AIRLINES MANAGEMENT SYSTEM

Team ID : LTVIP2024TMID11598

Team Size : 4

Team Leader : P HARIPRASATH

Team member : R YUVARAJ

Airlines Management System

Hardware Required:

laptops

System Required:

Windows 8 machine Install with two web browser Bandwidth of 30mbps

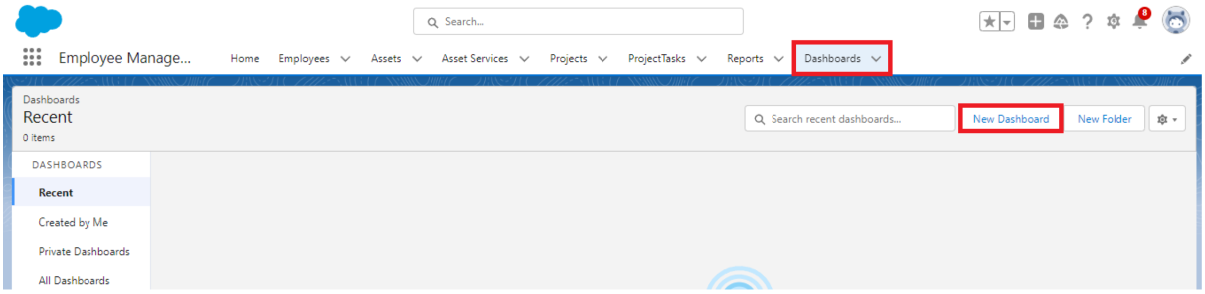
This project aims to enhance the efficiency and effectiveness of managing flights, reservations, and passenger information. The system enables airlines to manage their fleet, schedule flights, allocate seats, and handle bookings seamlessly. It provides functionalities for ticket reservations, seat availability checks, passenger check-ins, and baggage handling. Additionally, the system facilitates communication between airlines, airports, and passengers through automated notifications and alerts. With its user-friendly interface and robust database management, the Airlines Management System optimizes workflow, improves customer satisfaction, and ensures smooth operations for the entire airline industry.

MILESTONE 11 : DASHBOARDS

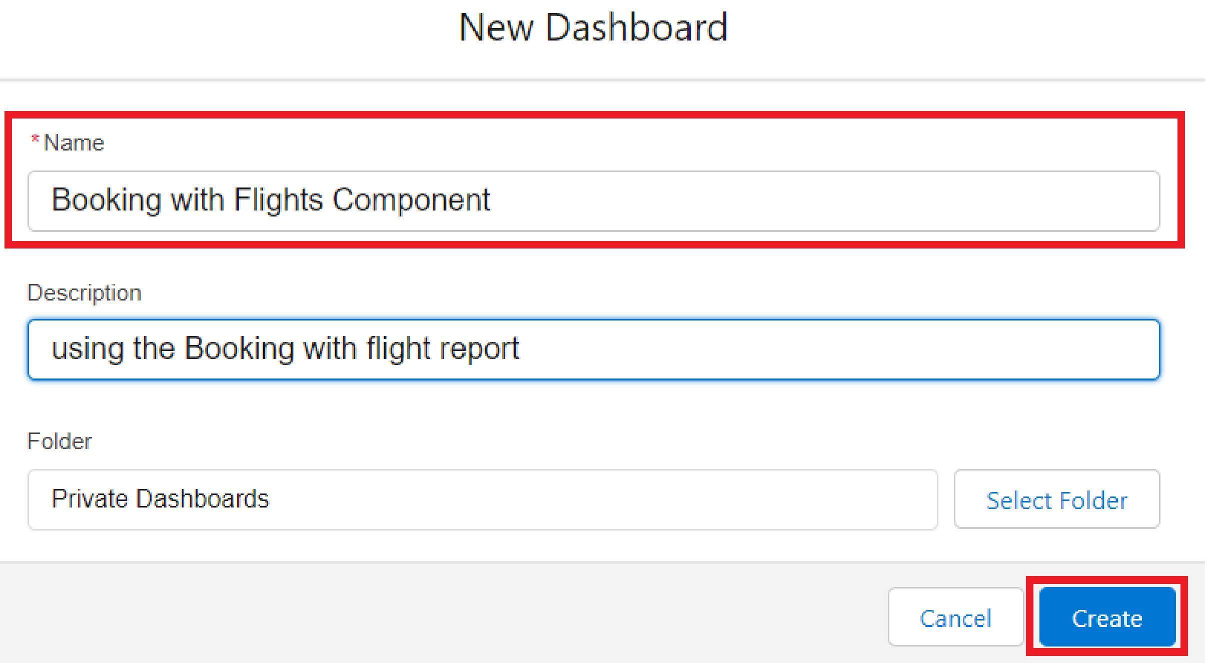
Activity 1

Create Dashboard

1. Go to the app --> click on the Dashboards tabs.



2. Give a Name and click on Create.



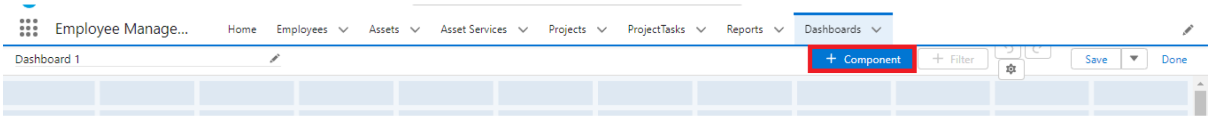
New Dashboard

* Name: Booking with Flights Component

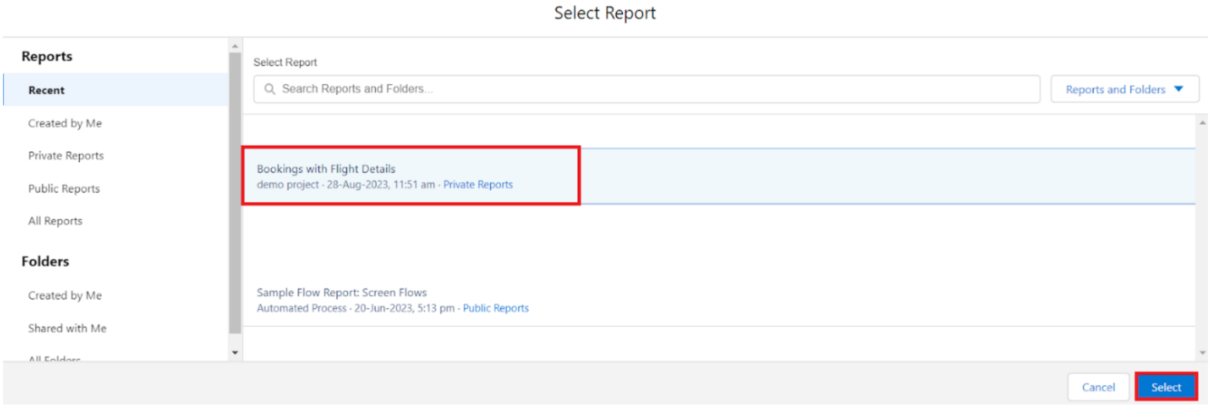
Description: using the Booking with flight report

Folder: Private Dashboards

3. Select add component.



4. Select a Report and click on select.

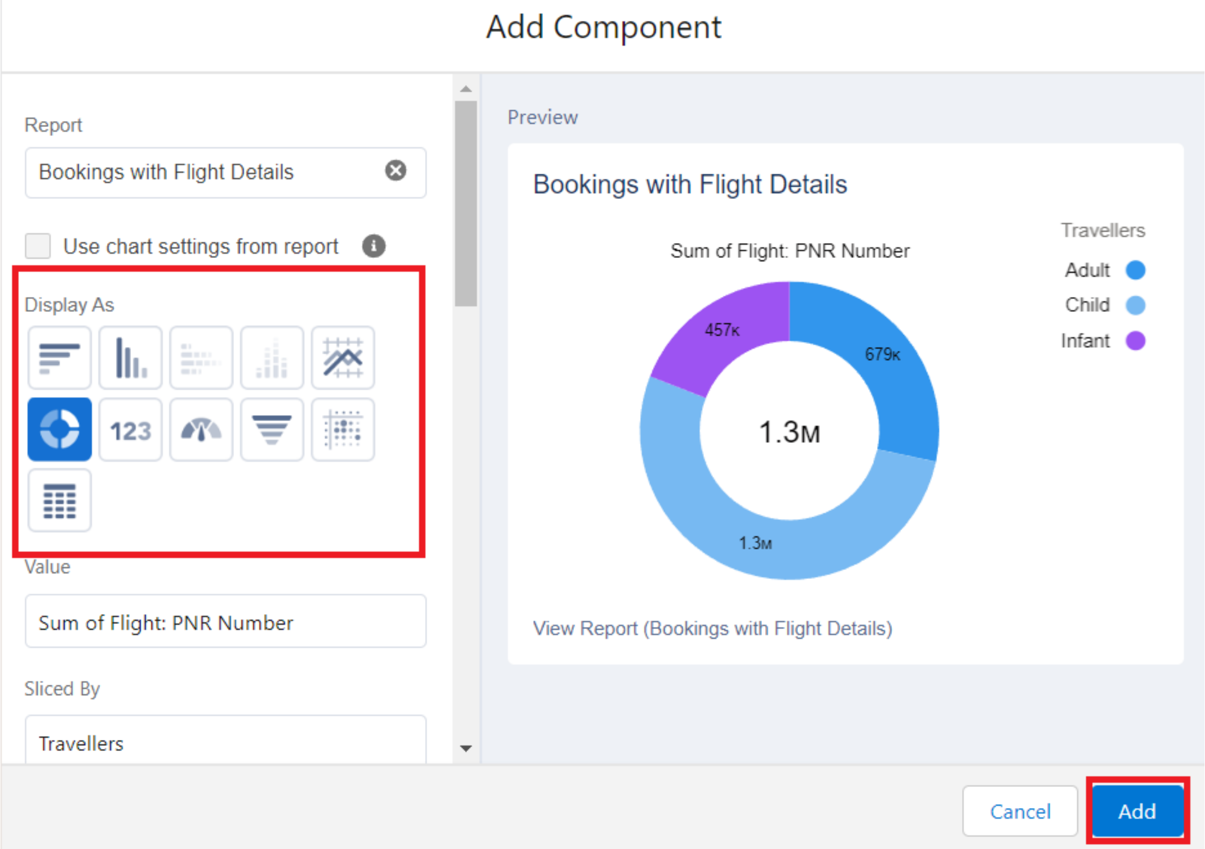


Select Report

Bookings with Flight Details
demo project · 28-Aug-2023, 11:51 am · Private Reports

Sample Flow Report: Screen Flows
Automated Process · 20-Jun-2023, 5:13 pm · Public Reports

5. Select the Donut option under display as and click on Add.



Add Component

Report

Bookings with Flight Details

☐ Use chart settings from report

Display As

Value

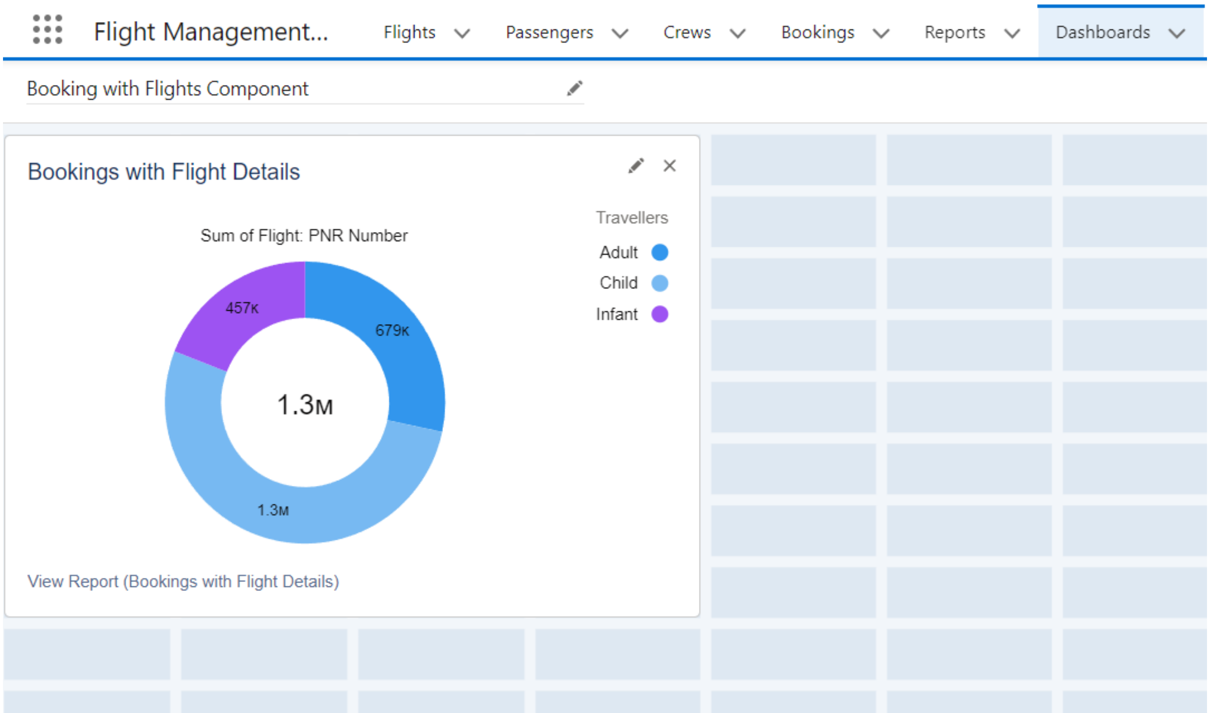
Sum of Flight: PNR Number

Sliced By

Travellers

Preview
Bookings with Flight Details
Sum of Flight: PNR Number
Travellers
Adult ●
Child ●
Infant ●
457K
679K
1.3M
1.3M
View Report (Bookings with Flight Details)
123
Cancel
Add

6. Your dashboard display as below:



Flight Management...
Flights
Passengers
Crews
Bookings
Reports
Dashboards
Booking with Flights Component
Bookings with Flight Details
Sum of Flight: PNR Number
Travellers
Adult ●
Child ●
Infant ●
457K
679K
1.3M
1.3M
View Report (Bookings with Flight Details)

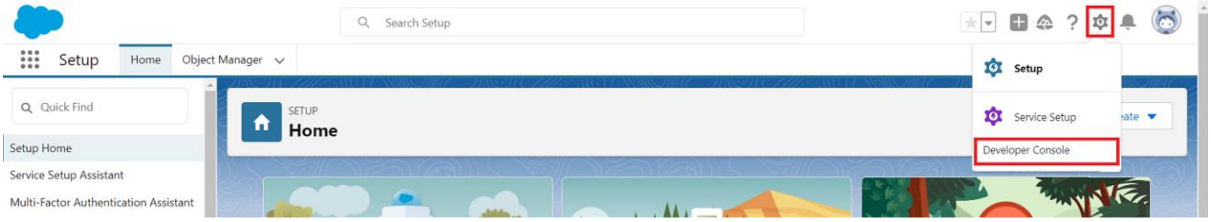
Activity 2

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

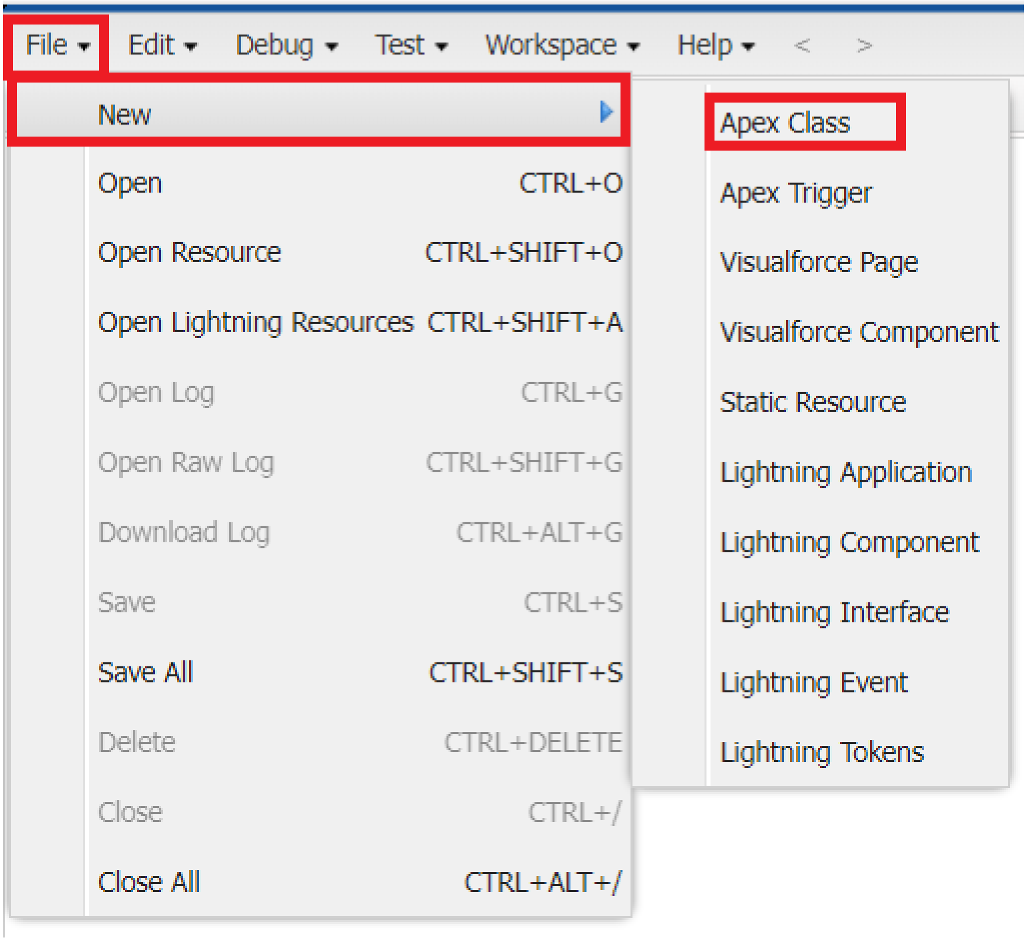
MILESTONE 12: APEX

Activity 1 : Create An Apex Class

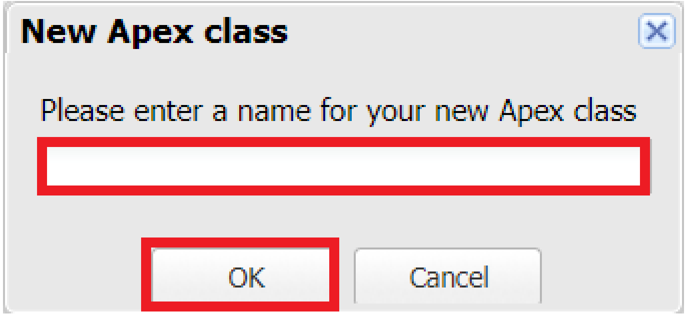
1. Go to Setup --> Click on the gear icon --> Select Developer Console.



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.



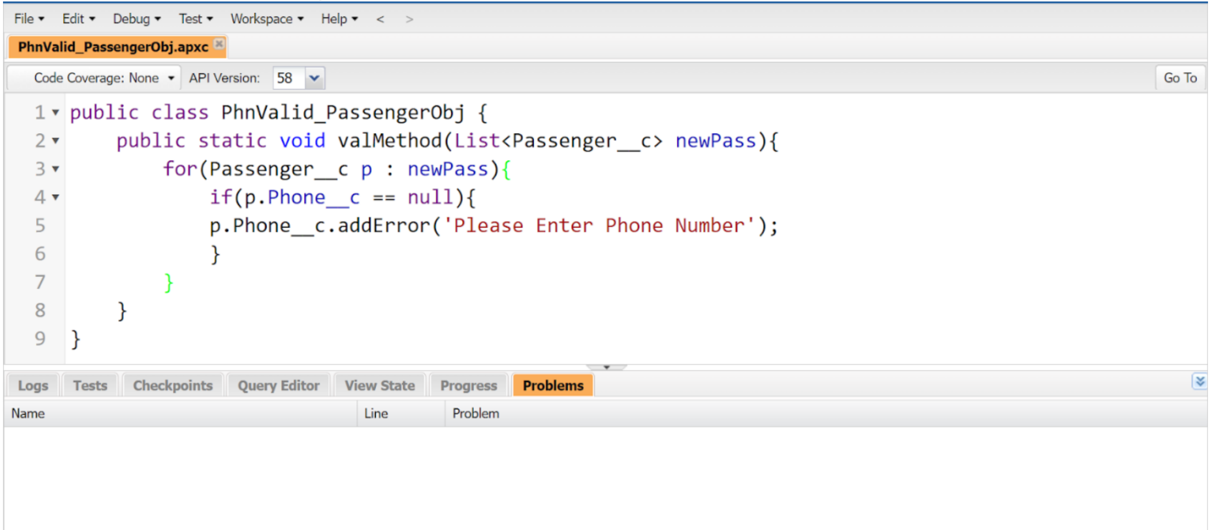
4. Give the Apex Class name as “PhnValid_PassengerObj”.



New Apex class

Please enter a name for your new Apex class

5. Click ok.
6. Now write the code logic here



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

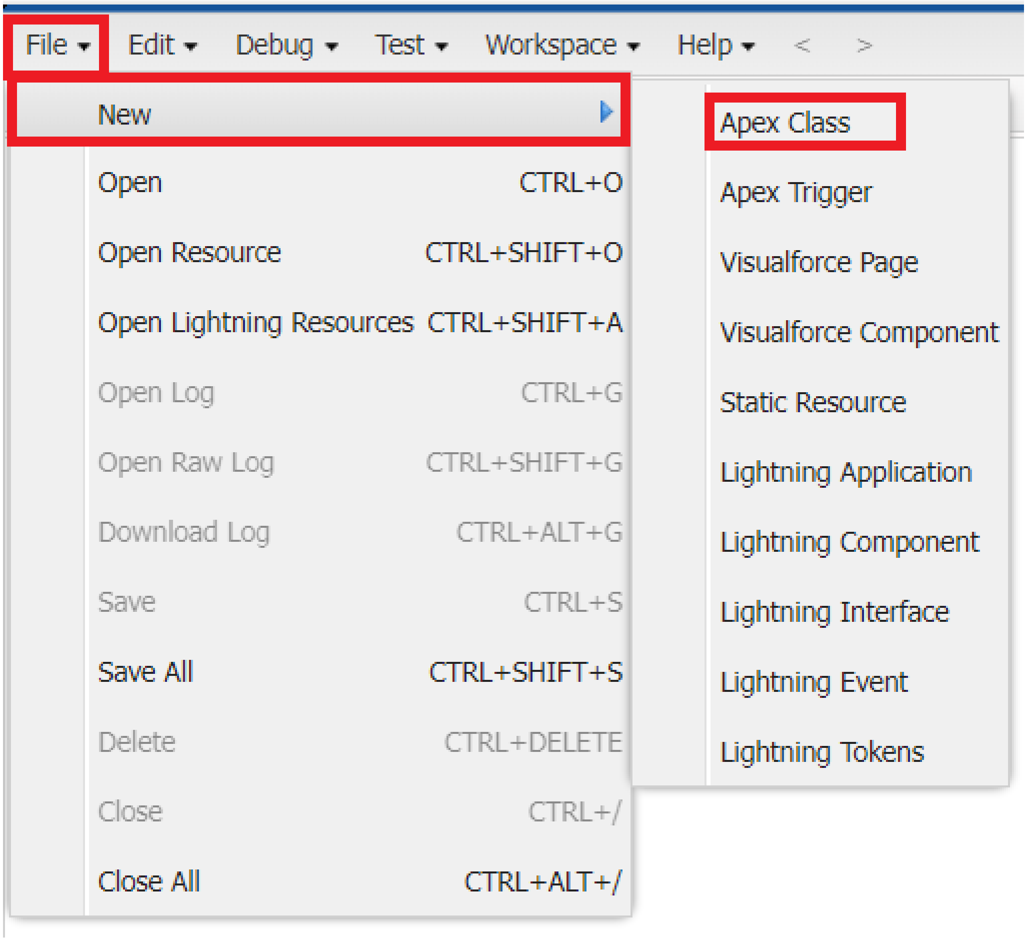
Source Snippet:

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

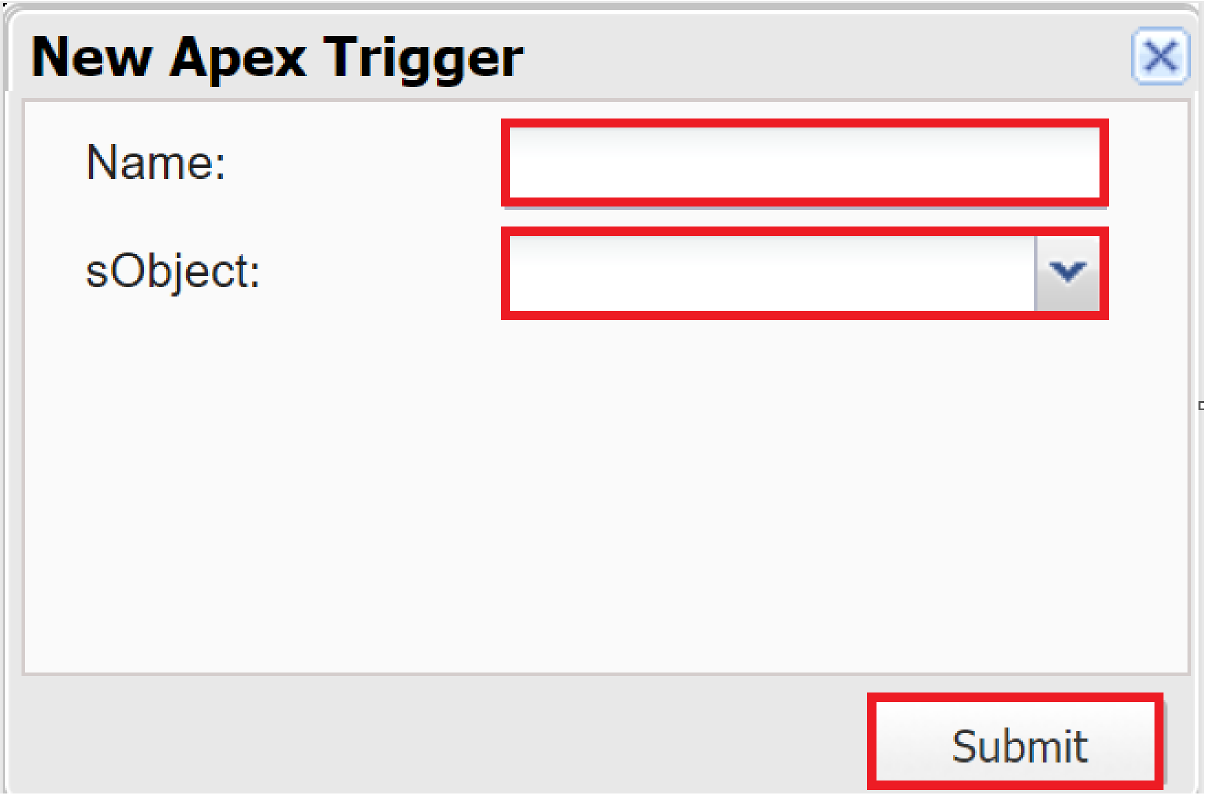
7. Save the code.(click on file --> Save)

Activity 2 : Create An Apex Trigger

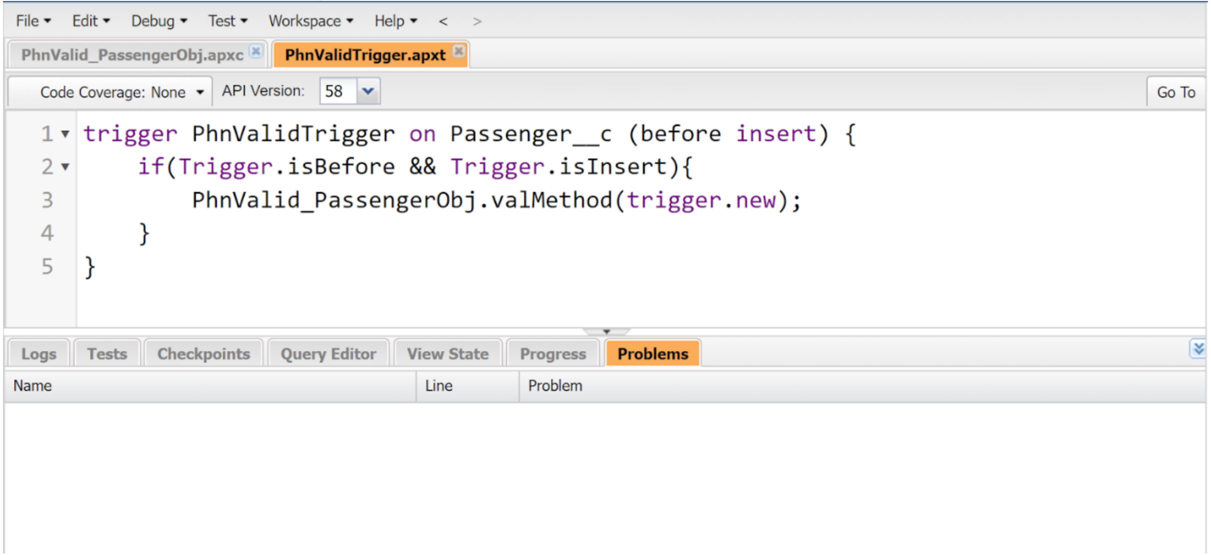
1. To create a new Apex Class follow the below steps:
   Click on the file --> New --> Apex Class.



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



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

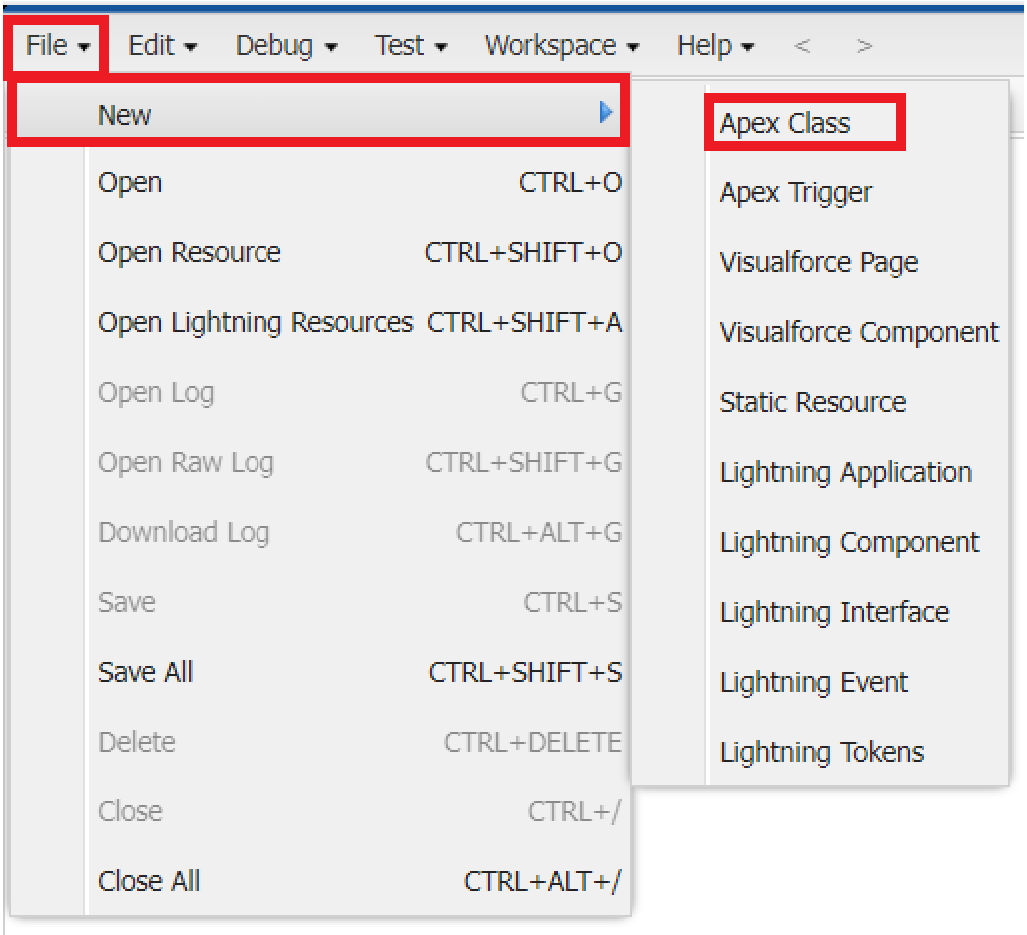
Source Code:

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

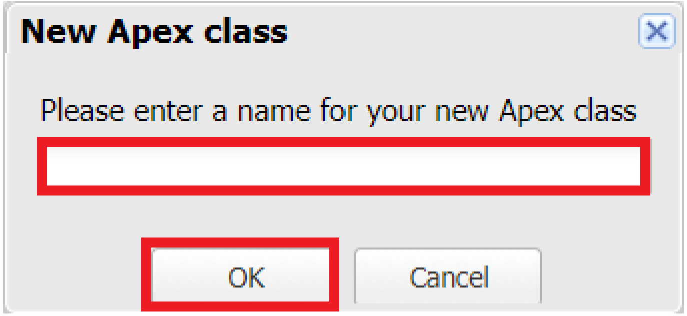
5. Save the code.(click on file --> Save)

Activity 3 : Create An Apex Test Class

1. To create a new Apex Class follow the below steps:
   Click on the file --> New --> Apex Class.



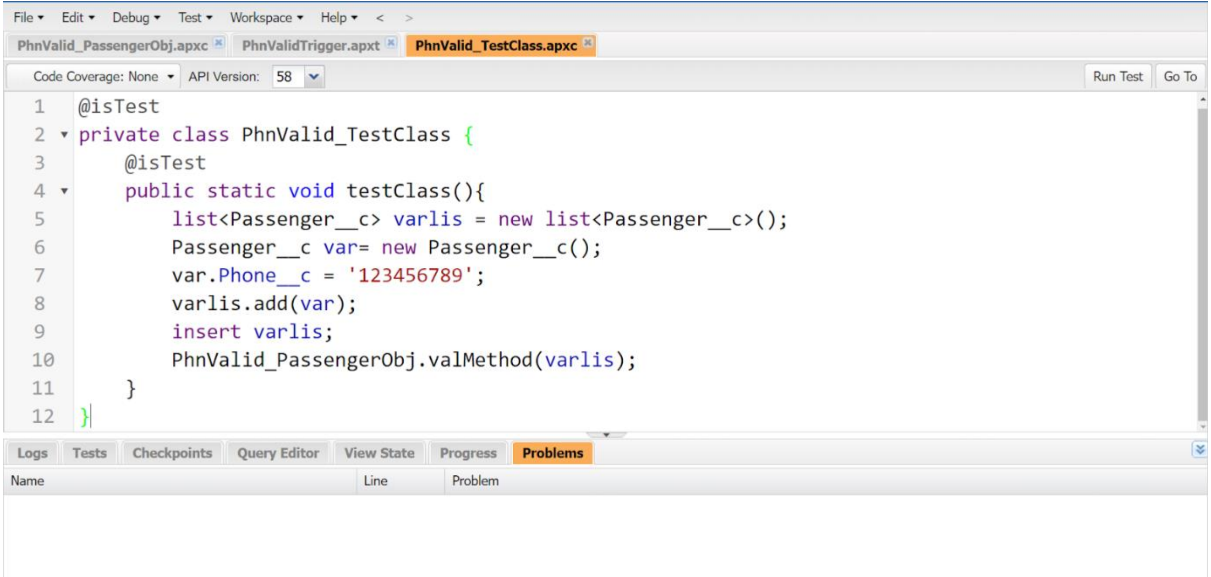
2. Give the Apex Class name as “PhnValid_TestClass”.


New Apex class

Please enter a name for your new Apex class

3. Click ok.

4. Now write the code logic here



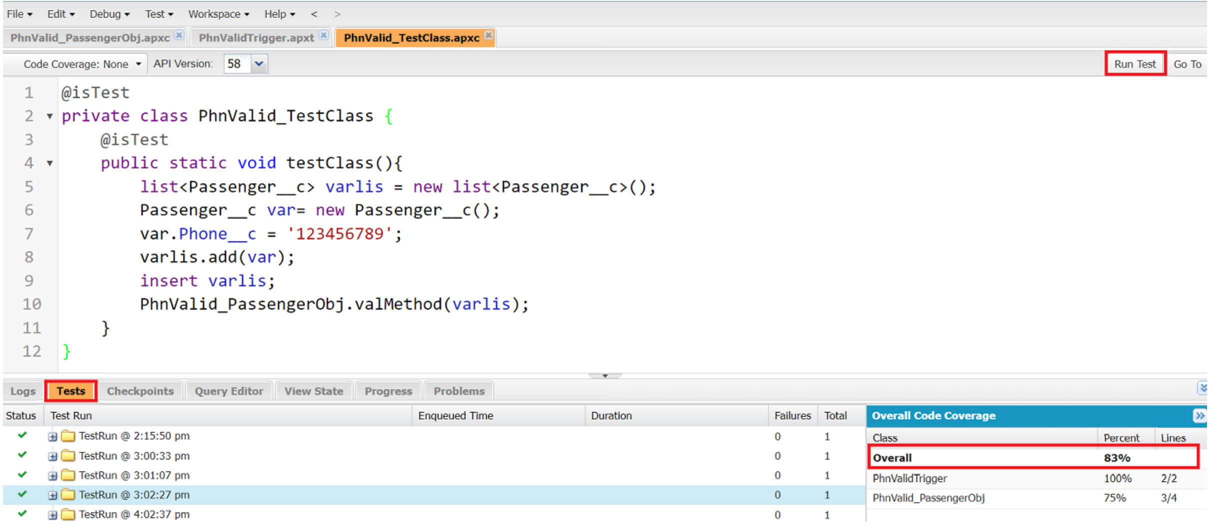
```
@isTest
private class PhnValid_TestClass {
    @isTest
    public static void testClass(){
        list<Passenger__c> varlis = new list<Passenger__c>();
        Passenger__c var= new Passenger__c();
        var.Phone__c = '123456789';
        varlis.add(var);
        insert varlis;
        PhnValid_PassengerObj.valMethod(varlis);
    }
}
```

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);
   }
}
```

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

6. 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%.



```
@isTest
private 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 = '123456789';
        varlis.add(var);
        insert varlis;
        PhnValid_PassengerObj.valMethod(varlis);
    }
}
```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 2:15:50 pm			0	1
✓	TestRun @ 3:00:33 pm			0	1
✓	TestRun @ 3:01:07 pm			0	1
✓	TestRun @ 3:02:27 pm			0	1
✓	TestRun @ 4:02:37 pm			0	1

Overall Code Coverage

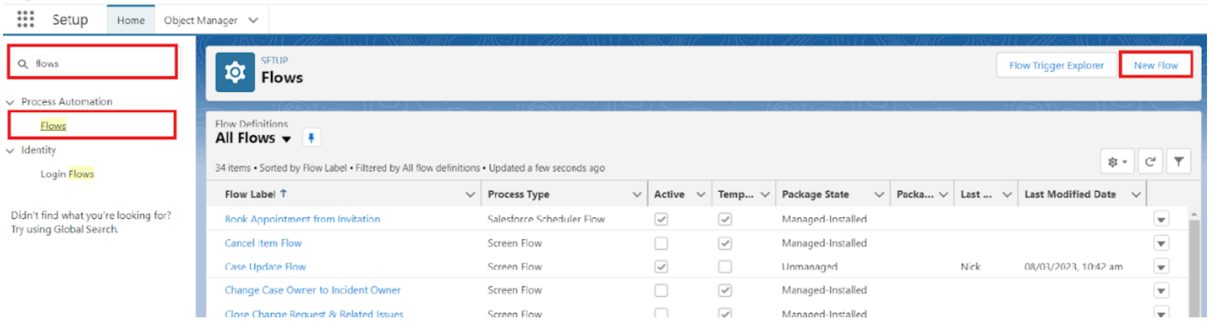
Class	Percent	Lines
Overall	**83%**	
PhnValidTrigger	100%	2/2
PhnValid_PassengerObj	75%	3/4

MILESTONE 13 : FLOWS

Activity 1 : Creating A Screen Element On Booking Object

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

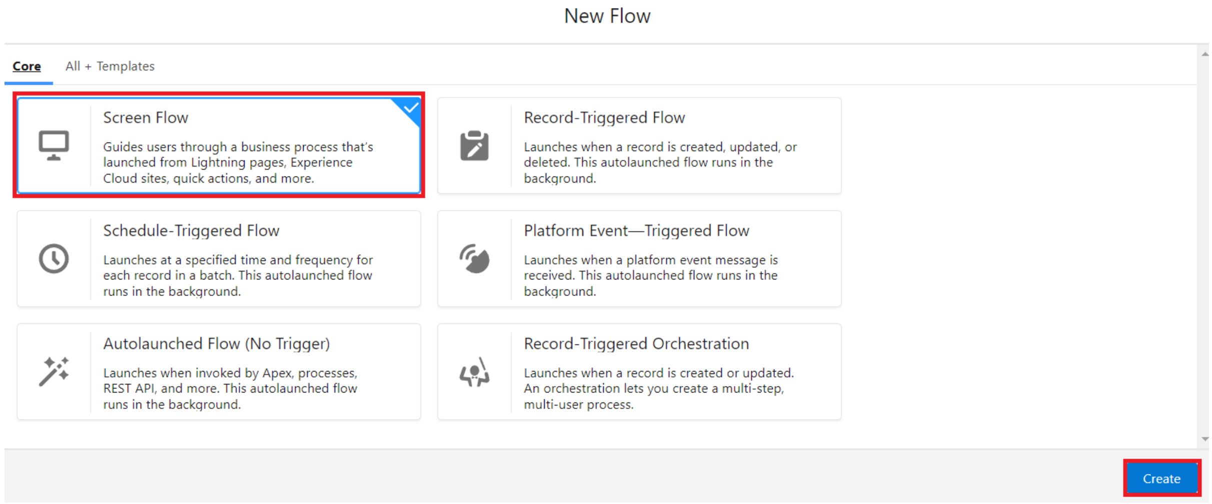
1. From Setup search for the “Flows” in quick find and select “New Flow”.



34 items • Sorted by Flow Label • Filtered by All flow definitions • Updated a few seconds ago

Flow Label ↑	Process Type	Active	Temp...	Package State	Packa...	Last ...	Last Modified Date
Book Appointment from Invitation	Salesforce Scheduler Flow	☑	☑	Managed-Installed			
Cancel Item Flow	Screen Flow	□	☑	Managed-Installed			
Case Update Flow	Screen Flow	☑	□	Unmanaged		Nick	08/03/2023, 10:42 am
Change Case Owner to Incident Owner	Screen Flow	□	☑	Managed-Installed			
Close Change Request & Related Issues	Screen Flow	□	☑	Managed-Installed			

2. Select “Screen Flow” and then click on “Create”.



New Flow

Core All + Templates

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.

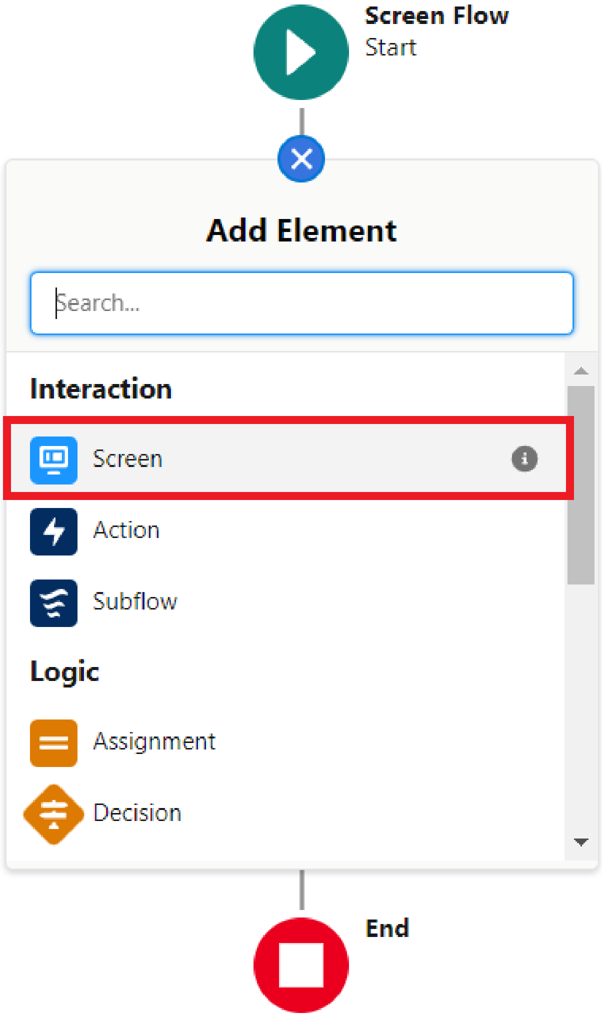
Platform Event—Triggered Flow
Launches when a platform event message is received. This autolaunched flow runs in the background.

Autolaunched Flow (No Trigger)
Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.

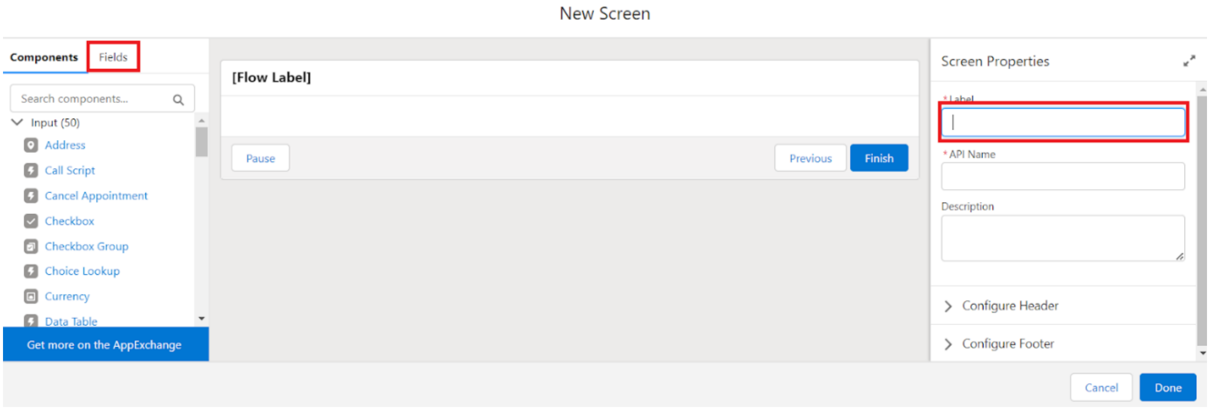
Record-Triggered Orchestration
Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

Create

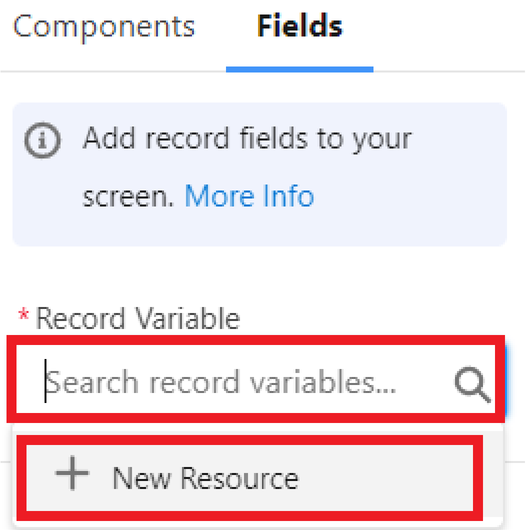
Place the cursor in between the Start and End element, a “+” icon appears, click on that and select “Screen”.



3. Give the label as “Booking Screen”, and select “Fields”.



4. Click on the lookup icon in the input field and select “New Resource”.



Add record fields to your screen. More Info

5. Filling the fields as given below:

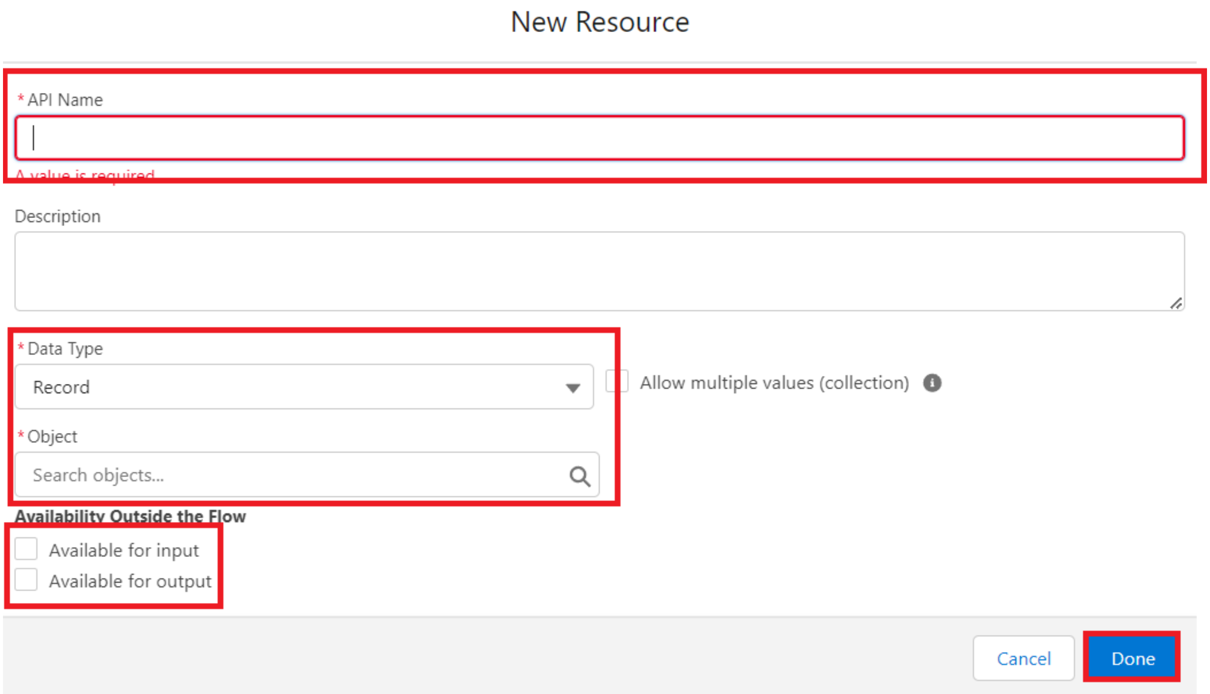
a. API name : bookingObject

b. Datatype : Record

c. Object : Booking

d. Available for input : Checked

e. Available for output: Checked



New Resource

* API Name

A value is required

Description

* Data Type

Record

□ Allow multiple values (collection)

* Object

Search objects...

Availability Outside the Flow

□ Available for input

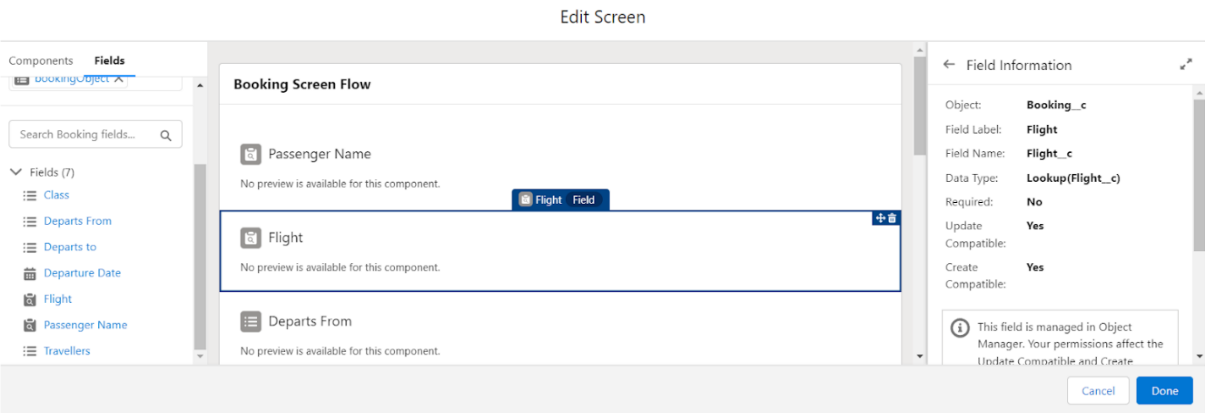
□ Available for output

7. Click on done

8. 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.

9. After selecting “bookingObject” you’ll have all the fields which are in the object, drag and drop each field on the screen.

10. Your screen will look like as shown below:



Edit Screen

Fields (7)
- Class
- Departs From
- Departs to
- Departure Date
- Flight
- Passenger Name
- Travellers

Booking Screen Flow

Passenger Name

No preview is available for this component.

Flight

No preview is available for this component.

Departs From

No preview is available for this component.

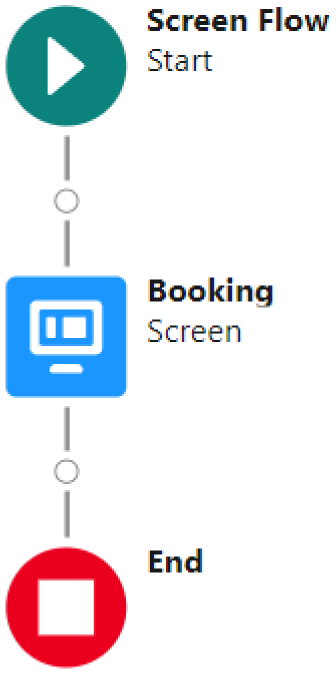
Field Information

Object:	Booking__c
Field Label:	Flight
Field Name:	Flight__c
Data Type:	Lookup(Flight__c)
Required:	No
Update Compatible:	Yes
Create Compatible:	Yes

This field is managed in Object Manager. Your permissions affect the Update Compatible and Create

11. Click on done and Save.

12. The flow will look like as shown below.



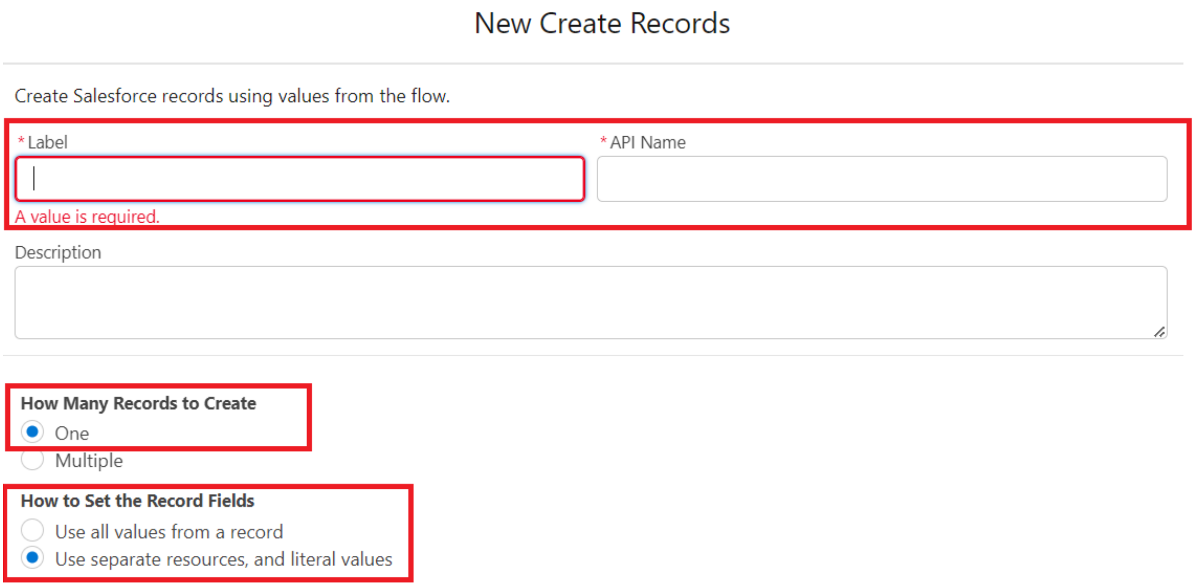
Let’s create a “Create Element” and add it to screen flow.

Activity 2 : 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



New Create Records

Create Salesforce records using values from the flow.

* Label

A value is required.

* API Name

Description

How Many Records to Create

● One

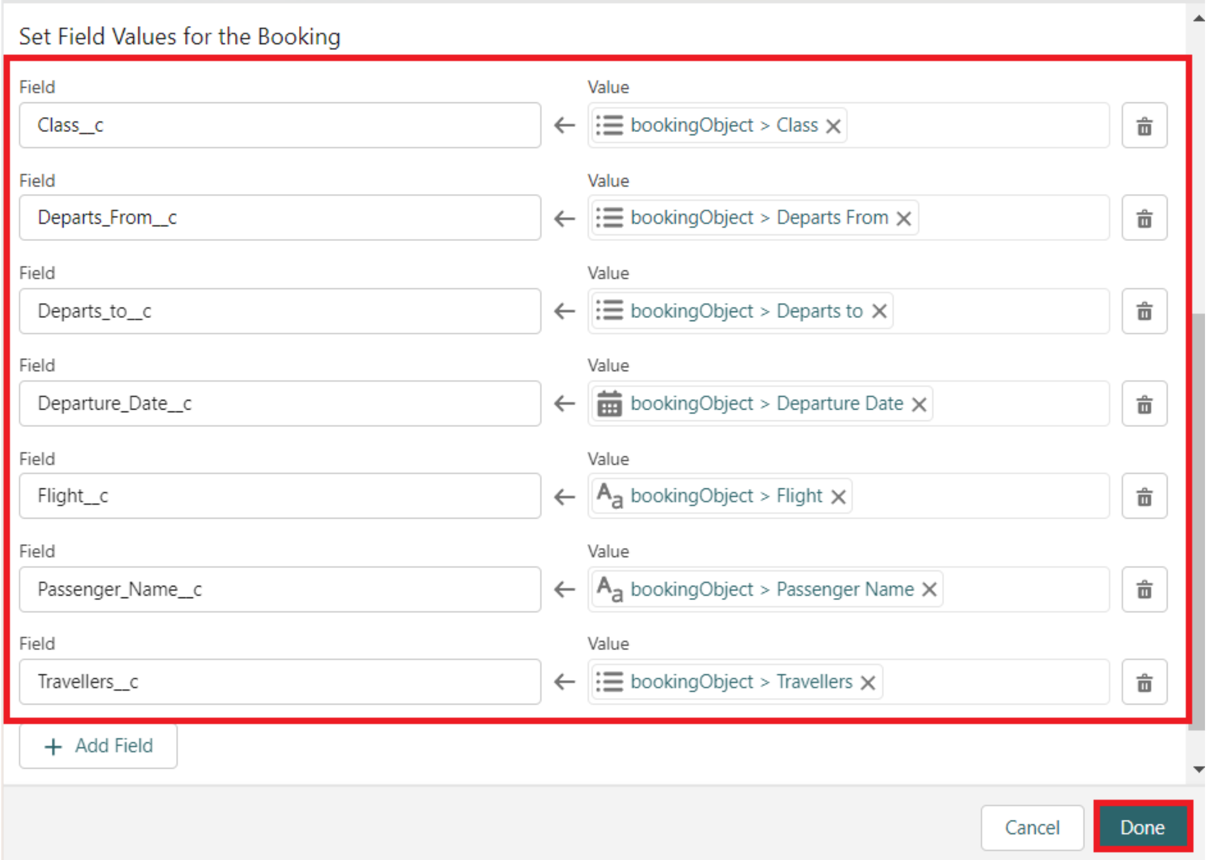
○ Multiple

How to Set the Record Fields

○ Use all values from a record

● Use separate resources, and literal values

4. Map the fields as shown below:



Set Field Values for the Booking

Field		Value
Class__c	←	bookingObject > Class
Departs_From__c	←	bookingObject > Departs From
Departs_to__c	←	bookingObject > Departs to
Departure_Date__c	←	bookingObject > Departure Date
Flight__c	←	bookingObject > Flight
Passenger_Name__c	←	bookingObject > Passenger Name
Travellers__c	←	bookingObject > Travellers

5. Click Done.

Activity 3 : 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
7. Save the flow and Activate it.

Thank you...!!