A CRM Application for

Public Transport Management System

Team ID: LTVIP2025TMID30581

|  |  |
| --- | --- |
| **S.NO** | **NAME OF THE STUDENT** |
| **1.** | Sravanam Padma Priya |
| **2.** | Ravikanth Yalla |
| **3.** | Pranav Sai Gudimella |
| **4.** | Narkidimilli M S S Praharshitha |

## The Public Transport(RTC - Regional Transport corporation) Management System is a comprehensive Salesforce application designed to streamline and manage various operational aspects of the Public Transport.  This system will enable the Transport department to efficiently maintain details of employees, their roles, bus stations, buses, ticket fares, daily bus trips, passenger counts, and the total ticket fare amount. By leveraging Salesforce's robust platform, the Transport Department can improve operational efficiency, data accuracy, and reporting capabilities.

## Objectives:

1. Real Time Salesforce Project
2. Data Modelling
3. Creating an Application
4. User Interface Customization
5. Object & Relationship in Salesforce
6. Formula fields and Validation rules.
7. Field Dependencies
8. Record Types
9. Cross object formula fields.
10. Conditional formatting.
11. Flows
12. Email alerts and email templates
13. Reports & Dashboards

## Salesforce

### Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don’t know where you should start on your learning journey? If you’ve answered yes to any of these questions, then you’re in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we’ll take you through these features and answer the question, “What is Salesforce, anyway?”.

### What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organised something like this:

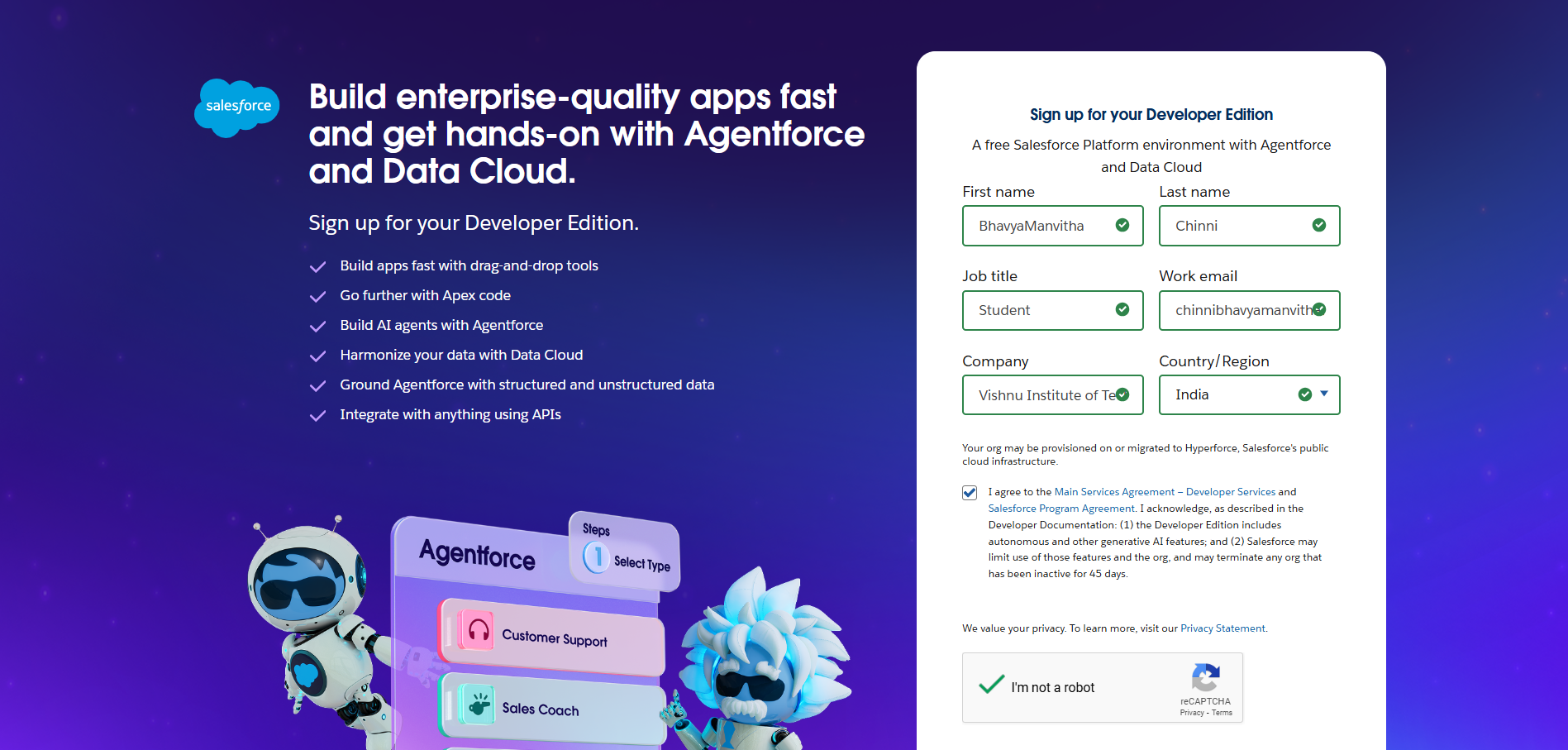
<https://youtu.be/r9EX3lGde5k>

## **Creating Developer Account**

Creating a developer org in salesforce.

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

On the sign up form, enter the following details:

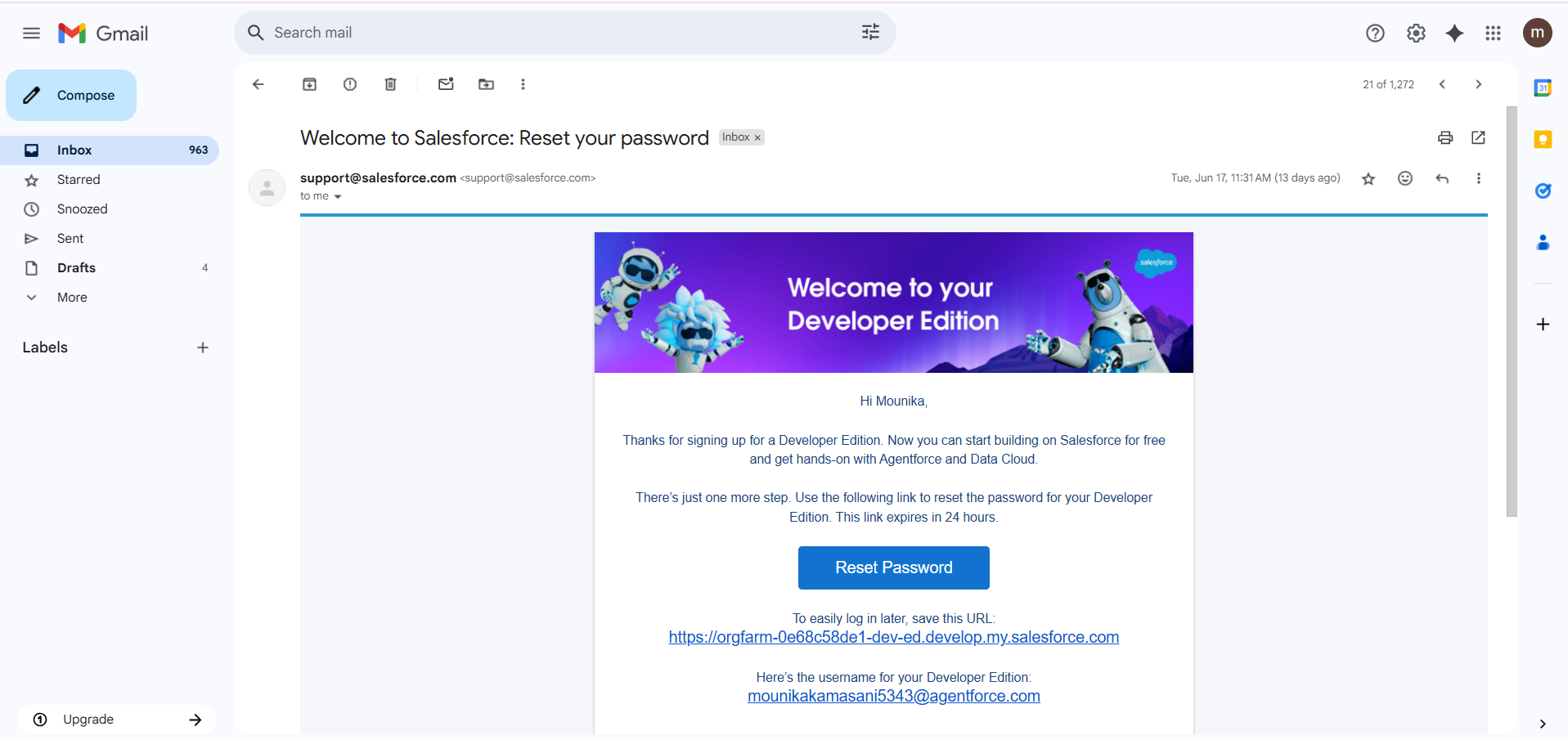
 

1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code
7. Username : should be a combination of your name and company

This need not be an actual email id, you can give anything in the format: username@organization.com

Click on sign me up after filling these.

## **Account Activation**

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.
2.   
   Click on Verify Account
3. Give a password and answer a security question and click on change password.
4. Then you will redirect to your salesforce setup page.

# Object

## **What Is an Object?**

Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects

Salesforce objects are of two types:

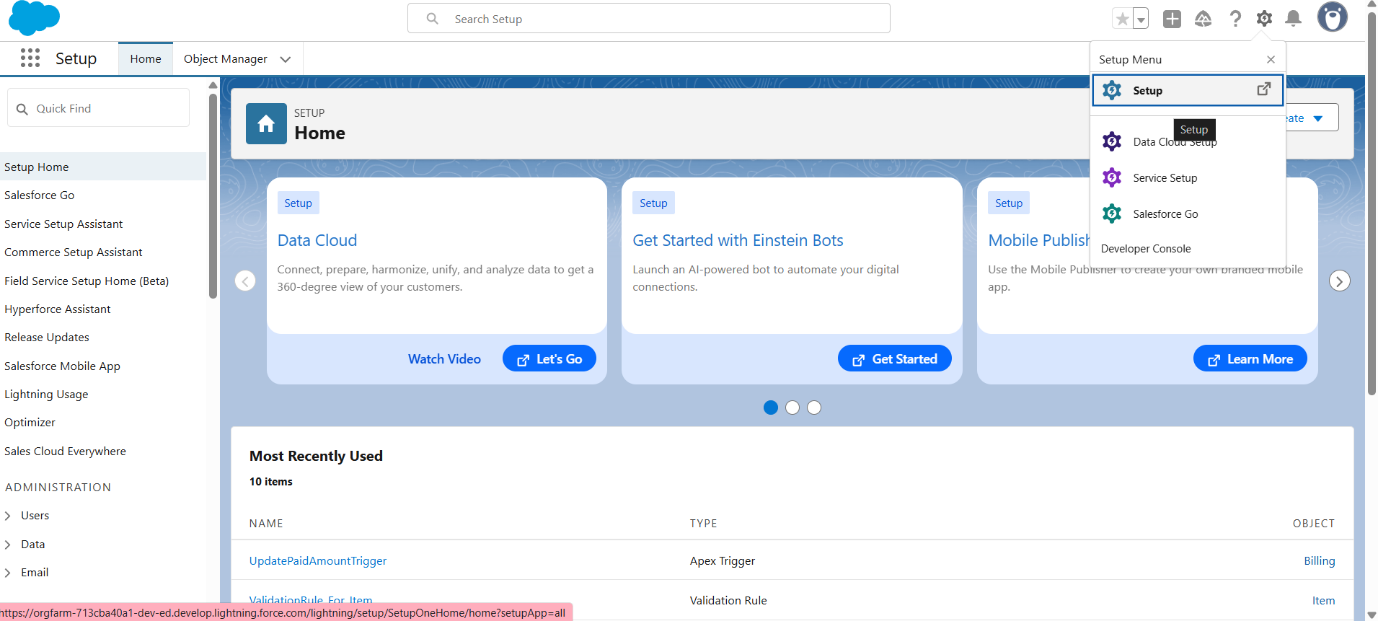
1. Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. Custom Objects: Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

## Use Case:

Creating an object in Salesforce organisation is essential for efficient data management and process automation. By defining custom objects, businesses can structure and store data specific to their needs, enabling streamlined workflows, personalised reporting, and enhanced user experiences. Objects serve as the foundation for organising and leveraging critical information within Salesforce.

To Navigate to Setup page:

Click on gear icon >> click setup.



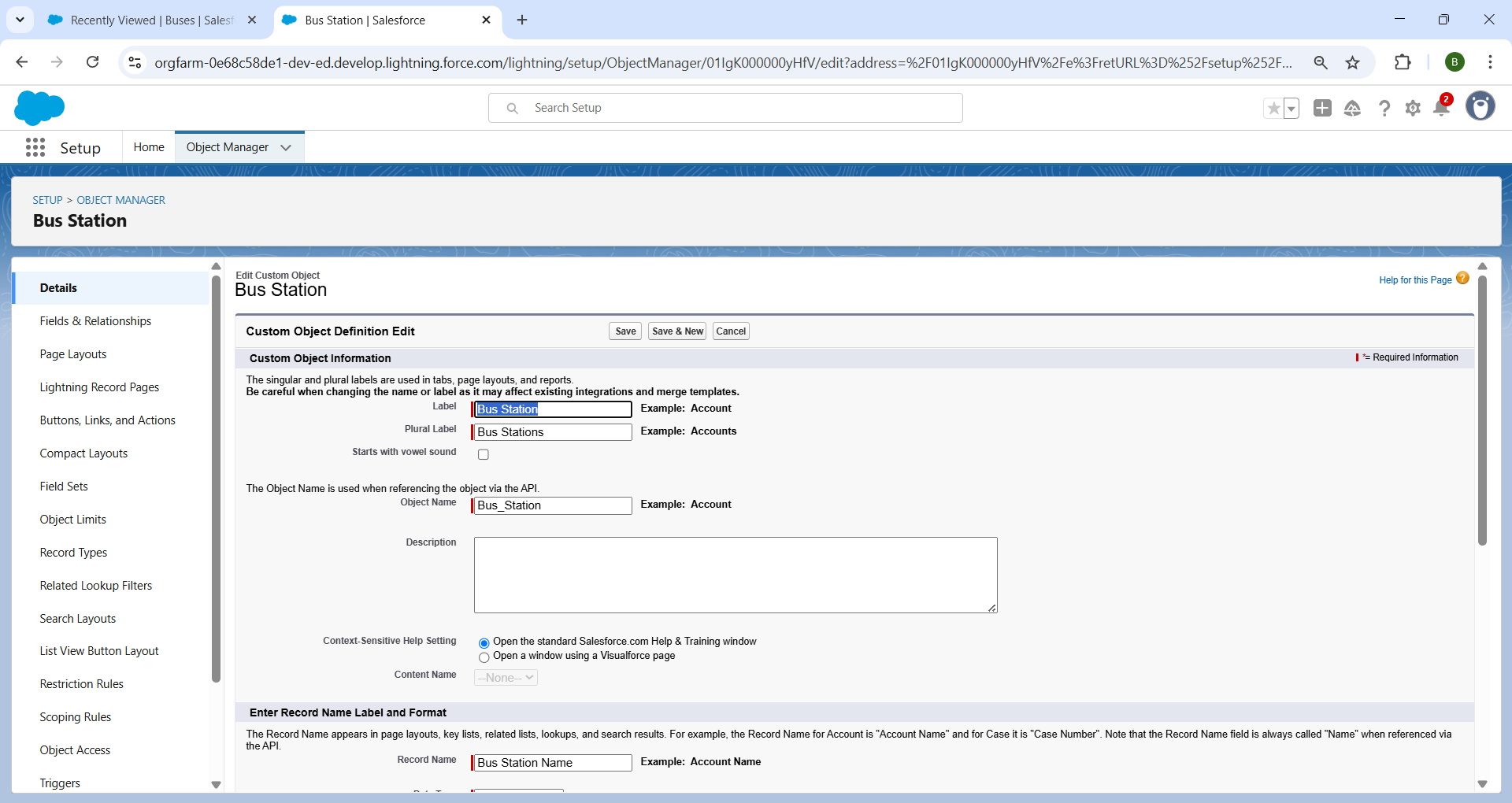
## 1. Creating A Bus Station Object

The purpose of creating a Bus Station custom object is to store and manage information about Bus Stops.

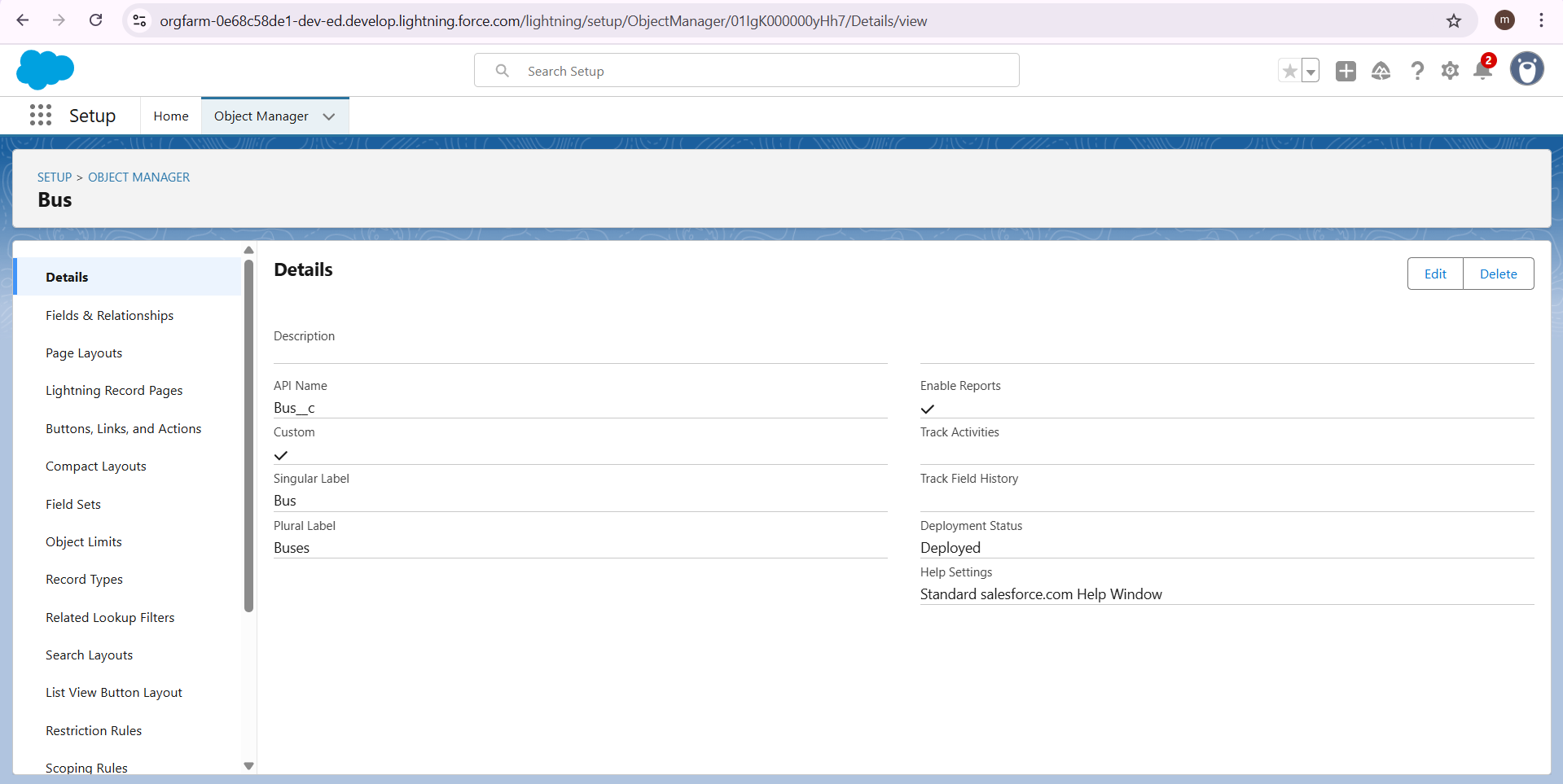
To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object.
4. Enter the label name as Bus Station
5. Enter Plural label name as Bus Stations
6. Enter Record Name as Bus Station Name
7. Select Data Type as Text.
8. Select Allow reports.
9. Select Allow search.
10. Click on Save and New

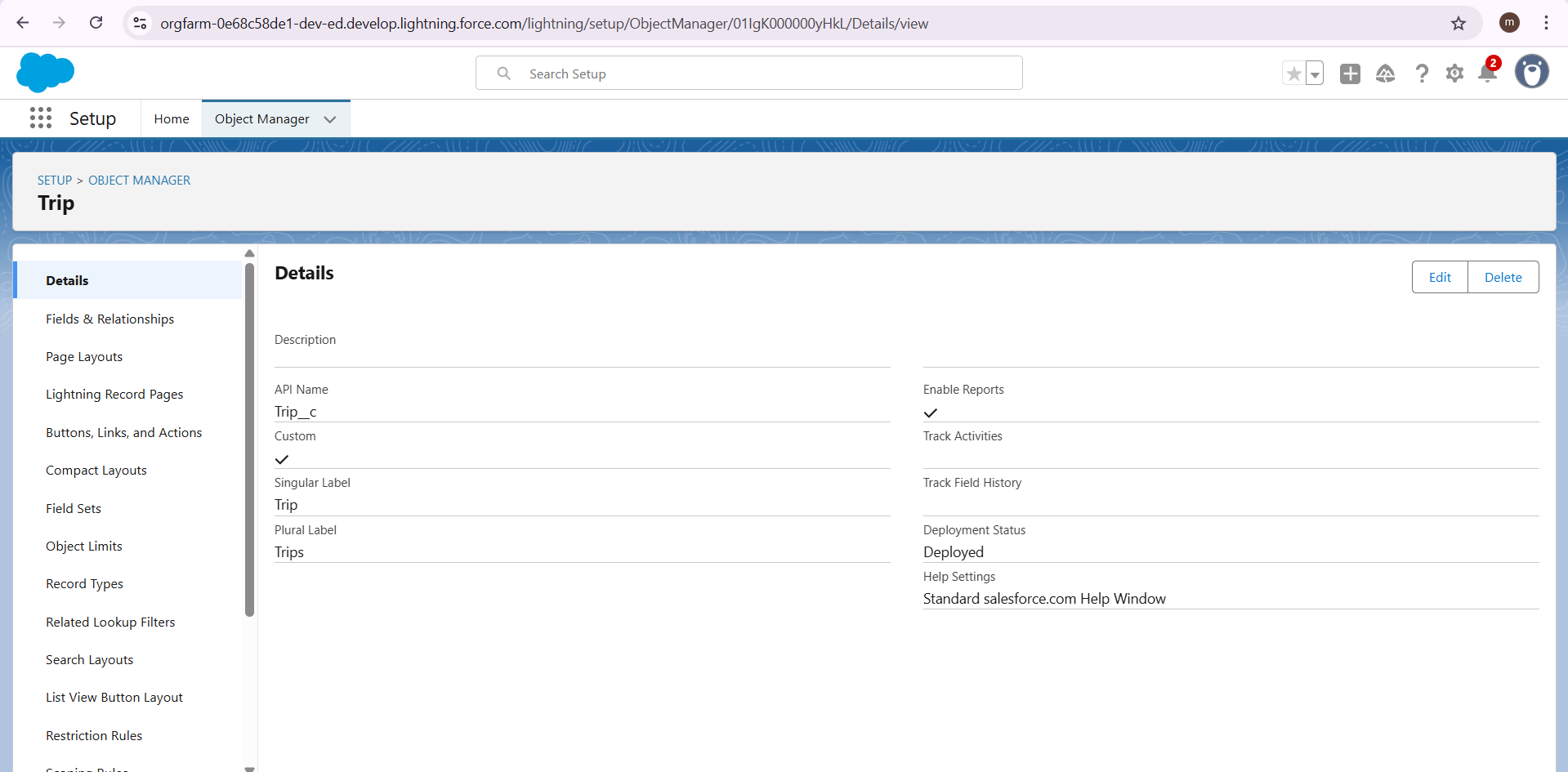
Bus Station Object



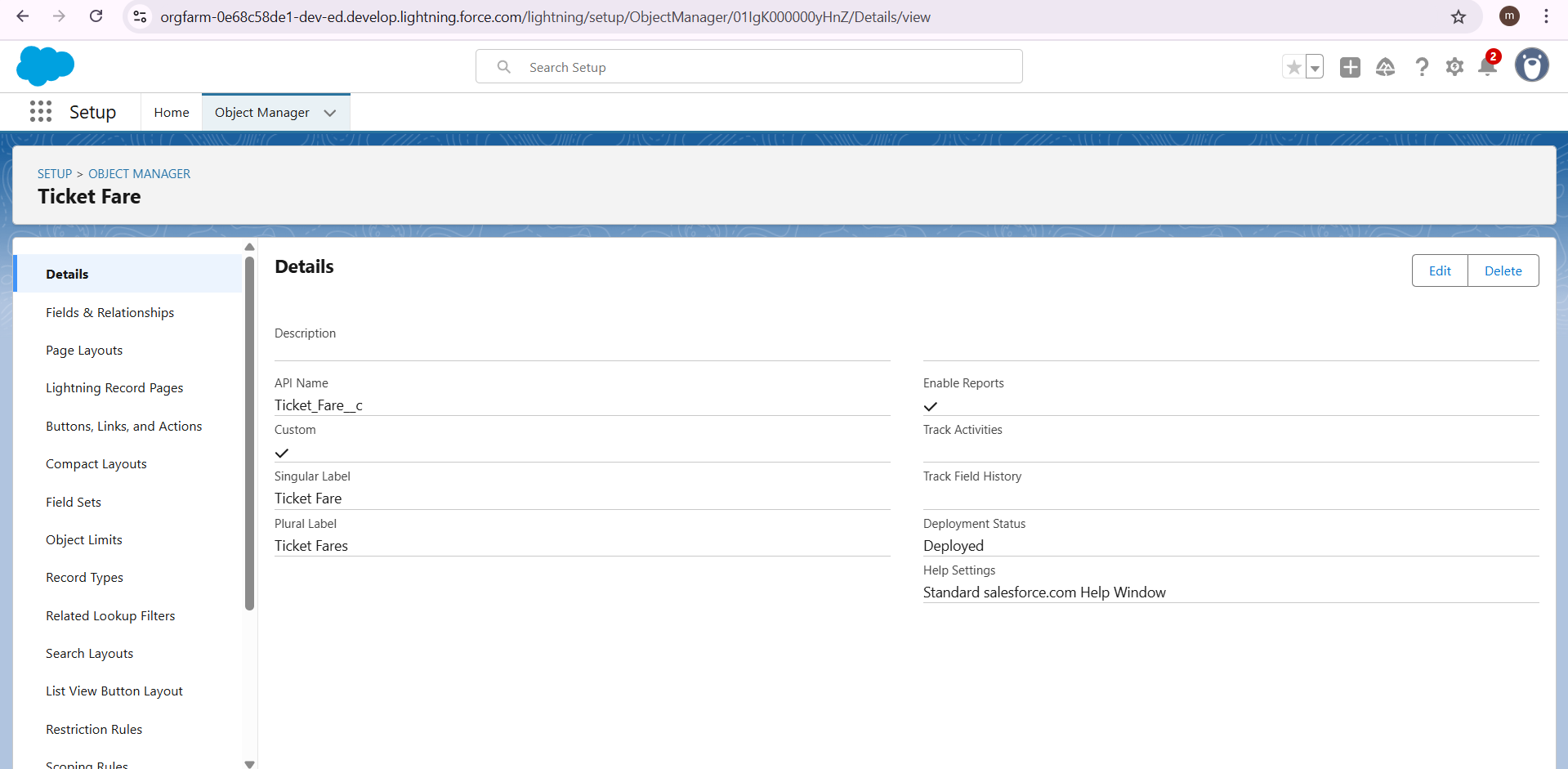
Bus



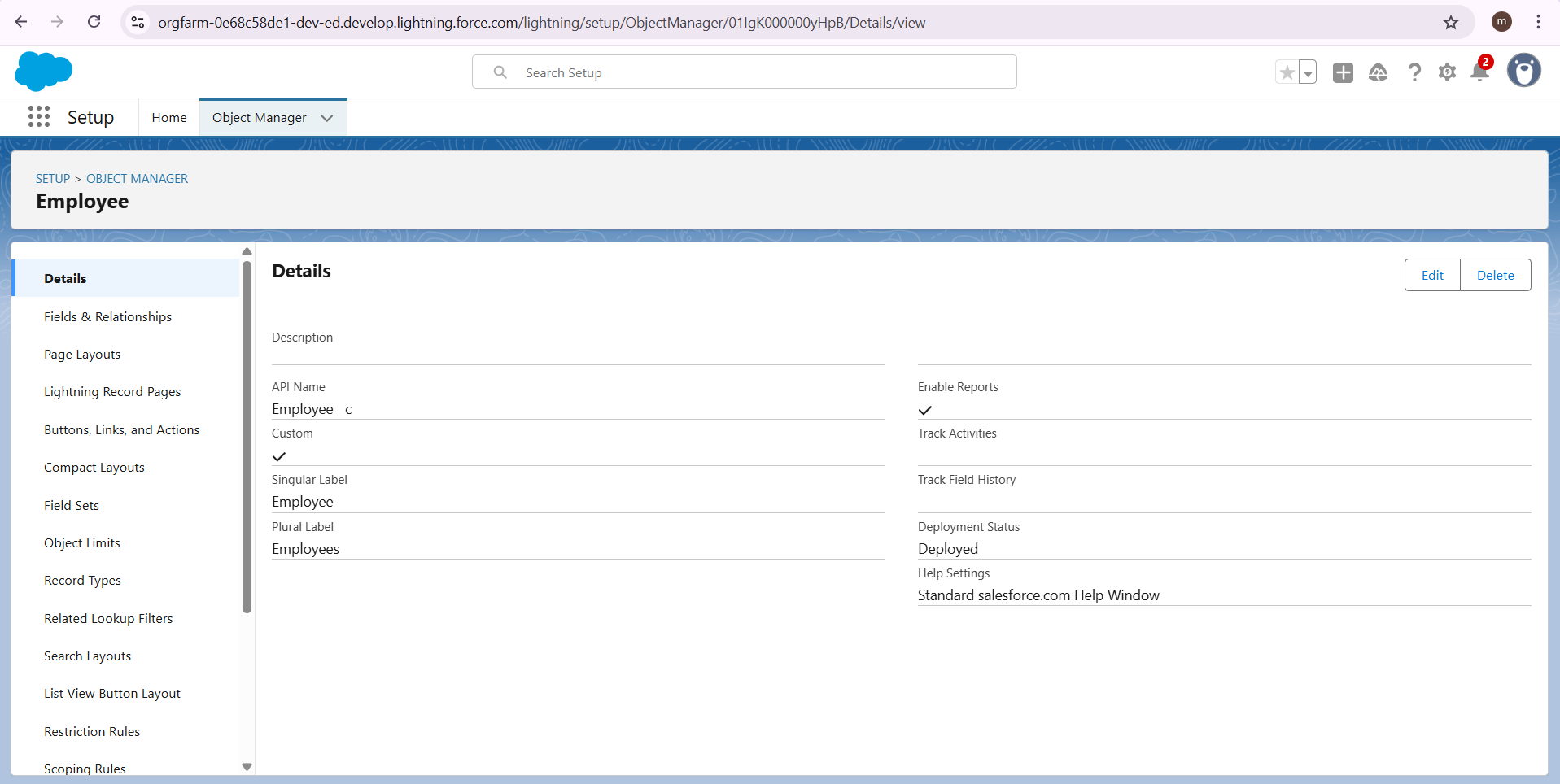
Trip



Ticket Fare



Employee



# Tabs

**What is Tab:**A tab is like a user interface that is used to build records for objects and to view the records in the objects.

**Types of Tabs:**

1. Custom Tabs

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

1. Web Tabs

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

1. Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

1. Lightning Component Tabs

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

1. Lightning Page Tabs

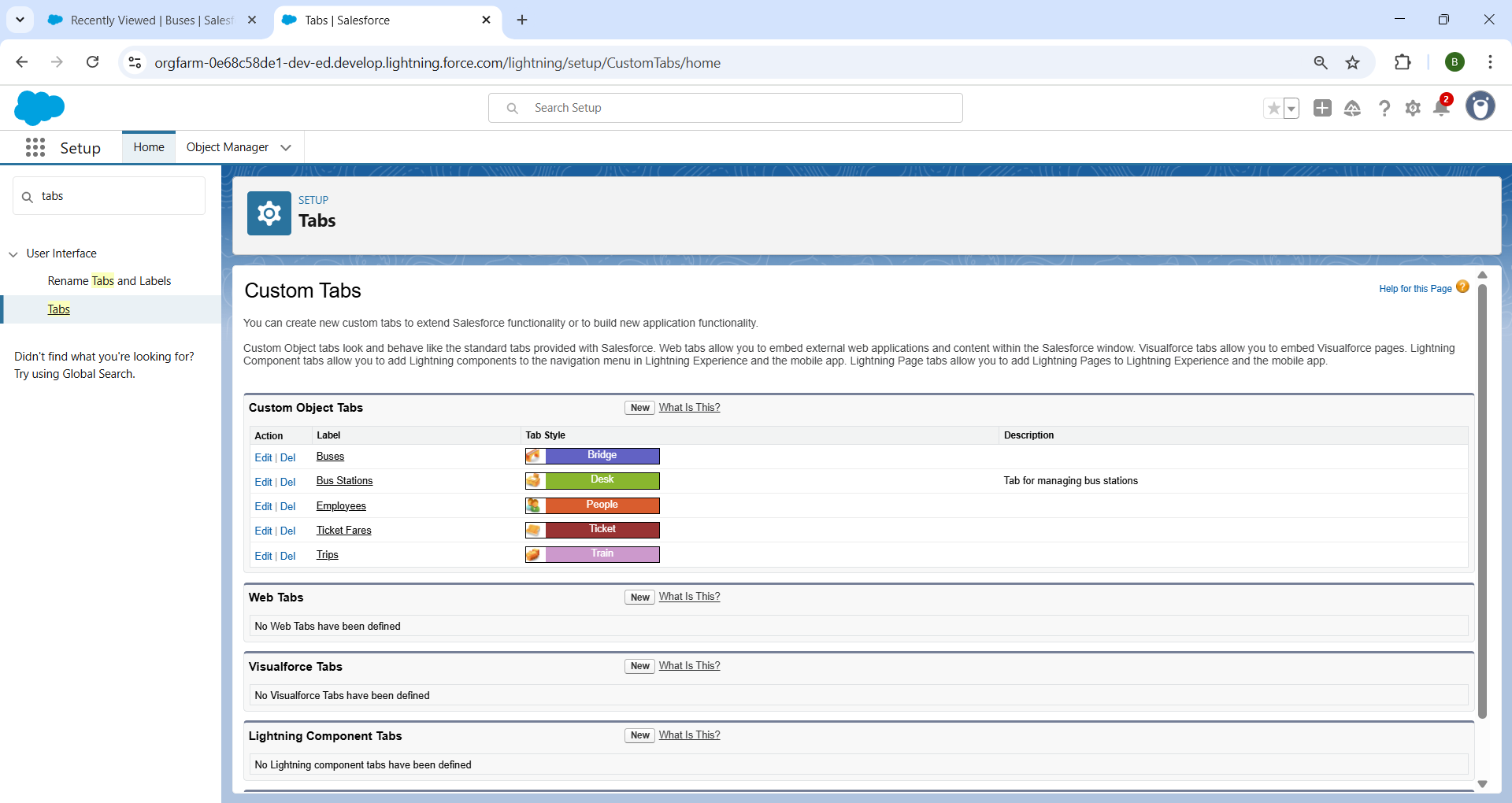
Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

## 1. Creating a Custom Tab

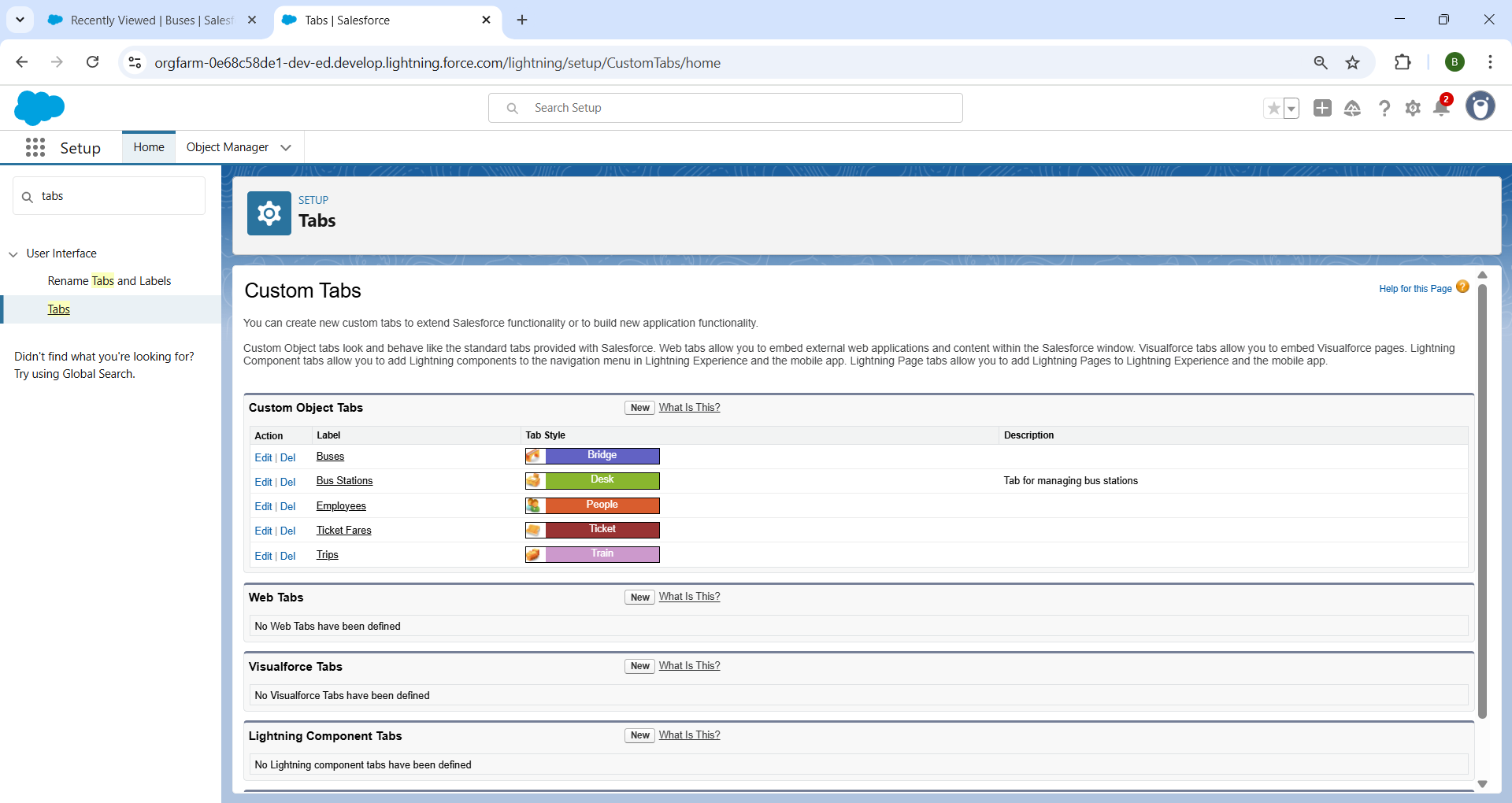
To create a Tab:(Customer)

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



## 2. To create a Tab:(Item)

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



# The Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps gives users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

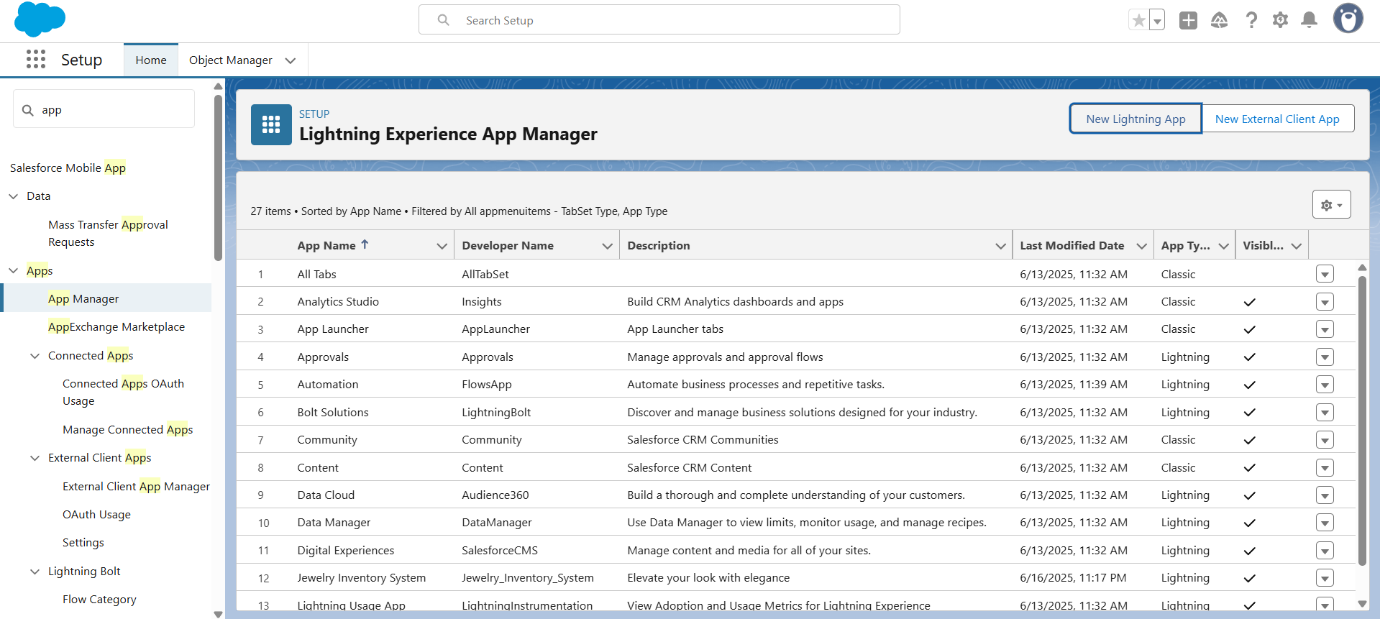
Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

## Use Case:

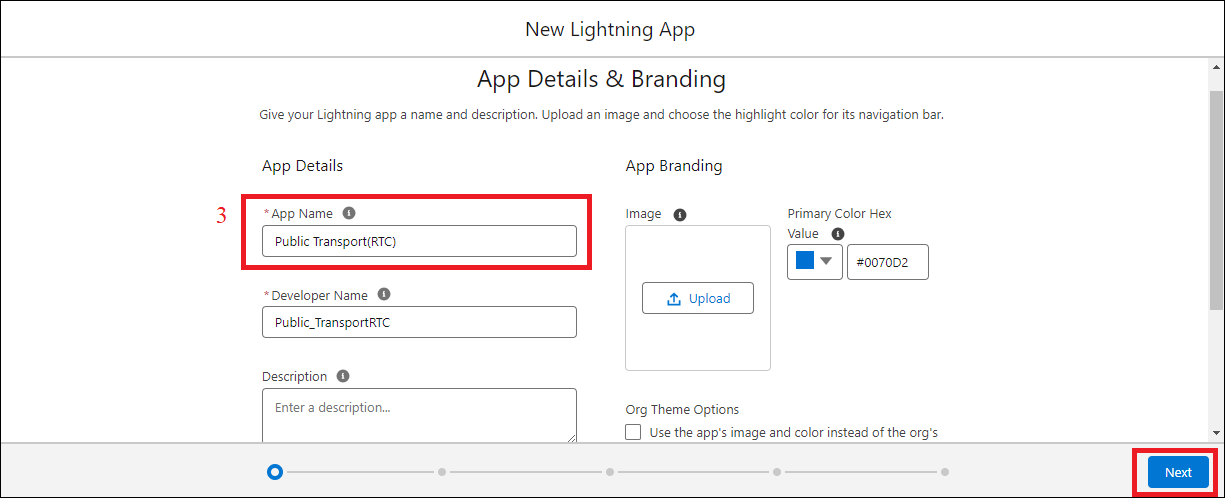
Well done you have reached close to your requirement by creating the objects to store the organization’s data. Making a database for an organization is just not enough to reach out the requirements, the task is how the users at the organization can access the objects you have created for them. As an Admin for the organization it's your duty to make sure every user of the organization is able to access the data modelling structure.

## **Create a Lightning App**

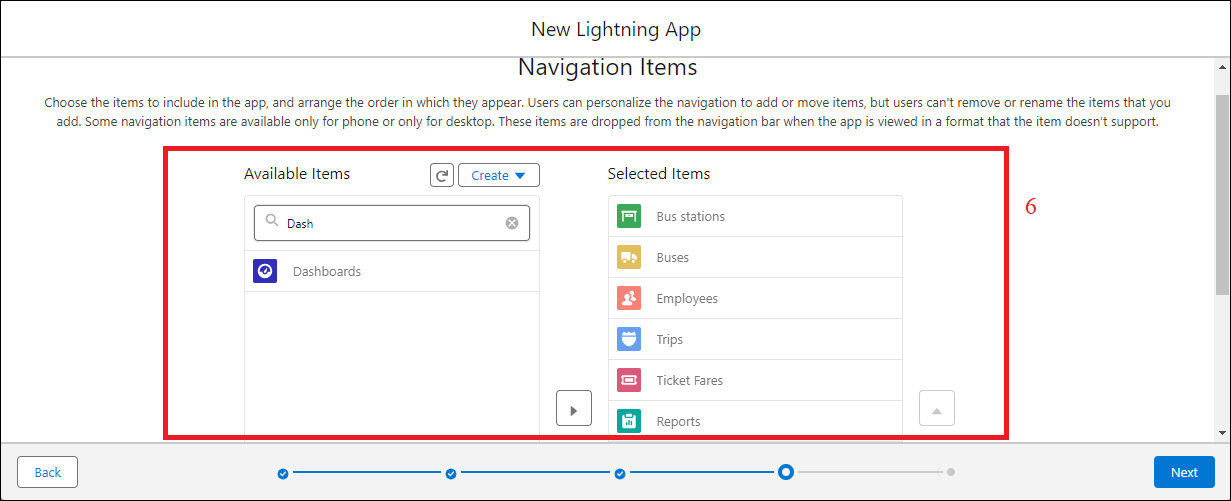
To create a lightning app page:



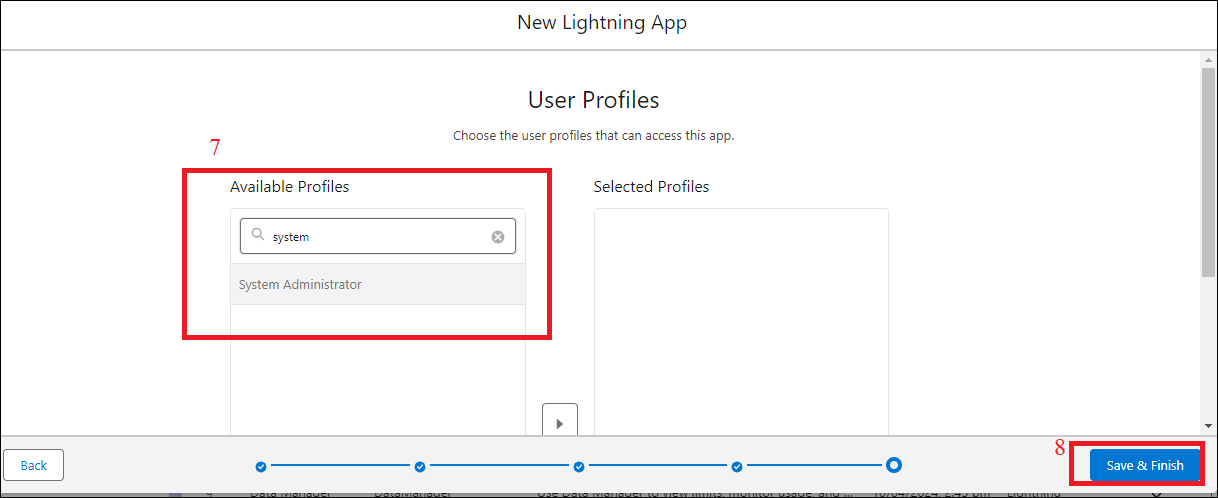
1. From Setup, enter App Manager in the Quick Find and select App Manager.
2. Click New Lightning App.
3. Enter Public Transport as the App Name, then click next



1. Under App Options, leave the default selections and click next.
2. Under Utility Items, leave as is and click Next.
3. From Available Items, select Bus Stations, Buses, Trips, Ticket Fares, Employees, Reports, and Dashboards and move them to Selected Item and Click Next.



1. From Available Profiles, select System Administrator and move it to Selected Profiles.



1. Click Save & Finish.

# Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Types of Fields :

1. Standard Fields
2. Custom Fields

Standard Fields:

As the name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard task. The main point is that you can’t simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salesforce application. They are,

>> Created By

>> Owner

>> Last Modified

>> Field Made During object Creation

Custom Fields:

On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organiser or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.

**Use Case:**

Now it’s time for you to think out of the box for your organisation. You have successfully created the database objects for the organisation but now all eyes turn on you as you have to define what sort of information the objects store which you have created. As a life saver of your organisation you come up with the idea of creating fields to store different types of data.

## 1.Creating a Role Picklist Field in Employee Object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box>> click on the Employee Object.
2. Now click on “Fields & Relationships” .
3. Click on New.
4. Select Data type as “Picklist” and click Next.
5. Enter Field Label as “Role”.
6. In values select “Enter values, with each value separated by a new line" and enter values as shown below.

Administrative Assistant

Cleaner

Conductor

Customer Service Representative

Driver

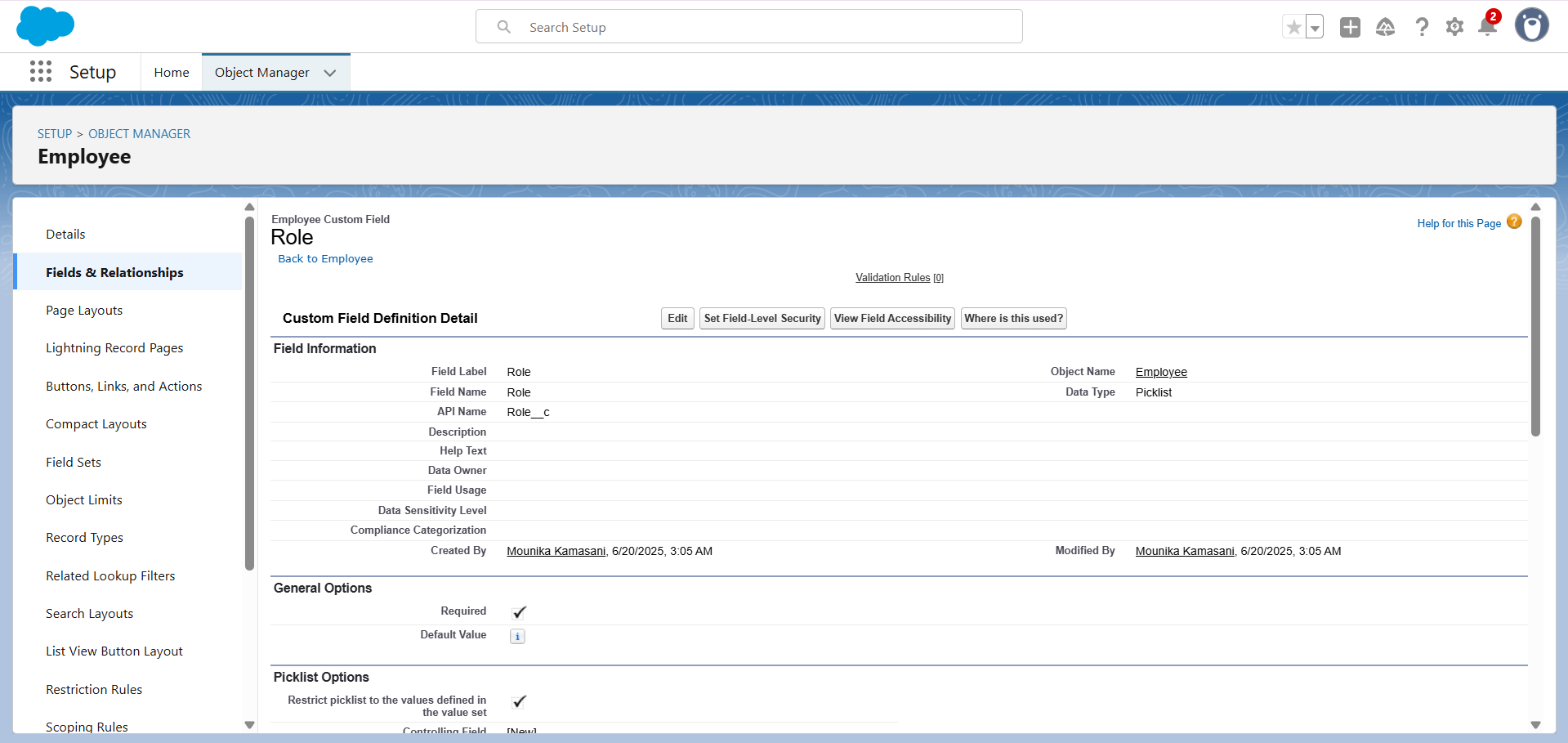
Safety Officer

Station Manager

Supervisor

Ticket Inspector

1. Select Display values alphabetically, not in the order entered .
2. Select Required, Always require a value in this field in order to save a record .
3. Click on Next, Next and Save.



## 2. Creating a Trip Date Field in Trip object

## To create fields in an object:

## Go to setup >> click on Object Manager >> type object name(Trip) in quick find box>> click on the Trip object.

## Now click on “Fields & Relationships”

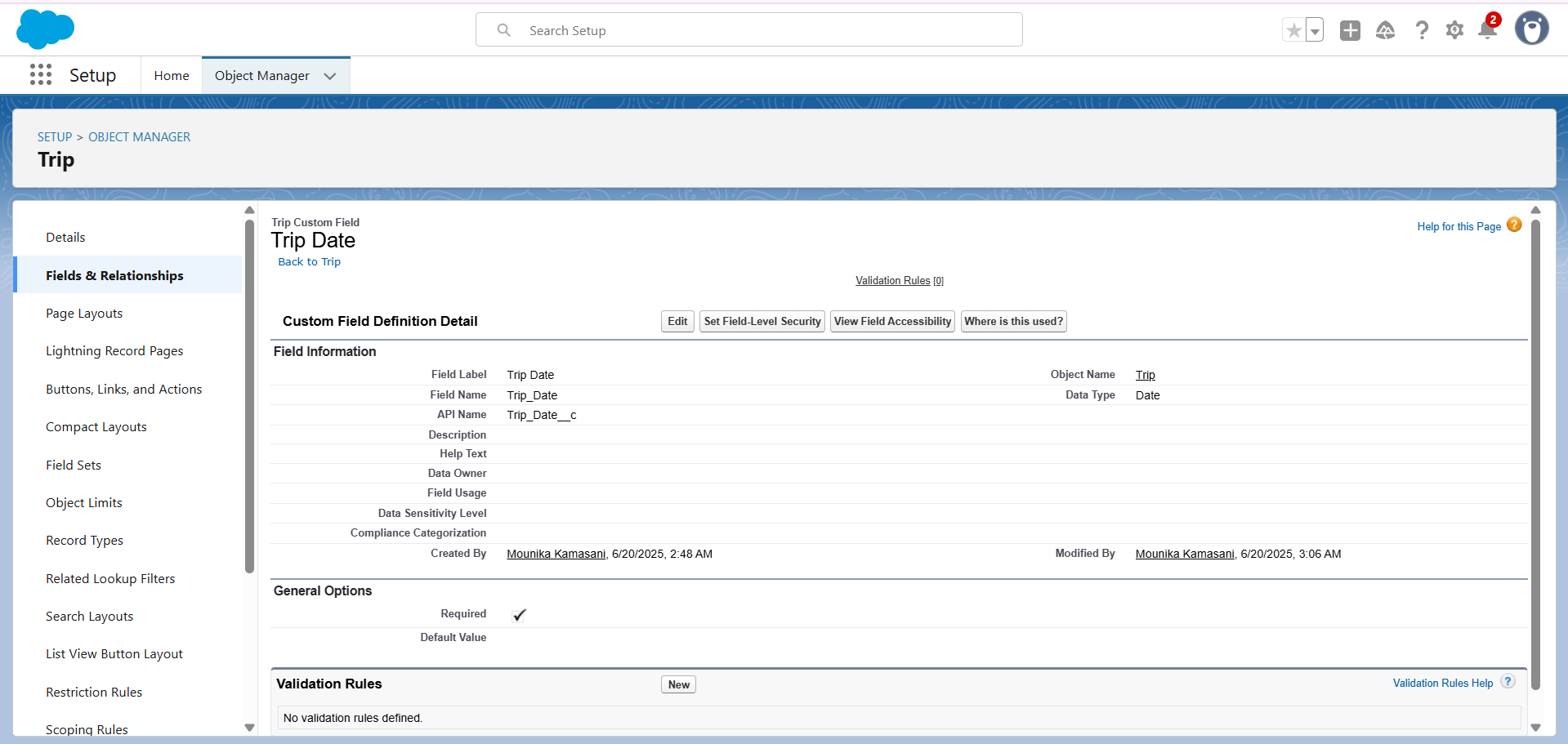
## Click on New.

## Select Data type as “Date” and click Next.

## Enter Field Label as “ Trip Date”.

## Select Required, Always require a value in this field in order to save a record .

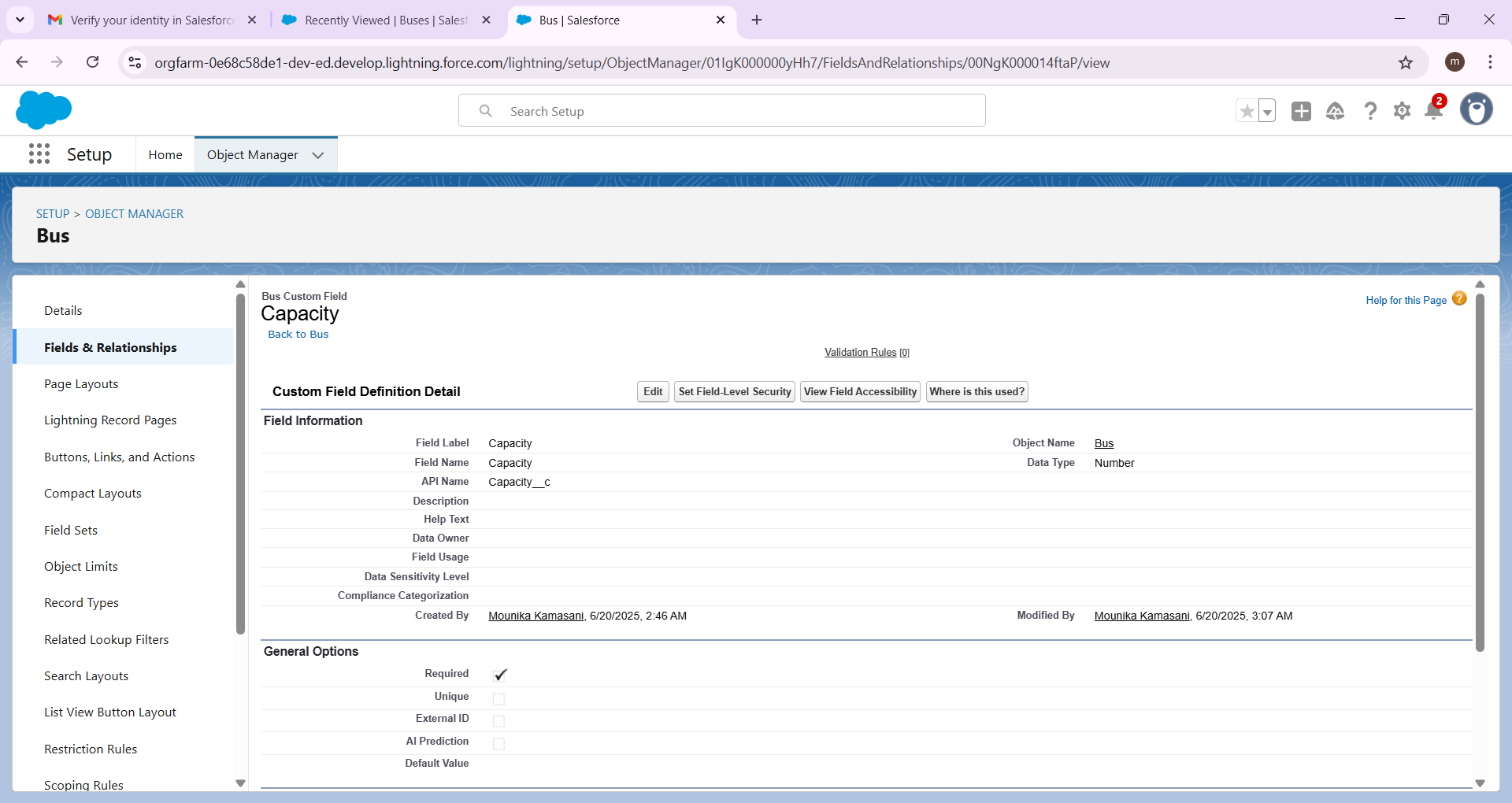
## 7.Click on Next, Next and Save.



## 3.Creating a Number Field in Bus object

 To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Bus) in quick find box >> click on the Bus object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Number” and click Next.
5. Enter Field Label as “ Capacity”.
6. Length - 4, Decimal Places - 0.
7. Select Required, Always require a value in this field in order to save a record .
8. Click on Next, Next and Save.

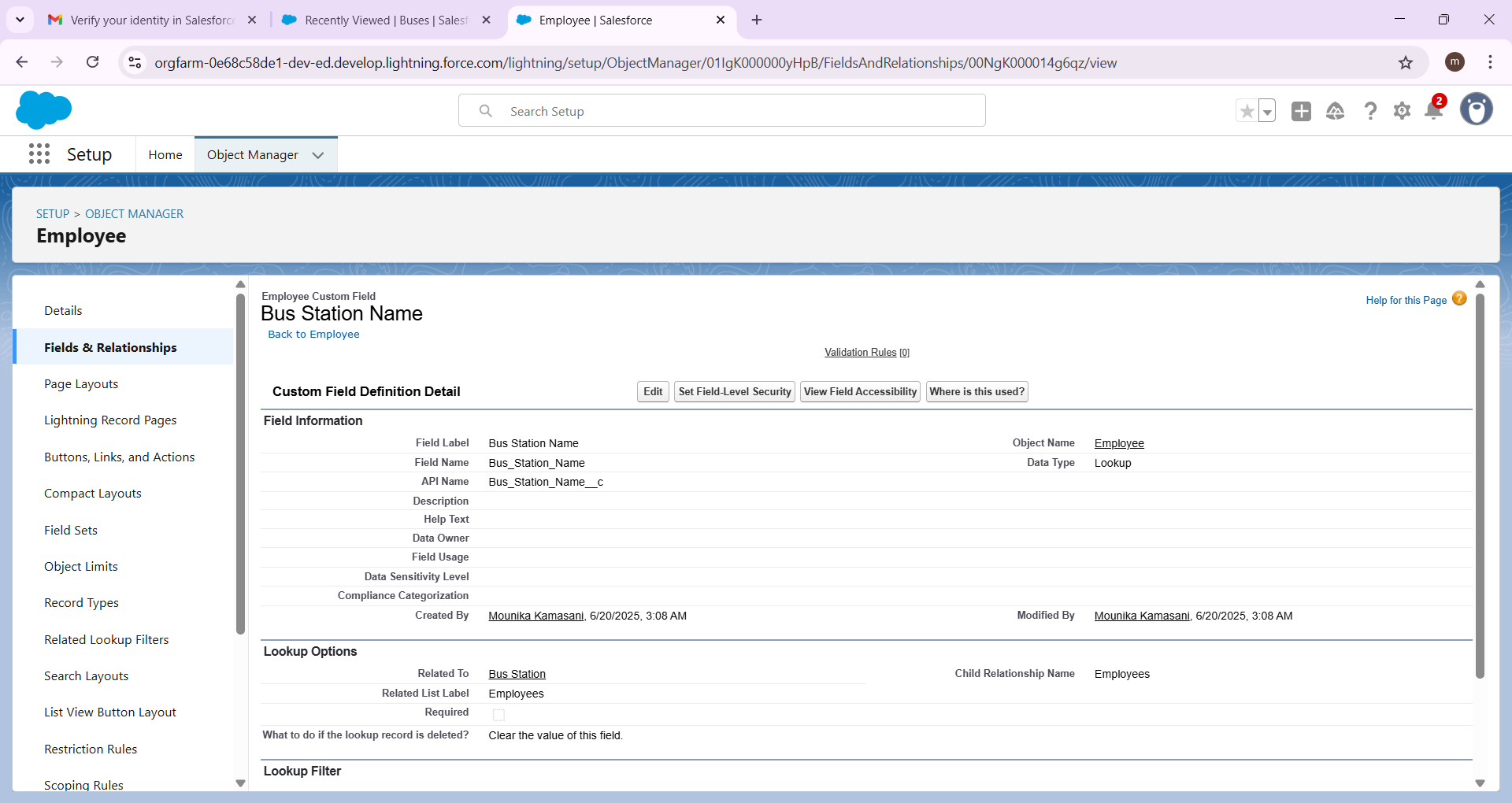


## 4. Creating Lookup Relationship

A Lookup relationship is a type of relationship in Salesforce that connects two objects together based on a field known as the Lookup field. It establishes a relationship between a child object and a parent object, allowing the child object to reference the parent object.

To Create a relationship from Employee to Bus Station .

1. Go to the Setup page >> click on Object manager >> type object name(Employee) in the quick find bar >> click on the Employee object.
2. Click on fields & relationship
3. Click on New.
4. Select “Lookup relationship” as data type and click Next.
5. Select the related object “ Bus Station”.
6. Click on Next
7. Give Field Label as “Bus Station Name” .
8. Click on Next , Next, Next , Save.



## 5. Creating a Checkbox Field in Bus Station object

To create fields in an object:

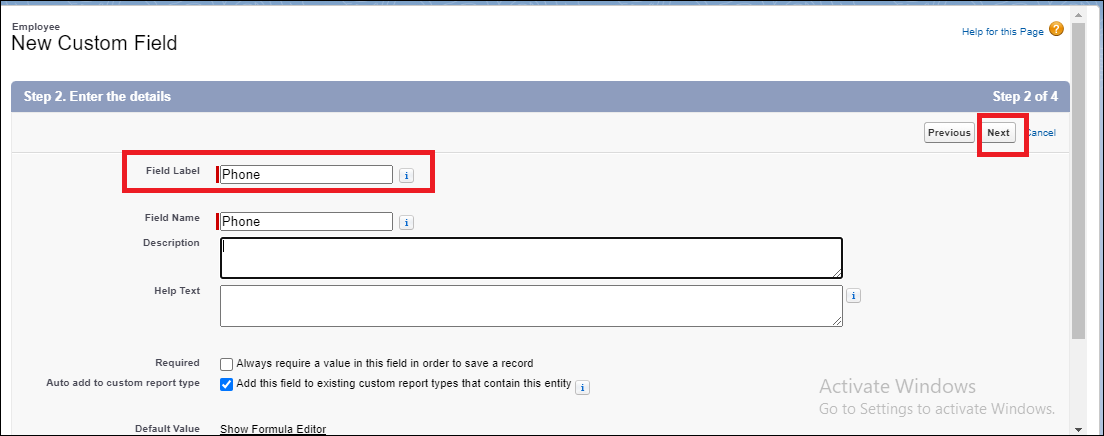
1. Go to setup >> click on Object Manager >> type object name(Bus Station) in quick find box>> click on the Bus Station object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Checkbox” and click Next.
5. Enter Field Label as “ Shelter available”.
6. Select Default value : Unchecked .
7. Click on Next, Next and Save.

## 

## 6. Creating a Phone Field in Employee object

To create fields in an object:

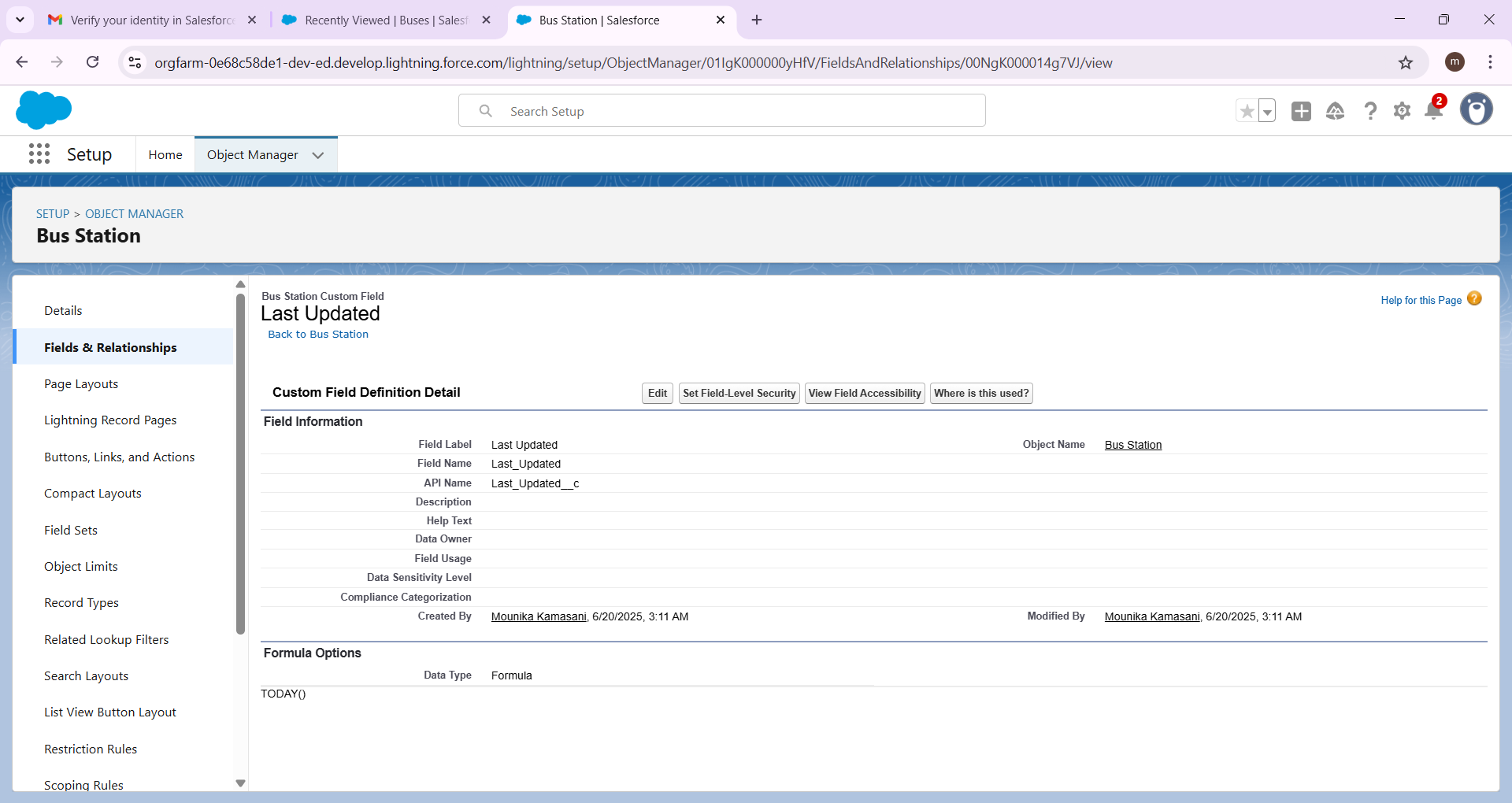
1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box>> click on the Employee object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Phone” and click Next.
5. Enter Field Label as “ Phone No”.
6. Click on Next, Next and Save.



## 7. Creating a Last Updated Formula Field in Bus Station object

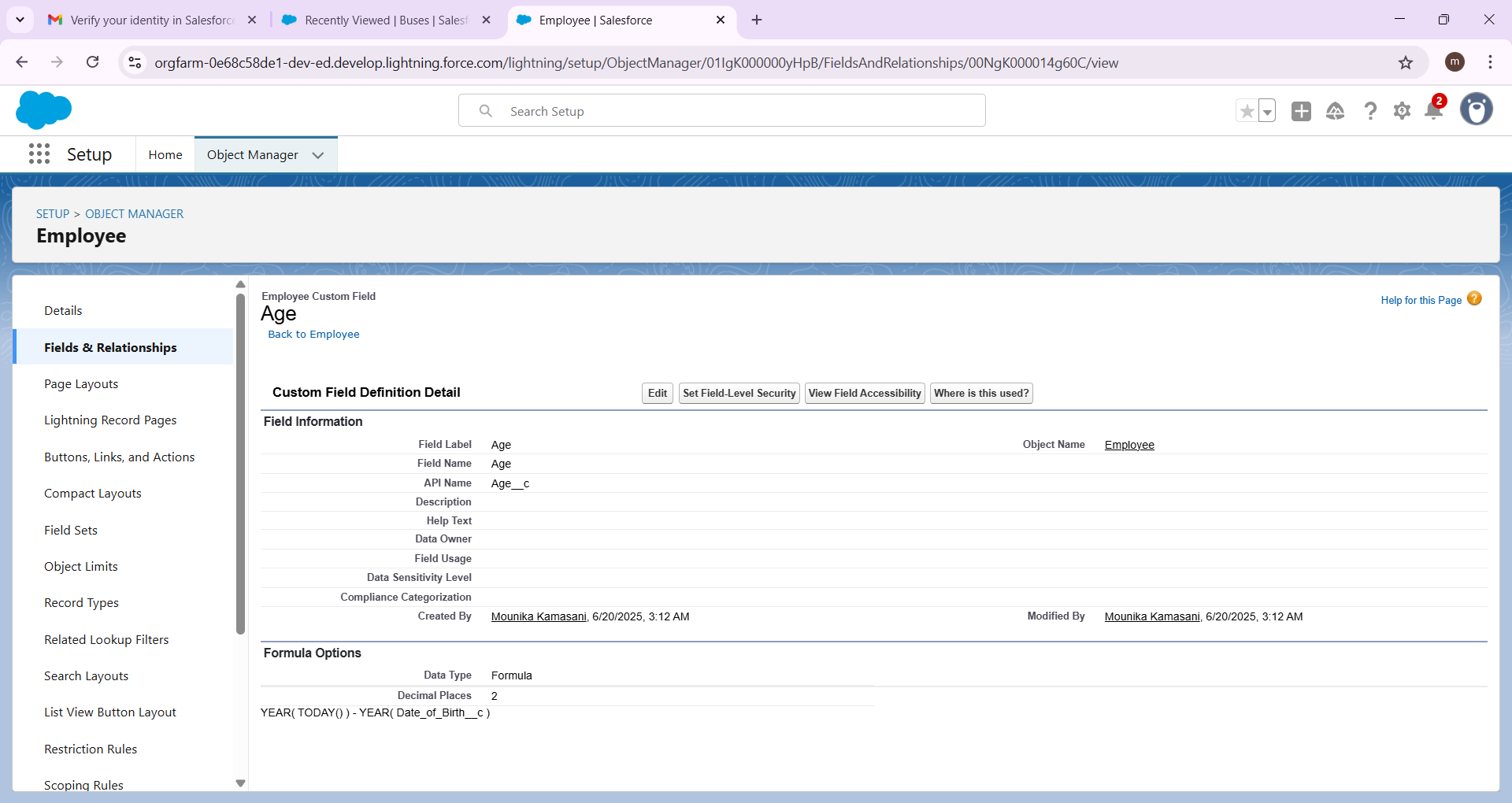
To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Bus Station) in quick find box >> click on the Bus Station object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Last Updated
6. Select formula return type Date, Click Next
7. Create and insert Advance formula:  TODAY()
8. Click Next, Next, then Save.



## 8. Creating a Age Formula Field in Employee object

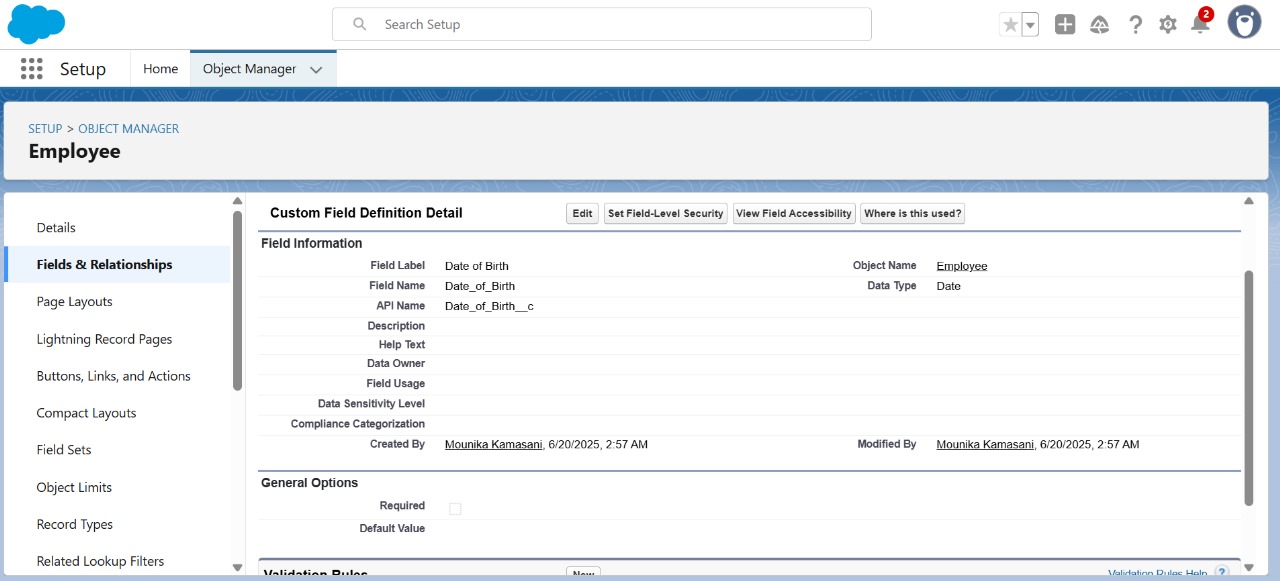
1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box >> click on the Employee object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Age
6. Select formula return type Number, Click Next
7. Create and insert Advance formula:  YEAR( TODAY() )  -   YEAR( Date\_of\_Birth\_\_c )
8. Click Next, Next, then Save.



## 9. Creating a Date of Retirement Formula Field in Employee object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box >> click on the Employee object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Date of Retirement
6. Select formula return type Date, Click Next
7. Create and insert Advance formula:  DATE(YEAR(  Date\_of\_Birth\_\_c  )  + 55,  MONTH(Date\_of\_Birth\_\_c) , DAY(Date\_of\_Birth\_\_c) )
8. Click Next, Next, then Save.



## 10. Creating Experience Formula Field in Employee object

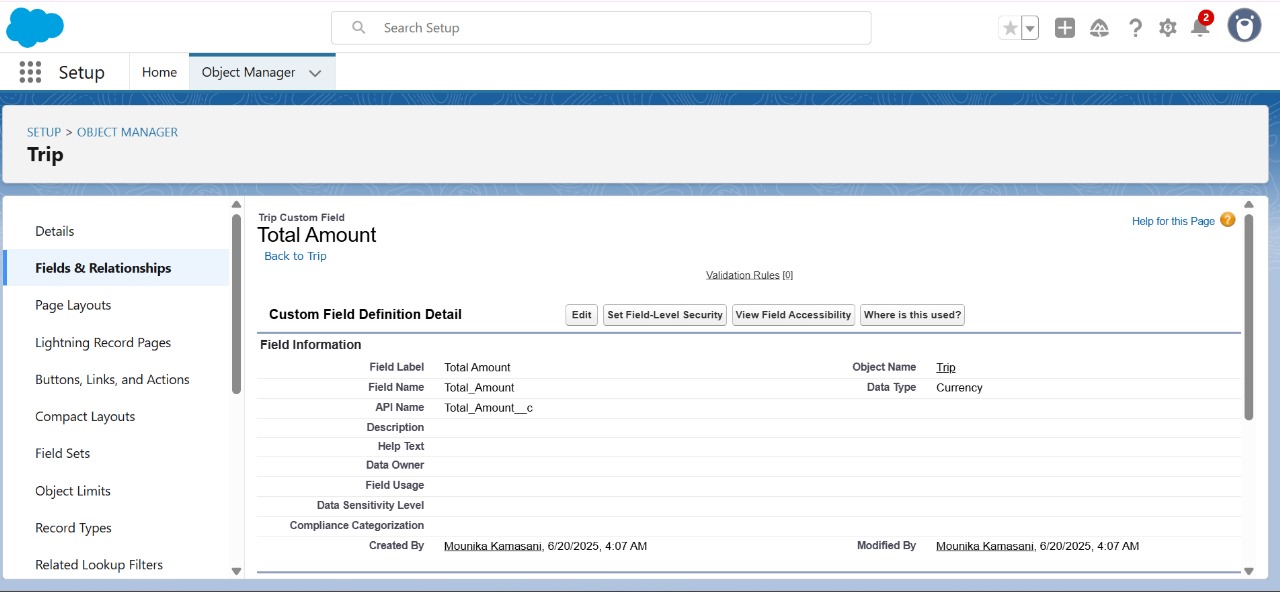
To create fields in an object:

1. Go to setup >> click on Object Manager  type object name(Employee) in quick find box >> click on the Employee object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Experience
6. Select formula return type Number, Click Next
7. Create and insert Advance formula:  YEAR( TODAY() )  -  YEAR( Date\_of\_joining\_\_c )
8. Click Next, Next, then Save.

## 11. Creating a Total Amount Formula Field in Trip object

 To create fields in an object:

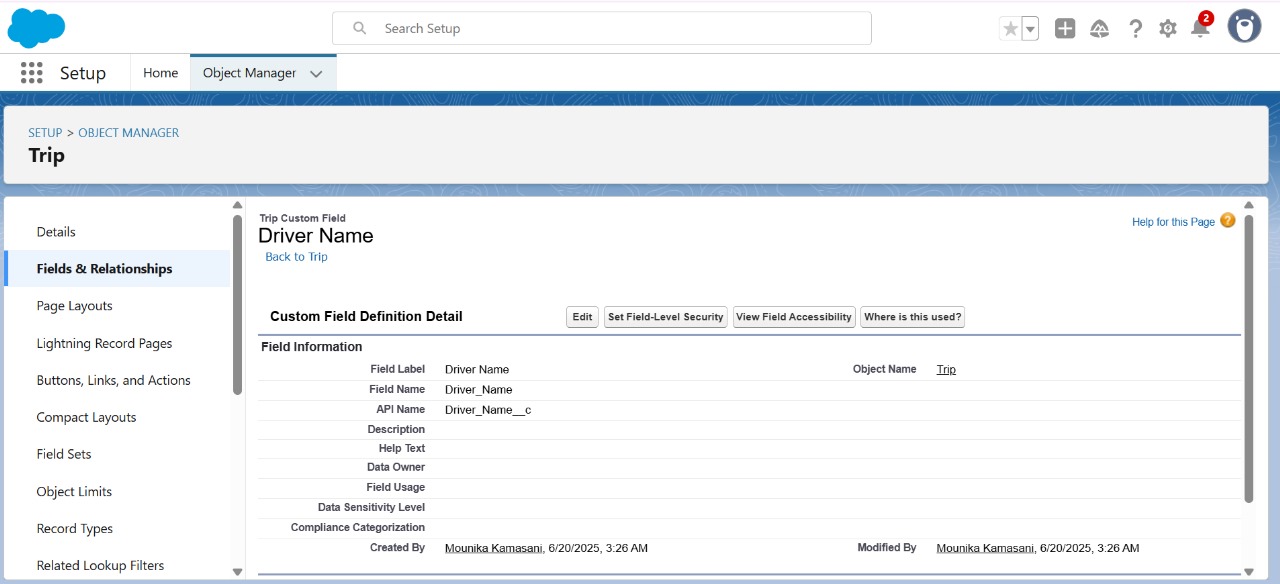
1. Go to setup >> click on Object Manager >> type object name(Trip) in quick find box >> click on the Trip object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Total Amount.
6. Select formula return type Currency, Click Next.
7. Create and insert Advance formula: Passenger\_Count\_\_c  \*   Ticket\_Fare\_\_c
8. Click Next, Next, then Save.



## 12. Creating a Driver Name Formula Field in Trip object

 To create fields in an object:

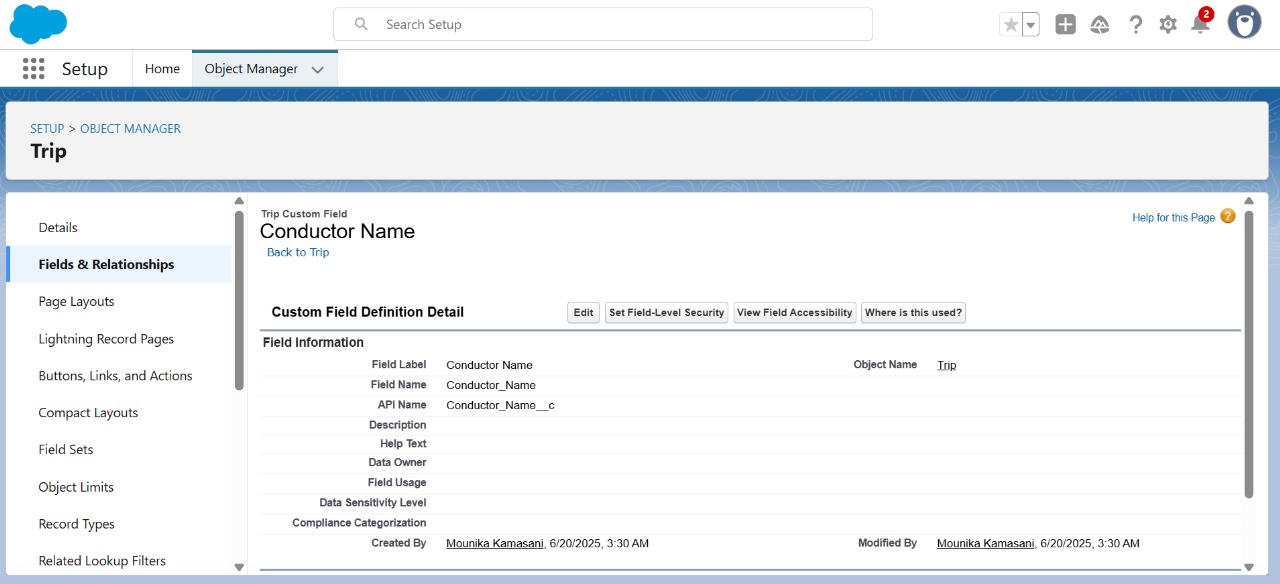
1. Go to setup >>  click on Object Manager >> type object name(Trip) in quick find box>> click on the Trip object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Driver Name.
6. Select formula return type Text, Click Next.
7. Create and insert Advance formula: Driver\_Id\_\_r.Employee\_Name\_\_c
8. Click Next, Next, then Save.



## 13. Creating a Conductor Name Formula Field in Trip object

 To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Trip) in quick find box >> click on the Trip object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Conductor Name.
6. Select formula return type Text, Click Next.
7. Create and insert Advance formula: Conductor\_Id\_\_r.Employee\_Name\_\_c
8. Click Next, Next, then Save.



## 14. Creating a Driver Name Formula Field in Trip object

1. From the Setup menu, enter "Picklist Value Sets" in the Quick Find box and select it.
2. Click on "New" to create a new global value set.
3. Enter the label Bus Time.
4. In values select “Enter values, with each value separated by a new line" and enter values as shown below.

6:00 AM

7:00 AM

8:00 AM

9:00 AM

10:00 AM

11:00 AM

12:00 PM

1:00 PM

2:00 PM

3:00 PM

4:00 PM

5:00 PM

6:00 PM

7:00 PM

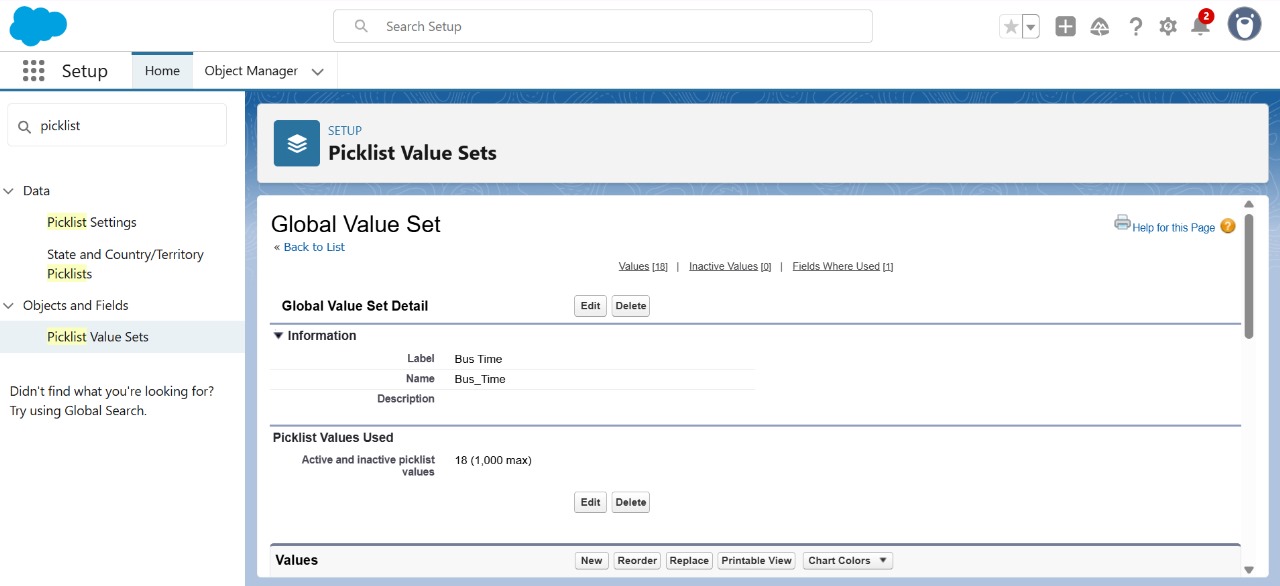
8:00 PM

9:00 PM

10:00 PM

11:00 PM

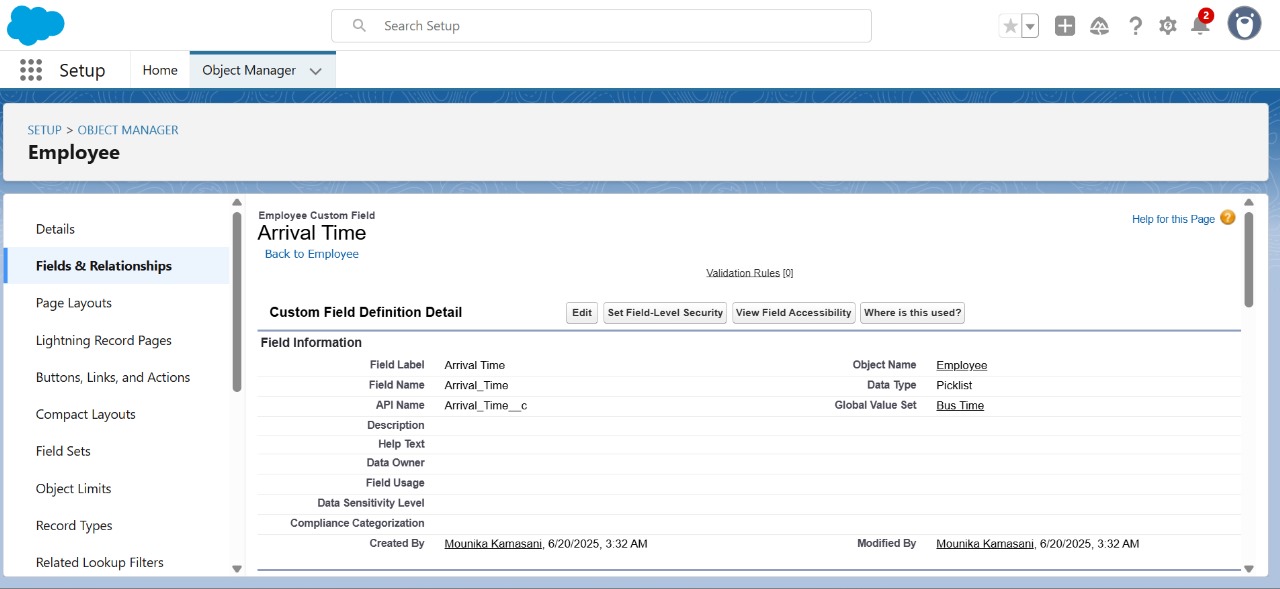
1. Click "Save" to create the global value set



## 15. Creating a Picklist Field using global picklist value set in Trip Object

 To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box>> click on the Employee Object.
2. Now click on “Fields & Relationships” .
3. Click on New.
4. Select Data type as “Picklist” and click Next.
5. Enter Field Label as “Arrival Time”.
6. Select "Use global picklist value set" and choose the global value set “Bus Time”.
7. Click on Next, Next and Save.



## 16. Creating a Picklist Field using global picklist value set in Trip Object

 Creating Controlling picklist field in Bus object:

1. Go to setup >> click on Object Manager >> type object name(Bus) in quick find box >> click on the Bus Object.
2. Now click on “Fields & Relationships” .
3. Click on New.
4. Select Data type as “Picklist” and click Next.
5. Enter Field Label as “Category”.
6. In values select “Enter values, with each value separated by a new line" and enter values as shown below.

Local

Intercity

InterState

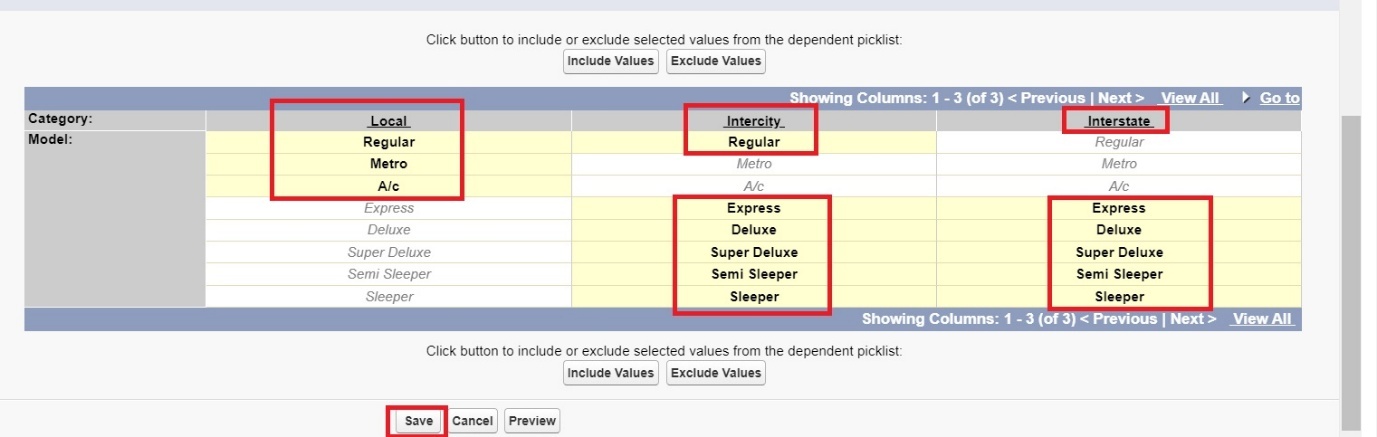
1. Select Required, Always require a value in this field in order to save a record .
2. Click on Next, Next and Save.

Creating Dependent picklist field in Bus object:

1. Go to setup >> click on Object Manager >> type object name(Bus) in quick find box >> click on the Bus Object.
2. Now click on “Fields & Relationships” .
3. Click on New.
4. Select Data type as “Picklist” and click Next.
5. Enter Field Label as “Category”.
6. Select "Use global picklist value set" and choose the global value set “Bus Time”.
7. Select Required, Always require a value in this field in order to save a record .
8. Click on Next, Next and Save.

Establish the Dependency

1. Go to setup >> click on Object Manager >> type object name(Bus) in quick find box >> click on the Bus Object
2. Now click on “Fields & Relationships” .
3. Now click on “Field Dependencies” .
4. Click on New.
5. Enter Controlling Field : Category
6. Enter Dependent field : Model
7. Click "Continue".
8. Matrix with the controlling field values on the top and the dependent field values on the side. Check the boxes to define which dependent picklist values should be available for each controlling picklist value.
9. Click Save

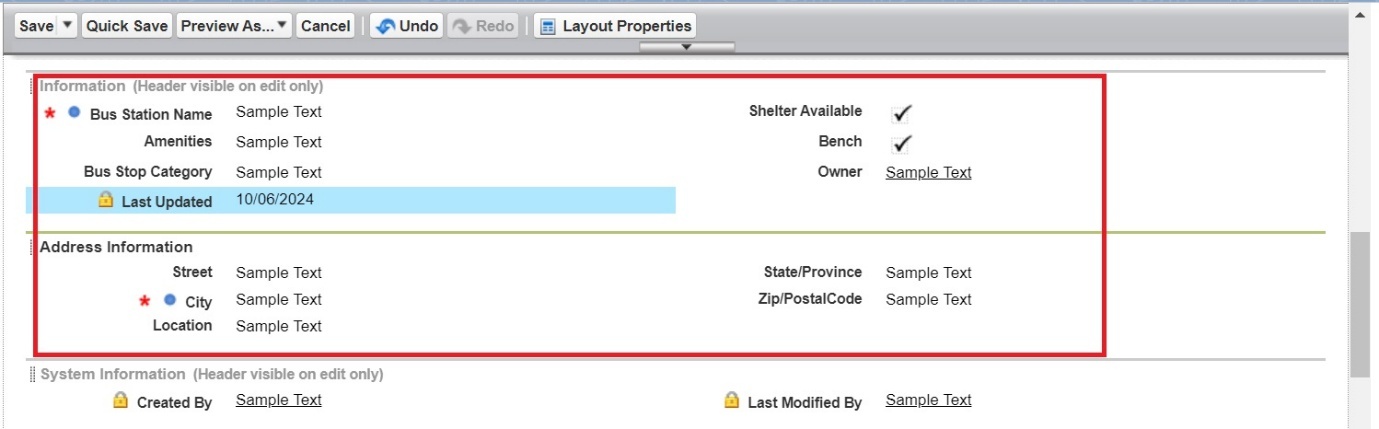


# Editing of Page Layouts

Page layouts in Salesforce control the layout and organization of fields, related lists, custom links, and other elements on a record detail or edit page. They are essential for managing how data is presented to users and can vary based on user roles and profiles.

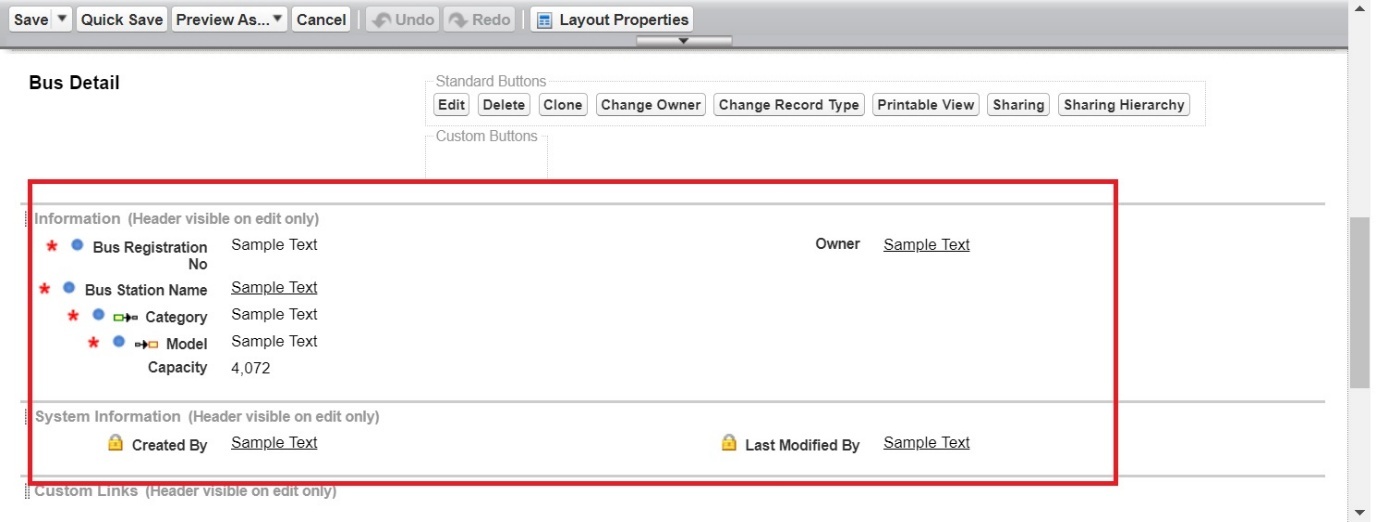
## 1. To edit a Page Layout in Bus Station Object

1. Go to setup >> click on Object Manager >> type object name(Bus Station) in quick find box >> click on the Bus Station object >> Page Layouts .
2. Click on the Bus Station Layout.
3. Drag and Arrange the field as shown below.
4. Save



## 2. To create a Page Layout in Bus Object

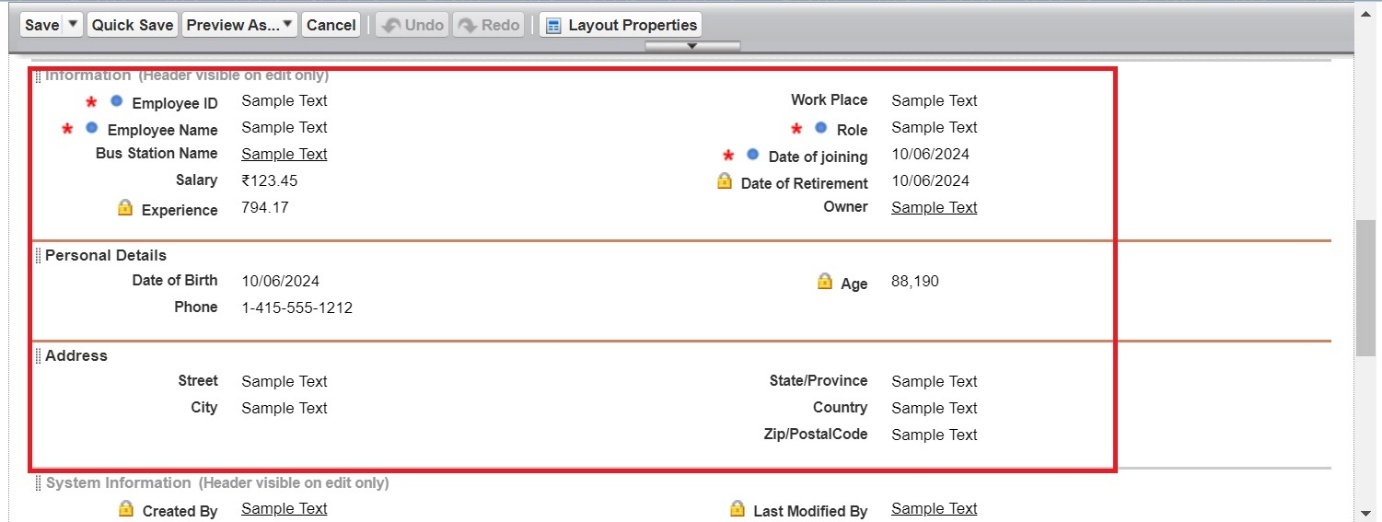
1. Go to setup >> click on Object Manager >> type object name(Bus) in quick find box >> click on the Bus object >> Page Layouts.
2. Click on the Bus Layout
3. Drag and Arrange the field as shown below



1. Click Save.

## 3. To create a Page Layout in Employee Object

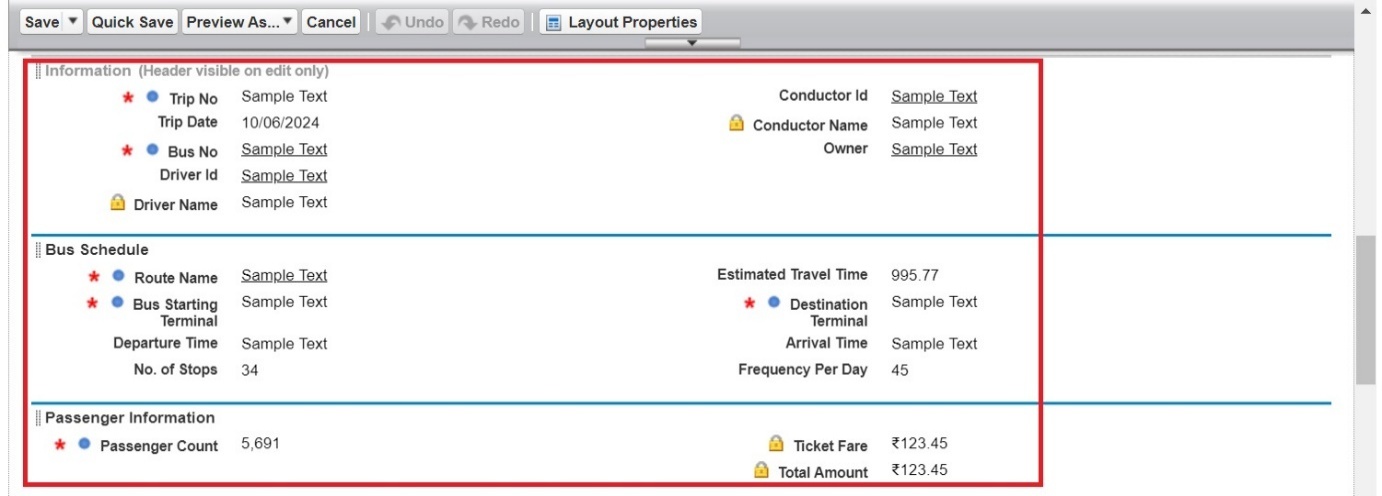
1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box >> click on the Employee object >> Page Layouts.
2. Click on the Employee Layout
3. Drag and Arrange the field as shown below



1. Click Save.

## 4. To create a Page Layout in Trip Object

1. Go to setup >> click on Object Manager >> type object name(Trip) in quick find box >> click on the Trip object >> Page Layouts.
2. Click on the Trip Layout
3. Drag and Arrange the field as shown below



1. Click on field Ticket Fare >> click on settings >> select Read Only and save it.
2. Click Save.

## 5. To create a Page Layout in Ticket Fare Object

## Go to setup >> click on Object Manager >> type object name(Ticket Fare) in quick find box >> click on the Ticket Fare object >> Page Layouts.

## Click on the Ticket Layout

## Drag and Arrange the field as shown below

## 

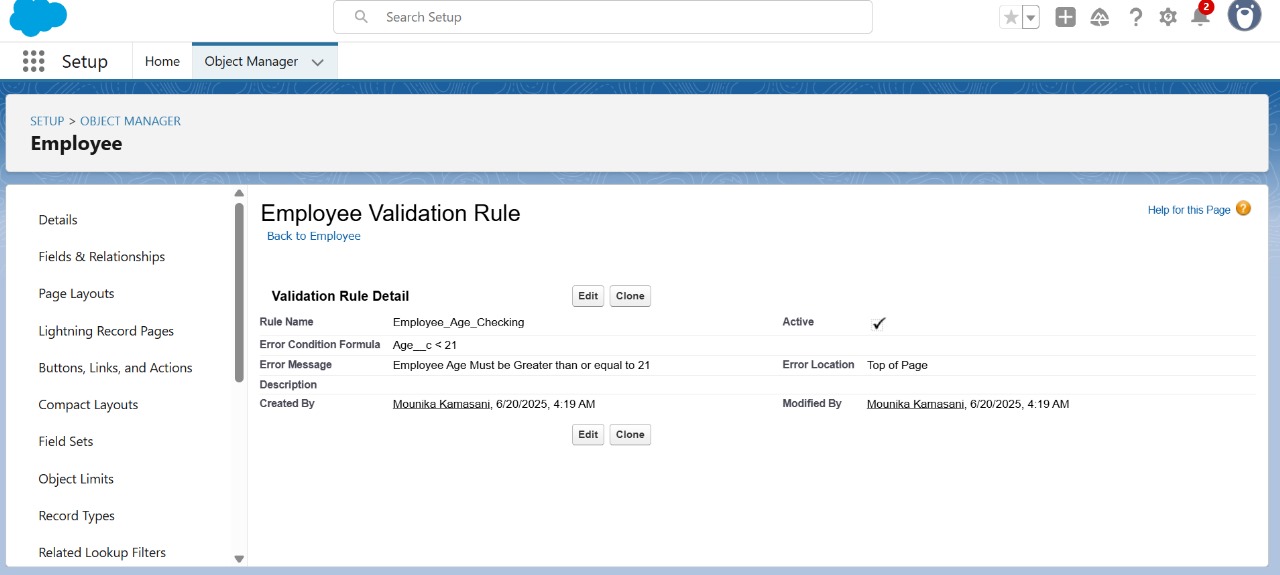
## Click Save.

# Creating the validation rule

Validation rules in Salesforce are used to ensure data integrity by preventing users from entering invalid data into fields. They consist of a logical formula or expression that evaluates the data in one or more fields and returns true or false. If the rule returns true, an error message is displayed, and the record is not saved.

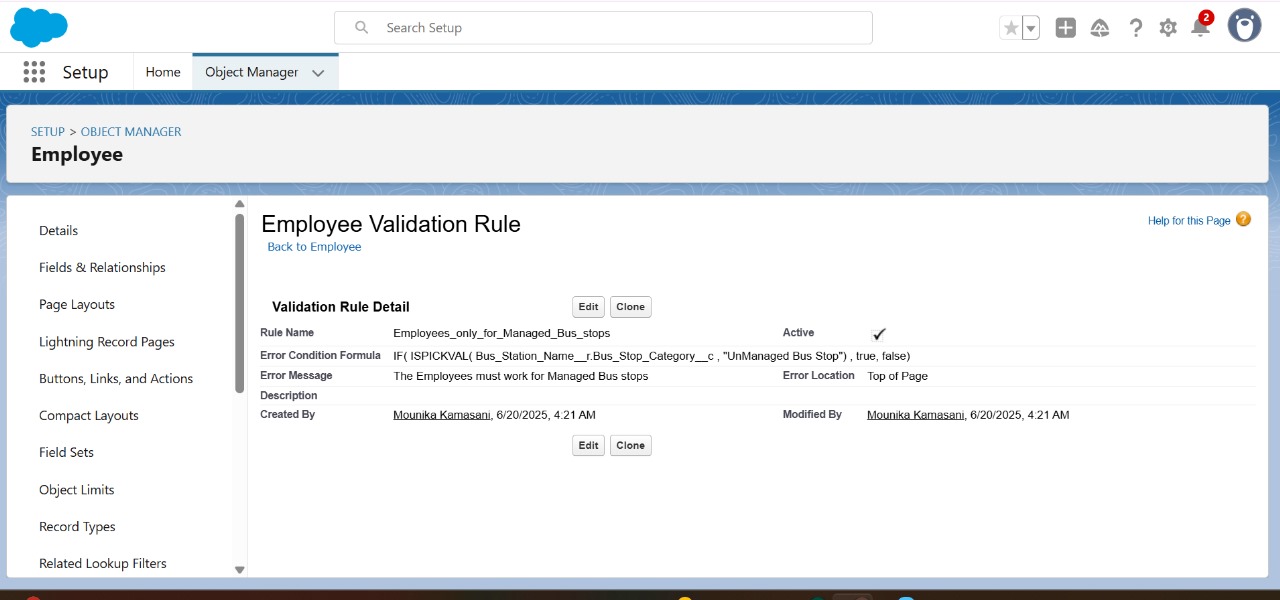
**1: To create a validation rule to a Employee Object**

1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box >> click on the Employee object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Employee\_Age\_Checking”.
4. Select Active
5. Insert the Error Condition Formula as :  
   Age\_\_c  < 21
6. Enter the Error Message as “Employee Age Must be Greater than or equal to 21”.
7. Select the Error location as Top of Page
8. Click Save.



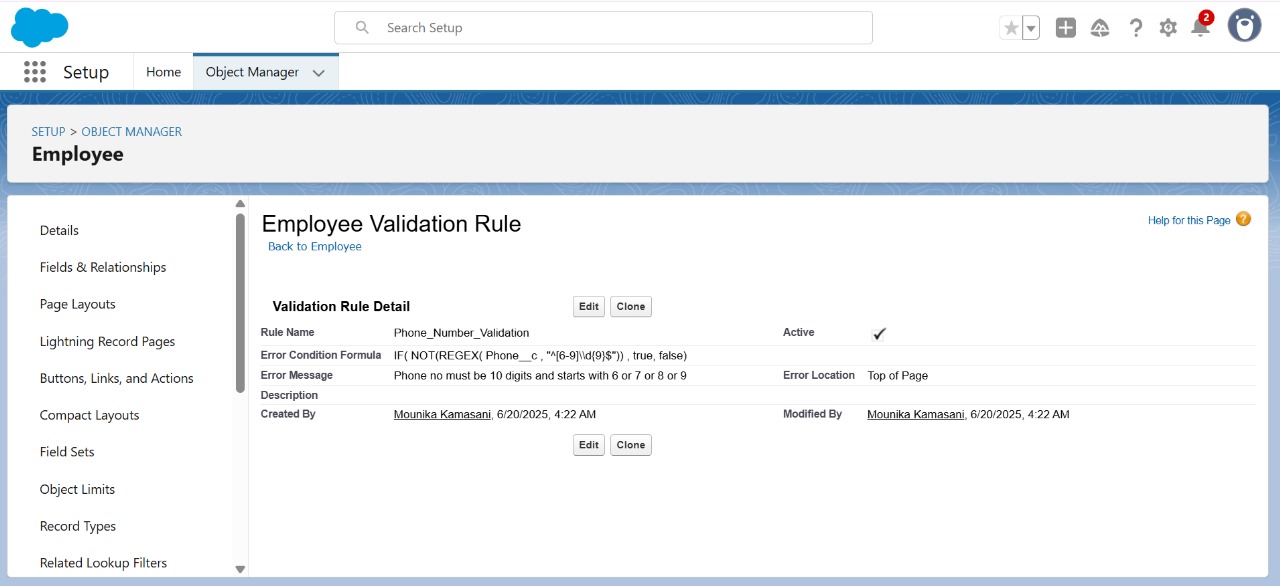
**2: To create a validation rule to a Bus Station on Employee Object**

1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box >> click on the Banquet Hall object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Employees\_only\_for\_Managed\_Bus\_stops”.
4. Select Active
5. Insert the Error Condition Formula as :  
   IF( ISPICKVAL( Bus\_Station\_Name\_\_r.Bus\_Stop\_Category\_\_c , "UnManaged Bus Stop") , true, false)
6. Enter the Error Message as “The Employees must work for Managed Bus stops”.
7. Select the Error location as Field and as Bus Station Name and click Save.



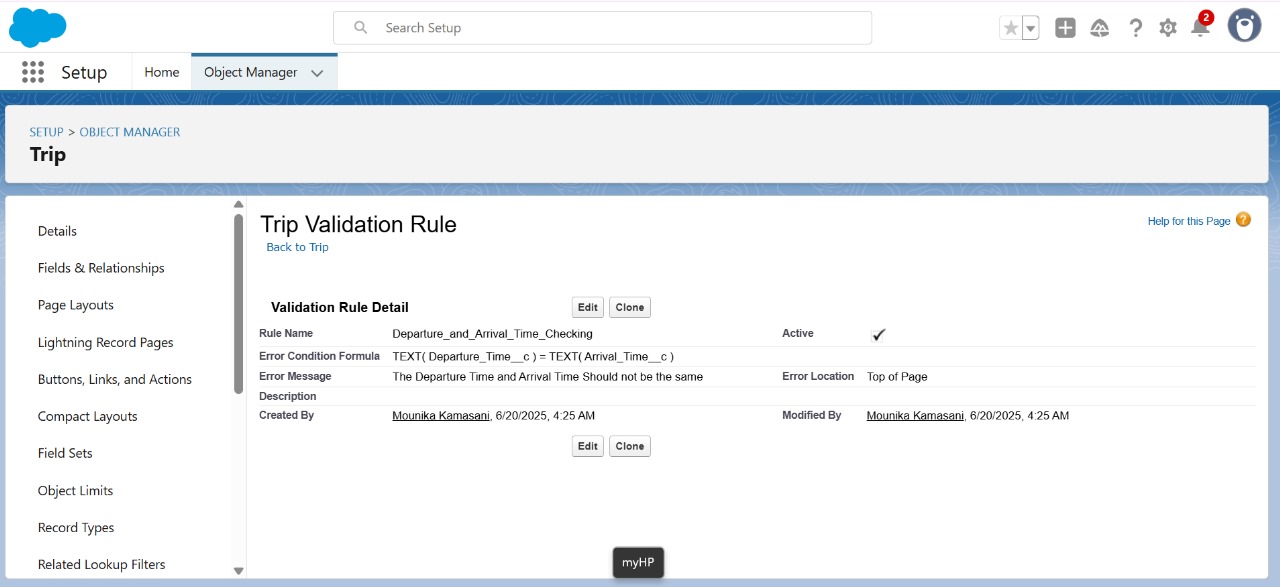
**3: To create a validation rule to a Phone No on Employee Object**

1. Go to setup >> click on Object Manager >> type object name(Employee) in quick find box>> click on the Employee object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Phone\_Number\_Validation”.
4. Select Active
5. Insert the Error Condition Formula as :  
   IF( NOT(REGEX( Phone\_\_c , "^[6-9]\\d{9}$")) ,    true,  false)
6. Enter the Error Message as “Phone no must be 10 digits and starts with 6 or 7 or 8 or 9”.
7. Select the Error location as Top of Page and click Save.



**4: To create a validation rule to a Trip Object**

1. Go to setup >> click on Object Manager >> type object name(Trip) in quick find box >> click on the Trip object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Departure\_and\_Arrival\_Time\_Checking”.
4. Select Active
5. Insert the Error Condition Formula as :  
   TEXT( Departure\_Time\_\_c )  =  TEXT( Arrival\_Time\_\_c )
6. Enter the Error Message as “The Departure Time and Arrival Time Should not be the same”.
7. Select the Error location as Top of Page and click Save.



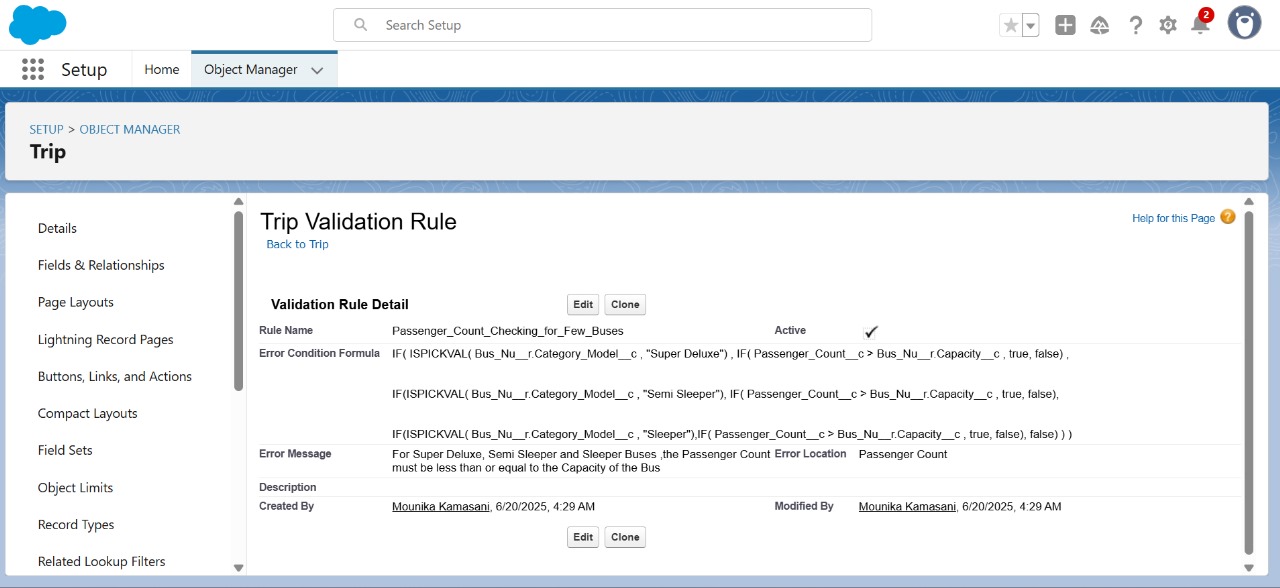
**5: To create a second validation rule to a Trip Object**

1. Go to setup >> click on Object Manager >> type object name(Trip) in quick find box >> click on the Trip object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Passenger\_Count\_Checking\_for\_Few\_Buses”.
4. Select Active
5. Insert the Error Condition Formula as :  
   IF( ISPICKVAL( Bus\_No\_\_r.Model\_\_c , "Super Deluxe") ,  IF(     Passenger\_Count\_\_c  >  Bus\_No\_\_r.Capacity\_\_c , true, false) ,

 IF(ISPICKVAL( Bus\_No\_\_r.Model\_\_c , "Semi Sleeper"), IF( Passenger\_Count\_\_c  >  Bus\_No\_\_r.Capacity\_\_c , true, false),

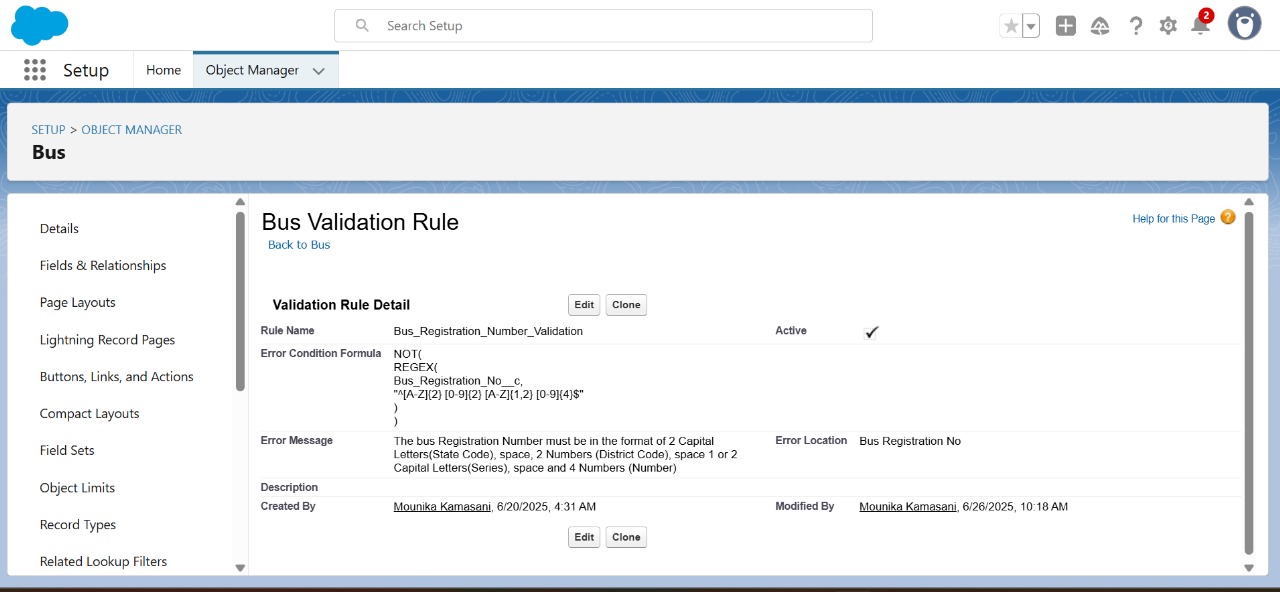
 IF(ISPICKVAL( Bus\_No\_\_r.Model\_\_c , "Sleeper"),IF( Passenger\_Count\_\_c  >  Bus\_No\_\_r.Capacity\_\_c , true, false), false) ) )

1. Enter the Error Message as “For Super Deluxe, Semi Sleeper and Sleeper Buses ,the Passenger Count must be less than or equal to the Capacity of the Bus”.
2. Select the Error location as Field and as Passenger Count and click Save.



**6: To create a validation rule to a Bus Object**

1. Go to setup >> click on Object Manager >> type object name(Bus) in quick find box >> click on the Bus Hall object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Bus\_Registration\_Number\_Validation”.
4. Select Active
5. Insert the Error Condition Formula as :  
   IF( NOT( REGEX( Name ,"^[A-Z]{2}\\s\\d{2}\\s[A-Z]{1,2}\\s\\d{4}$" )), true, false)
6. Enter the Error Message as “The bus Registration Number must be in the format of 2 Capital Letters(State Code), space, 2 Numbers (District Code), space 1 or 2 Capital Letters(Series), space and 4 Numbers (Number).”.
7. Select the Error location as Field and as Bus Registration No and click Save.



# Flows

In Salesforce, a Flow is an automation tool that allows you to create complex business processes using a visual interface. Flows can be used to collect, update, delete, and create records, as well as to guide users through a series of screens to complete a process.

**Creating Flow to Fetch Ticket Fare for Bus.**

1. Go to setup ? type Flow in quick find box ? Click on the Flow and Select the New Flow.
2. Select the record Triggered flow.Click on create
3. Under Object select “Trip”
4. Select  A record is created or updated
5. Set Entry Conditions : None
6. Select Actions and Related Records and click on Done
7. Under the record trigger flow click on the “+” icon and select Get Records.
8. Enter Label as “ Fetching Route Ticket Fares ”.
9. For Object select Ticket Fare
10. For Condition Requirements , select All Conditions are Met(AND)

        For the first condition select as follows:

                                 Field: Id

                                 Operator: Equals

                          Value: {!$Record.Route\_Name\_\_c}

1. For How many Records to store Select Only the First Record.
2. For How to Store Record Data select Choose fields and let Salesforce do the rest. Select Field: Ticket\_Fare\_\_c.  Click on Done.
3. From the Toolbox drag and drop Decision element.
4. Enter the Decision label as “Bus Model Checking”.
5. For Outcome Details:

* Label : Ticket Fare Checking
* Outcome API Name : Ticket\_Fare\_Checking
* Condition Requirements to Execute Outcome : All Conditions are Met (AND)
* Resource : {!Fetching\_Route\_Ticket\_Fares.Ticket\_Fare\_\_c}
* Operator : Is Null
* Value : {!$GlobalConstant.False}

1. Click Done
2. From the Toolbox drag and drop Update Records element and connect to Decision element for Ticket Fare Fetching Output.
3. Enter the label as “Updating Trip Object Ticket Fare Field”.
4. How to Find Records to Update and Set Their Values : Use the trip record that triggered the flow
5. Set Filter Conditions : None -Always Update Record
6. Set Field Values for the Trip Record as

 Field :  Ticket\_Fare\_\_c

 Value : {!Fetching\_Route\_Ticket\_Fares.Ticket\_Fare\_\_c}

       And click Done

1. From the Toolbox drag and drop Custom Error Message element and connect to Default Outcome of Decision element..
2. Enter the label as “Route with Bus Model  Does not Exists”.
3. For Where to Show the Error Message: Select In a window on a record page
4. Error Message: There is no Record with the Route {!$Record.Route\_Name\_\_r.Name} and the Bus Model {!$Record.Bus\_No\_\_r.Model\_\_c} in the Ticket Fares
5. Click Done
6. Save the flow as “Fetching Ticket Fare For Bus”
7. Activate the flow.



# Trigger

## Triggers in Salesforce are pieces of Apex code that execute before or after specific database operations, such as insert, update, delete, or undelete. They allow you to perform custom actions on records in Salesforce when certain events occur. Triggers are particularly powerful for enforcing business logic and automating workflows.

**Create a Trigger to validate whether the Driver Id and Conductor Id are correct or not.**

Step 1 : Login to Salesforce:

Log in to your Salesforce account with administrative privileges.

Step 2:

i)Navigate to Setup: Once logged in, click on the gear icon ?? (Setup) located at the top-right corner of the page. This will open the Setup menu.

ii)Click on Developer Console: Click on the "Developer Console" option from the Setup menu.   This will open the Developer Console in a new browser tab or window.

Step 3:

1) In the Developer Console window, go to the top menu and click on "File".

2)Select New: From the dropdown menu under "File", select "New".

3)Choose Apex Class: In the submenu that appears, select "Apex Class". This will open a new Apex Class editor tab.

Give Class Name : TripTriggerHandlerClass

Create an Apex Class:

Public class TripTriggerHandlerClass {

// Checking whether the entered Driver Id belongs to a Driver or not

 public Static void driverValidation(List<Trip\_\_c> tripList){

      List<Employee\_\_c> driverList = [SELECT Id, Name FROM Employee\_\_c WHERE Role\_\_c ='Driver' ];

     If(driverList != null){

        Map<Id, String> driverMap = new Map<Id, String>();

        for(Employee\_\_c emp : driverList ){

            driverMap.put(emp.Id, emp.Name);

        }

        for(Trip\_\_c trip : tripList ){

            If(trip.Driver\_Id\_\_c!=null){

                 Boolean hasDriverId = driverMap.containskey(trip.Driver\_Id\_\_c); // hasDriverId will be true

                 If(hasDriverId == false){

                    trip.addError('The assigned person is not a Driver.');

                 }

            }

        }

     }

    }

// Checking whether the entered conductor Id belongs to a Conductor or not

    public Static void conductorValidation(List<Trip\_\_c> tripList){

        List<Employee\_\_c> conductorList = [SELECT Id, Name FROM Employee\_\_c WHERE Role\_\_c ='Conductor' ];

      Map<Id, String> conductorMap = new Map<Id, String>();

        for(Employee\_\_c emp : conductorList ){

            conductorMap.put(emp.Id, emp.Name);

        }

        for(Trip\_\_c trip : tripList ){

            If(trip.Conductor\_Id\_\_c!=null){

                 Boolean hasConductorId = conductorMap.containskey(trip.Conductor\_Id\_\_c); // hasConductorId will be true

                 If(hasConductorId == false){

                    trip.addError('The assigned person is not a Conductor.');

                 }

            }

        }

    }

}  
  
Step 4:

i) In the Developer Console window, go to the top menu and click on "File".

ii)Select New: From the dropdown menu under "File", select "New".

iii)Choose Apex Class: In the submenu that appears, select "Apex Trigger". This will open a new Apex Trigger editor tab.

Create an Apex Trigger:

trigger TripTrigger on Trip\_\_c (before insert, before update) {

    if(trigger.isBefore){

        if(trigger.isInsert || trigger.isUpdate){

            // Validating the Conductor Id in Trip is really a Conductor or not

            TripTriggerHandlerClass.driverValidation(trigger.new);

            // Validating the Conductor Id in Trip is really a Conductor or not

            TripTriggerHandlerClass.conductorValidation(trigger.new);

        }

    }

}



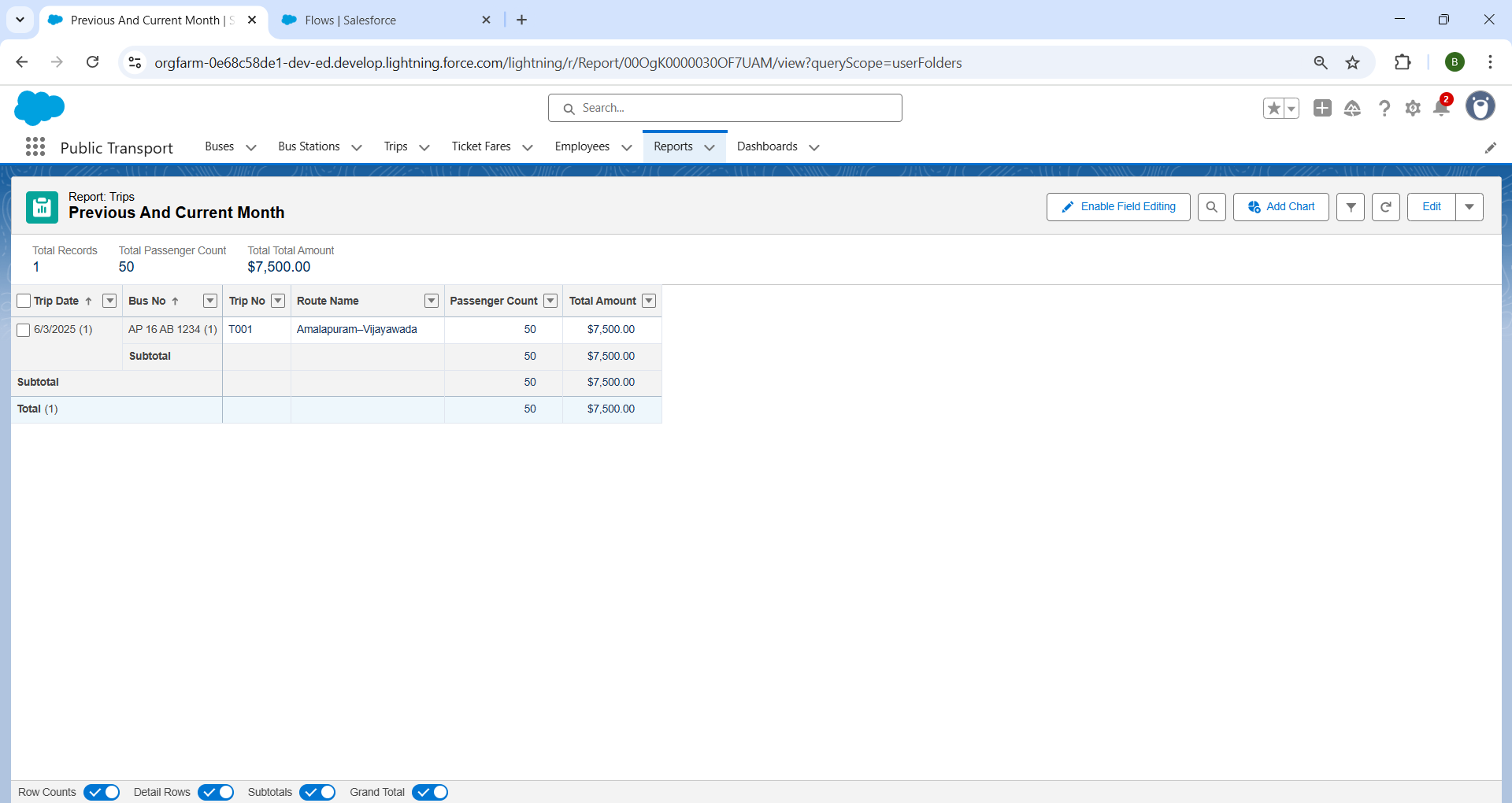
### Reports

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

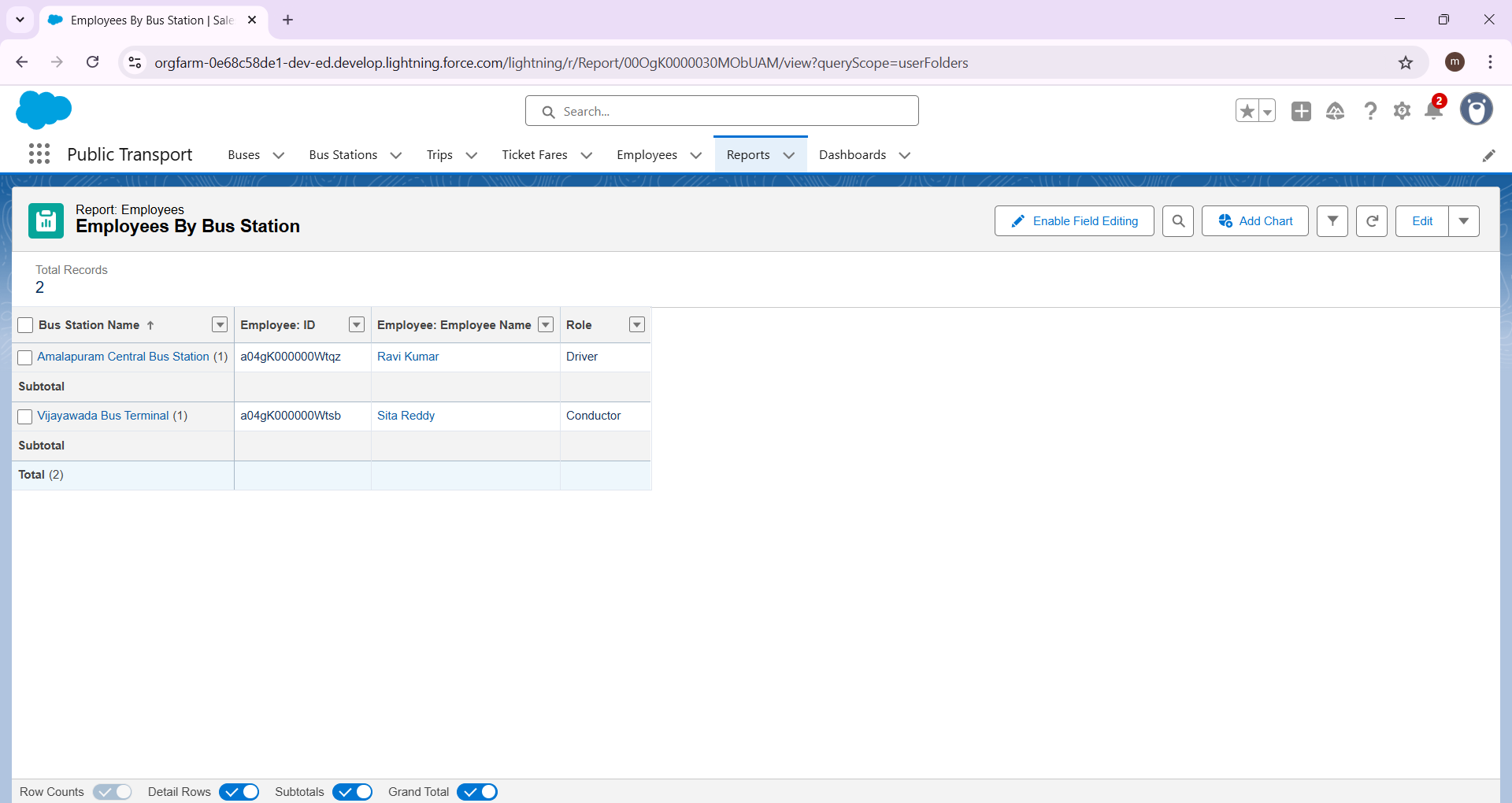
Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix
4. Joined Reports

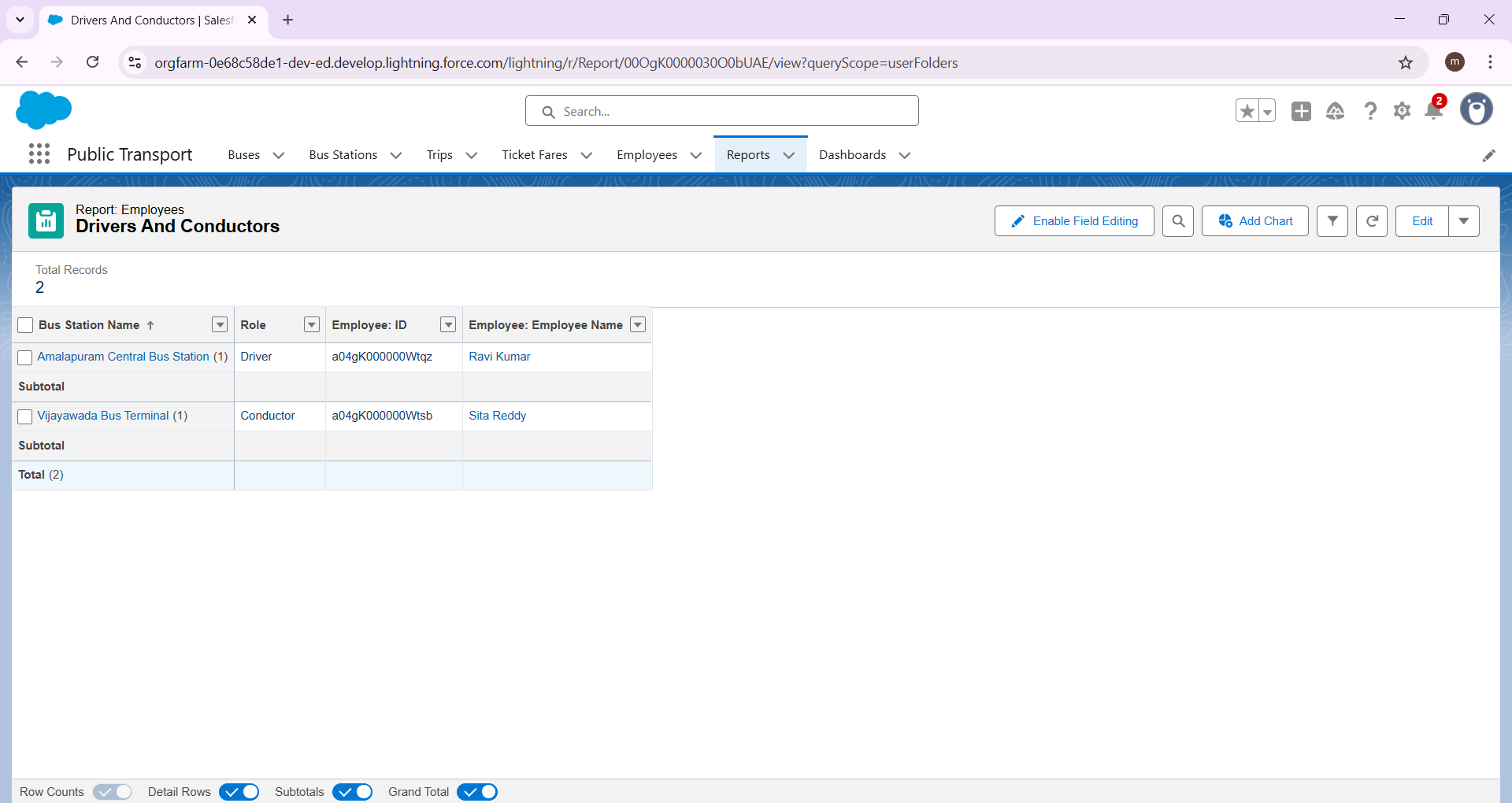
**1.Previous And Current Month**



**2.Employees By Bus Station**



**3.Drivers and Conductors**

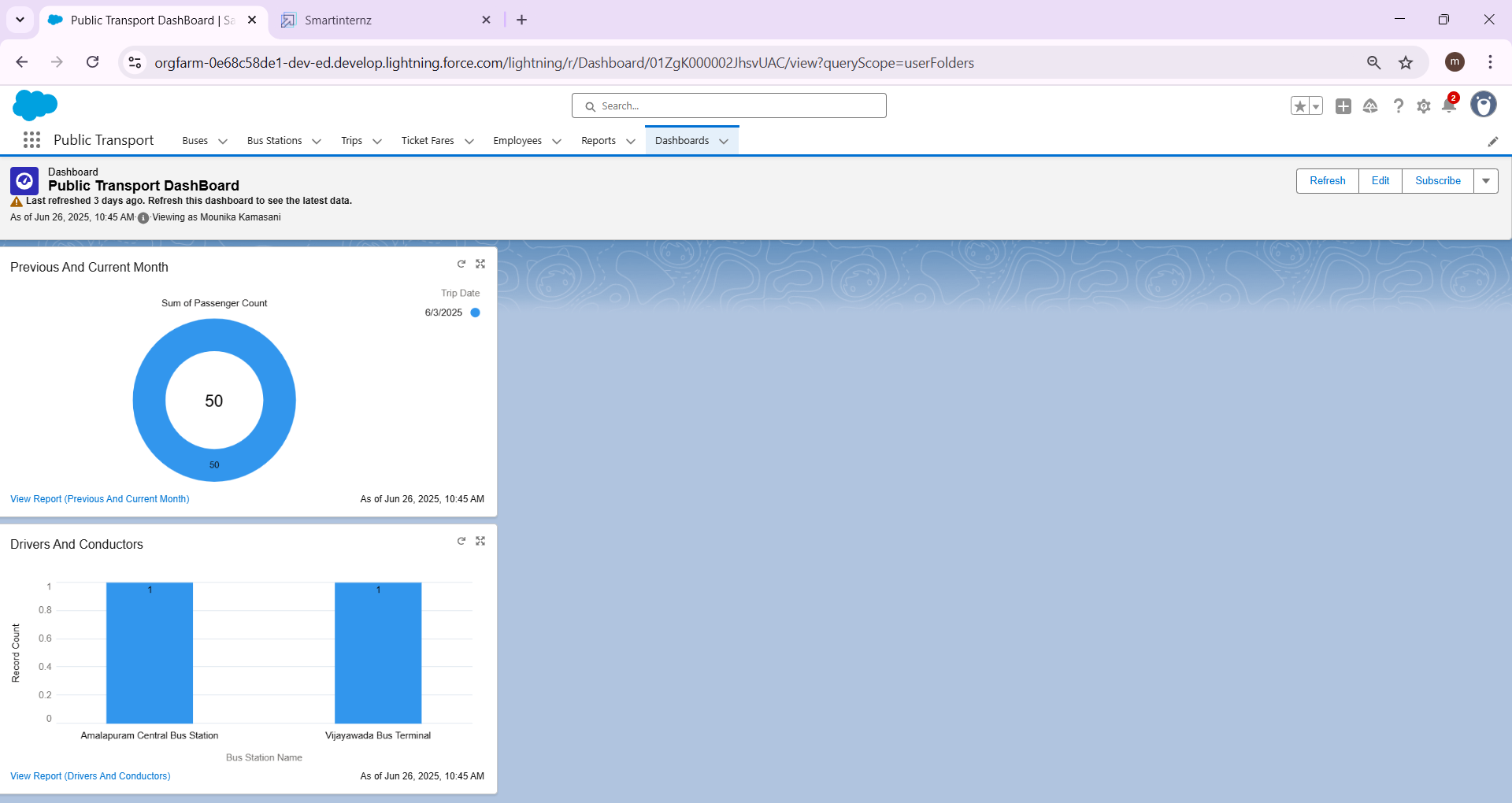
****

# Dashboards

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you’ve gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

Steps to Create:

1. Click on the Dashboards tab from the Public Transport(RTC) application.
2. Click on the new dashboard.
3. Give name - Public Transport DashBoard
4. Click create
5. Click on +widget
6. Select the  Previous and Current Month Trips Details   Report
7. For the data visualization select any of the charts, tables etc. as per your choice/requirement
8. Click add.
9. Click on +widget
10. Select the  Drivers and Conductors Information Report
11. For the data visualization select any of the charts, tables etc. as per your choice/requirement
12. Click add.
13. Click save.



# Conclusion

The **Public Transport (RTC) Management System** developed on the Salesforce platform offers a comprehensive, scalable, and user-friendly solution to streamline and modernize the operations of a public transport organization. Leveraging Salesforce’s powerful CRM capabilities, custom objects, flows, triggers, and reports, the system effectively manages critical entities such as Bus Stations, Buses, Trips, Employees, and Ticket Fare structures.

By centralizing operational data and automating routine tasks, the application enhances data accuracy, reduces administrative overhead, and improves service delivery. Features like role-based dashboards, validation rules, and real-time reporting empower the management to make informed, data-driven decisions. Additionally, user-friendly interfaces and well-structured workflows support cross-functional collaboration and better coordination among departments.

Overall, this Salesforce-based CRM solution not only addresses the existing inefficiencies in transport management but also establishes a strong digital foundation for future expansions, including mobile accessibility, passenger engagement, and integration with other government systems.

## ------------------------------------THE END----------------------------------