Capstone Project Report
On
“**Travel App**”

Submitted by:

Name: SAGAR KUMAR GUPTA
Batch Wipro Salesforce B13 (WIP-SF-13)
LMS Id: MGSA_490
Date: 28/02/2023

Table of contents

Abstract

Salesforce, Inc. is an American cloud-based software company headquartered in San Francisco, California. It provides customer relationship management (CRM) software and applications focused on sales, customer service, marketing automation, analytics, and application development.

Salesforce's main technologies are tools for customer management. Other products enable customers to create apps, integrate data from other systems, visualize data, and offer training courses.

Force.com applications are built using declarative tools, backed by Lightning and Apex, a proprietary Java-like programming language for Force.com, as well as Visualforce, a framework including an XML syntax typically used to generate HTML. The Force.com platform typically receives three complete releases a year. As the platform is provided as a service to its developers, every single development instance also receives all these updates.

In here we work on how to Set up the Company Profile, Configuring the user Interface, setting up Activities and Calendars, Configuring Search Settings, Setting up Chatter Groups, Mobile Access with salesforce.

Introduction

About Salesforce:-

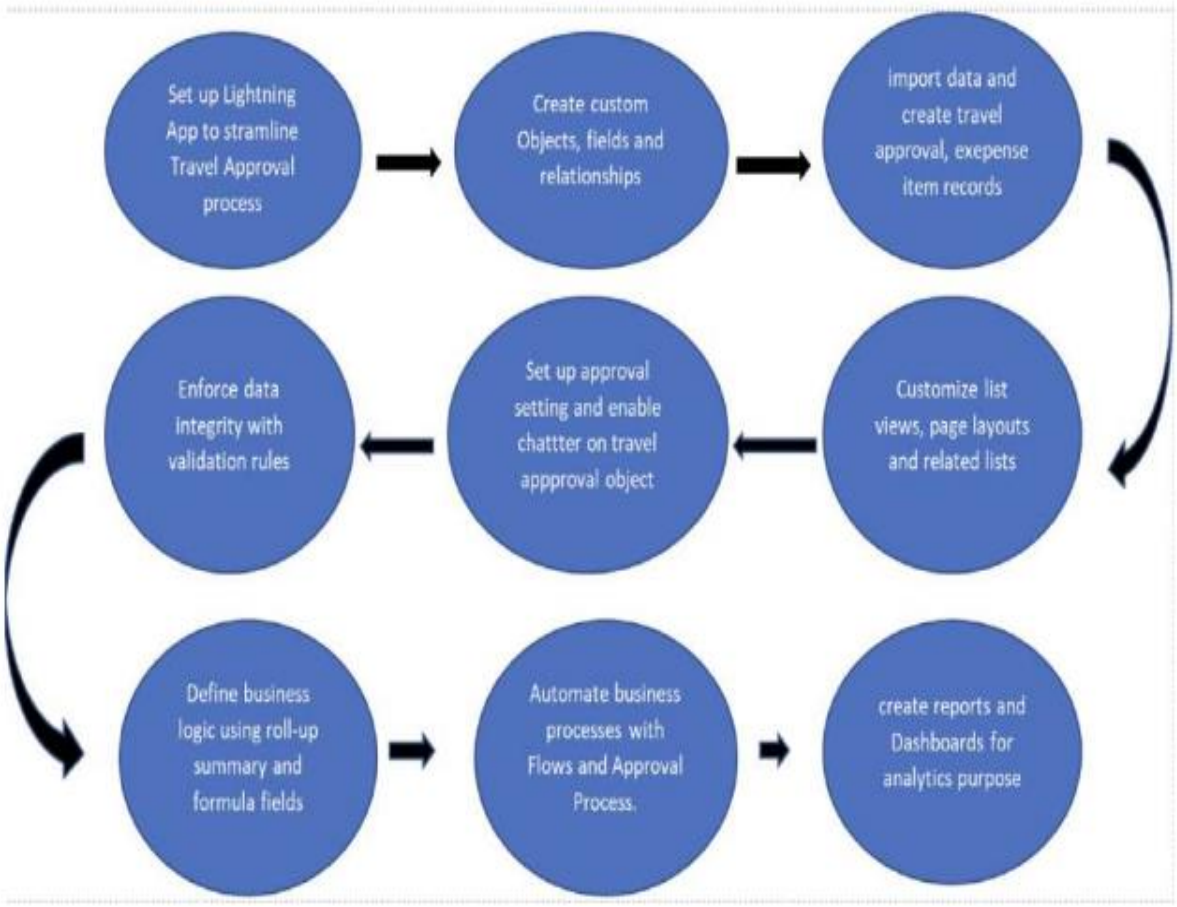
Salesforce is a cloud-based software company that provides its customers with a platform to develop their own applications without following the tough steps that they used to follow in the legacy system. The software or application once created can be uploaded onto the cloud allowing the end-users to view them.

Salesforce is currently providing various software solutions and platforms for developers to create and distribute custom software/applications. Tech giants like Google, Twitter, Amazon, and Facebook are using Salesforce either in the form of SaaS or PaaS.

Salesforce developers can make an application on the cloud and share it with multiple companies across multiple domains by using Salesforce.

Talking about HR systems, every company across the globe has an HR team. Each HR team would require an HR application to store employee records. Almost all specifications for such an application would be common for all companies. So, as a developer, it would be very easy to create a Salesforce application for such specifications, post it onto the cloud, and provide it as a service to multiple clients at the same time. Maintenance of the same can be done altogether too. So basically, the problem of scalability gets eliminated.

Flow of the Project



Software Requirements

For the fastest and most stable experience, we recommend:

- An Octane 2.0 score of 30,000 or greater
- Network latency of 150 ms or less
- Download speed of 3 Mbps or greater
- At least 8 GB of RAM, with 3 GB available for Salesforce browser tabs

Minimum requirements are:

- An Octane 2.0 score of 20,000 or greater
- Network latency of 200 ms or less
- Download speed of 1 Mbps or greater
- At least 5 GB of RAM, with 2 GB available for Salesforce browser tabs

OR

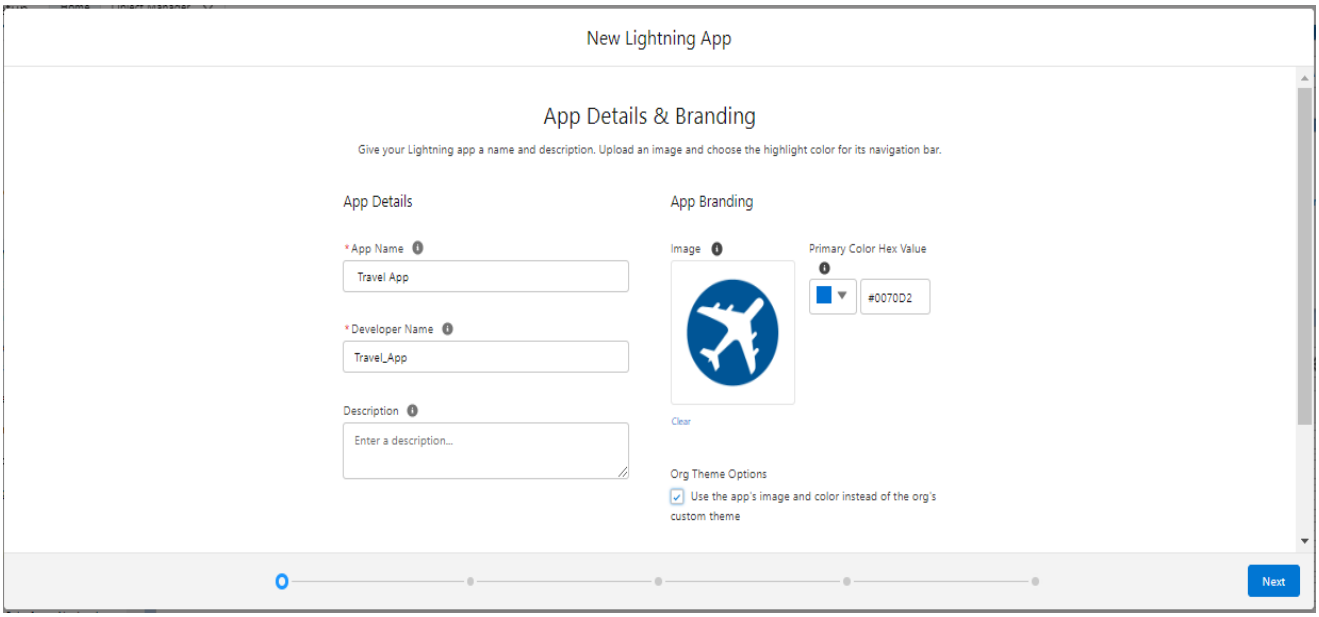
Requirements	
Windows	
Operating system	Windows 8.1 64-bit, Windows 8 64-bit, Windows 7 Service Pack 1 64-bit, Windows Vista Service Pack 2 64-bit
CPU	Core 2 Quad Q6600 at 2.4 GHz or AMD Phenom 9850 at 2.5 GHz
Memory	4 GB RAM
Free space	65 GB of free space
Graphics hardware	DirectX 10-compatible GPU: GeForce 9800GT 1GB or ATI Radeon HD 4870 1GB
Sound hardware	DirectX 10 compatible sound card

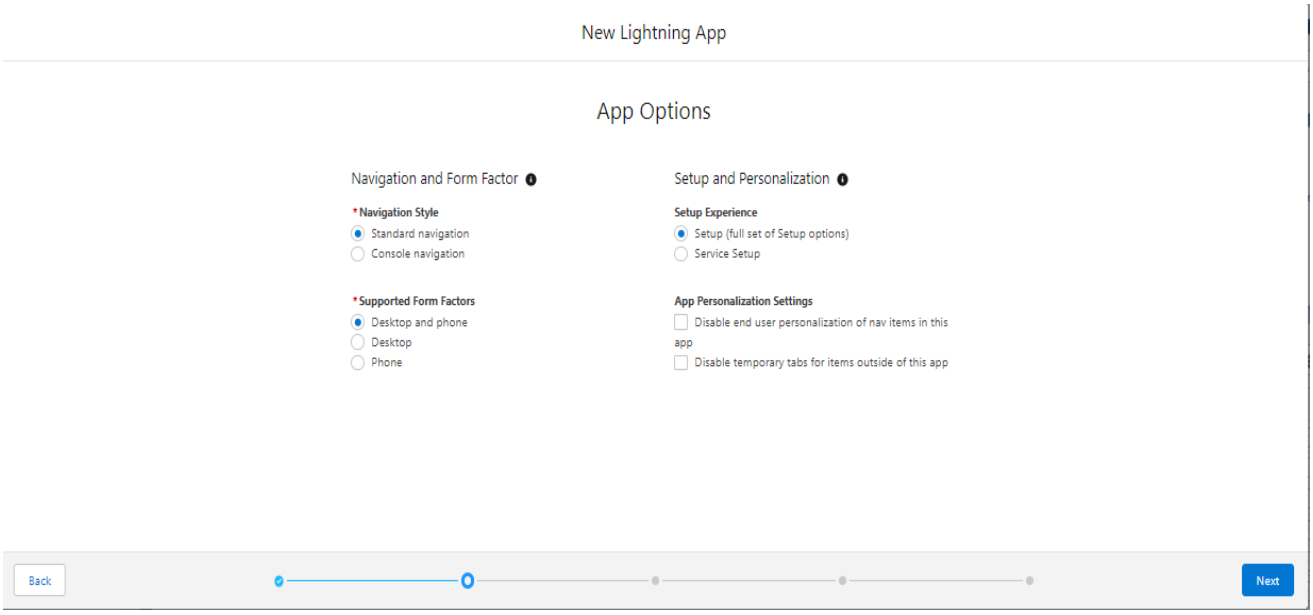
Screen shots

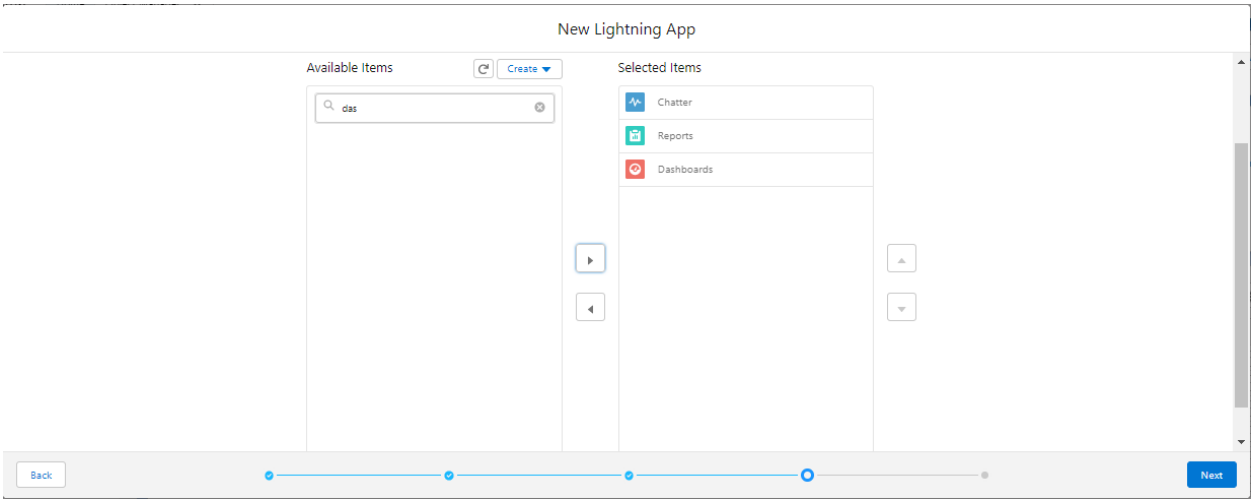
Module – 1

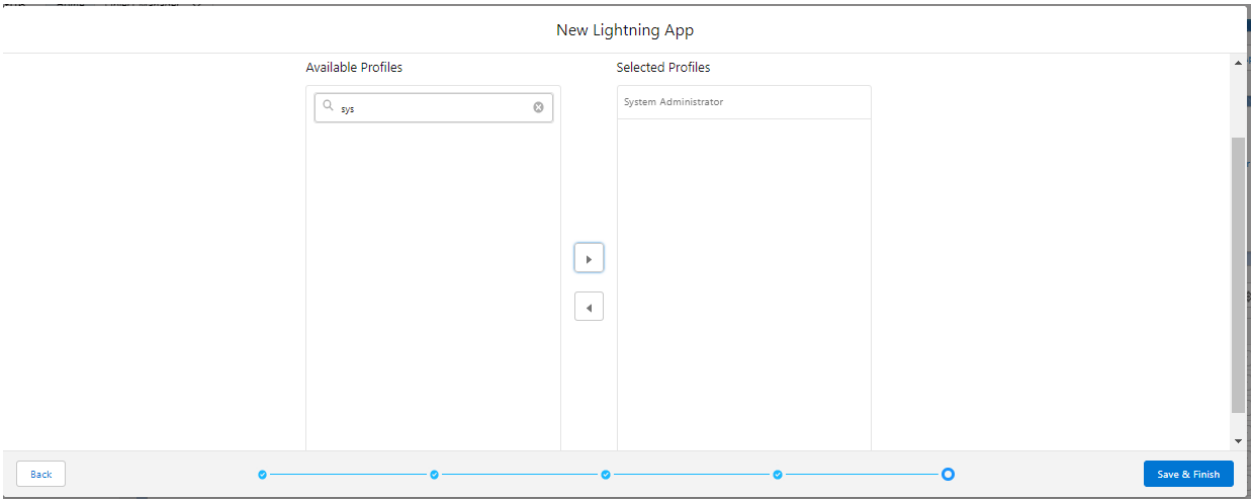
Exercise 1: -

Step 1: - Create a new custom lightning App, name: **Travel App**

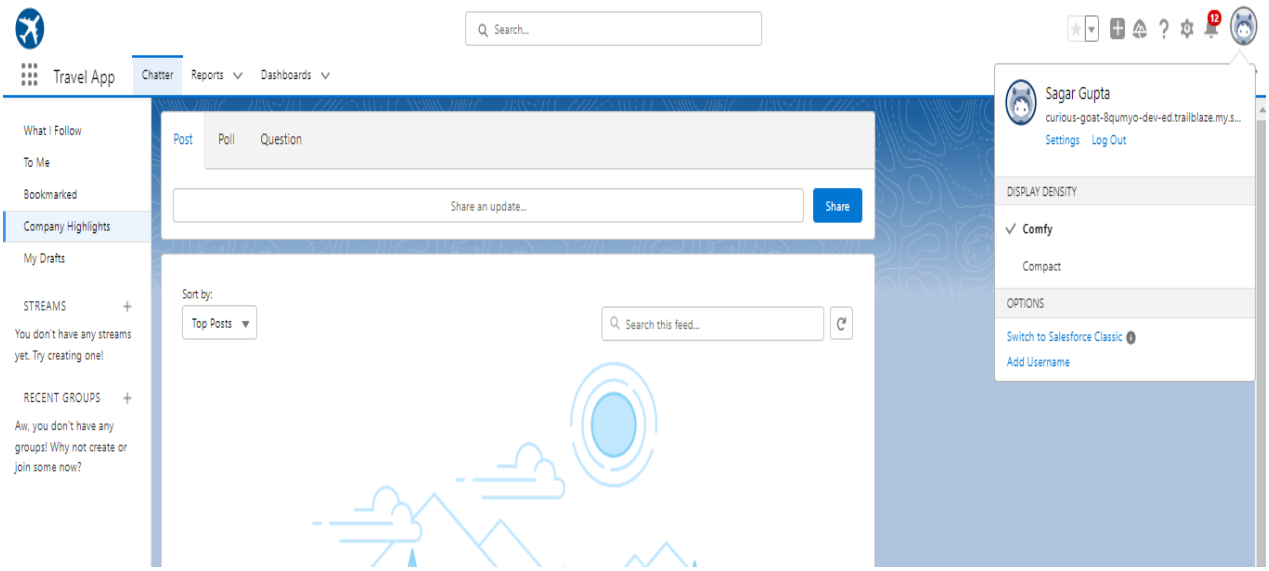






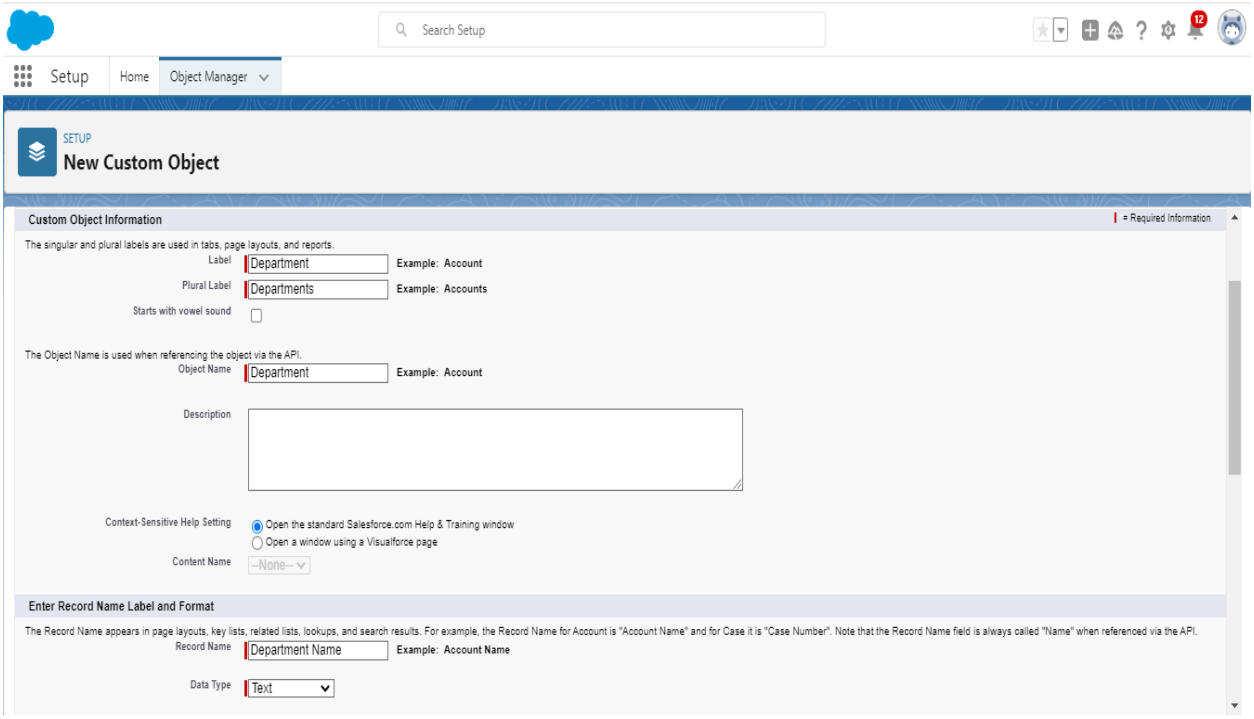


Final output of step 1 -



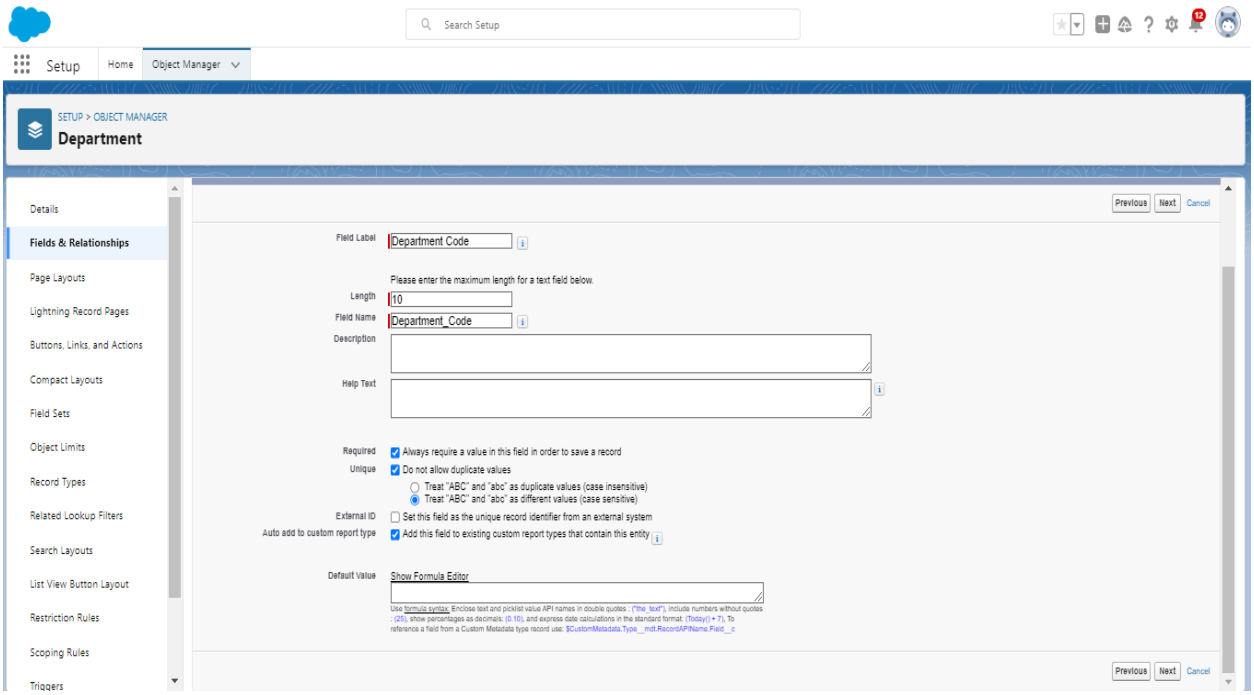
Step 2 : - Create a Department custom object.

Setup->Object Manager->Create Custom object (From top left dropdown button)

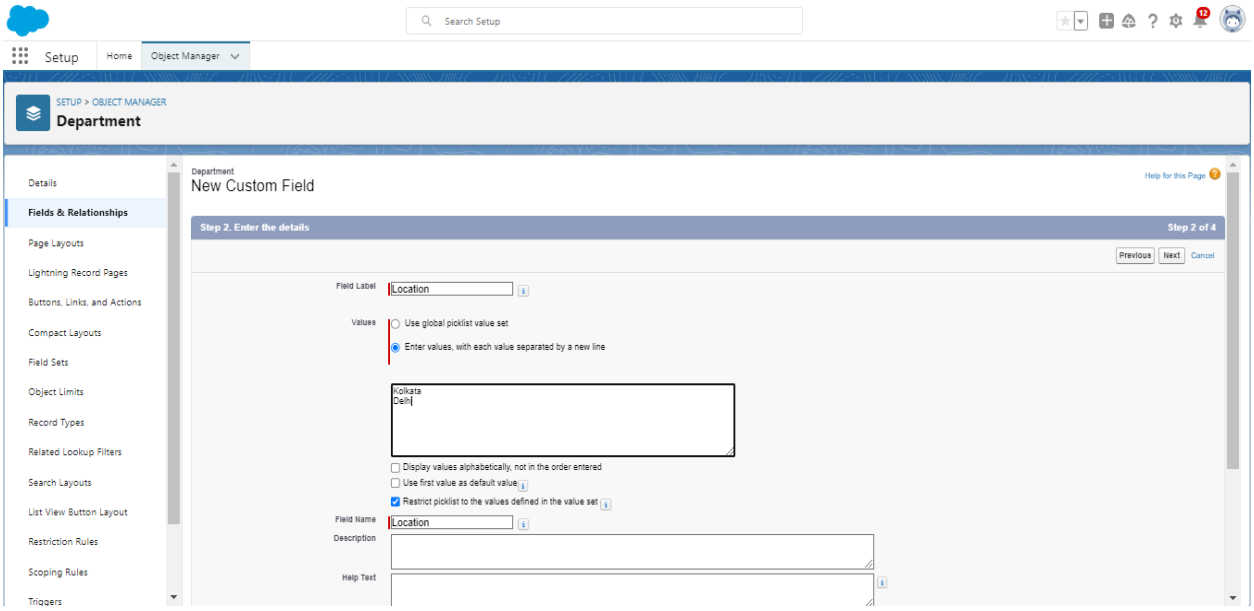


Step 3 :-Create the following Custom Field in Department Object.

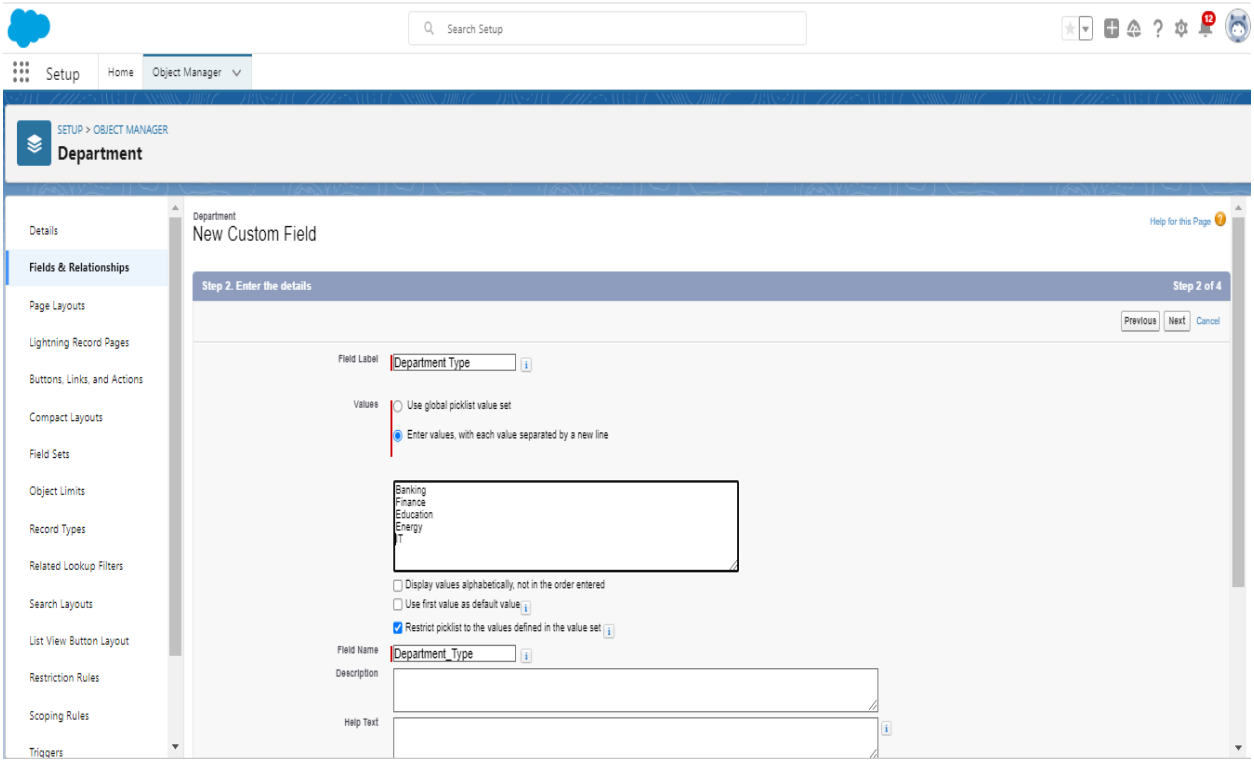
(a). Department Code, Text, Length = 10, Required, Select Unique & Case sensitive



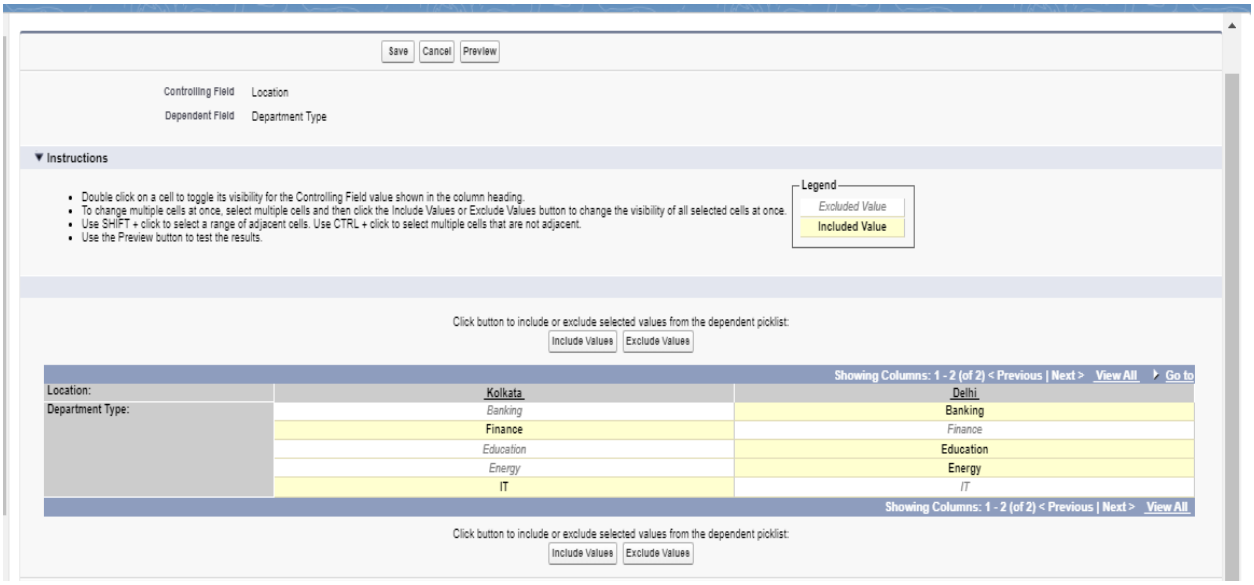
(b). Location, Picklist, Value: Kolkata, Delhi.



(c). Department Type, Picklist, Values: Banking, Finance, Education, Energy, IT.

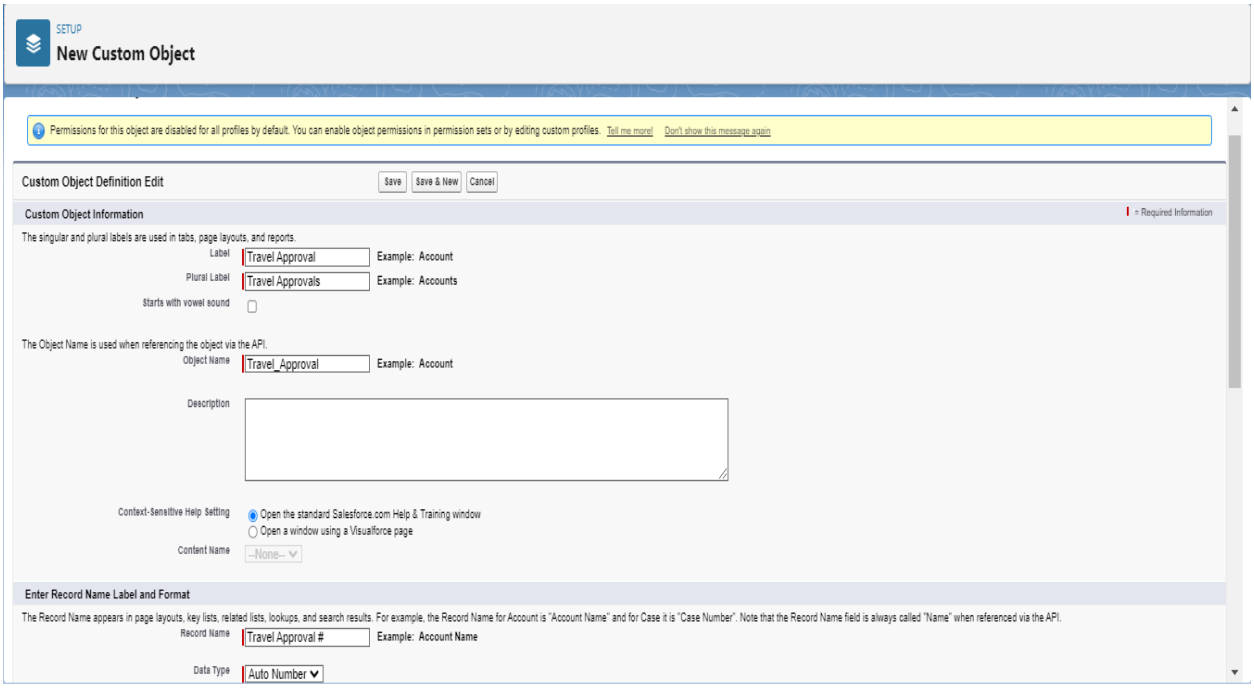


(d). Create Field Dependency, Controlling field =Location, Dependent field = Department Type



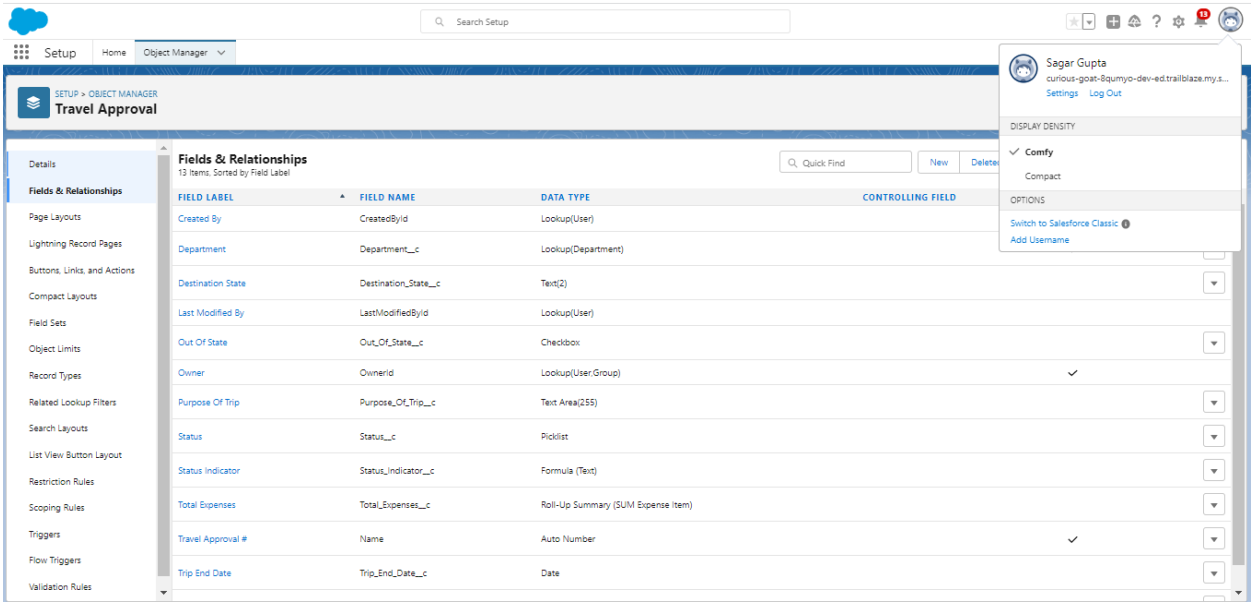
Step 4 : - Create a Travel Approval custom object.

Setup->Object Manager->Create Custom object (From top left dropdown button)



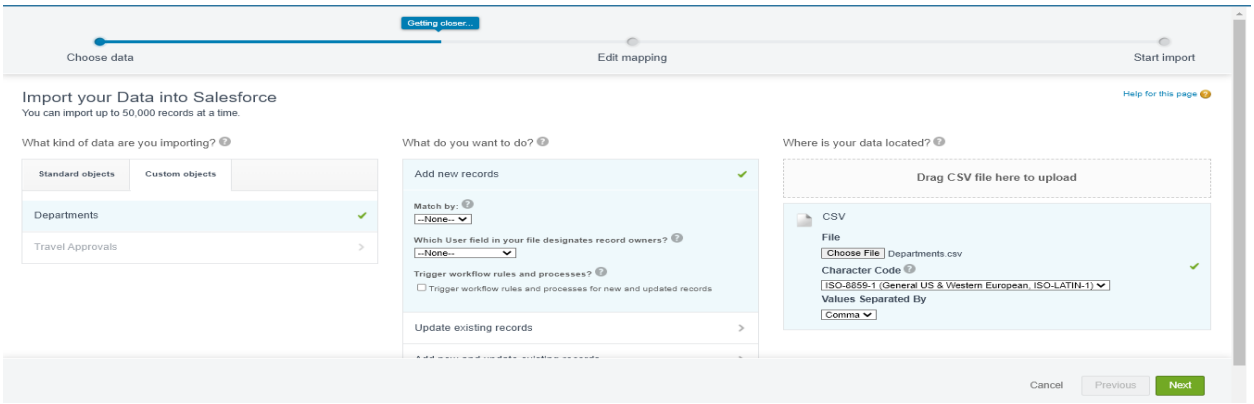
Now, Create the following Custom Field in Travel Approval Object.

- Purpose of Trip, Text Area.
- Status, Picklist, Values = New, Submitted, Pending Approval, Approved, Rejected, Draft.
- Trip Start Date, Date.
- Trip End Date, Date.
- Out of State, Checkbox.
- Destination State, Text, Length = 2.
- Department, Lookup, Related to = Department custom object.
- Save

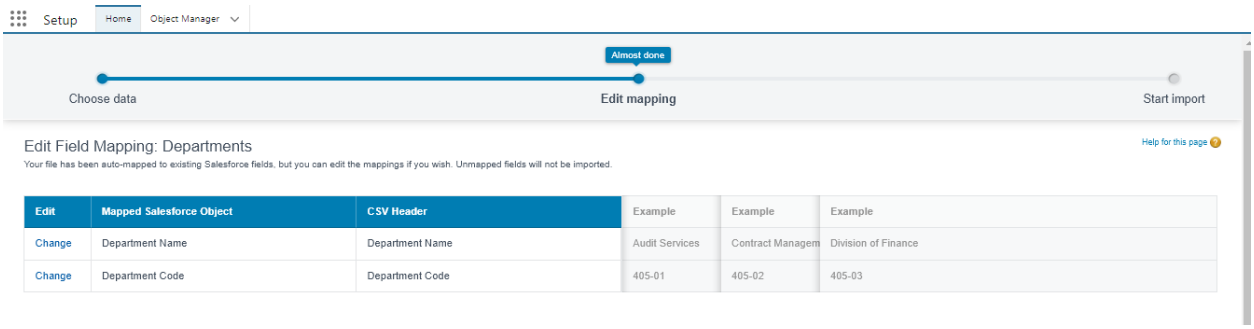


Step 5 : - Import the data store in Department.csv by using “Data Import wizard” tool.

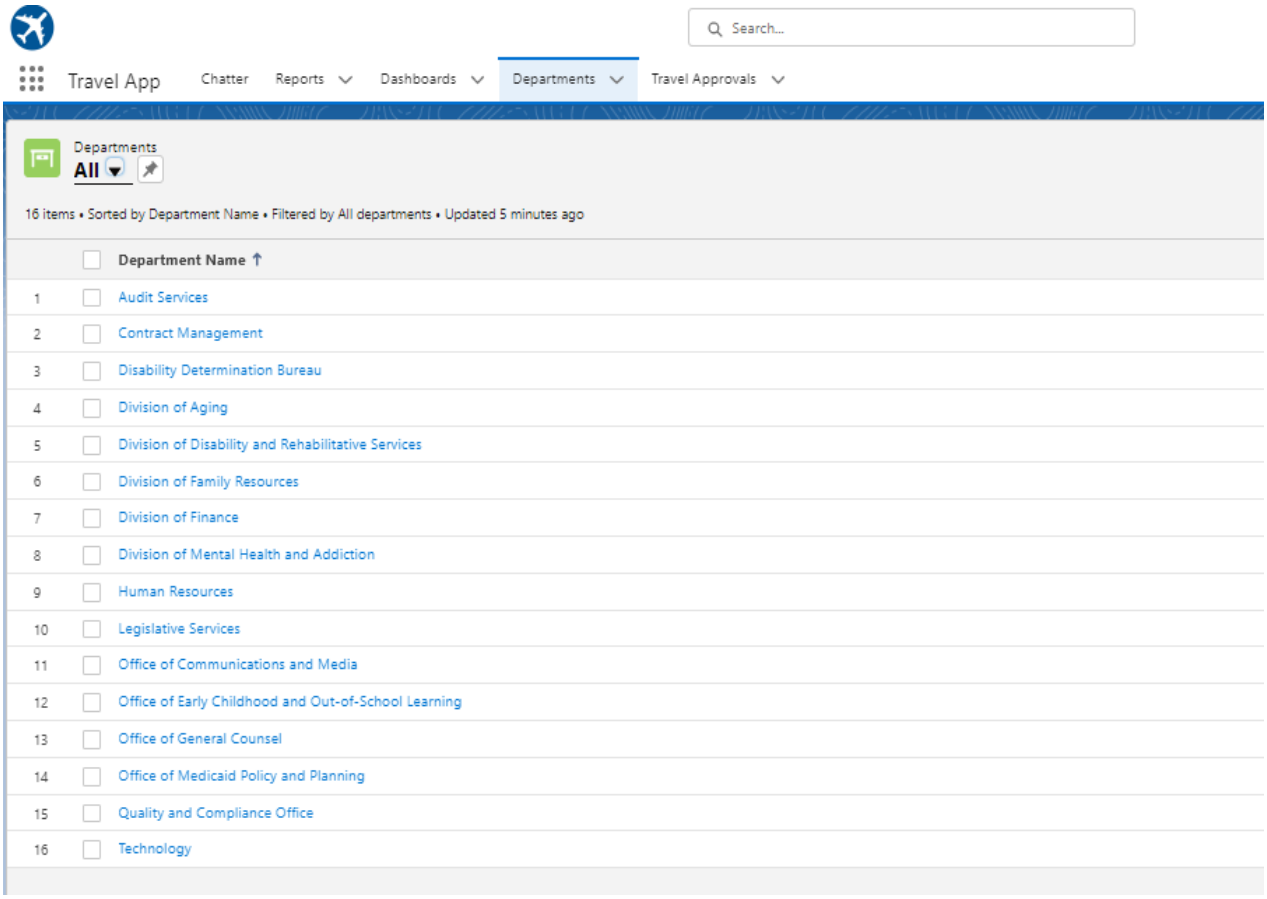
5(a) -



5(b) -



-**Test the App**, till yet what we have done.

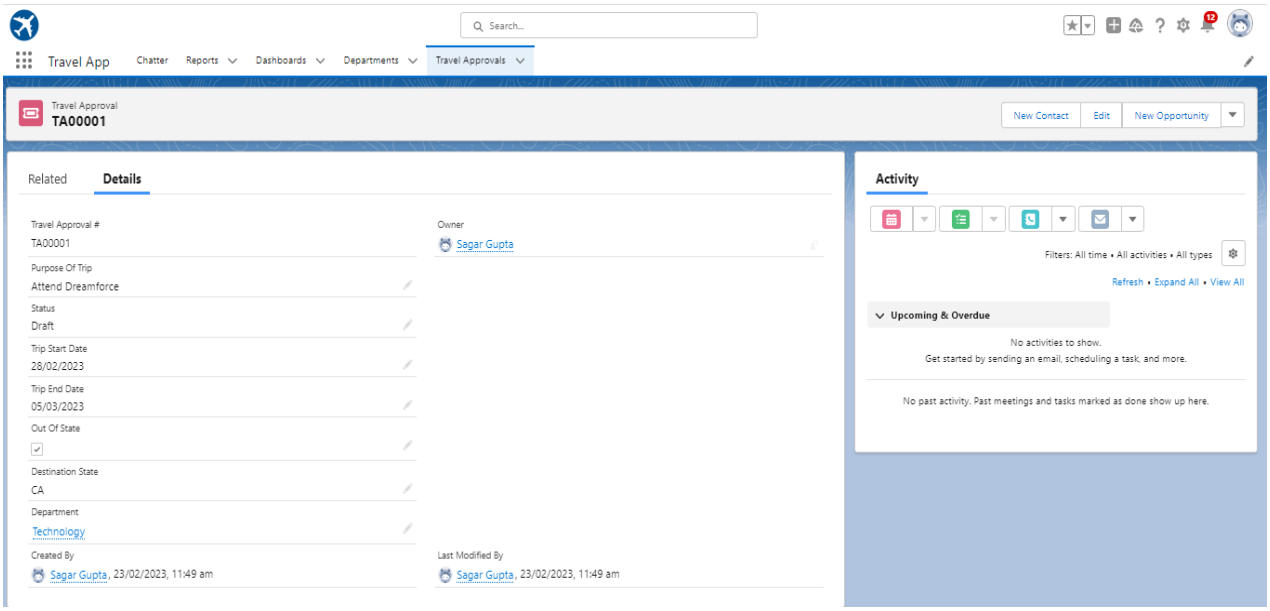


See the data is imported Successfully.

Note :- Before Importing the data make sure you clean the data otherwise some data will not be imported successfully.

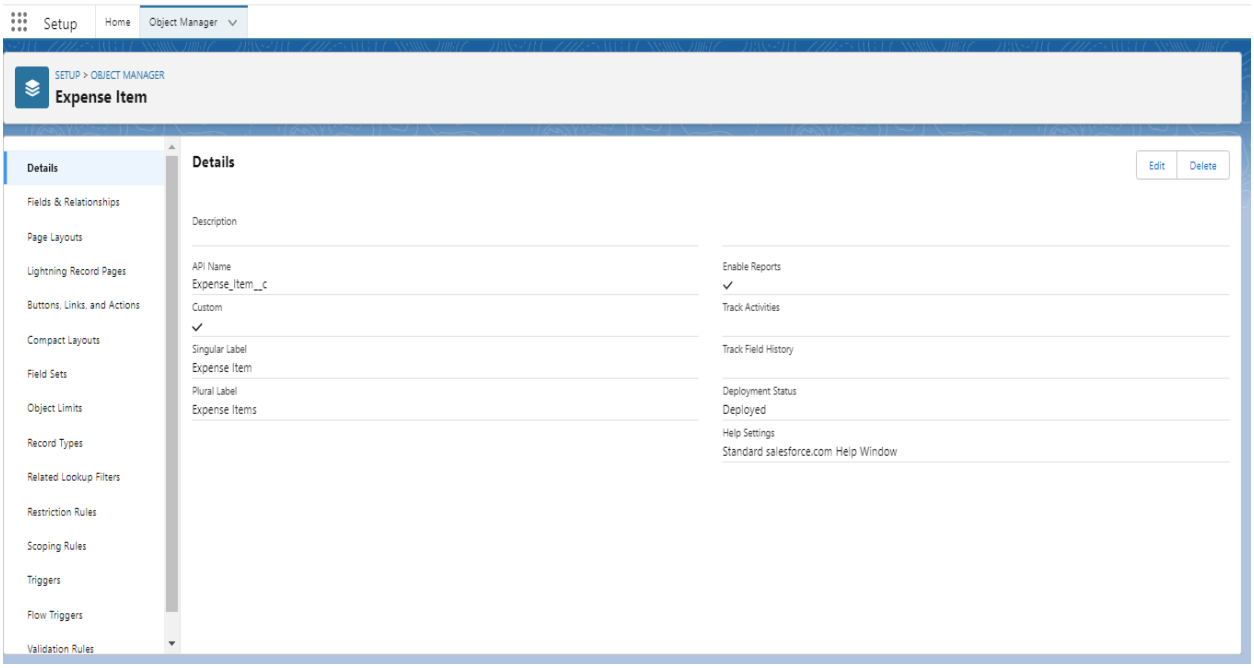
Exercise 2: -

Step 1: - Create Travel Approval Record.



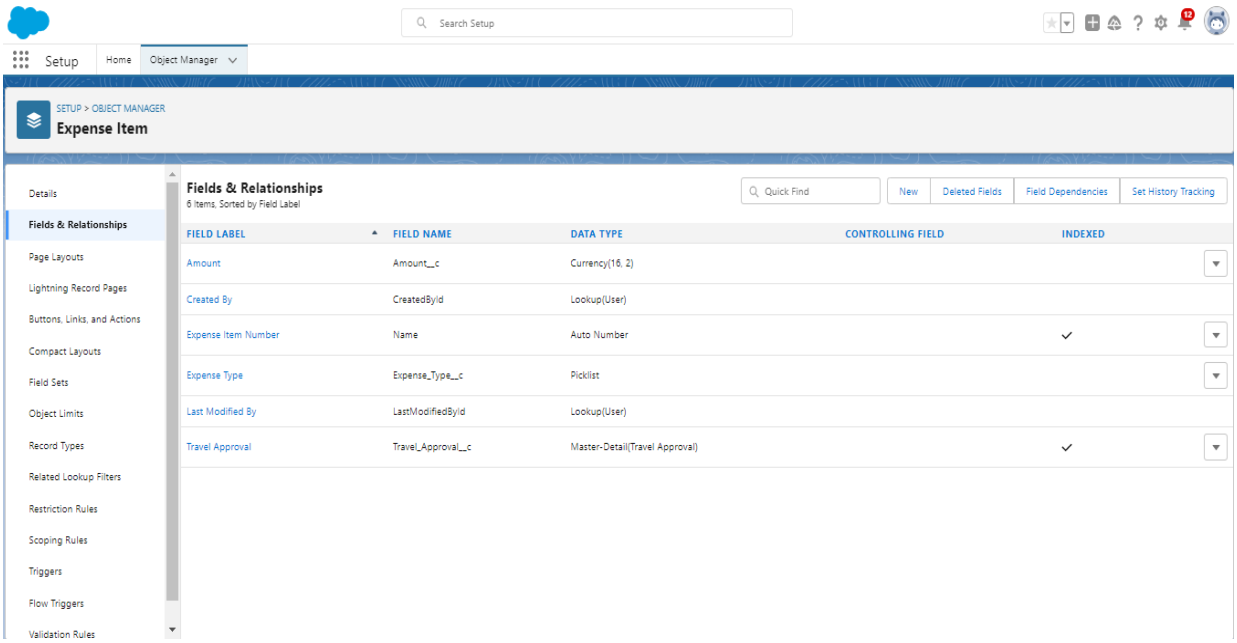
Step 2 : - Create a Expense Items custom object.

Setup->Object Manager->Create Custom object (From top left dropdown button)



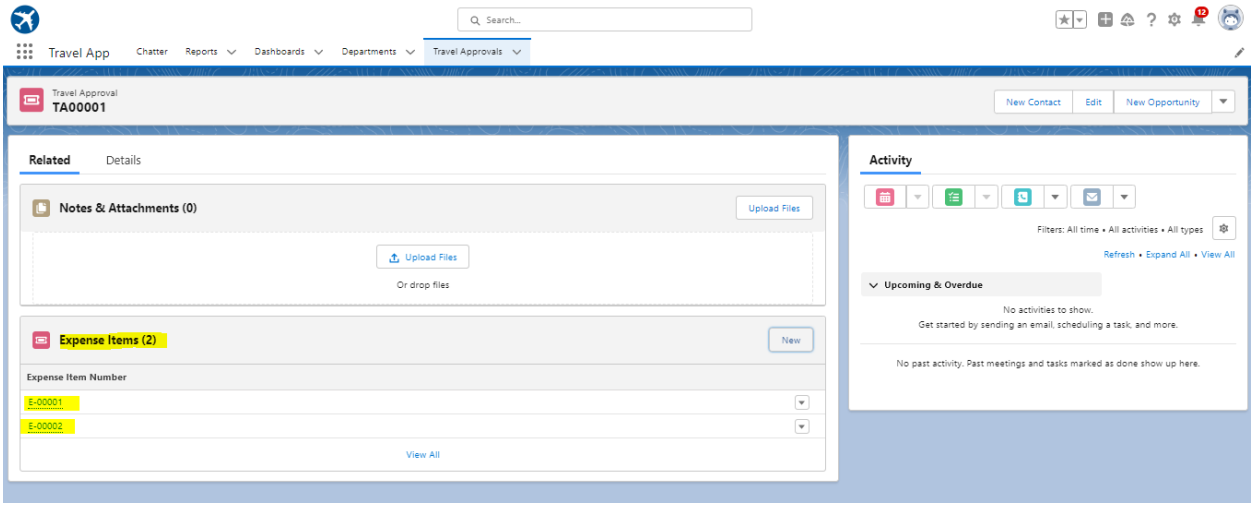
Step 3 :-Create the following custom fields on Expense Item Object.

- Amount, Length = 16, Type = Currency, Decimal = 2, Required = True.
- Expense Type, Type = Picklist, Values = Airfare, Hotel, Rental Car, Meals, Other, Required = True.
- Travel Approval, Type = Master-Detail, Related To – Travel Approval.



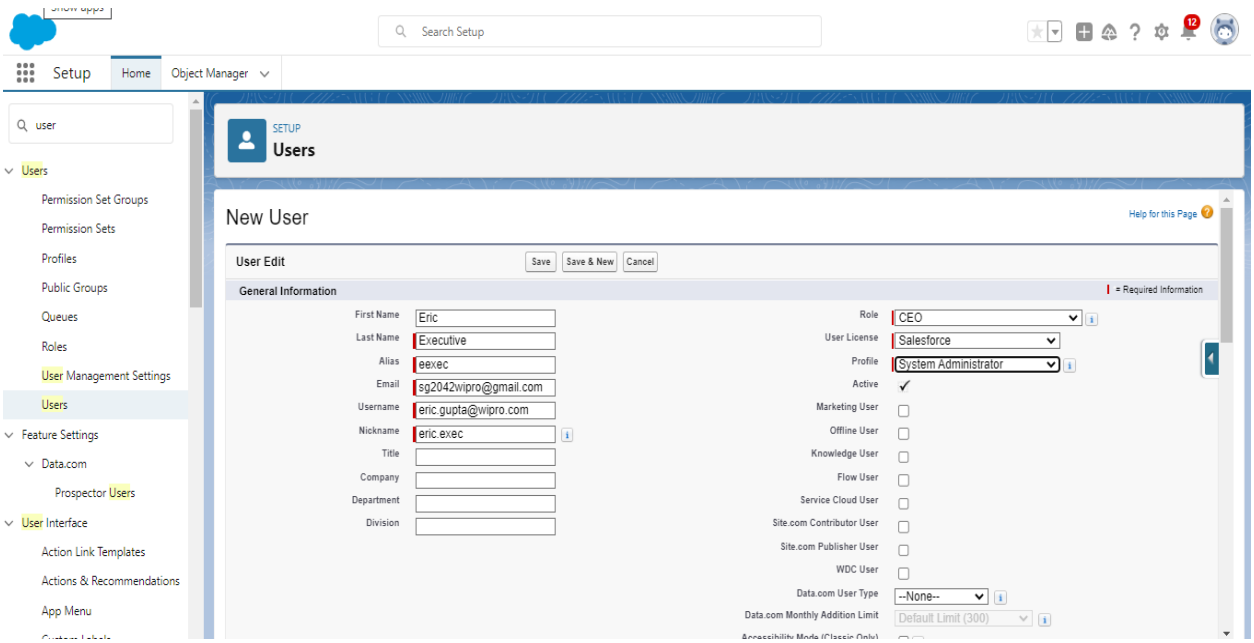
Step 4 :-Create the Expense Items .

- Amount = 870, Expense Type = “Hotel”.
- Save.
- Amount = “450”, Expense Type = “Airfare”
- Save.

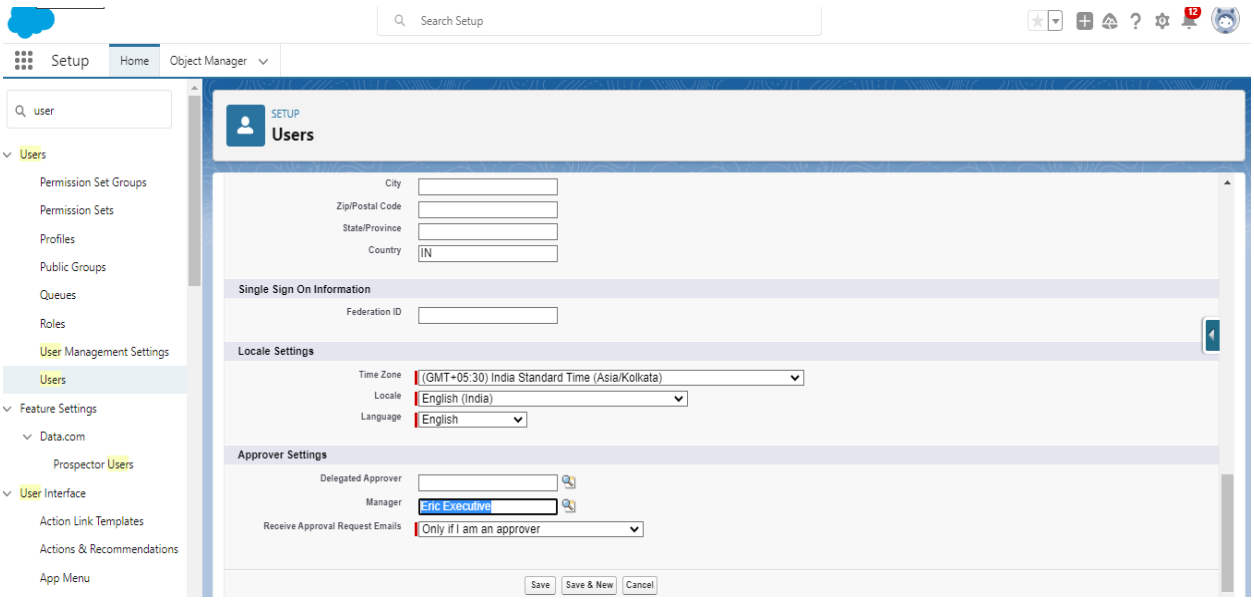


Step 5 :- Create a user.

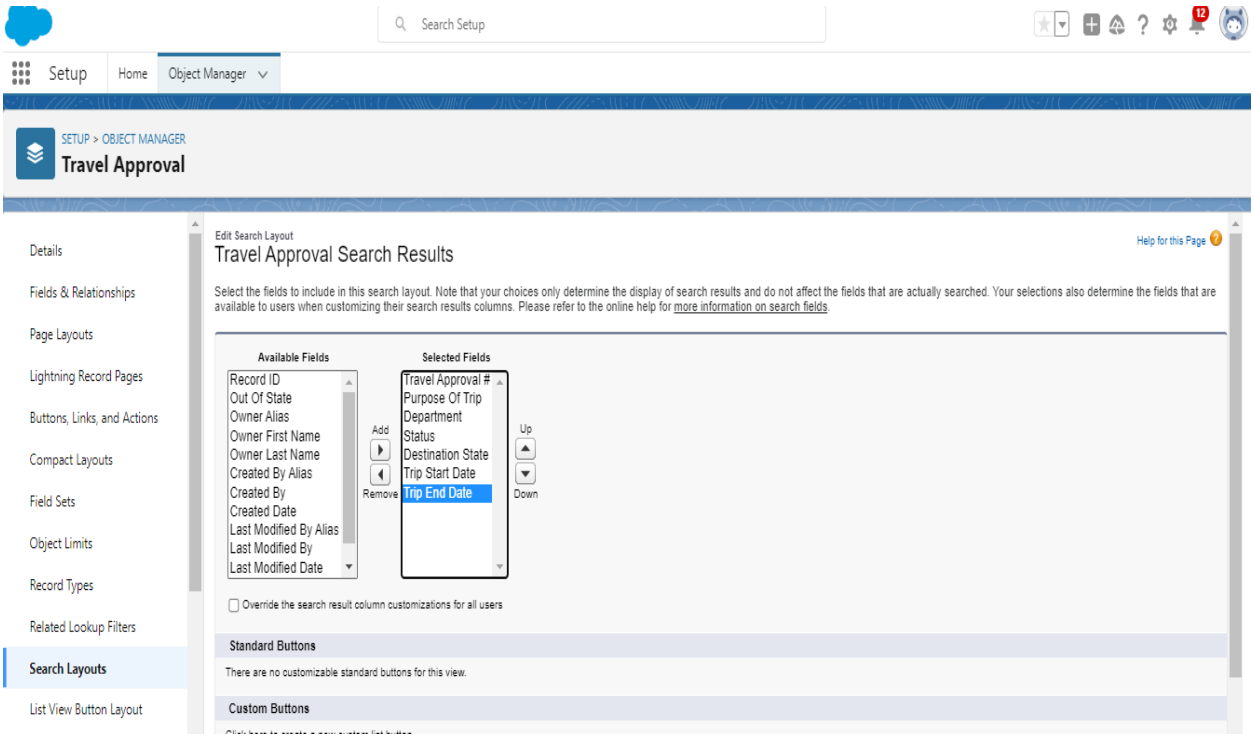
- First Name = “Eric”, Last Name = “Executive”, Email = “Use your own email”,
- Username Name = “Choose a Unique username”
- Role = “CEO”
- License = Salesforce.
- Profile = System Administrator.
- Save.



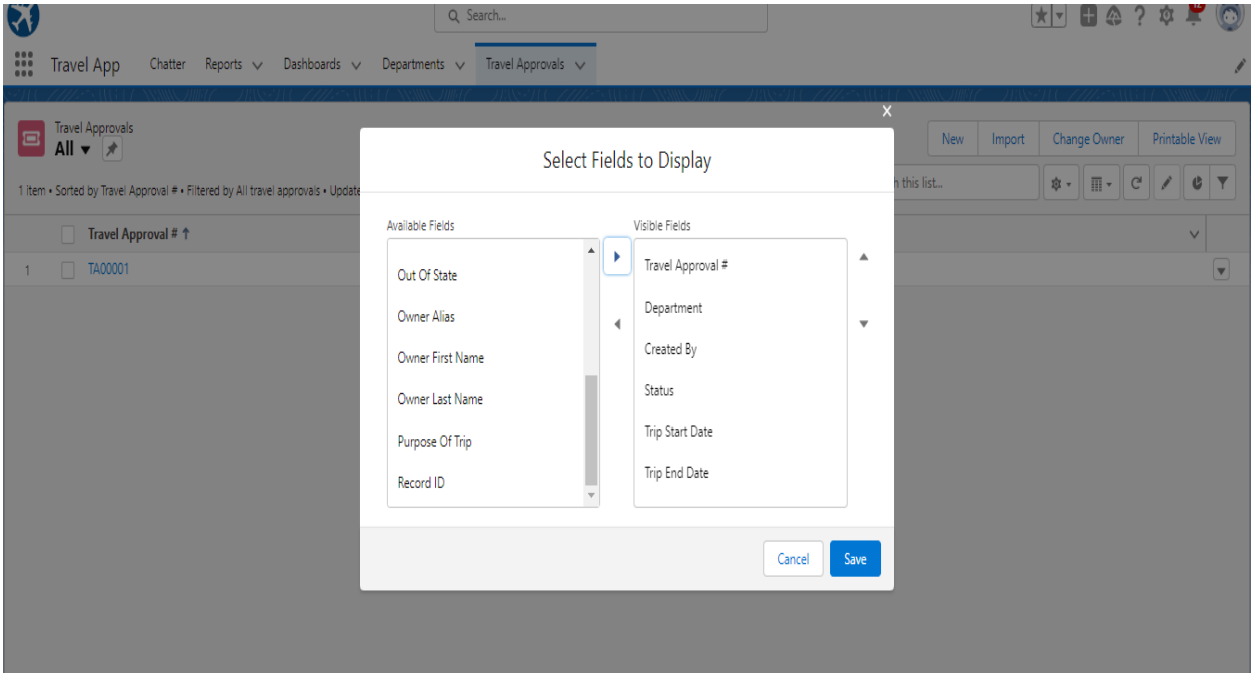
Step 6:- Add user Eric Executive as your manager as shown in the screen shot.



Step 7:- Customize the Travel Approval Default search layout as shown in the screen shot & Save.

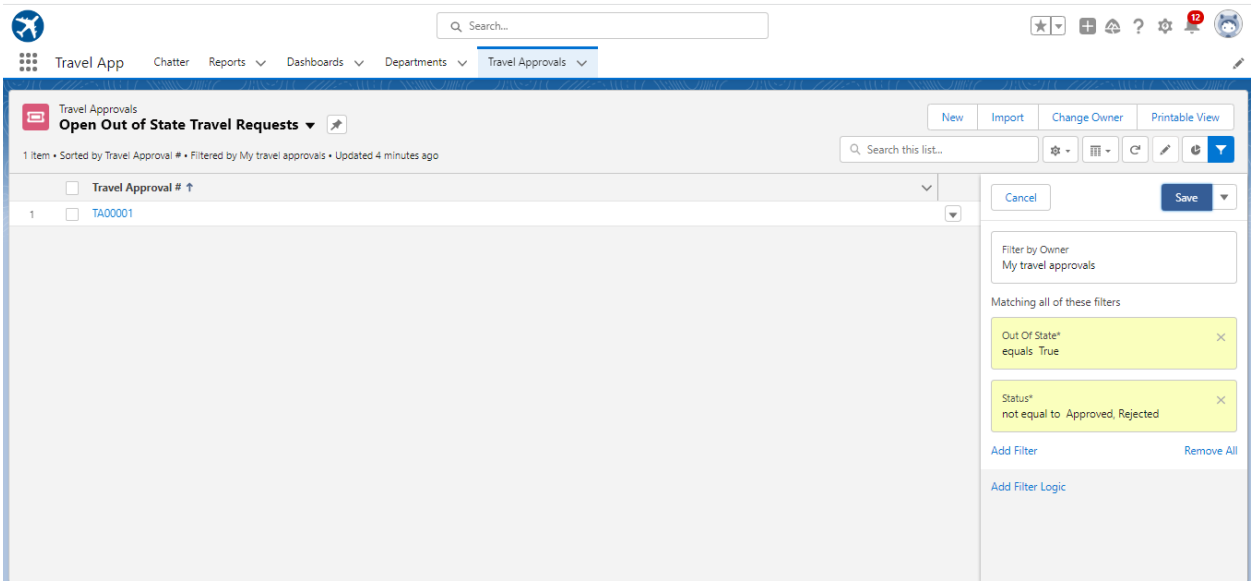


Step 8:- Select fields to display in the Travel Approval “All” List view, as shown in the screen shot & Save.

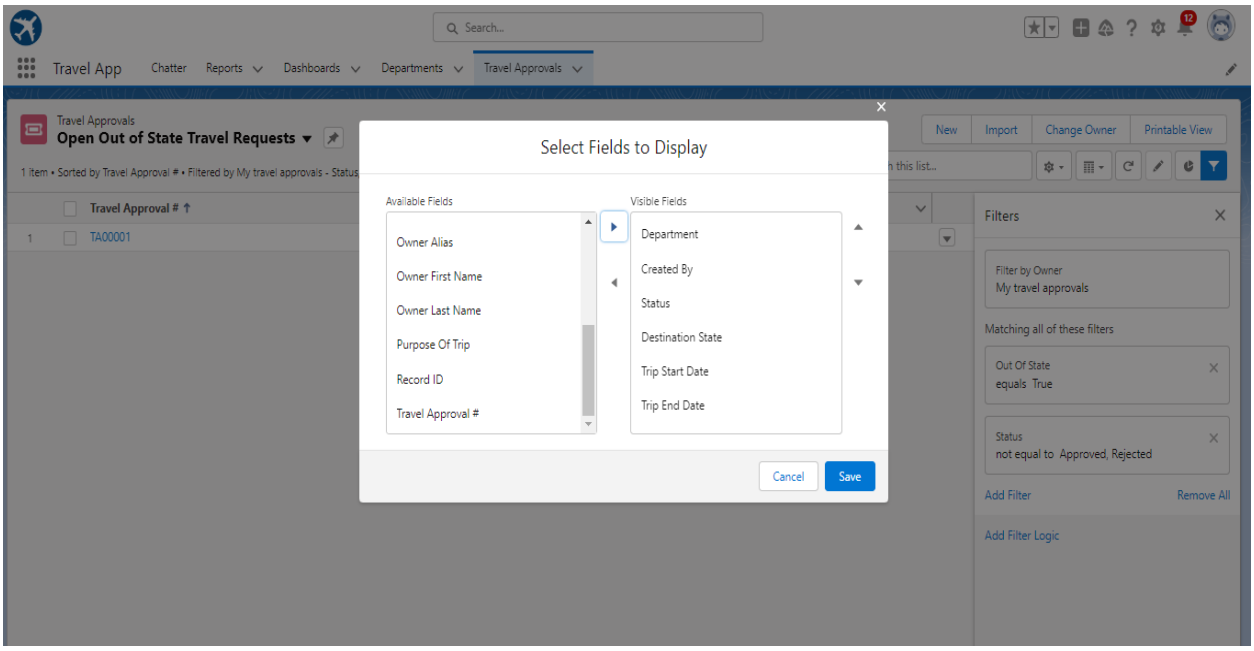


Step 9 :- Create Travel approval custom List View “Open Out of State Travel Requests” as shown in the screen shot & Save.

• All users should be able to see this list view.

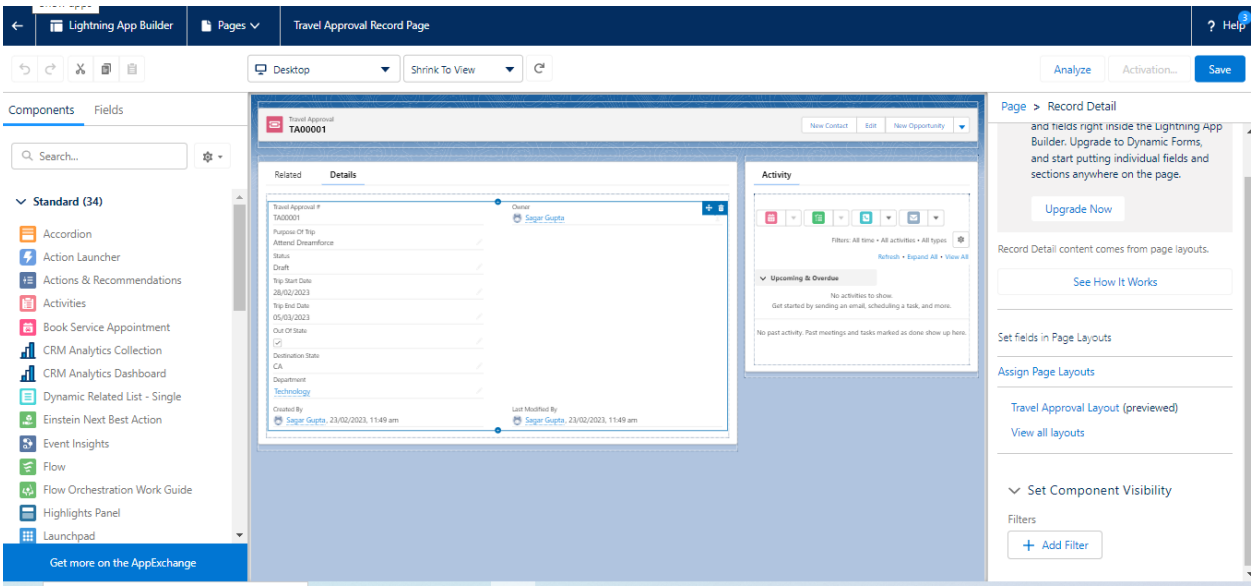


Step 10 :- Select fields to display in the Travel Approval “Open Out of State Travel Requests” List view, as shown in the screen shot & Save.



Step 11 :- (a) Customize the Travel Approval Page Layout as shown in the screen shot.

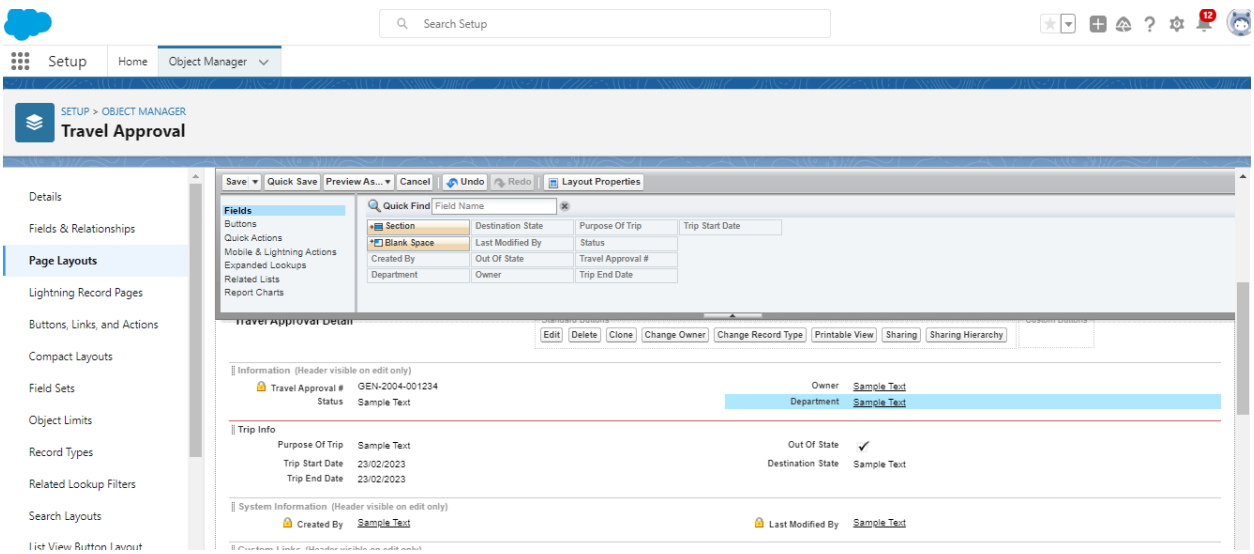
- In the travel app, click travel approval tab and open TA-00001
- Then click Gear button and select “Edit Button”
- It will open as shown as screenshots.



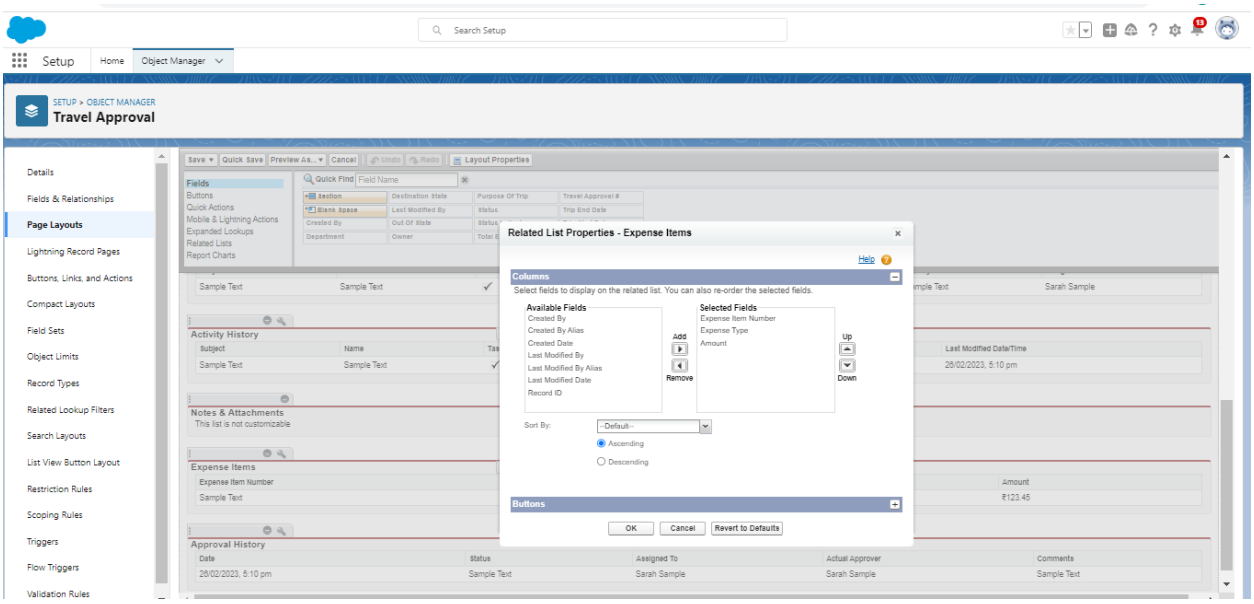
(b) :- Travel Approval Page Layout should look as shown in the screen shot.

Note: Add a section to the page layout called “Trip Info” and add the fields as shown.

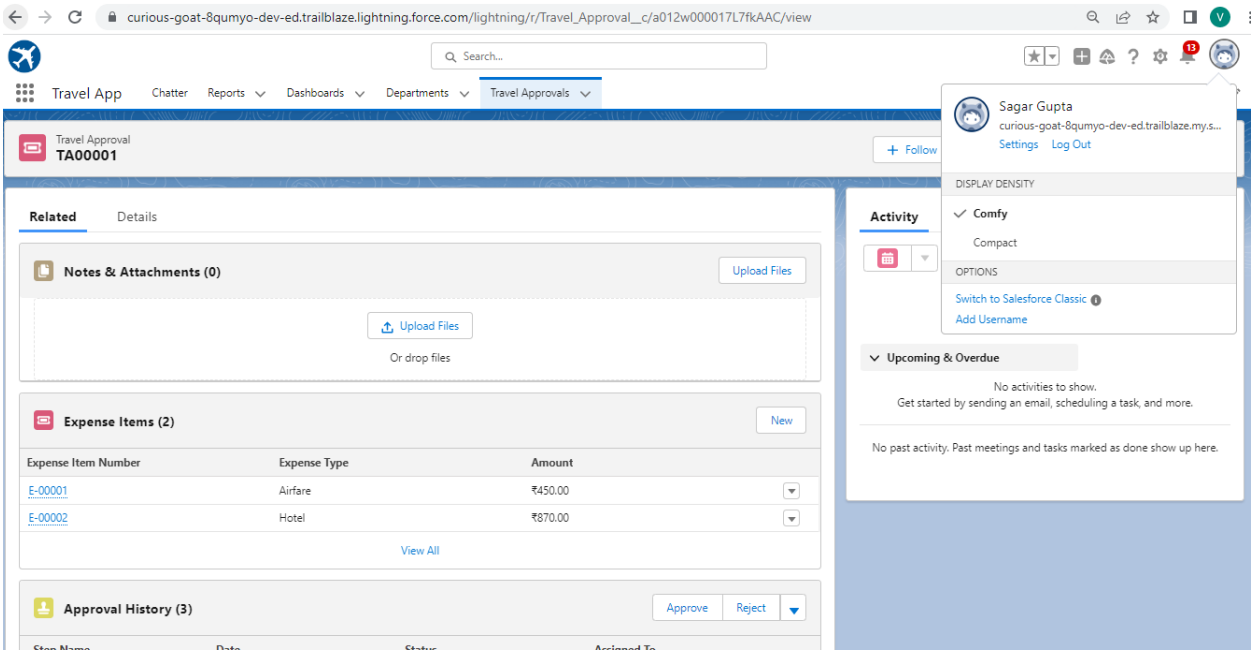
Click Save.



Step 12 :- (a) Customize the Expense Item Related List under the Travel Approval page layout as shown in the screen shots.

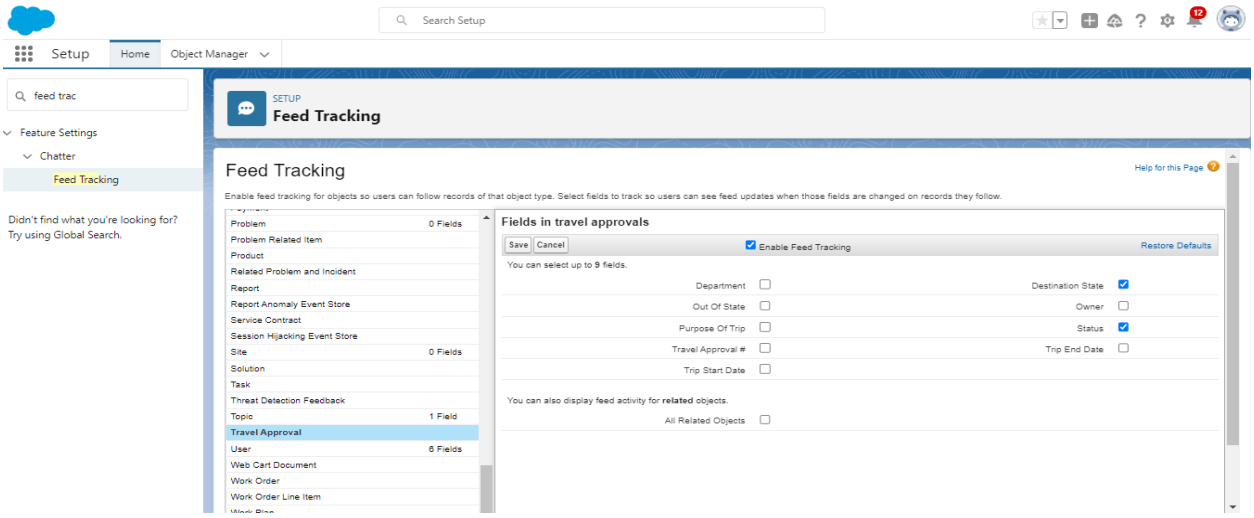


(b)- The Travel Approval App should look as shown in the screenshot.



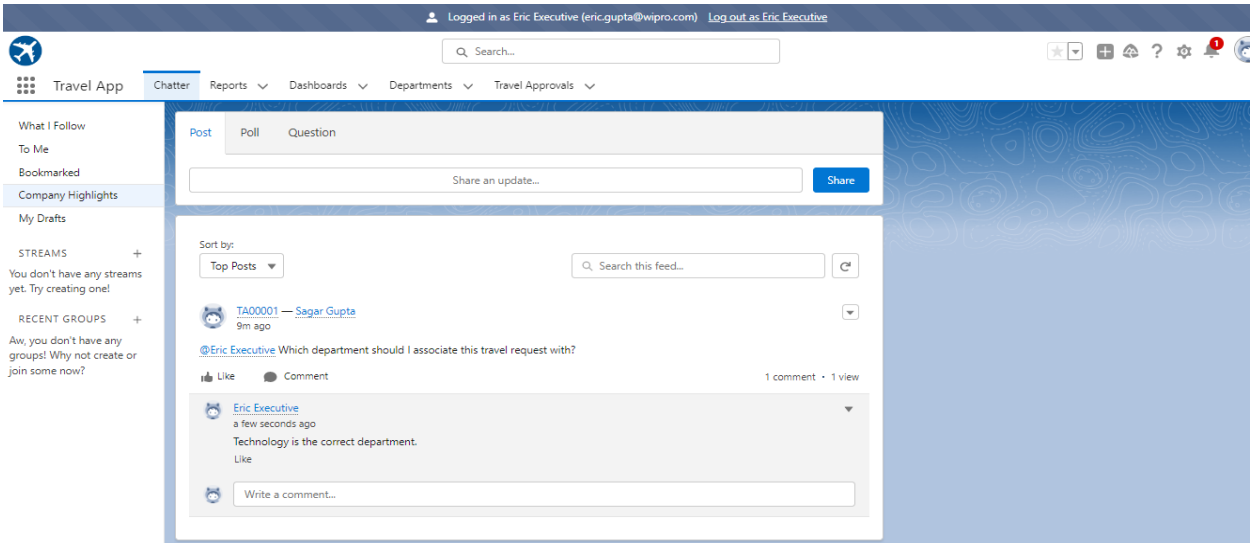
Step 13 :- (a)

- Enable Chatter on the Travel Approval Object.
- Enable “Feed Tracking” for Travel ApprovalObject.
- Select these 2 fields:
- Destination State
- Status
- Save.



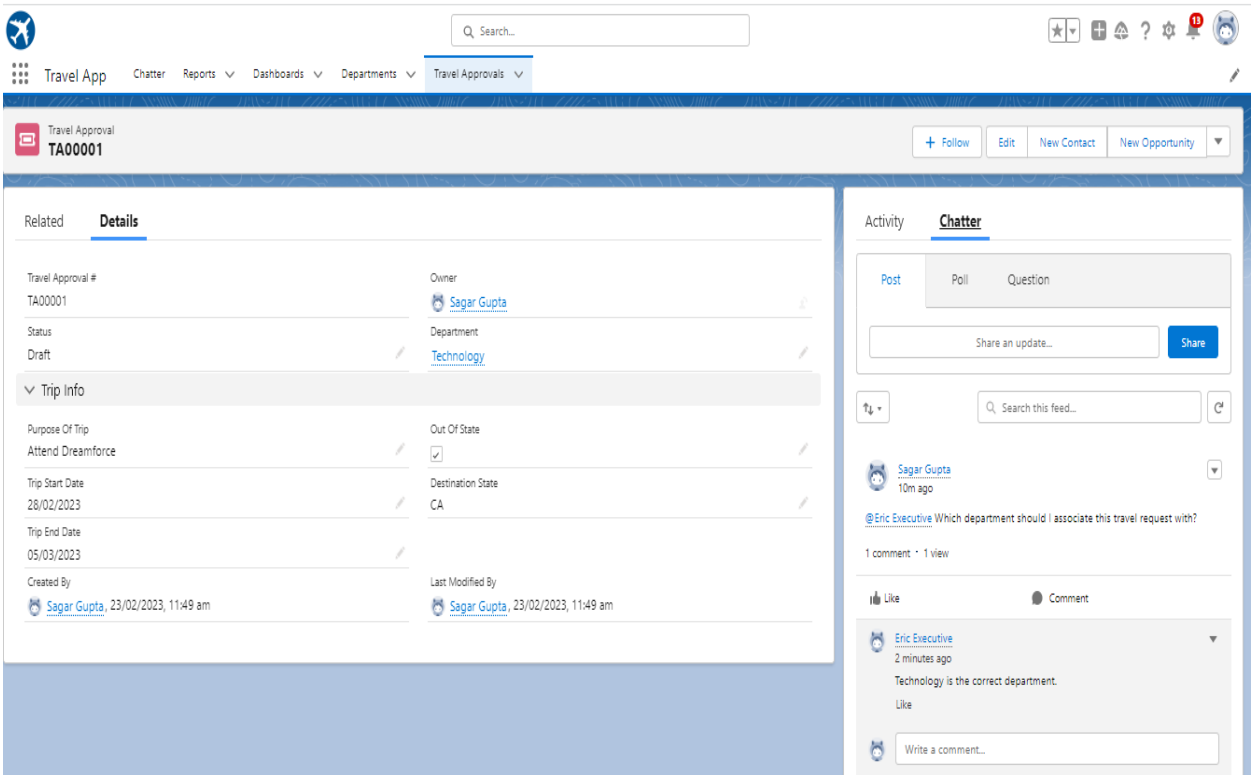
(b) Test Collaboration **:-**

• Open a Travel Approval record.
• Click on Chatter Tab.
• Share a post: Which Department should I associate this travel request with?
• Mention user Eric Executive on the Post using @.
• Note: Login in as Eric and reply to the email, saying: “Technology is the correct department”.
• Note: Enable “Administrator can Log in as any user.



Test the App , Let see how its look like

- The Travel Approval App should look as shown in the screenshot.



After completing the exercise 2 you will see the user can post their query by using chatter and other can reply to that query.

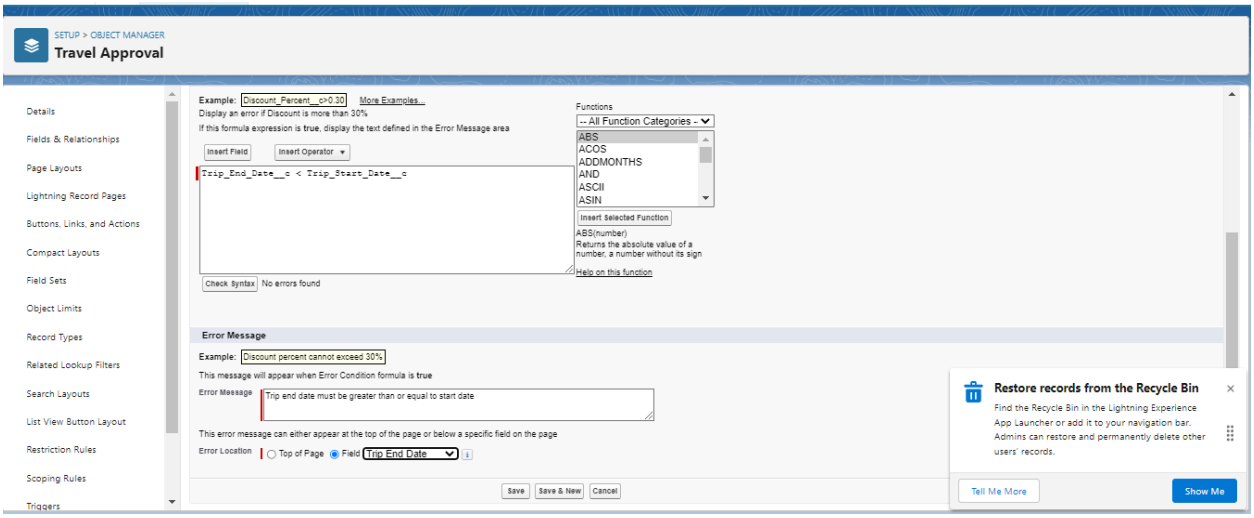
Now, Here Our module 1 is completed lets move to our next module 2.

Module – 2

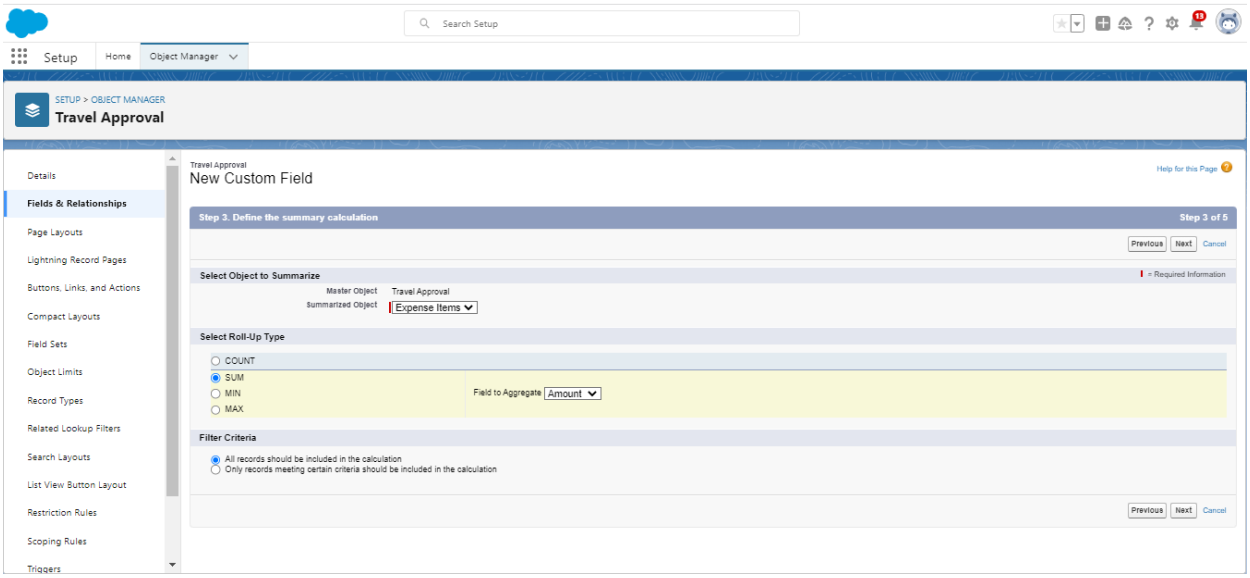
Exercise 1: -

Step 1:- Create Validation Rule.

Business Logic: Trip end date must always be greater than (>=) the trip start date.

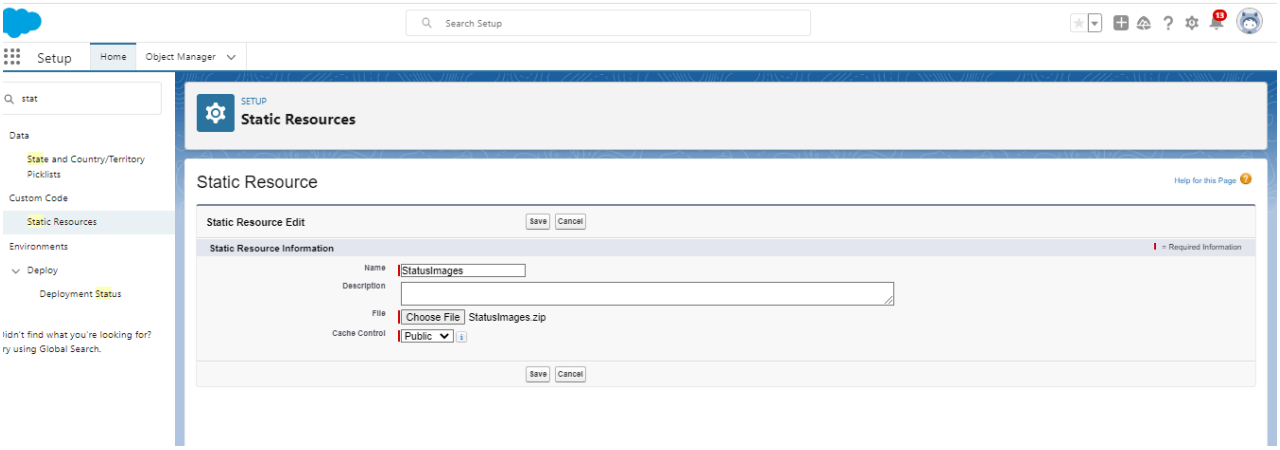


Step 2 :- Create a Roll-Up Summary Field on Travel Approval object.



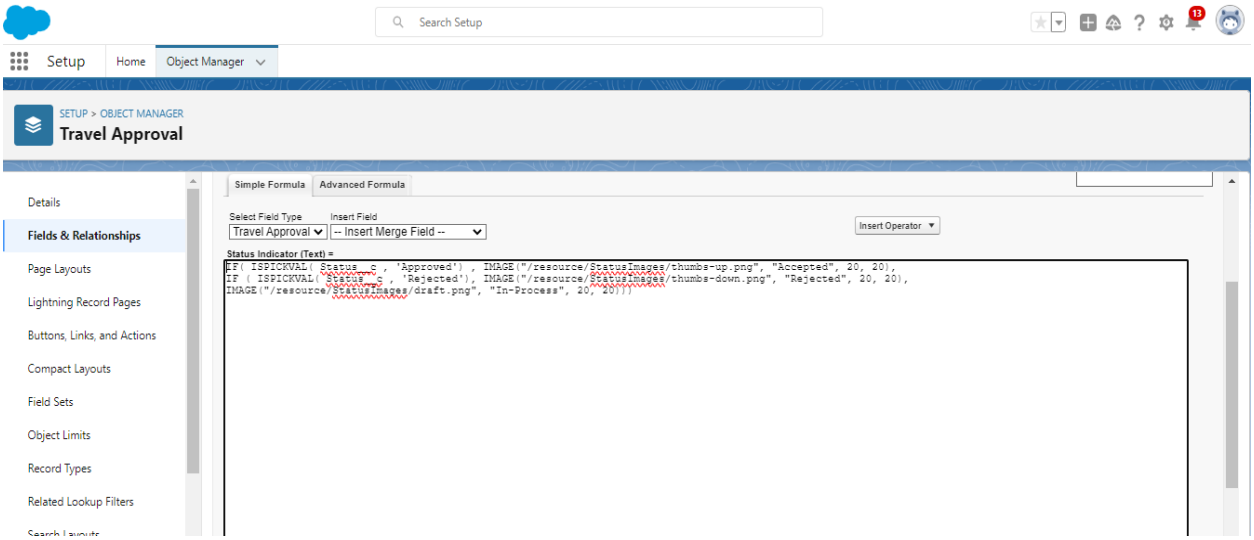
Step 3 :- Business Logic: Create a field that shows a visual indicator based on the value of the Status field

- Setup | Custom Code | Static Resource | New



Step 4 :- Create a Formula field on the Travel Approval object to show an image based on the Status field.

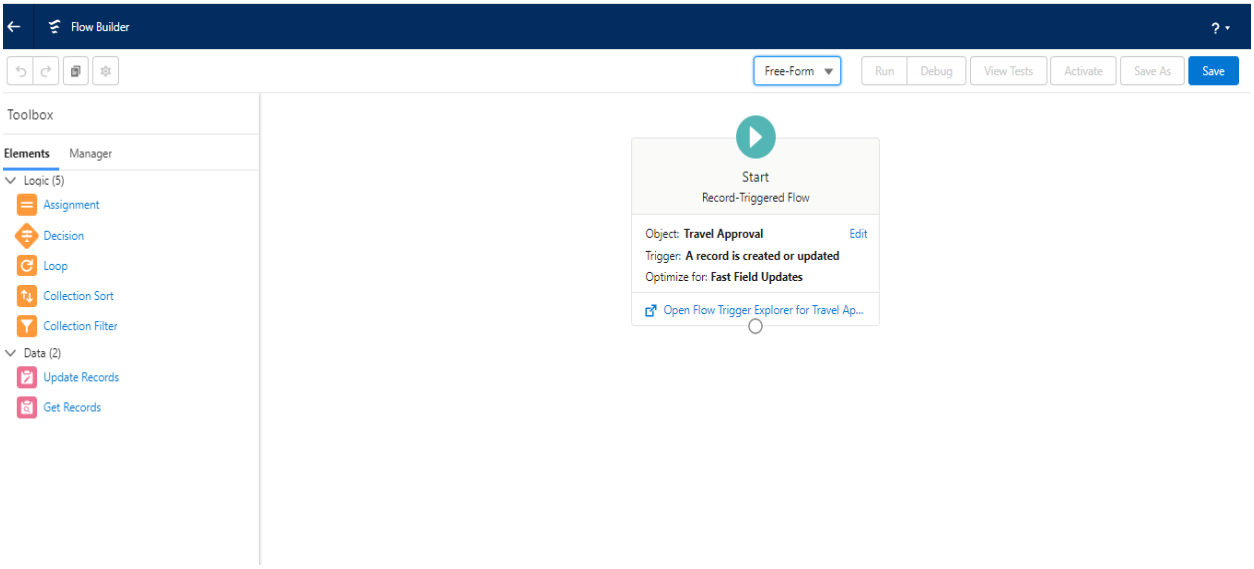
- Field Label: Status Indicator
- Formula Return Type = Text
- Formula: IF(ISPICKVAL(Status c , 'Approved') , IMAGE("/resource/StatusImages/thumbs-up.png", "Accepted", 20, 20), IF (ISPICKVAL(Status c , 'Rejected'), IMAGE("/resource/StatusImages/thumbs-down.png", "Rejected", 20, 20), IMAGE("/resource/StatusImages/draft.png", "In-Process", 20, 20)))
- Save



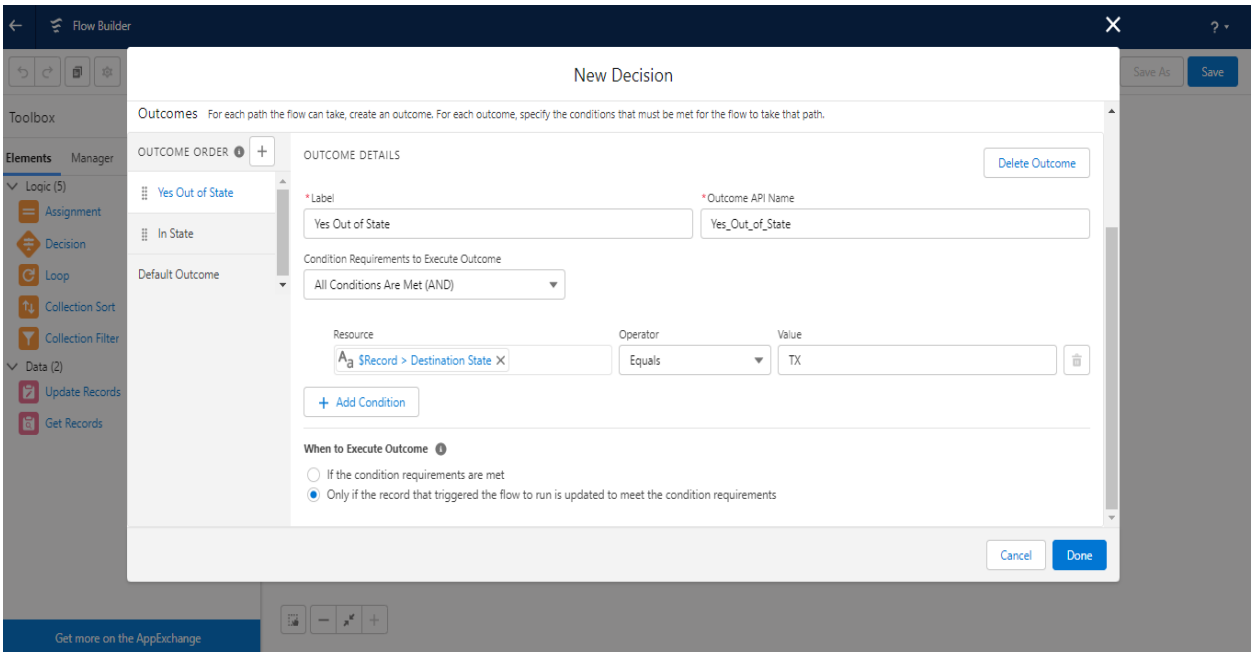
Step 5 :-

(a).Create a Record – Triggered Flow.

• Flow should look like this:

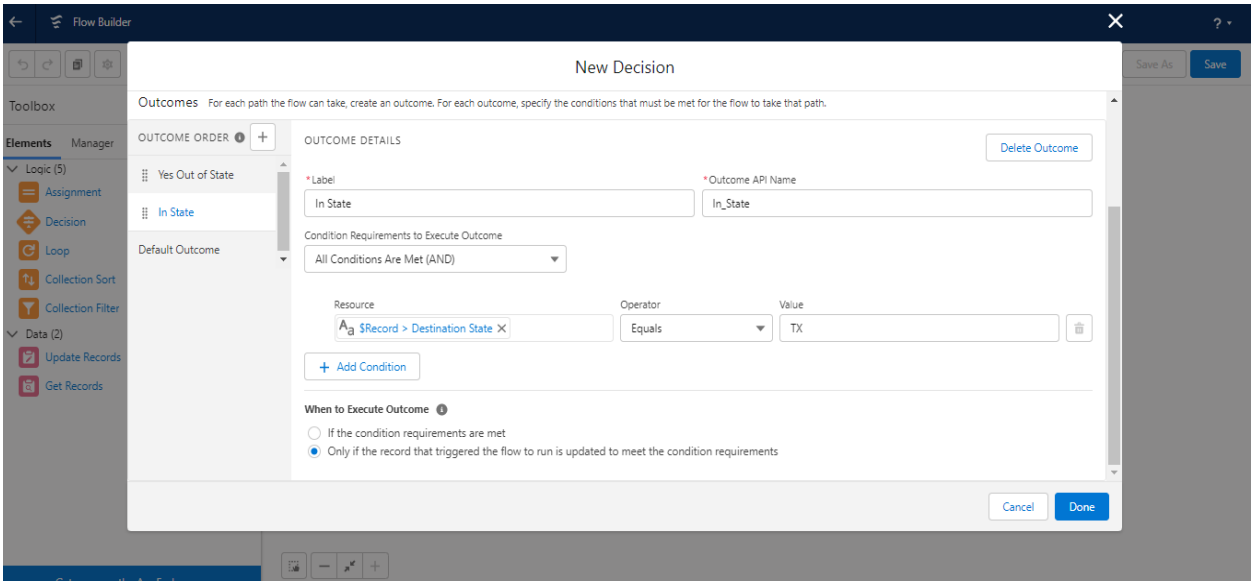


(b). Add a Decision Element to the Flow.

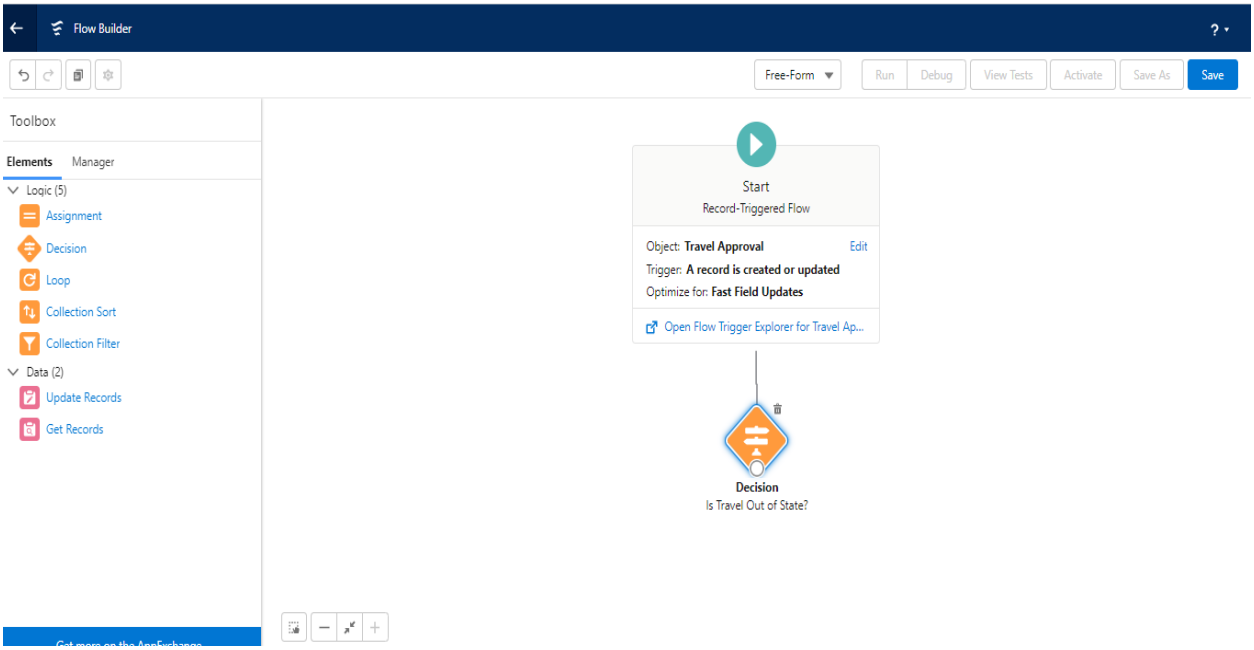


(c). Next to Outcome Order click the + button to add another outcome.

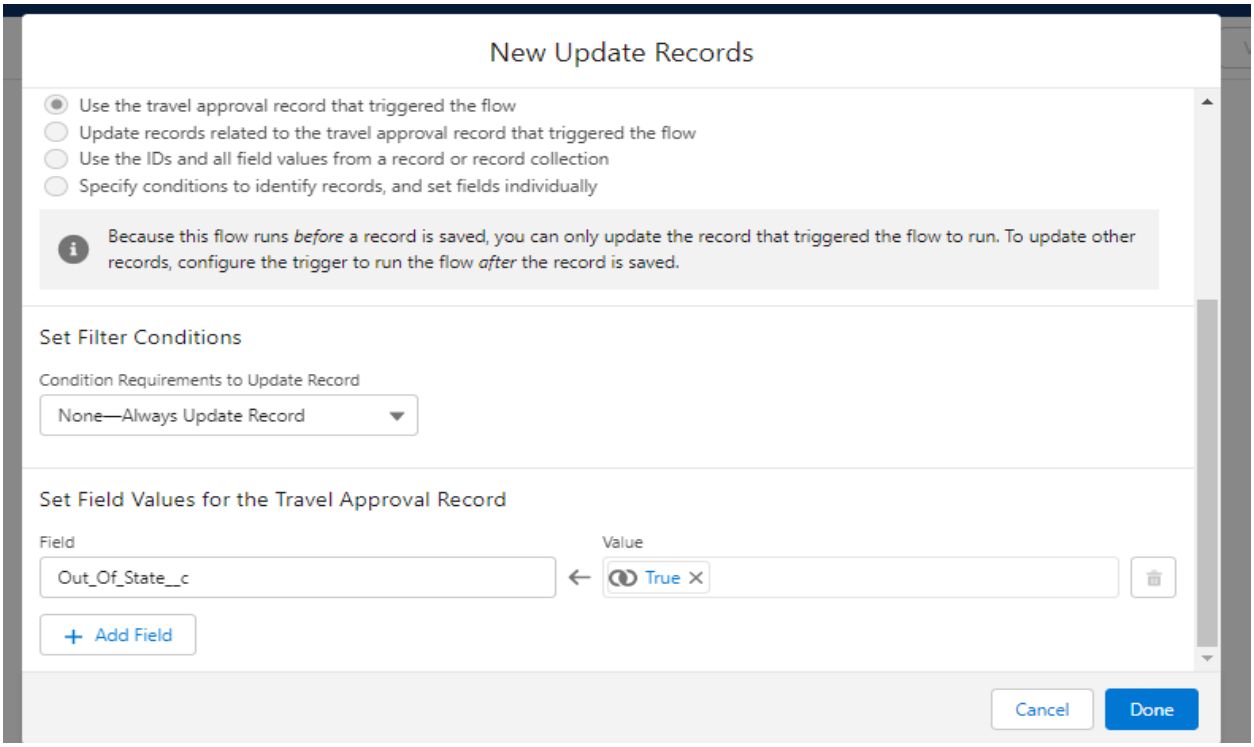
• Flow should look like this:



(d). Connect the Start Flow element to the Decision element.



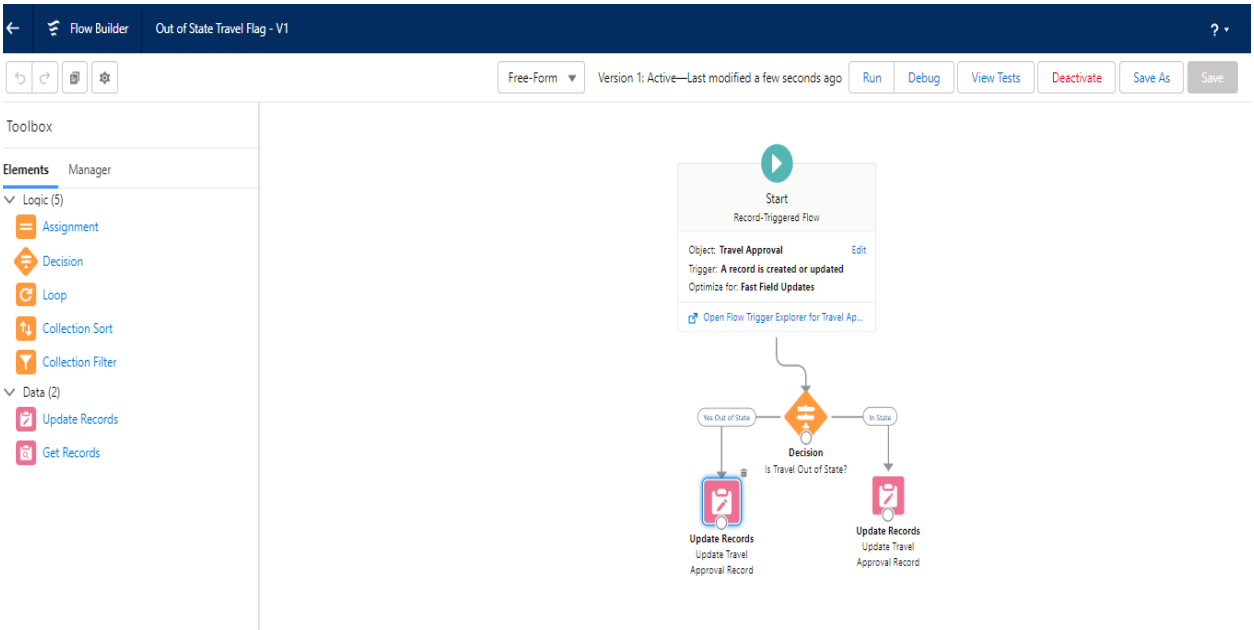
(e). Create an Action for the Flow Using Update Records Elements.



(f). Test The Flow Screen

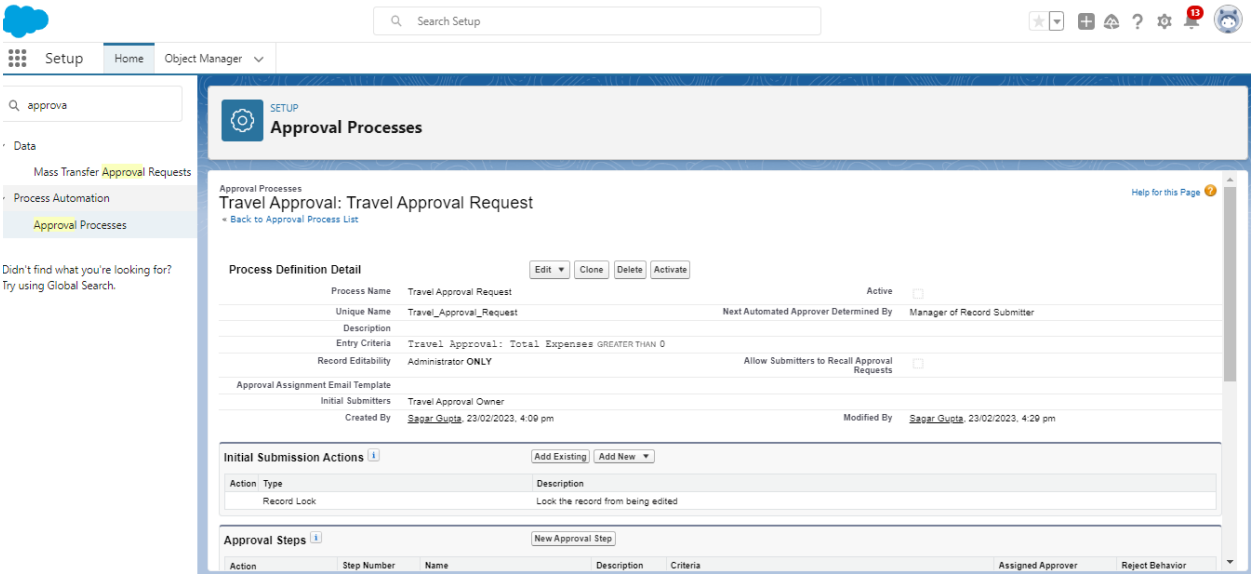
• Drag the white circle from the Decision Node to the Update Records you just created and select the decision In State | Done.

• Flow screen should look this:



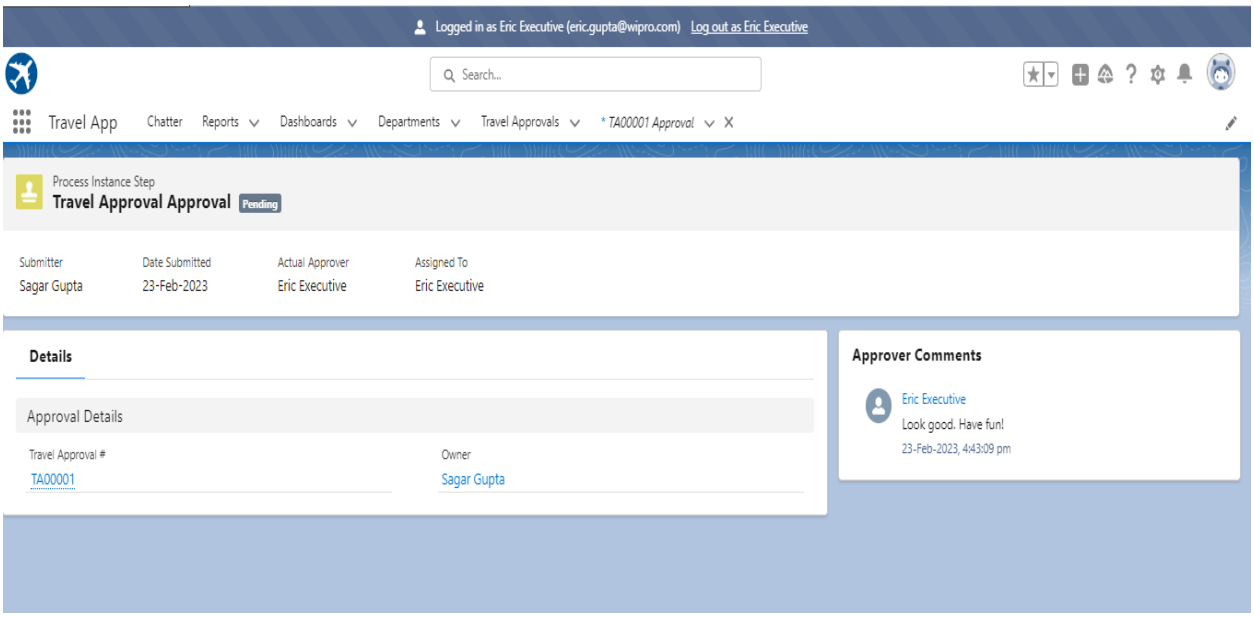
Step 6 :-

(a). Create an Approval Process to send Travel approvals to Manager or Travel coordinator.



(b). Test The Approval Process

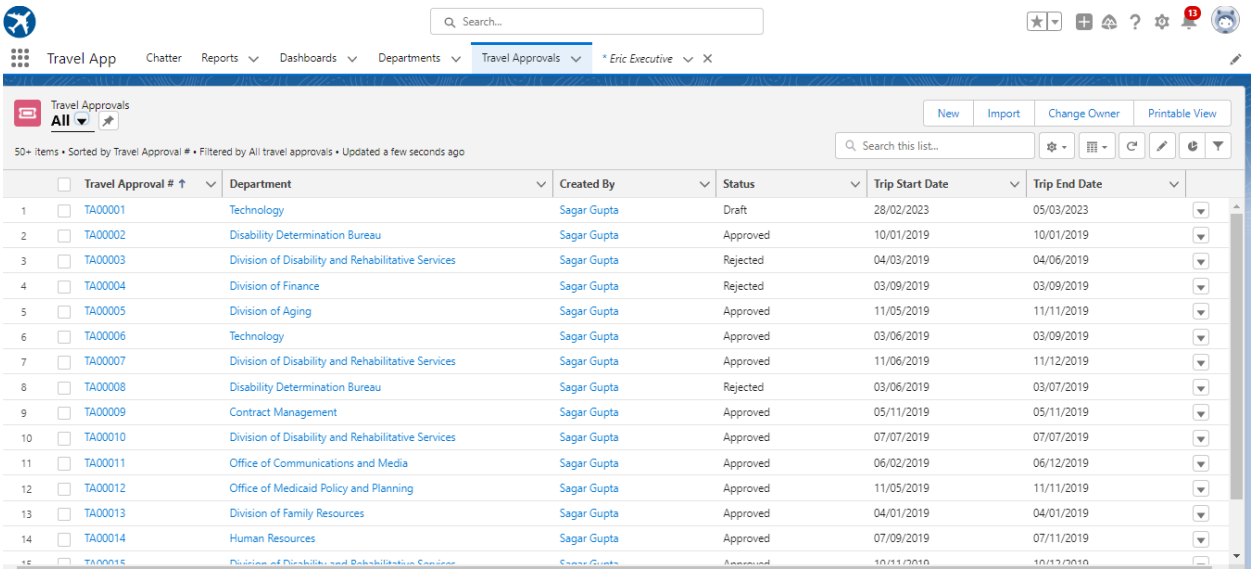
- Create few Travel Approval records and Submit for Approval.
- Login as Eric, approve and reject the records randomly as shown in the screen shot:



Exercise 2: -

Step 1 :- Use Data Import Wizard to import Travel Approval records.

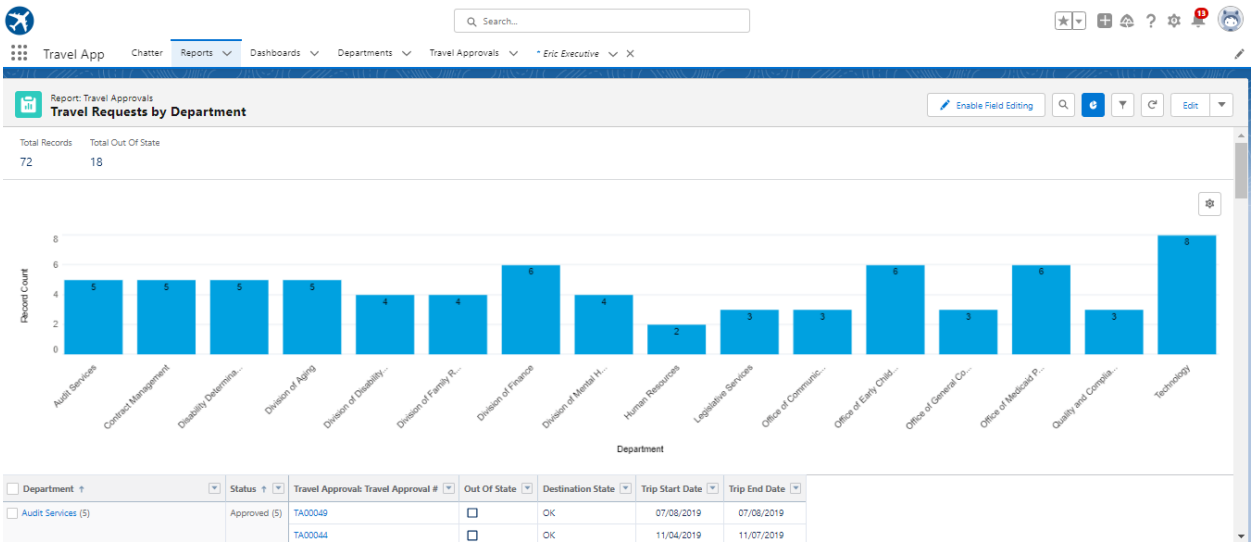
- Test The App
  • Make sure all the Travel Approval records are successfully imported, check the screenshot:



	Travel Approval #	Department	Created By	Status	Trip Start Date	Trip End Date
1	TA00001	Technology	Sagar Gupta	Draft	28/02/2023	05/03/2023
2	TA00002	Disability Determination Bureau	Sagar Gupta	Approved	10/01/2019	10/01/2019
3	TA00003	Division of Disability and Rehabilitative Services	Sagar Gupta	Rejected	04/03/2019	04/06/2019
4	TA00004	Division of Finance	Sagar Gupta	Rejected	03/09/2019	03/09/2019
5	TA00005	Division of Aging	Sagar Gupta	Approved	11/05/2019	11/11/2019
6	TA00006	Technology	Sagar Gupta	Approved	03/06/2019	03/09/2019
7	TA00007	Division of Disability and Rehabilitative Services	Sagar Gupta	Approved	11/06/2019	11/12/2019
8	TA00008	Disability Determination Bureau	Sagar Gupta	Rejected	03/06/2019	03/07/2019
9	TA00009	Contract Management	Sagar Gupta	Approved	05/11/2019	05/11/2019
10	TA00010	Division of Disability and Rehabilitative Services	Sagar Gupta	Approved	07/07/2019	07/07/2019
11	TA00011	Office of Communications and Media	Sagar Gupta	Approved	06/02/2019	06/12/2019
12	TA00012	Office of Medicaid Policy and Planning	Sagar Gupta	Approved	11/05/2019	11/11/2019
13	TA00013	Division of Family Resources	Sagar Gupta	Approved	04/01/2019	04/01/2019
14	TA00014	Human Resources	Sagar Gupta	Approved	07/09/2019	07/11/2019

Step 2 :- Create a Travel Requests by Department Report.

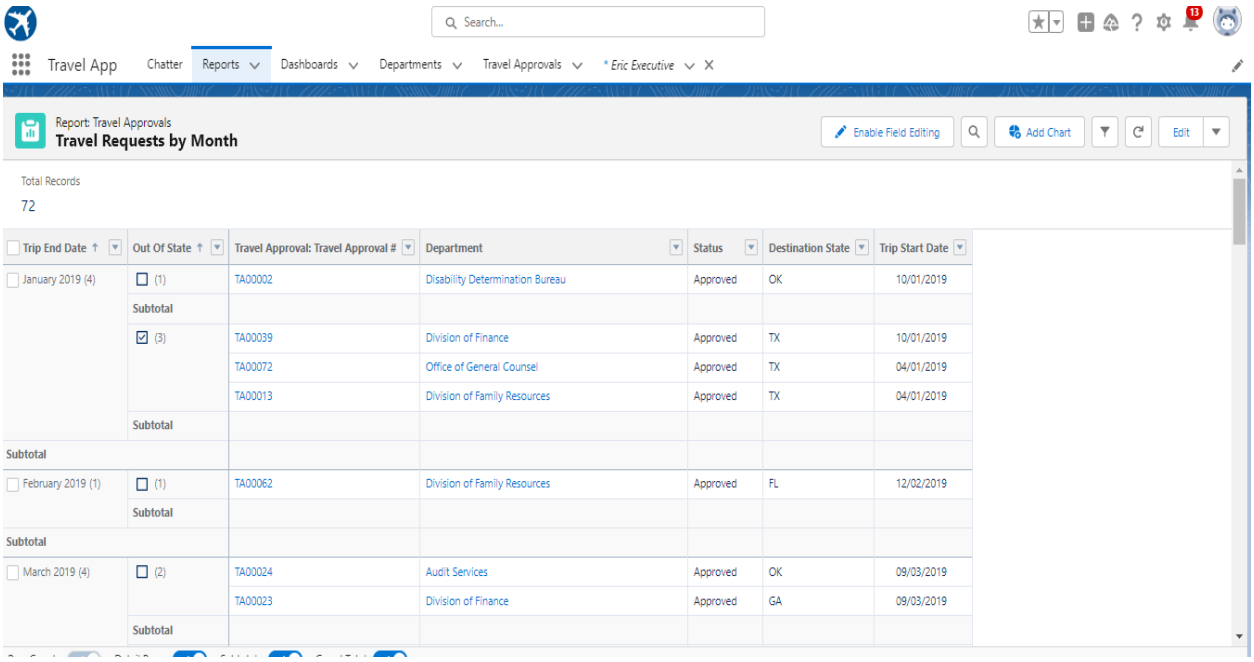
- Test The Report
- Report might look as per the screen shot:



Department	Status	Travel Approval: Travel Approval #	Out Of State	Destination State	Trip Start Date	Trip End Date
Audit Services (5)	Approved (5)	TA00049	☐	OK	07/08/2019	07/08/2019
		TA00044	☐	OK	11/04/2019	11/07/2019

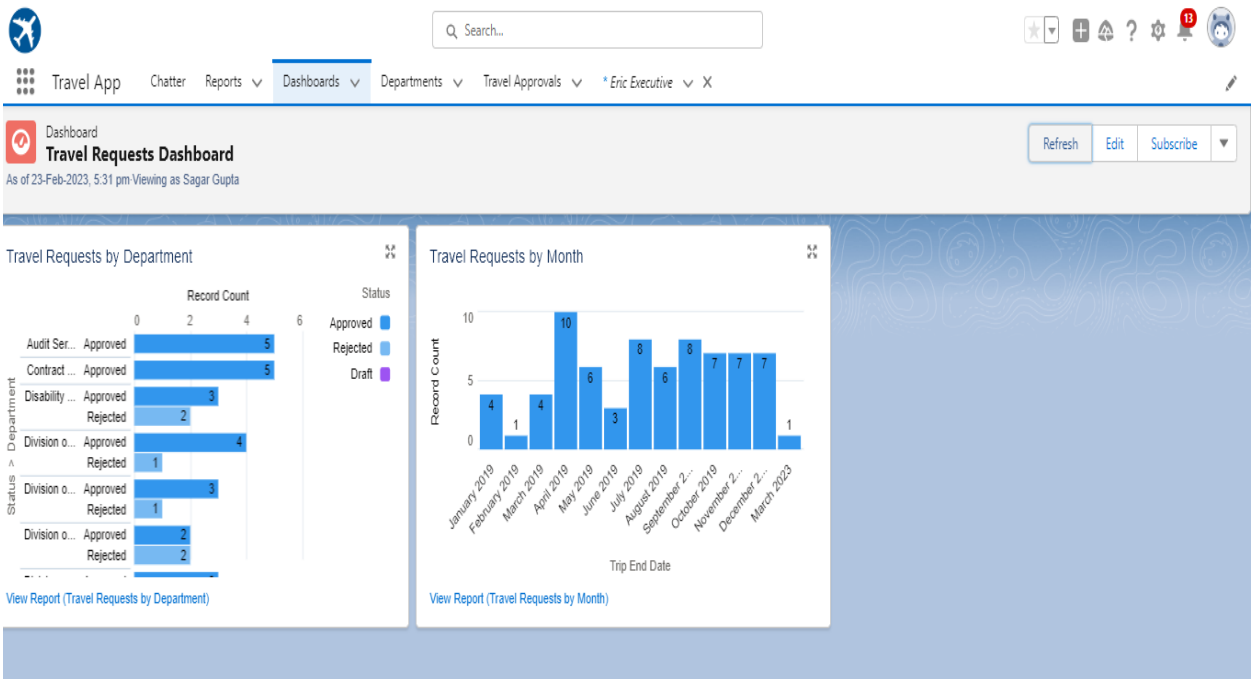
Step 3 :- Create a Travel Requests by Month Report

- Test The Report
- Report might look as per the screen shot:



Step 4 :- Create a Travel Approvals Dashboard.

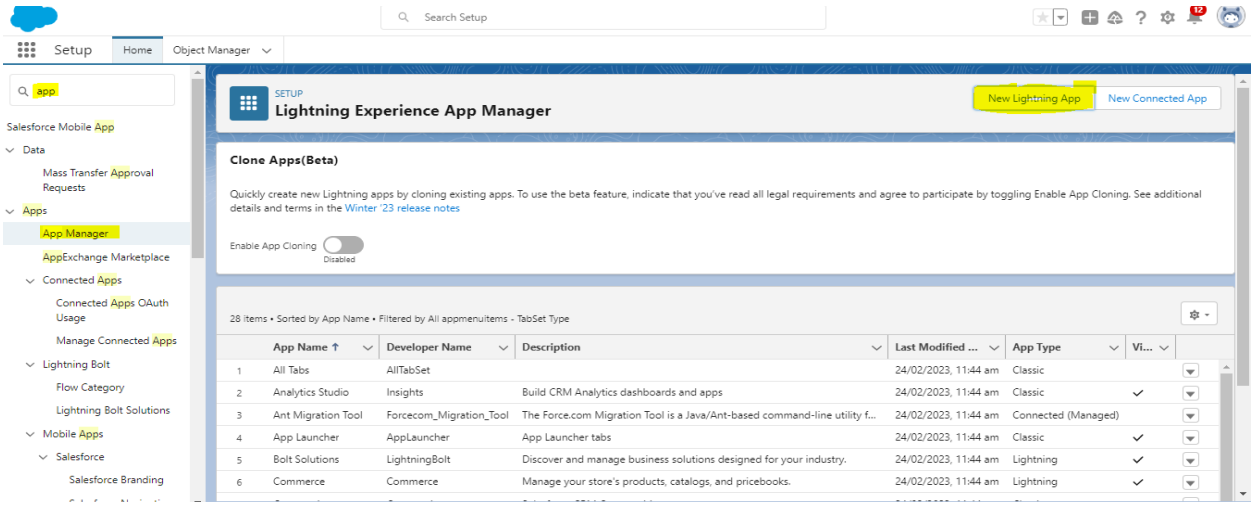
- Test The Dashboard
- Dashboard will look as per the screen shot:

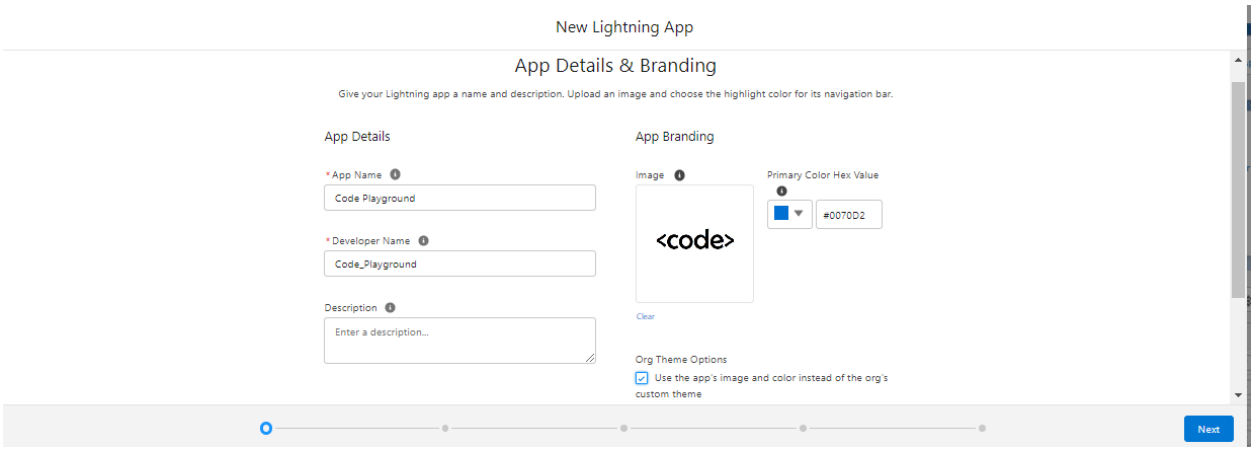


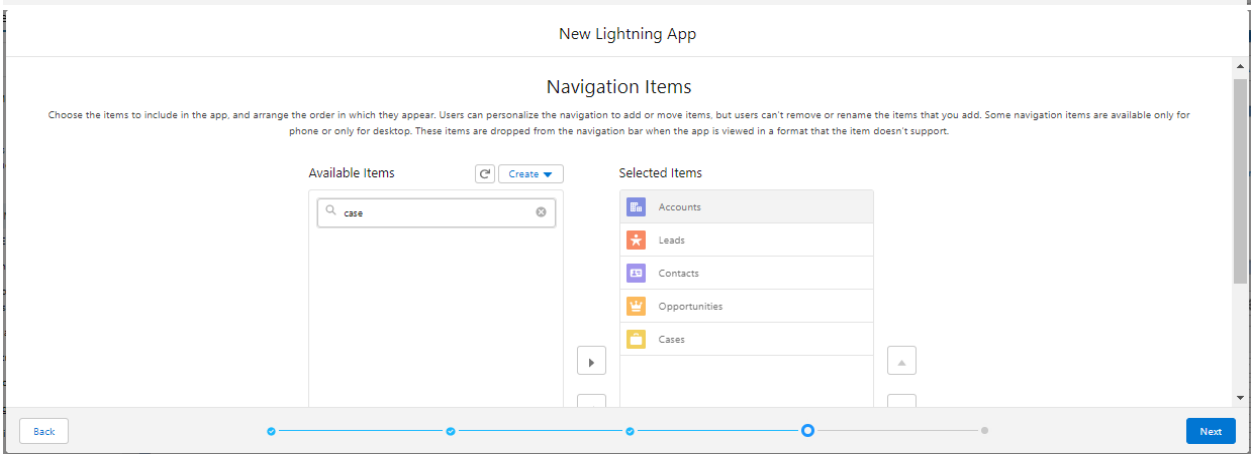
Module – 3

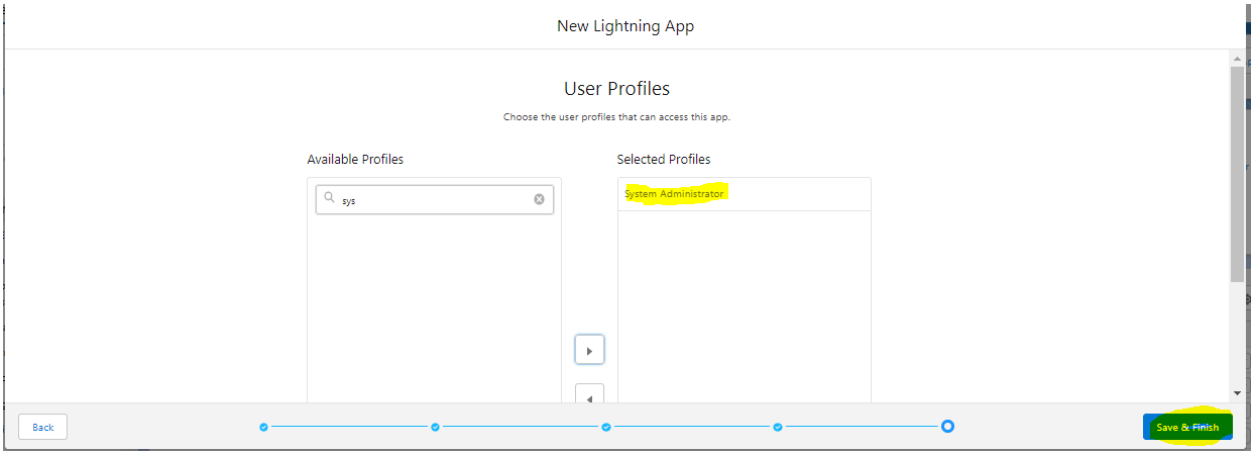
Exercise 1 : -

Step 1:- Create a new custom lightning App, name: Code Playground.

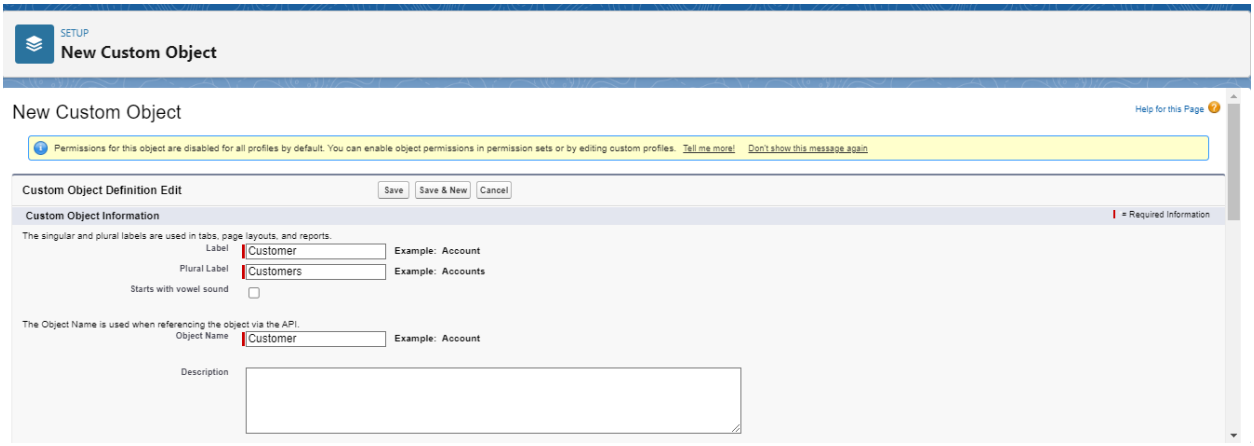


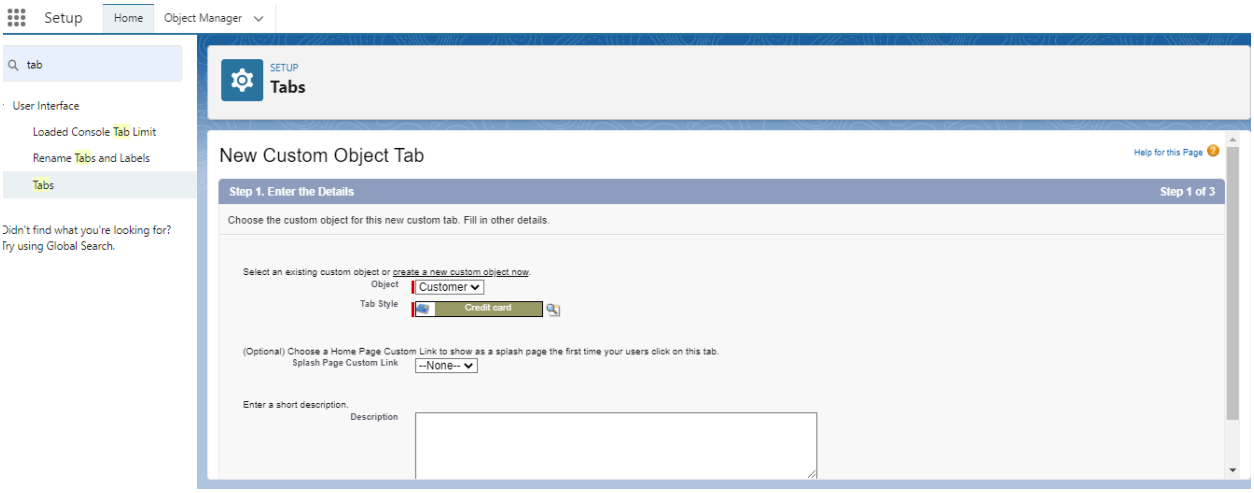






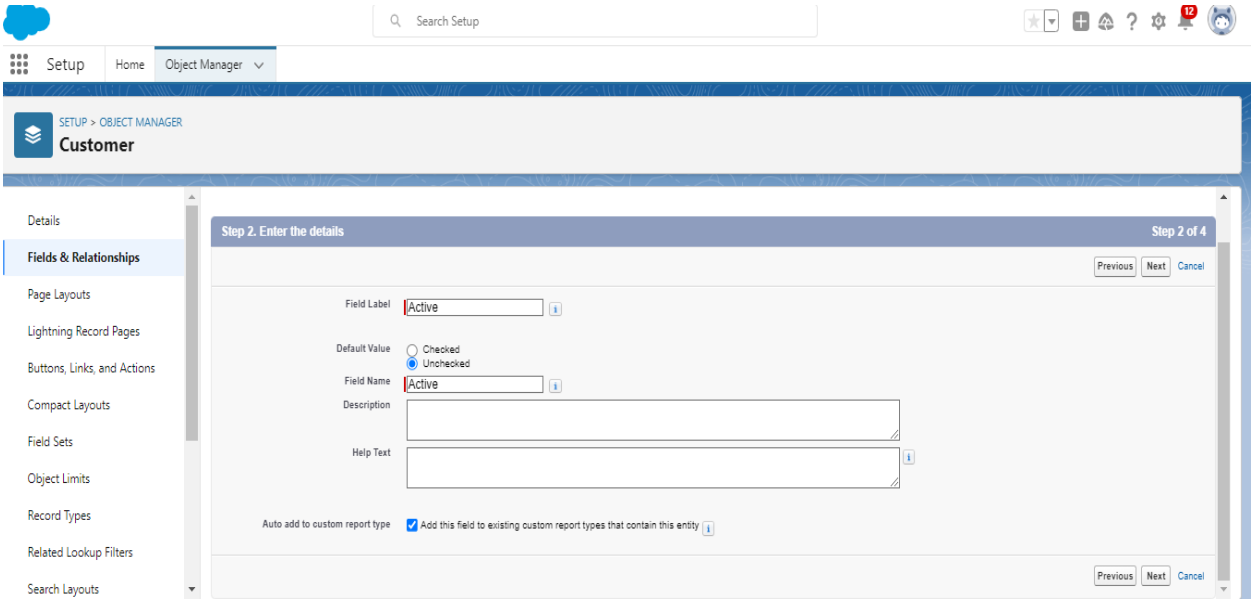
Step 2:- Create a Custom Object and tab for Customers.



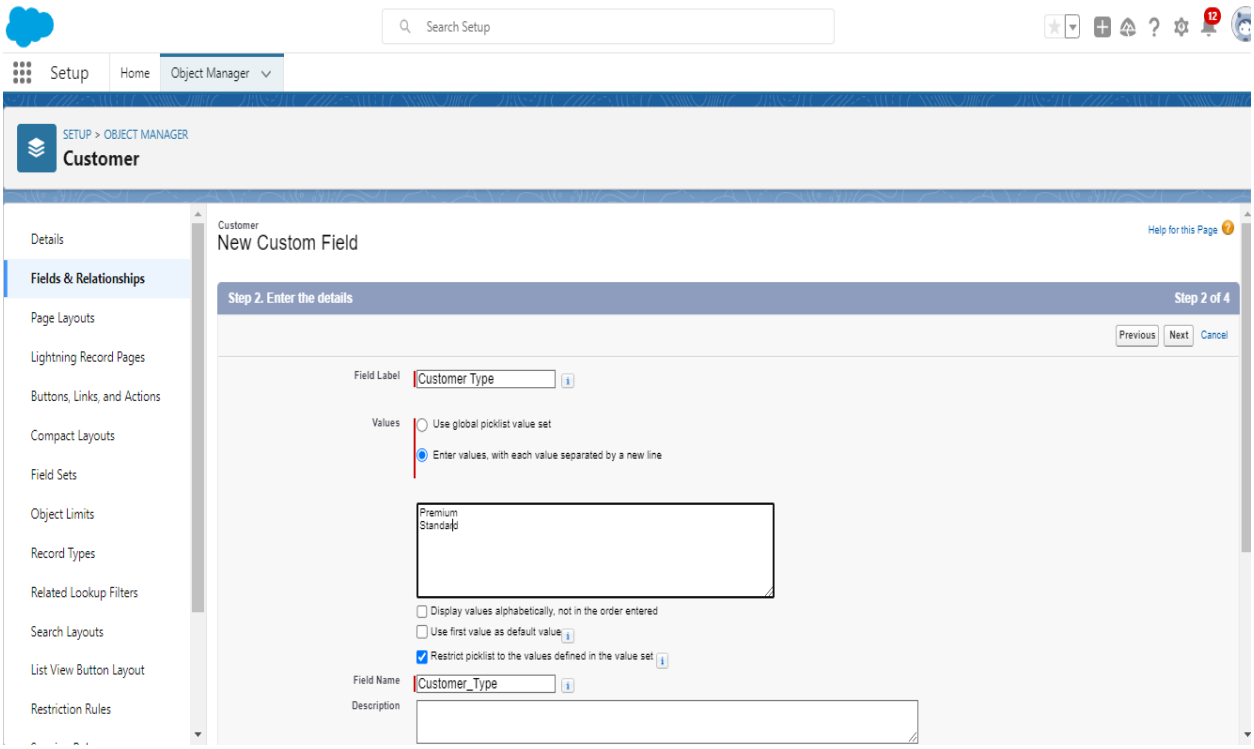


Step 3:- Create a Custom fields for Customer Object

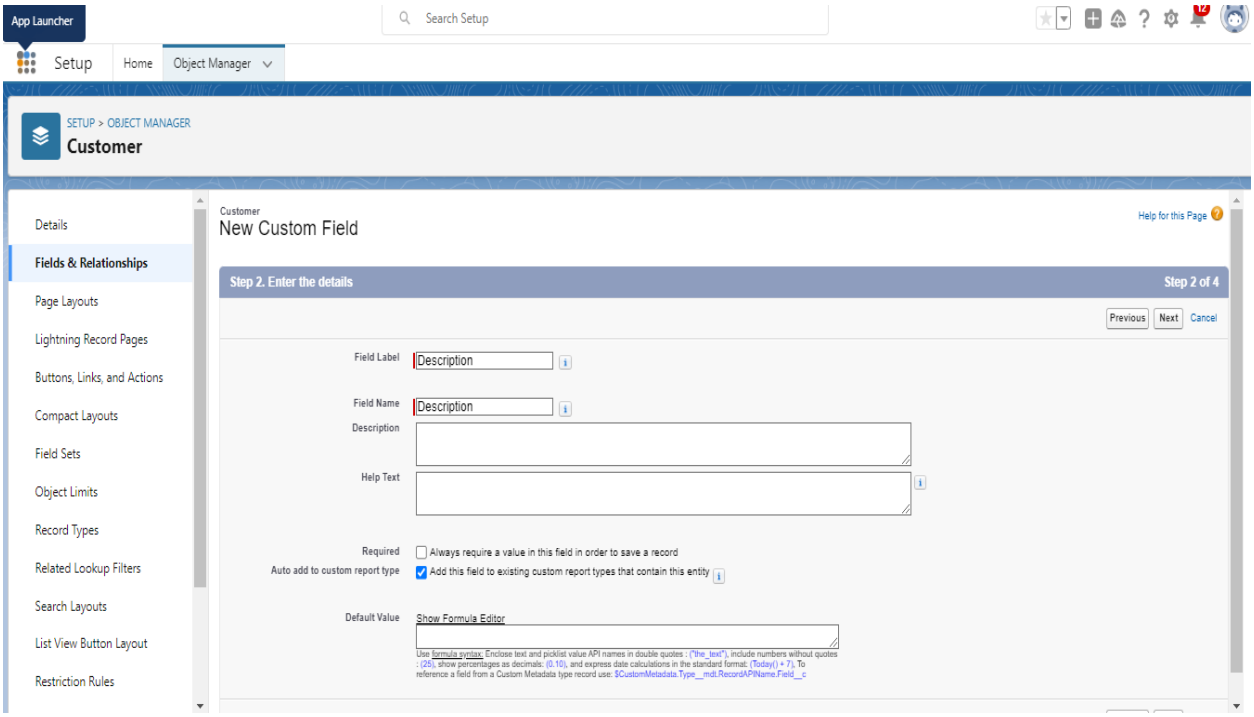
a. Label = Active, Checkbox, Save.



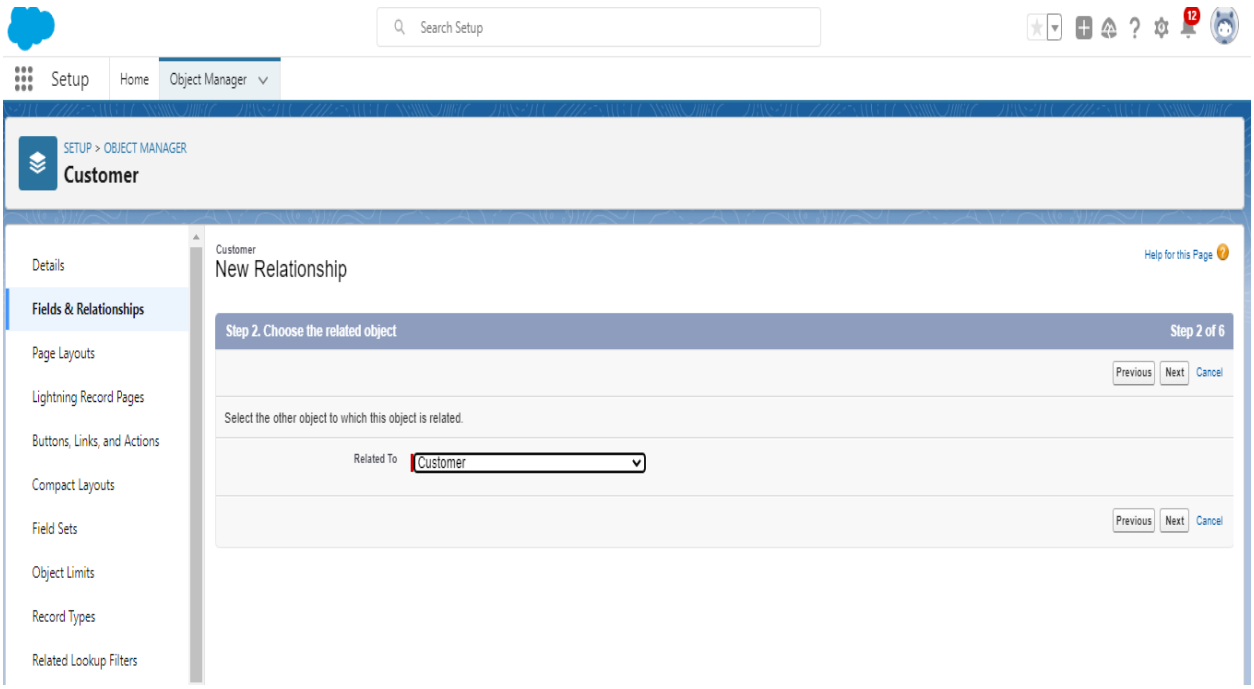
b. Label = Customer Type, Picklist, Values: Premium, Standard



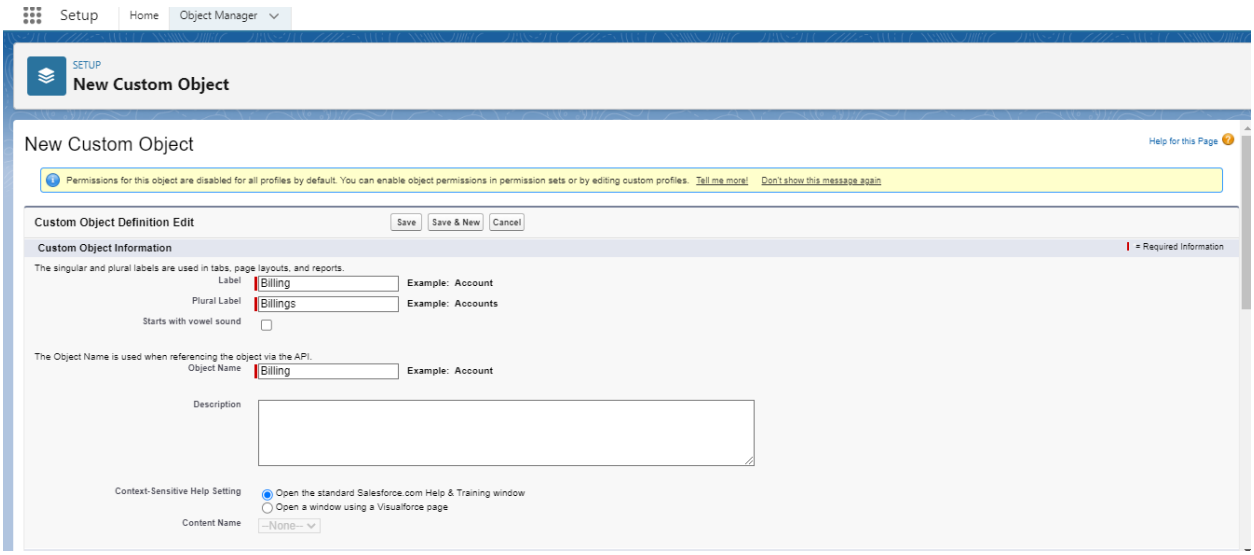
c. Label = Description, Text Area, Save.

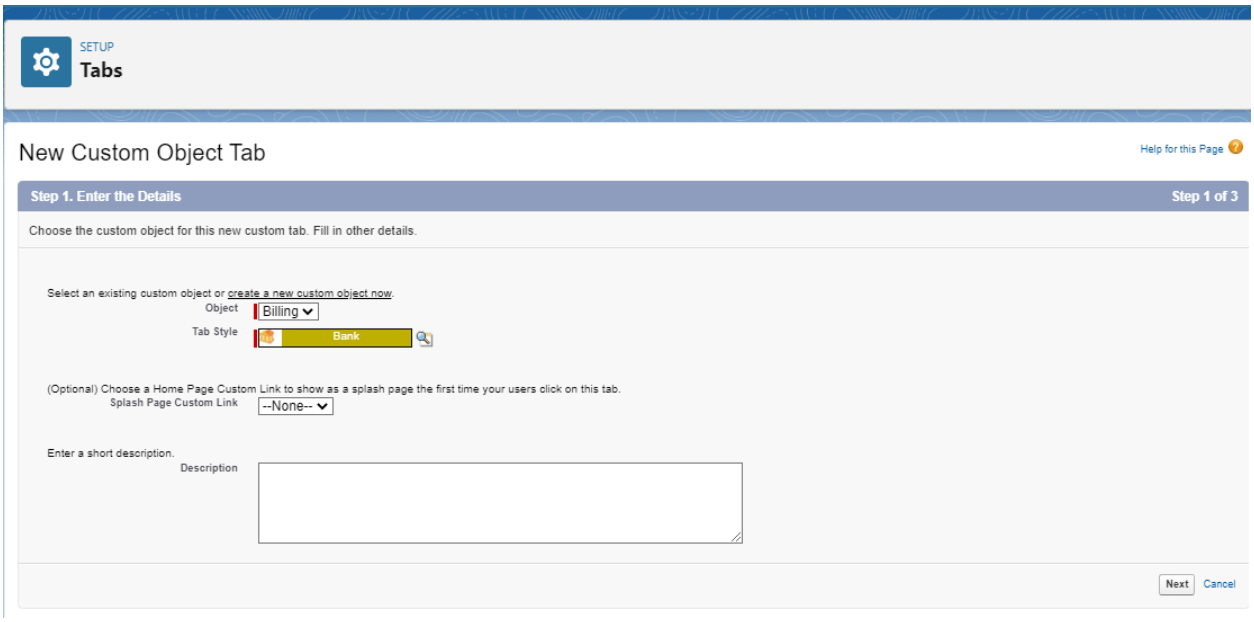


d. Label = Customer, Master-Detail, Related To – Customer custom object



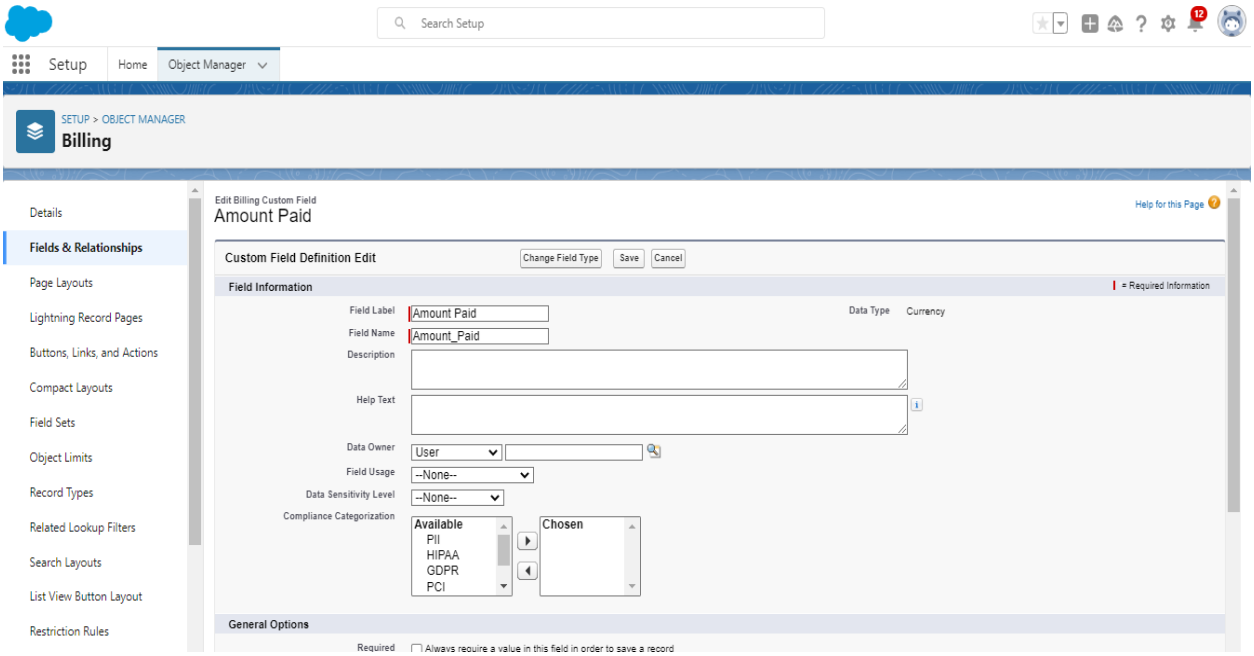
Step 4:- Create a Custom Object for Billing.



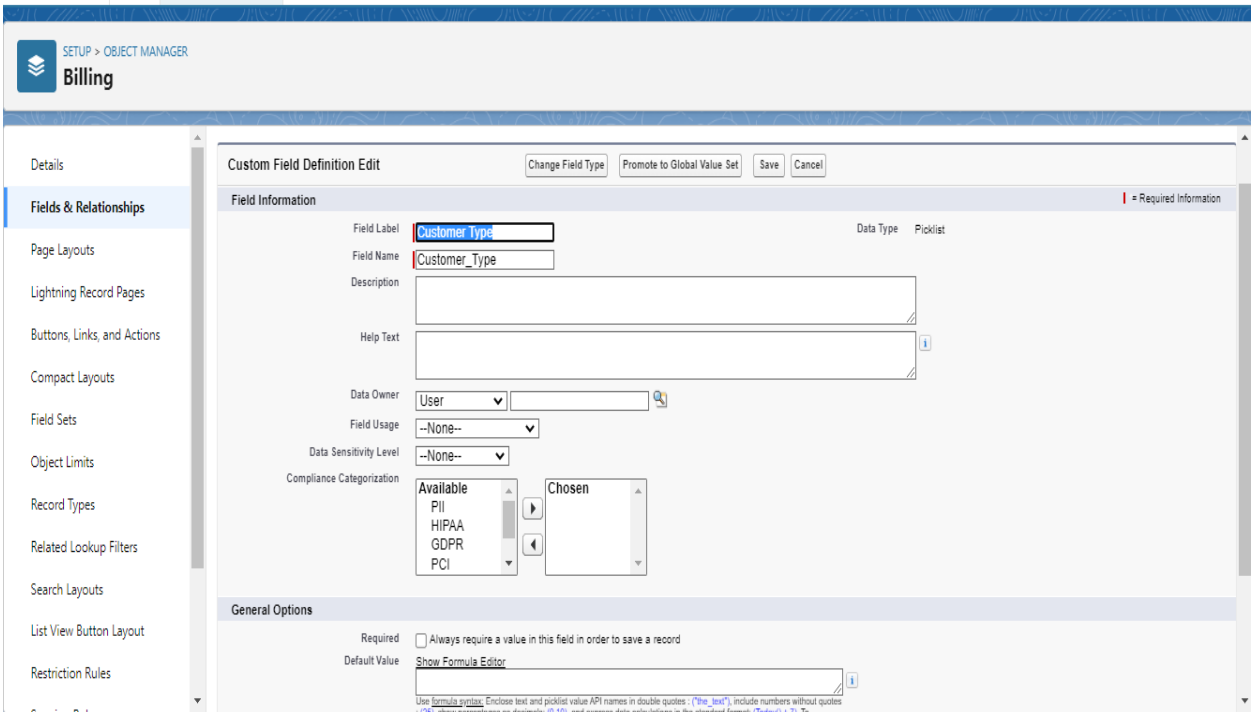


Step 5:- Create a Custom fields for Billing Object.

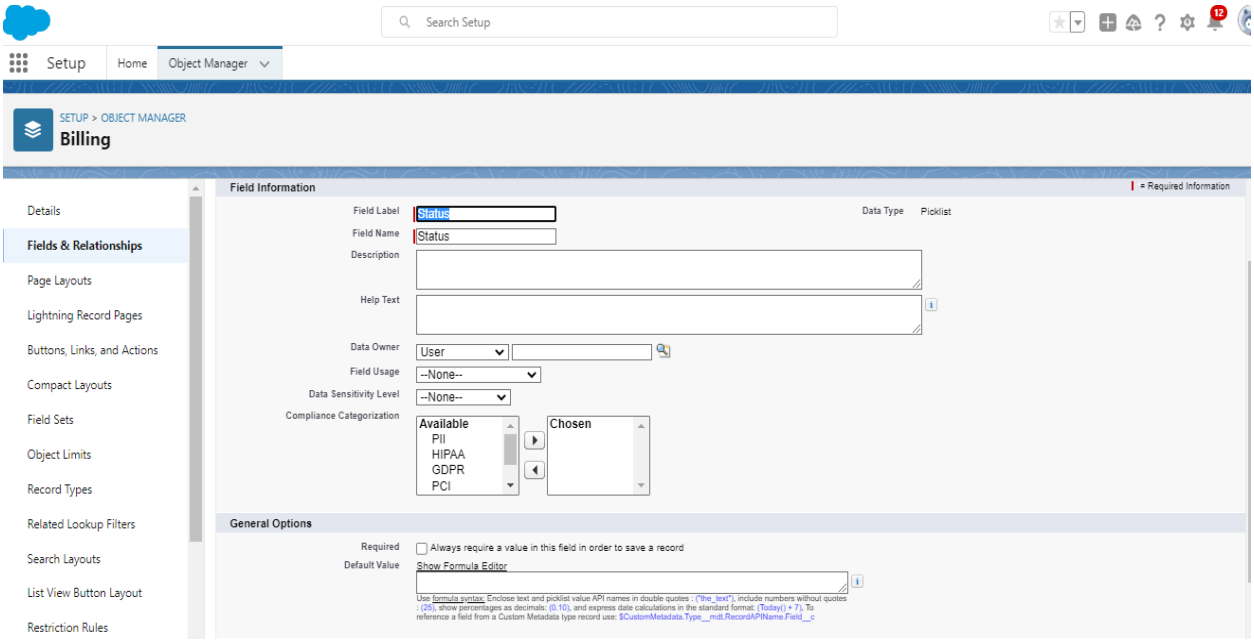
a. Label = Amount Paid, Currency, Save.



b. Label = Customer Type, Picklist, Values: Premium, Standard.

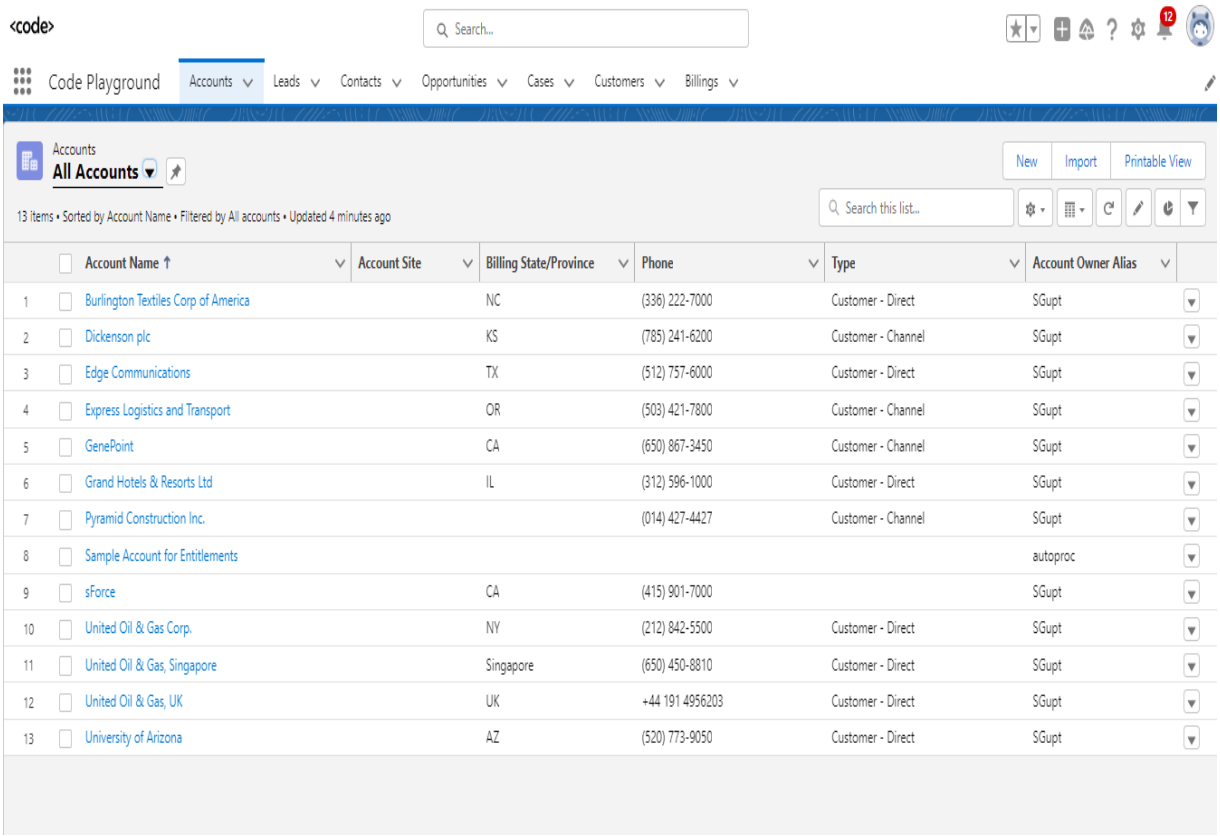


c. Label = Status, Picklist, Values: Paid, Unpaid.



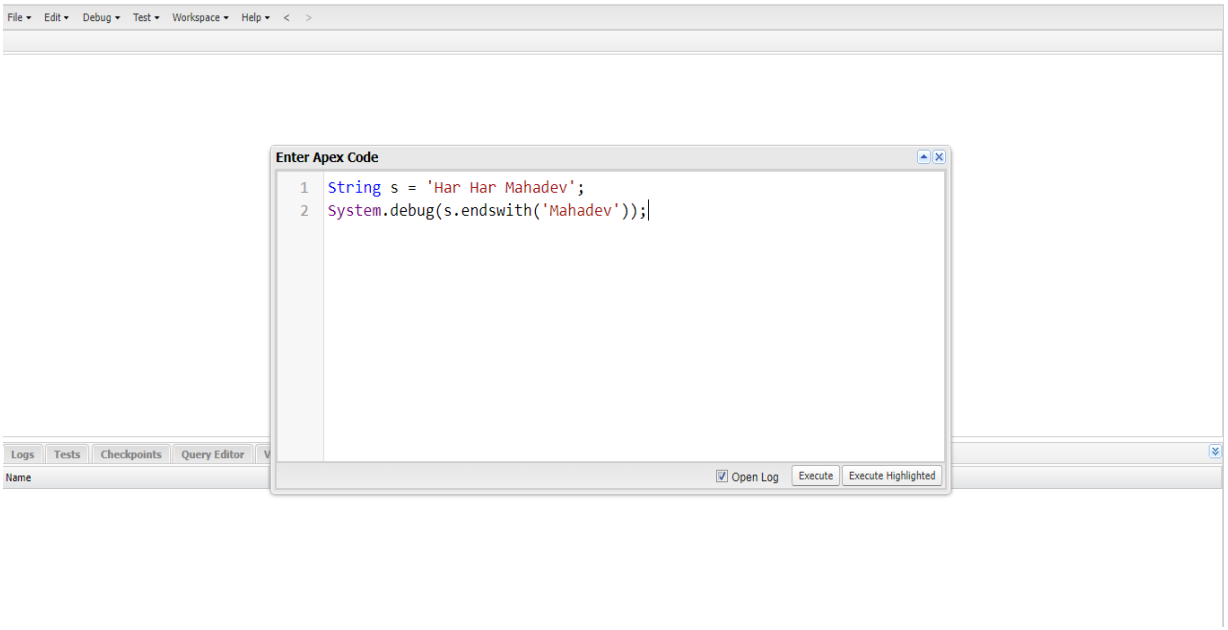
Test The App

• Code Playground App should look like this:

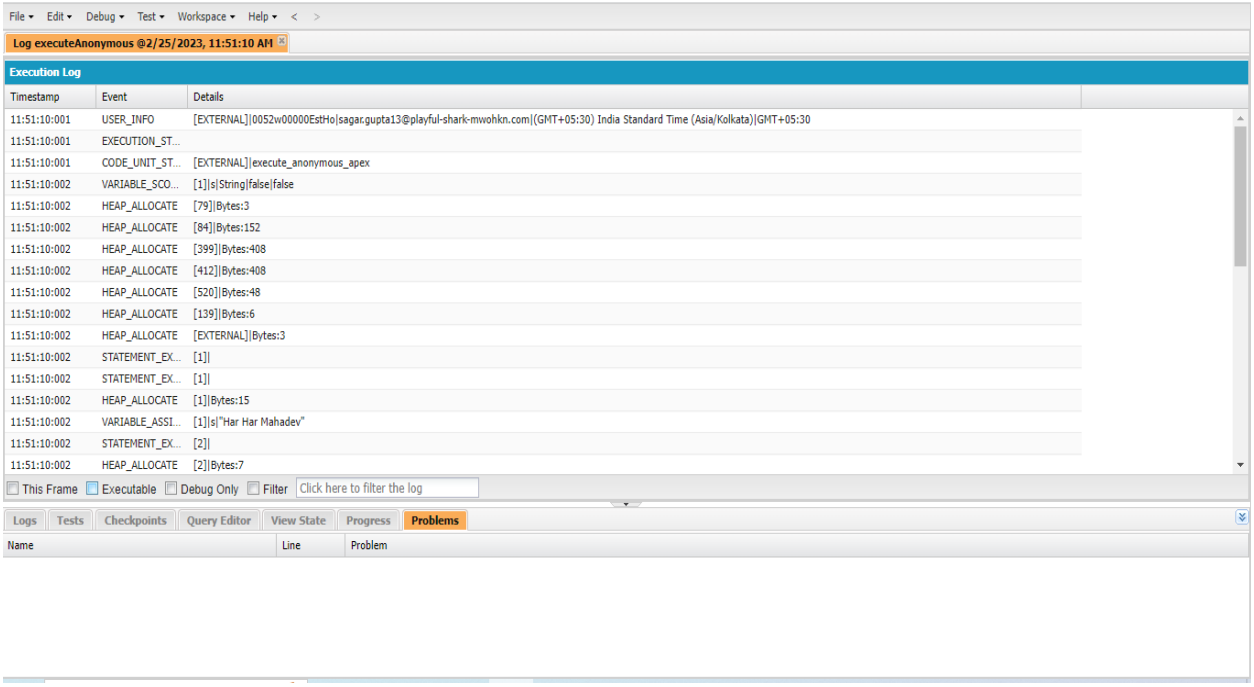


Exercise 2 : - Use Execute Anonymous to define and execute the following code:

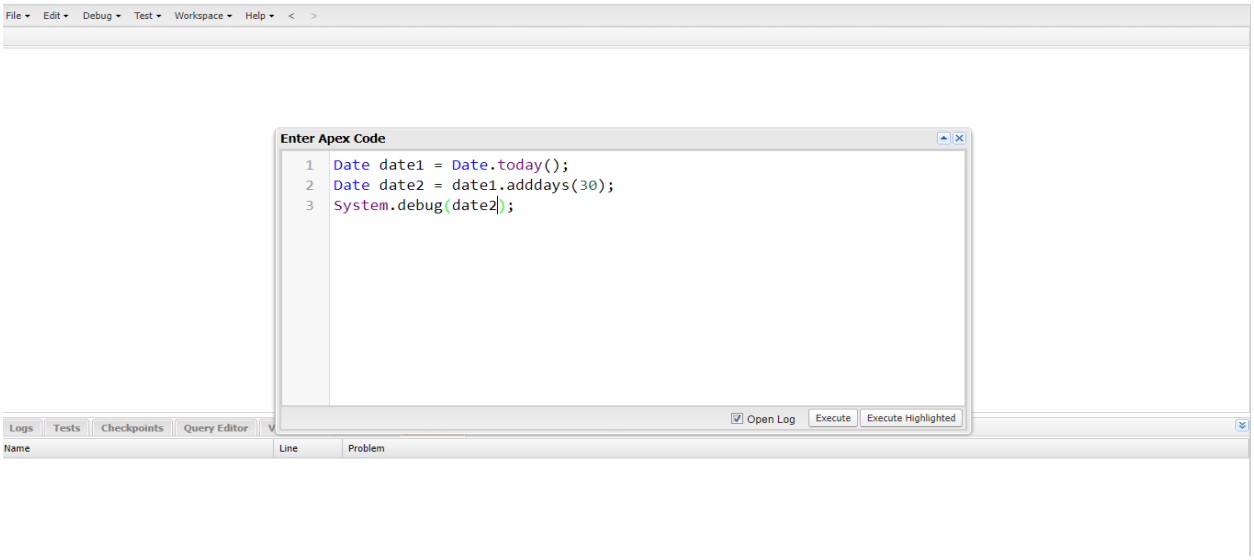
Code 1:- Define a String Variable & use string method ‘endsWith’ to display the output.



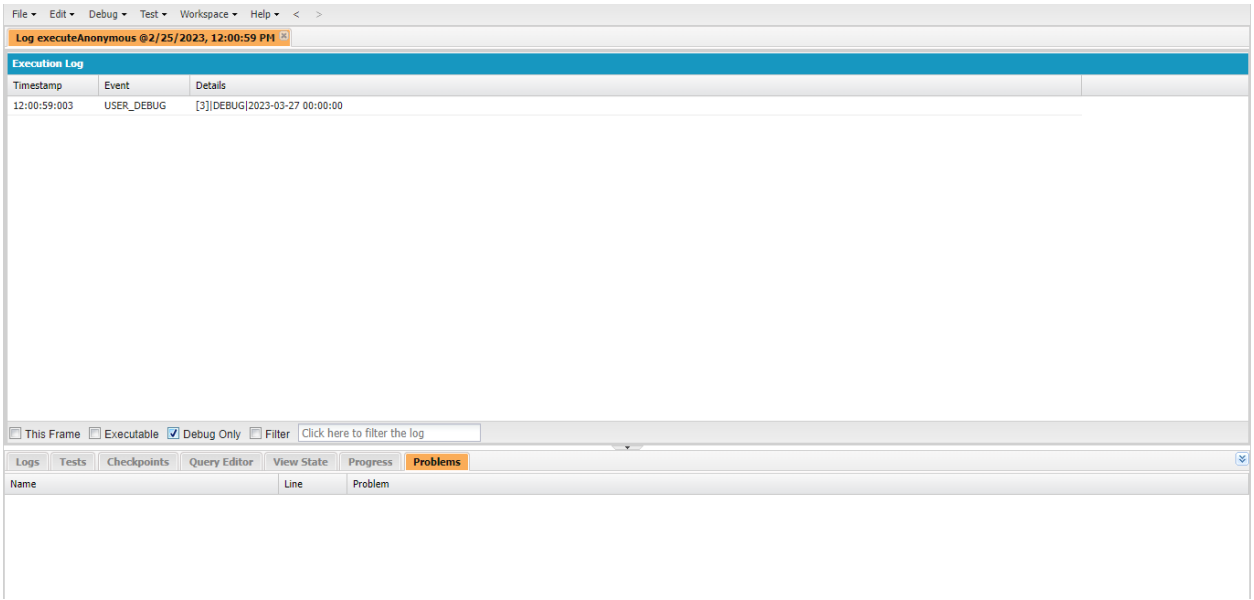
Output :



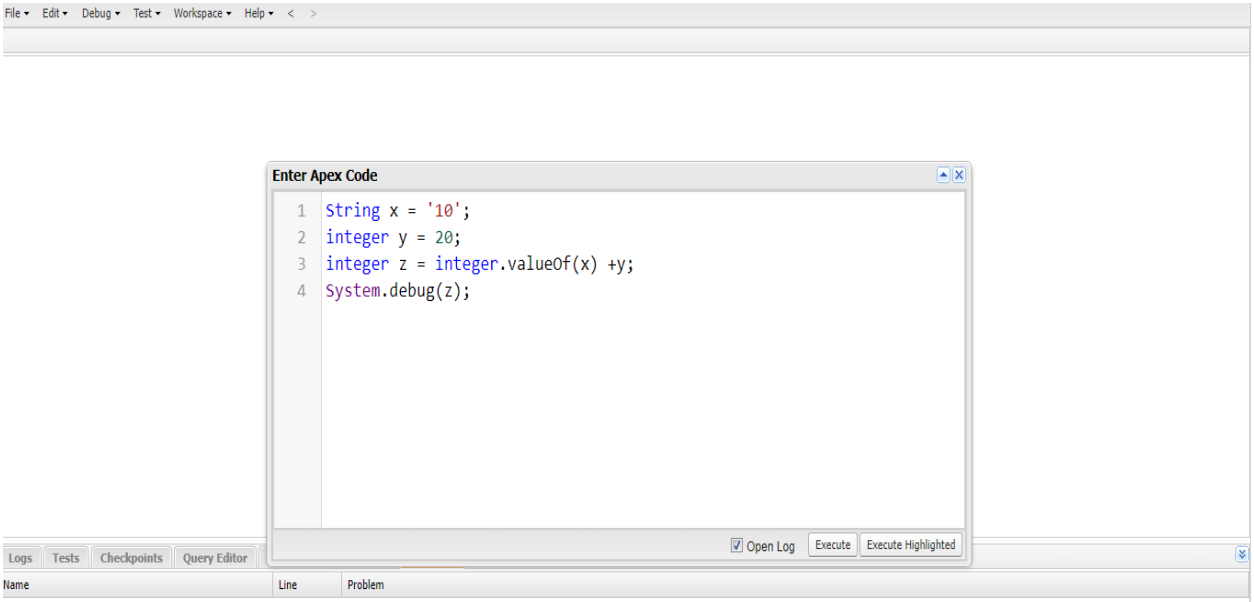
Code 2 : Define 2 Date type variables, use Date method today() & addDays(30) to display the output.



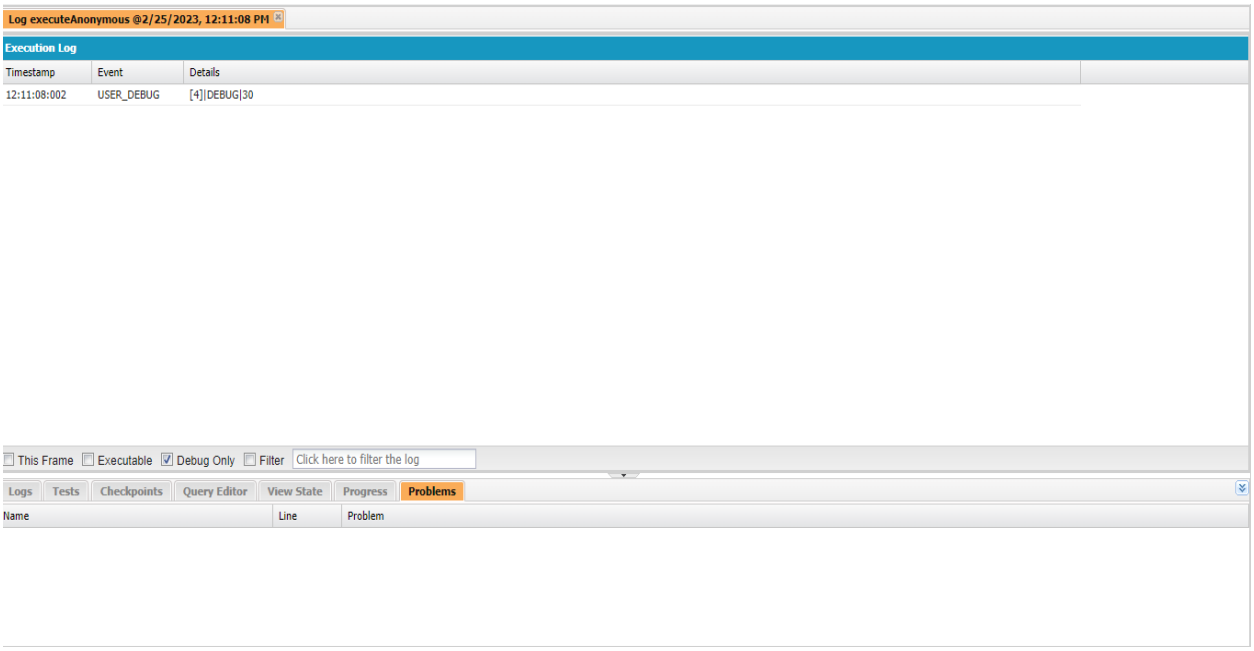
Output :



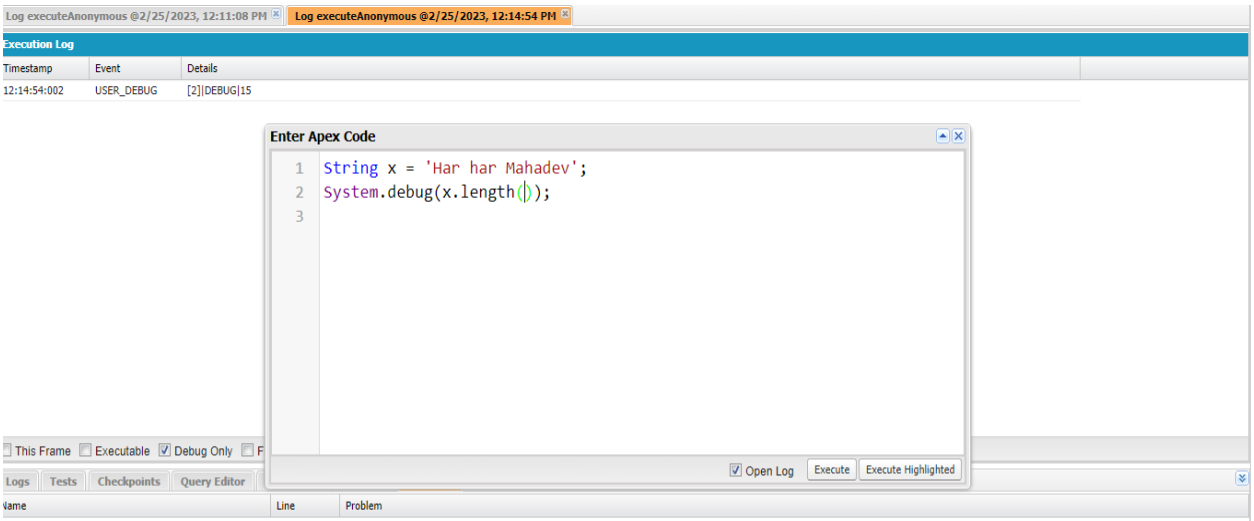
Code 4 : Display the output of an Integer variable from string ‘10’ and then add 20 to it.



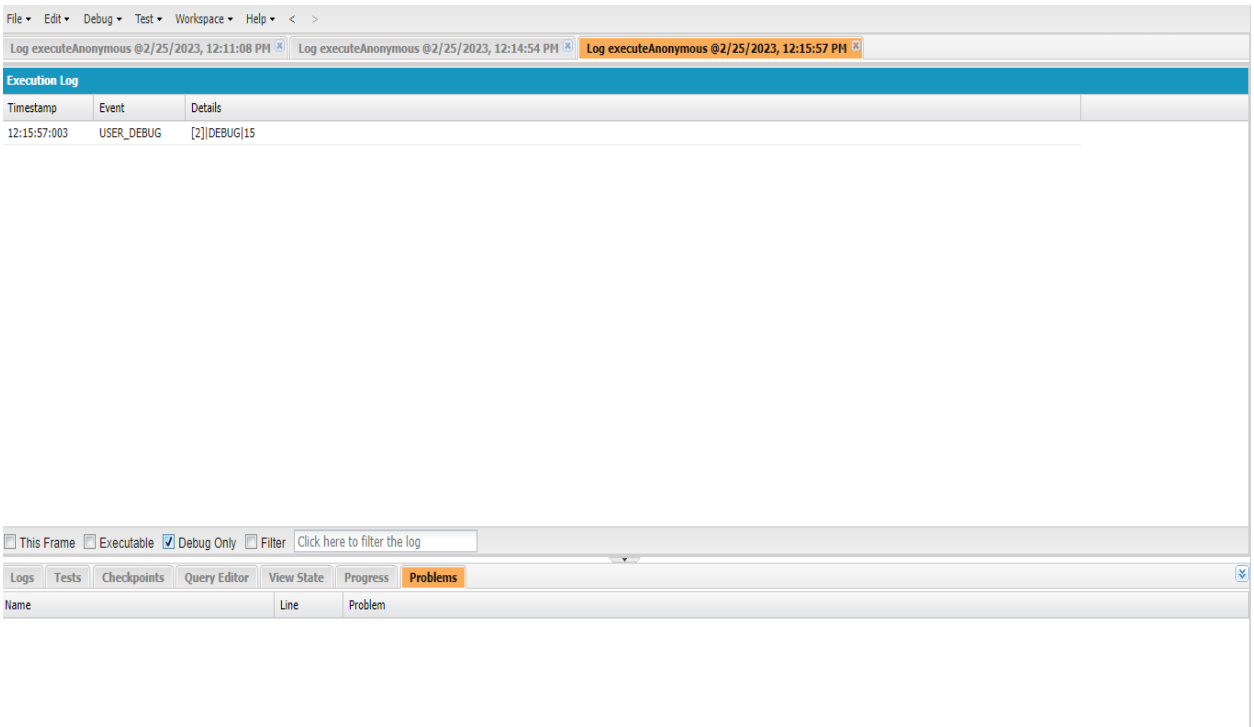
Output :



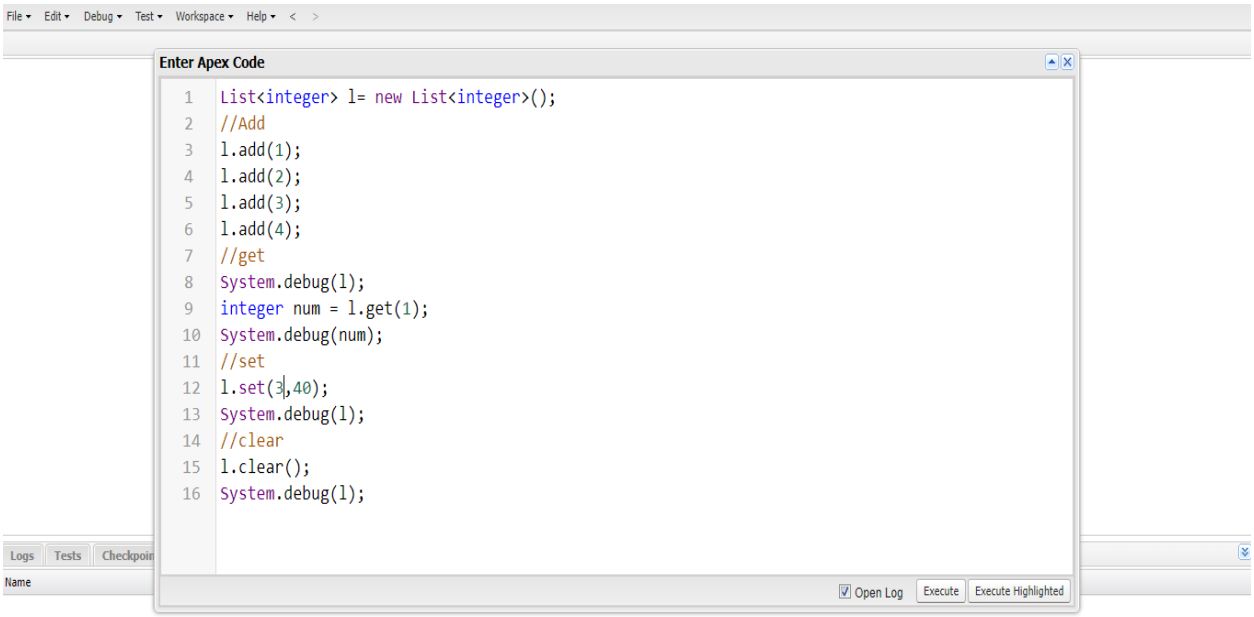
Code 5 : Define a String Variable & use string method length() to display the output.



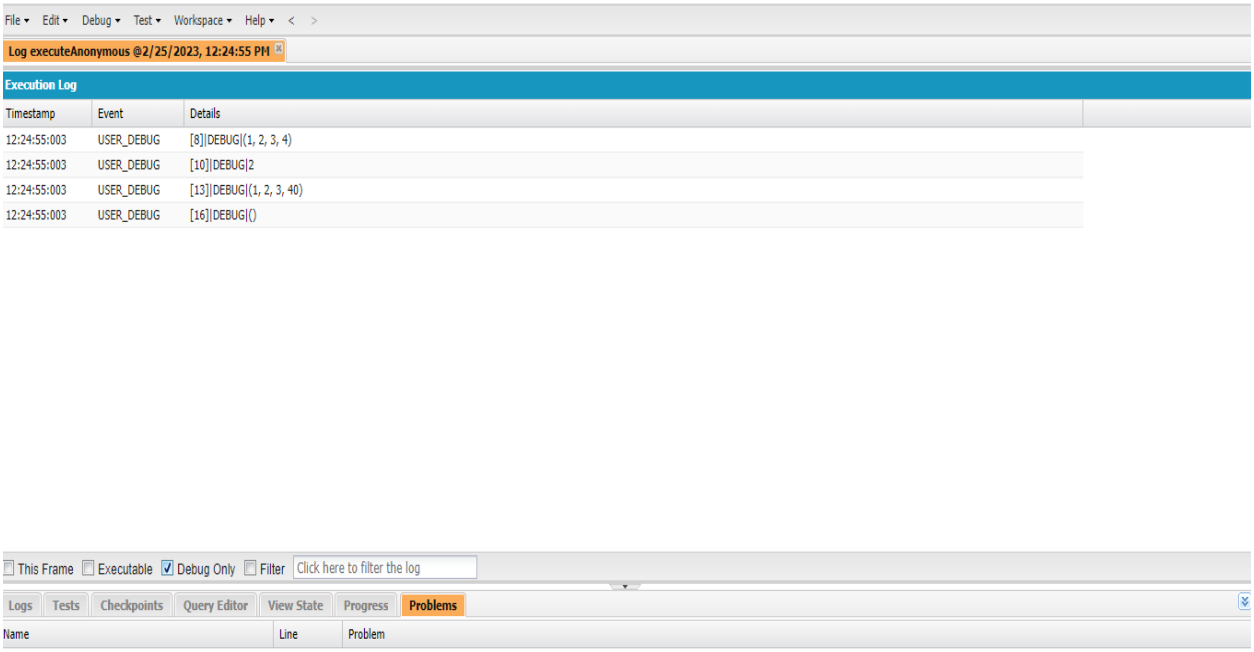
Output :



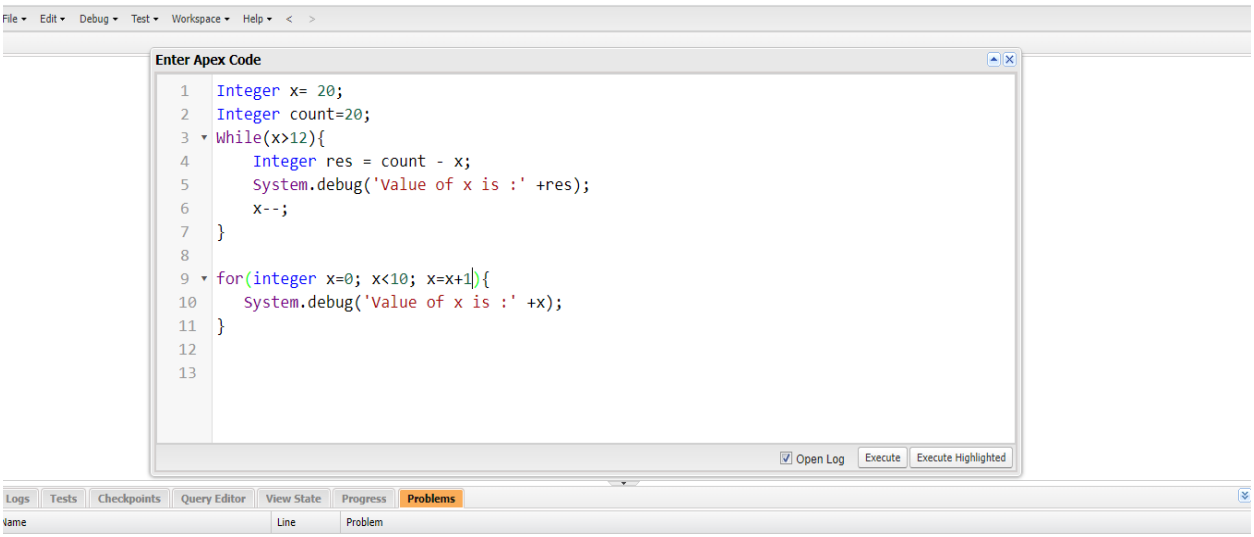
Code 6 : Define a List of integer and display the output using add(), get(), set(), clear(), methods



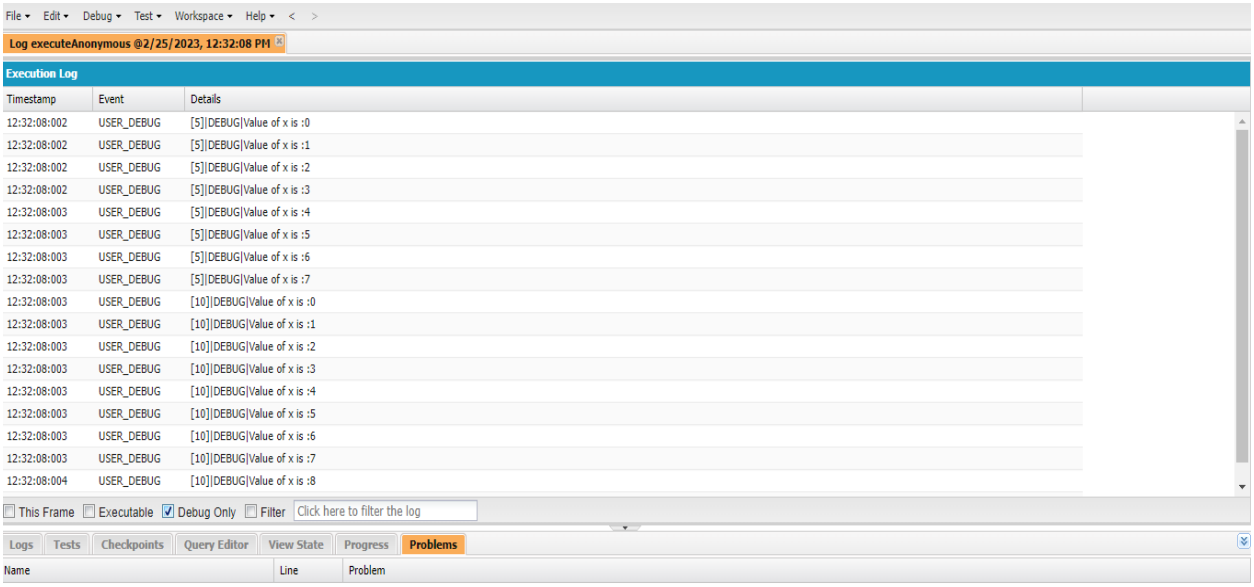
Output :



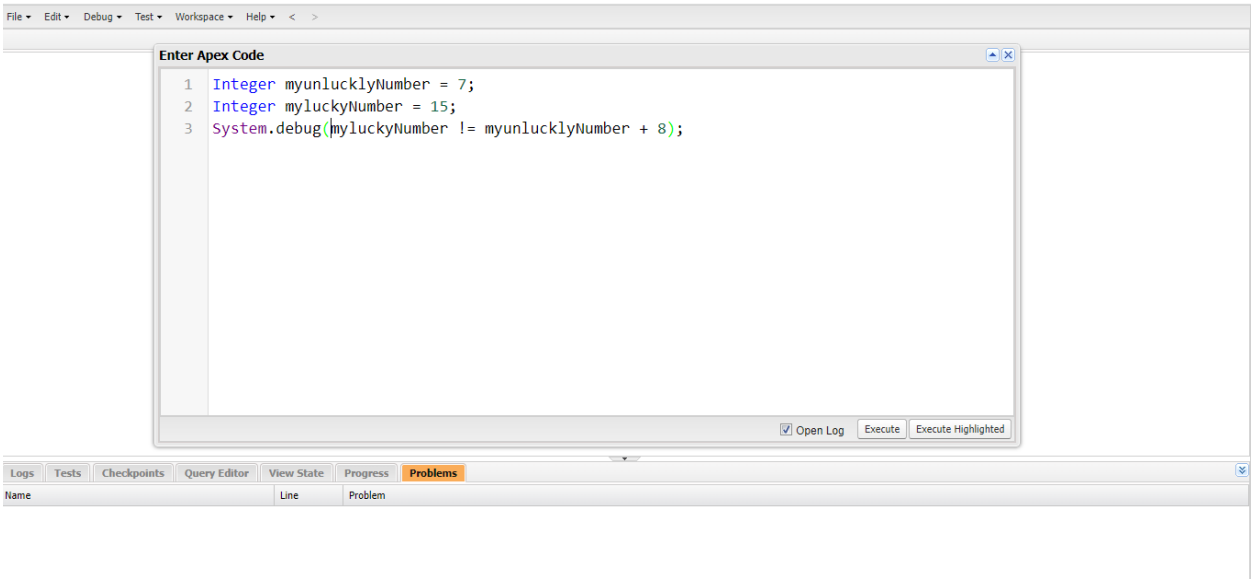
Code 7 : Use Execute Anonymous to define and execute the following code to display the value of x = 0 to 9.



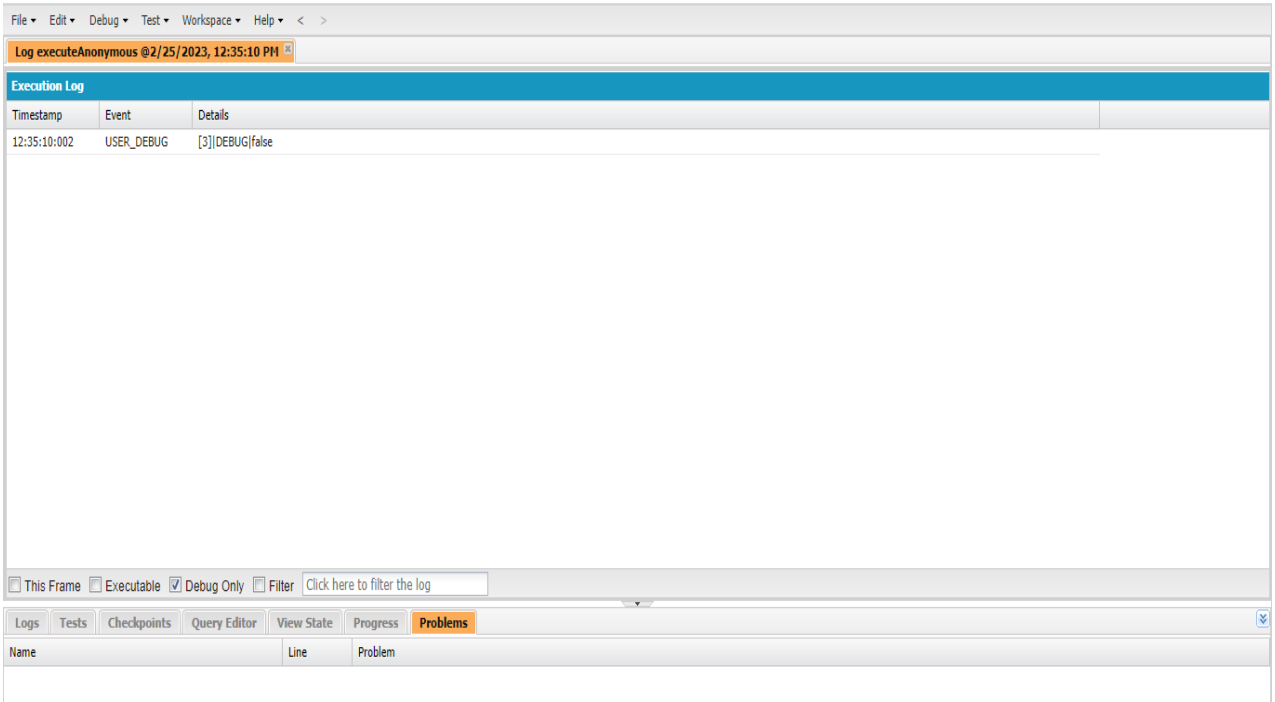
Output :



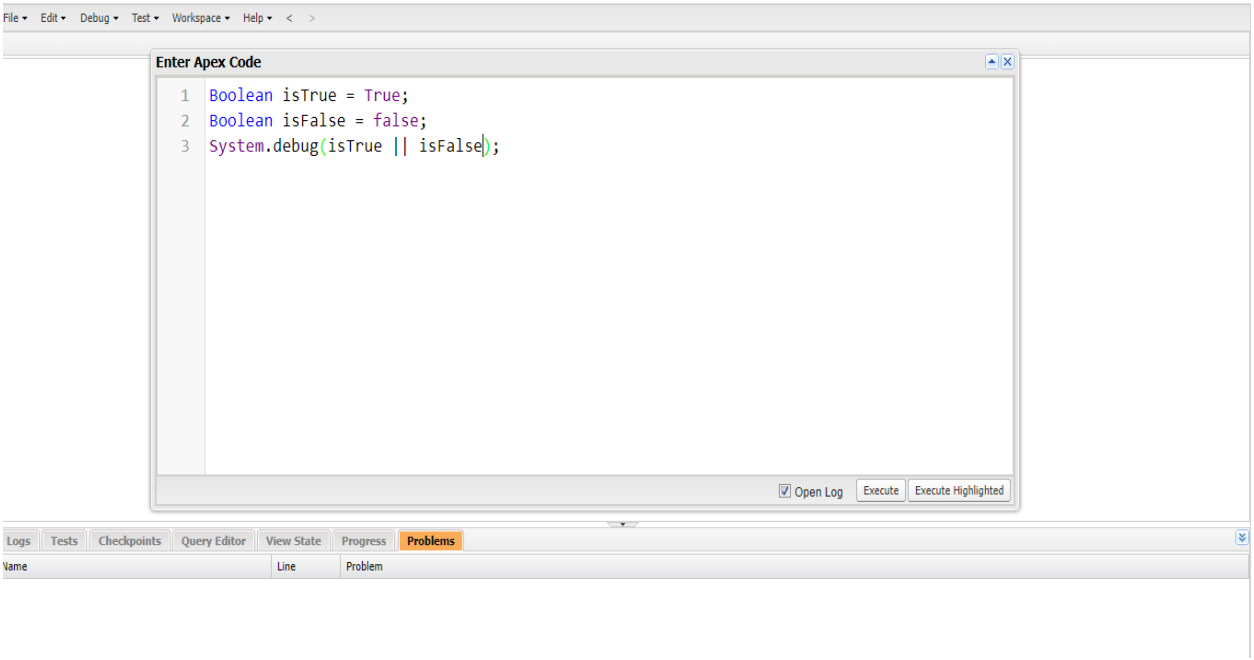
Exercise 3 : - Answer the following in True Or False:



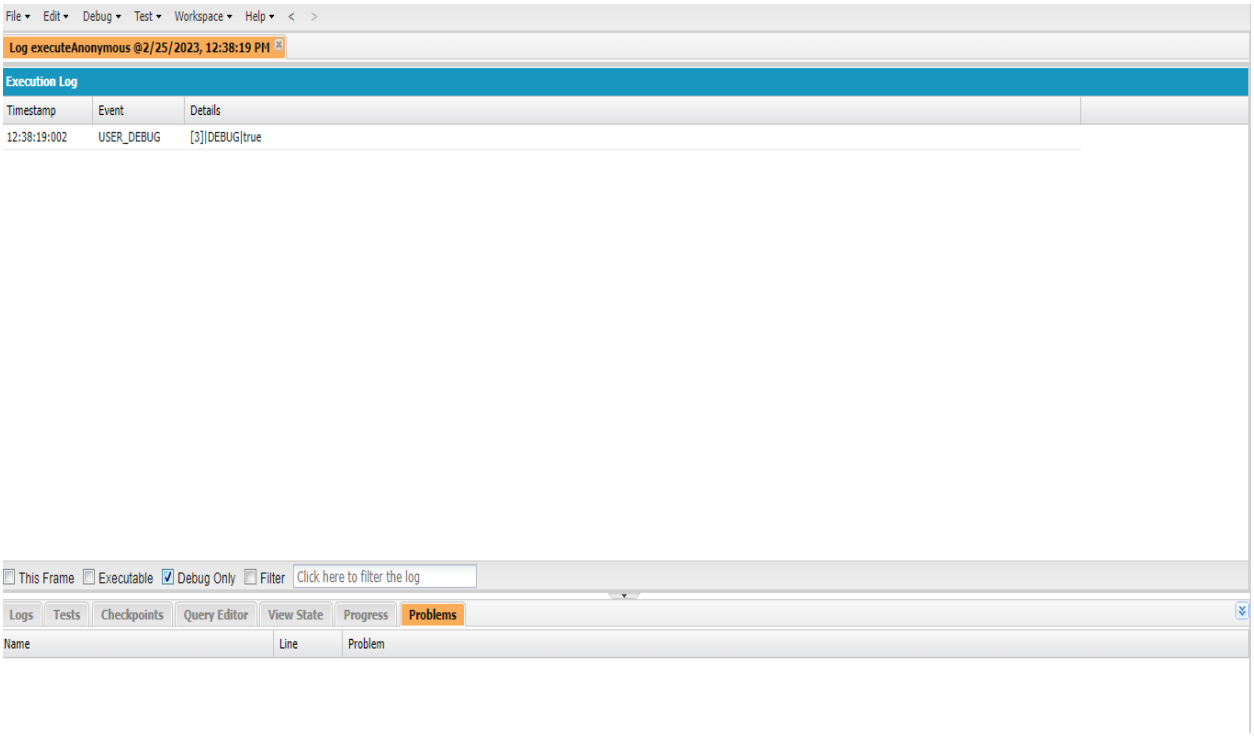
Output :



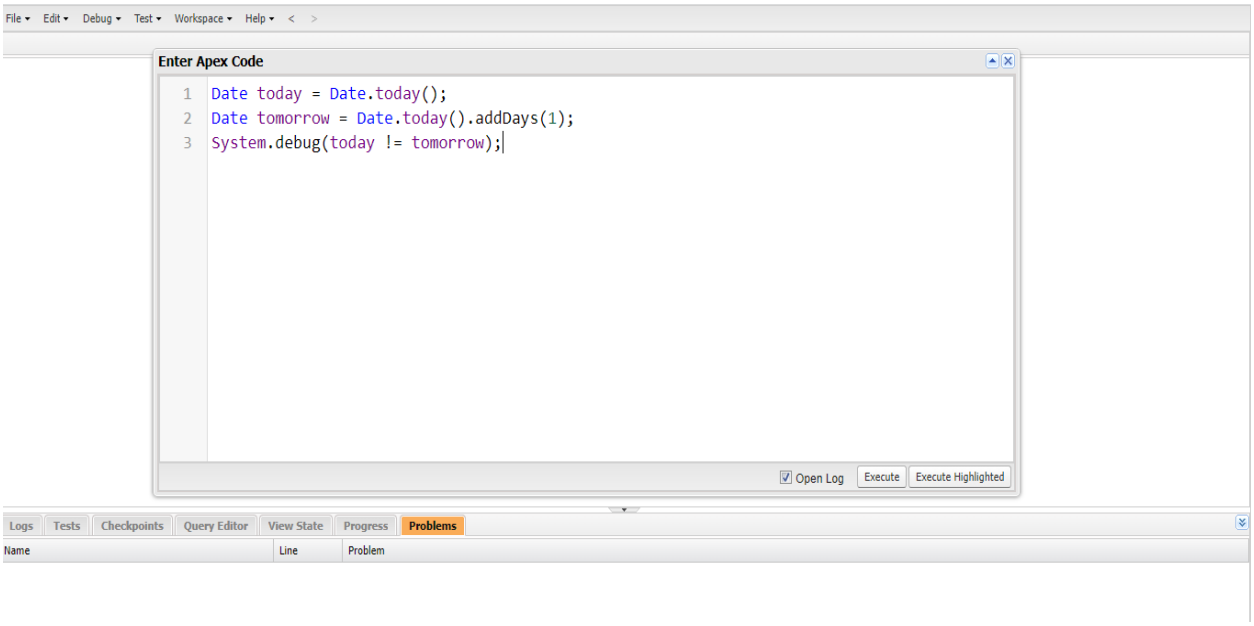
Exercise 4 : - Answer the following in True Or False:



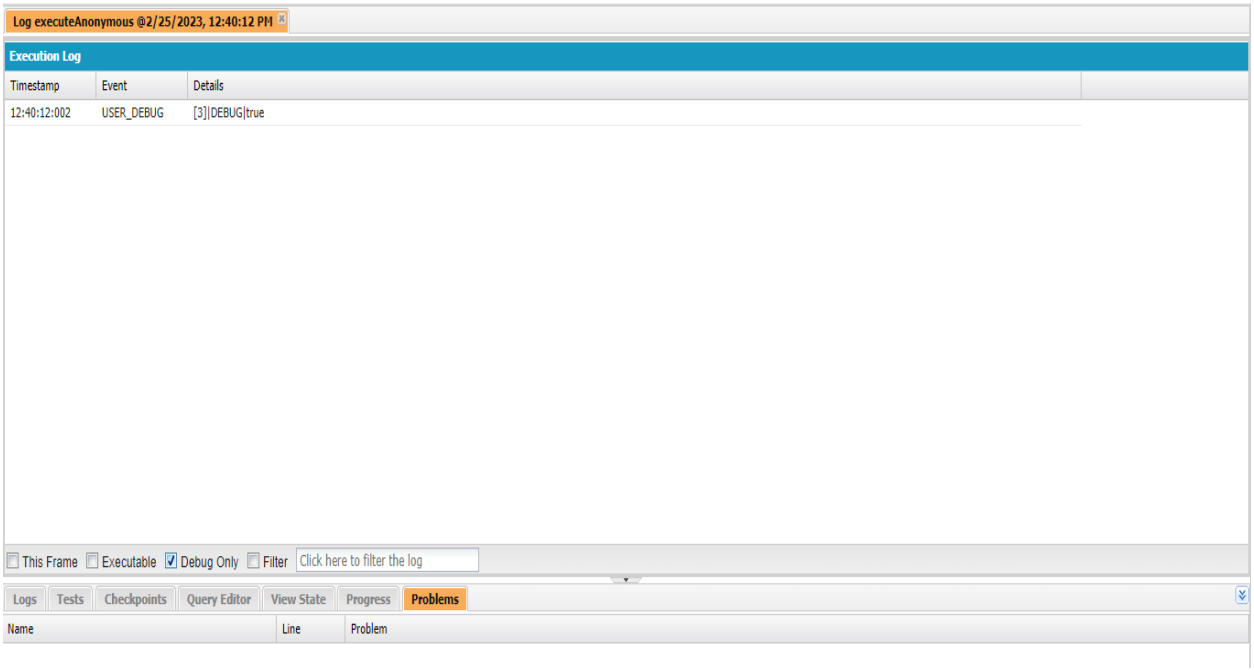
Output :



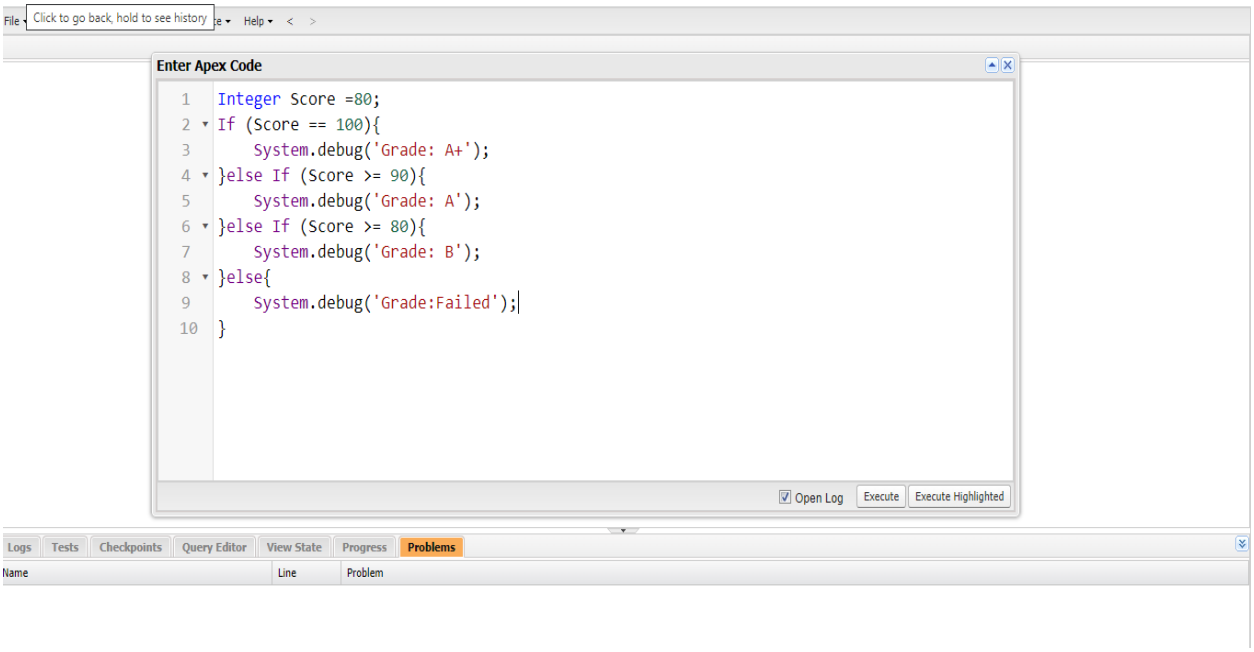
Exercise 5 : - Answer the following in True Or False:



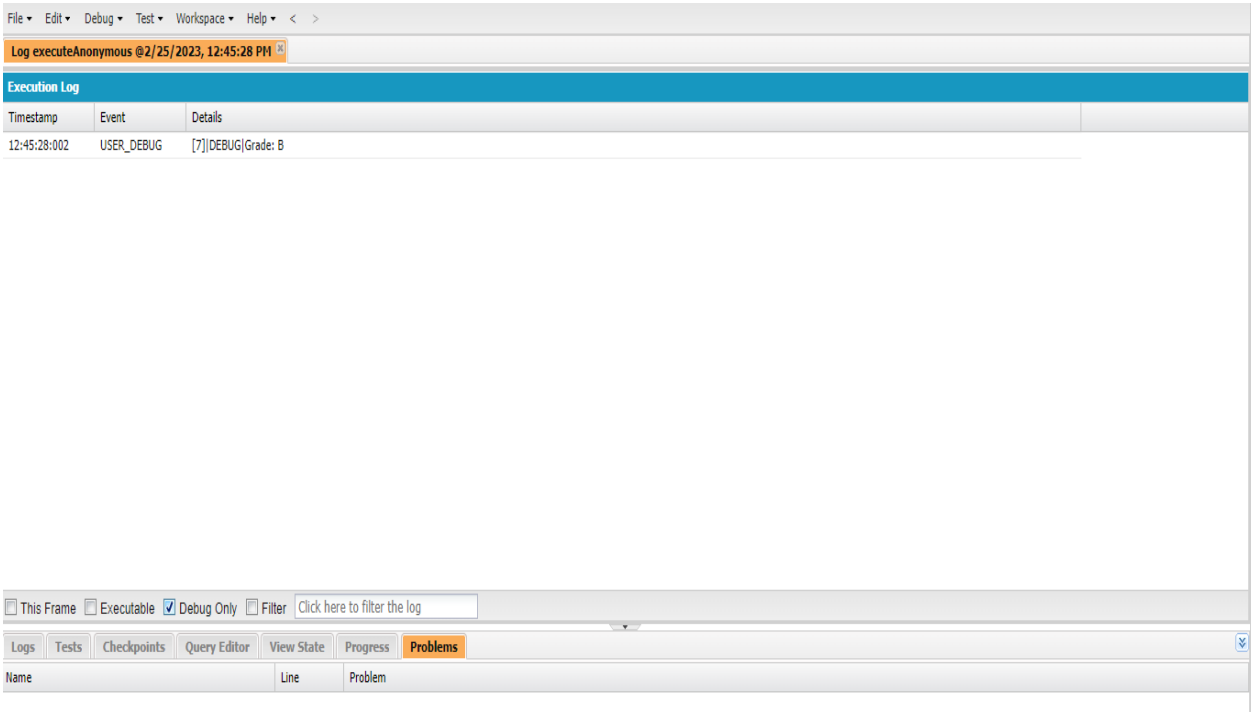
Output :



Exercise 6 : - Write a program and execute to demo the use of “If..else if...else”.

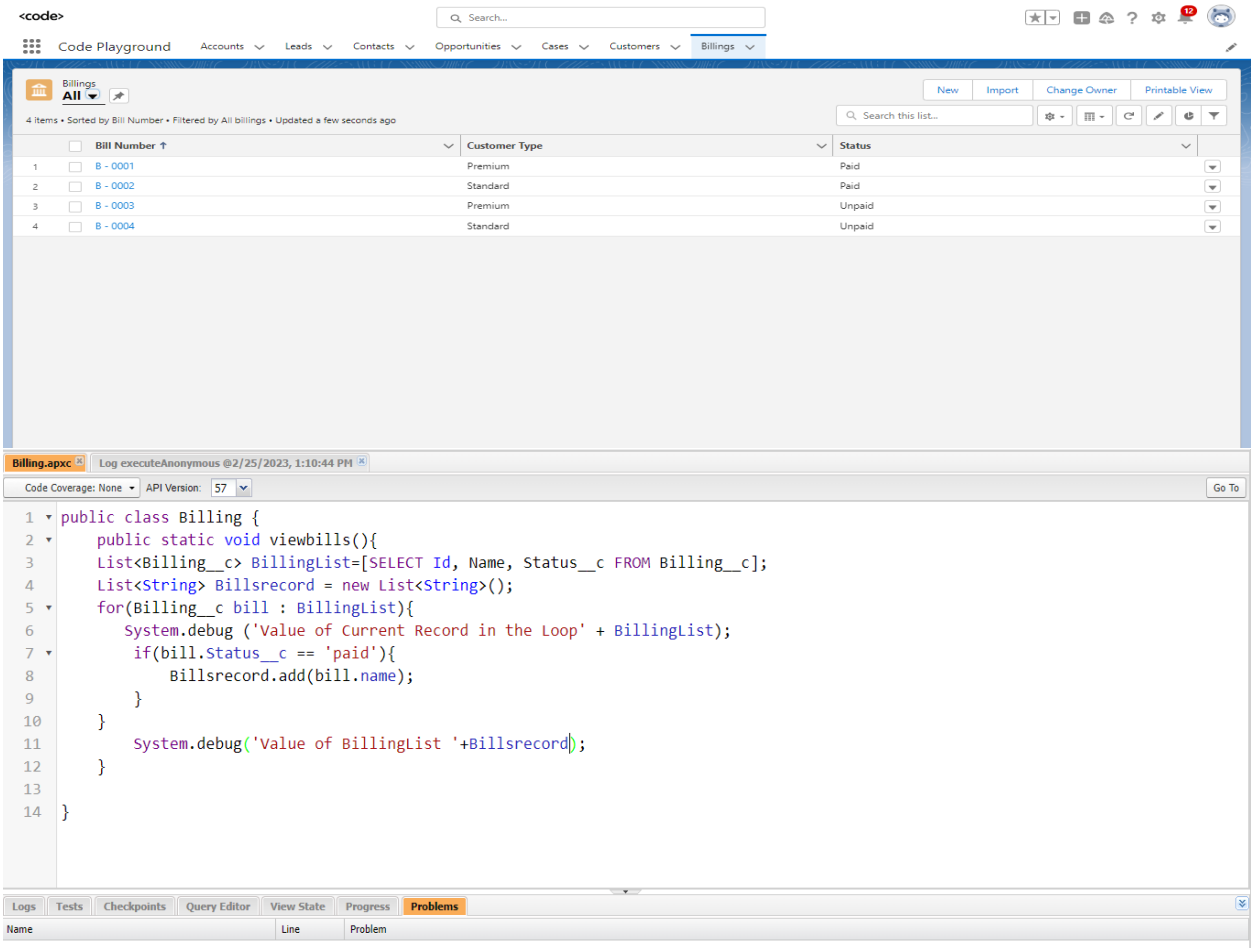


Output :

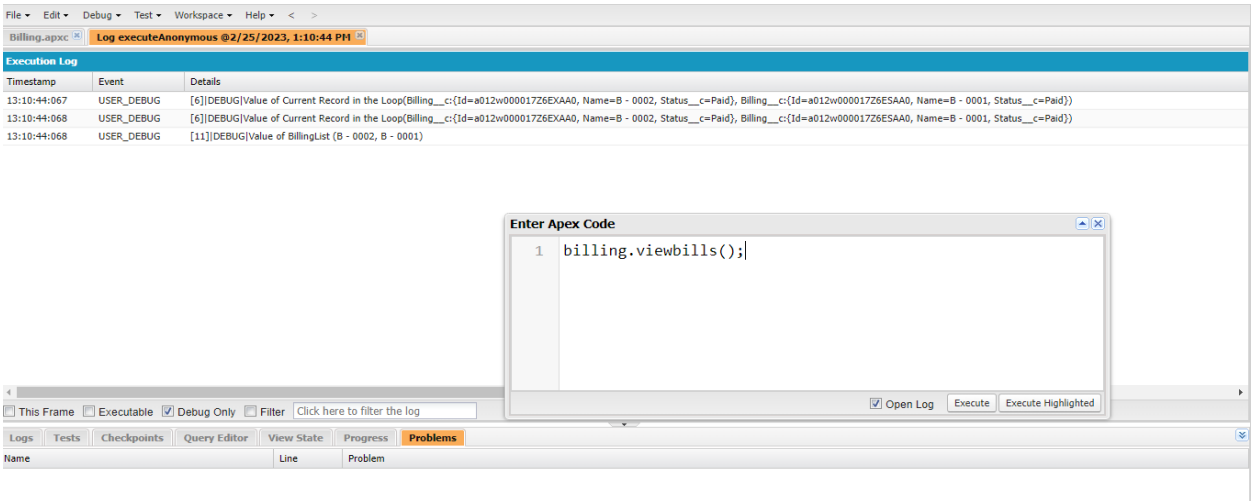


Exercise 7 : - Write a program to execute and demo the use of “Apex – for Loop”

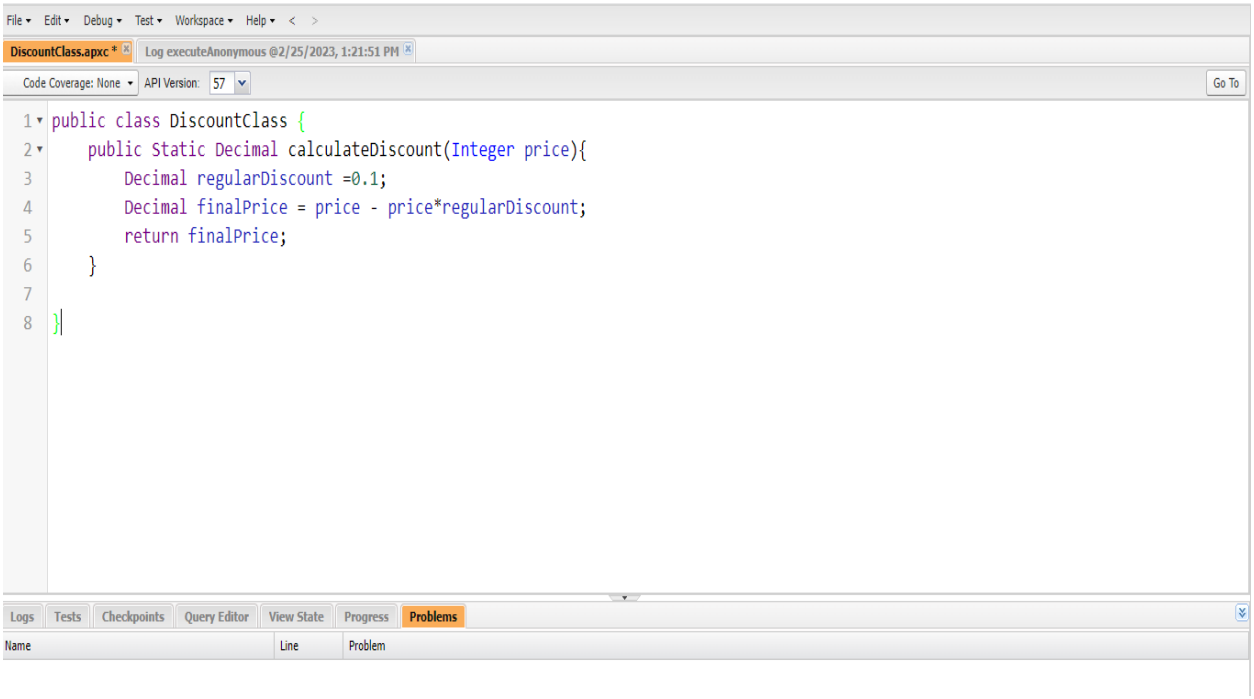
Note: Create at least 2 Billing records with Status = Paid for this exercise.



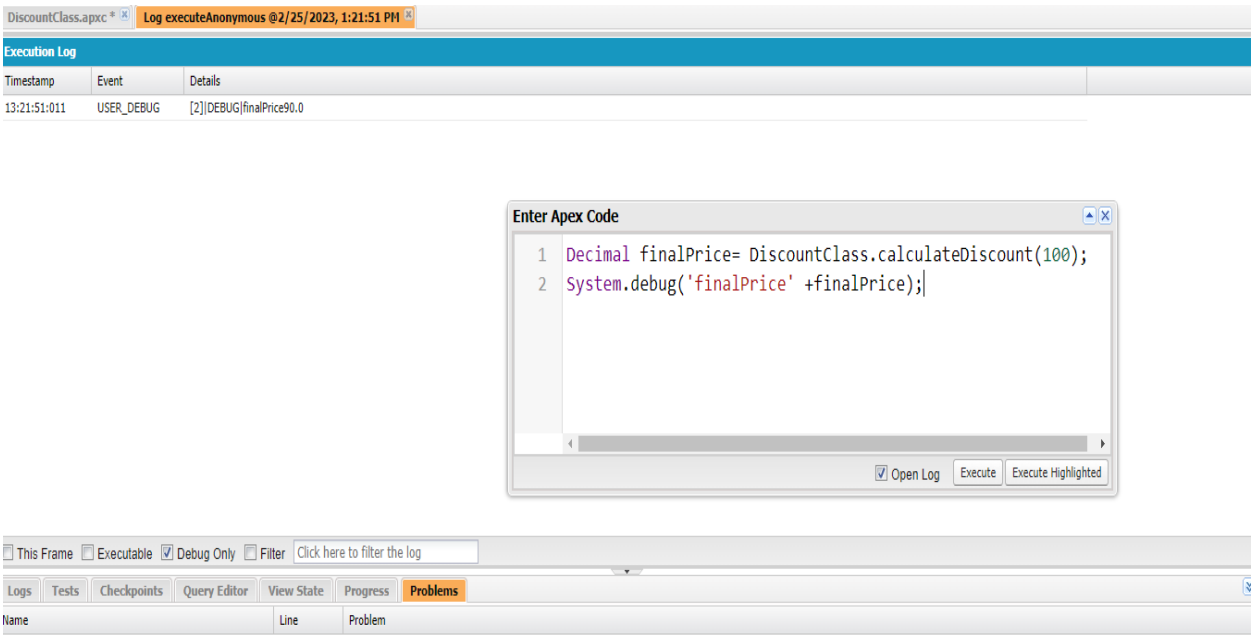
Output :



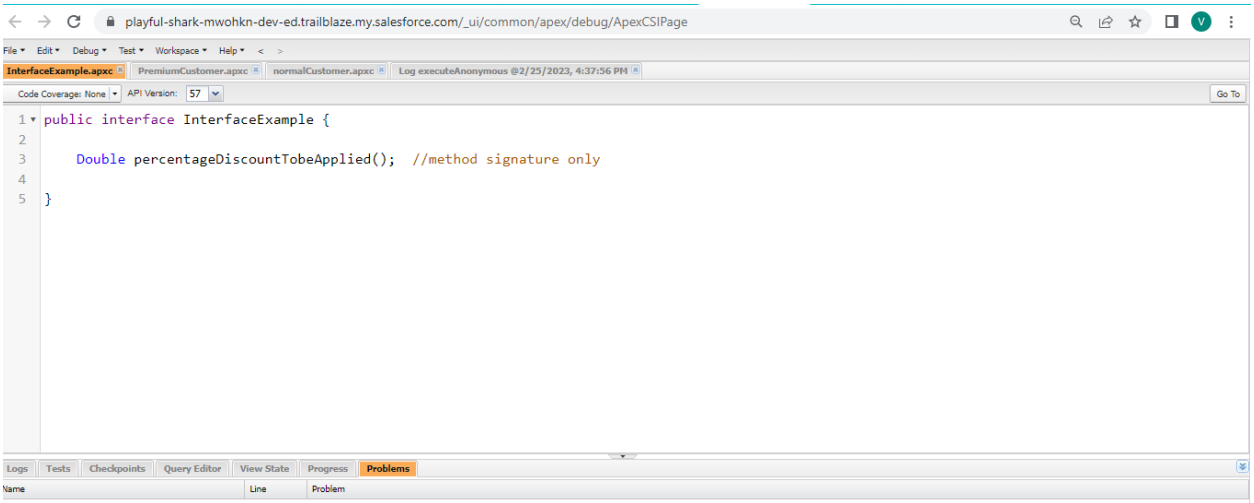
Exercise 8 : - Write a Class to demo the use of Constants in Apex



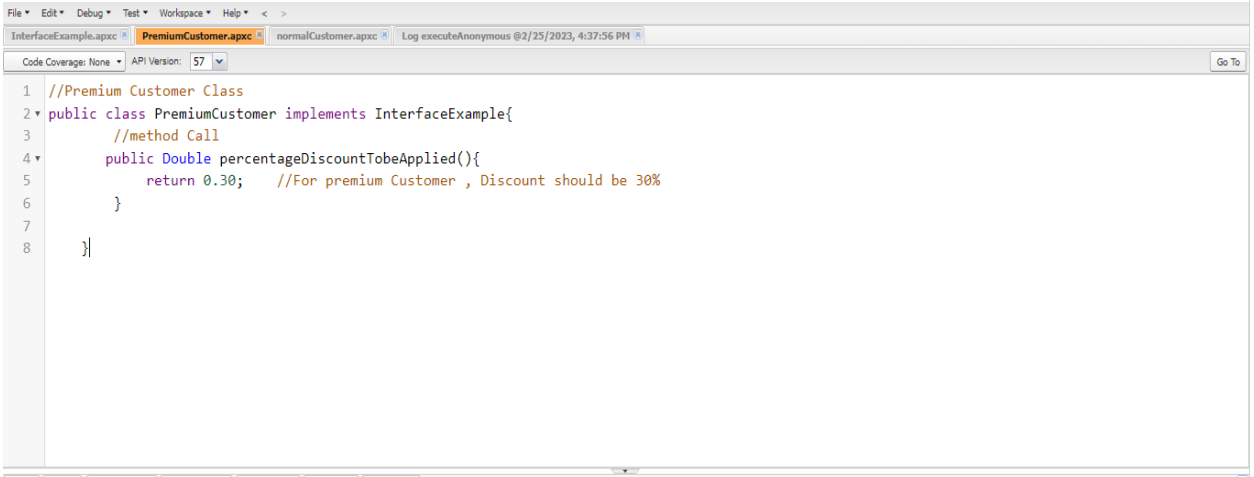
Output :



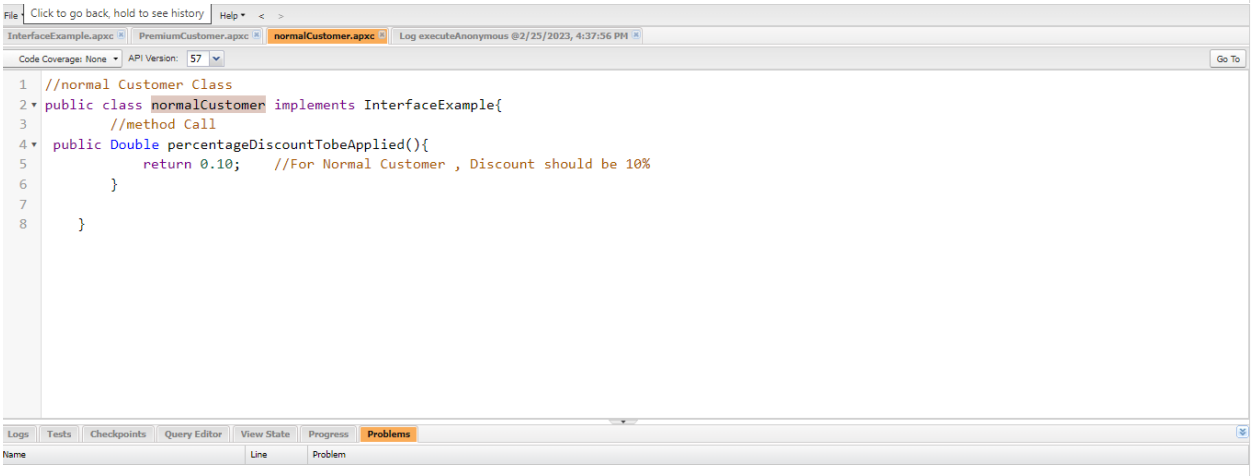
Exercise 9 : - Write a Class to demo the use of Interface in Apex



```
public interface InterfaceExample {

    Double percentageDiscountTobeApplied();  //method signature only

}
```

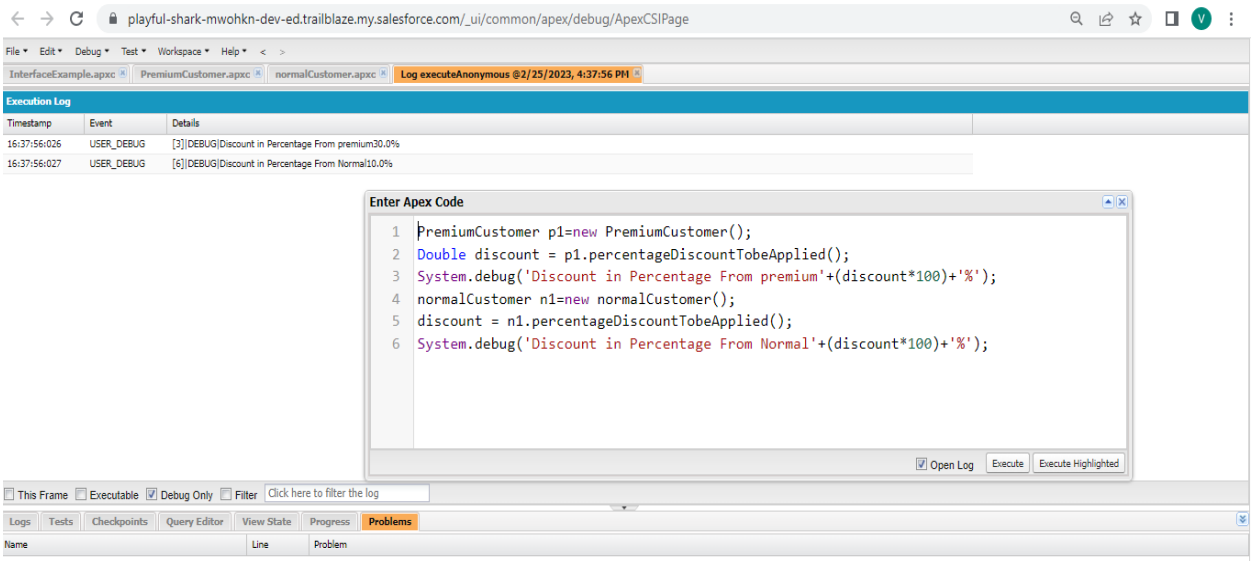


```
//Premium Customer Class
public class PremiumCustomer implements InterfaceExample{
        //method Call
       public Double percentageDiscountTobeApplied(){
            return 0.30;    //For premium Customer , Discount should be 30%
        }

    }
```

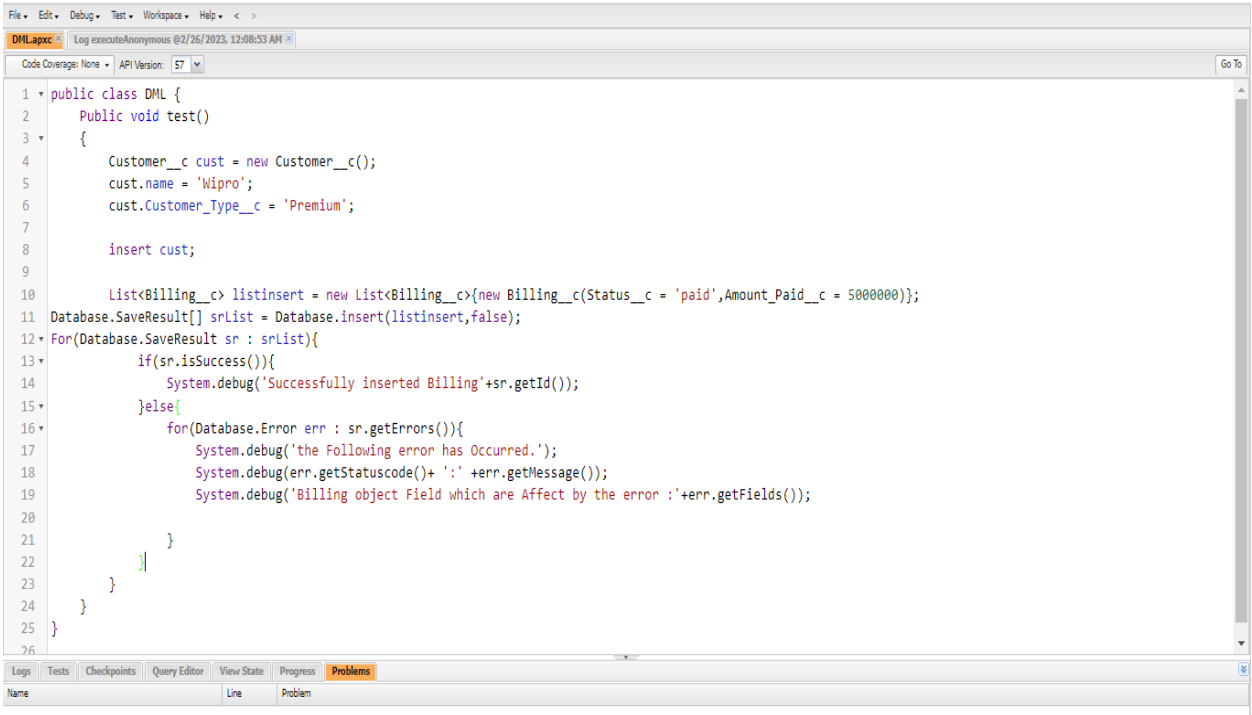


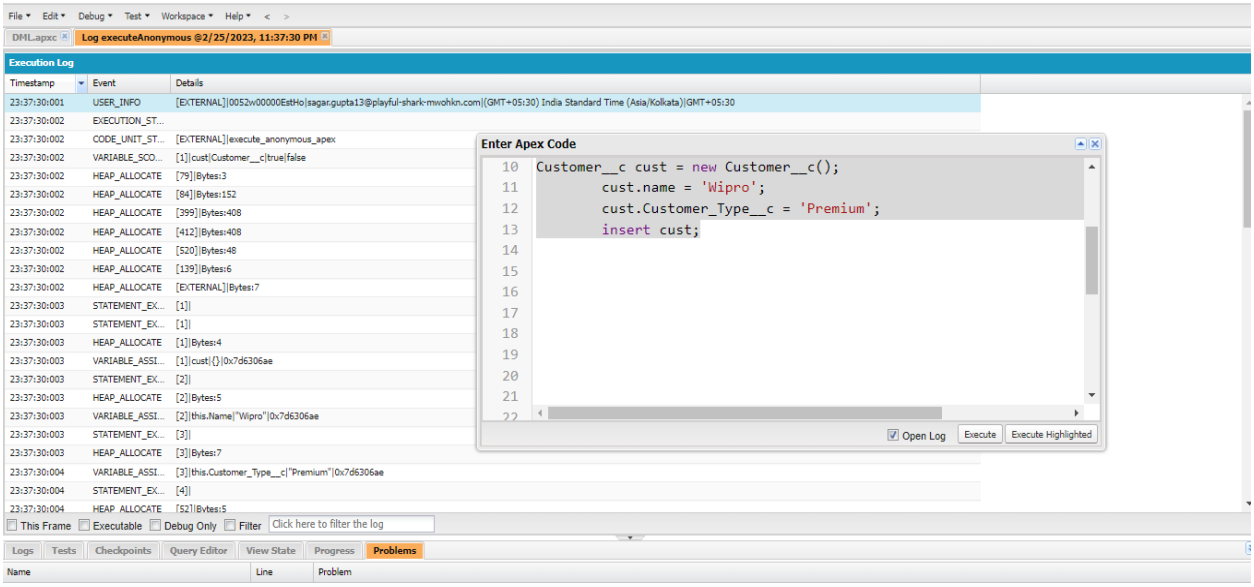
```
//normal Customer Class
public class normalCustomer implements InterfaceExample{
        //method Call
 public Double percentageDiscountTobeApplied(){
            return 0.10;    //For Normal Customer , Discount should be 10%
        }

    }
```

Output :

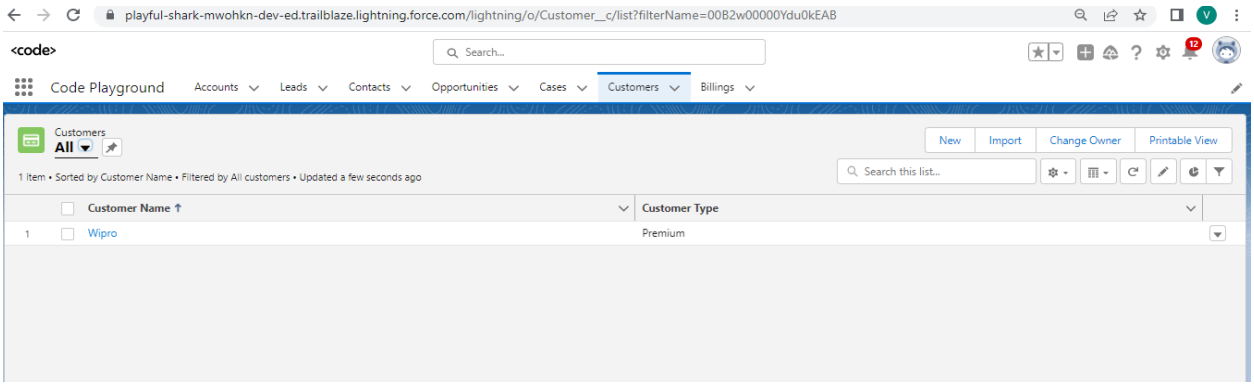


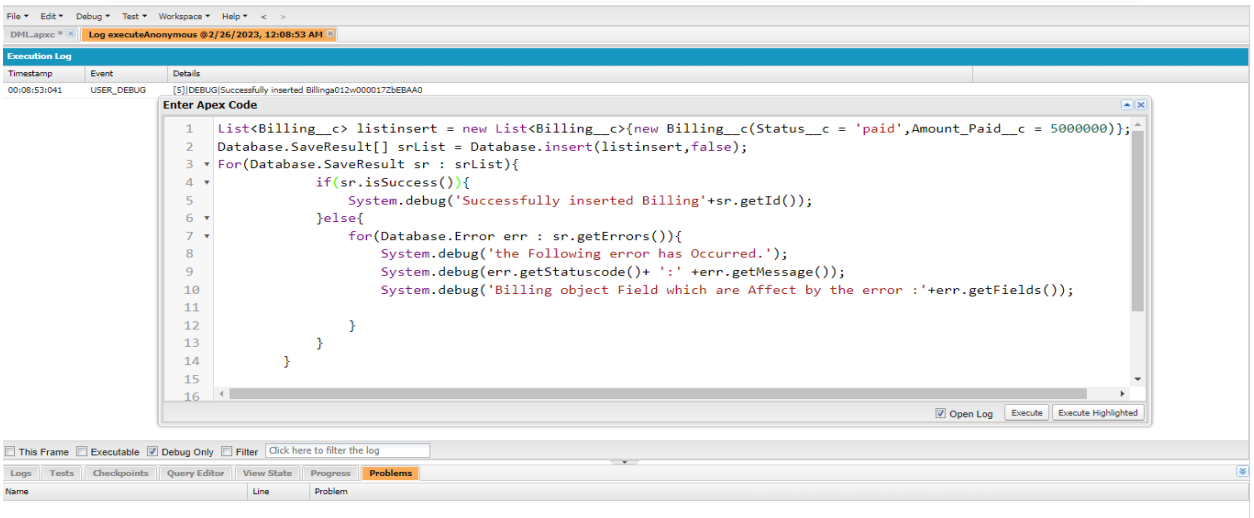
Exercise 10 : - Demo on DML Insert Operation Using Database methods



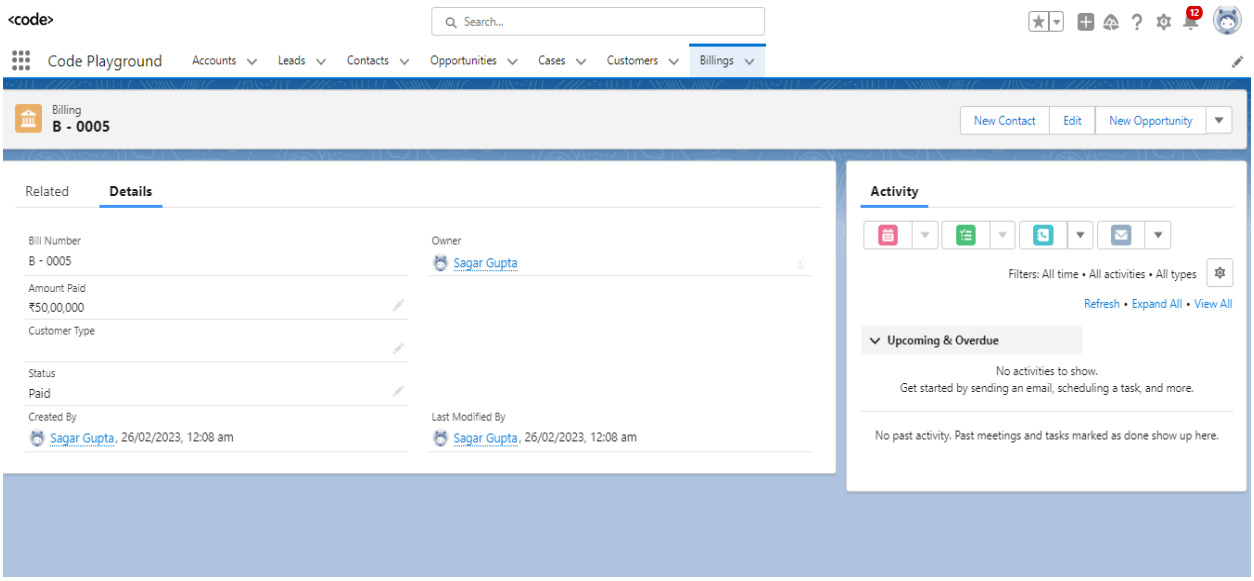

Log executeAnonymous @2/25/2023, 11:37:30 PM
Execution Log
Enter Apex Code
Customer__c cust = new Customer__c();
cust.name = 'Wipro';
cust.Customer_Type__c = 'Premium';
insert cust;
Open Log
Execute
Execute Highlighted

Output :


playful-shark-mwohkn-dev-ed.trailblaze.lightning.force.com/lightning/o/Customer__c/list?filterName=00B2w00000Ydu0kEAB
Code Playground
Accounts
Leads
Contacts
Opportunities
Cases
Customers
Billings
Customers
All
New
Import
Change Owner
Printable View
Customer Name
Customer Type
Wipro
Premium

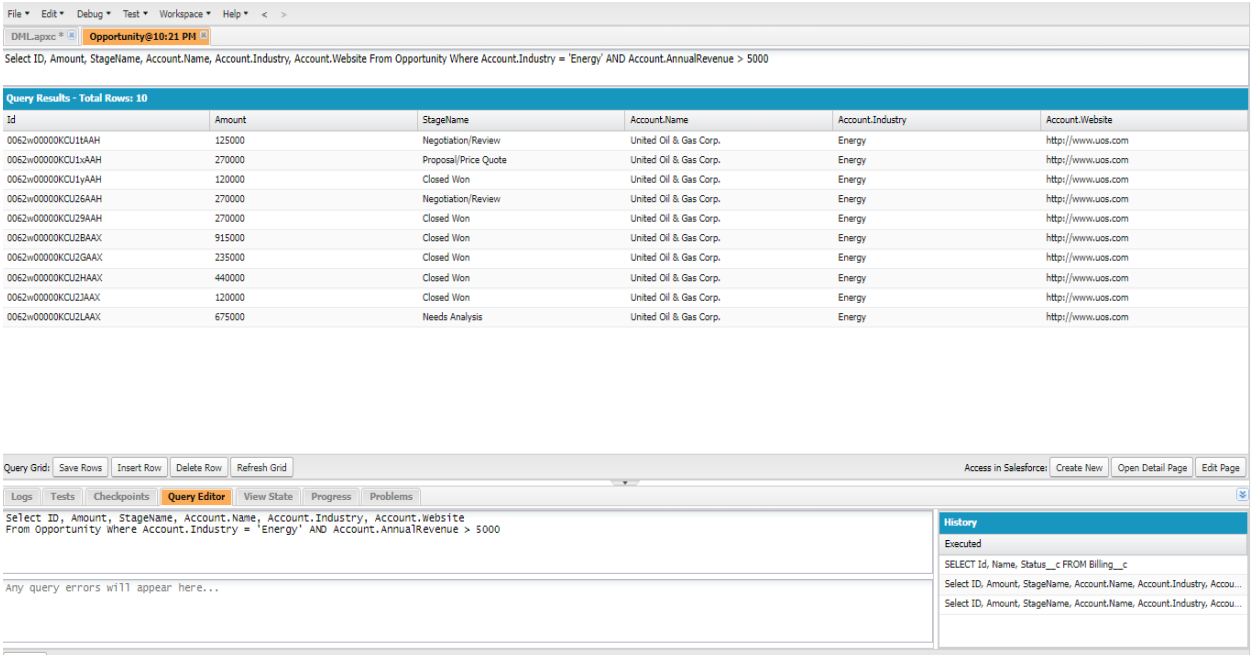

Log executeAnonymous @2/26/2023, 12:08:53 AM
Execution Log
Enter Apex Code
List<Billing__c> listinsert = new List<Billing__c>{new Billing__c(Status__c = 'paid',Amount_Paid__c = 5000000)};
Database.SaveResult[] srList = Database.insert(listinsert,false);
For(Database.SaveResult sr : srList){
if(sr.isSuccess()){
System.debug('Successfully inserted Billing'+sr.getId());
}else{
for(Database.Error err : sr.getErrors()){
System.debug('the Following error has Occurred.');
System.debug(err.getStatuscode()+ ':' +err.getMessage());
System.debug('Billing object Field which are Affect by the error :'+err.getFields());
}
}
}
Open Log
Execute
Execute Highlighted

Output :

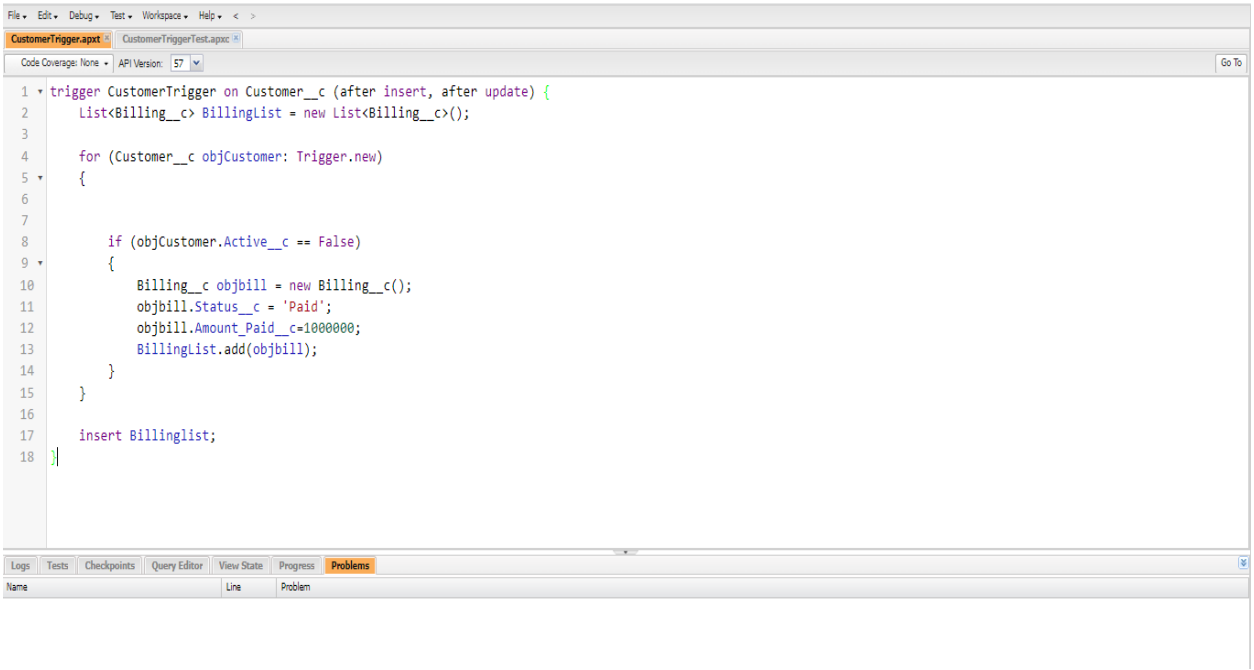


Exercise 11 :- Write and execute SOQL queries from Developer Console.

Output :



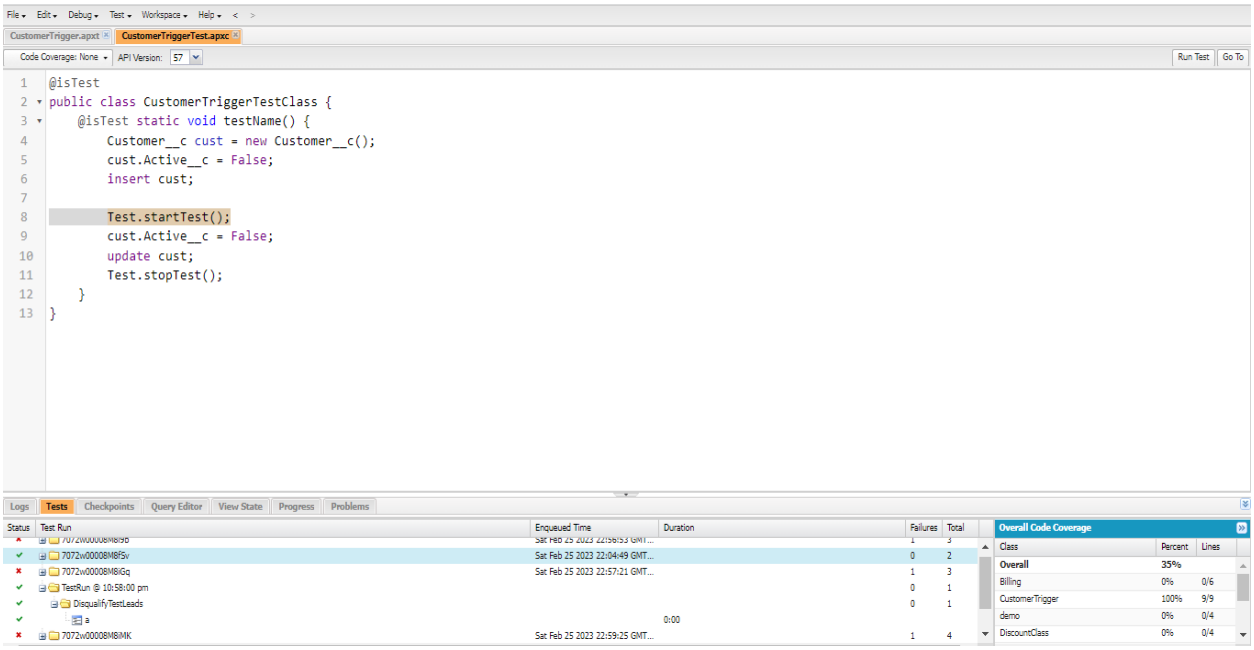
Exercise 12 :- Write an Apex Trigger, Name = CustomerTrigger.



```
trigger CustomerTrigger on Customer__c (after insert, after update) {
    List<Billing__c> BillingList = new List<Billing__c>();

    for (Customer__c objCustomer: Trigger.new)
    {


        if (objCustomer.Active__c == False)
        {
            Billing__c objbill = new Billing__c();
            objbill.Status__c = 'Paid';
            objbill.Amount_Paid__c=1000000;
            BillingList.add(objbill);
        }
    }

    insert Billinglist;
}
```

Exercise 13 :- Write a Test Class for Customer Trigger.

Output :



```
@isTest
public class CustomerTriggerTestClass {
    @isTest static void testName() {
        Customer__c cust = new Customer__c();
        cust.Active__c = False;
        insert cust;

        Test.startTest();
        cust.Active__c = False;
        update cust;
        Test.stopTest();
    }
}
```

Exercise 14 :- Write an Apex Trigger, Name = DisqualifyTestLeads.

DisqualifyTestLeads.apxt | TestDisqualifyTestLeads.apxc

Code Coverage: None | API Version: 57 | Go To

```
trigger DisqualifyTestLeads on Lead (before insert) {
     List<Lead> llist = new List<Lead>();
    for(Lead le:Trigger.new)
    {
      if(le.FirstName.containsIgnoreCase('test')|| string.isBlank(le.FirstName)
         || le.LastName.containsIgnoreCase('test')||string.isBlank(le.LastName))
        {
            system.debug(le.FirstName + ' ' + le.LastName + 'Will be disqualified!');
            llist.add(le);
        }
    }
    for(Lead l :llist){
        l.status='Disqualified';
    }
}
```

Exercise 15 :- Write a Test Class for DisqualifyTestLeads.

Output :

DisqualifyTestLeads.apxt | TestDisqualifyTestLeads.apxc

Code Coverage: None | API Version: 57 | Run Test | Go To

```
@isTest
public class DisqualifyTestLeads {
@isTest static void a()
    {
        Lead le=new Lead();
        le.FirstName = 'Role';
        le.LastName = 'Test';
        le.Company = 'Wipro';


        Test.startTest();
        insert le;
        Test.stopTest();
    }
}
```

Logs | Tests | Checkpoints | Query Editor | View State | Progress | Problems 1

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7072w00008M8f5v	Sat Feb 25 2023 22:04:49 GMT...		0	2
✗	7072w00008M8iGq	Sat Feb 25 2023 22:57:21 GMT...		1	3
✓	TestRun @ 10:58:00 pm			0	1
✓	DisqualifyTestLeads			0	1
✓	a		0:00		
✗	7072w00008M8iMK	Sat Feb 25 2023 22:59:25 GMT...		1	4
✓	TestRun @ 11:04:22 pm			0	1

Overall Code Coverage

Class	Percent	Lines
Overall	20%	
Billing	0%	0/6
demo	0%	0/4
DiscountClass	0%	0/4
DisqualifyTestLeads	100%	8/8
DML	0%	0/13

Exercise 16 :- Create a Visualforce page which displays Opportunity fields as output fields.

Opportunity.vfp

Preview API Version: 57

```
<apex:page standardController = "Opportunity">

    <apex:pageBlock title = "Opportunities">
        <apex:pageBlockSection >
            <apex:outputField value="{! Opportunity.Name}"/>
            <apex:outputField value="{! Opportunity.Amount}"/>
            <apex:outputField value="{! Opportunity.CloseDate}"/>
            <apex:outputField value="{! Opportunity.Account.Name}"/>

        </apex:pageBlockSection>
    </apex:pageBlock>
</apex:page>
```

Logs | Tests | Checkpoints | Query Editor | View State | Progress | Problems

Name | Line | Problem

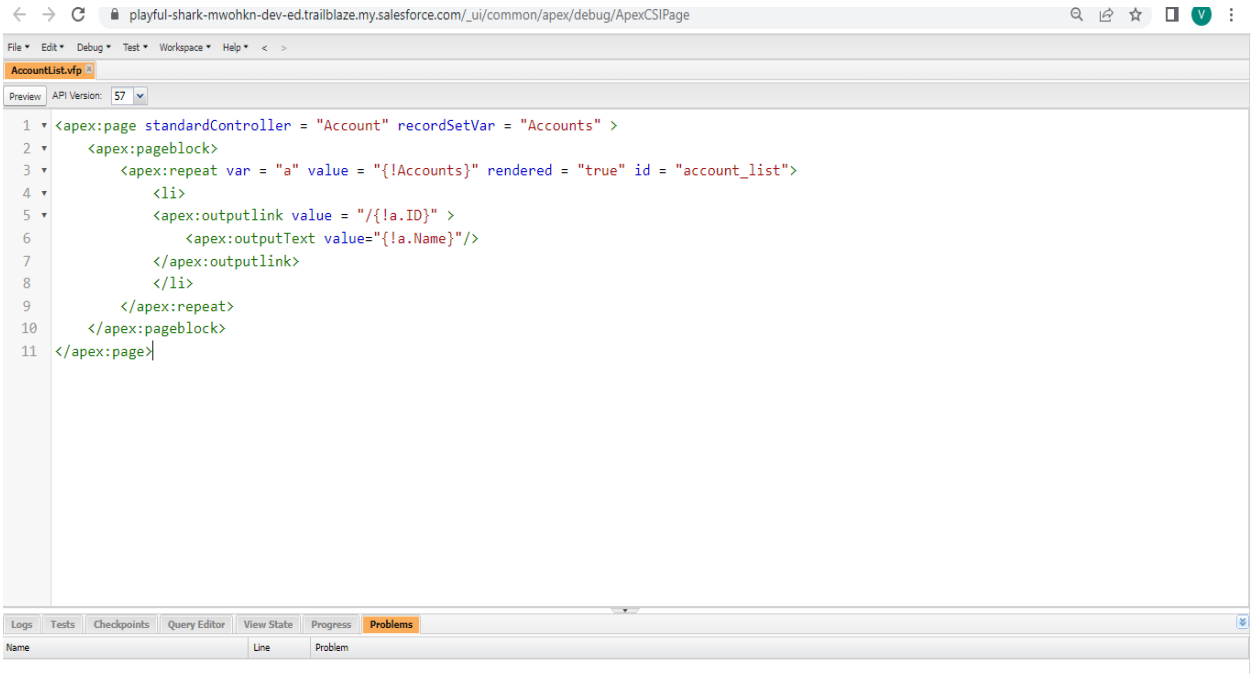
Output :

playful-shark-mwohkn-dev-ed--c.trailblaze.vf.force.com/apex/Opportunity?core.apexpages.request.devconsole=1

Opportunities

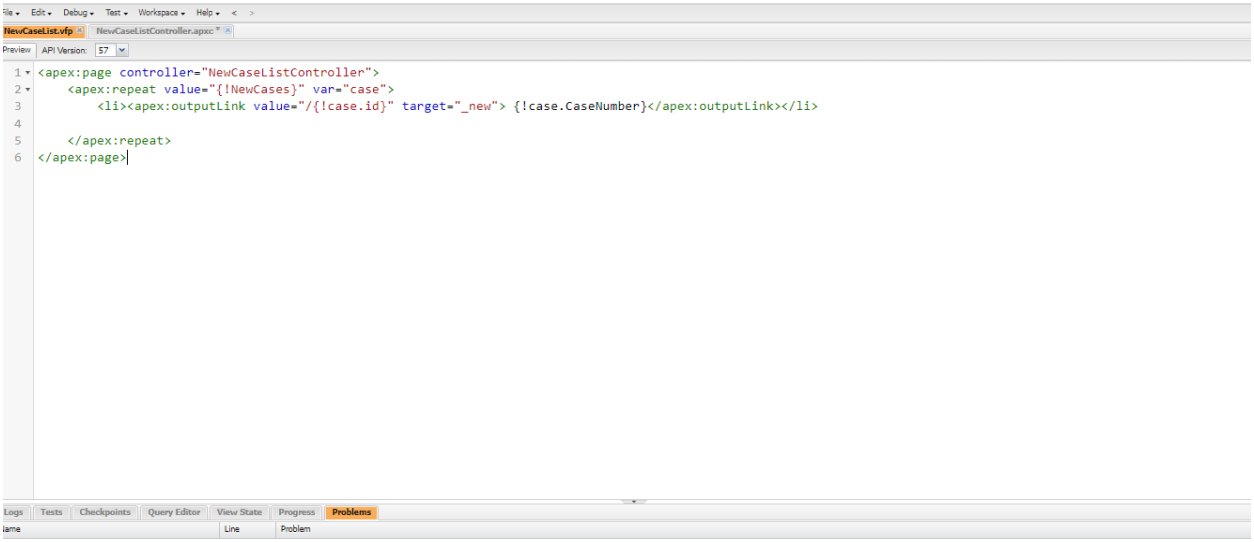
Opportunity Name	Amount
Close Date	Account Name

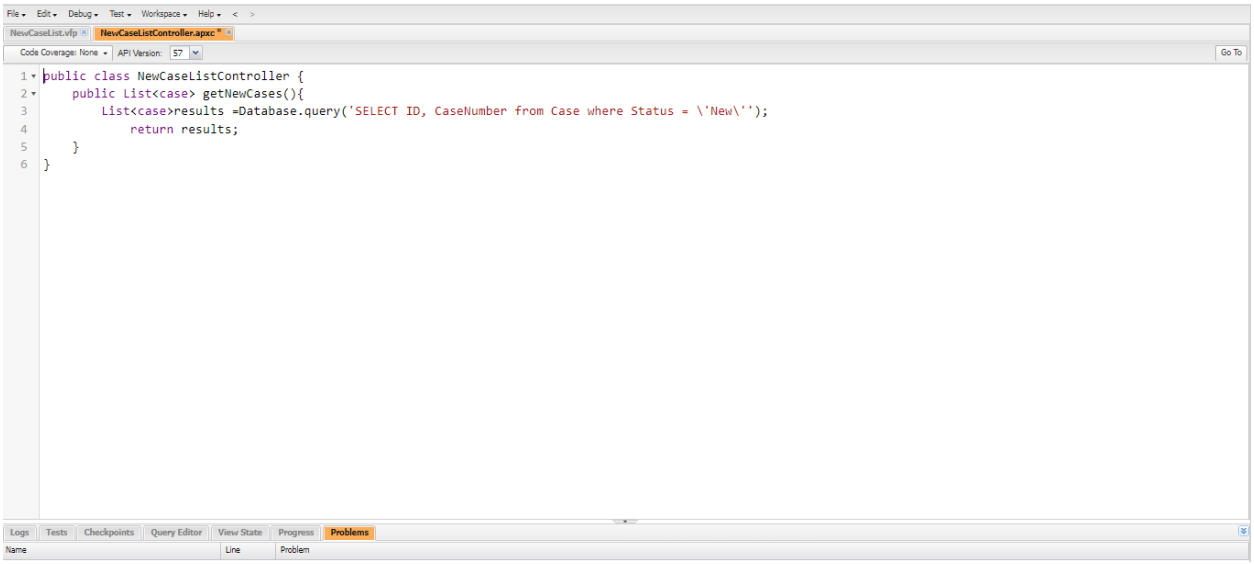
Exercise 17 :- Create a Visualforce page which shows a list of Accounts linked to their record page.



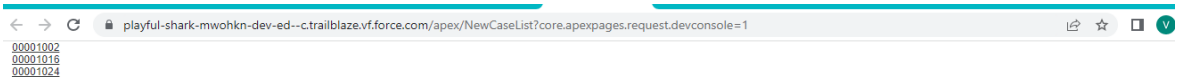
Output :

Exercise 18 :- Create a Visualforce page that uses a custom controller to display a list of cases with the status of 'New'. The page must be named NewCaseList.





Output :



References

1.Trailhead: https://trailhead.salesforce.com/

2. Developer Guide: Salesforce Developers

3.Manage sales - Salesforce IN

4.Salesforce - ADX201 Administrative Essentials for New Admins in Lightning Experience (SFADX201) (qa.com)

5.Understand the Salesforce Architecture Unit | Salesforce Trailhead