give a Name and click on Create.
4. Select add component.
5. Select a Report and click on select.

Create two more Dashboards using the reports which we have created in the milestone 10 activity 2.

Report Name	Description	Folder	Created By	Created On	Subscribed
Bookings with Flight Details	Private Reports	Alluri sonali	13/4/2024, 4:20 pm		
Booking with Passenger Name	Private Reports	Alluri sonali	13/4/2024, 4:25 pm		
Crew with Flight	Private Reports	Alluri sonali	13/4/2024, 4:29 pm		

Milestone 12: Apex

Create An Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
4. Give the Apex Class name as “PhnValid_PassengerObj”.
5. Click ok.
6. Now write the code logic here

Source Snippet:

```
public class PhnValid_PassengerObj {
    public static void valMethod(List<Passenger__c> newPass){
        for(Passenger__c p:newPass){ if(p.Phone__c==null){
            p.Phone__caddError('please Enter phone Number');
        }
    }
}
```

7.Save the code.(click on file --> Save)

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [gayatriidgrecollge133-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage](http://gayatriidgrecolllege133-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The tab bar has three tabs: PhnValid_PassengerObj.apxc (highlighted in orange), PhnValidTrigger.apxt, and PhnValid_TestClass.apxc. The main area displays the Apex code for PhnValid_PassengerObj:

```
1 public class PhnValid_PassengerObj {  
2  
3     public static void valMethod(List<Passenger__c> newPass){  
4  
5         for(Passenger__c p:newPass){  
6  
7             if(p.Phone__c==null){  
8  
9                 p.Phone__c.addError('please Enter phone Number');  
10            }  
11        }  
12    }  
13 }  
14  
15 }  
16  
17 }
```

Below the code editor is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is selected. The status bar at the bottom shows the date and time as 17-04-2024 14:07.

Create An Apex Trigger

1. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
Give the Apex Trigger name as “PhnValidTrigger”, and select “Passenger__c” from the dropdown for sObject.
2. Give the Apex Trigger name as “PhnValidTrigger”, and select “Passenger__c” from the dropdown for sObject.
3. Click Submit.
4. Now write the code logic here

Source Code:

```
trigger PhnValidTrigger on passenger__c (before insert) { if(trigger.isBefore  
&& trigger.isInsert){  
    PhnValid_PassengerObj.valMethod(trigger.new); }  
}
```

5. Save the code.(click on file --> Save)

```

trigger PhnValidTrigger on passenger__c (before insert) {
    if(trigger.isBefore && trigger.isInsert){
        PhnValid_PassengerObj.valMethod(trigger.new);
    }
}

@isTest
public class PhnValid_TestClass {
    @isTest
    public static void testClass(){
        list<Passenger__c> varlis = new list<Passenger__c>();
        Passenger__c var = new Passenger__c();
        var.Phone__c = null;
        varlis.add(var);
        insert varlis;
        PhnValid_PassengerObj.valMethod(varlis);
    }
}

```

Create An Apex Test Class

1. To create a new Apex Class follow the below steps: Click on the file --> New --> Apex Class.
2. Give the Apex Class name as “PhnValid_TestClass”.
3. Click ok.
4. Now write the code logic here
5. Source Code:

```

@isTest
public class PhnValid_TestClass {
    @isTest
    public static void testClass(){
        list<Passenger__c> varlis = new list<Passenger__c>();
        Passenger__c var = new Passenger__c();
        var.Phone__c = null;
        varlis.add(var);
        insert varlis;
        PhnValid_PassengerObj.valMethod(varlis);
    }
}

```

6. Save the code.(click on file --> Save).
7. Click on “Run Test” and then click on Test under the terminal section and do check that your overall code coverage should be more than 75%.

```

1 @isTest
2 public class PhnValid_TestClass {
3
4     @isTest
5
6     public static void testClass(){
7         list<Passenger__c> varlis = new list<Passenger__c>();
8         Passenger__c var = new Passenger__c();
9         var.Phone__c = null;
10        varlis.add(var);
11        insert varlis;
12        PhnValid_PassengerObj.valMethod(varlis);
13    }
14 }
15

```

Overall Code Coverage	
Class	Percent
Overall	100%
PhnValidTrigger	100%
PhnValid_PassengerObj	100%

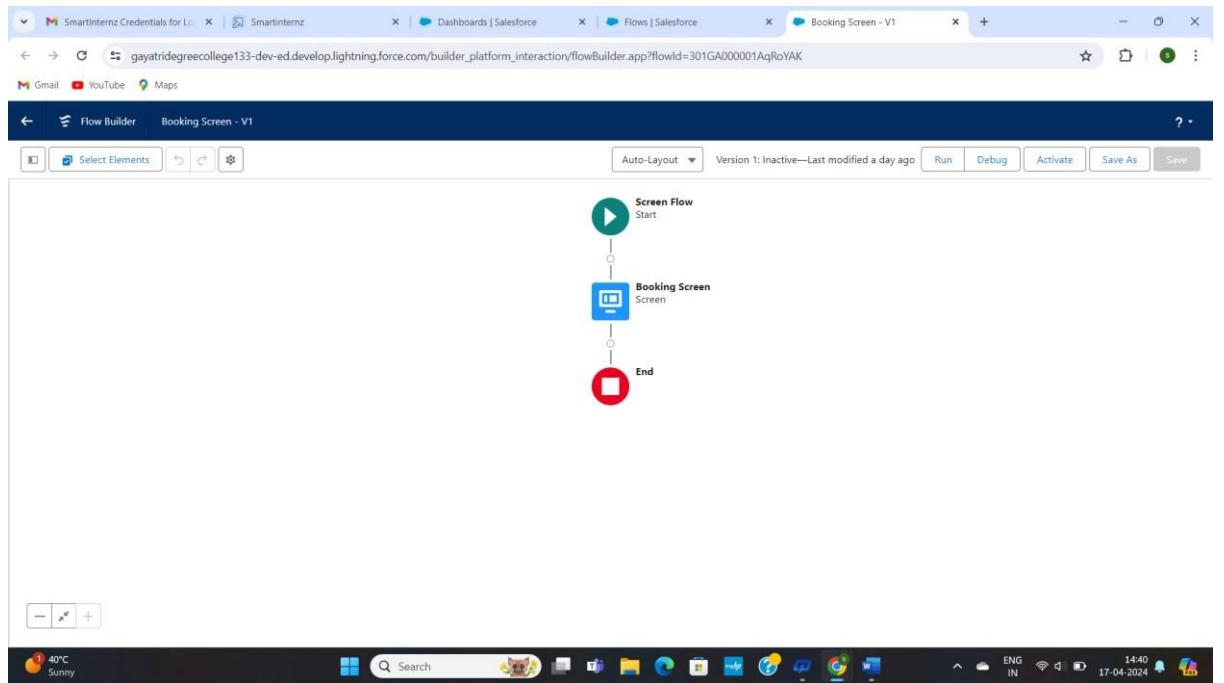
Milestone 13 : Flows

Creating A Screen Element On Booking Object

Let's start by creating a screen flow and adding a screen to it.

1. From Setup search for the “Flows” in quick find and select “New Flow”.
Select “Screen Flow” and then click on “Create”.
2. Place the cursor in between the Start and End element, a “+” icon appears, click on that and select “Screen”.
3. Give the label as “Booking Screen”, and select “Fields”.
4. Click on the lookup icon in the input field and select “New Resource”.
5. Filling the fields as given below:
 - API name : bookingObject
 - Datatype : Record
 - Object : Booking
 - Available for input : Checked
 - Available for output: Checked
6. Click on done
7. Once you are done with the creation of a resource with the name “bookingObject” go back to step 5 and click on the lookup icon and select “bookingObject” from the list.
8. After selecting “bookingObject” you'll have all the fields which are in the object, drag and drop each field on the screen.
9. Your screen will look like as shown below:
10. Click on done and Save.
11. The flow will look like as shown below.

12. Let's create a "Create Element" and add it to screen flow.

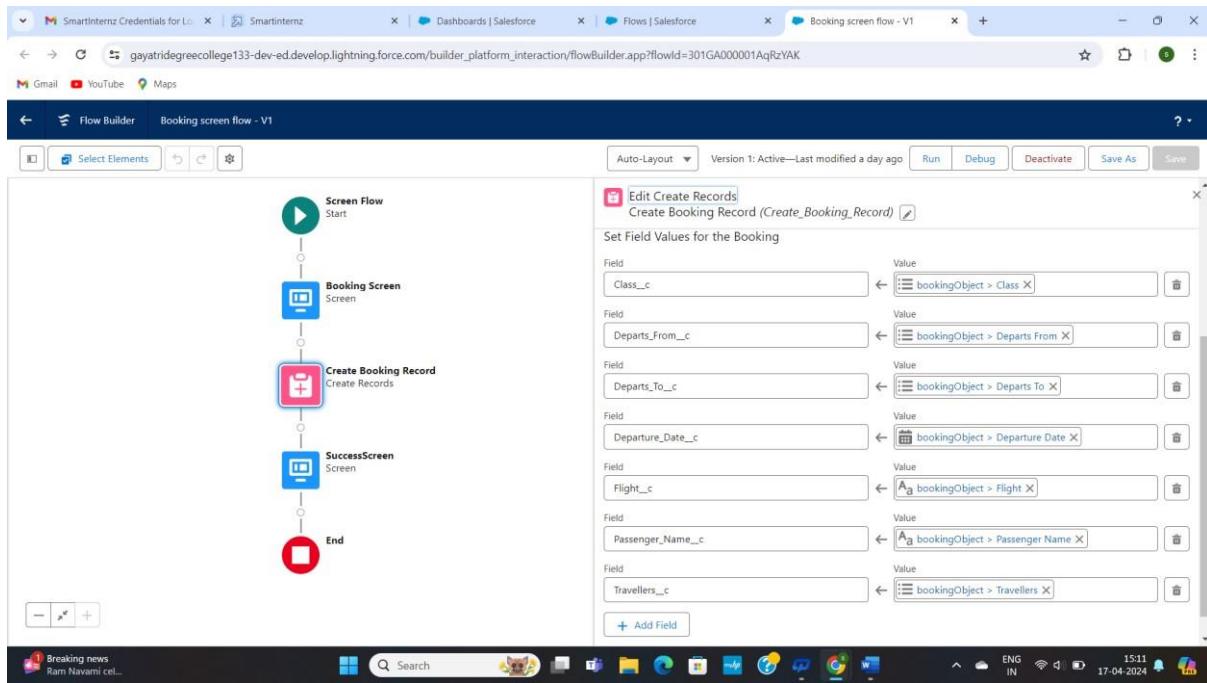


Creating A Create Element On Booking Object

1. In the Create New path, hover over circle and click '+' icon in between booking screen and end.
2. Click Create Records.
3. In the New Create Records window, enter these values.

Field	Value
Label	Create Booking Record
API name	Gets autopopulated
How many records to create	one
How to set the record fields	Use separate resources and literal values
Object	Booking

Map the fields as shown below:



Click Done.

Creating A Success Screen Element

1. In the Create New path, hover over circle and click '+' icon in between Create Booking Record and end.
2. Click the Screen element.
3. Give a label name as “SuccessScreen”.
4. From components search “Display text” and drag that component in the canvas and give an API Name as “SuccessMessage”.
5. Inside the text area copy this message.

We are happy to inform you that your booking for [tour name] is confirmed! Get ready to create some unforgettable memories. We've made things easy for you and included all of your booking details in this very email. All you need to do is show us this email on the day you arrive, and you'll be good to go

Date: {!bookingObject.Departure_Date__c}

Departs from: {!bookingObject.Departs_From__c}

Departs to: {!bookingObject.Departs_to__c}

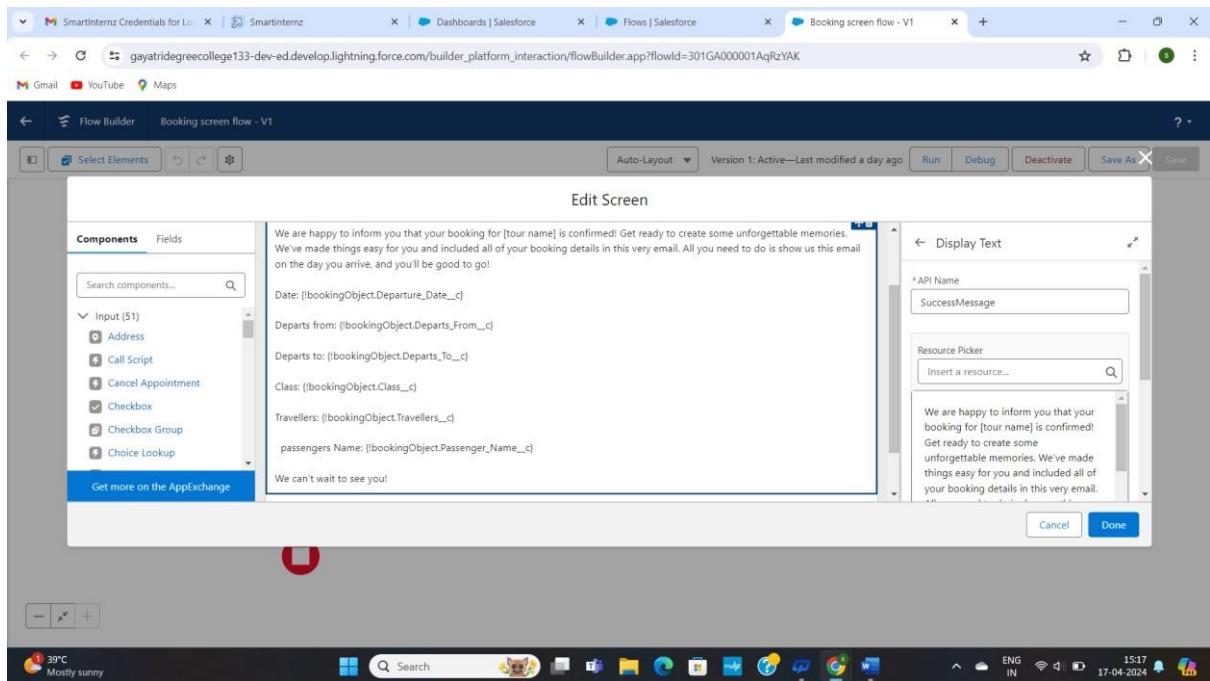
Class: {!bookingObject.Class__c}

Travellers: {!bookingObject.Travellers__c}

Passengers Name: {!bookingObject.Passenger_Name__r.Passenger_Name__c}

We can't wait to see you!

6. Click Done



6. Save the flow and Activate it.

