Capstone Project Report

On

"Travel Application and Code Playground Application"

Submitted by:

Name: Nadavaluru Medhasi

Batch WIP-SF-13

LMS Id: MGSA_516

Date: 27/02/2023

Github Link: https://github.com/Medhasi2001/Capstone-Project.git

Table of contents

1. Abstract

2. Introduction

3. Software Requirements

4. Screen shots

    4.1 Module 1

    4.2 Module 2

    4.3 Module 3

5. References

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.Here we mainly work on Travel Application and Code Playground Application.

Introduction

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.

Business Requirements

Module 1:

Set up a Lightning app to streamline the travel approval process.
You need to create a custom application that meets these requirements:
Build a Lightning app, add tabs, and customize page layouts.
Create custom objects and fields for the app.
Define relationships between objects.
Import data.
Create Travel Approvals, Travel Expenses & Users.
Customize List views, Page Layouts,Related Lists.
Setup Approval settings for User.
Enable Chatter on the Travel Approval Object.

Module 2:

Add Business logic & Analytics to the Travel Approval App.
You will be adding Business logics, Reports and Dashboards to:
Enforce data integrity with validation rules.
Define business logic using roll-up summary and formula fields
Automate business processes with Flows.
Define Approval Process for Travel Approval App.
Create Reports to analyze Travel Approvals.
Build Dashboards for Travel Approval Management.

Module 3:

Demonstrate your Apex programming skills to:
Write and execute codes from Anonymous window.
Demonstrate the use of Data types & Variables.
Demonstrate the use of Control Flow Statement.
Define relationships between objects. Data Manipulation using DML.
Querying the Data base using SOQL.
Create Apex Classes, Objects & Interfaces
Write Apex Triggers
Write Test Classes.
Create VisualForce Pages.

Software Requirements

- Platform : Salesforce developer edition
- Browser : Chrome/Safari/Mozilla etc
- Ram : 4 GB and above
- OS : windows/linux/mac

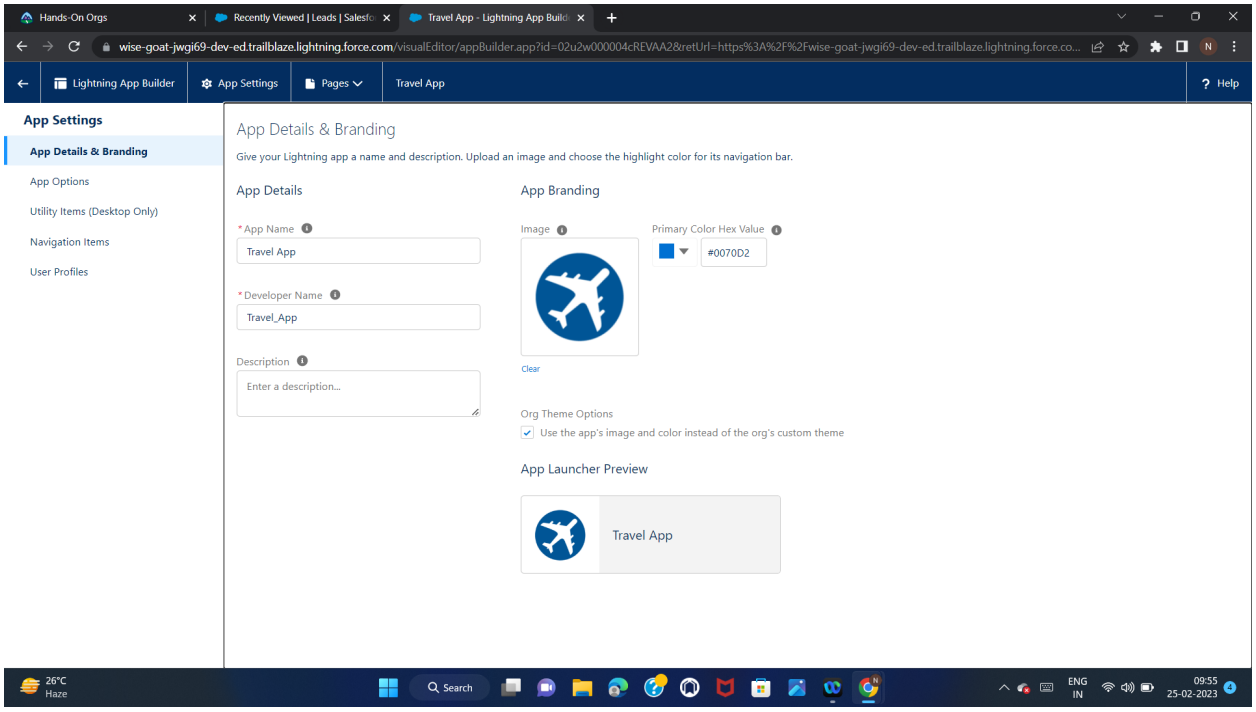
Screenshots

Module 1

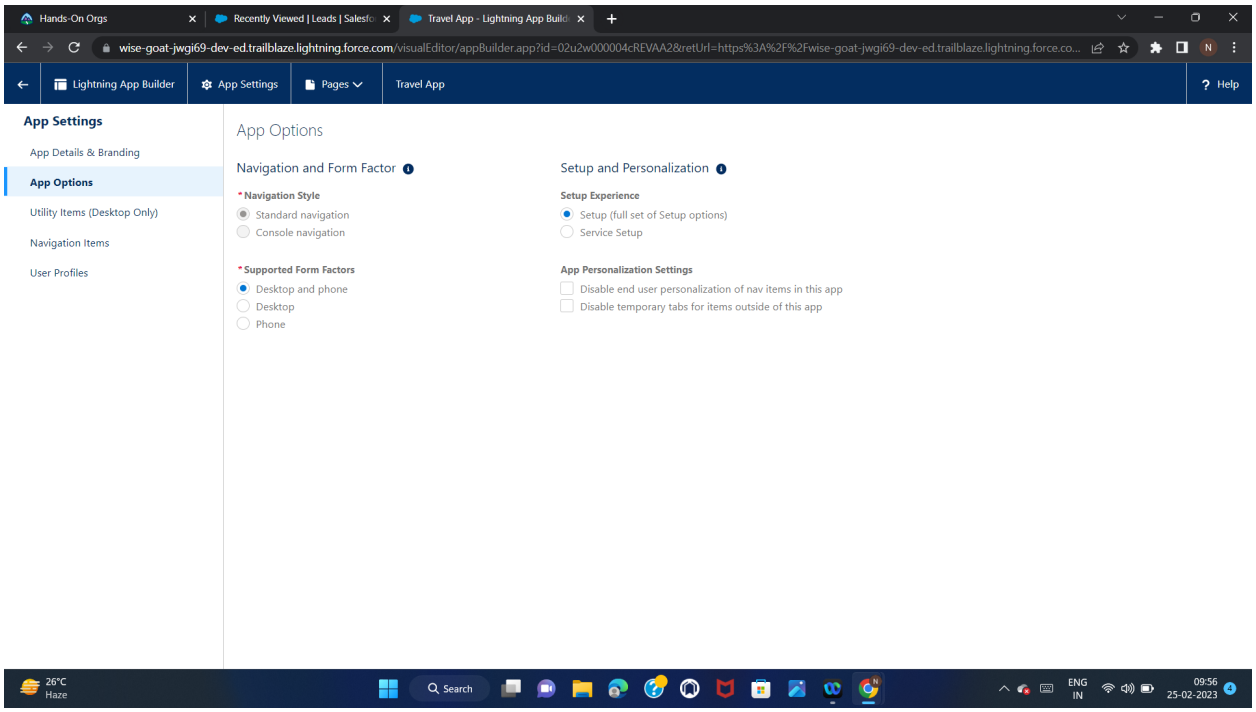
Exercise 1:

Step 1:

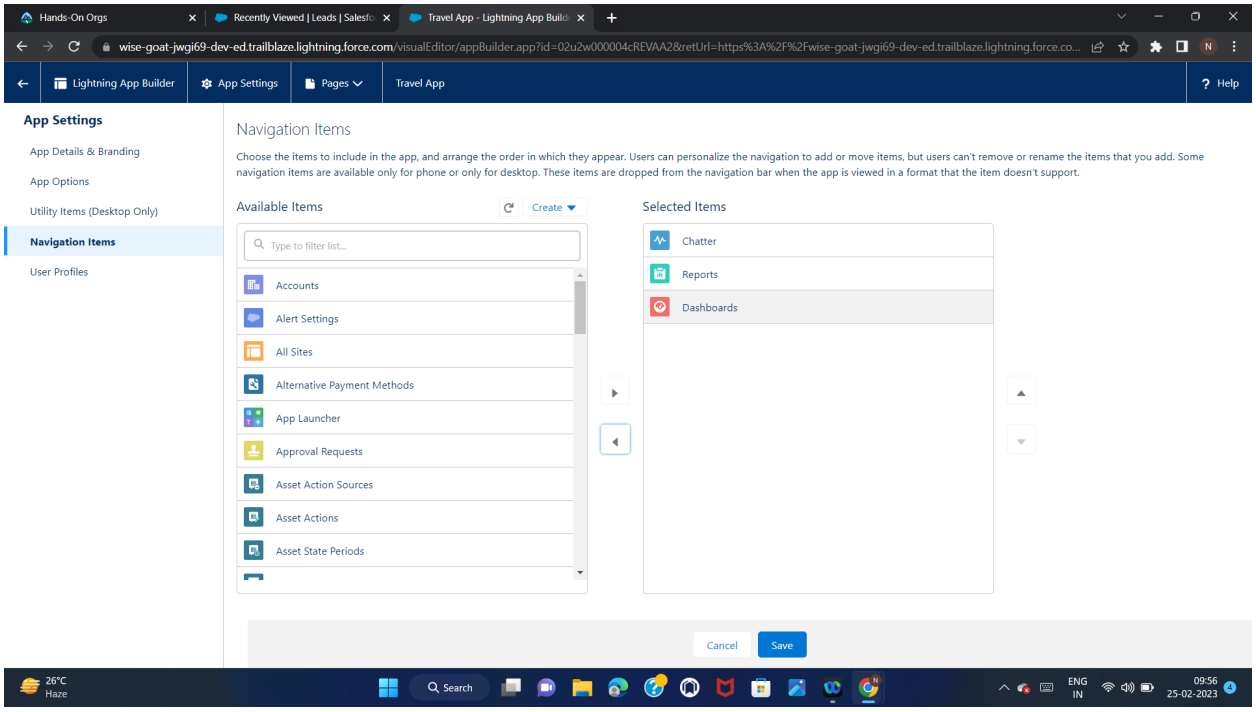
Create a new custom lightning App, name: Travel App.
Upload Travel.png as App image, Keep “Org Theme options” checked.



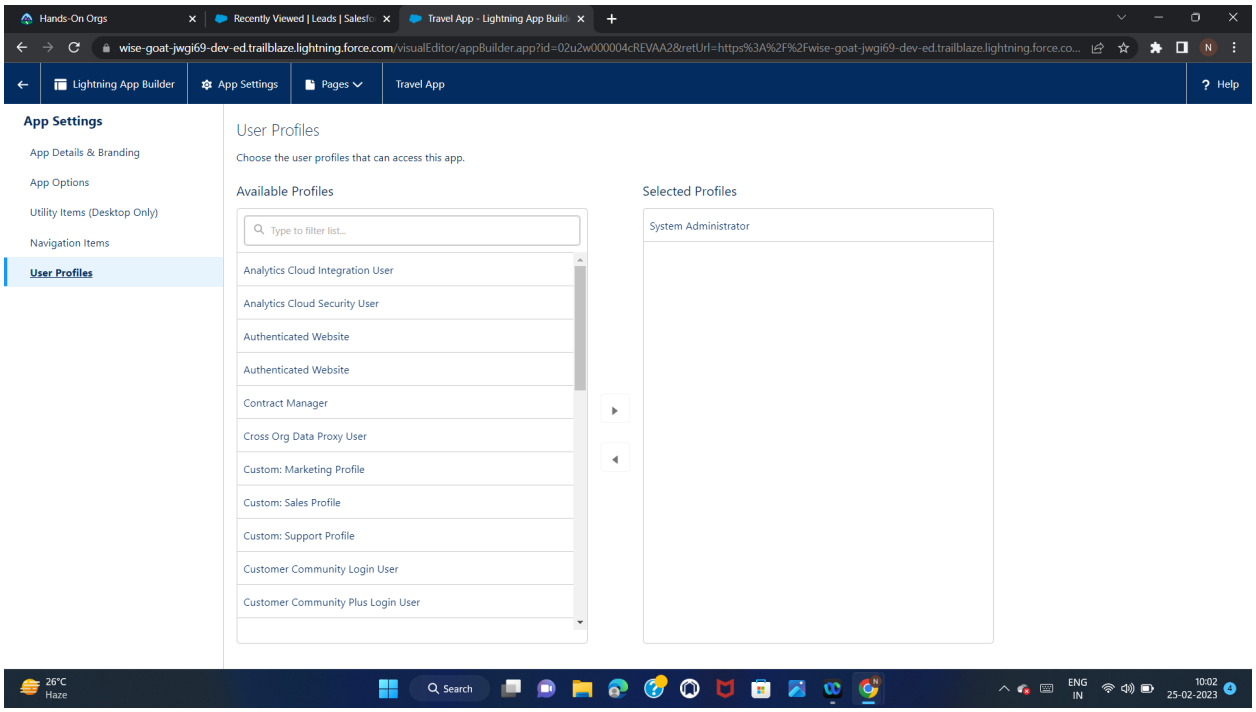
Navigation Style=”Standard Navigation”,leave rest as it is.



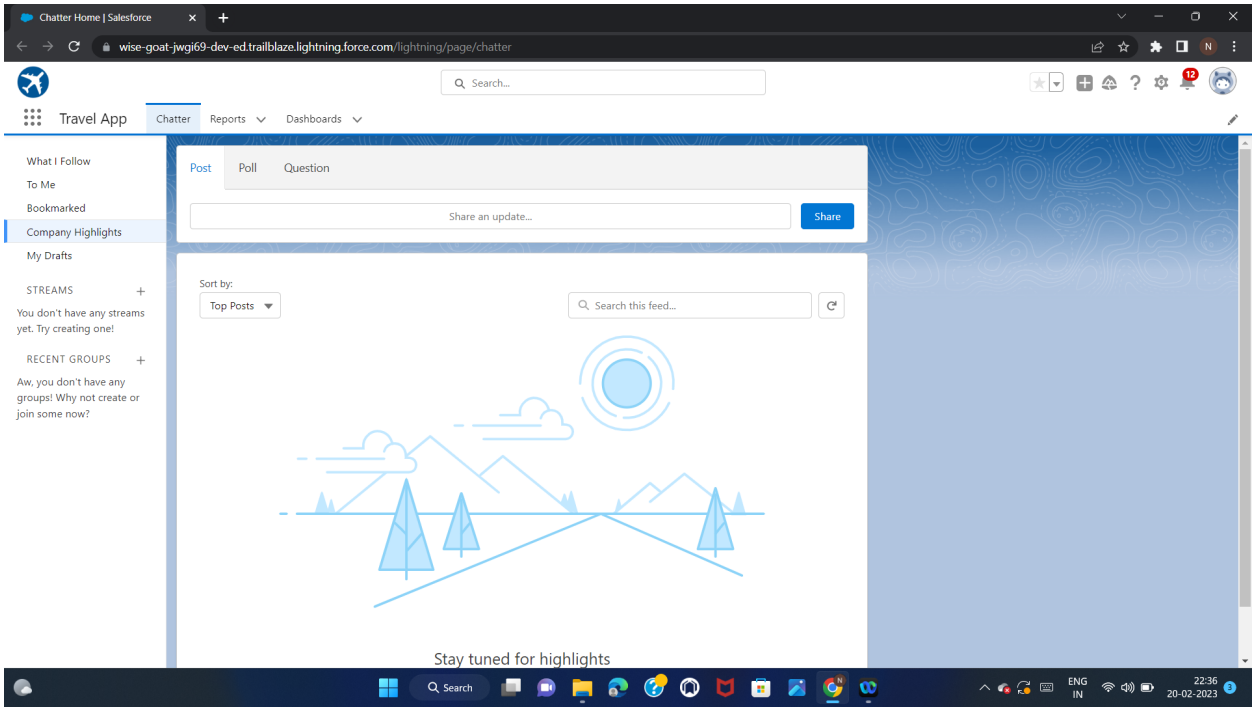
Select and add Chatter,Reports and dashboards to the Navigation items.



User Profile=”System Administrator”
Save and Finish

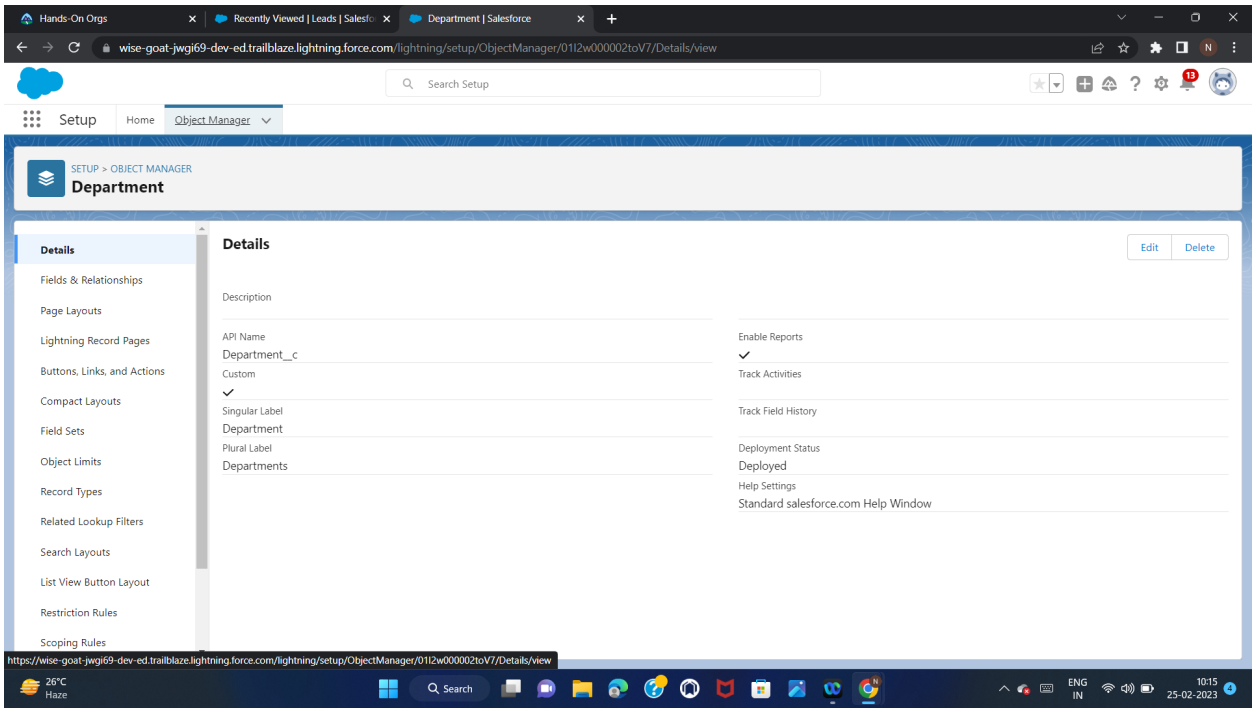


Travel App looks like this



Step 2:

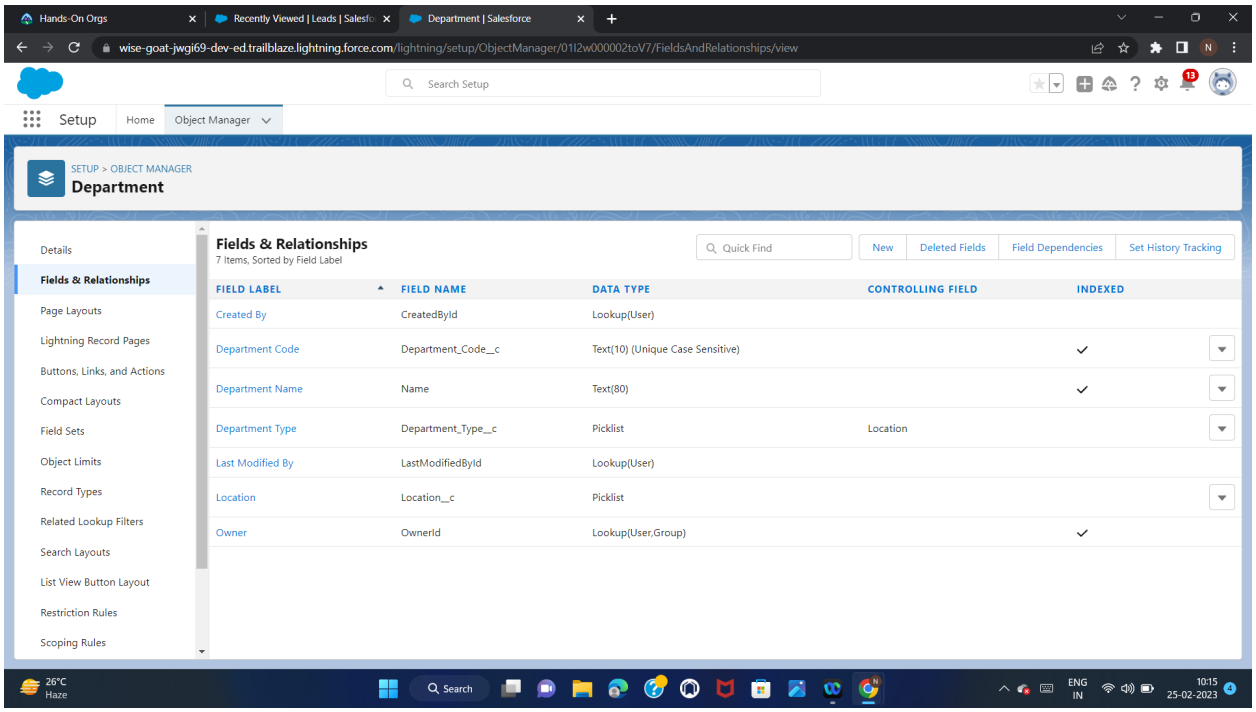
Create a Department custom object
Label=”Department”
Plural=”Departments”
Data type=”text”
Select “Allow Reports”,”Allow Search”
Keep other options as it is
Use custom tab and include it only for Travel App
Tab visibility=As it is,Save



Step 3:

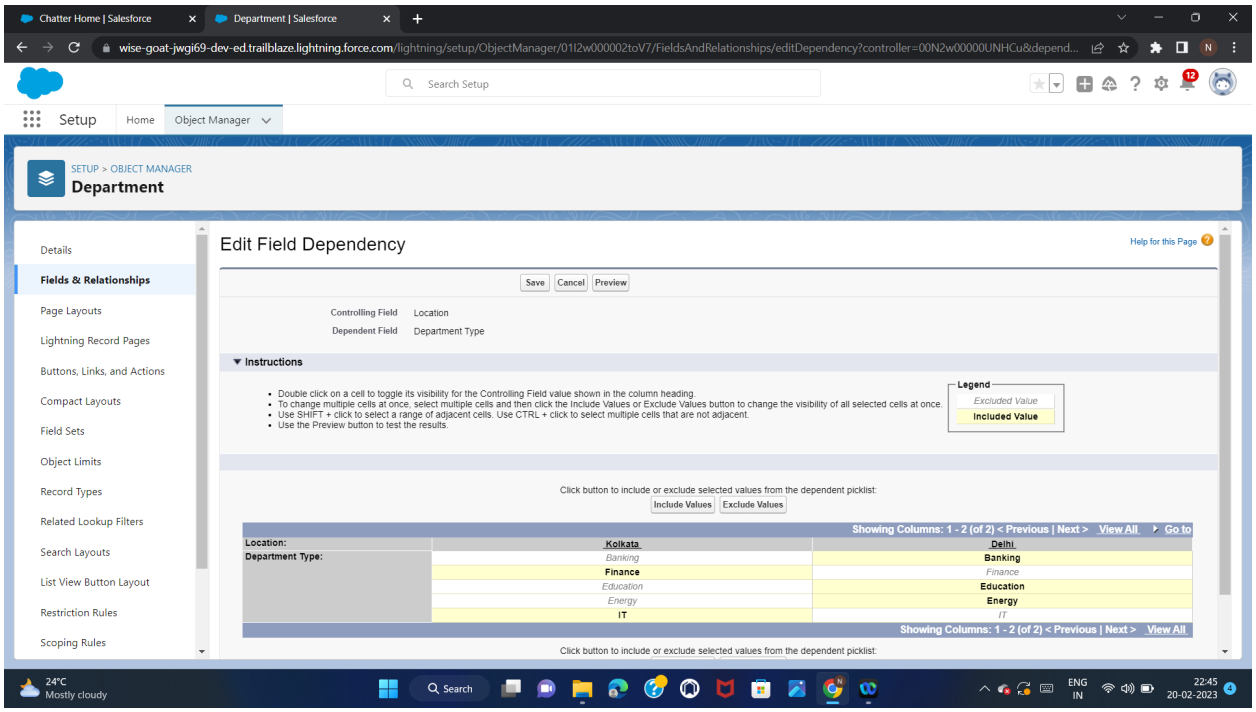
Create the following custom fields:

1.Department Code,Text ,length=10,Required,Select Unique and Case sensitive
2.Location,Picklist, Values:Kolkata,Delhi
3.Department Type,Picklist,Values:Banking,Finance,Education,Energy,IT



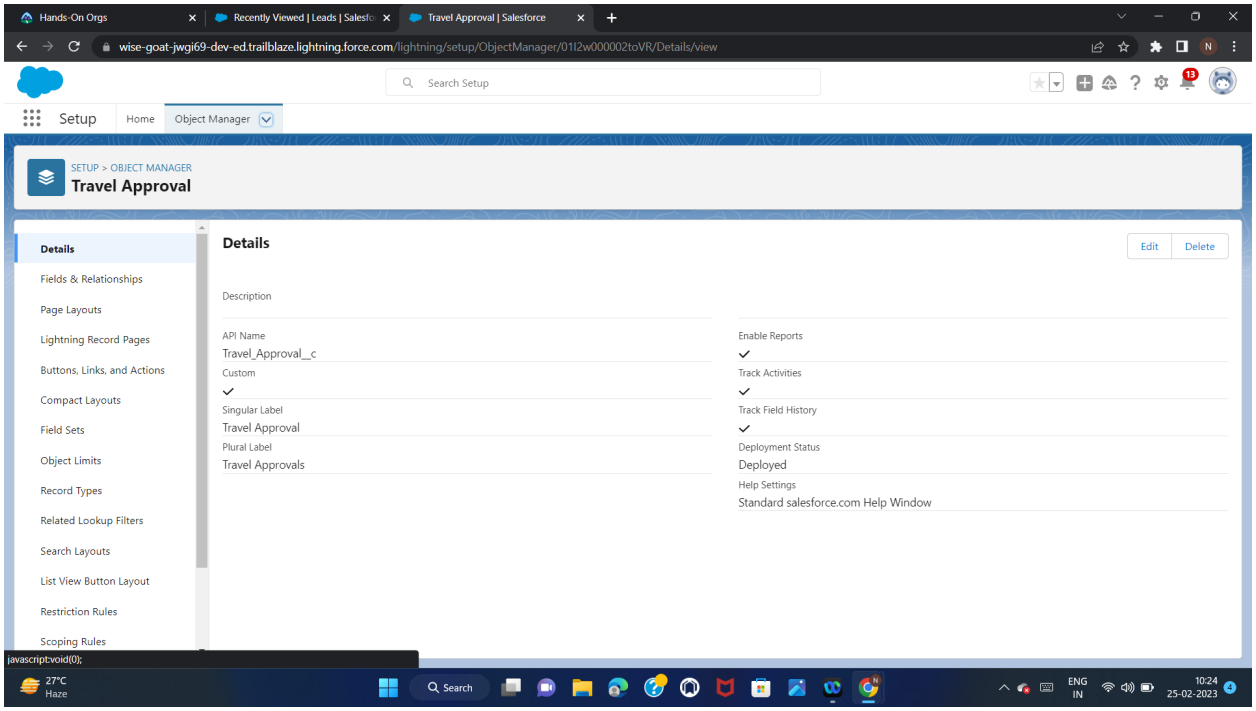
Create Field Dependency

Controlling field=Location,
Dependent field=Department Type



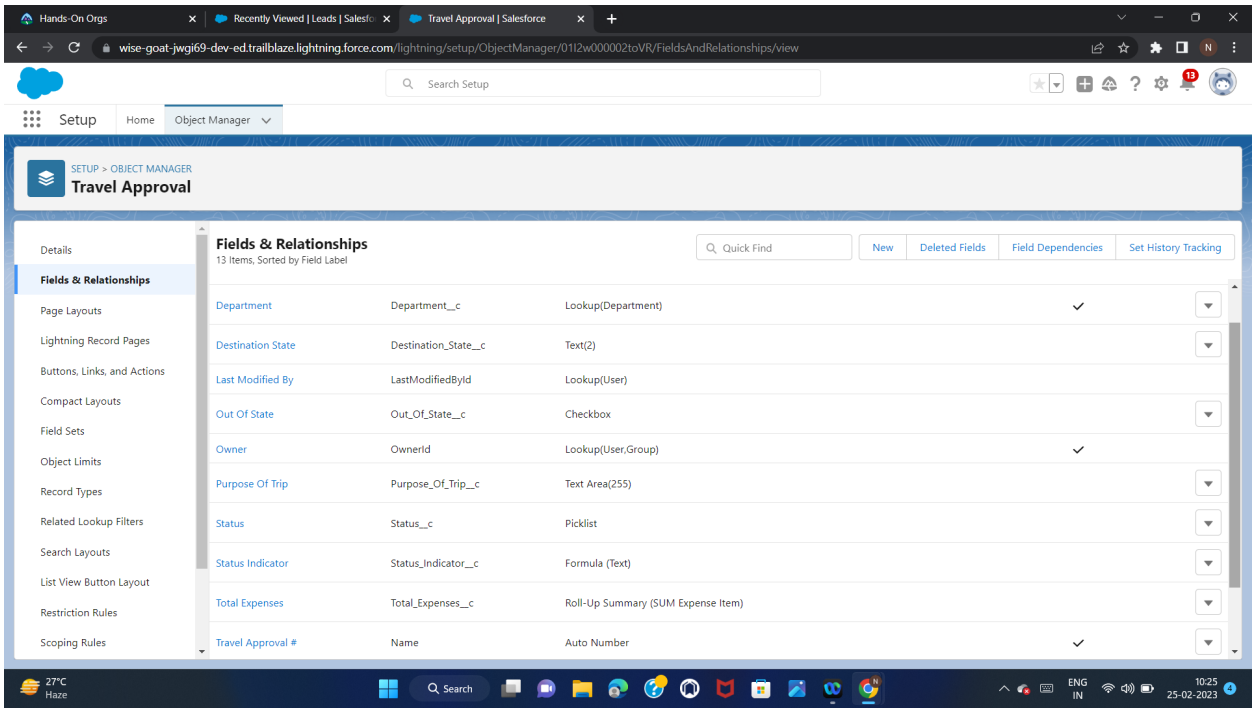
Step 4:

Create a Travel Approval Object.
Label=”Travel Approval”
Plural=”Travel Approvals”
Object Name=”Travel_Approval”
Record Name=”Travel Approval #”
Data Type=”Auto Number”
Display format “TA{00000}”
Starting Number=1
Allow Reports
Allow Activities
Track Field History
Allow Search
Add Notes and Attachments
Create a custom tab
Add the tab only to Travel App,Save

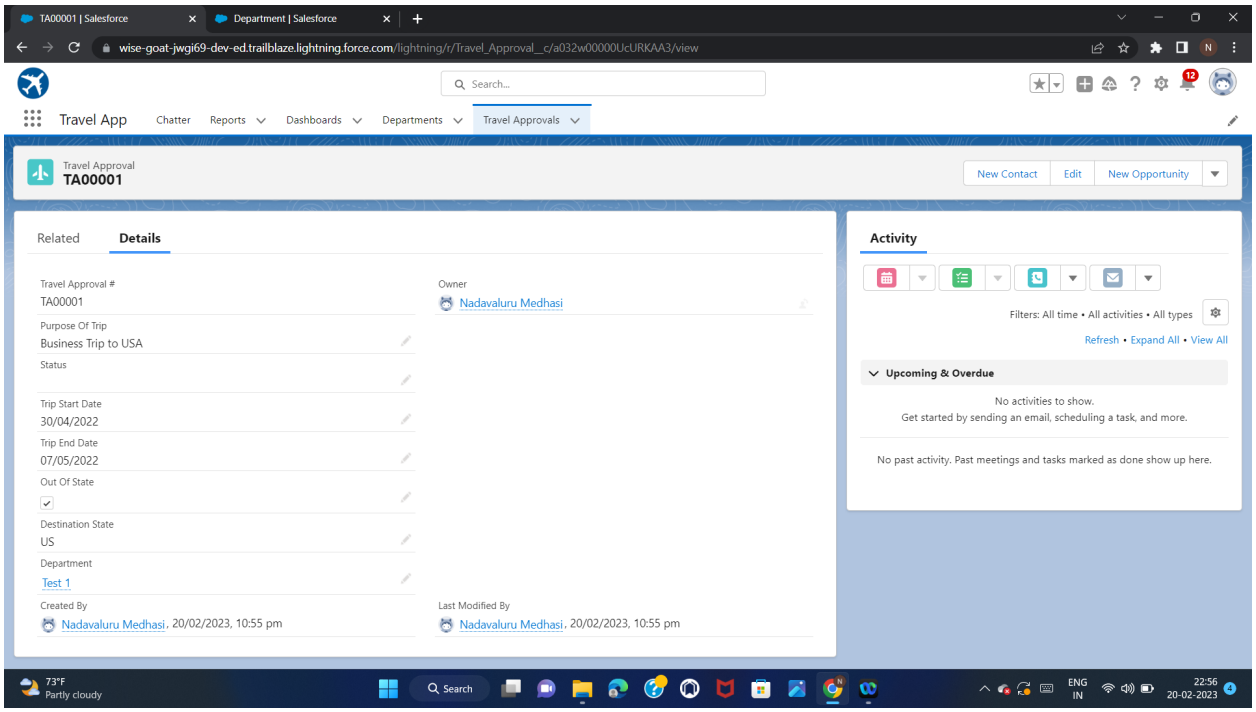


Create the Custom fields:

1.Purpose Of Trip,Text Area
2.Status,Picklist,
  Values=New,Submitted,Pending,Approval,Approved,Rejected,Draft
3.Trip Start Date,Date
4.Trip End Date,Date
5.Out Of State,Checkbox
6.Destination State,Text,Length=2
7.Department,Lookup,Related To=Department custom object

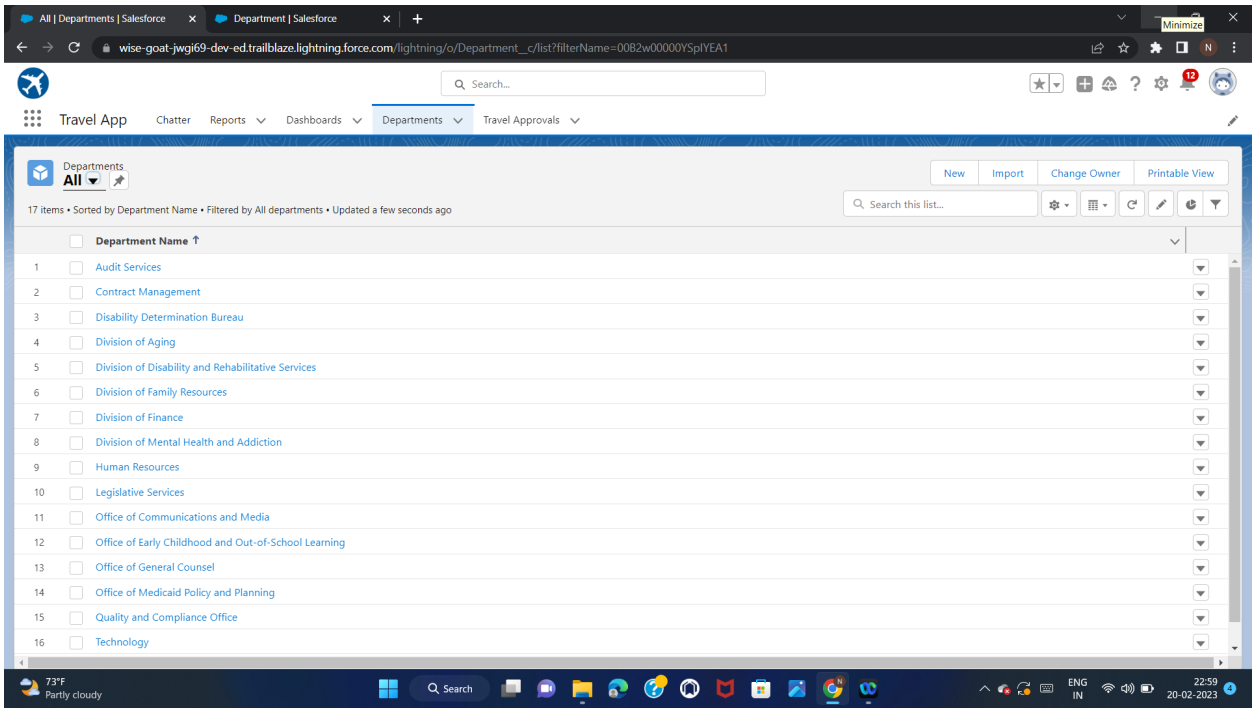


Travel App looks like this.



Step 5:

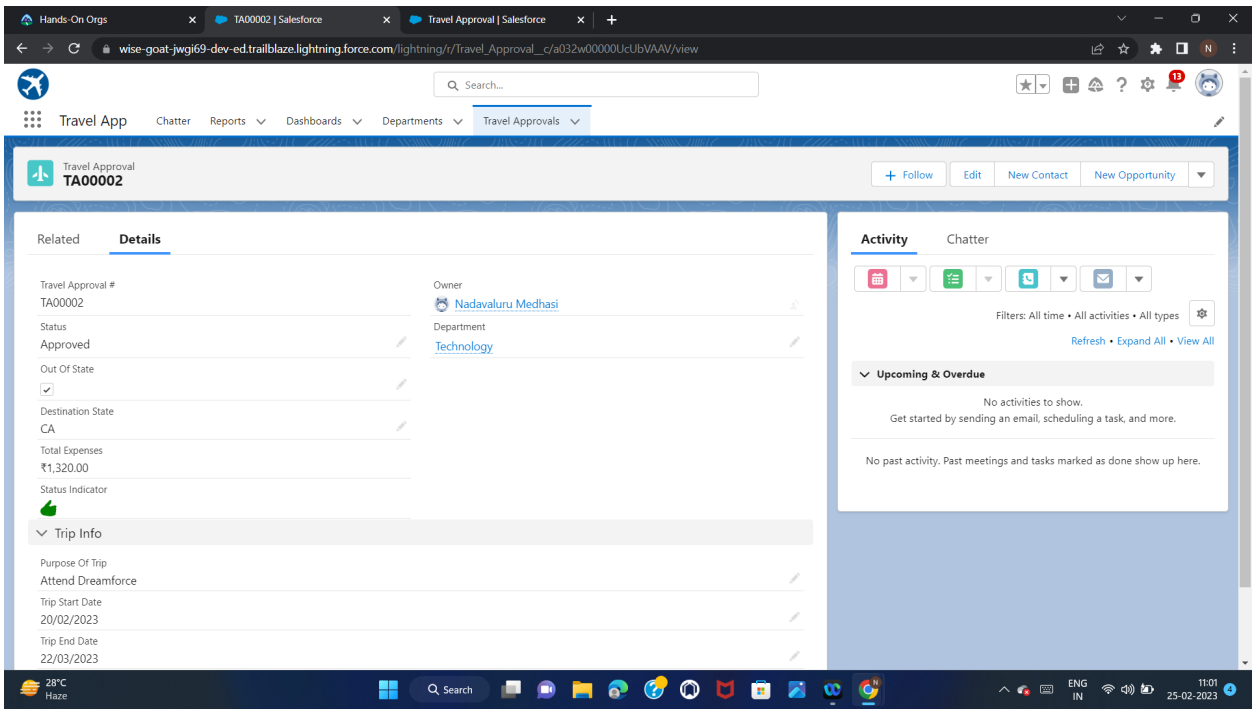
Import Departments
Download the Department.CSV File
Open “Data Import Wizard”,Click “Launch Wizard”
Select “Departments” custom object
Add new records ,upload the file by clicking the csv icon
Map the fields,click Next
Start Import
Make sure all the records are inserted.



Exercise 2:

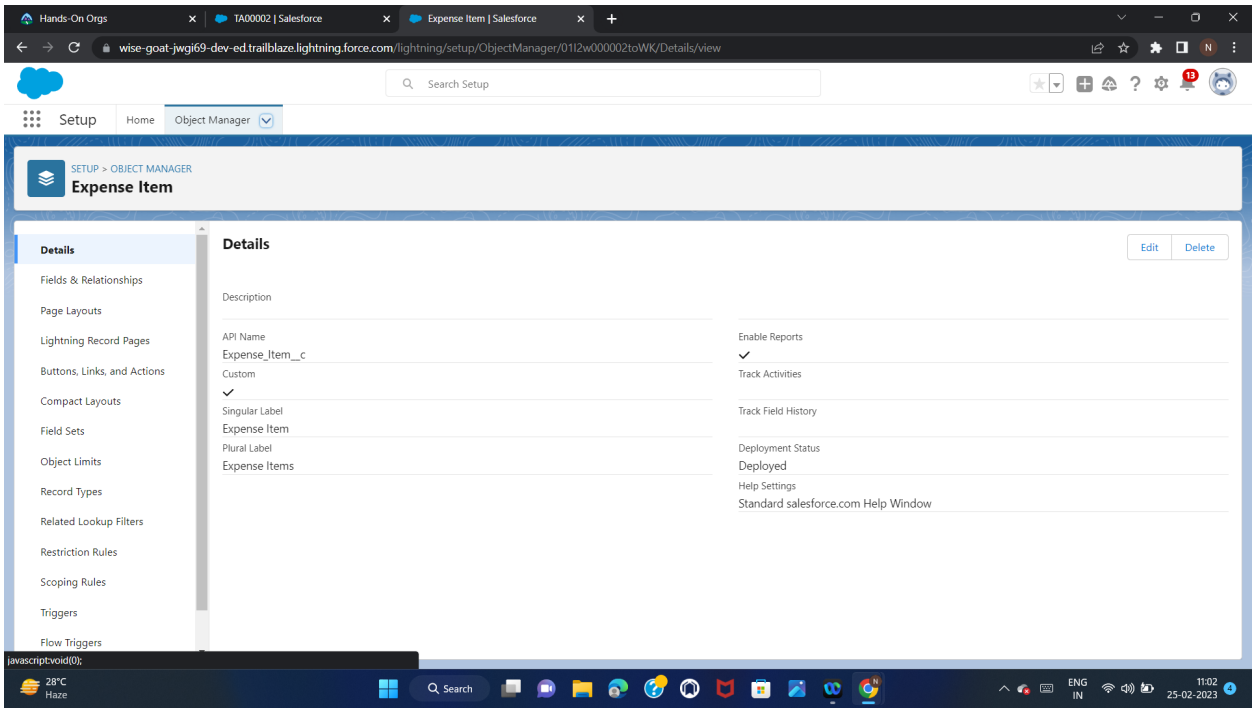
Step 1:

Create a Travel Approval record.
Purpose of Trip=”Attend Dreamforce”
Status=”Draft”
Trip Start Date/End Date=”Pick any Date”
Out Of State=True
Destination State=”CA”
Department=Technology,Save



Step 2:

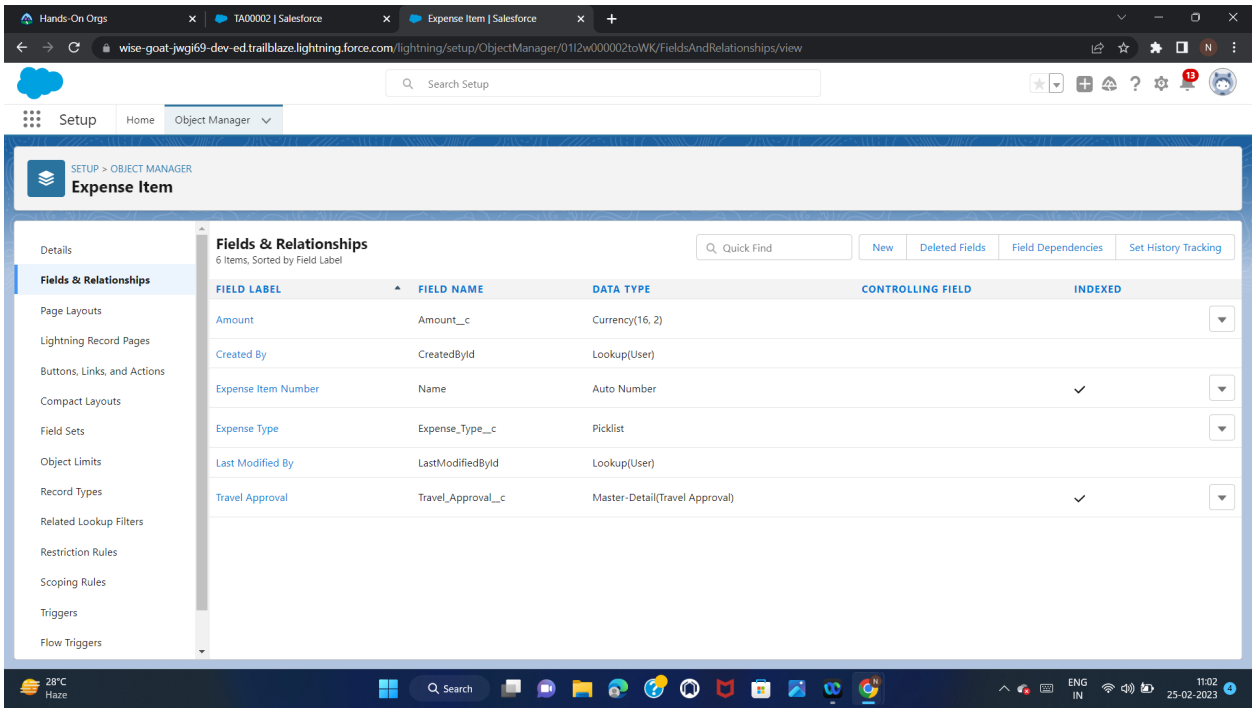
Create Expense Item Object.
Label=”Expense Item”,Plural=”Expense Items”
Starts with vowel sound=Checked
Record Name=Expense Item Number,Data Type=Auto Number,
Display format=E-{00000}
Allow Reports,Do not create Custom Tab,Save



Step 3:

Create the following custom fields:

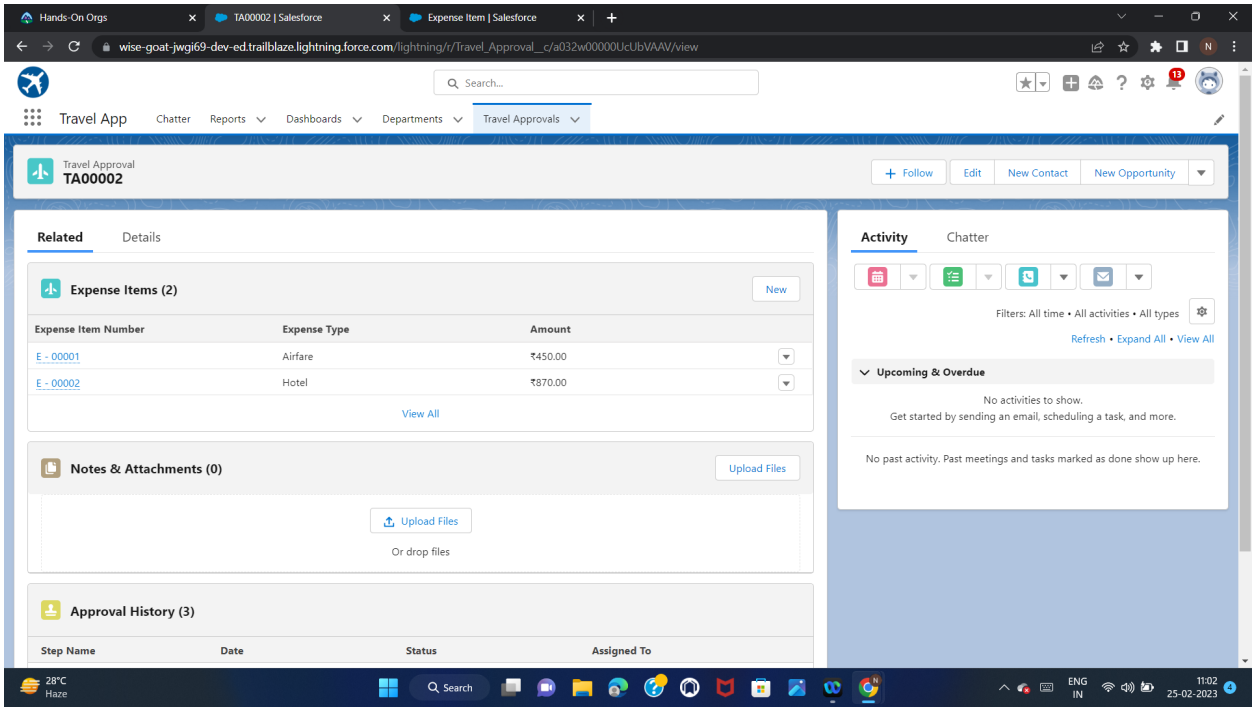
1.Amount,length=16,Type=Currency,Decimal=2,Required=True
2.Expense Type,Type=Picklist,
Values=Airfare,Hotel,Rental Car,Meals,Other,Required=true
3.Travel Approval,Type=Master-Detail,Related To-Travel Approval



Step 4:

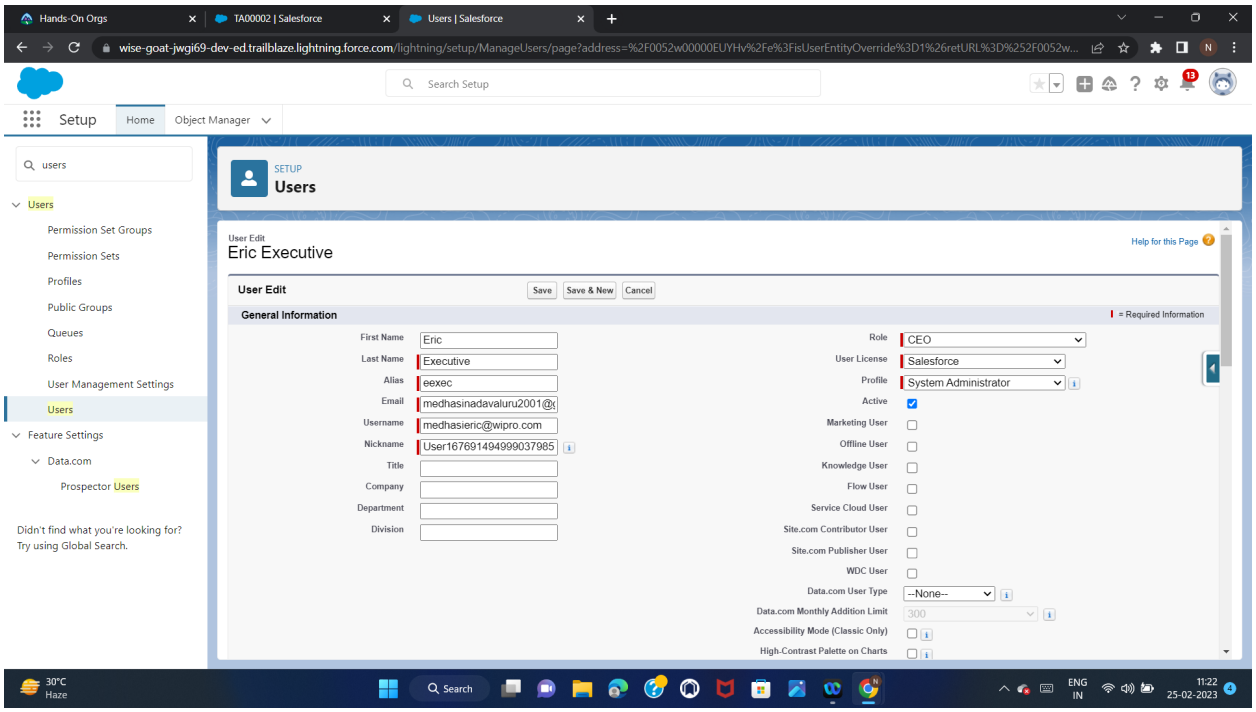
Create Expense Items.

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



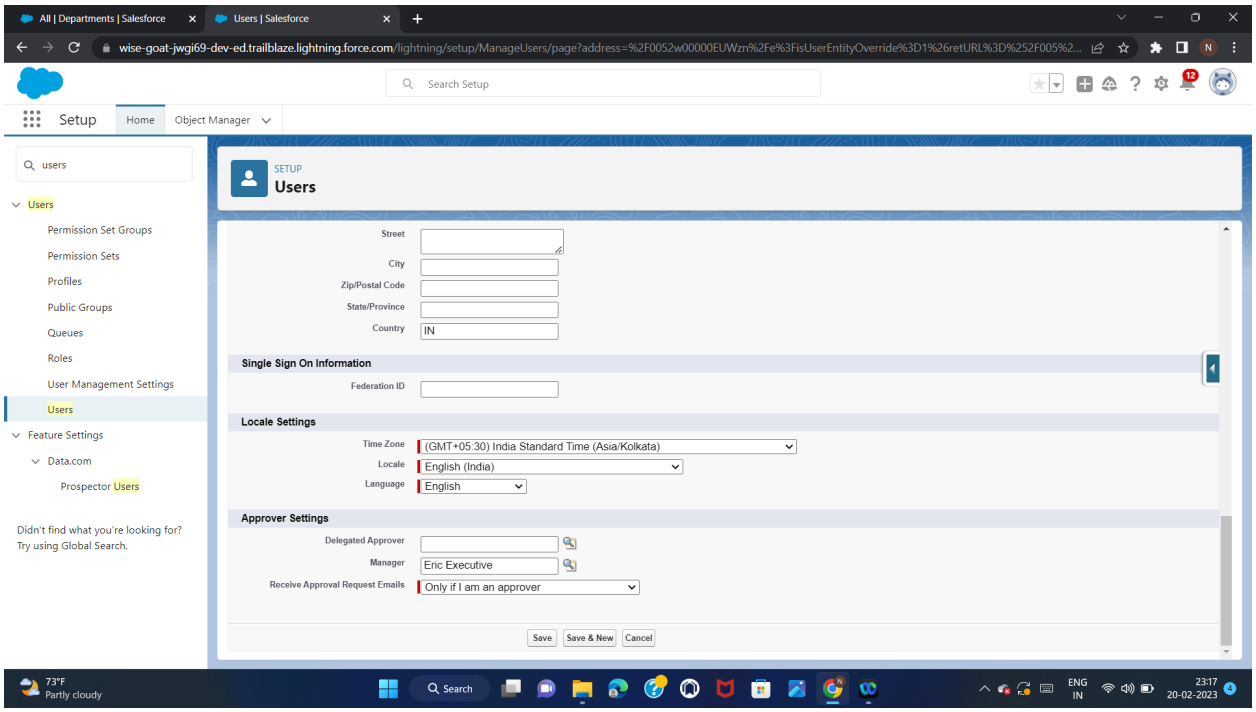
Step 5:

Create a User
First Name=”Eric”,Last Name=”Executive”,Email=”Use your own email”
UserName=”Choose a Unique username”
Role=”CEO”
License=Salesforce
Profile=System Administrator
Save



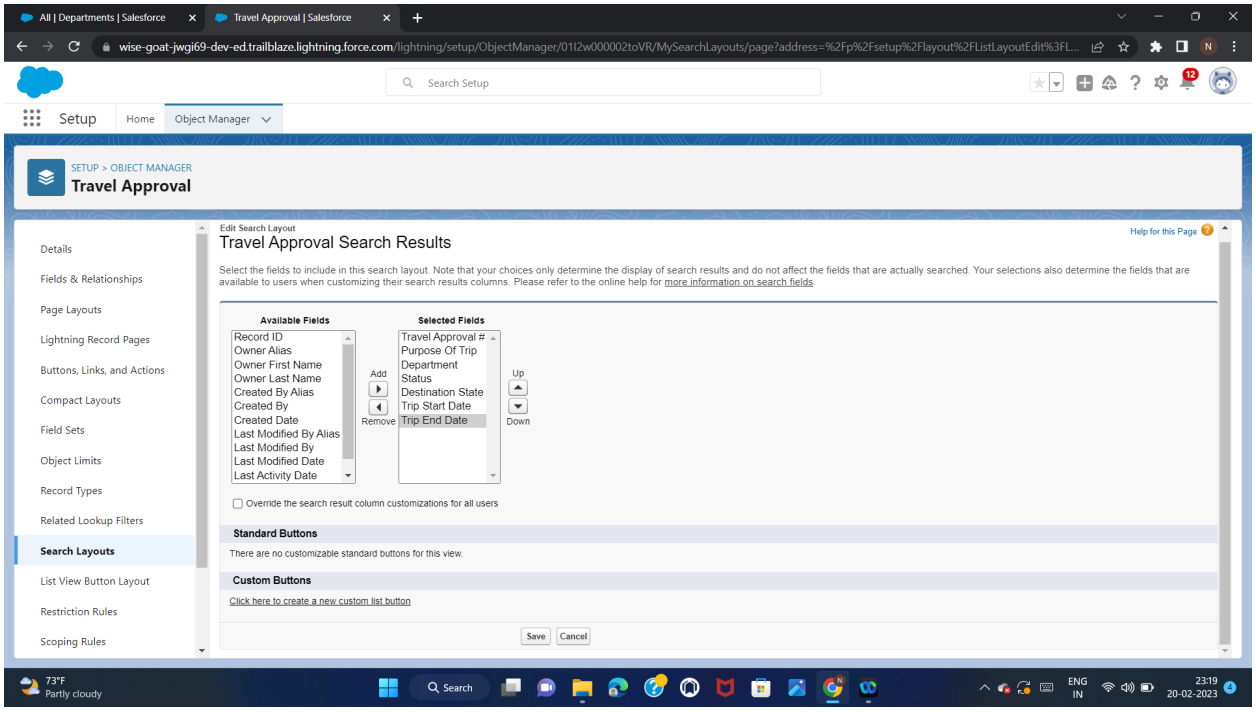
Step 6:

Add user Eric Executive as your manager.



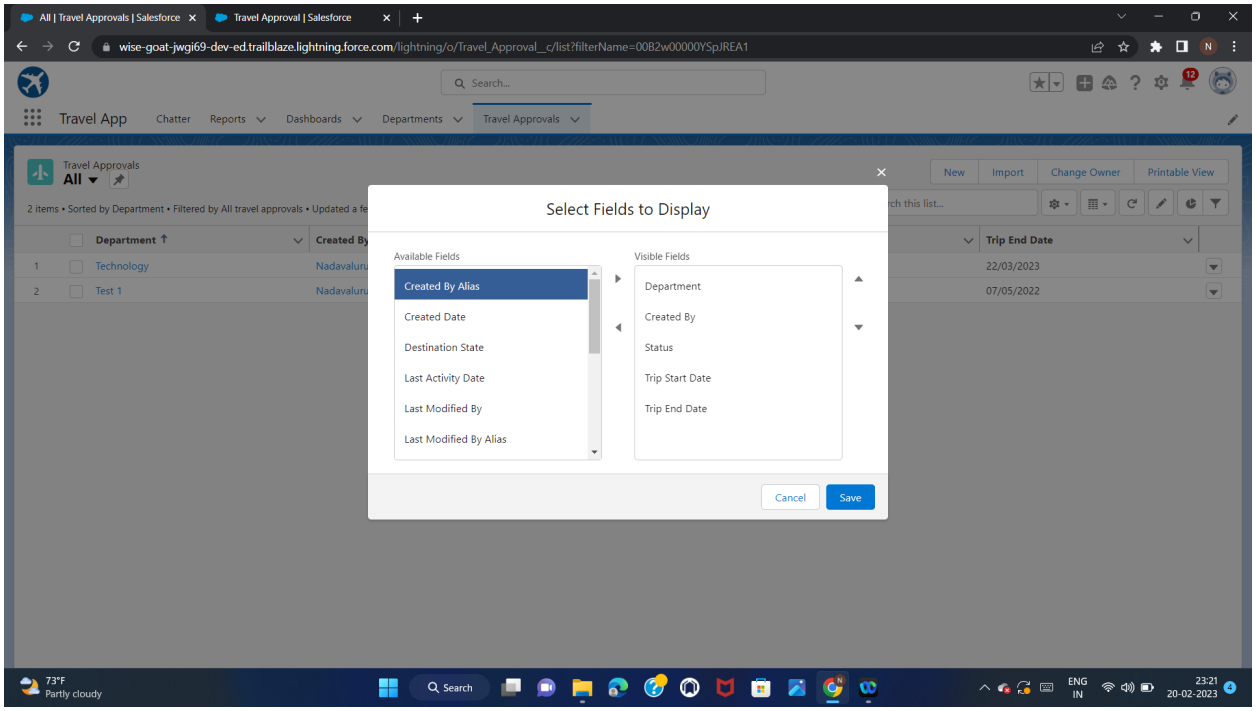
Step 7:

Customize the Travel Approval default search layout as shown in the screenshot.



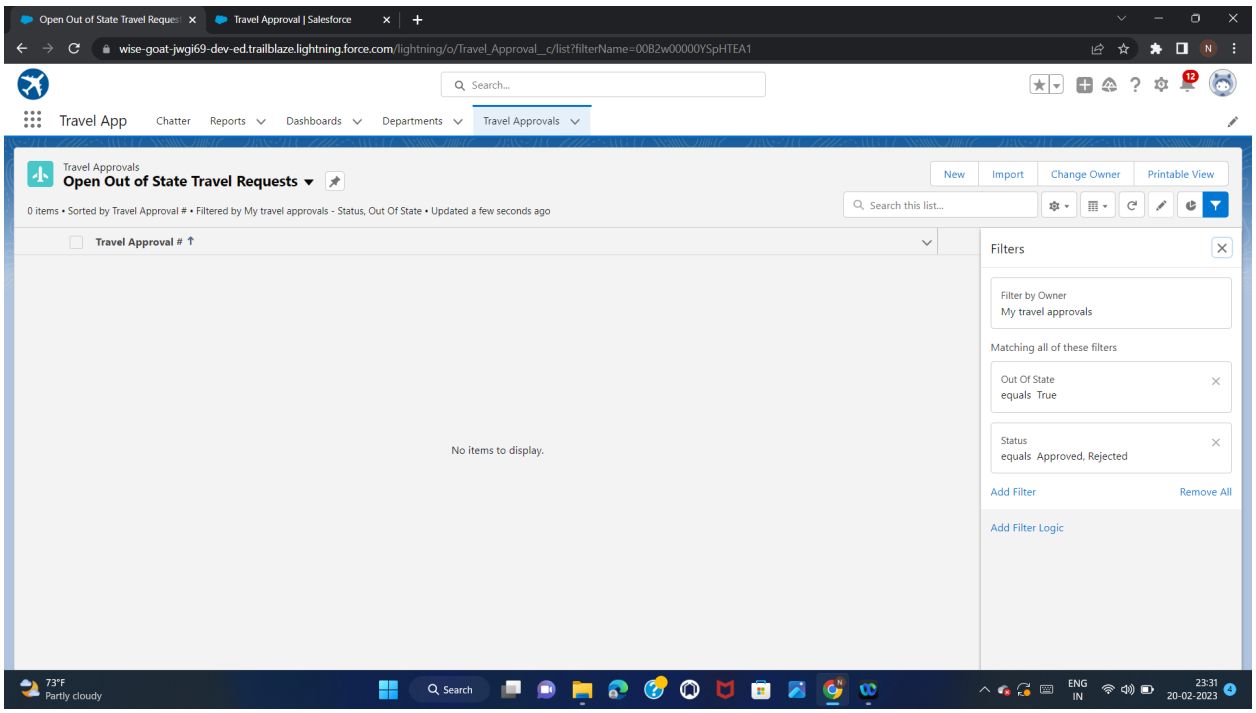
Step 8:

Select fields to display in the Travel Approval "All" List view as shown in the screenshot.



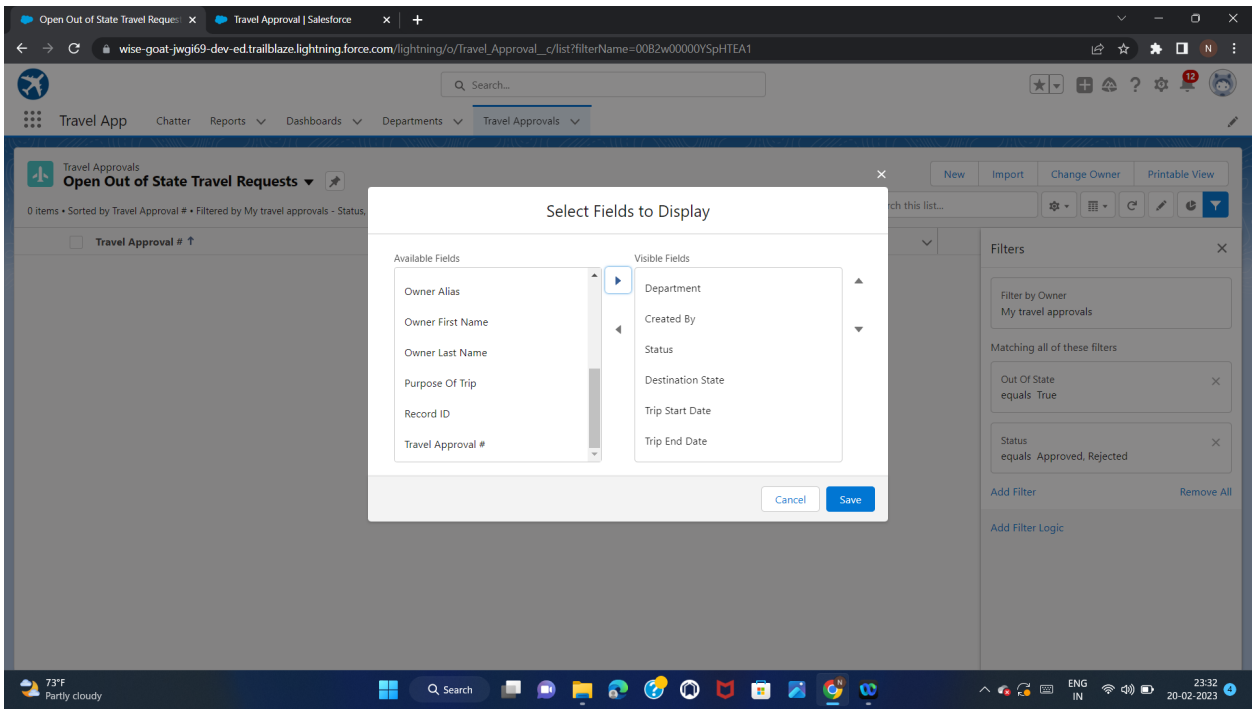
Step 9:

Create Travel Approval custom List View “Open Out of State Travel Requests”
All users should be able to see this list view



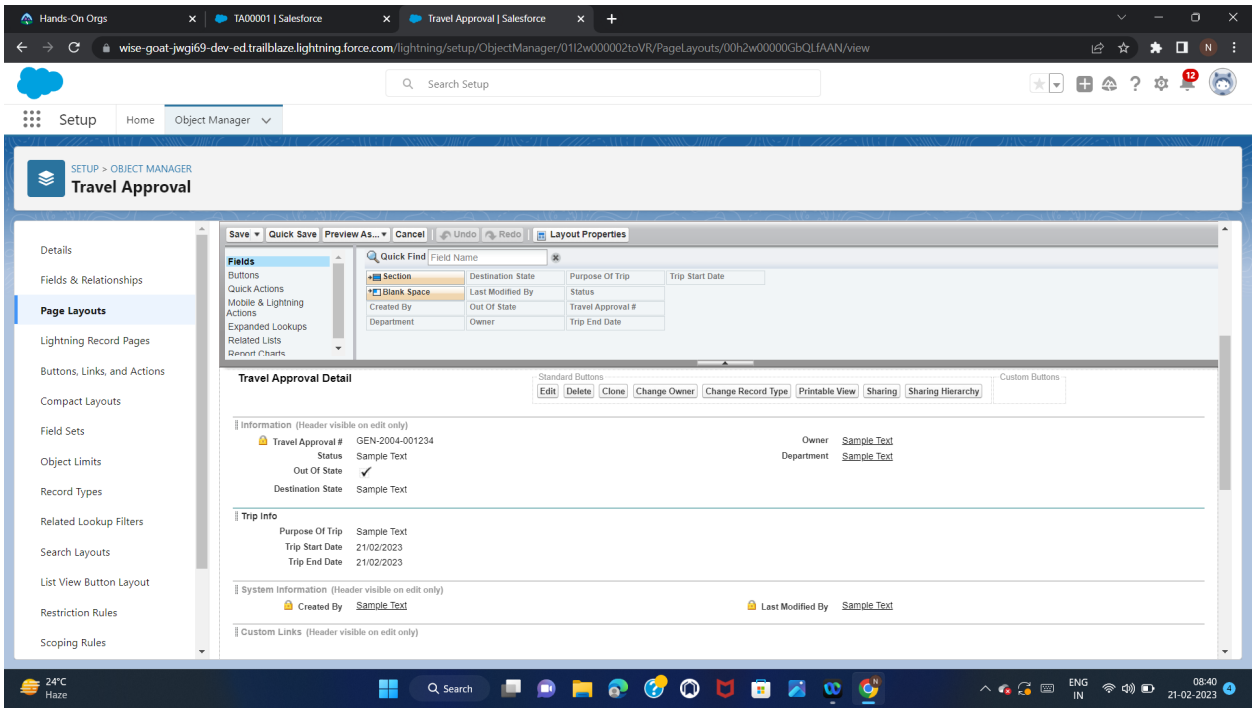
Step 10:

Select fields to display in the Travel Approval “Open Out of State Travel Requests” List view as shown in screenshot and save



Step 11:

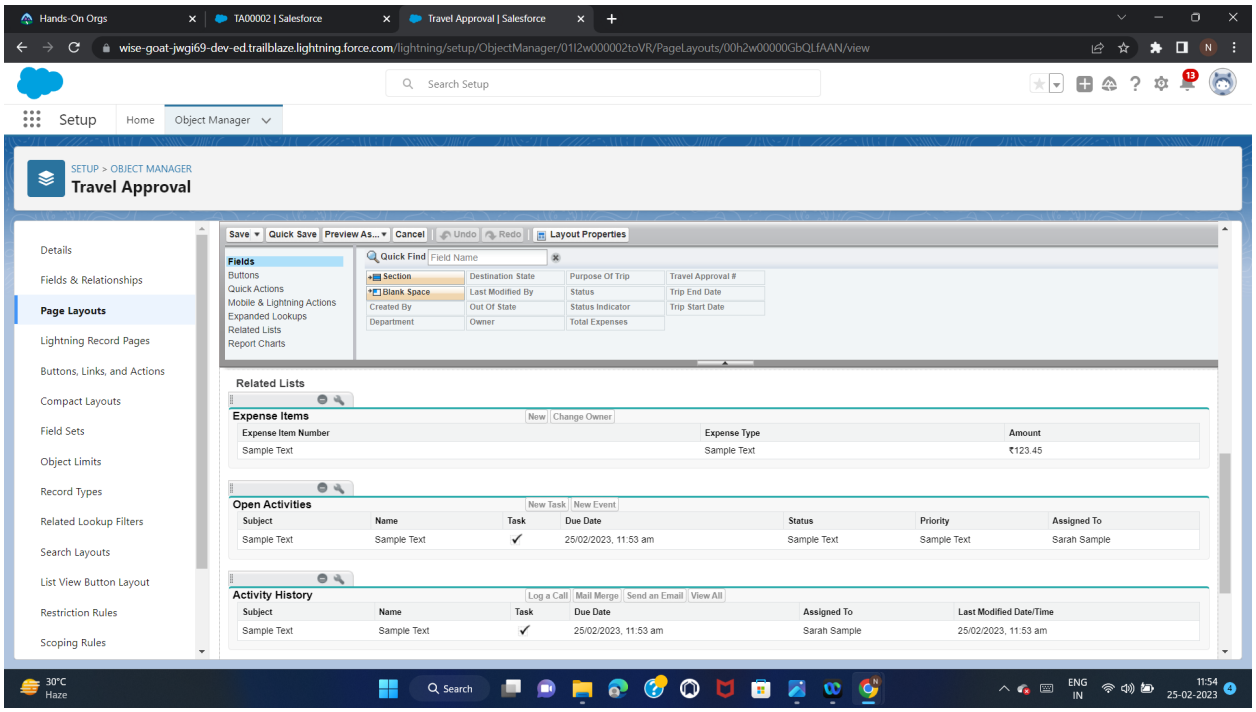
Customize the Travel Approval Page layout.
1.In the Travel App app,click the Travel Approvals tab and open TA-00001
2.Click Setup and then select Edit Page
3.Left click once in the middle of the form to select it.you should see a light blue border around the form.
4.On the right-hand side,Click Travel Approval Layout(previewed)
5.Scroll down to the Travel Approval Detail section.Here's where you can change the order and position of the fields.
6.Add a section to the page layout called "Trip Info" and add the fields as shown in the screenshot.



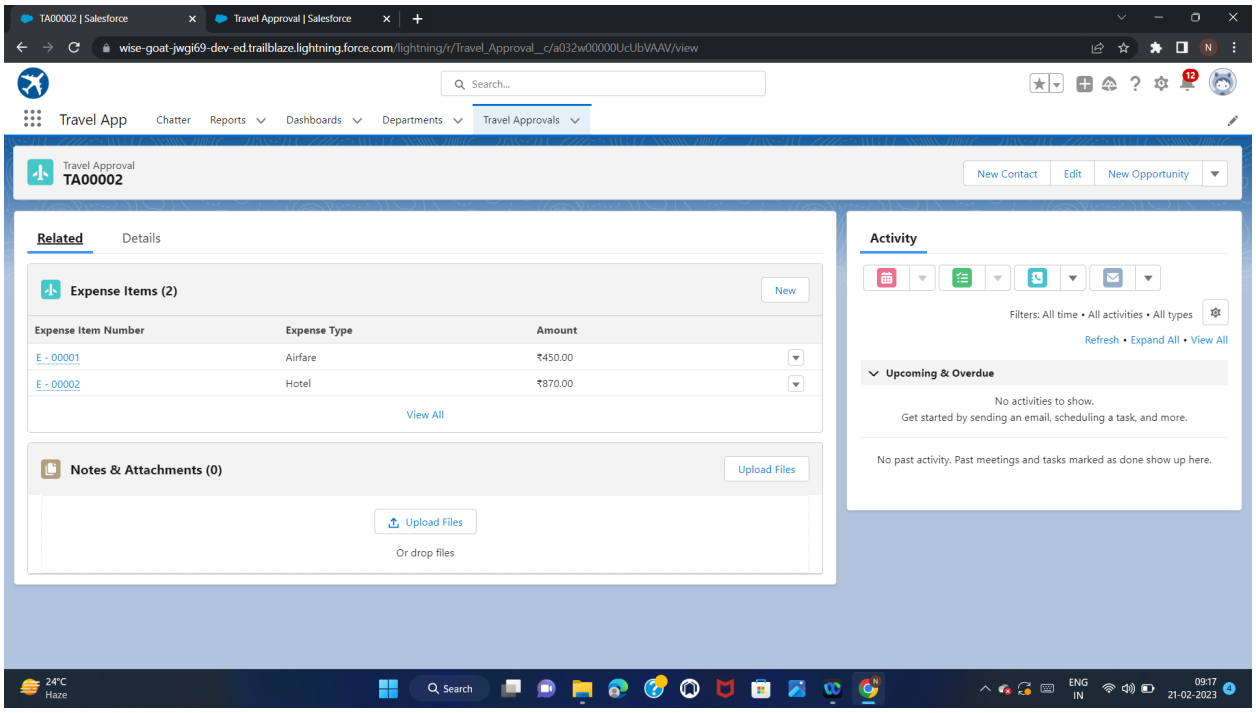
Step 12:

Edit the Expense Item Related List

Related lists can be added to any salesforce page they are related or linked to.
1.From Page Layouts in the Object Manager,Select Travel Approval Layout.Scroll down to the Expense Items related list.click the wrench icon in the header tab.
2.Move the Expense Type and Amount fields from Available fields to Selected fields.
3.Clock OK.Click Yes if you are asked if you want to Overwrite Users Related Lists Customizations.
4.Click Save.
5.Navigate back to a Travel Approval Record and look at the related lists.you should see the extra columns you added to the expense item related lists.



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

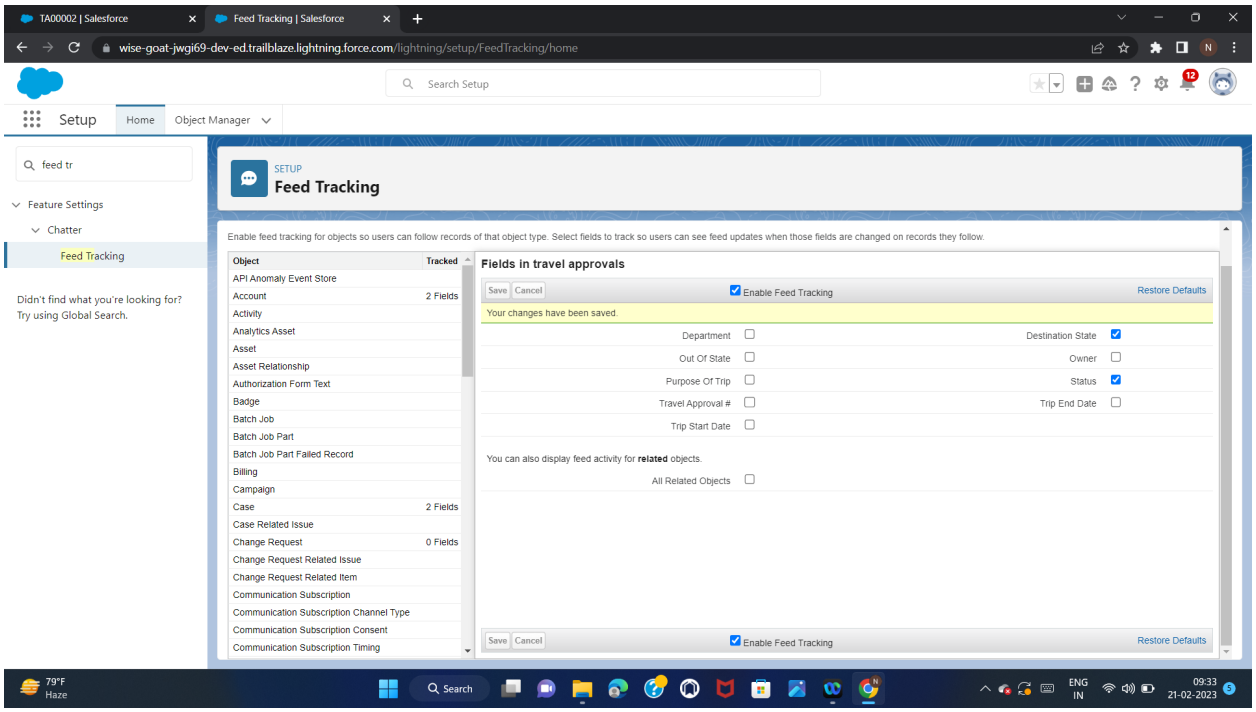


Step 13:

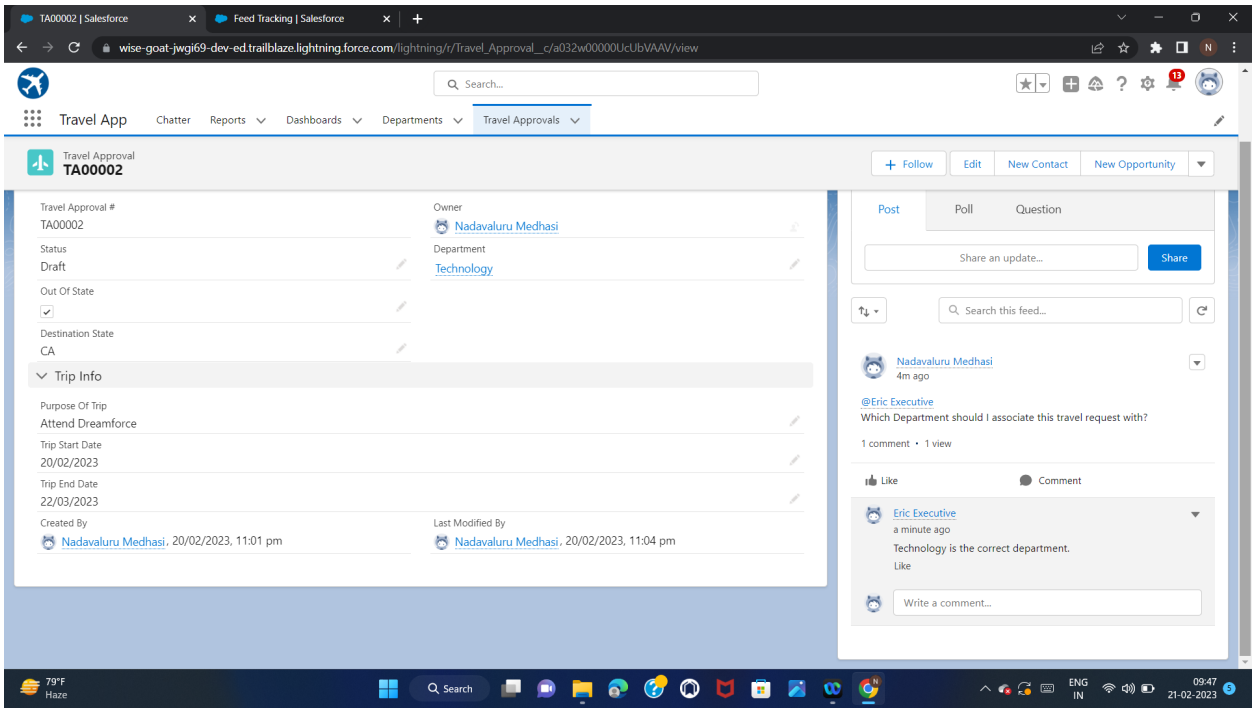
Enable Chatter on the Travel Approval Object.

Enable “Feed Tracking” for Travel Approval Object.

Select these 2 fields:Destination State,Status and then save.



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 @.
Login as Eric and reply to the email,saying:”Technology is the correct department”
Enable Administrator can log in as any user.

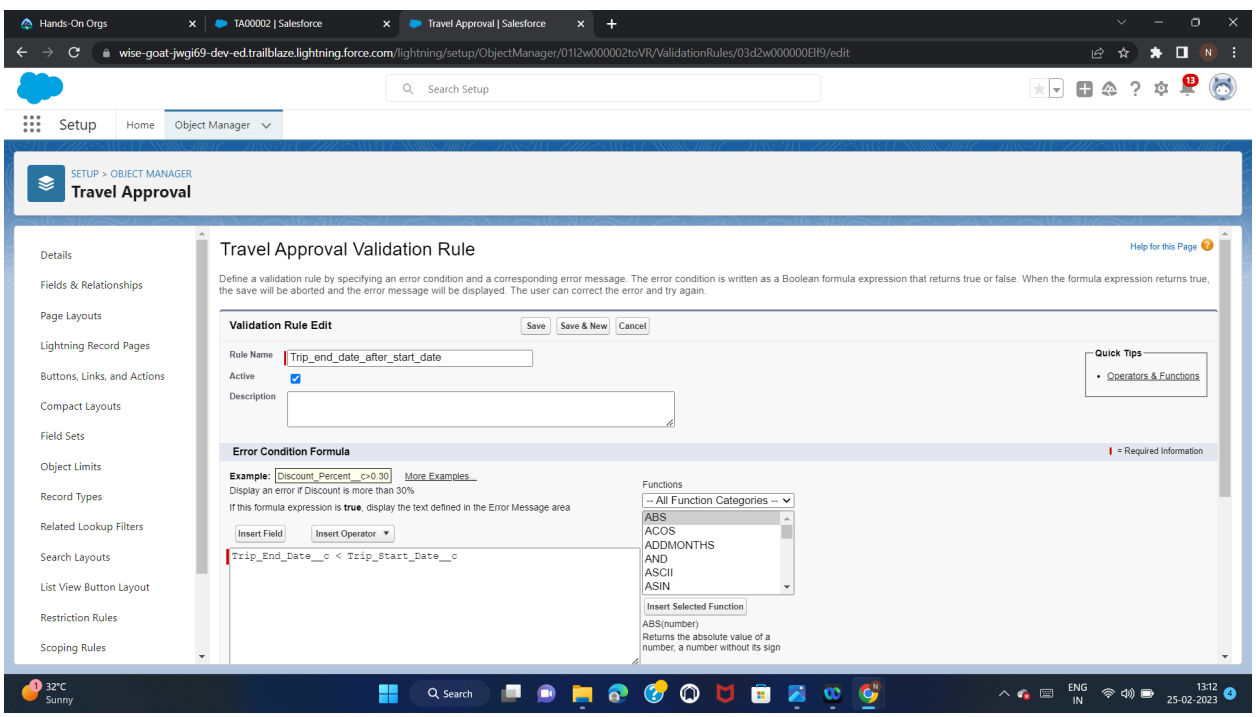


Module 2

Exercise 1:

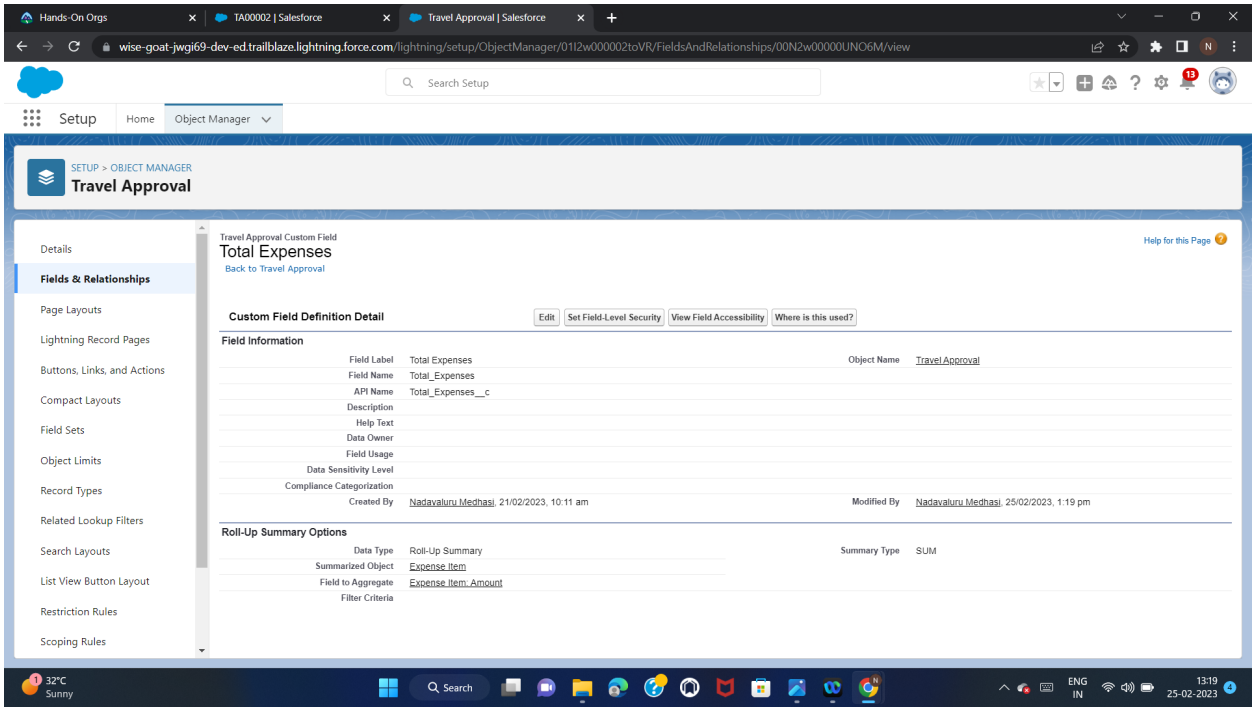
Step 1:

Create a Validation Rule for Travel Approval Object
Business Logic: Trip end date must always be greater than(>=) the trip start date
Name:Trip end date after start date
Make sure to keep “Active” selected/checked
Error Message:Trip end date must be greater than or equal to start date
Error Location:Select field and pick Trip End Date as the location for the error



Step 2:

Create a Roll-Up Summary field on Travel Approval object
Business Logic:Automatically sum of the total amount of expenses from the related Expense Item
Field Label:Total Expenses
Field Name:Total_Expenses
Roll-Up Type:Sum
Field to Aggregate:Amount
Filter Criteria:All records should be included in the calculation
Save



Step 3:

Create Formula Fields

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

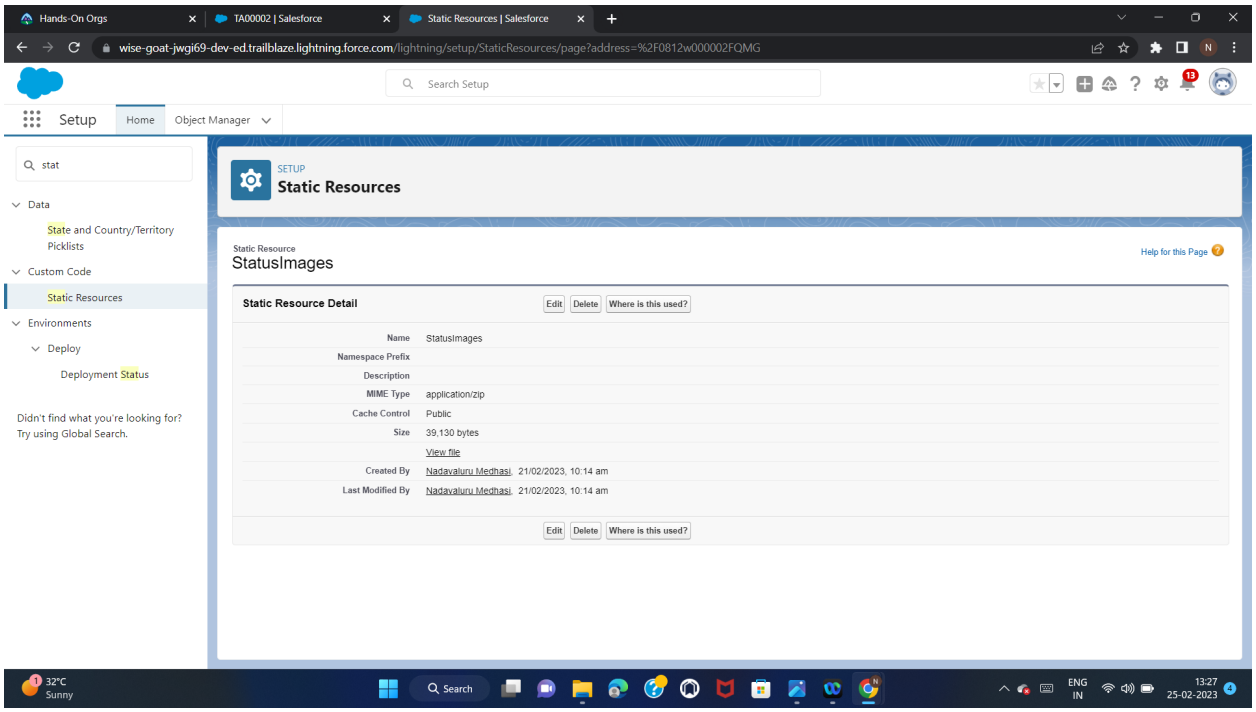
Click Setup select Custom Code | Static Resource | New

Name=StatusImages

File=StatusImages.zip(upload it from the project folder)

Cache-Control=Public

Save



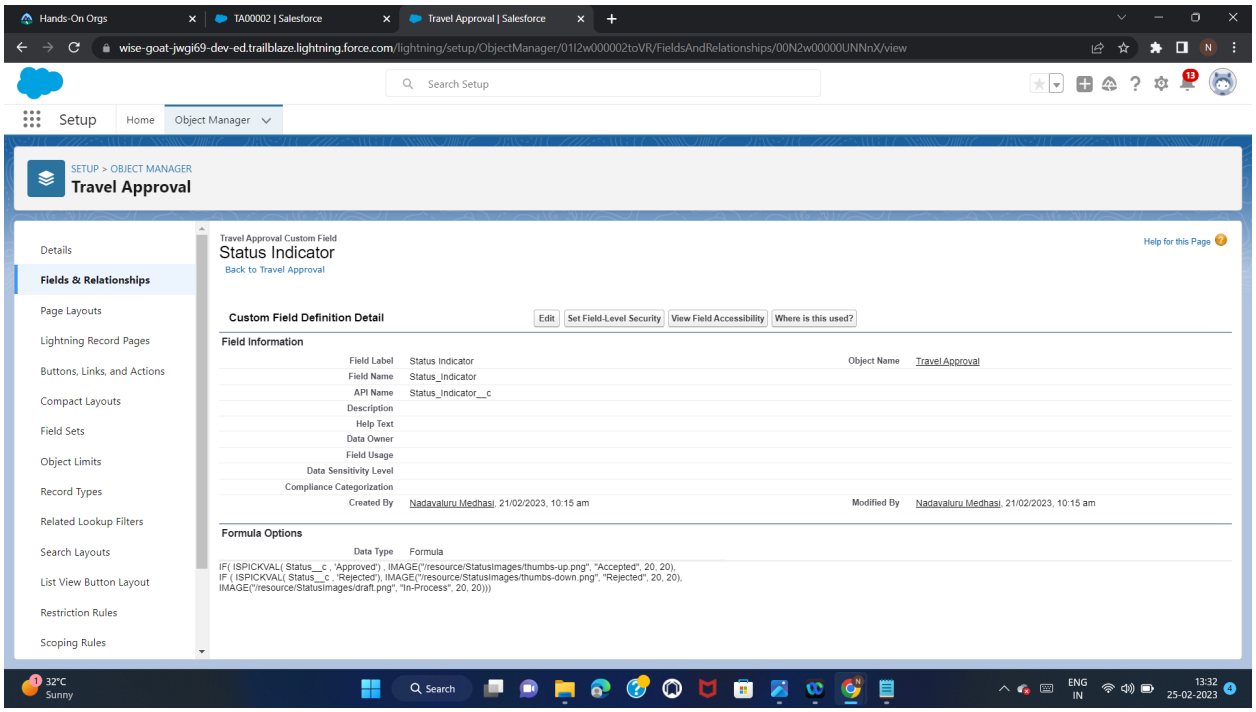
Step 4:

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

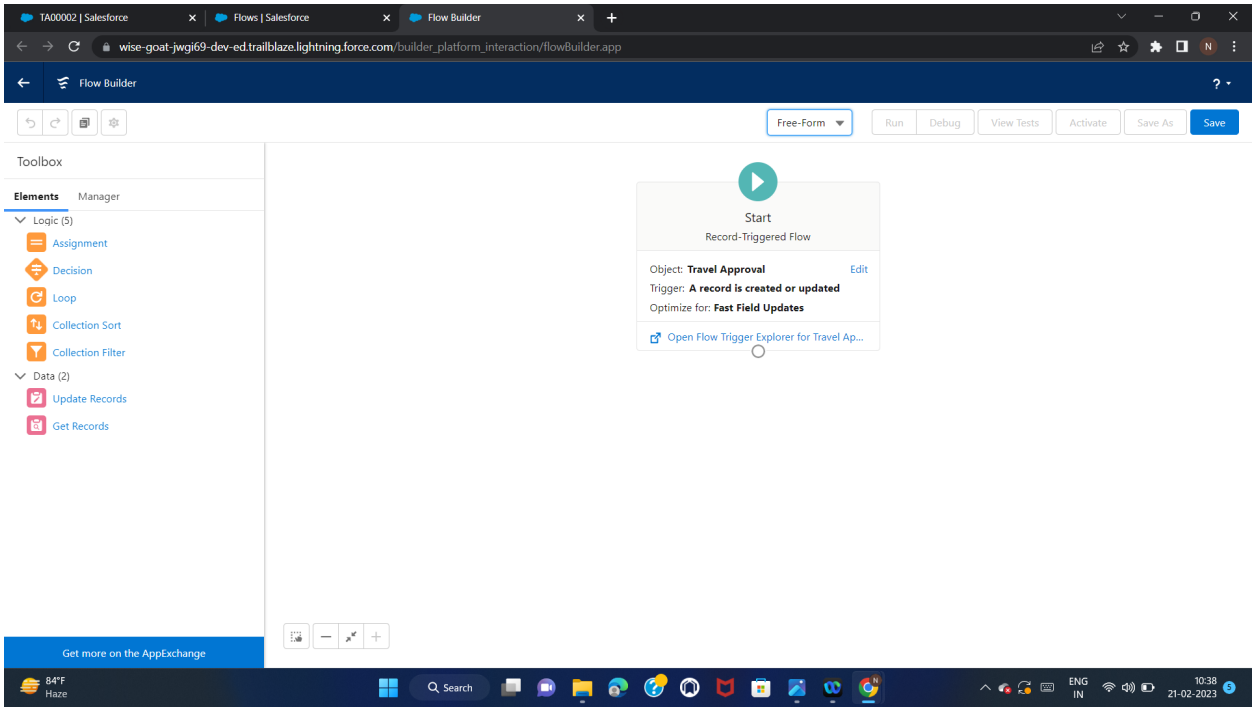
Copy & Paste the formula: IF(ISPICKVAL(Status c , 'Approved') ,
IMAGE("/resource/StatusImages/thumbs-up.png", "Accepted", 20, 20),
IF (ISPICKVAL(Status c , 'Rejected'),
IMAGE("/resource/StatusImages/thumbs-down.png", "Rejected", 20, 20),
IMAGE("/resource/StatusImages/draft.png", "In-Process", 20, 20)))

Save

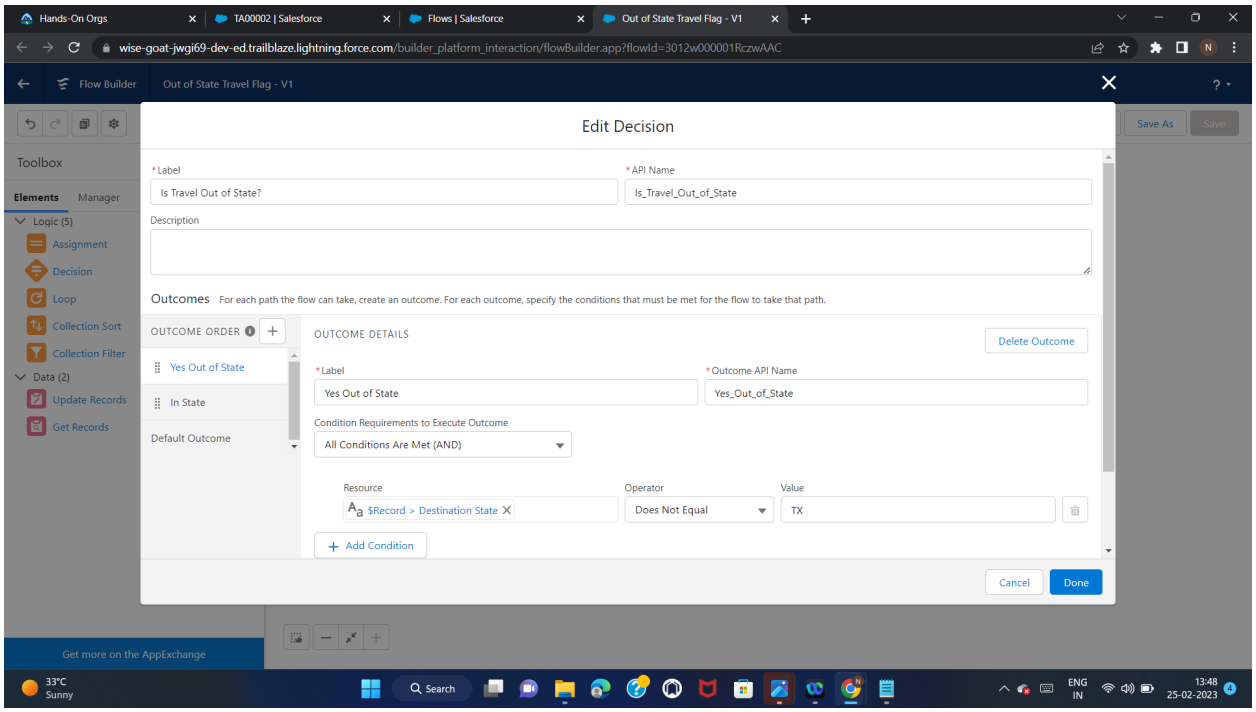


Step 5:

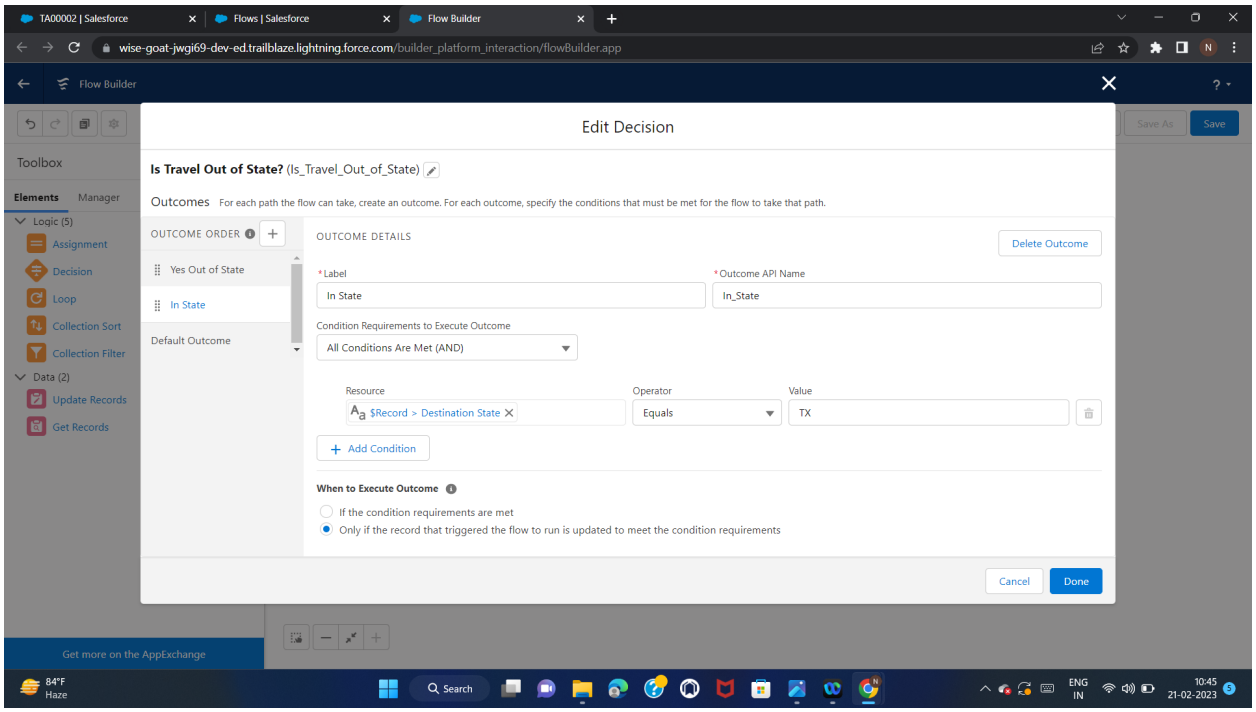
Create a Record – Triggered Flow.
Set the Out-of-State checkbox field on the Travel Approval object if out-of-state travel has been chosen.
Object: Travel Approval.
Configure Trigger: Trigger the flow when: A record is created or updated.
Condition Requirements = None.
Optimize the Flow For: Fast Field Updates.
Done.
Select “Freeform”.



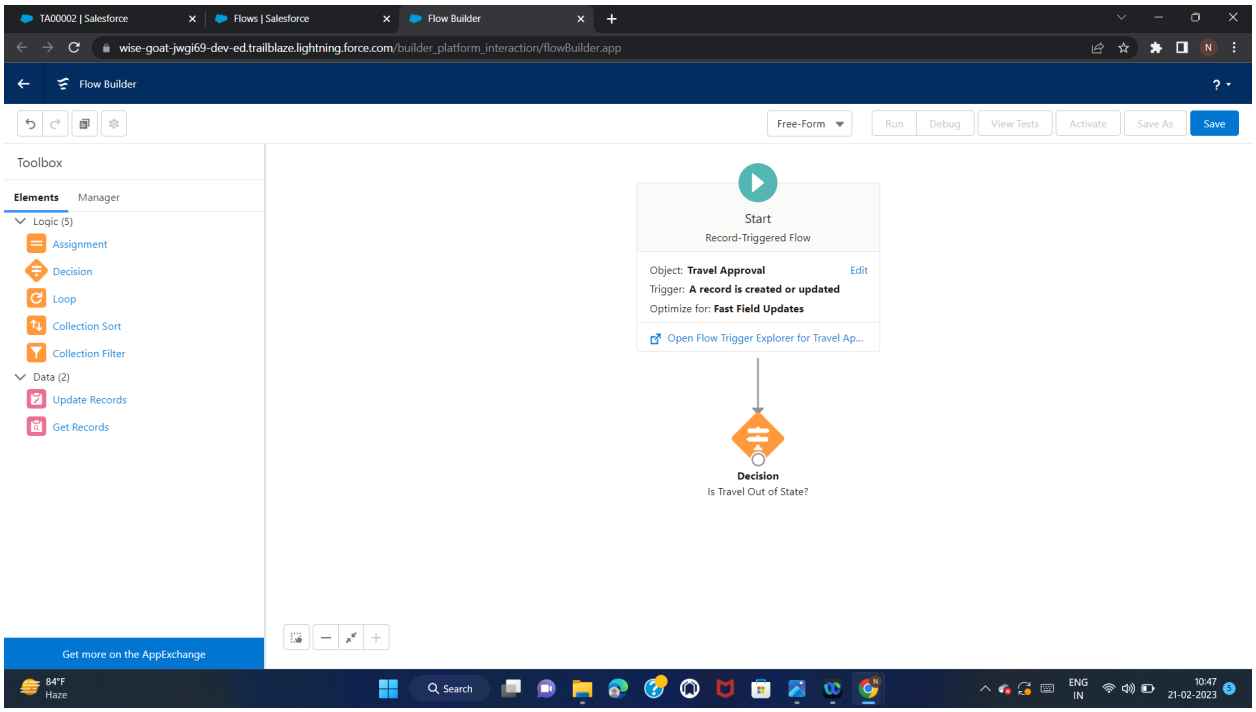
Add a Decision Element to the Flow.
Label = Is Travel Out of State?
Set the Outcomes, For the first outcome, set these parameters:
Label = Yes Out of State, Condition Requirements = All Conditions Are Met (AND).
Resource = $Record > Destination State
Operator: Does Not Equal, Value: Tx
When to Execute the Outcome: Only if the record that triggered the flow to run is updated to meet the condition requirements.



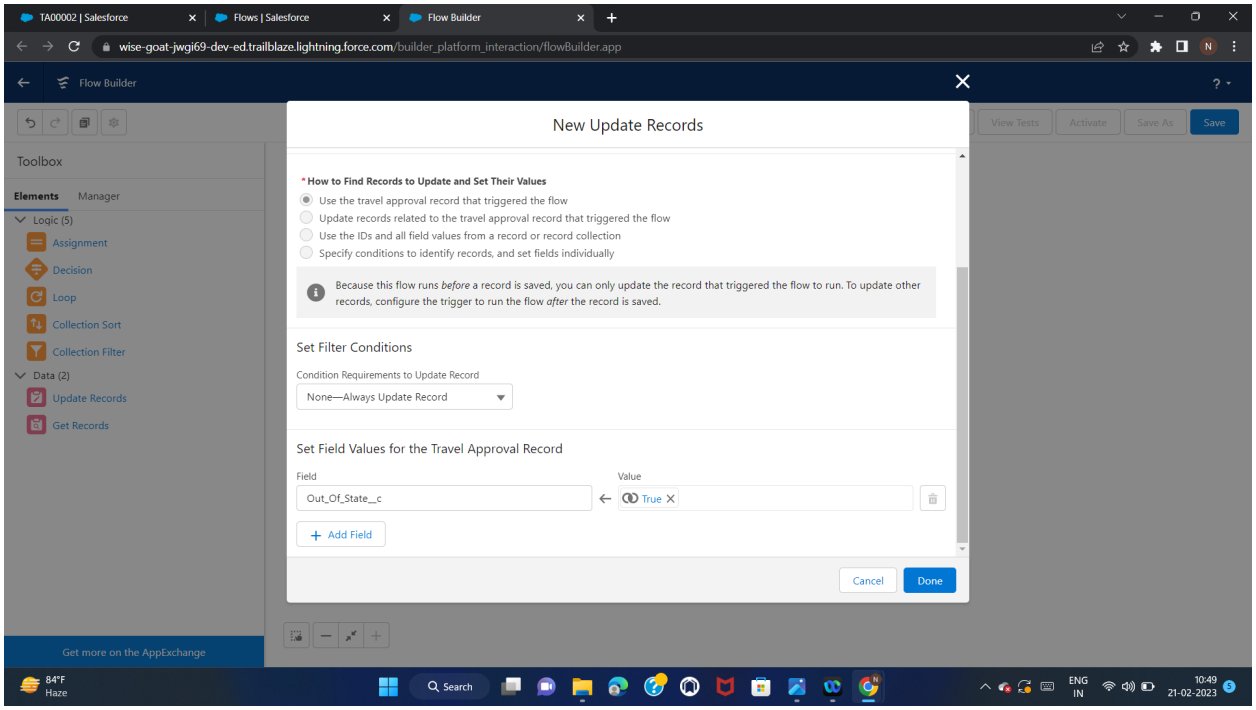
Next to Outcome Order click the + button to add another outcome.
Label = In State
Condition Requirements = All Conditions Are Met (AND).
Resource = $Record > Destination State
Operator: Equal, Value: Tx
When to Execute the Outcome: Only if the record that triggered the flow to run is updated to meet the condition requirement.
Done.



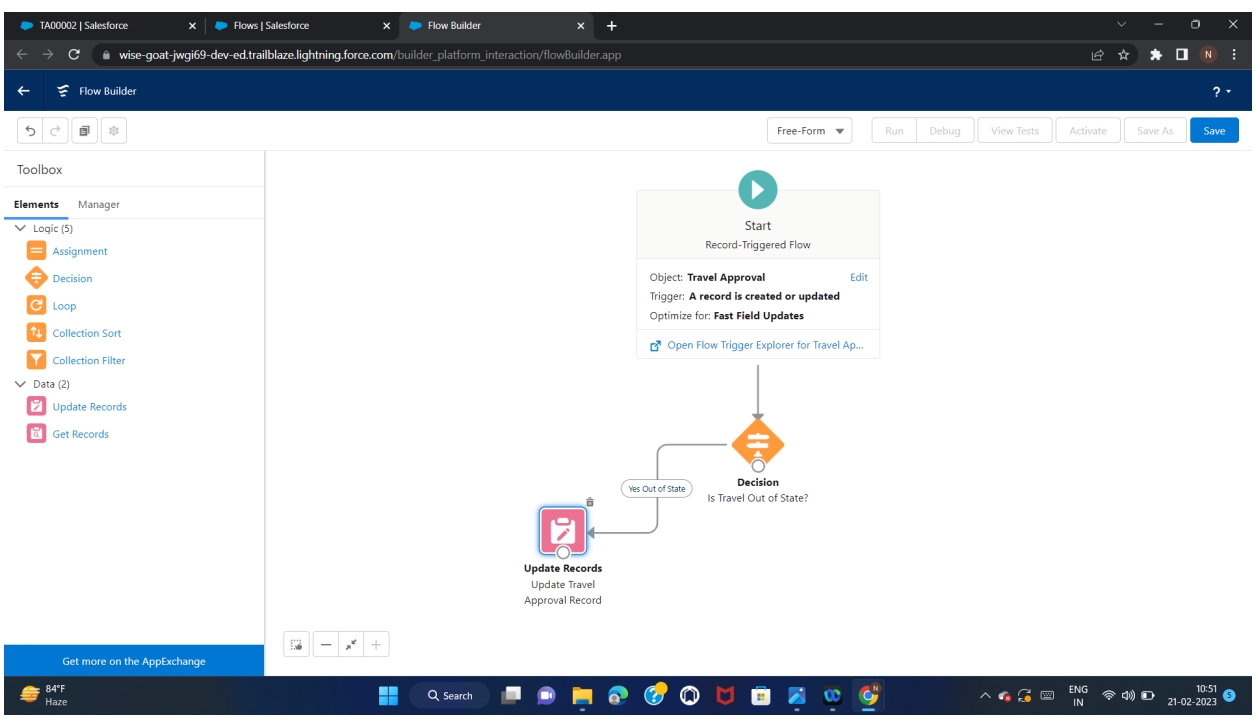
Connect the Start flow element to the decision element



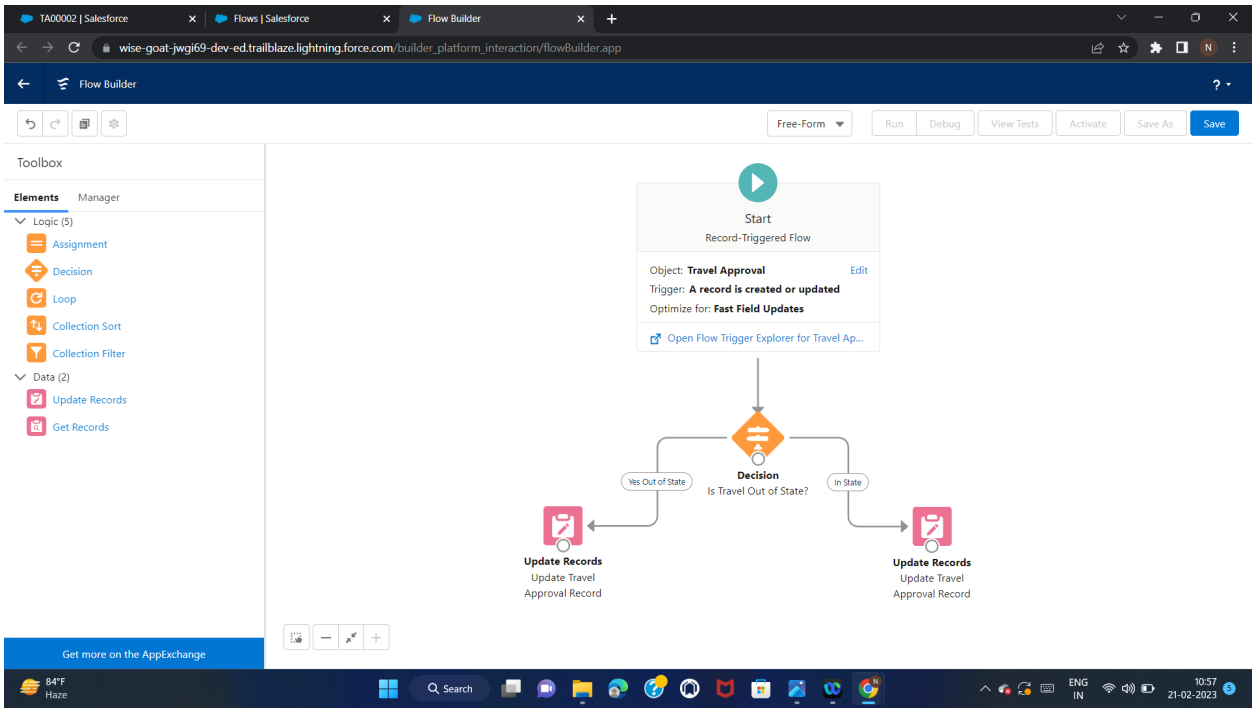
Create an Action for the Flow Using Update Records Elements.
From the flow toolbox, drag an Update Records element onto the flow screen.
Label = Update Travel Approval Record
How to find Records = Use the travel approval record that triggered the flow
Condition Requirements to Update the Record = None—Always Update the Record
Field: Out_of_State c
Value = $GlobalConstant.True (start typing True and this value will come up),
Done.



Drag the white circle from the decision node to the update records you just created,and select the decision Yes Out Of State then Done.
Flow screen should look like this.



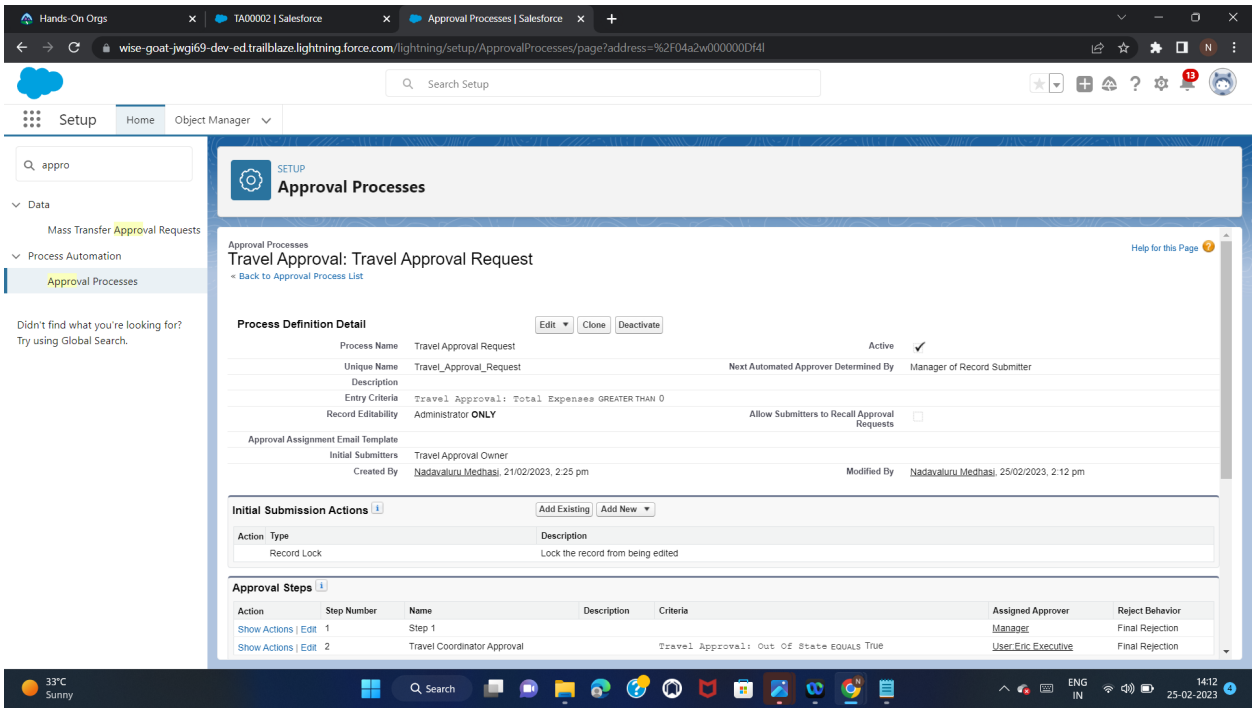
Create an Action for the Flow Using Update Records Elements.
From the flow toolbox, drag an Update Records element onto the flow screen.
Label = Update Travel Approval Record
How to find Records = Use the travel approval record that triggered the flow
Condition Requirements to Update the Record = None—Always Update the Record
Field: Out_of_State c
Value = $GlobalConstant.False (start typing False and this value will come up),
Done.



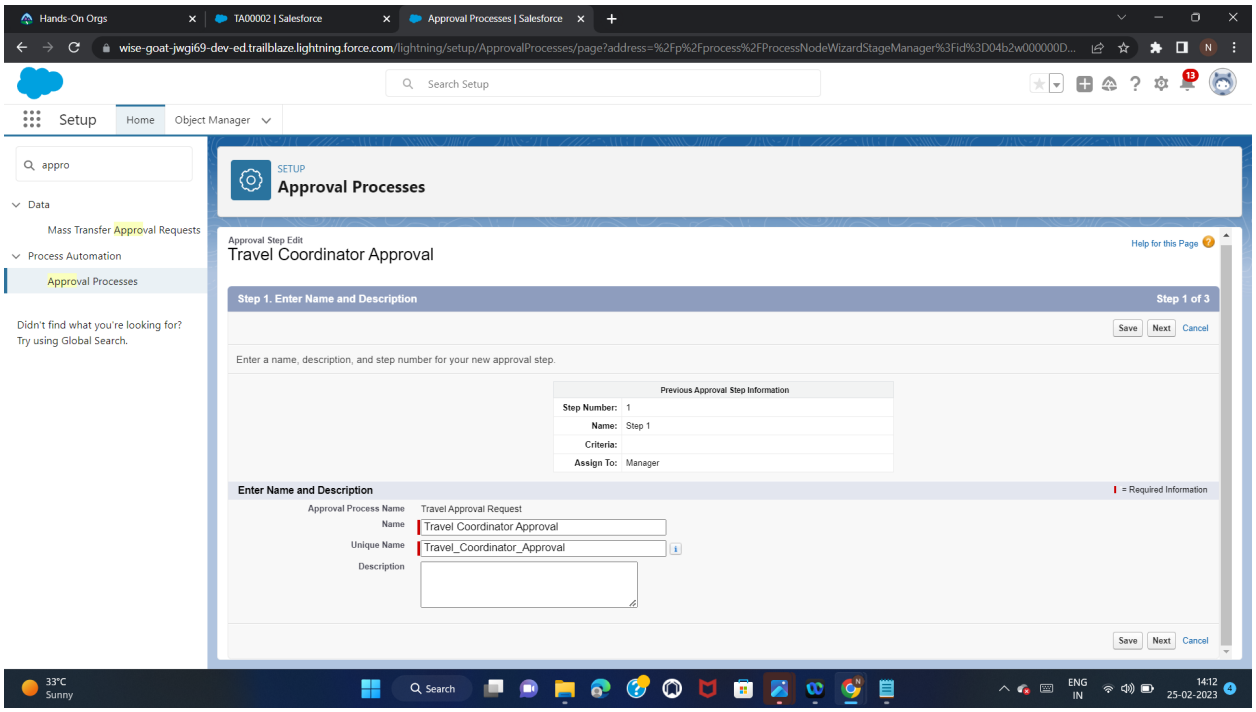
Make sure to save and activate the flow.
Click Save.
Flow Label = Out of State Travel Flag
How to find Records = Use the travel approval record that triggered the flow
Click Save.
Click Activate.

Step 6:

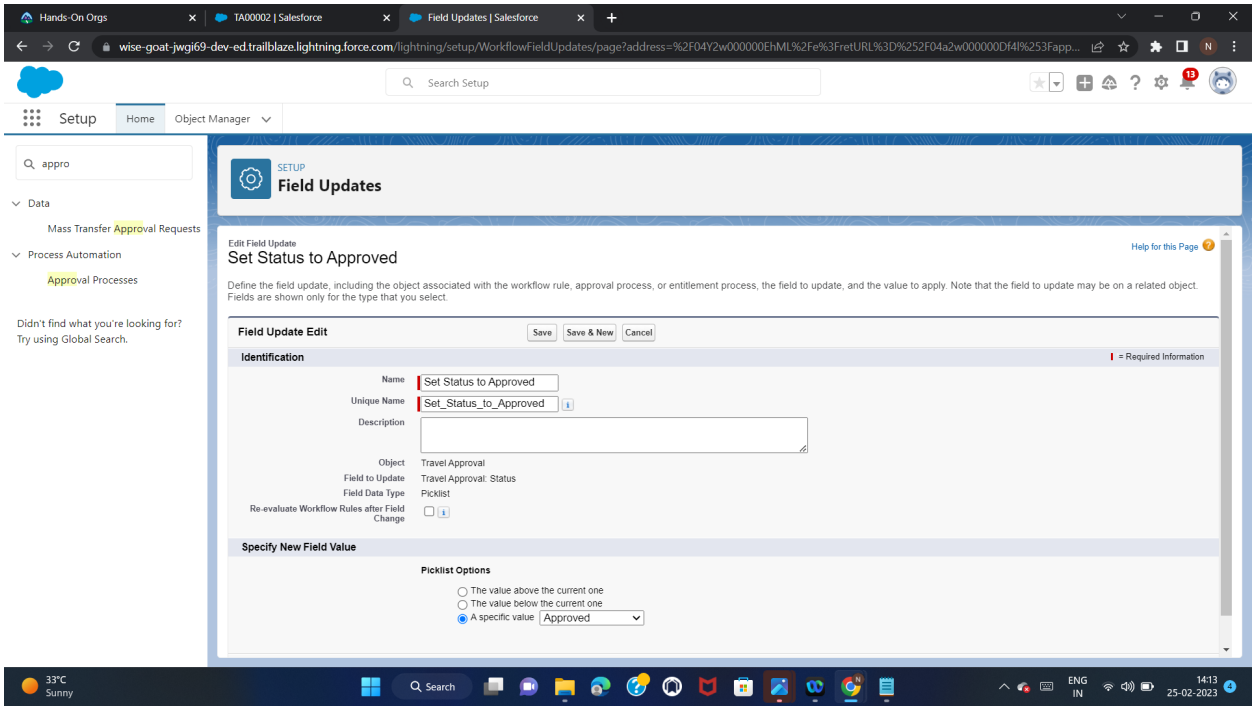
1.Create an Approval Process to send Travel approvals to Manager or Travel coordinator.
2.Approval Process Name: Travel Approval Request (Leave rest of the options as it is)
3.Specify Entry Criteria: Use this approval process if the following: Formula Evaluates to True.
4.In Formula space, Add “True”. There will be no Error.
5.Select Approver: Automatically assign an approver using a standard or custom hierarchy field: Manager
6.Click Save.
7.Click View Approval Process Detail Page.



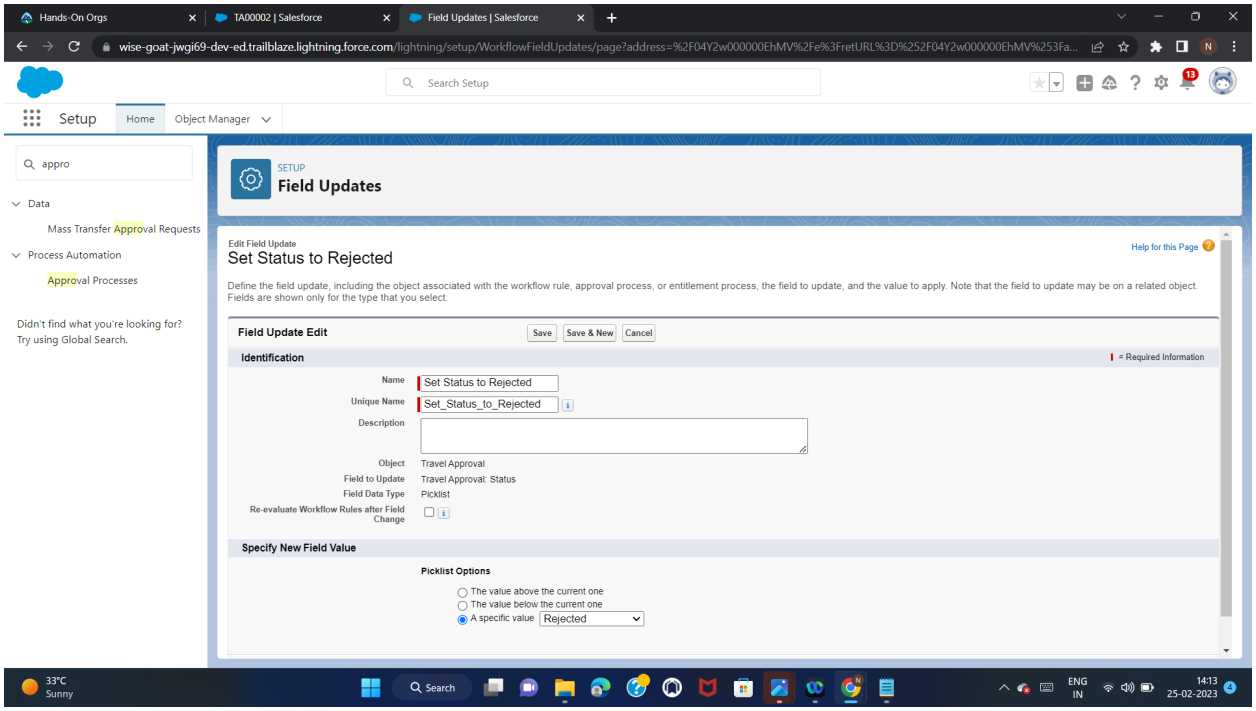
Create an Approval Step for Out-of-State Travel
Name: Travel_Coordinator_Approval
Step Number: 2, Next
Enter this step if the following: criteria are met
formula criteria: Field: Travel Approval: Out-of-State = True.
Automatically assign to the approver(s): User: select Eric Executive as the travel coordinator.
Save. No, I will do this later.



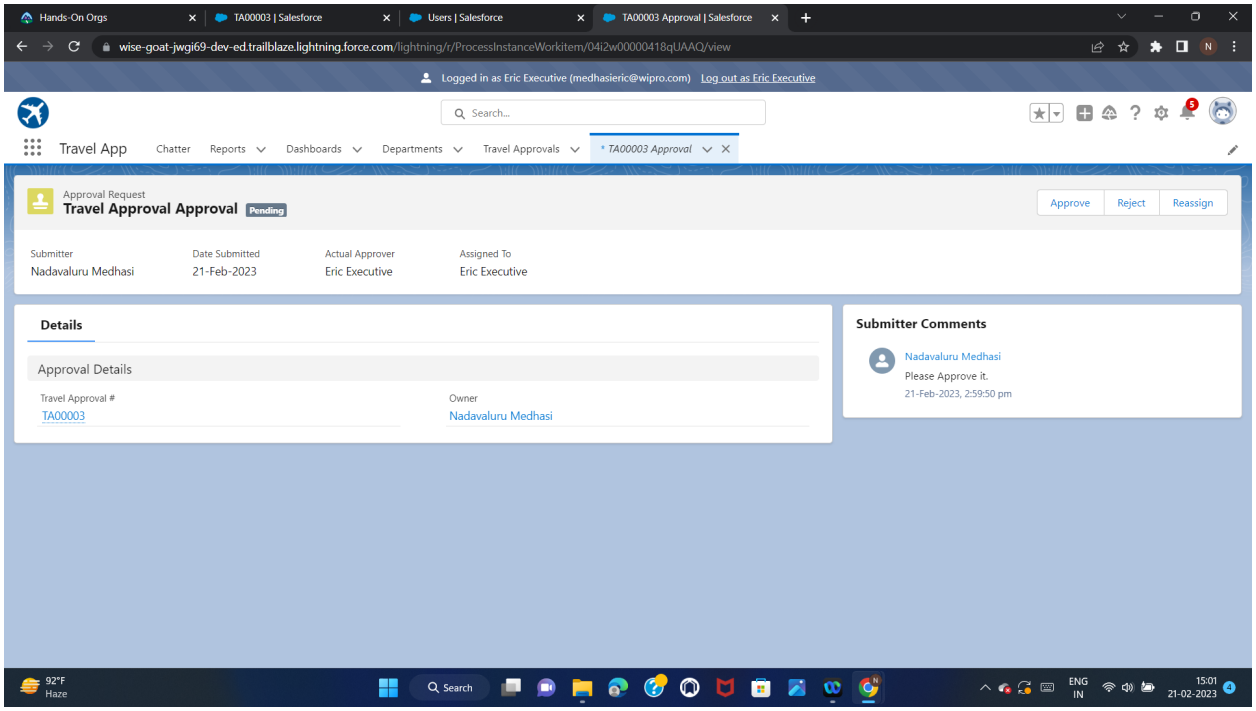
Create Final Approval action.
Click Add New in the Final Approval Actions.
Select Field Update.
Name: Set Status to Approved
Field to Update: Status
Picklist Options: Select A specific value and select Approved from the dropdown list.
Save.



Create Final Rejection action.
Click Add New in the Final Rejection Actions.
Select Field Update.
Name: Set Status to Rejected
Field to Update: Status
Picklist Options: Select A specific value and select Rejected from the dropdown list.
Save, Activate.

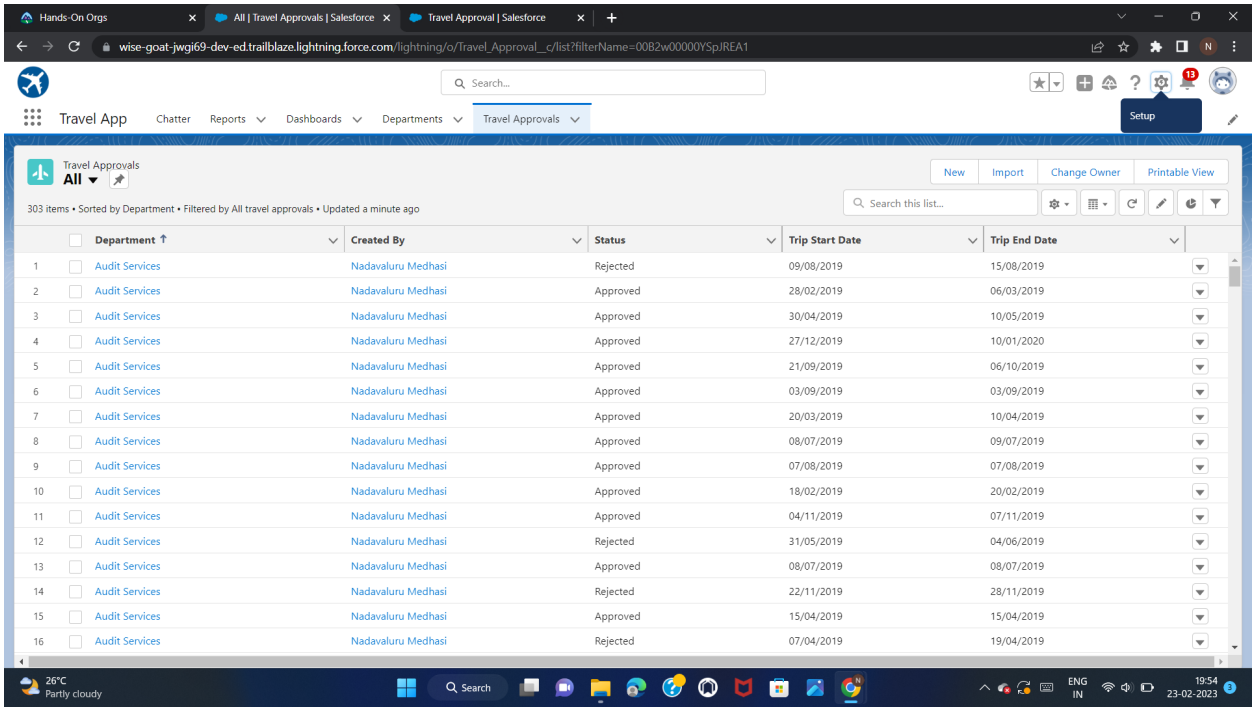


Create few Travel Approval records and Submit for Approval.
Login as Eric,approve and reject the records randomly as shown in the screenshot.



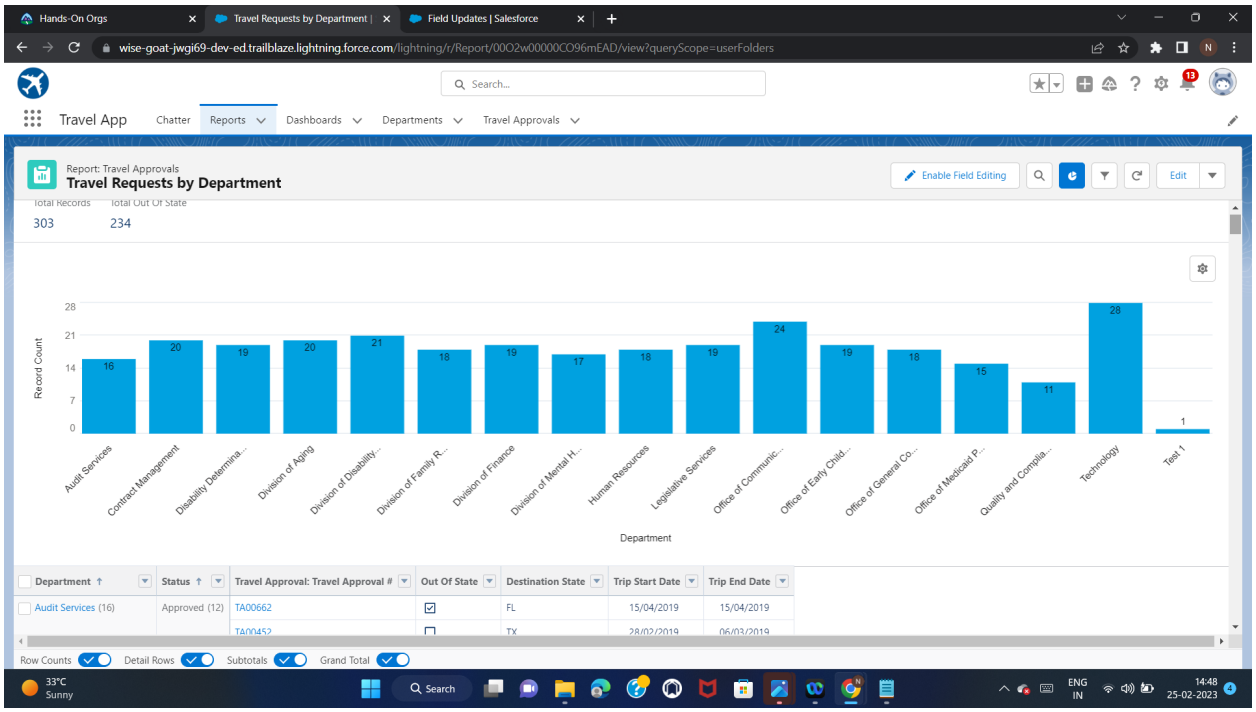
Exercise 2:
Step 1:

Use Data Import Wizard to import Travel Approval records.
Open Import Wizard | Custom Object = Travel Approvals.
Add New Records
Which Department field in your file do you want to match...= Department Name
Trigger workflow rules and processes for new and updated records = Checked.
.CSV File = TravelApprovals
Edit Field Mapping: Fields should be Mapped Automatically, Next, Start Import.



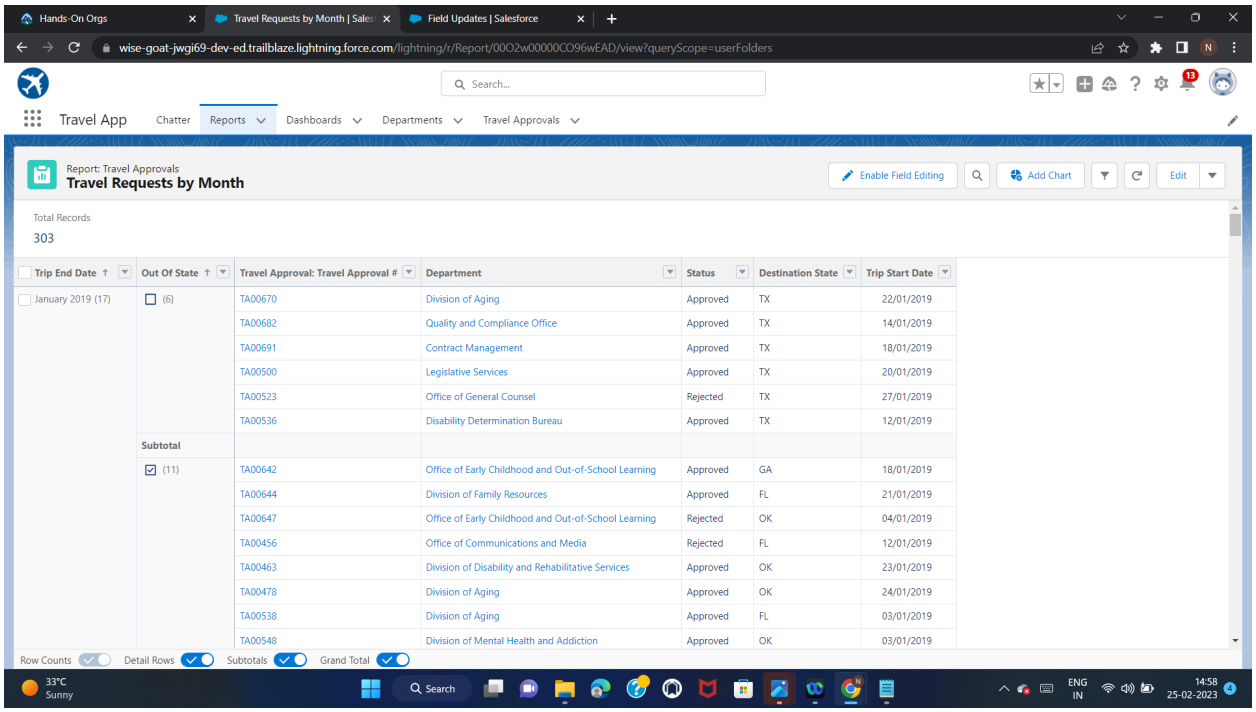
Step 2:

Create a Travel Requests by Department Report
Click on Reports Tab | Click “New Report”
Report types = Travel Approvals.
Add column:Department, Status,Out-of-State,Destination State,Trip Start/End Date
Group Rows by This Field: Department & Status
Add a Chart(Column), X Axis = Department, Y Axis = Record Count,
Show Values = Checked.
Save report as: Travel Requests by Department, Folder: Public Reports.



Step 3:

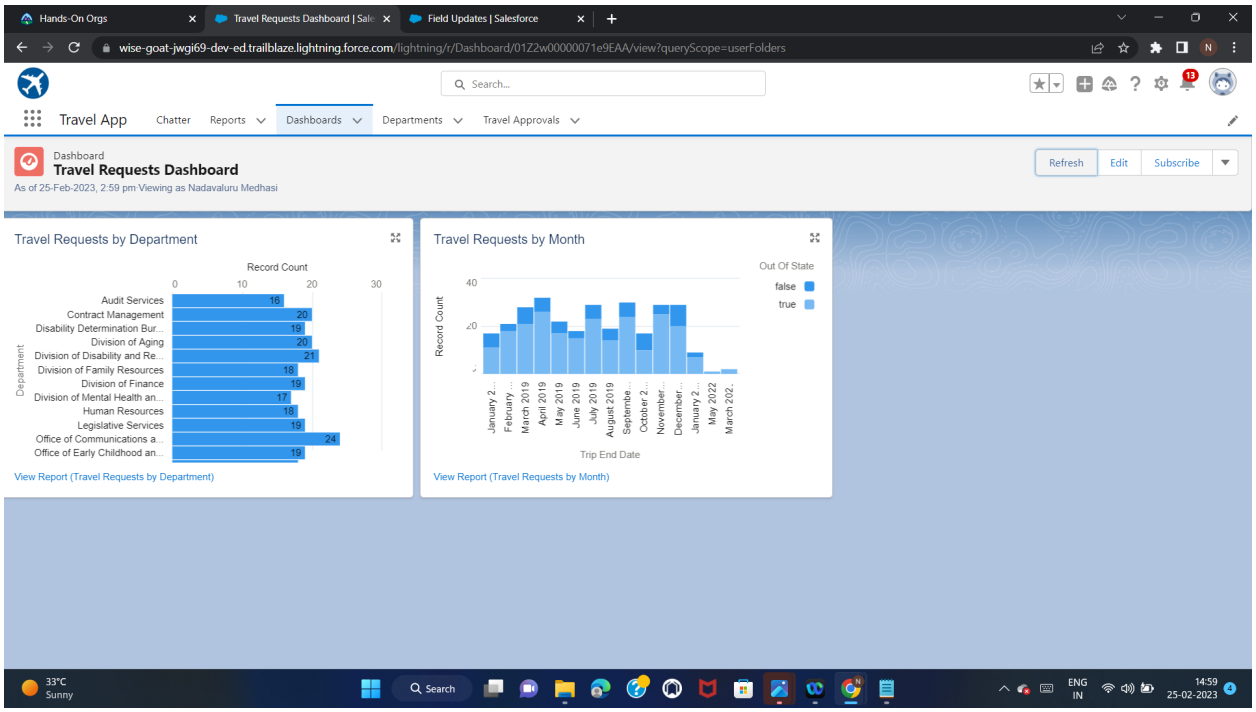
Create a Travel Requests by Month Report
Click on Reports Tab | Click “New Report”
Report types = Travel Approvals.
Add column:Department,Status,Out-of-State,Destination State, Trip Start/End Date
Group Rows by This Field: Trip End Date, select Group Dates By | Calendar Month.
Group Rows by This Field: Out-Of-State.
Save report as: Travel Requests by Month, Folder: Public Reports.



Trip End Date	Out Of State	Travel Approval: Travel Approval #	Department	Status	Destination State	Trip Start Date
January 2019 (17)	(6)	TA00670	Division of Aging	Approved	TX	22/01/2019
		TA00682	Quality and Compliance Office	Approved	TX	14/01/2019
		TA00691	Contract Management	Approved	TX	18/01/2019
		TA00500	Legislative Services	Approved	TX	20/01/2019
		TA00523	Office of General Counsel	Rejected	TX	27/01/2019
		TA00536	Disability Determination Bureau	Approved	TX	12/01/2019
	Subtotal					
	(11)	TA00642	Office of Early Childhood and Out-of-School Learning	Approved	GA	18/01/2019
		TA00644	Division of Family Resources	Approved	FL	21/01/2019
		TA00647	Office of Early Childhood and Out-of-School Learning	Rejected	OK	04/01/2019
		TA00456	Office of Communications and Media	Rejected	FL	12/01/2019
		TA00463	Division of Disability and Rehabilitative Services	Approved	OK	23/01/2019
		TA00478	Division of Aging	Approved	OK	24/01/2019
		TA00538	Division of Aging	Approved	FL	03/01/2019
		TA00548	Division of Mental Health and Addiction	Approved	OK	03/01/2019

Step 4:

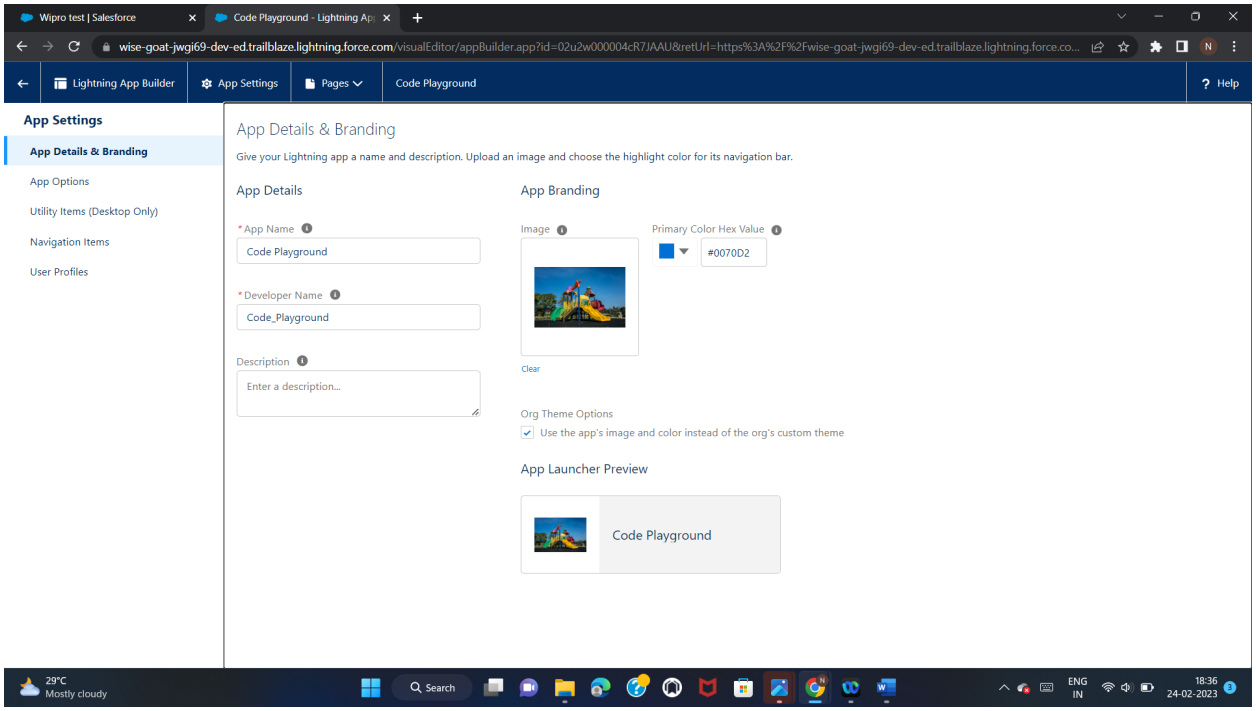
1.Create a Travel Approvals Dashboard
2.Click on Dashboard Tab | Click “New Dashboard”
3.Name: Travel Requests Dashboard, Folder: Private Dashboard. Click Create.
4.Click + Component, Click the Travel Request by Department report and click Select
5.Make sure the Y-Axis is set to Department and the X-Axis is set to Record Count. Click Add
6.Click + Component, Click the Travel Request by Month report and click Select Display As | Stacked Vertical Bar Chart, Sort By | Trip End Date, Click Save, click Done.



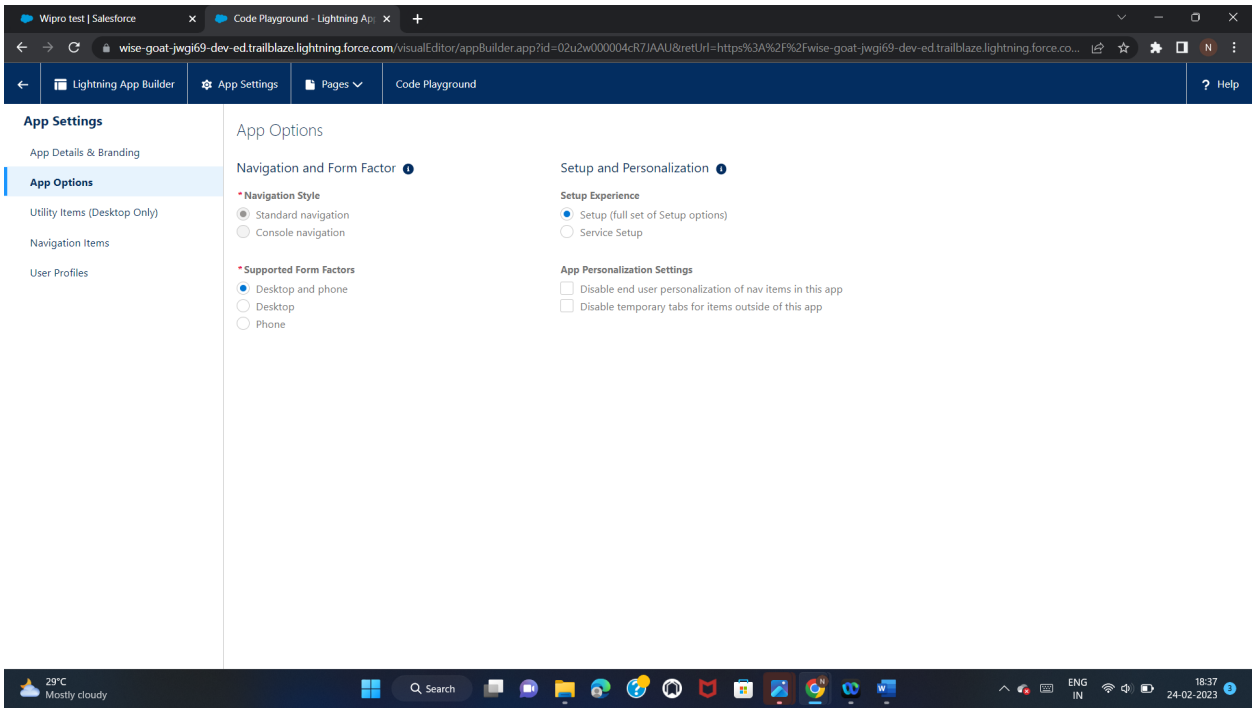
Module 3

Exercise 1:

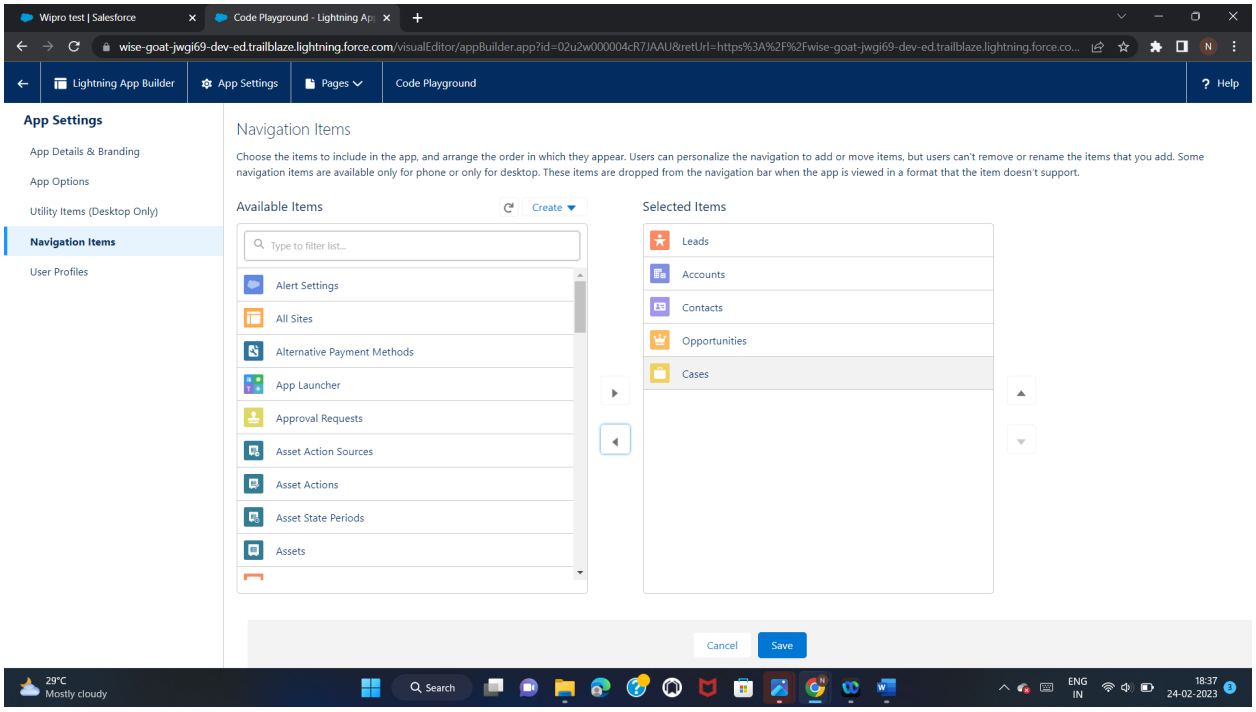
Create a new custom lightning App, name: Code Playground.
Upload an App image, Keep “Org Theme options” checked.



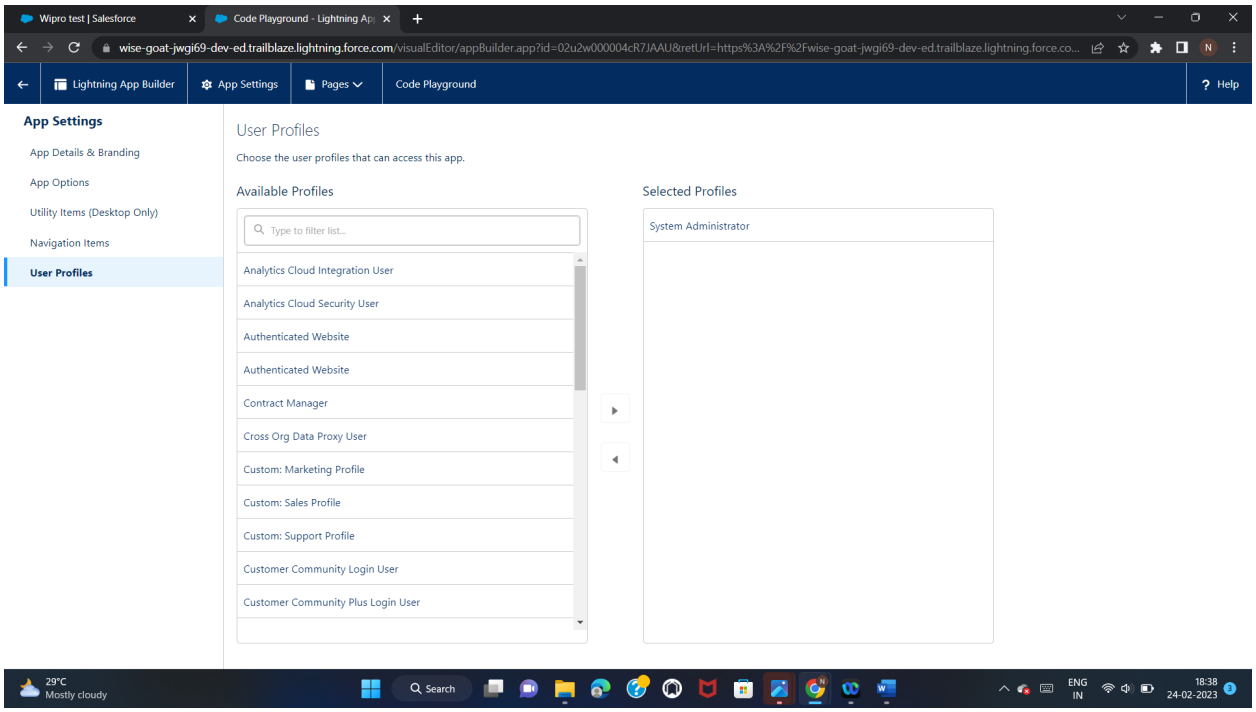
Navigation Style=”Standard Navigation”,leave rest as it is.



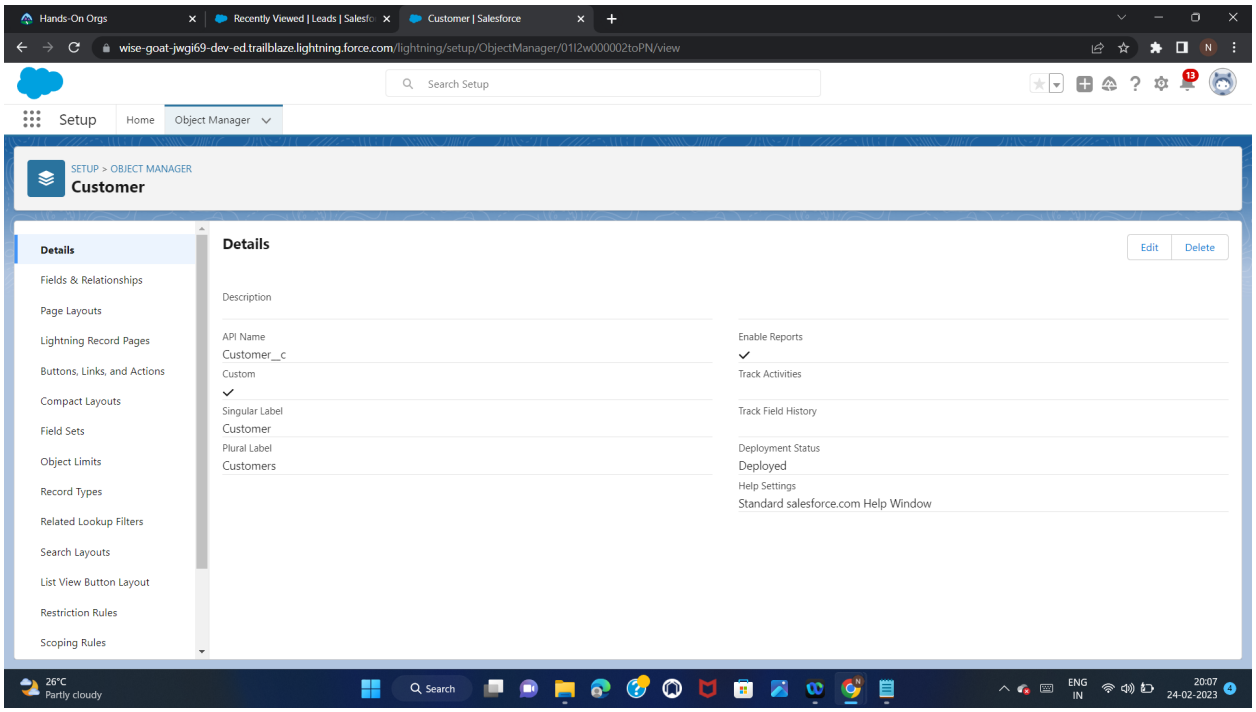
Add Lead,Account,Contact,Opportunity,Case to the Navigation items.



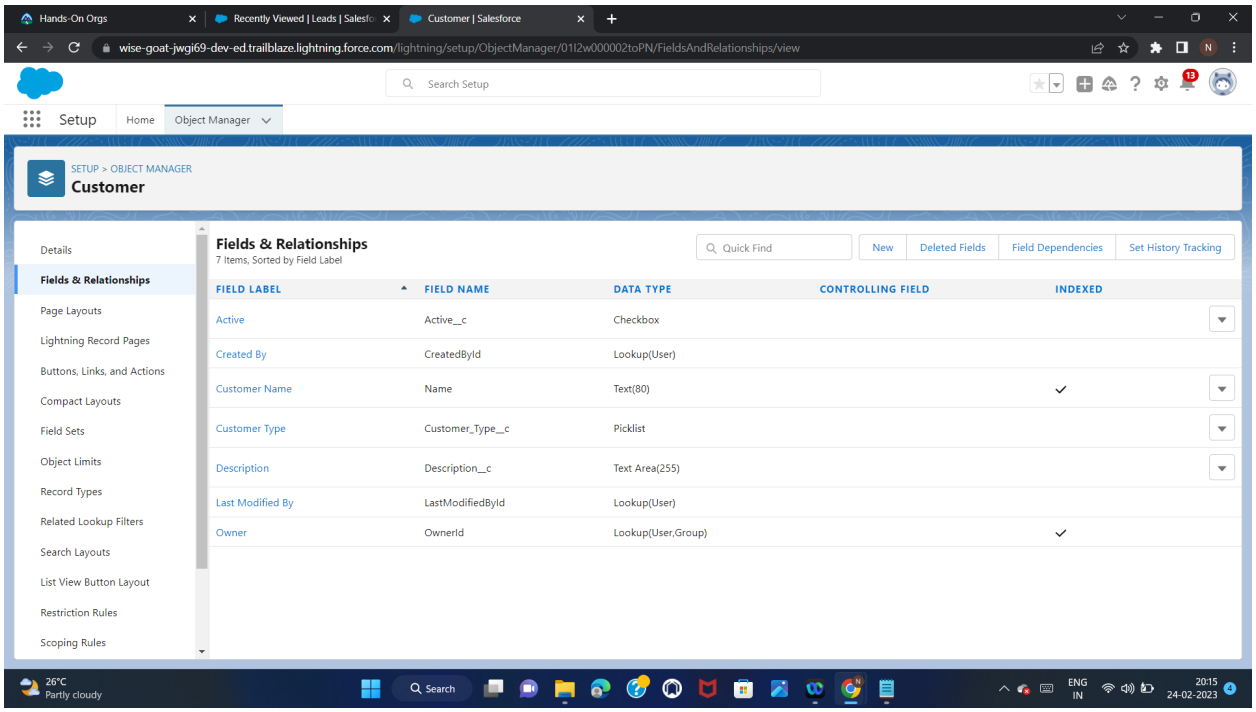
User Profile=”System Administrator”
Save and Finish.



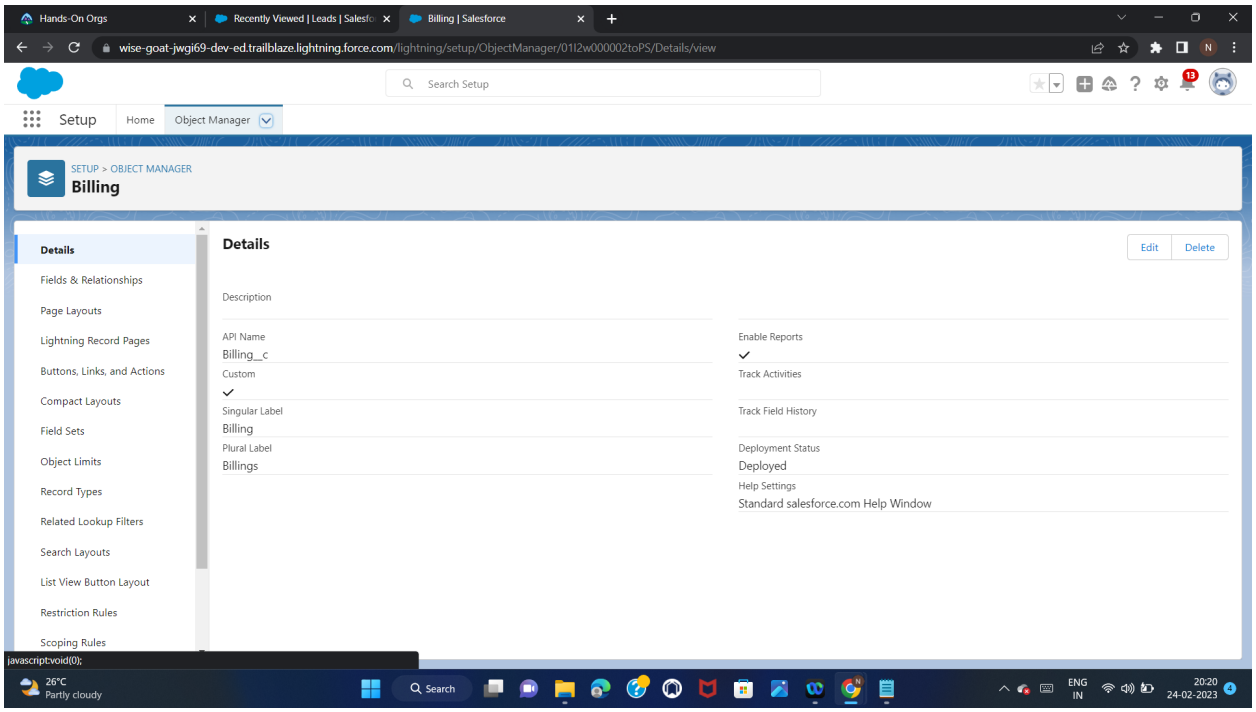
Create a Custom Object.
Label=Customer, Plural Label=Customers
Record Name=Customer Name,Data Type=Text
Allow Report,Allow Search
Create a New Custom Tab



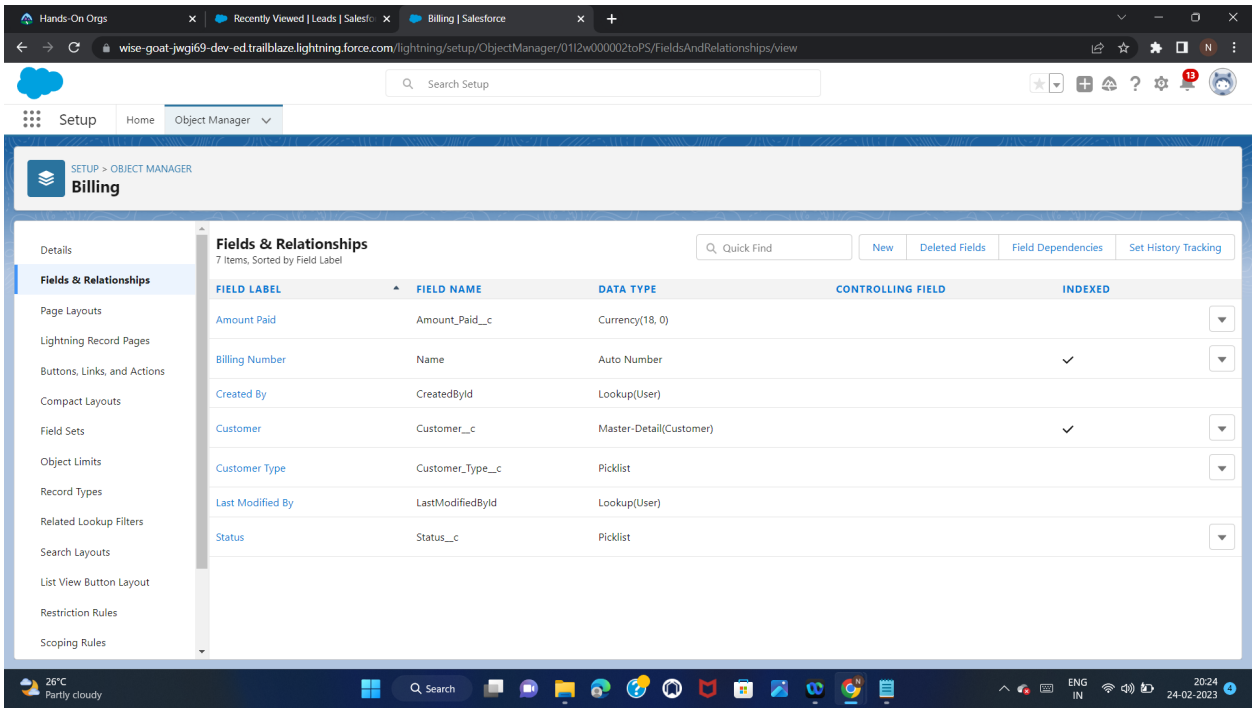
Create a Custom fields for Customer Object.
Label=Active,Checkbox,Save
Label=Customer Type,Picklist Values: Premium,Standard
Label=Description,Text Area,Save



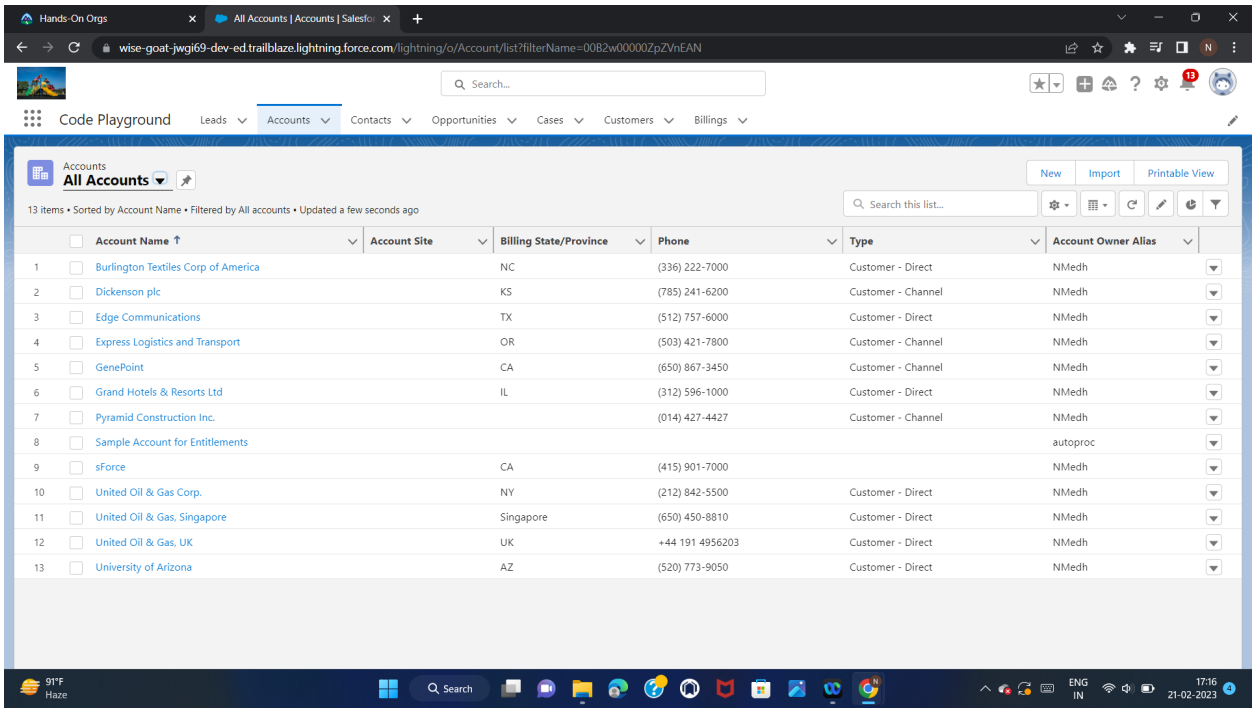
Create a Custom Object
Label=Billing,Plural Label=Billings
Record Name=Bill Number
Type=Auto Number,Format=B-{0000},Starting number 1
Allow Report,Allow Search
Create a New Custom Tab
Add it to “Code Playground” App only,Save



Create a custom fields for Billing Object
Label=Amount Paid,Currency,Save
Label=Customer Type,Picklist,Values: Premium,Standard
Label=Status,Picklist,Values:Paid,Unpaid
Label=Customer,Master-Detail,Related To - Customer custom object

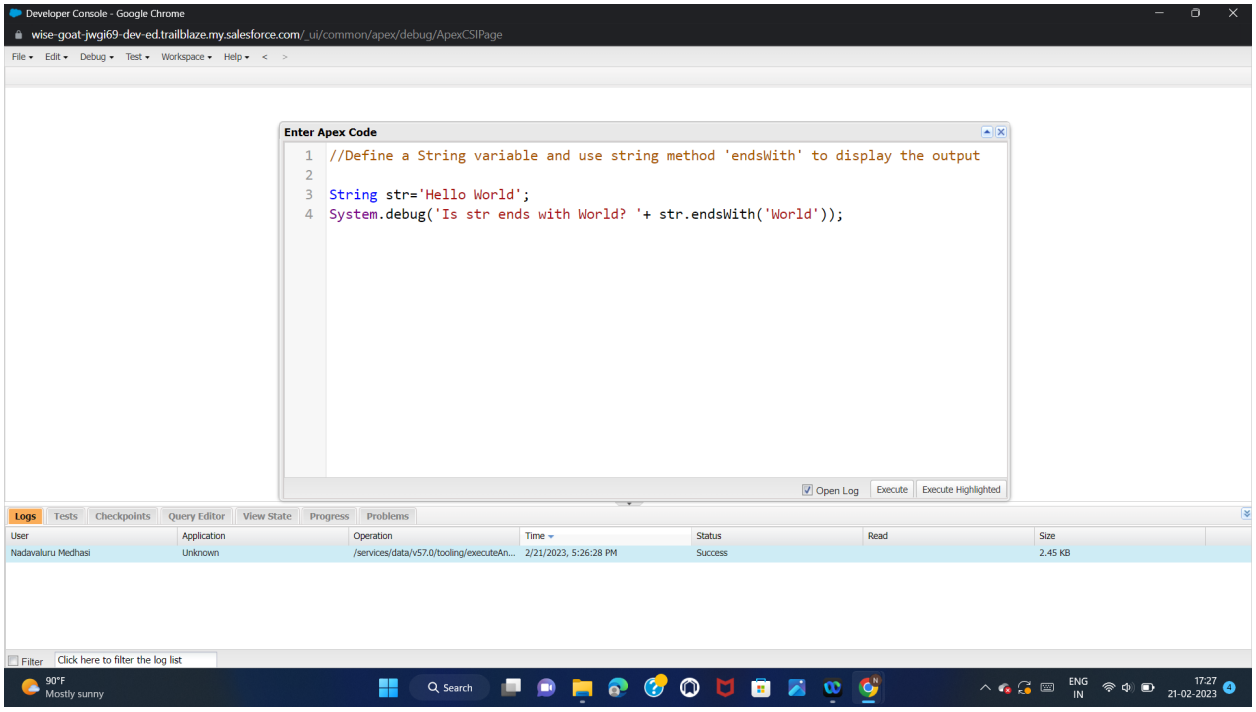


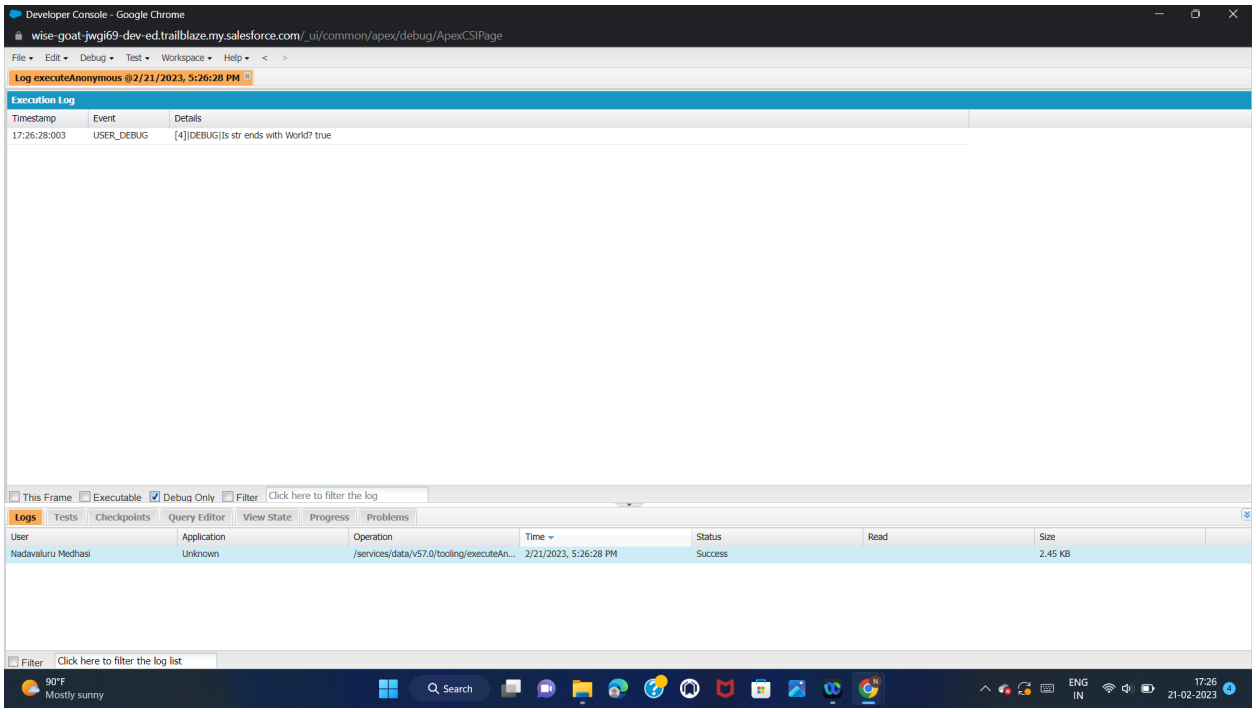
Code Playground looks like this.



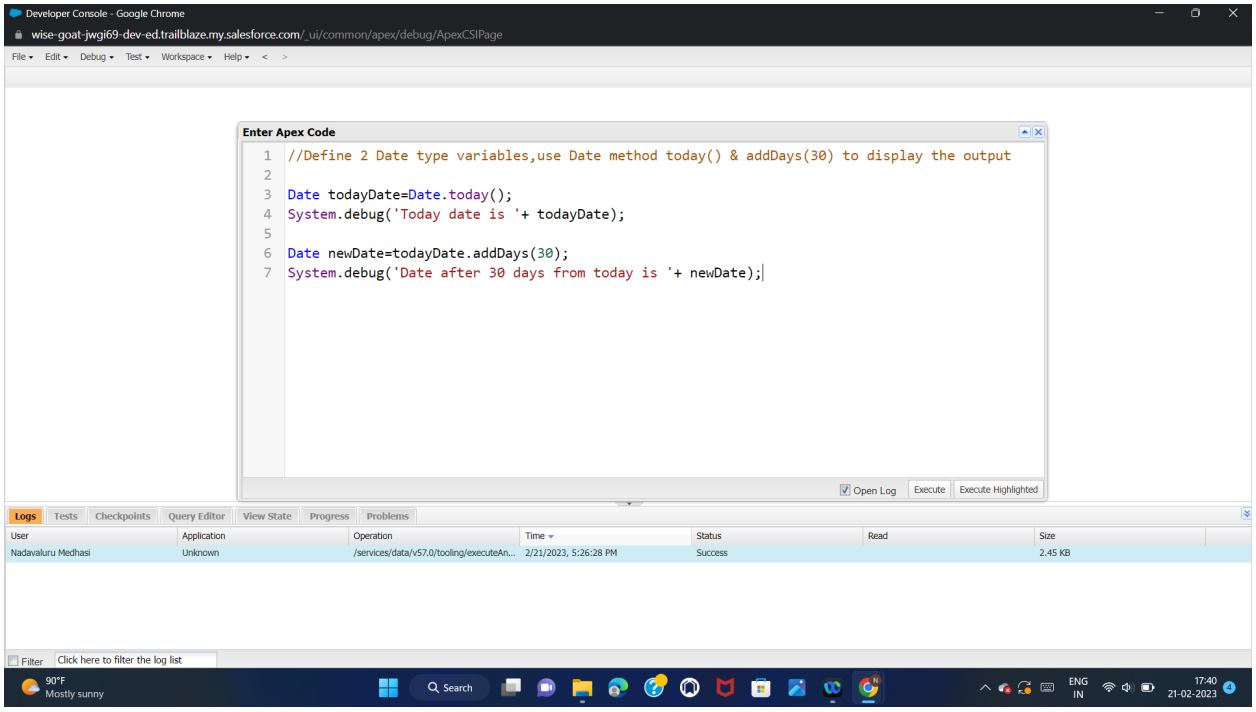
Exercise 2:

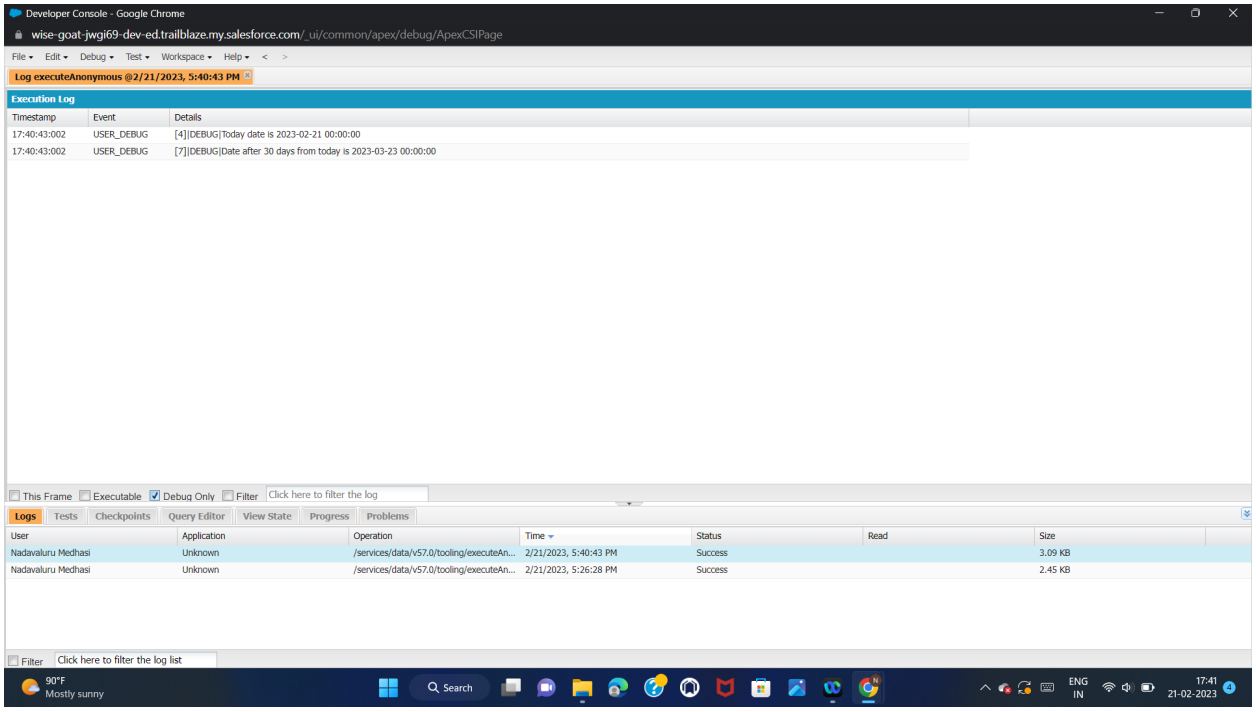
Use Execute Anonymous to define and execute the following code:
Define a String and use string method ‘endsWith’ to display the output



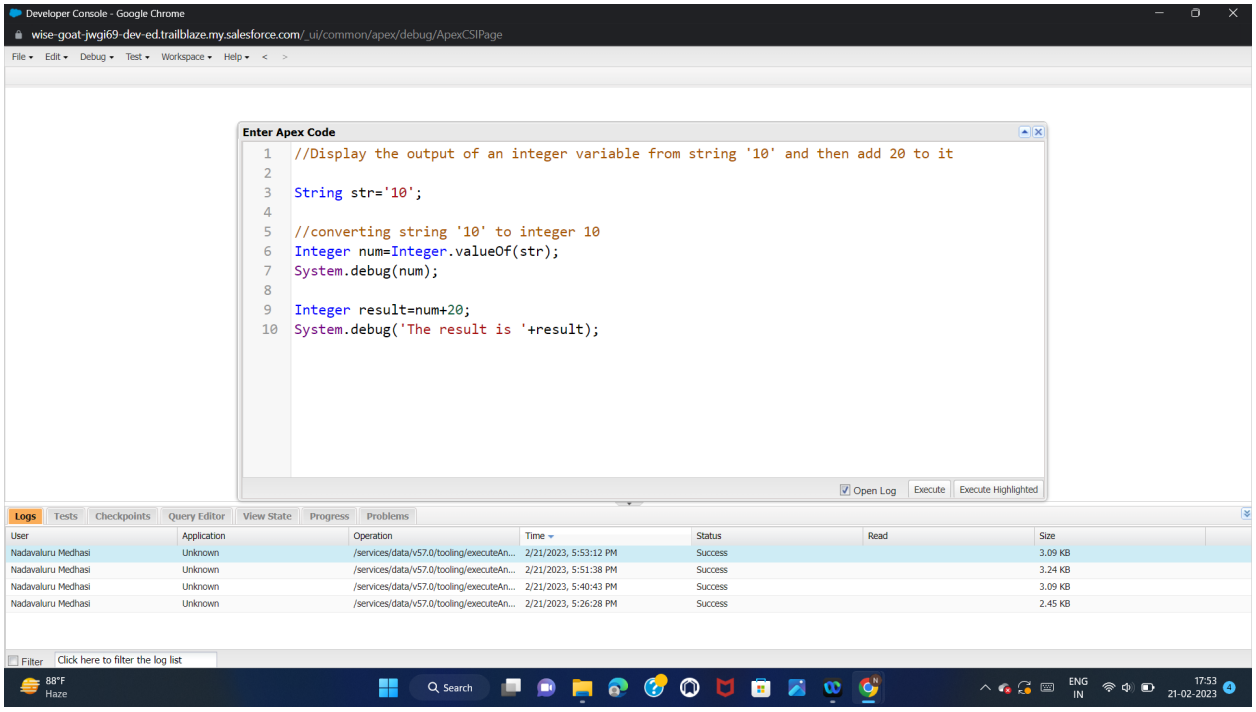


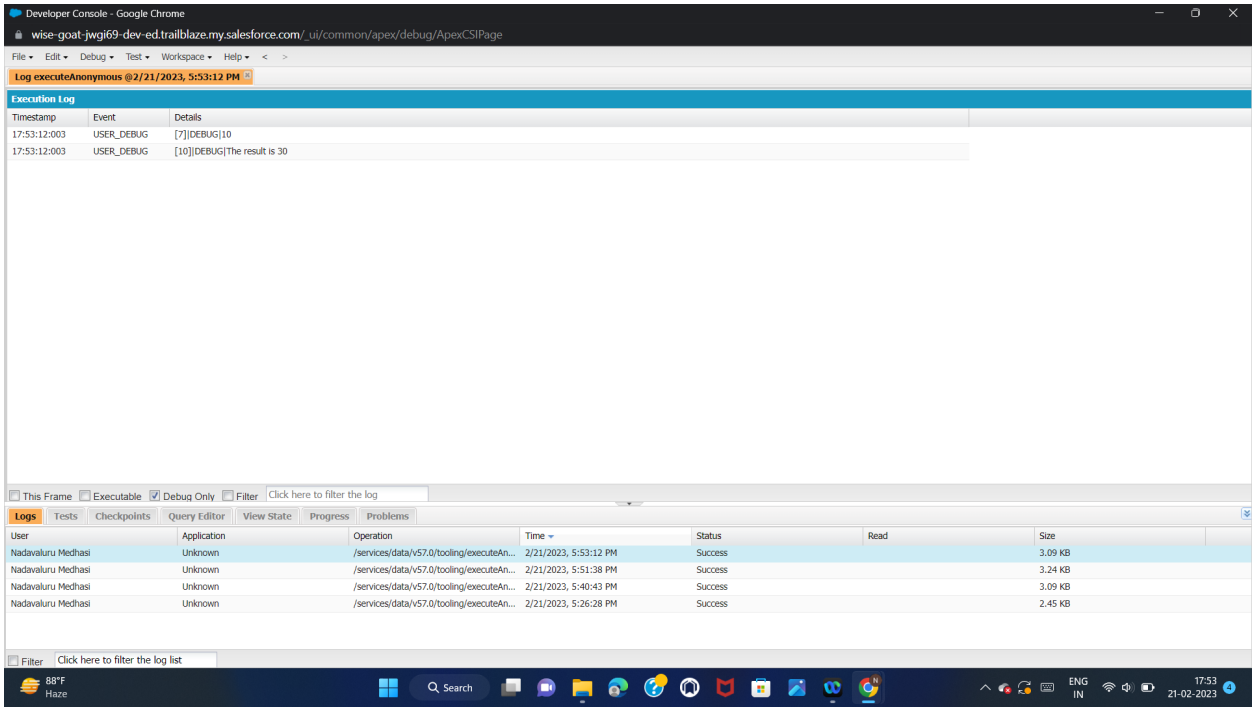
Define 2 Date type variables,use Date method today() and addDays(30) to display the output



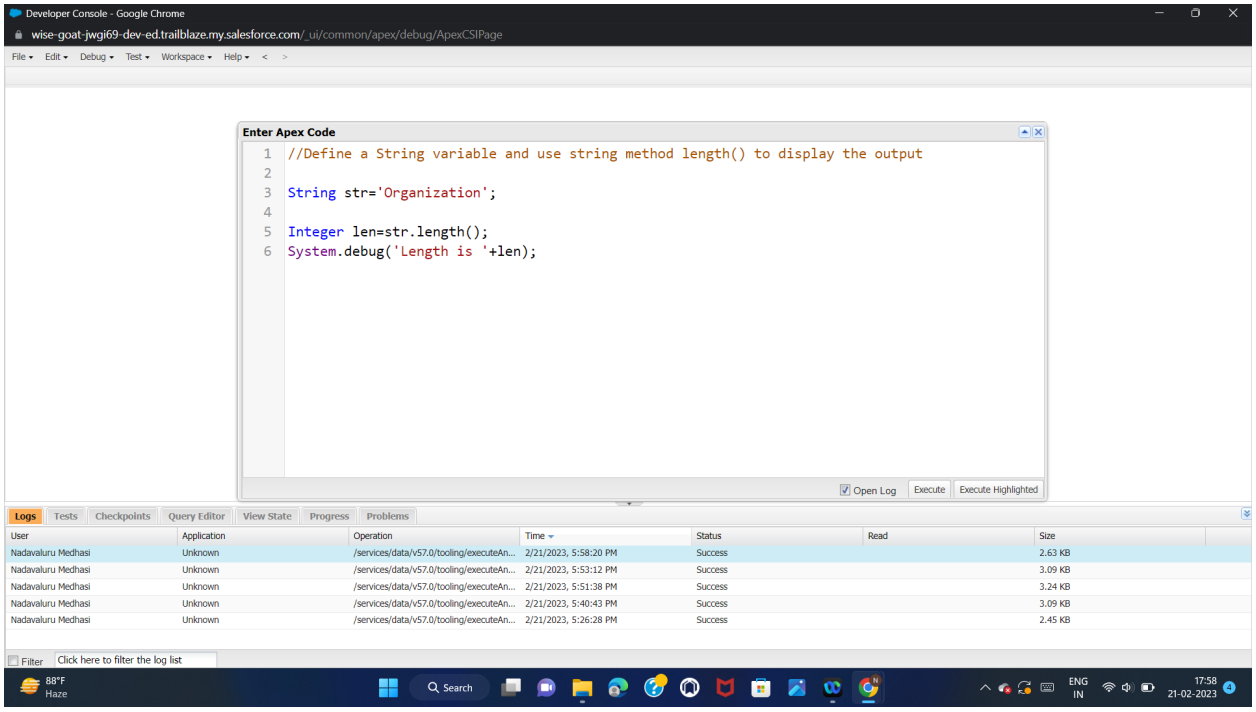


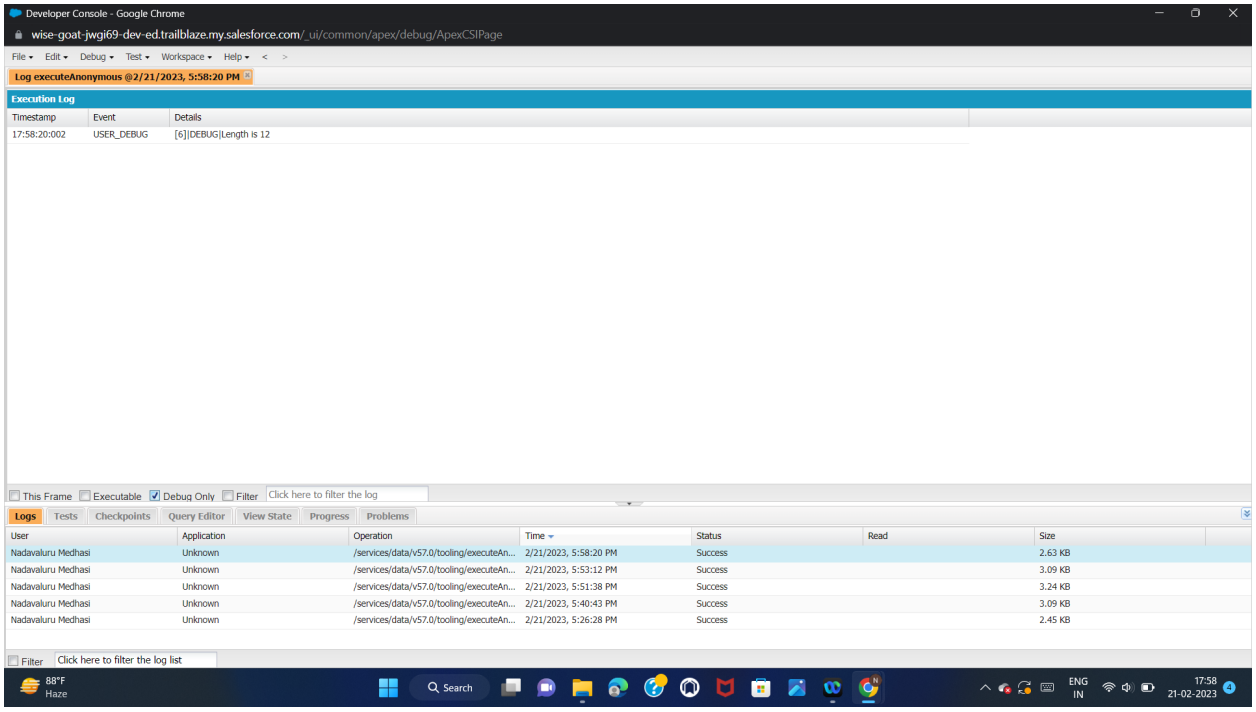
Display the output of an integer variable from string ‘10’ and then add 20 to it



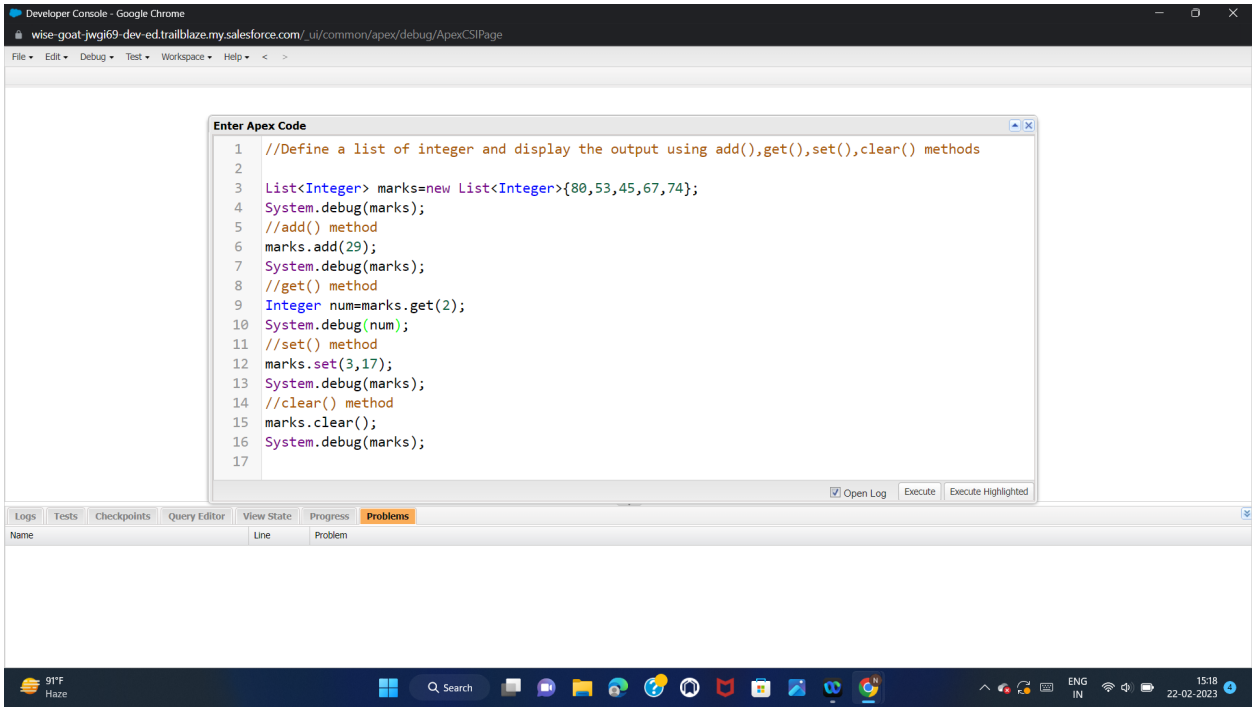


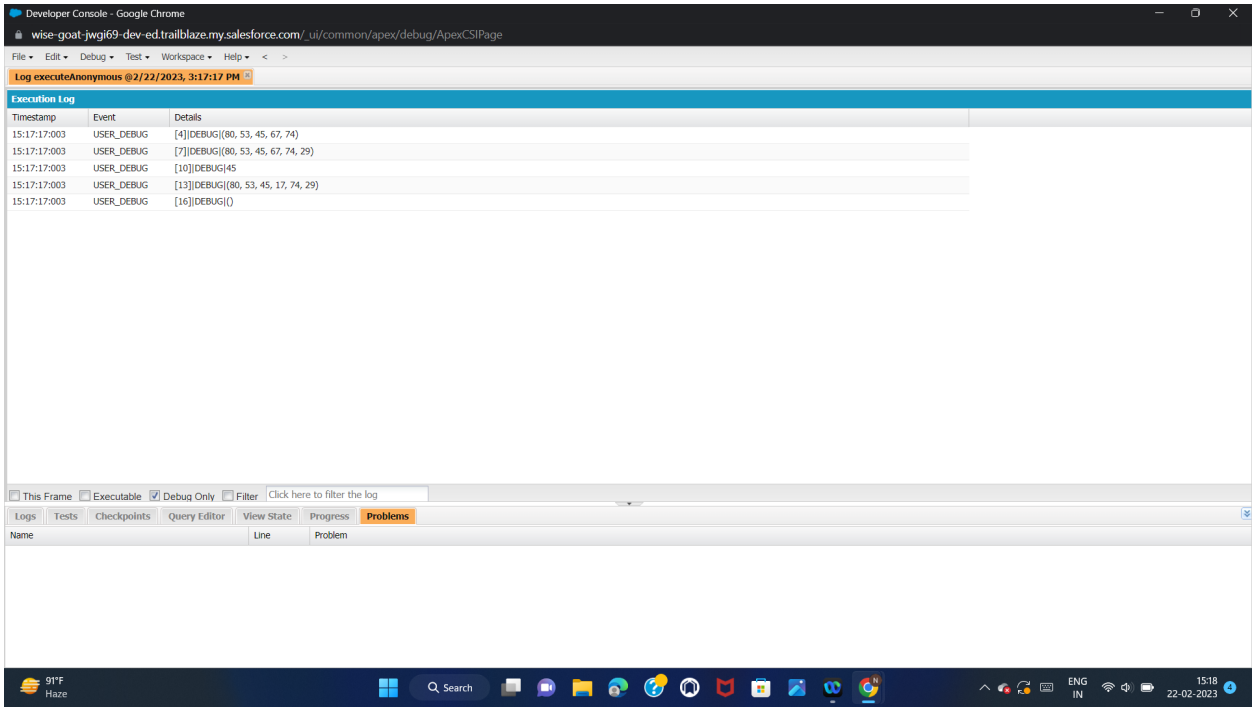
Define a String variable & use string method length() to display the output





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





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

Integer x=20

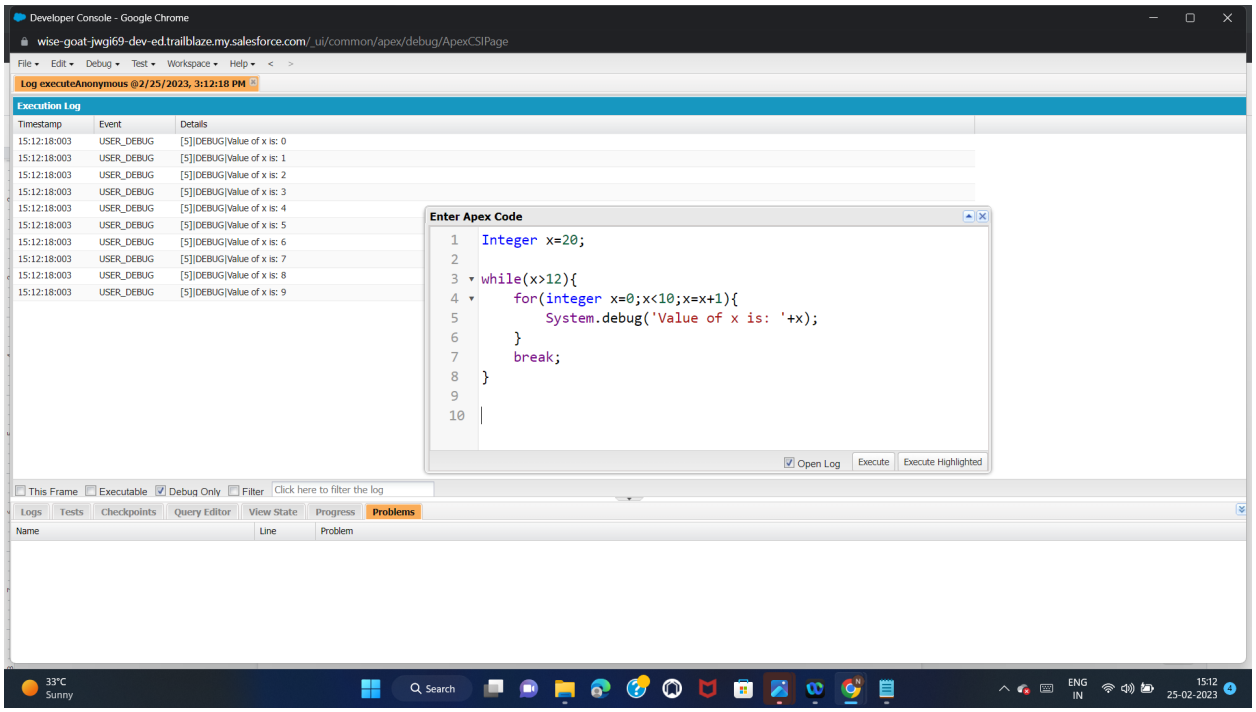
while(x>12){define the code}

for(Integer x=?;x<10;x=x+1)

{

System.debug(‘Value of x is: ‘+x);

}



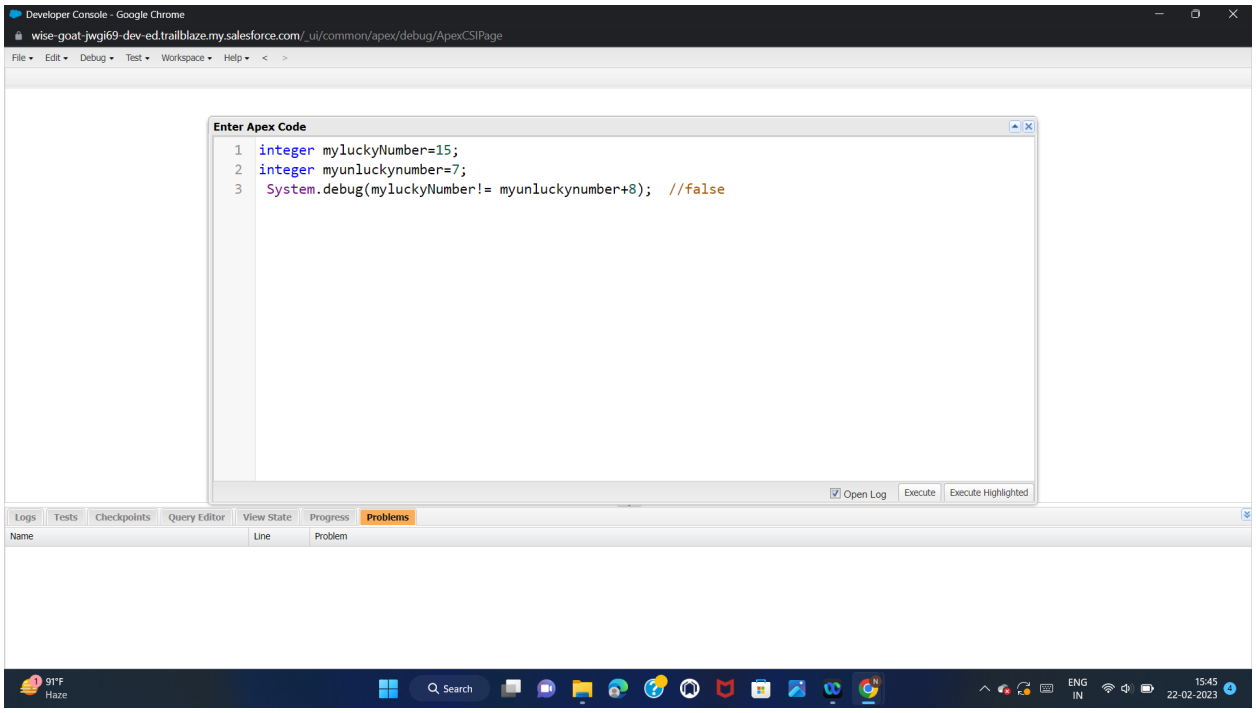
Exercise 3:

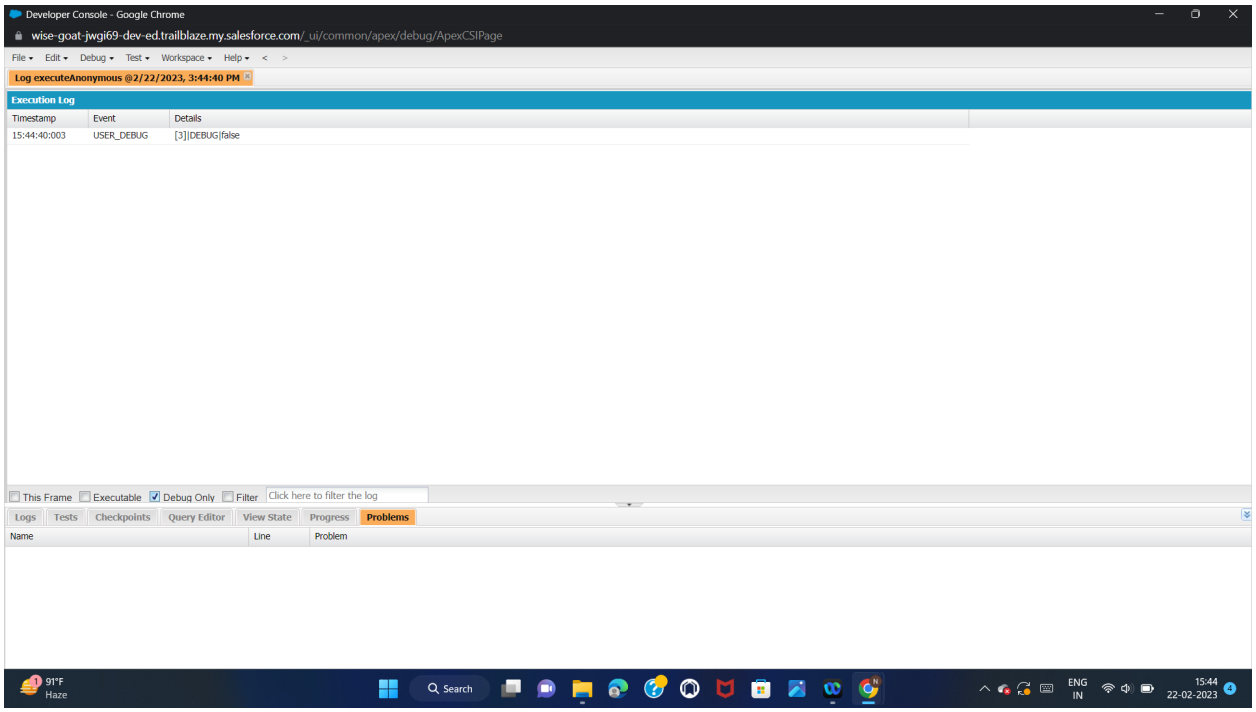
Answer the following in True or False:

Integer myluckyNumber=15;

Integer myunluckyNumber=7;

myluckyNumber != myunluckyNumber





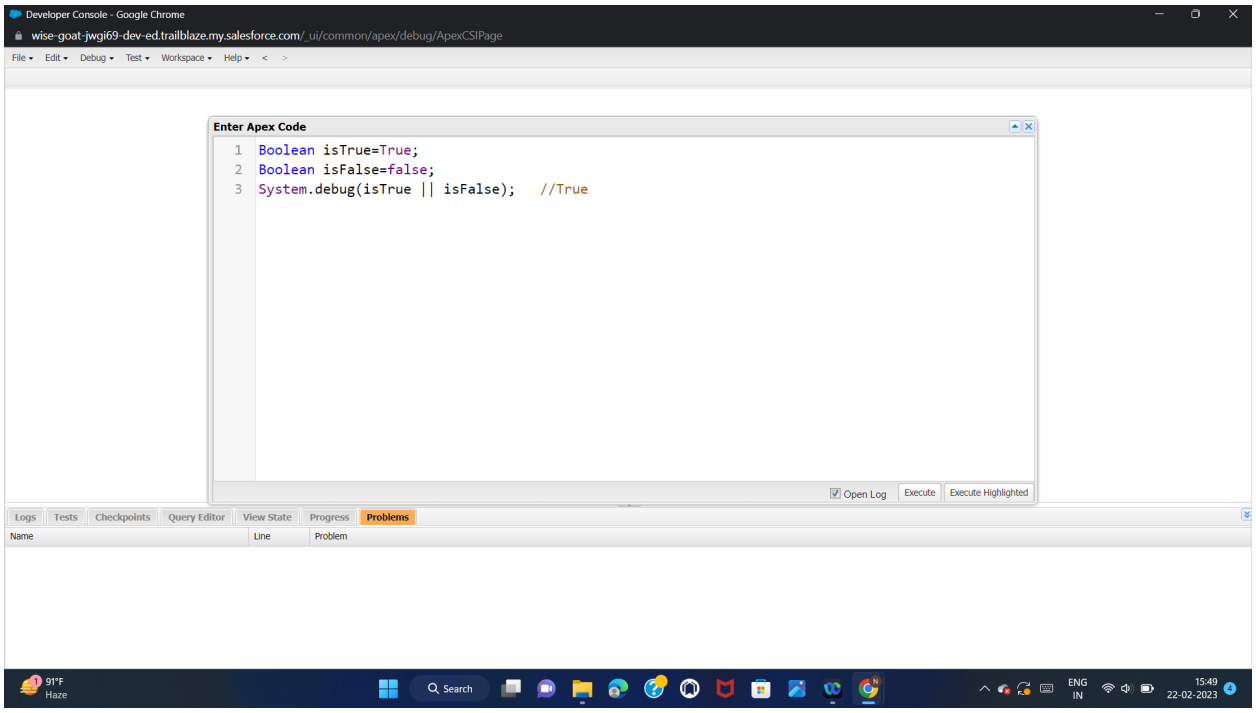
Exercise 4:

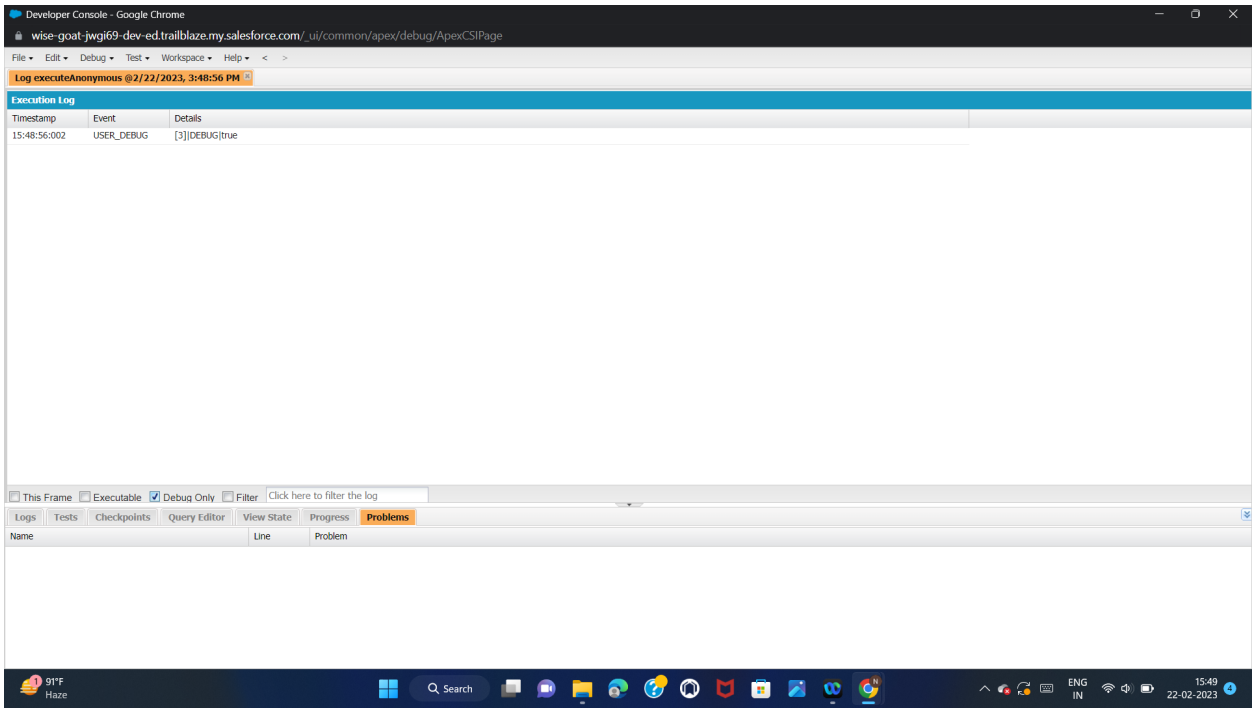
Answer the following in True or False:

Boolean isTrue=True;

Boolean isFalse=false;

isTrue || isFalse





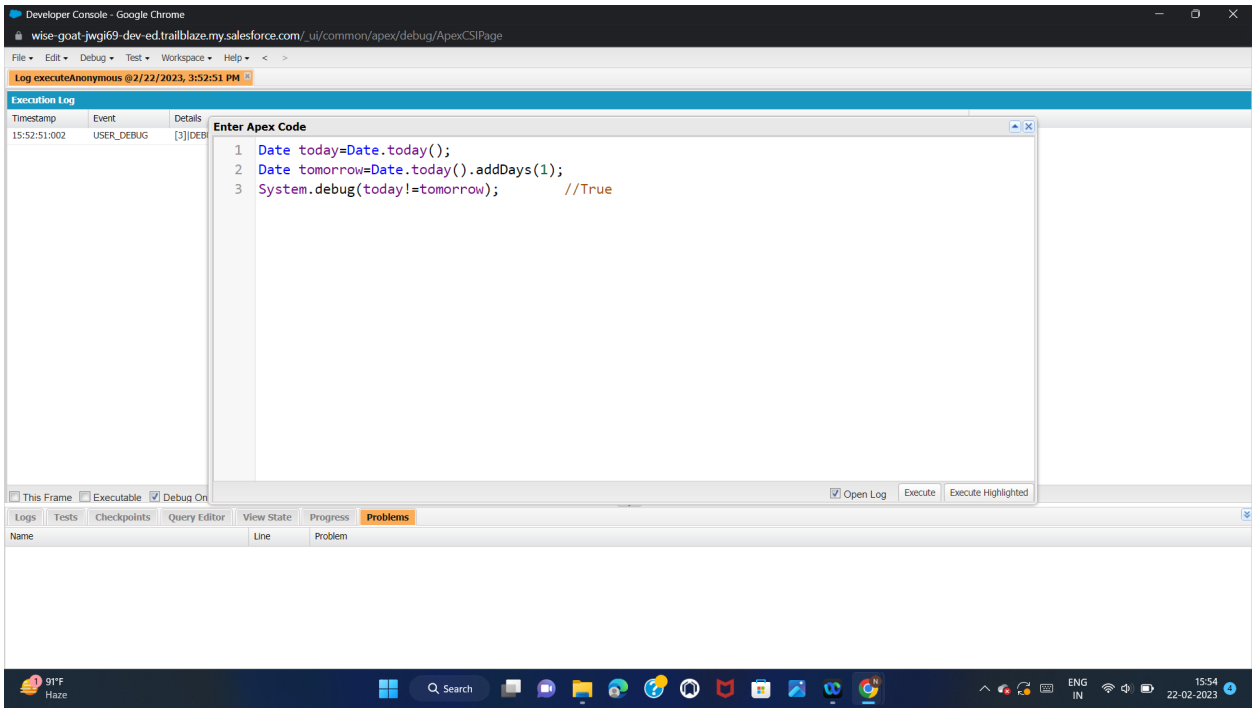
Exercise 5:

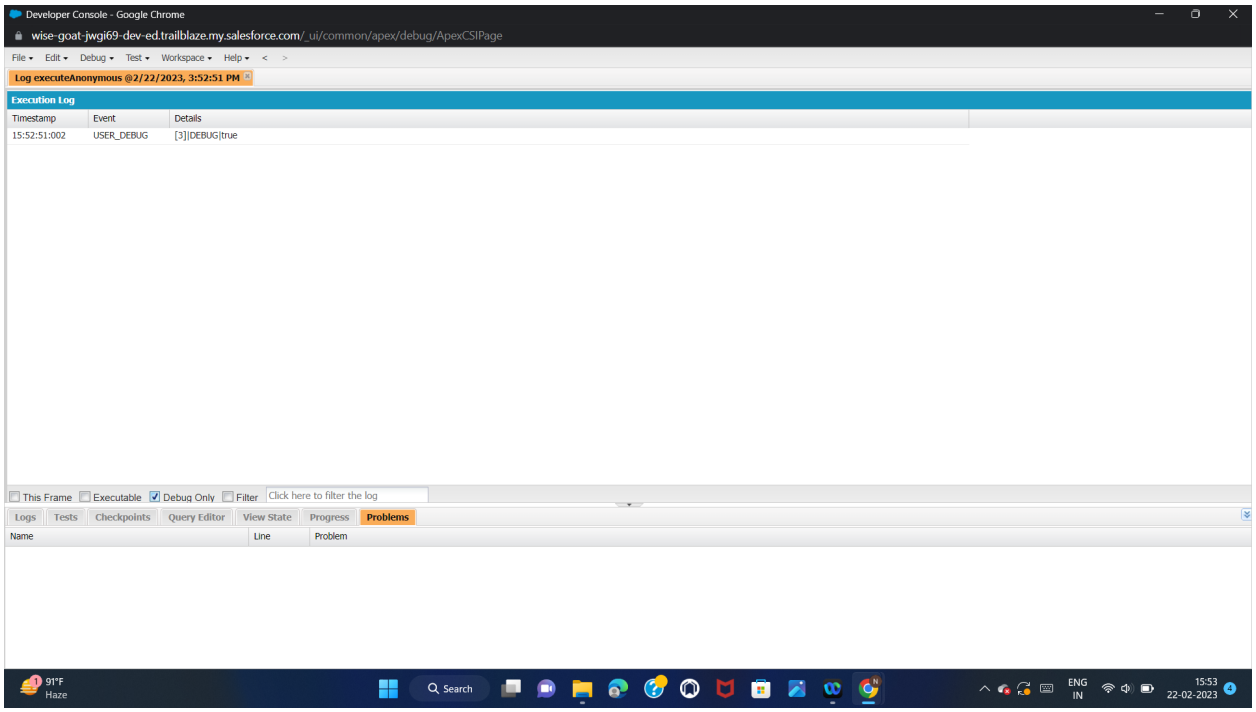
Answer the following in True or False:

Date today=Date.today();

Date tomorrow=Date.today().addDays(1);

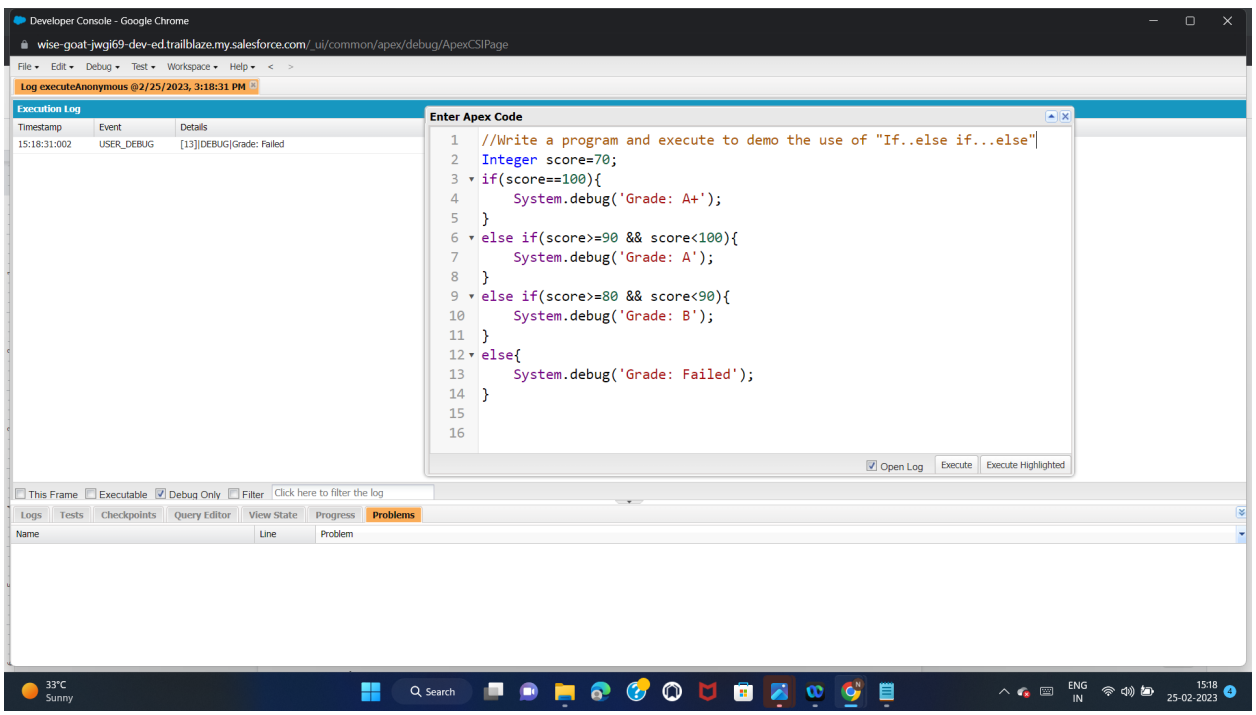
today != tomorrow





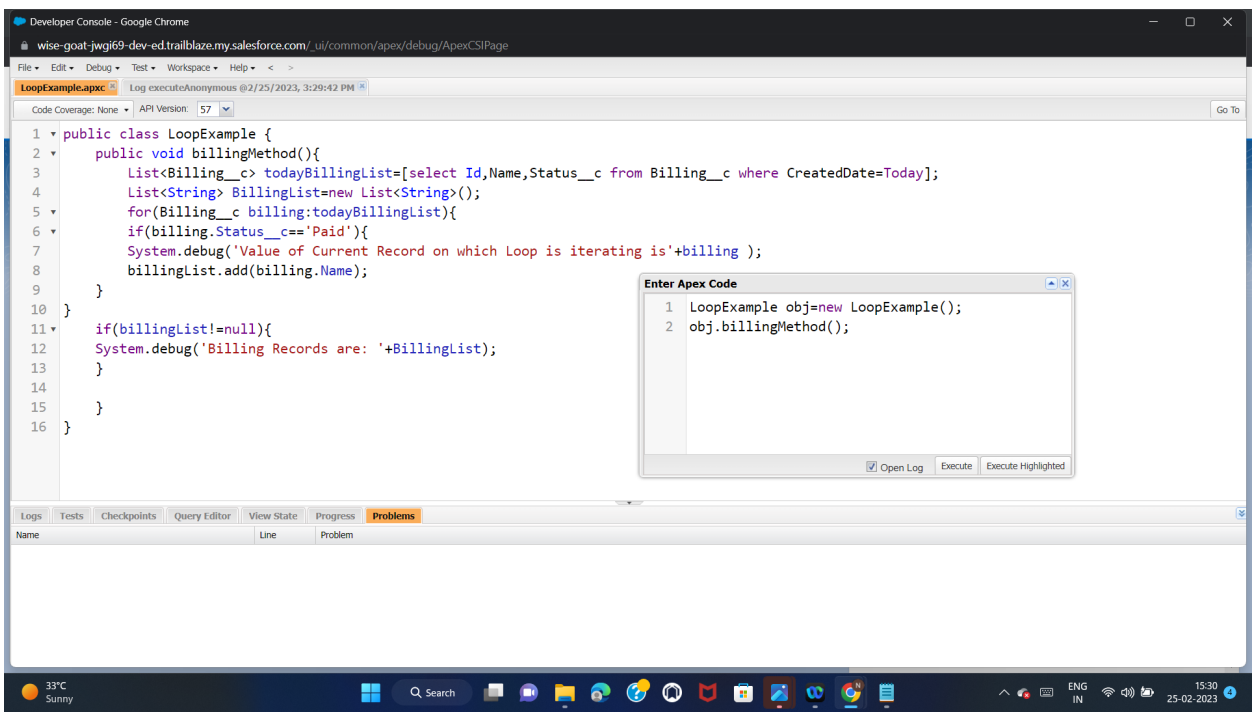
Exercise 6:

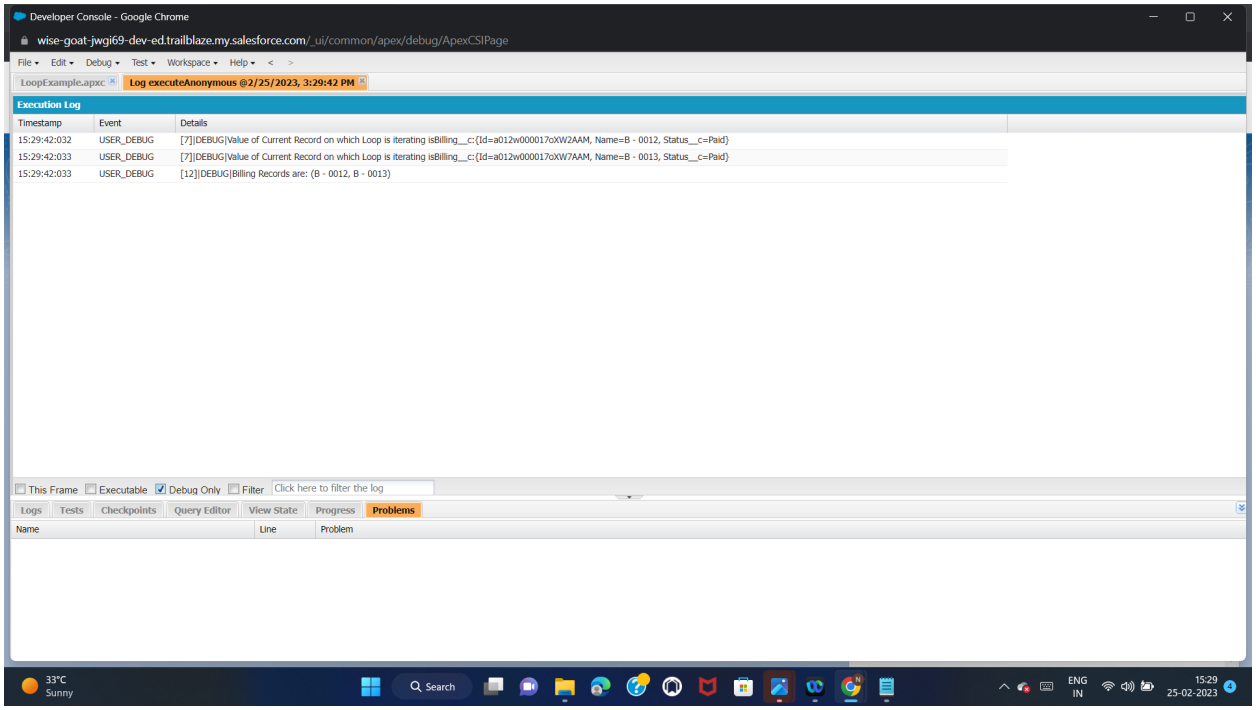
Write a program and execute to demo the use of “If..else if...else”
Define an Integer variable called Score.
System.debug (‘Grade: A+’), if Score is equal to 100.
System.debug (‘Grade: A’), if Score is greater than or equals to 90.
System.debug (‘Grade: B’), if Score is greater than or equals to 80.
System.debug (‘Grade: Failed’), define as final else.



Exercise 7:

Write a program to execute and demo the use of “Apex – for Loop”
Initialize the Billing object list to store the Billing Records created today, use SOQL.
Use for Loop to iterate over the list and check if the Billings status = Paid.
System.debug (‘Value of Current Record in the Loop’+ Billing List);
if Status is paid then add the Billings into a List of String.
System.debug('Value of BillingList '+BillingList);
Note: Create at least 2 Billing records with Status = Paid for this exercise.





Exercise 8:

Write a Class to demo the use of Constants in Apex
Class Name = DiscountClass.
Define a Final Decimal variable called “regularDiscount”.
Define a Decimal variable called “finalPrice”.
Define a Method called “calculateDiscount(Integer price)”, takes a Decimal data type.
finalPrice = price - price*regularDiscount;
Don’t forget to return a Decimal data type.

Developer Console - Google Chrome

wise-goat-jwgi69-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

DiscountClass.apxc | Log executeAnonymous @2/24/2023, 9:54:56 PM

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

```
//Write a class to demo the use of constants in Apex

public class DiscountClass {
    Final Decimal regularDiscount=0.05;
    Decimal finalPrice;

    public Decimal calculateDiscount(Integer price){
        finalPrice=price-price*regularDiscount;
        return finalPrice;
    }
}
```

Enter Apex Code

```
DiscountClass obj=new DiscountClass();
Decimal finalPrice=obj.calculateDiscount(100);
System.debug(finalPrice);
```

Open Log | Execute | Execute Highlighted

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

Name	Line	Problem

25°C Partly cloudy | Search | ENG IN | 21:55 24-02-2023

Developer Console - Google Chrome

wise-goat-jwgi69-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

DiscountClass.apxc | Log executeAnonymous @2/24/2023, 9:54:56 PM

Execution Log

Timestamp	Event	Details
21:54:56:033	USER_DEBUG	[3]|DEBUG|95.00

This Frame | Executable | Debug Only | Filter | Click here to filter the log

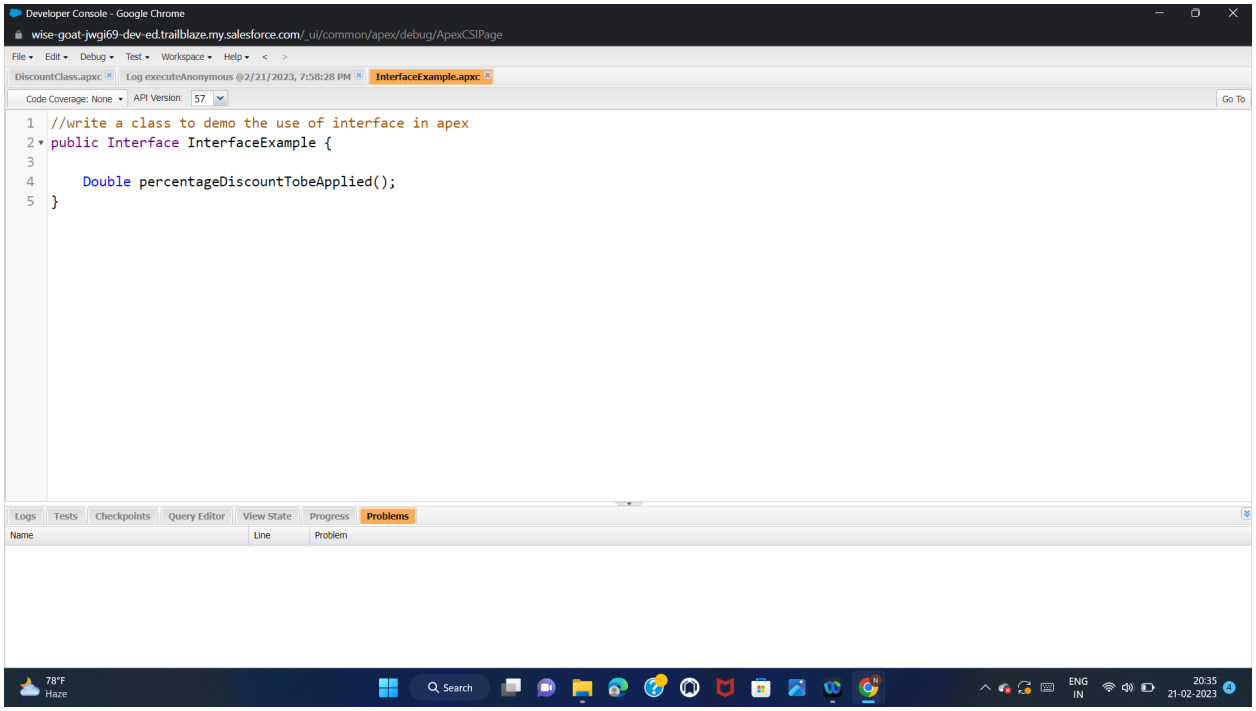
Logs | Tests | Checkpoints | Query Editor | View State | Progress | Problems

Name	Line	Problem

25°C Partly cloudy | Search | ENG IN | 21:56 24-02-2023

Exercise 9:

Write a Class to demo the use of Interface in Apex
Interface Class Name = InterfaceExample.
Method Signature = Double percentageDiscountTobeApplied();
Create an Apex Class to implement the Interface Class, Name = PremiumCustomer
Call the interface method to return 30% Discount.
Create another Apex Class to implement the Interface Class, Name = normalCustomer
Call the interface method to return 10% Discount.



Developer Console - Google Chrome

wise-goat-jwgi69-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

InterfaceExample.apxc | PremiumCustomer.apxc

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

```
public class PremiumCustomer implements InterfaceExample {

    public  Double percentageDiscountTobeApplied(){
        return 0.3;
    }
}
```

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

Name	Line	Problem

78°F Haze | Search | ENG IN | 20:38 21-02-2023

Developer Console - Google Chrome

wise-goat-jwgi69-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

InterfaceExample.apxc | PremiumCustomer.apxc | NormalCustomer.apxc | Log executeAnonymous @2/25/2023, 4:00:30 PM

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

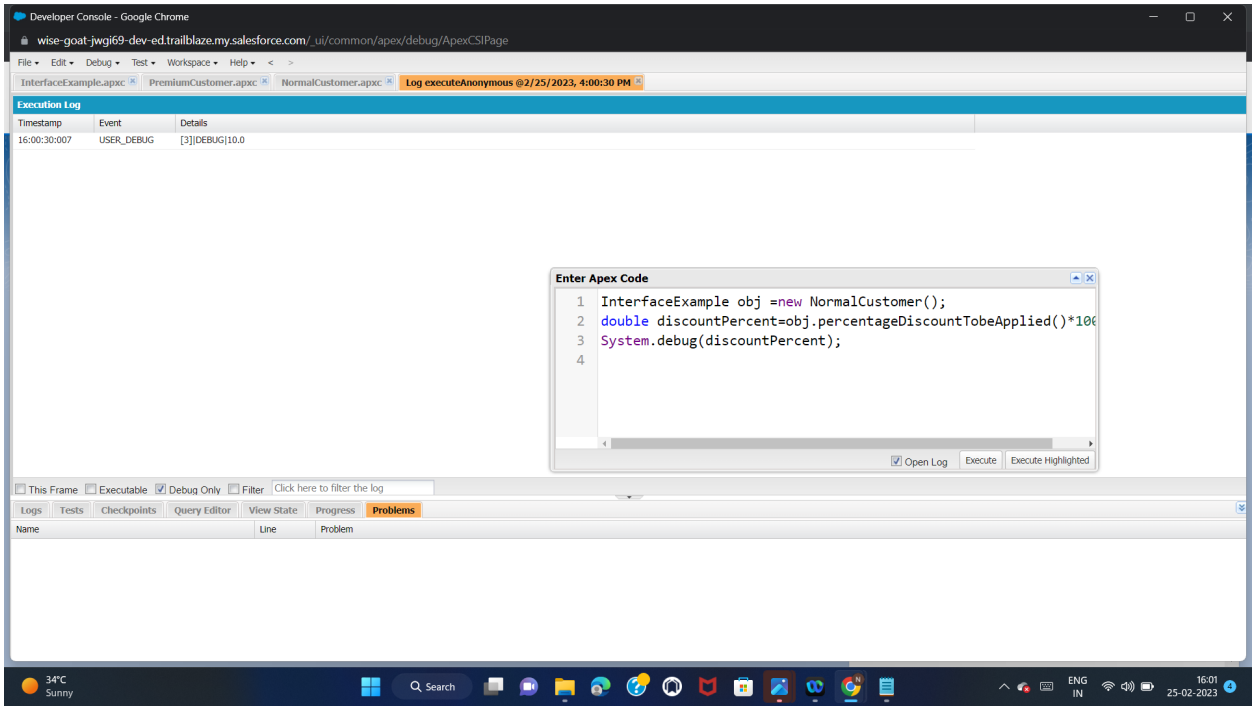
```
public class  NormalCustomer implements InterfaceExample {

    public  Double percentageDiscountTobeApplied(){
        return 0.1;
    }
}
```

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

Name	Line	Problem

34°C Sunny | Search | ENG IN | 16:01 25-02-2023



Exercise 10:

Demo on DML Insert Operation Using Database methods
Insert a Customer Record First using simple DML Statement.
Customer Name = “Wipro”, Type = “Premium”.
Insert Billing record using Database methods.
Billing Status = Paid, Amount Paid = 5000000, link Billing record to Customer using the customer ID.
Use a List<Billing c> to perform the Database insert method & store the result in srList.
Iterate through the Success/Error result, show inserted record ID, or the error message with fields.

Developer Console - Google Chrome

wise-goat-jwgi69-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

BillingRecordInsertion.apxc

Code Coverage: None API Version: 57 Go To

```
public class BillingRecordInsertion {
    public void recordInsertion(){
        Customer__c cus=new Customer__c();
        cus.Name='Wipro';
        cus.Customer_Type__c='Premium';
        insert cus;
        ID cust_id=cus.Id;
        Billing__c[] bill=new List<Billing__c>{
            new Billing__c(Status__c='Paid',Amount_Paid__c=5000000,Customer__c=cust_id)

        };
        Database.SaveResult[] srList=Database.insert(bill,false);
        for(Database.SaveResult saveres:srList){
            if(saveres.isSuccess()){
                System.debug('Record Inserted. Billing ID: '+saveres.getId());
            }
            else{
                for(Database.Error err :saveres.getErrors()){
                    System.debug(err.getStatusCode()+': '+err.getMessage());
                    System.debug('Billing fields which have error: '+err.getFields());
                }
            }
        }
```

Enter Apex Code

```
BillingRecordInsertion obj=new BillingRecordInsertion();
obj.recordInsertion();
```

Open Log Execute Execute Highlighted

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem

27°C Partly cloudy Search ENG IN 19:12 23-02-2023

Developer Console - Google Chrome

wise-goat-jwgi69-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

BillingRecordInsertion.apxc Log executeAnonymous @2/23/2023, 7:01:08 PM

Execution Log

Timestamp	Event	Details
19:01:08:080	USER_DEBUG	[15]|DEBUG|Record Inserted. Billing ID: a012w000017ncc9AAA

This Frame Executable Debug Only Filter Click here to filter the log

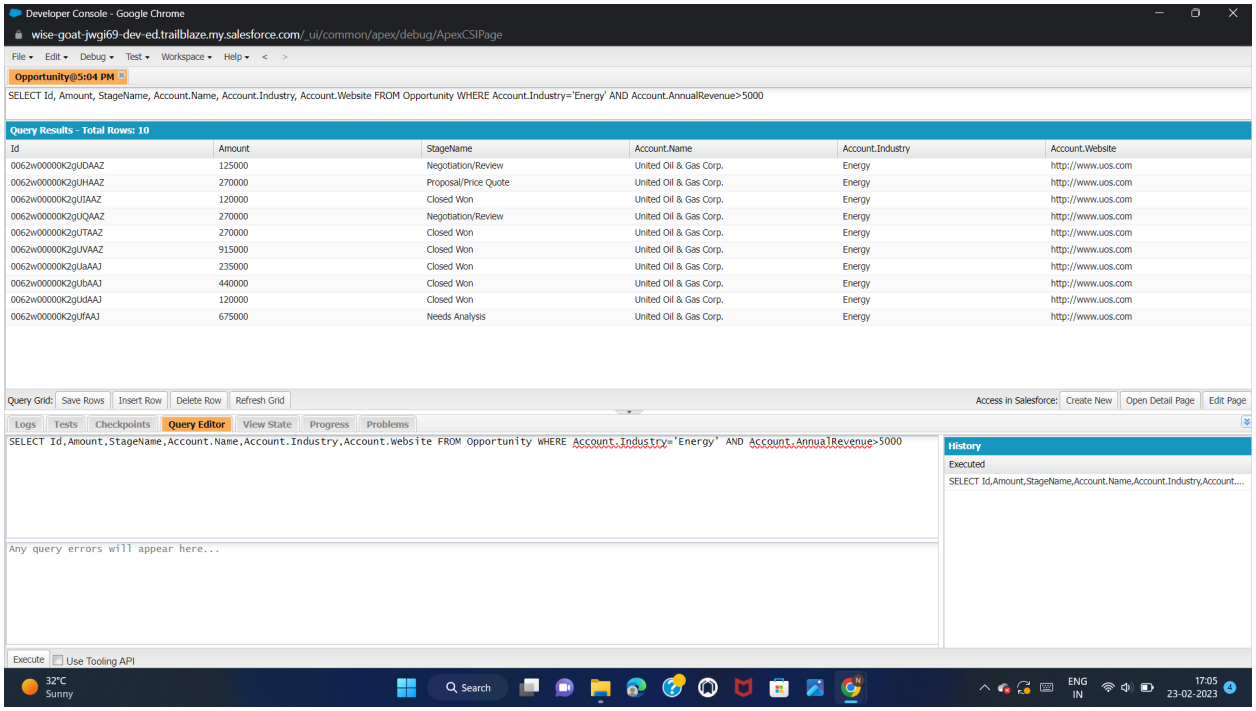
Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem

29°C Partly cloudy Search ENG IN 19:03 23-02-2023

Exercise 11:

Write and execute SOQL queries from Developer Console
Display ID, Amount, Stage, Account Name, Account Industry, Account Website From Opportunity
Add a Where Clause “Account Industry = Energy”
AND “Account Annual Revenue > 5000”



Exercise 12:

Write an Apex Trigger, Name = CustomerTrigger.
Create a Billing Record when on Customer Object the 'Status' field changes to Active from Inactive.
Billing “Status” must be: Paid, Amount Paid = 1000000
Perform the DML on a List.

Developer Console - Google Chrome

wise-goat-jwgi69-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

DisqualifyTestLeads.apxt | DisqualifyTest.apxc | TestRun @ 7:56:31 pm | CustomerTrigger.apxt

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

```
trigger CustomerTrigger on Customer__c (after update) {
    List<Billing__c> billingRecords=new List<Billing__c>();
    for(Customer__c cust:Trigger.new){
        if(cust.Active__c==true && Trigger.oldmap.get(cust.id).Active__c==false){
            Billing__c bill=new Billing__c();
            bill.Status__c='Paid';
            bill.Amount_Paid__c=1000000;
            bill.Customer__c=cust.Id;
            billingRecords.add(bill);
        }
    }
    if(billingRecords!=null){
        insert billingRecords;
    }
}
```

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

Status	Test Run	Enqueued Time	Duration	Failures	Total

Overall Code Coverage

Class	Percent	Lines
Overall	**16%**	
BillingRecordInsertion	0%	0/12
CustomerTrigger	0%	0/10
DiscountClass	0%	0/4
DisqualifyTestLeads	100%	8/8
LoopExample	0%	0/7

25°C Partly cloudy | Search | ENG IN | 19:30 26-02-2023

Hands-On Orgs | Wipro | Salesforce

wise-goat-jwgi69-dev-ed.trailblaze.lightning.force.com/lightning/r/Customer__c/a002w00000ZIBUAAA5/view

Search...

Code Playground | Leads | Accounts | Contacts | Opportunities | Cases | Customers | Billings

Customer
Wipro

New Contact | Edit | New Opportunity

Related | **Details**

Customer Name
Wipro

Owner
Nadavaluru Medhasi

Active

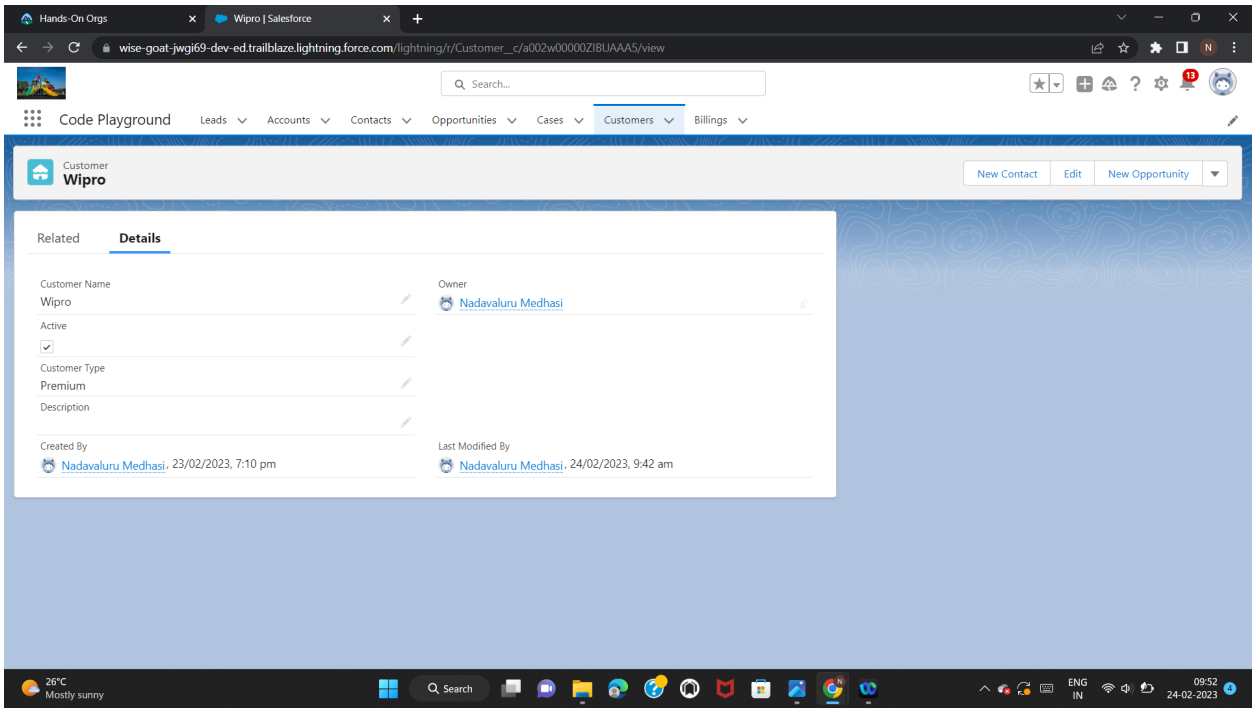
Customer Type
Premium

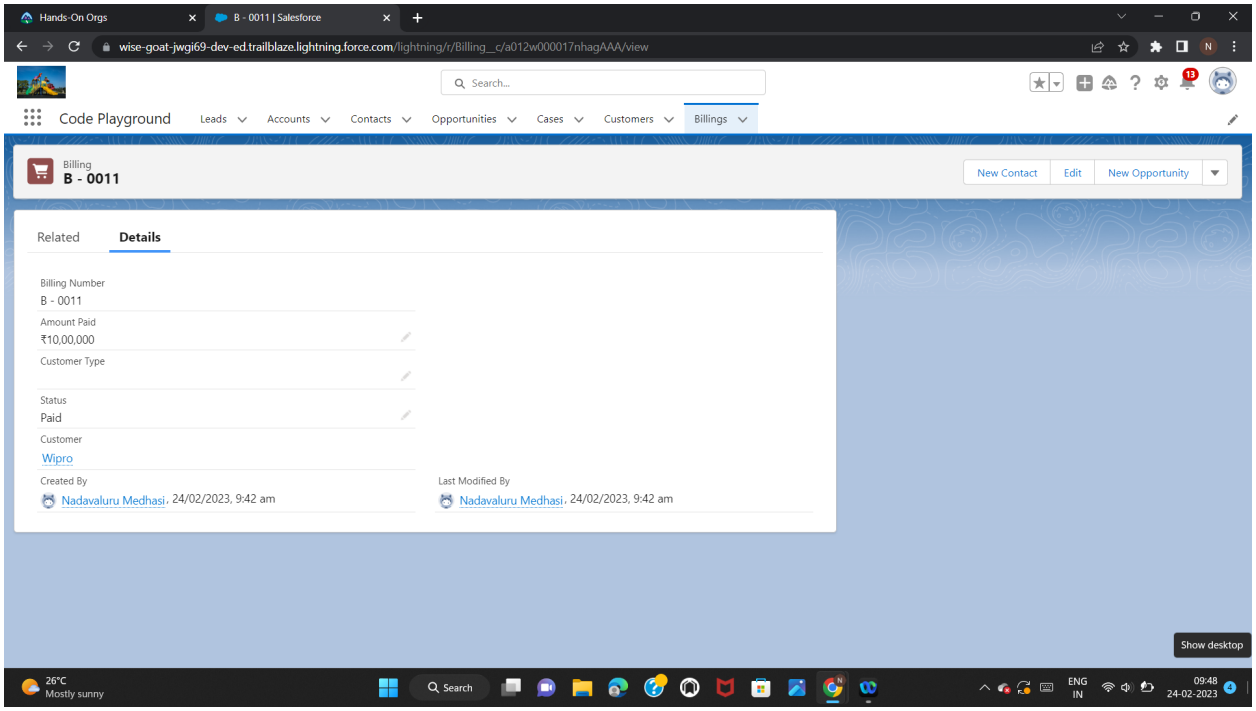
Description

Created By
Nadavaluru Medhasi, 23/02/2023, 7:10 pm

Last Modified By
Nadavaluru Medhasi, 23/02/2023, 7:10 pm

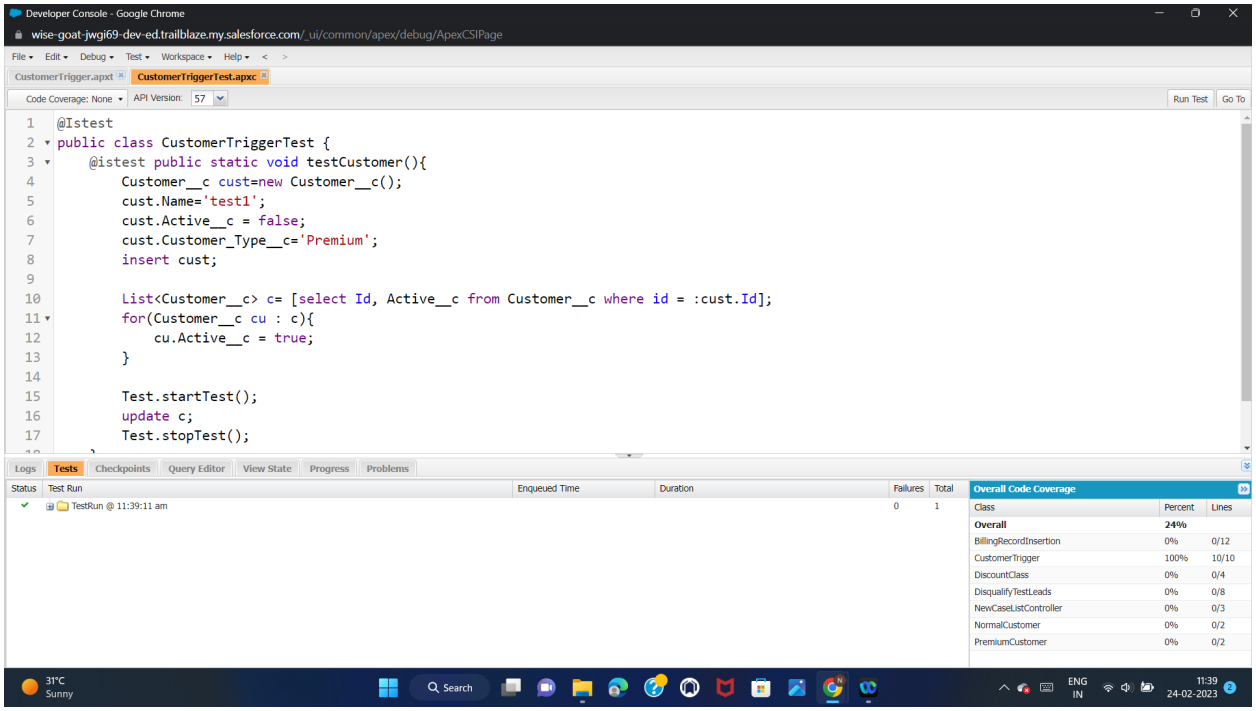
Search | ENG IN | 09:41 24-02-2023

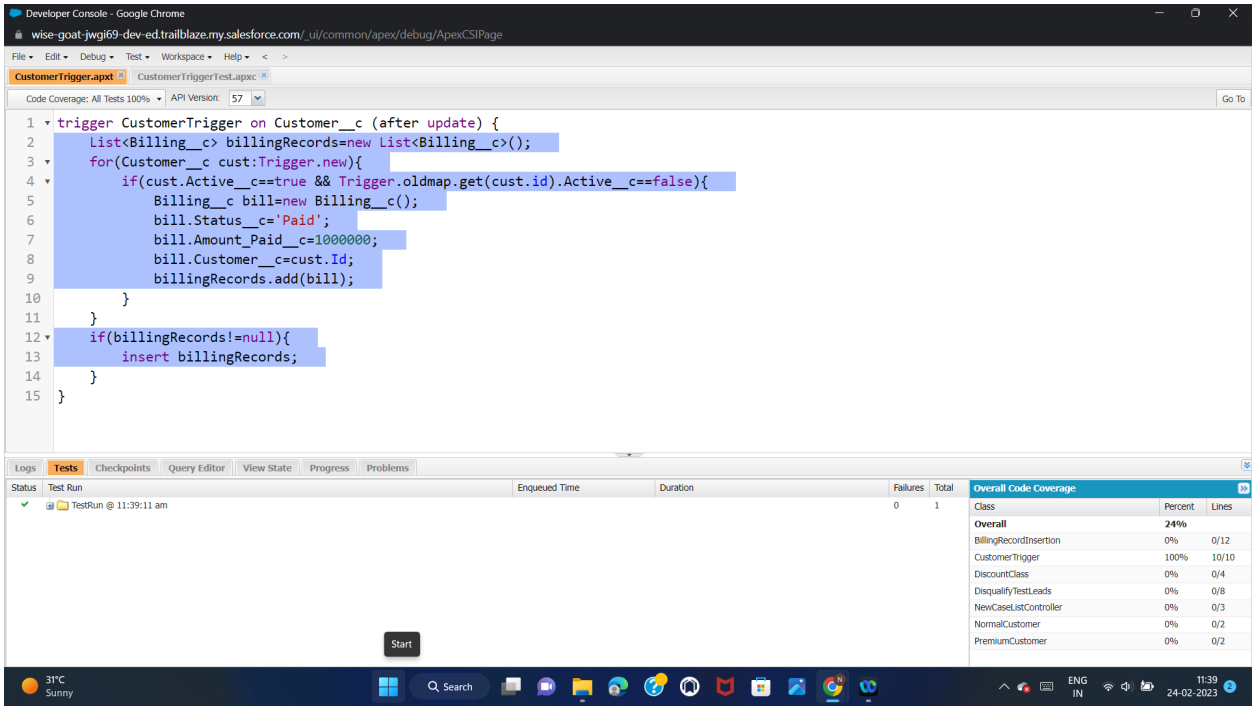

Hands-On Orgs
Wipro | Salesforce
wise-goat-jwgi69-dev-ed.trailblaze.lightning.force.com/lightning/r/Customer__c/a002w00000ZIBUAAA5/view
Search...
Code Playground
Leads
Accounts
Contacts
Opportunities
Cases
Customers
Billings
Customer
Wipro
New Contact
Edit
New Opportunity
Related
Details
Customer Name
Wipro
Owner
Nadavaluru Medhasi
Active
Customer Type
Premium
Description
Created By
Nadavaluru Medhasi, 23/02/2023, 7:10 pm
Last Modified By
Nadavaluru Medhasi, 24/02/2023, 9:42 am
26°C
Mostly sunny
Search
ENG
IN
09:52
24-02-2023


Hands-On Orgs
B - 0011 | Salesforce
wise-goat-jwgi69-dev-ed.trailblaze.lightning.force.com/lightning/r/Billing__c/a012w000017nhagAAA/view
Search...
Code Playground
Leads
Accounts
Contacts
Opportunities
Cases
Customers
Billings
Billing
B - 0011
New Contact
Edit
New Opportunity
Related
Details
Billing Number
B - 0011
Amount Paid
₹10,00,000
Customer Type
Status
Paid
Customer
Wipro
Created By
Nadavaluru Medhasi, 24/02/2023, 9:42 am
Last Modified By
Nadavaluru Medhasi, 24/02/2023, 9:42 am
Show desktop
26°C
Mostly sunny
Search
ENG
IN
09:48
24-02-2023

Exercise 13:

Write a Test Class for CustomerTrigger.
Create the Data for the Customer custom Object
Use Test.startTest() & Test.stopTest().
Use assert statement to validate the output, Test Class must have full code coverage.





Exercise 14:

Write an Apex Trigger, Name = DisqualifyTestLeads.
Define logic to find Leads with ‘Test’ in the Name, Ignore Case.
Validate that the Lead First/Last Name must not be Blank.
IF ‘Test Lead’ found, system.debug(Lead First Name + ‘ ’ + Lead Last Name + ‘Will be disqualified!’);
Add the disqualified Leads to a List<Lead>, use for loop to iterate through the list and update Lead status field to ‘Disqualified’.

Developer Console - Google Chrome

wise-goat-jwgi69-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

Saving: DisqualifyTestLeads.apxt * | DisqualifyTest.apxc *

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

```
trigger DisqualifyTestLeads on Lead (before insert,before update) {
    List<Lead> leadList=new List<Lead>();
    for(Lead ld:Trigger.new){
        if(ld.FirstName==null){
         ld.addError('Enter name');
        }
        else{
            if((ld.FirstName=='test' && ld.FirstName.length()!=0) || (ld.LastName=='test' && ld.LastName.length()!=0)){
                System.debug(ld.FirstName+' '+ld.LastName+' '+'Will be Disqualified');
                leadList.add(ld);

            }
            for(Lead led:leadList){
                led.Status='Disqualified';
            }
        }
    }
}
```

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

Name	Line	Problem

26°C Partly cloudy | Search | ENG IN | 19:42 25-02-2023

Hands-On Orgs | Wipro test | Salesforce

wise-goat-jwgi69-dev-ed.trailblaze.lightning.force.com/lightning/r/Lead/00Q2w00000STYTYEA5/view

Search...

Code Playground | Leads | Accounts | Contacts | Opportunities | Cases | Customers | Billings

Lead
Mr. Wipro test

+ Follow | Convert | Edit | New Case

Title | Company: wipro | Phone (2) | Email

Open - Not Contacted | Working - Contacted | Closed - Not Converted | Converted | Mark Status as Complete

Activity | **Details** | Chatter | News

Lead Owner: Nadavaluru Medhasi	Phone
Name: Mr. Wipro test	Mobile
Company: wipro	Fax
Title	Email
Lead Source	Website
Industry	Lead Status: Disqualified
Annual Revenue	Rating

Related

We found no potential duplicates of this Lead.

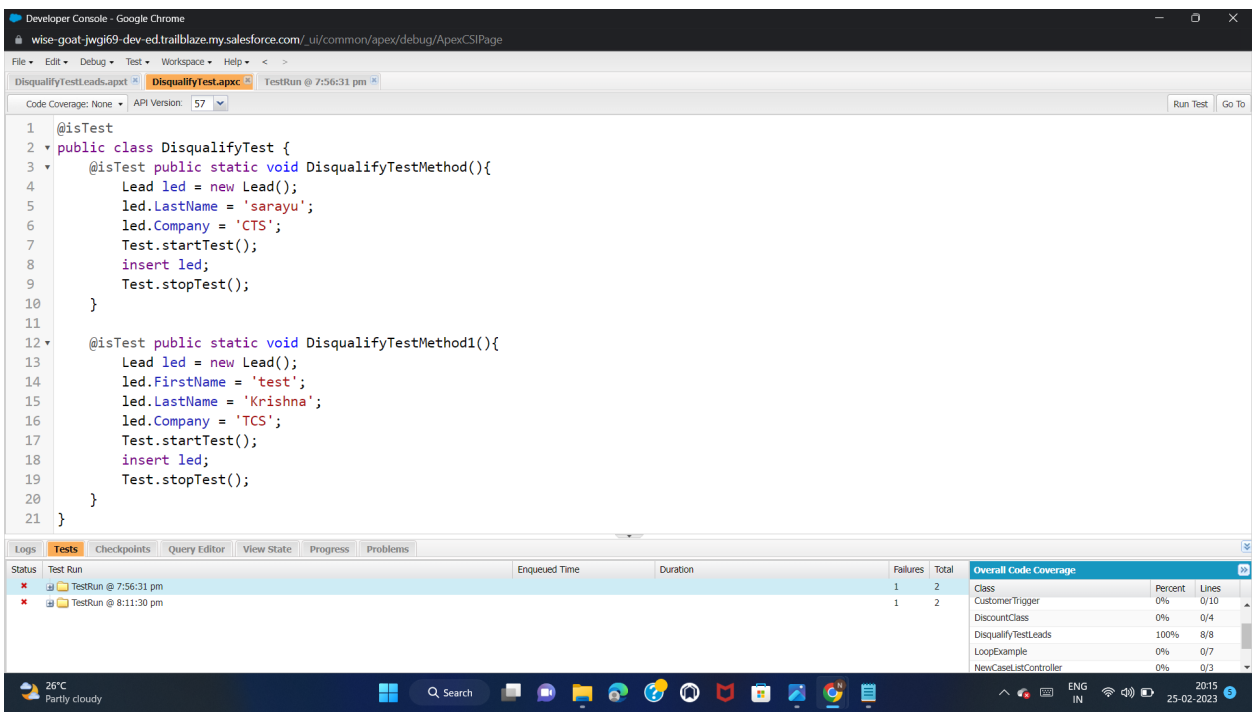
No duplicate rules are activated. Activate duplicate rules to identify potential duplicate records.

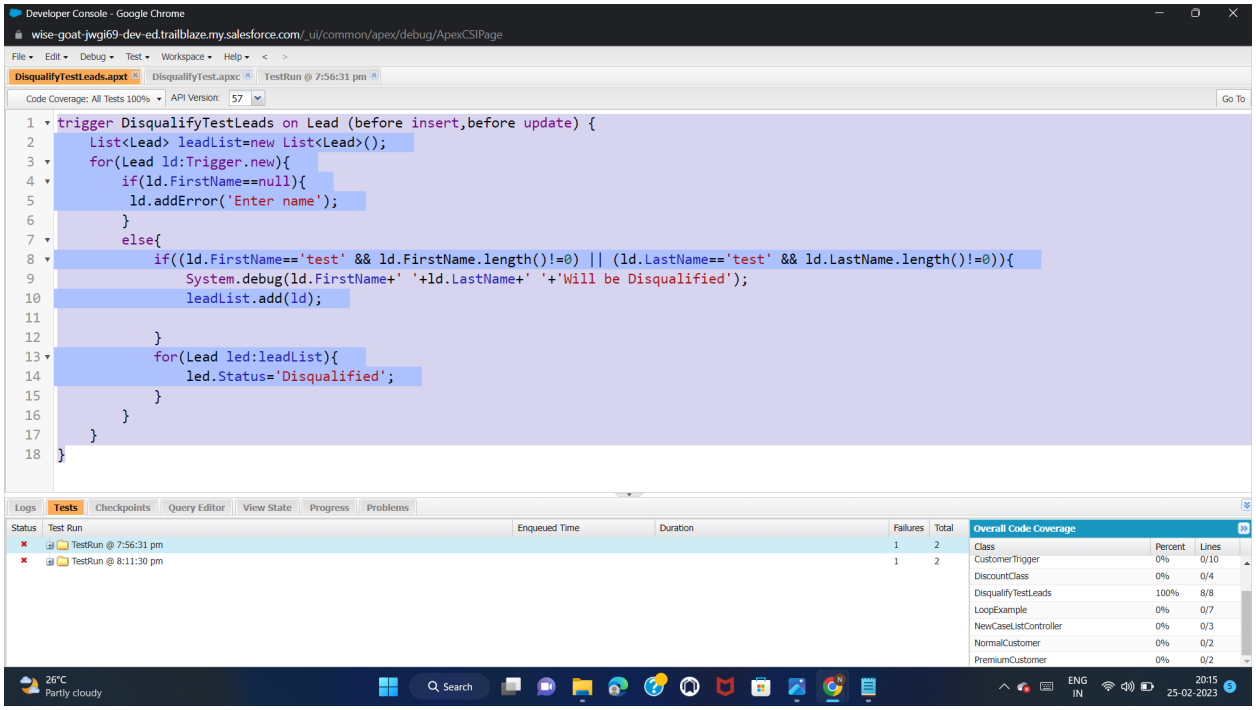
Campaign History (0)

33°C Sunny | Search | ENG IN | 15:04 24-02-2023

Exercise 15:

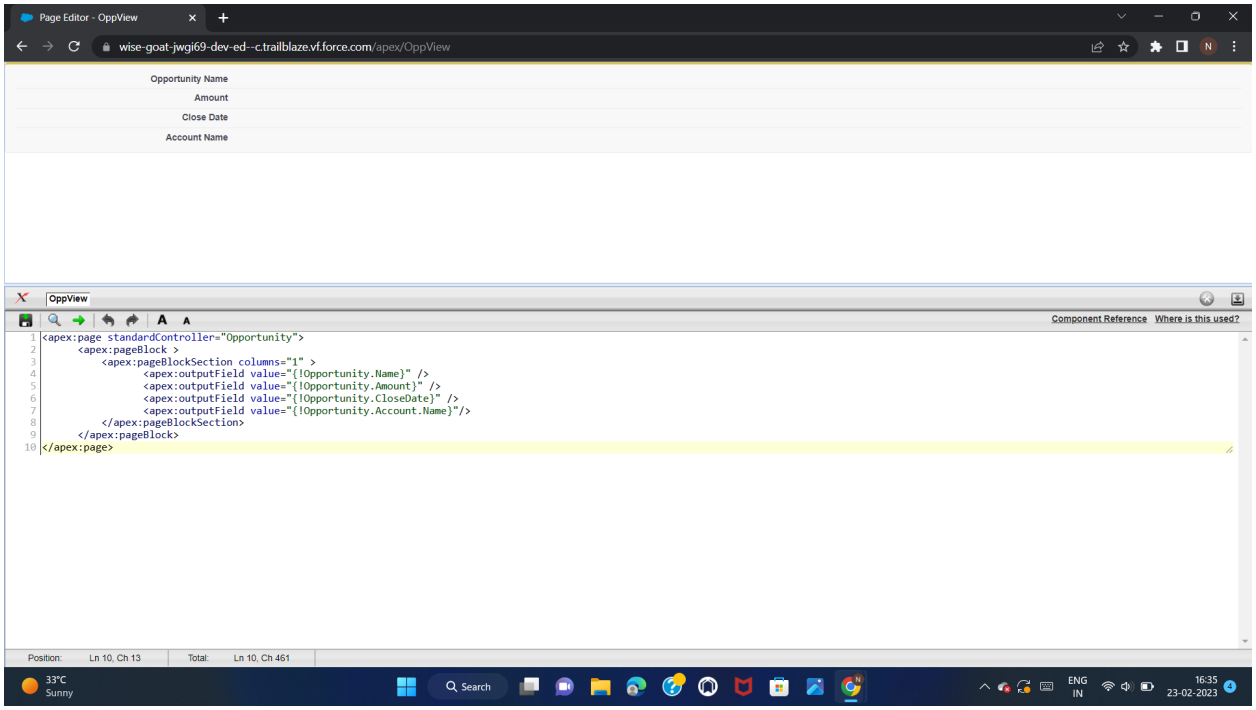
Write a Test Class for DisqualifyTestLeads.
Create the Data for the Lead Object
Use Test.startTest() & Test.stopTest().
Use assert statement to validate the output, Test Class must have full code Coverage.





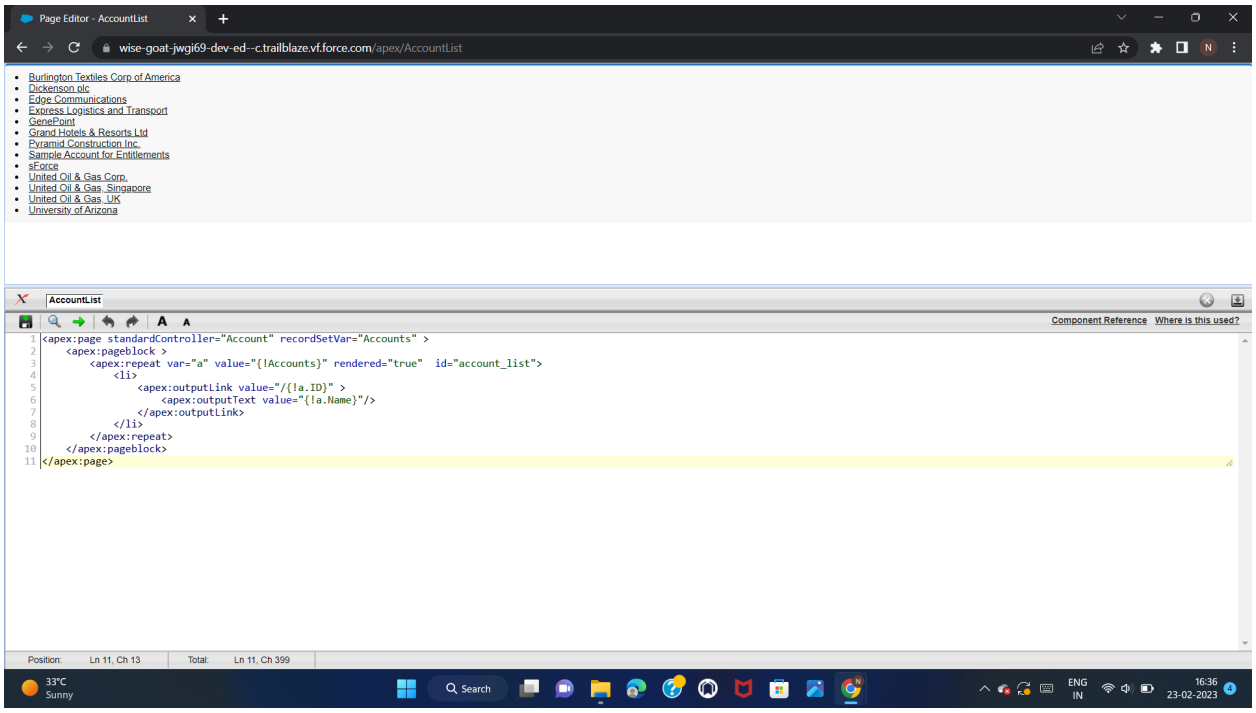
Exercise 16:

Create a Visualforce page which displays Opportunity fields as output fields.
The page must be named OppView, It must reference the Opportunity standard controller.
Define 4 apex:outputField components:
Opportunity Name, Amount, Close Date, Opportunity Account Name.



Exercise 17:

Create a Visualforce page which shows a list of Accounts linked to their record pages.
The page must be named AccountList
It must reference the Account standard controller.
It must have a recordSetVar attribute equal to accounts
It must have a Visualforce apex:repeat component, with the following:
Use the var attribute set to “a”, Use the <li> HTML list tag
Use the apex:outputLink component to link to the respective record detail page



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.

The custom controller Apex class must be named NewCaseListController and include the following:

A publicly scoped method named getNewCases | Use the return type of List<Case>
Return a list of case records that includes the ID and CaseNumber fields.
Filter the results returned to only have a status of New.
The NewCaseList Visualforce page must use an apex:repeat component, which is: Bound to newCases, Refers to the var attribute as case, bind an apex:outputLink component to the ID of the case.

Page Editor - AccountList | Page Editor - NewCaseList

wise-goat-jwgi69-dev-ed--c.trailblaze.vf.force.com/apex/NewCaseList

New Cases List

Case Number
00001002
00001016
00001024

NewCaseList | NewCaseListController

Component Reference | Where is this used?

```
<apex:page controller="NewCaseListController" >
    <apex:pageBlock title="New Cases List">
        <apex:pageBlockTable value="{!newCases}" var="case">
            <apex:column headerValue="Case Number">
                <apex:outputLink value="/{!case.Id}">{!case.CaseNumber}</apex:outputLink>
            </apex:column>
        </apex:pageBlockTable>
    </apex:pageBlock>
</apex:page>
```

Position: Ln 9, Ch 13 | Total: Ln 9, Ch 379

32°C Sunny | Search | ENG IN | 17:02 23-02-2023

References

1. Manage sales - Salesforce IN

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

3. Understand the Salesforce Architecture Unit | Salesforce Trailhead