

CRM APPLICATION FOR JEWEL MANAGEMENT
SALESFORCE DEVELOPER NAAN MUDHALVAN
PROJECT REPORT

Submitted By

M.Divya(611420104019)
P.Kalaiyarasi(611420104029)
M.Kowsalya(611420104030)
T.ReenaDevi(611420104064)

in partial fulfillment for the award of the
degree of

BACHELOR OF ENGINEERING
in

COMPUTER SCIENCE AND ENGINEERING

MAHENDRA ENGINEERING COLLEGE
FOR WOMEN

Tiruchengode 637205

BONAFIDE CERTIFICATE

Certified that this project report titled “**CRM APPLICATION FOR JEWEL MANAGEMENT**” is the Bonafide work of “**M.Divya(611420104019), P.Kalaiyarasi(611420104029), M.Kowsalya(611420104030), T.ReenaDevi(611420104064)**” who carried out the project work under my supervision.

SIGNATURE

Dr.A.KANCHANA M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of Computer
Science and Engineering ,
Mahendra Engineering College
for Women ,
Tiruchengode,
637205 .

SIGNATURE

MrS.N.KOWSALYA M.E.,

FACULTY MENTOR

ASSISTANT PROFESSOR

Department of Computer
Science and Engineering,
Mahendra Engineering College
for Women,
Tiruchengode,
637205 .

SPOC

HEAD OF THE DEPARTMENT

ACKNOWLEDGEMENT

At the outset, we express our heartfelt gratitude to **GOD**, who has been our strength to bring this project to light.

At this pleasing moment of having successfully completed our project, we wish to convey our sincere thanks and gratitude to our beloved president **Mr. C. BALAKRISHNAN**, who has provided all the facilities to us.

We would like to convey our sincere thanks to our beloved Principal **Dr. PSS. SRINIVASAN**, for forwarding us to do our project and offering adequate duration in completing our project.

We express our sincere thanks to our Head of the Department **Dr. M. RAMKUMAR**, Department of Computer Science and Business Systems for fostering the excellent academic climate in the Department.

We express our pronounced sense of thanks with deepest respect and gratitude to our Faculty Mentor **Mr.B.VENKATARAMANEN**, Department of Computer Science and Business Systems for their valuable and precious guidance and for having amicable relation.

With deep sense of gratitude, we extend our earnest and sincere thanks to our SPOC **Mr.T.KARTHIKEYAN**, Assistant Professor, Department of Computer Science and Engineering for his guidance and encouragement during this project.

We would also like express our thanks to all the faculty members of our Department, friends and students who helped us directly and indirectly in all aspects of the project work to get completed successfully.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	INTRODUCTION	1
2	PROJECT SPECIFICATIONS	2
	2.1 Project Goal	
	2.2 Project Scope	
	2.3 Technical Requirements	
	2.4 Functional Requirements	
3	PREPARATION DATA MODELING	7
4	USERS & DATA SECURITY	22
5	AUTOMATION	31
6	REPORTS & DASHBOARD	39
	GitHub & Project Video Demo Link	42

1.INTRODUCTION

Salesforce, a leading cloud-based Customer Relationship Management (CRM) platform, is a pivotal tool for organizations to manage customer data, optimize sales processes, and elevate customer interactions. Its multifaceted features include Sales Cloud, which enhances sales management through lead tracking, opportunity management, and seamless email integration. Service Cloud focuses on exceptional customer support, featuring case management, knowledge base development, and multi-channel support. Marketing Cloud empowers businesses with marketing automation, email campaigns, social media engagement, and in-depth analytics. Salesforce's hallmark is its customization, allowing businesses to tailor the platform to meet specific requirements, while robust integration capabilities facilitate seamless connections with other business applications.

The platform equips businesses with powerful reporting and analytic tools, enabling data-driven decisions and insightful, customized reports and dashboards. Salesforce ensures mobile accessibility, enabling users to stay connected and productive while on the move. A paramount emphasis on data security and compliance guarantees data protection and privacy. Whether you're a small start-up or a large enterprise, Salesforce offers scalability to accommodate your evolving needs.

Through Salesforce, organizations foster improved customer relationships, increased sales efficiency, and superior customer support. It empowers businesses to make data-driven decisions, streamline operations, and create impact, targeted marketing campaigns. This introduction encapsulates Salesforce's capabilities and benefits, offering a concise overview for your project document, allowing for a better understanding of how the platform can contribute to your specific project goals.

2.PROJECT SPECIFICATIONS

2.1 Project Goal

The goal of the **CRM APPLICATION FOR JEWEL MANAGEMENT** developed on the Salesforce platform is to revolutionize the pizza delivery industry by providing an efficient, user-friendly, and seamless experience for both customers and delivery personnel. Through robust customization and integration of Salesforce tools, the app aims to optimize order management, enhance customer satisfaction, and improve operational efficiency. By automating key processes, ensuring real-time data visibility, and empowering users with insightful analytics, the project strives to transform pizza delivery services into a streamlined, reliable, and delightful experience, setting new standards for the industry.

2.2 Project Scope

- **Creation of Developer Account (Milestone 1):** In this phase, the project scope involves setting up a Salesforce developer account, ensuring access to necessary tools and resources, and initiating the groundwork for the Pizza Delivery App development process.
- **Object Creation (Milestone 2):** This milestone encompasses defining crucial objects such as "Order," "Customer," and "Menu Items" within Salesforce, specifying their fields and data types, laying the foundation for organizing and managing pizza delivery data effectively.
- **Tabs Creation (Milestone 3):** The scope here involves creating intuitive user interface tabs for different functionalities like order management, customer details, and menu browsing,

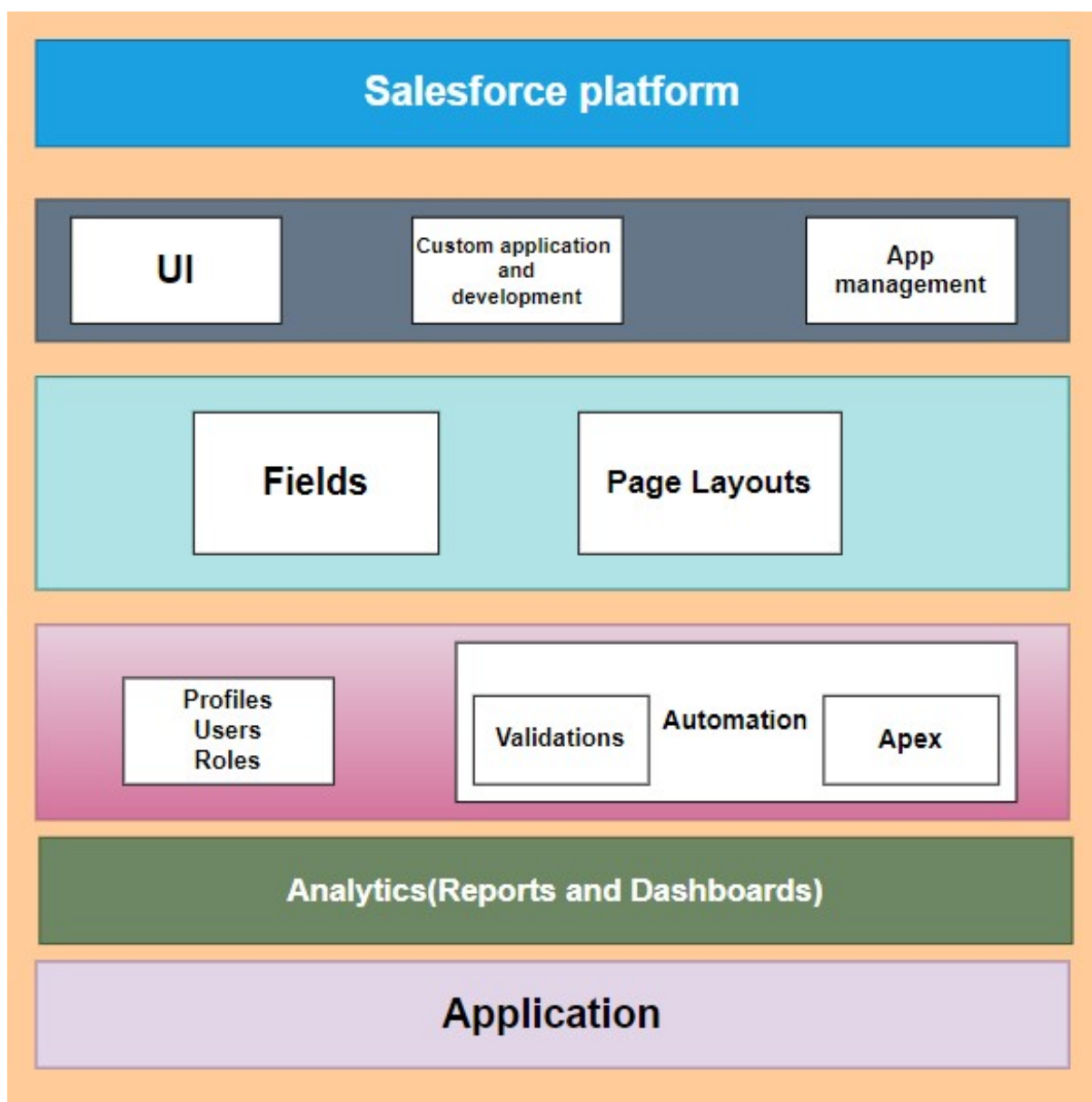
ensuring seamless navigation and user experience within the Pizza Delivery App.

- **Create App (Milestone 4):** This phase focuses on developing the Pizza Delivery App within the Salesforce platform, integrating the previously defined objects and tabs into a cohesive application framework, ready for further customization and enhancements.
- **Fields & Relationships (Milestone 5):** The project scope includes fine-tuning the app's data model by defining fields for capturing specific information and establishing relationships between objects, ensuring comprehensive data management capabilities.
- **Profile (Milestone 6):** This milestone involves implementing record types for differentiating between various entities, such as customer profiles and order types, allowing tailored user experiences and ensuring accurate data processing based on specific criteria.
- **Profile (Milestone 7):** The project scope comprises creating and configuring user profiles and permission sets, ensuring that different roles have appropriate access levels and privileges within the Pizza Delivery App.
- **User Adoption (Milestone 8):** The scope for this phase involves designing training modules and user guides, ensuring smooth onboarding and adoption of the Pizza Delivery App among employees and stakeholders, enhancing overall user satisfaction and productivity.
- **OWD (Milestone 9):** The project scope includes setting up Organization Wide Defaults, defining the baseline level of access for different objects and records, establishing the foundational security framework for the Pizza Delivery App.
- **Reports (Milestone 10):** This phase focuses on creating customized reports and dashboards within the Pizza Delivery App, enabling real-time data analysis, performance tracking, and

informed decision-making for optimizing delivery operations and customer service.

- **Apex Triggers (Milestone 11):** The project scope here involves implementing Apex Triggers to automate specific business logic and processes, enhancing the app's functionality by enabling actions like real-time notifications, validations, and data updates based on predefined conditions.

2.3 Technical Requirements



2.4 Functional Requirements

User Registration and Login:

- Users should be able to register accounts with valid credentials.
- Users should have the option to log in using email or social media accounts.

Menu Browsing:

- Customers should be able to view the menu, including pizza types, toppings, and prices.
- Customers should be able to customize their pizza orders based on available toppings and sizes.

Order Placement:

- Customers should be able to place orders by selecting pizzas from the menu.
- Customers should provide delivery addresses and contact details during order placement.

Order Tracking:

- Customers should be able to track the status of their orders in real-time.
- Delivery personnel should have access to GPS-based tracking to find the most efficient route for delivery.

Payment Processing:

- Customers should be able to make payments online securely.
- The system should support multiple payment methods, including credit/debit cards and digital wallets.

Order Management:

- Restaurant staff should receive notifications of new orders.
- Staff should have the ability to confirm, modify, or cancel orders based on availability and customer requests.

Customer Feedback and Ratings:

- Customers should be able to provide feedback and ratings for their orders and overall experience.

- Feedback data should be stored for analysis and improvement purposes.

Admin Dashboard:

- Admins should have access to a dashboard showing real-time order statistics, customer feedback, and delivery performance.
- Admins should be able to manage menu items, prices, and availability.

Customer Support:

- There should be a customer support interface allowing customers to contact support for order-related queries.
- Support staff should be able to view order details and assist customers effectively.

Reporting and Analytics:

- Generate reports on sales trends, popular menu items, and customer behaviour.
- Provide analytical insights to optimize inventory management and marketing strategies.

Security and Privacy:

- Ensure data encryption for user profiles, payment information, and order details.
- Comply with data protection regulations to safeguard customer privacy.

Integration with External Systems:

- Integrate with external services for geolocation, payment processing, and customer relationship management (CRM) if applicable.

3.PREPARATION DATA MODELING

Objects:

Salesforce objects are database tables that permit you to store data that is specific to an organization. It consists of fields (columns) and records (rows).

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.

In This Application We Use 8 Custom Objects:

1. EveryDay Values
2. Beverages
3. Customer Details
4. Best sellers
5. New launches
6. Pizzas
7. Mode of payments

1)Create A Custom Object for EveryDay Values:

1. Click on the **Gear Icon** >> From **setup** click on **object manager**
2. Click **create**, select **custom object**.
- 3.Fill in the label as "**EveryDay Value**".
4. Fill in the plural label as "**EveryDay Values**".
5. Record name : "**EveryDay Value**"
6. Select the data type as "**Text**".
7. In the Optional Features section, select Allow **Reports** and **Track** Field History.
- 8.In the Deployment Status section, ensure **Deployed** is selected.

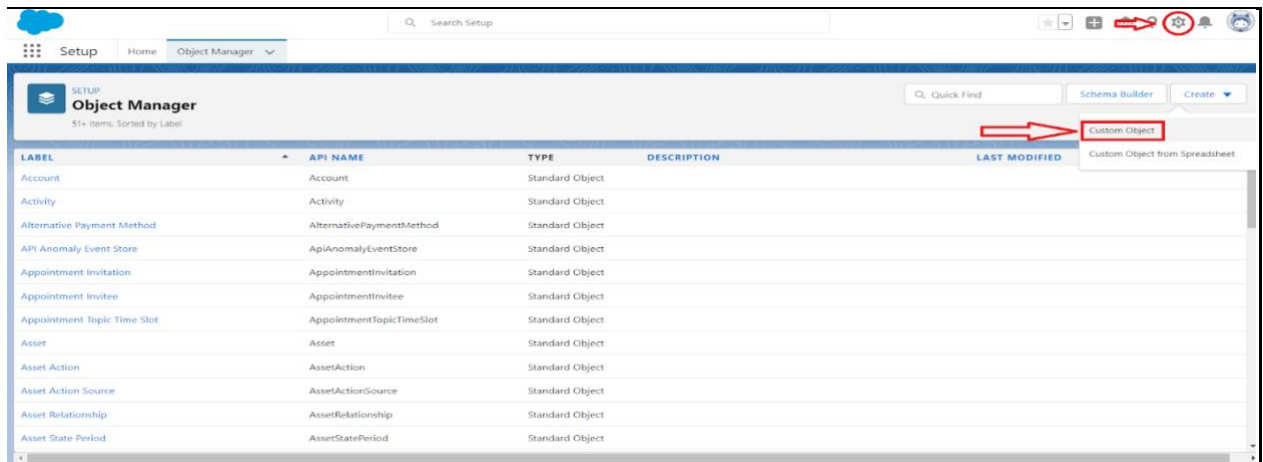
9. In the Search Status section, select **Allow Search**.
10. In the Object Creation Options section, select select these options: Add Notes and Attachments related list to default **page layout** Launch New Custom Tab Wizard after saving this custom object
11. Leave everything else as is, and click **Save**.

2)Create A Custom Object For Beverages:

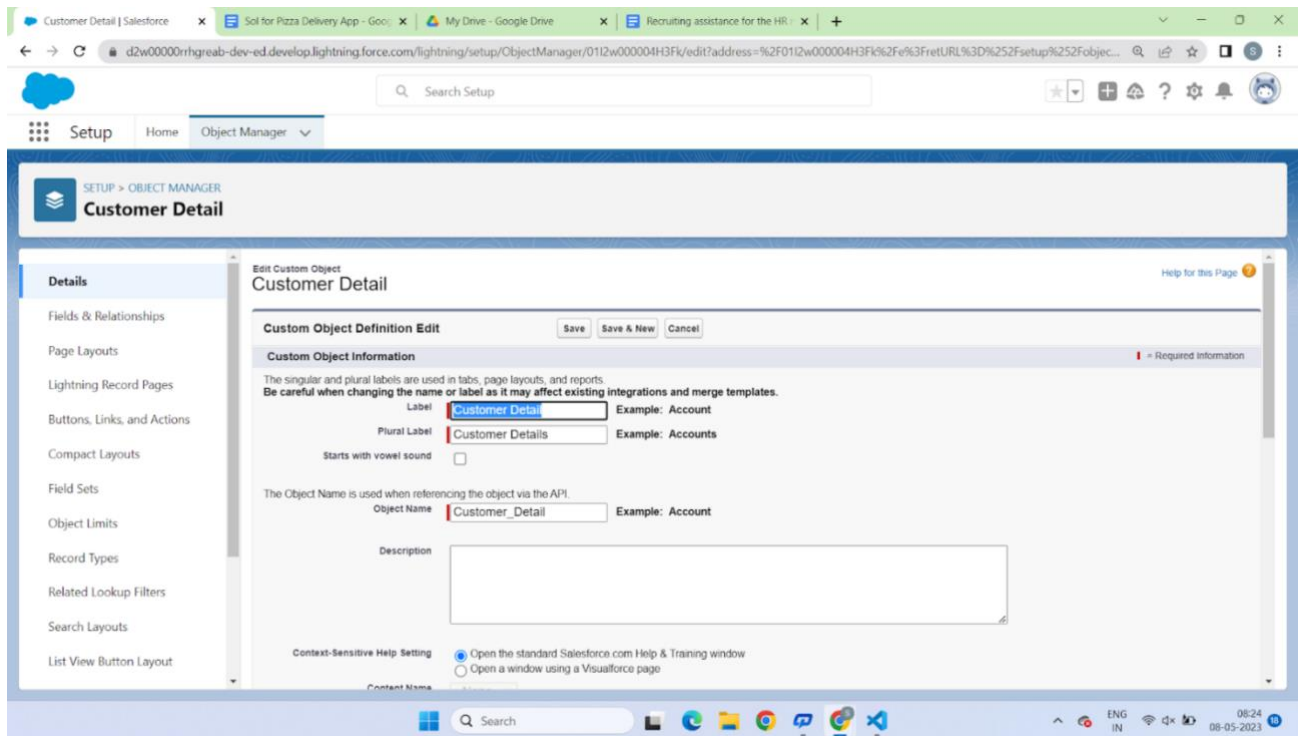
1. Click on the **Gear Icon >>** From **setup** click on **object manager**
2. Click **create**, select **custom object**.
3. 3. Fill in the label as "**Beverage**".
4. Fill in the plural label as "**Beverages**".
5. Record name : "**Beverage**"
6. Select the data type as "**Text**".
7. In the Optional Features section, select Allow **Reports** and **Track** Field History.
8. In the Deployment Status section, ensure **Deployed** is selected.
9. In the Search Status section, select **Allow Search**.
10. In the Object Creation Options section, select select these options:
Add Notes and Attachments related list to default **page layout**
Launch New Custom Tab Wizard after saving this custom object
11. Leave everything else as is, and click **Save**

3)Create A Custom Object For Customer Detail :

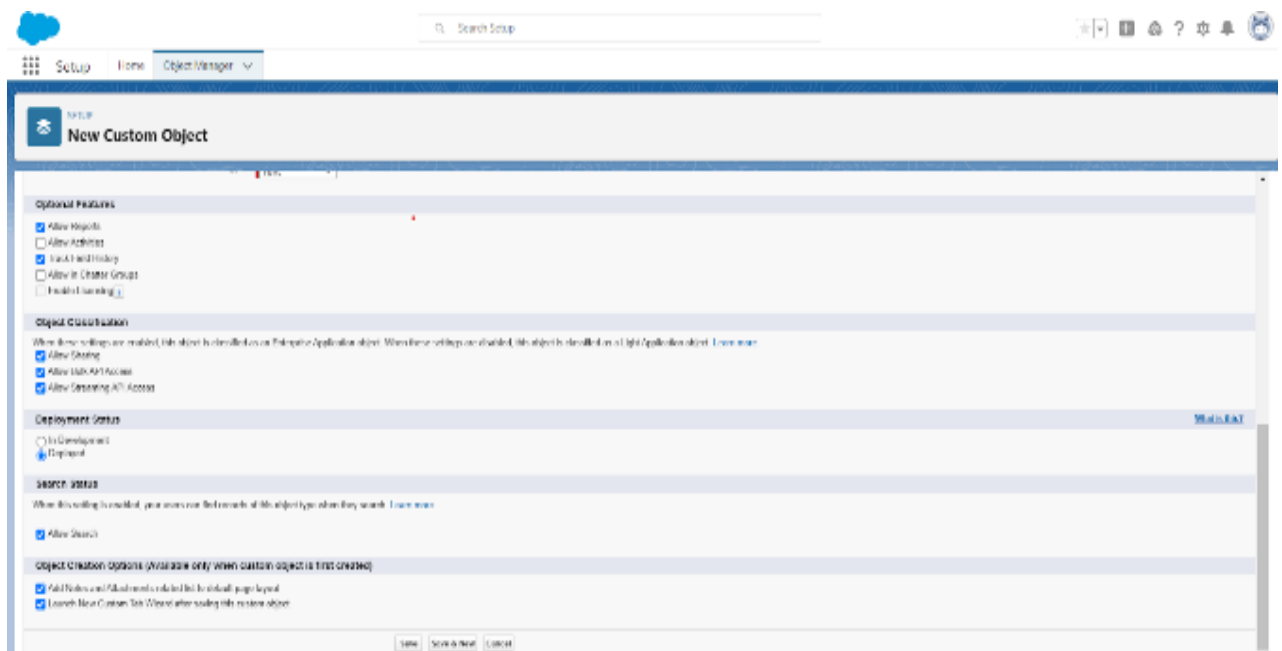
1. Click on the Gear icon >> From **setup** click on **object manager**
- 2.Click **create**, select **custom object**.



3. Fill in the label as "**Customer Detail**".
4. Fill in the plural label as "**Customer Details**".
5. Record name : "**Customer Name**".
6. Select the data type as "**Text**".
7. In the Optional features section, select Allow Reports and Track field history.
8. In the Deployment Status section, ensure **Deployed** is selected.
9. In the search status section select , Allow Search.
10. In the Object creation Options section , select the options.



Add Notes and Attachments related list to default **page layout** Launch New Custom Tab Wizard after saving this custom object
11. Leave everything else as is, and click **Save**.



4)Create A Custom Object For Best Seller :

1. Click on the **Gear Icon** ? From **setup** click on **object manager**
2. Click **create**, select **custom object**.
3. Fill in the label as "**Best Seller**".
4. Fill in the plural label as "**Best Sellers**".
5. Record name : "**Best Seller**"
6. Select the data type as "**Text**".
7. In the Optional Features section, select Allow **Reports** and **Track** Field History.
8. In the Deployment Status section, ensure **Deployed** is selected.
9. In the Search Status section, select **Allow Search**.
10. In the Object Creation Options section, select select these options:
Add Notes and Attachments related list to default **page layout**
Launch New Custom Tab Wizard after saving this custom object
11. Leave everything else as is, and click **Save**

5)Create A Custom Object For New Launches :

1. Click on the **Gear Icon** ? From **setup** click on **object manager**
2. Click **create**, select **custom object**.
3. Fill in the label as "**New Launch**".
4. Fill in the plural label as "**New Launches**".
5. Record name : "**New Launch**"
6. Select the data type as "**Text**".
7. In the Optional Features section, select Allow **Reports** and **Track** Field History.
8. In the Deployment Status section, ensure **Deployed** is selected.
9. In the Search Status section, select **Allow Search**.
10. In the Object Creation Options section, select select these options:
Add Notes and Attachments related list to default **page layout**
Launch New Custom Tab Wizard after saving this custom object
11. Leave everything else as is, and click **Save**

6)Create A Custom Object for Pizzas:

1. Click on the **Gear Icon** >> From **setup** click on **object manager**
2. Click **create**, select **custom object**.
3. Fill in the label as "**New Launch**".
4. Fill in the plural label as "**New Launches**".
5. Record name : "**New Launch**"
6. Select the data type as "**Text**".
7. In the Optional Features section, select Allow **Reports** and **Track** Field.
8. In the Deployment Status section, ensure **Deployed** is selected.
9. In the Search Status section, select **Allow Search**.
10. In the Object Creation Options section, select select these options:
Add Notes and Attachments related list to default **page layout**
Launch New Custom Tab Wizard after saving this custom object
11. Leave everything else as is, and click **Save**.

7) Create A Custom Object For Mode Of Payment:

1. Click on the **Gear Icon** >> From **setup** click on **object manager**
2. Click **create**, select **custom object**.
3. 3. Fill in the label as "**Mode Of Payment**".
4. Fill in the plural label as "**Mode Of Payments**".
5. Record name : "**Mode Of Payment**"
6. Select the data type as "**Text**".
7. In the Optional Features section, select Allow **Reports** and **Track** Field History.
8. In the Deployment Status section, ensure **Deployed** is selected.
9. In the Search Status section, select **Allow Search**.
10. In the Object Creation Options section, select select these options:
Add Notes and Attachments related list to default **page layout**
Launch New Custom Tab Wizard after saving this custom object
11. Leave everything else as is, and click **Save**

Tabs:

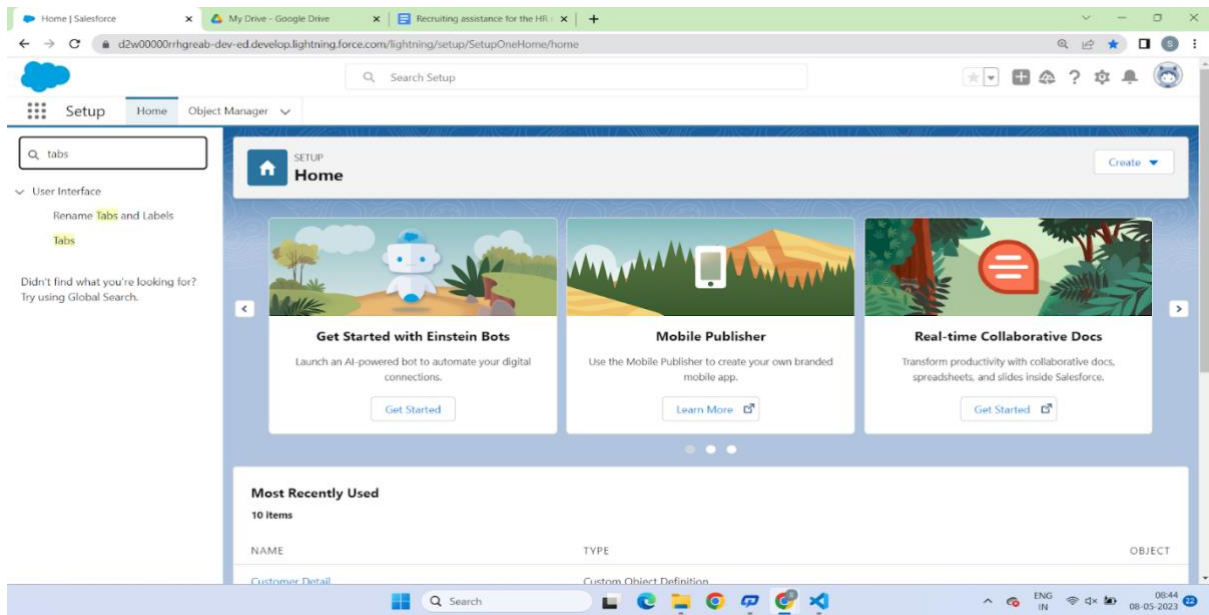
Tabs in Salesforce help users view the information at a glance. It displays the data of objects and other web content in the application.

There are mainly 4 types of tabs:

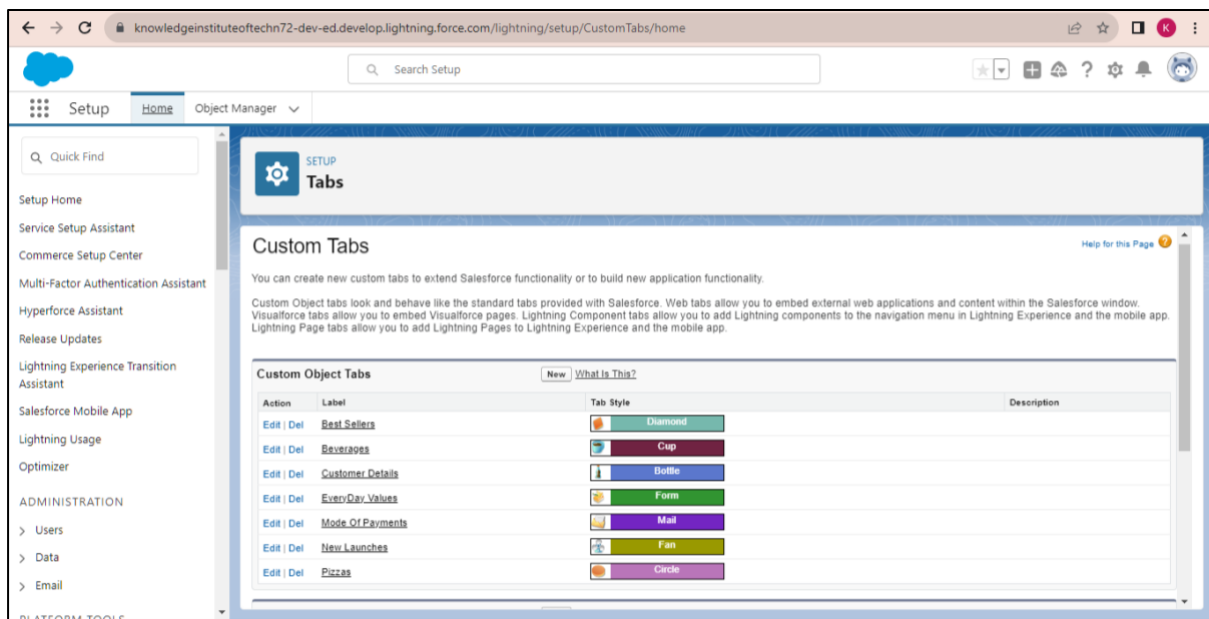
- 1.**Standard Object Tabs:** Standard object tabs display data related to standard objects.
- 2.**Custom Object Tabs:** Custom object tabs display data related to custom objects. These tabs look and function just like standard tabs.
- 3.**Web Tabs:** Web Tabs display any external Web-based application or Web page in a Salesforce tab.
- 4.**Visualforce Tabs:** Visualforce Tabs display data from a Visualforce Page.

1)Creation of EveryDay Value:

1. To Create the tab >> click on the gear Icon >> then click Home button >> In the quick find box search for the Tabs.



3. Then Click on the new button to create tab, in the Tab style you can select whatever you want to select



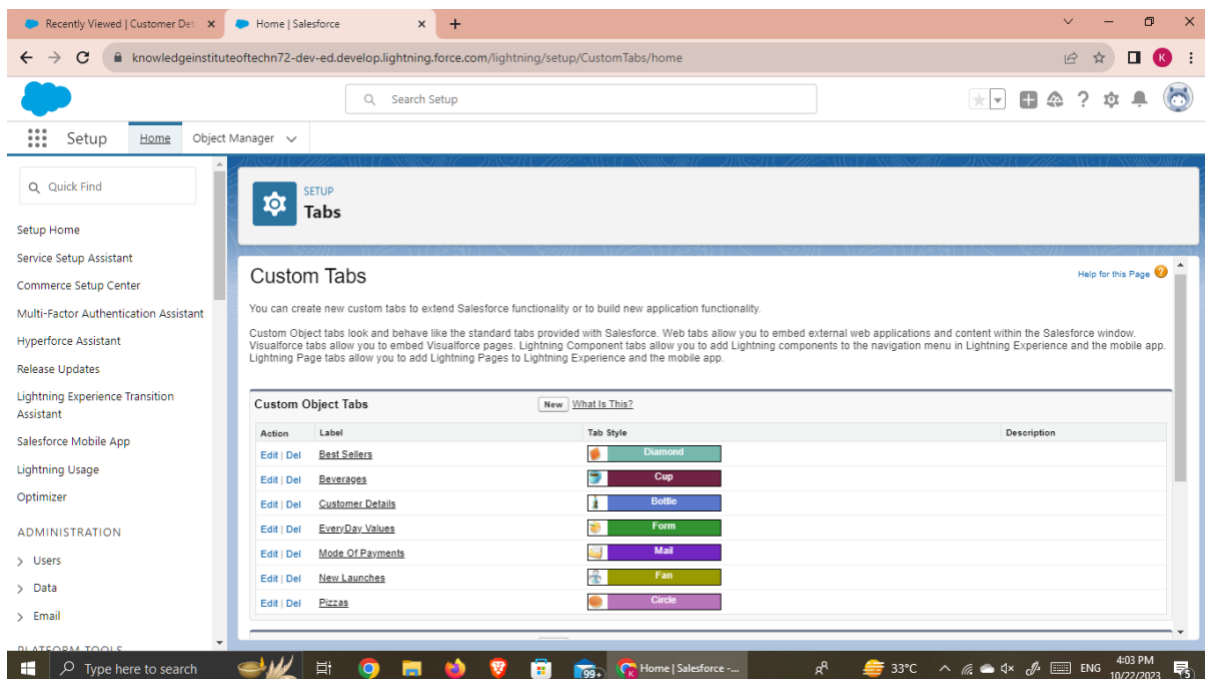
4. Than select the EveryDay Values Object and select the Icon from the search button and click next next and save

2)Creation of Beverages Tab:

1. To Create the tab >> click on the gear Icon >> then click Home button >> In the quick find box search for the Tabs.
2. Then Click on the new button to create tab, in the Tab style you can select whatever you want to select
3. Than select the Beverages Object and select the Icon from the search button and click next next and save

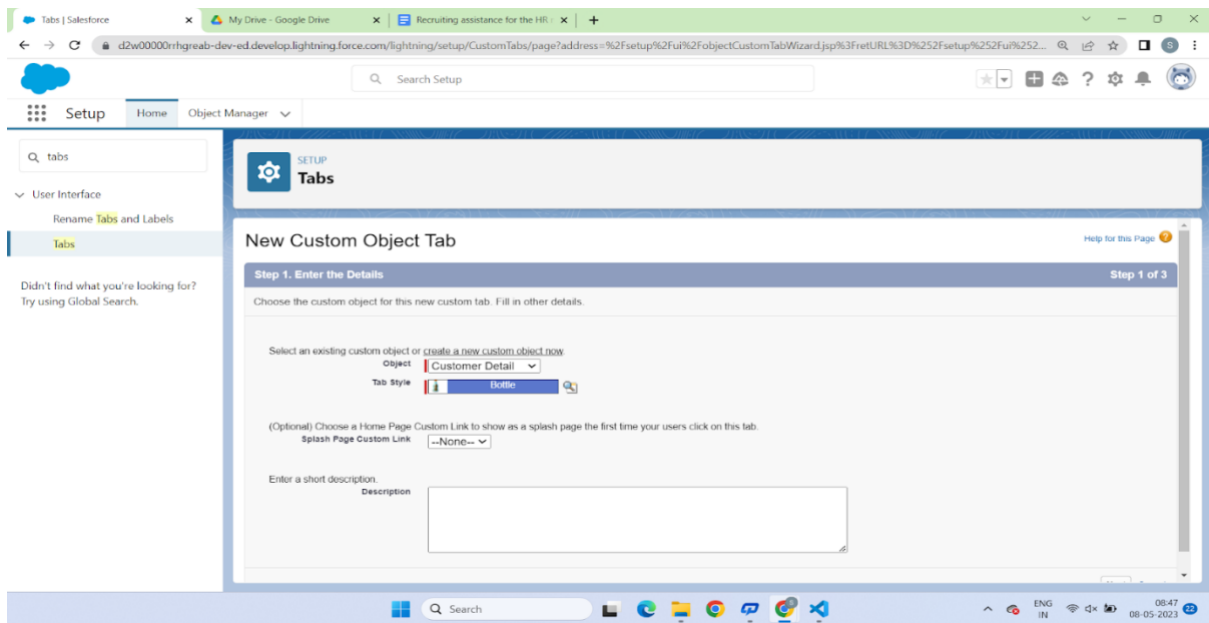
3)Creation of Customer Detail Tab:

1. To Create the tab >> click on the gear Icon >> then click Home button >> In the quick find box search for the Tabs.
2. Then Click on the new button to create tab, in the Tab style you can select whatever you want to select



3. Than select the Customer Detail Object and select the Icon from the

search button and click next next and save



4)Creation of Best sellers Tab:

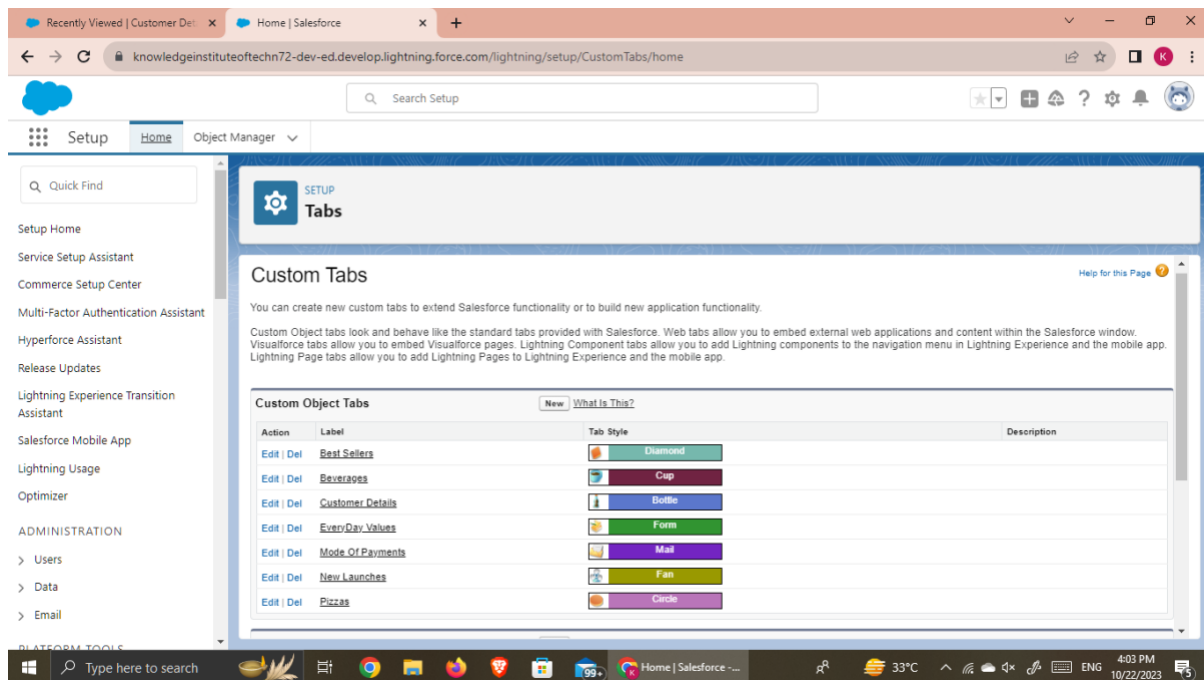
1. To Create the tab >> click on the gear Icon >> then click Home button >> In the quick find box search for the Tabs.
2. Then Click on the new button to create tab, in the Tab style you can select whatever you want to select
3. Than select the New launches Object and select the Icon from the search button and click next next and save

5)Creation of New launches Tab:

1. To Create the tab >> click on the gear Icon >> then click Home button >> In the quick find box search for the Tabs.
2. Then Click on the new button to create tab, in the Tab style you can select whatever you want to select
3. Than select the New launches Object and select the Icon from the search button and click next next and save

6)Creation of Pizza Tab:

1. To Create the tab >> click on the gear Icon >> then click Home button >> In the quick find box search for the Tabs.
2. Then Click on the new button to create tab, in the Tab style you can select whatever you want to select



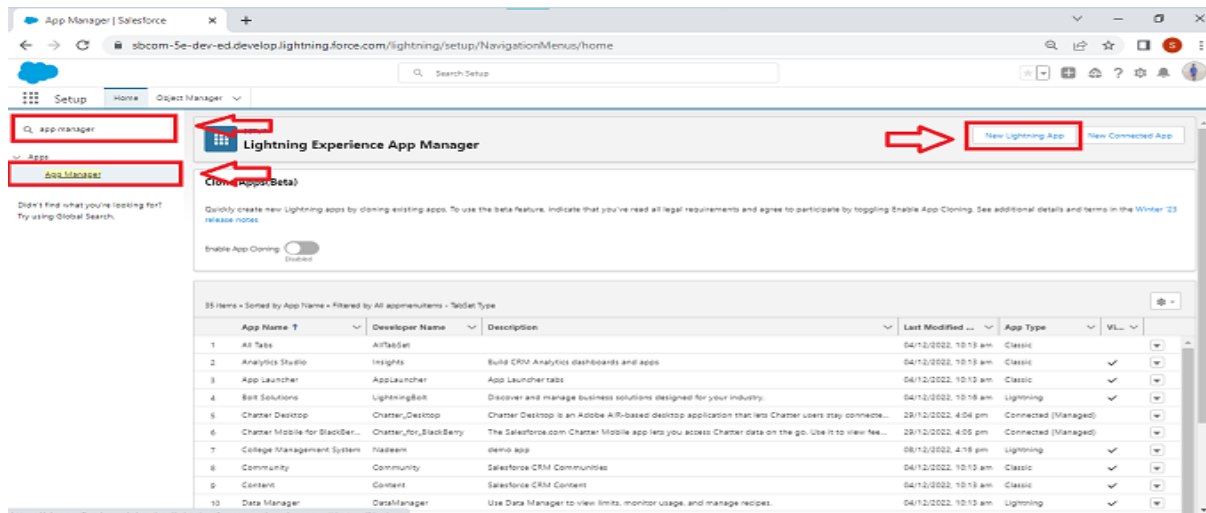
3. Than select the Pizza Object and select the Icon from the search button and click next next and save

7)Creation of Mode of payments:

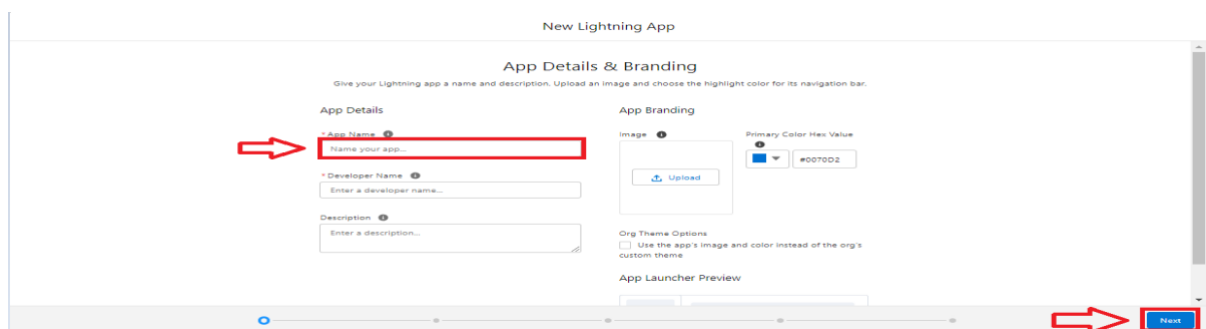
1. To Create the tab >> click on the gear Icon >> then click Home button >> In the quick find box search for the Tabs.
2. Then Click on the new button to create tab, in the Tab style you can select whatever you want to select
3. Than select the Mode of payments Object and select the Icon from the search button and click next next and save

Lightning App:

1. Go to the Gear Icon >> Click on setup >> click on the Home Button >> Go to the quick find box and select the App Manager



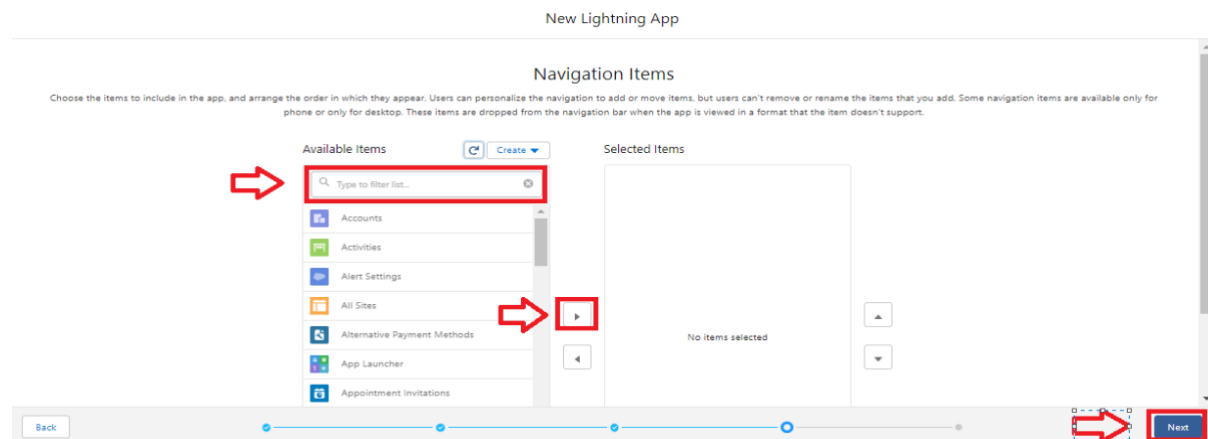
2. Fill the app name as a Pizza Delivery in app details and branding >> Next >> (App option page) keep it as default >> Next 3. Utility Items keep it as default >> Next >> (Add Navigation Items)(add tabs Customer Details, Every Day Values, Beverages, Best Seller, New Launches, pizza and Mode of payment) >> Next >> (Add User Profile) Add System Administrator, Salesforce platform user, Standard User >> Next.



4. To Add Navigation Items: Select the items from the search bar and move it using the arrow button >> Next. select all the tabs which you have created

5. To Add User Profiles: Search profiles in search bar >> click on the

arrow button & select Standard user, standard Platform user & System Admin Profile >> save & finish.



Fields and Relationship:

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also 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

- Standard Fields
- 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,

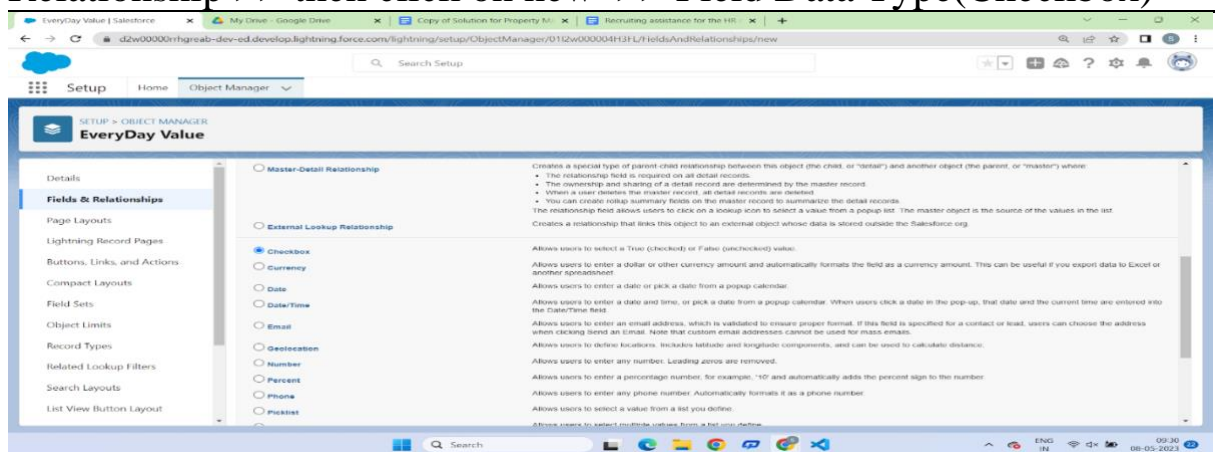
- Created By
- Owner
- Last Modified
- Field Made During object Creation

Custom Fields:

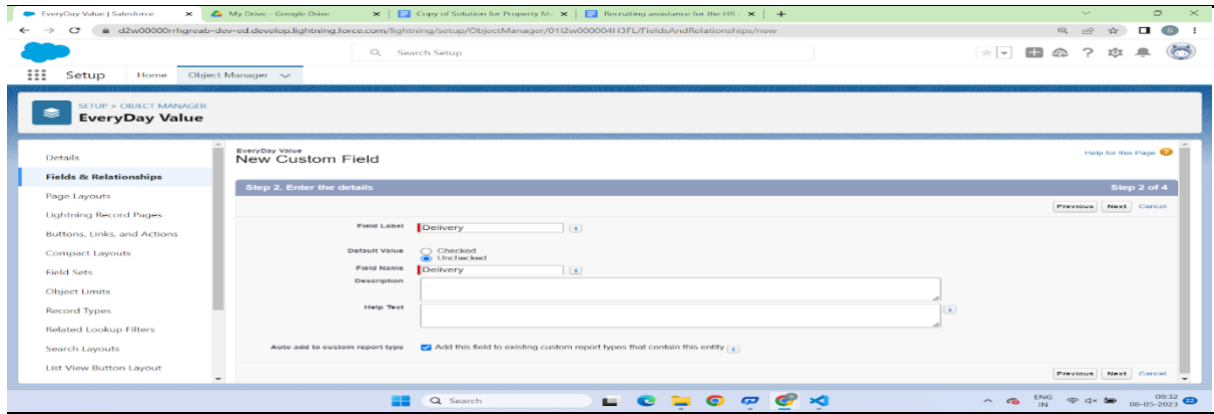
On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organizer or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on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

1) Create The Field EveryDay Value:

1. Go to setup >> click on Object Manager ? type object name in search bar >> click on the object >> EveryDay Value ? Field and Relationship >> then click on new >> Field Data Type(Checkbox)



2. Click On Next >> Field Label >> Delivery >> next >> next >> Save.



3. Follow the above steps and create the Remaining Fields.

- Pickup(Checkbox)
- Select City(Picklist)(New Delhi, Mumbai, Bangalore, Pune, Chennai, Hyderabad, Kolkata, Noida, Agra, AhmadNagar, Ahmedabad, Nasik)
- Select Store(Picklist)(Deolali Camp, Bytco Point, Satpur Nashik, Hitech City, Gachibowli, Indira nagar, Kelambakkam, Noida City)
- Delivery(Checkbox)
- Pickup(Checkbox)

2)Creation of Fields for (Beverages):

Types of Cold Drinks(Picklist)(Mountain Dew Ice, Nagpur Orange, Pepsi 475ml, 7Up 475ml,Mirinda)

3)Create The Field(Customer Detail):

1. Types of Roasted Chicken
2. Add To Cart

4)Creation of Fields for (Best Seller Objects):

(Best Seller obj >> **Size field**)

Crust(Picklist)(New Hand Tossed >> 459,

100% Wheat thin crust >> 524, Cheese Burst >> 578,
Fresh pan Pizza >> 514)

5)Creation of Fields for (New Launches):

1. New Beverages(Picklist)
2. Options

6)Creation of Fields for (Pizza):

Types of Pizzas(PickList)(Paneer Makhani, Margaretha, Onion
Pizza, Tomato pizza, Paneer Chicken, Chicken pizza)

Amount(PickList >> 100, 200, 350, 400, 500, 550,750, 850)

Add On >> (Number Field Data type)

The screenshot shows the Salesforce Object Manager interface for the 'Pizza' object. The 'Fields & Relationships' section is active, displaying a table of fields. The table has columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Add On	Add_On_c	Number(18, 0)		
Amount	Amount__c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Pizza Name	Name	Text(80)		✓
Record Type	RecordTypeId	Record Type		✓
Types of Pizzas	Types_of_Pizzas__c	Picklist		

7)Create Picklist Fields (Mode Of Payment):

1. **Mode Of Payment**(Picklist)(Debit Card, Credit Card, UPI Payment)
2. Master detail Relationship with(Customer Details)

4. USERS & DATA SECURITY

Profile

- A profile is a group/collection of settings and permissions that define what a user can do in salesforce.
- Profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges.
- You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

Types of profiles in salesforce Standard profiles:

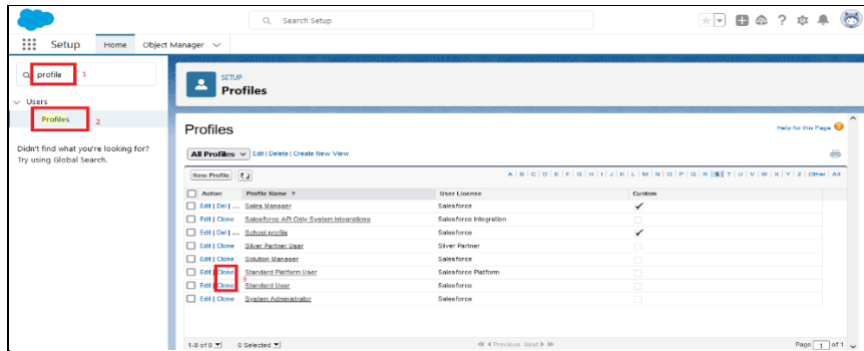
- By default salesforce provide below standard profiles
- We cannot deleted standard ones
- Each of these standard one includes a default set of permissions for all of the standard objects available on the platform

2. Custom Profiles:

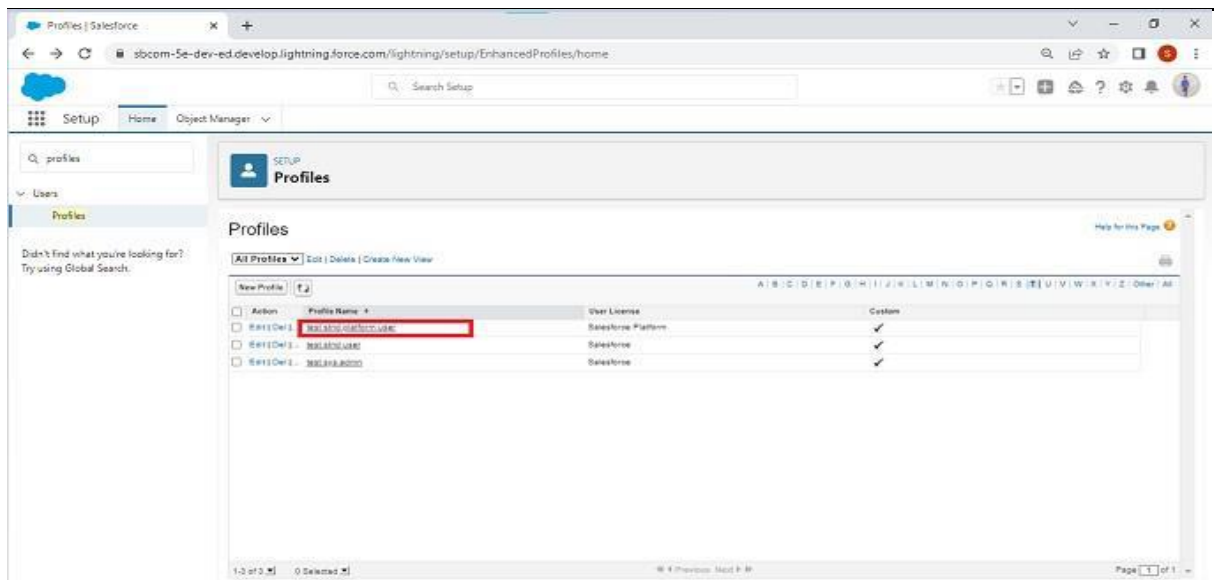
- 1. Custom ones defined by us.
- 2. They can be deleted if there are no users assigned with that particular one

1)Create A Custom Profile

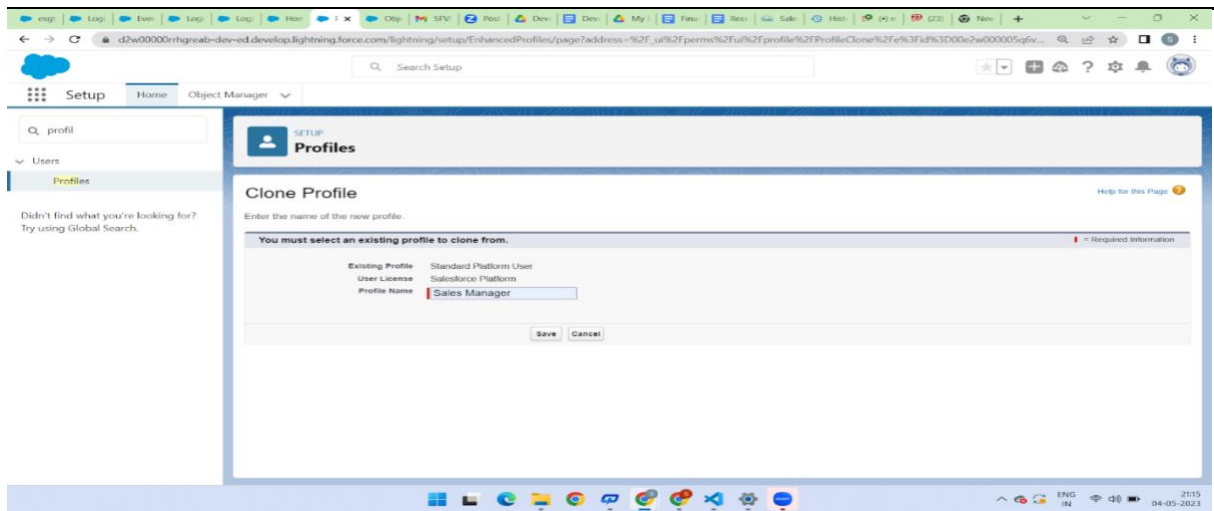
1. Go to setup >> type profiles in quick find box > > click on profiles >> clone the desired profile (standard platform user is pref) and clone that profile



2. Enter a Profile Name(Sales Manager) And click on Save
3. Click on the new created profile



4. While still on the profile page, then click Edit.



6. Similarly clone the standard platform profile and give the name as Sales Executive And give the same access to the as of sales manager.

7. Again clone the standard User profile and this time give all access for the objects, tabs and field And give the profile name as Delivery Person

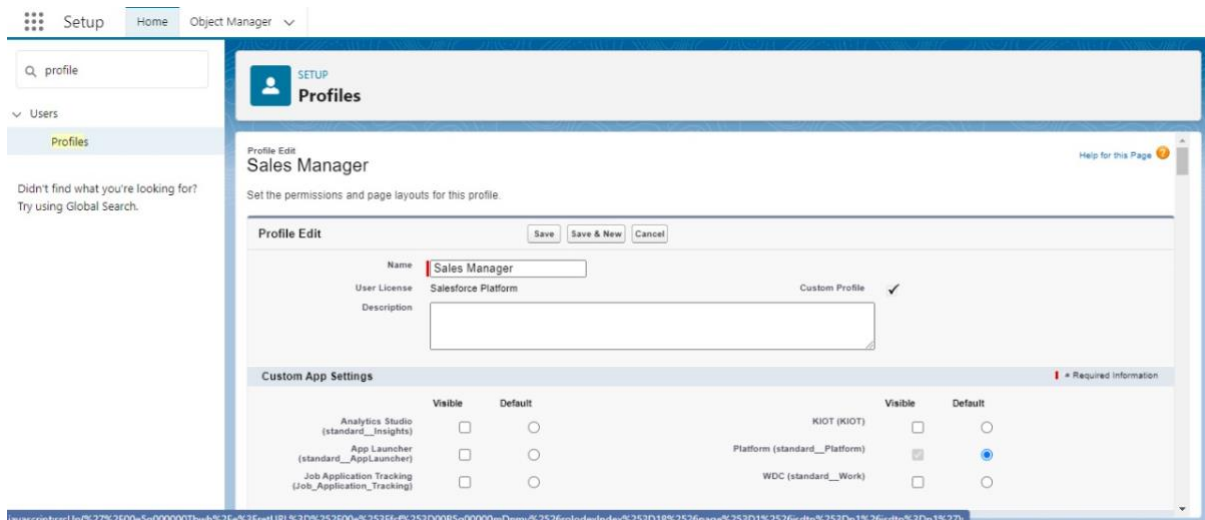
2)Create A Custom Profile-2

1.Create a profile with the profile name as “Sales Manager”.

2.From setup, enter profiles in Quick Find box

3.Select profiles (Standard user).

4.Click clone.

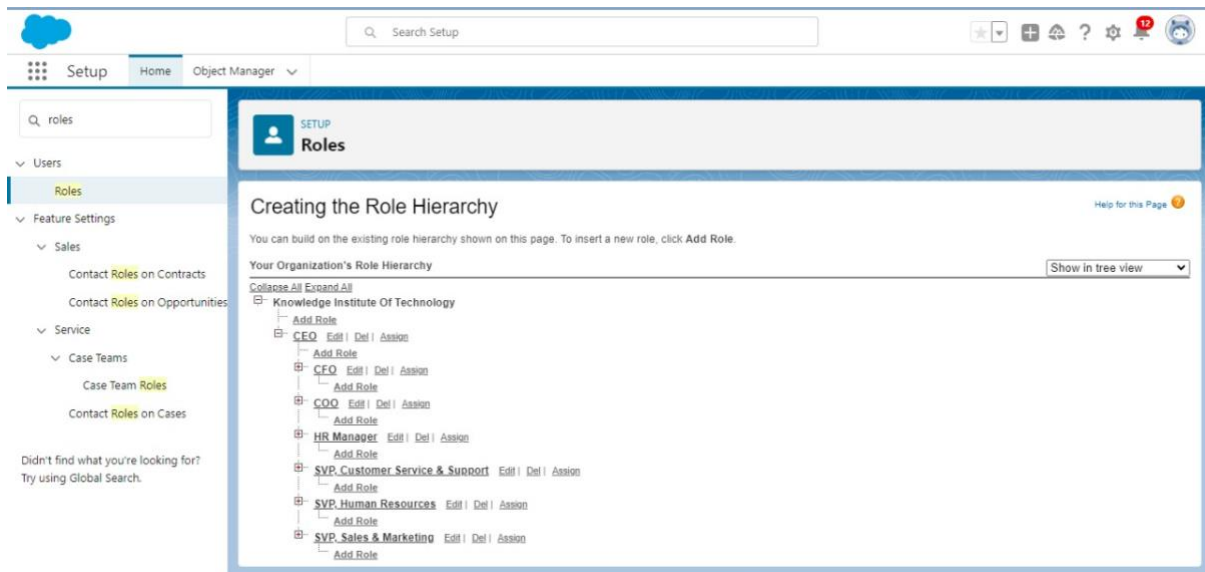


Role

In Salesforce, roles are used to determine which users have access to certain data and functions within the system. They are also used to define the reporting hierarchy within an organization. Users with higher roles have greater access to data and more control over the system

1)Creation of Role

- 1.From the Quick find box search for the role and click on the roles option
- 2.select the set-up roles option
- 3.Below the CEO click on add role and enter the label name as a" HR Manager" and role name will be Automatically populated and click on save.

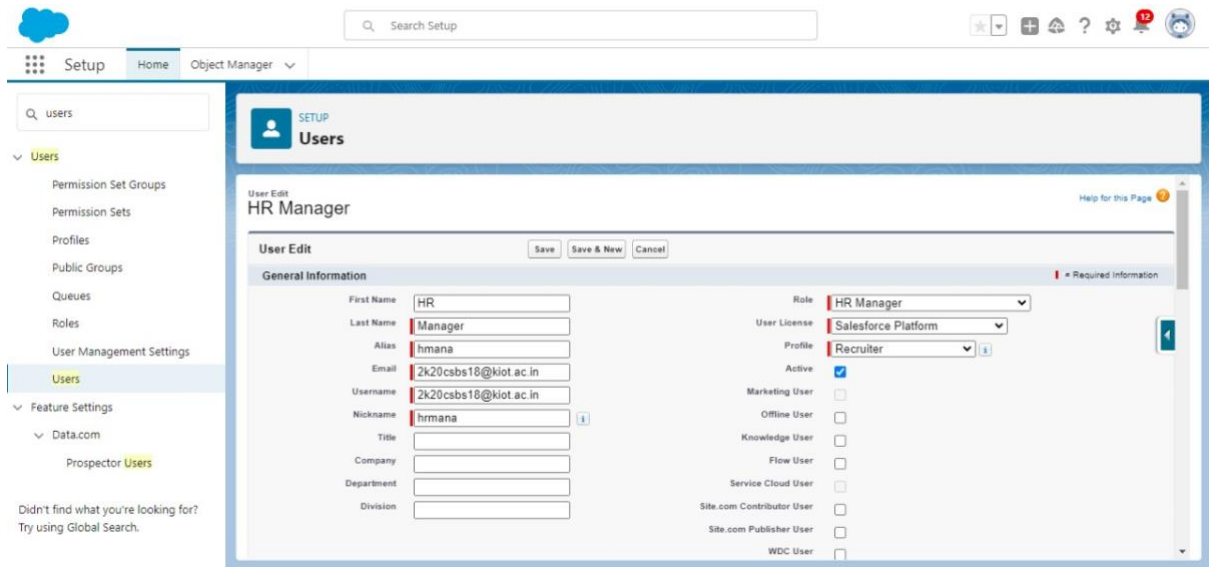


User

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

1)To Create A User

- 1.From Setup, enter Users in the Quick Find box, then select Users.
- 2.Click New User.
- 3.Enter First name as HR and last name as Manager.
- 4.Enter the user's name and email address and a unique username in the form of an email address. By default, the username is the same as the email address.
- 5.Then create a new role HR Manager.
- 6.Select user License as Standard Platform User.
- 7.Select profile (Recruiter).



8. Click save

2) To Create A User

1. From Setup, enter Users in the Quick Find box, then select Users.

2. Click New User.

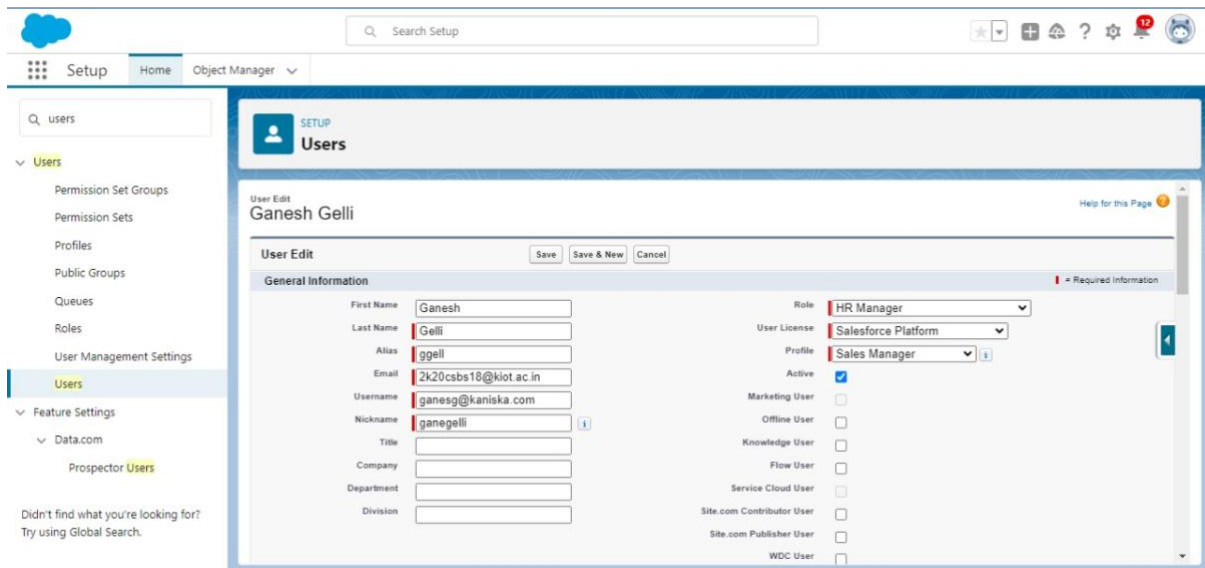
3. Enter First name as Ganesh and last name as Gili.

4. Enter the user's name and email address and a unique username in the form of an email address. By default, the username is the same as the email address.

5. Then create a new role HR Manager.

6. Select user License as Standard Platform User.

7. Select profile (Sales Manager).



8. Click save

Sharing Rules

Sharing rules help users to share records based on conditions. It is basically created for objects whose organization-wide defaults (OWD) are set to public read-only or private because sharing rules can only extend the access and not restrict it.

Types of sharing rules,

1. Owner-based Sharing Rules
2. Criteria-based Sharing Rules

1) Create A Sharing Rule

1. Go to Sharing Settings, which can be found under the Quick Find section.
2. Scroll down and find the candidate object where a sharing rule needs to be added, and then click on New to create a new sharing rule.
3. Add the label of the sharing rule you want to make.
4. Select your rule type based on the criteria.
5. Select the field can join immediately check field from the candidate object.

6. Select the State as equal and value is Rajasthan.

7. And in selecting the users to share with the section select roles and in that select Hr Manager.

8. And in the section of select the level of access for the users give the access Read/Write.

9. And save the rule.

Q shar

Security

Guest User **Sharing Rule Access** Report

Sharing Settings

Didn't find what you're looking for? Try using Global Search.

SETUP
Sharing Settings

Note: "Roles and subordinates" includes all users in a role, and the roles below that role.
You can use sharing rules only to grant wider access to data, not to restrict access.

Label:

Rule Name:

Description:

Step 1: Select your rule type

Criteria	Field	Operator	Value
	State	equals	Rajasthan
	--None--	--None--	
	--None--	--None--	
	--None--	--None--	
	--None--	--None--	

Additional Options

☒ Include records owned by users who can't have an assigned role

Share with Role: HR Manager

Access Level: **Read/Write**

Created By: Kamala Padmanaban, 18/10/2023, 7:13 pm

Modified By: Kamala Padmanaban, 18/10/2023, 7:13 pm

Save Cancel

2)Activity 2

Create a Sharing Rule to Share the records of Job Application to Hr Manager with the Access of Read/Write.

Create A Sharing Rule

1. Go to Sharing Settings, which can be found under the Quick Find section.

2. Scroll down and find the Job Application object where a sharing rule needs to be added, and then click on New to create a new sharing rule.

3. Add the label of the sharing rule you want to make.

4. Select your rule type based on the criteria.

5. Select the field can join immediately check field from the Job Application object.

6.Job application number contains some number.

7.And in selecting the users to share with the section select roles and in that select Hr Manager.

8.And in the section of select the level of access for the users give the access Read/Write.

9.And save the rule.

The screenshot shows the 'Sharing Settings' section of a system setup interface. The main heading is 'Job Application Sharing Rule'. Below this, there is a description: 'Use sharing rules to make automatic exceptions to your organization-wide sharing settings for defined sets of users. Note: "Roles and subordinates" includes all users in a role, and the roles below that role. You can use sharing rules only to grant wider access to data, not to restrict access.'

The configuration form includes the following fields:

- Label:** Job Application
- Rule Name:** Job_Application
- Description:** (empty text box)

Step 1: Select your rule type

Criteria	Field	Operator	Value	
	Job Application Number	contains	Some number	AND
	--None--	--None--		AND
	--None--	--None--		AND
	--None--	--None--		AND
	--None--	--None--		

[Add Filter Logic...](#)

Additional Options
☐ Include records owned by users who can't have an assigned role

Share with
Access Level: Read/Write

Created By: Kanaka Padmanaban, 18/10/2023, 7:30 pm
Modified By: Kanaka Padmanaban, 18/10/2023, 7:30 pm

5.AUTOMATION

Apex Trigger

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions

before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

- insert
- update
- delete
- merge
- upsert
- undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

Before Trigger: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

After Trigger: This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In

other words, the after trigger makes changes to the value from the data inserted in some other record.

Activity- 1

Use Case: This Trigger works for the Pizza Object where the scenario is like whenever the customer is selecting the Pizza whether it is veg Pizza or Non-veg Pizza According to the selection of Pizza The Amount will be reflected in the “Amount” Field.

Trigger:

```

1 * trigger PizzaApplication on Pizza__c (before insert, before update) {
2 *   if(trigger.isbefore && trigger.isinsert){
3
4       PizzaApplicationHandler.pizza(trigger.new);
5       //UpdatePizzaApplicationHandler.updatePizza(trigger.new);
6   }
7 *   if(trigger.isupdate){
8       PizzaApplicationHandler.pizza(trigger.new);
9   }
10 }
11
12 }

```

Trigger Handler:

```

1 * public class PizzaApplicationHandler {
2 *   public static void pizza(List<Pizza__c> pizzaRec){
3 *     for ( Pizza__c pz : pizzaRec){
4 *       if(pz.Pizza__c.contains('Margherita') && pz.Pizza__c.contains('Paneer Naan')) ){
5 *         pz.Amount__c = '500';
6 *       }
7 *       if (pz.Pizza__c.contains('Paneer Naan'))){
8 *         pz.Amount__c = '350';
9 *       }
10 *       if(pz.Pizza__c == 'Margherita'){
11 *         pz.Amount__c = '200';
12 *       }
13 *       if(pz.Pizza__c == 'Tomato Pizza'){
14 *         pz.Amount__c = '180';
15 *       }
16 *       if(pz.Pizza__c == 'Onion Pizza'){
17 *         pz.Amount__c = '180';
18 *       }
19 *       if(pz.Pizza__c.contains('Margherita') && pz.Pizza__c.contains('Paneer Naan') && pz.Pizza__c.contains('Tomato Pizza') && pz.Pizza__c.contains('Onion Pizza')){
20 *         pz.Amount__c = '750';
21 *       }
22 *     }
23 *   }
24 * }
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *

```

Trigger Code:

trigger PizzaApplication on Pizza__c (before insert, before update) {

```

if(trigger.isbefore && trigger.isinsert){

    PizzaApplicationHandler.pizza(trigger.new);
    //UpdatePizzaApplicationHandler.updatePizza(trigger.new);
}
if(trigger.isupdate){
    PizzaApplicationHandler.pizza(trigger.new);
}
}

```

Trigger Handler

```

public class PizzaApplicationHandler {
    public static void pizza(List<pizza__c> pizzaRec){
        for ( Pizza__c pz : pizzaRec){
            if(pz.Pizza__c.contains('Margretha') &&
pz.Pizza__c.contains('Paneer Makhani') ){
                pz.Amount__c = '550';
            }
            if( pz.Pizza__c.contains('Paneer Makhani')){
                pz.Amount__c = '350';
            }
            if(pz.Pizza__c == 'Margretha'){
                pz.Amount__c = '200';
            }
            if(pz.Pizza__c == 'Tomato Pizza'){
                pz.Amount__c = '100';
            }
            if(pz.Pizza__c == 'Onion Pizza'){
                pz.Amount__c = '100';
            }

            if(pz.Pizza__c.contains('Margretha') && pz.Pizza__c.contains('Paneer
Makhani') && p z.Pizza__c.contains('Tomato Pizza') &&
pz.Pizza__c.contains('Onion Pizza') ){
                pz.Amount__c = '750';
            }
        }
    }
}

```



```

if(pz.Pizza__c.contains('Margretha') && pz.Pizza__c.contains('Paneer
Makhani') && pz.Pizza__c.contains('Tomato Pizza'))
    pz.Amount__c = '750';
}
if(pz.Pizza__c == 'Chicken Pizza'){
    pz.Amount__c = '400';
}
if(pz.Pizza__c == 'Paneer Chicken'){
    pz.Amount__c = '400';
}
if(pz.Pizza__c.contains('Paneer Chicken') &&
pz.Pizza__c.contains('Chicken Pizza') ){
    pz.Amount__c = '800';
}
    if(pz.Pizza__c.contains('Paneer Chicken') &&
pz.Pizza__c.contains('Paneer Makhani') ){
        pz.Amount__c = '750';
    }
    if(pz.Pizza__c.contains('Paneer Chicken') &&
pz.Pizza__c.contains('Margretha') ){
        pz.Amount__c = '750';
    }
if(pz.Pizza__c.contains('Paneer Chicken') &&
pz.Pizza__c.contains('Tomato Pizza') ){
    pz.Amount__c = '500';
}
if(pz.Pizza__c.contains('Paneer Chicken') && pz.Pizza__c.contains('Onion
Pizza') ){
    pz.Amount__c = '500';
}
}
}
}
}

```

Schedule Apex

Scheduled Apex in Salesforce is a feature that allows you to schedule the execution of Apex classes to run at specified times. This capability is useful for automating repetitive tasks, such as data updates or calculations, on a regular basis. To create a scheduled Apex job, you need to define an Apex class that implements the Schedulable interface. This interface requires you to implement a single method called execute, which contains the logic you want to run at the scheduled time.

```
public class MyScheduledClass implements Schedulable {  
    public void execute(SchedulableContext context) {  
    }  
}
```

To schedule the execution of this class, you can use the Salesforce Developer Console or the Salesforce user interface. Here's an example of scheduling this class to run every day at 2 PM.

```
String cronExp = '0 0 14 * * ?'; // Cron expression for 2 PM every  
day
```

```
System.schedule('My Scheduled Job', cronExp, new  
MyScheduledClass());
```

In the example above, System.schedule is used to create a new scheduled job. The first parameter is the name of the job, the second parameter is the cron expression that defines the schedule, and the third parameter is the instance of the Apex class that should be executed at the scheduled time.

Note that the cron expression follows a specific syntax that allows you to define complex schedules with minute, hour, day, month, and day-of-week precision.

Once the scheduled Apex job is created, it will run automatically according to the specified schedule. You can monitor the execution

and view any debug logs generated by the scheduled job in the Salesforce user interface

Schedule Apex For Frequently Visited Customer:

```
1 public class PizzaDiscountScheduler implements Schedulable {
2     public void execute(SchedulableContext sc) {
3         // Logic for sending the email
4         // if (System.now() == System.DayOfWeek.Sunday) {
5             List<Customer_Detail__c> pz = new List<Customer_Detail__c>();
6             String s='gmail.com';
7             for(Customer_Detail__c c:pz)
8             {
9                 if(c.Email__c.contains(s))
10                {
11                    system.debug('haiiiii');
12                }
13            }
14
15            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
16            email.setToAddresses(new List<String>{'user@example.com'});
17            email.setSubject('Special Sunday Discount');
18            email.setPlainTextBody('Enjoy a 20% discount on all pizzas today!');
19            Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
20        }
21    }
22 }
```

```
public class PizzaDiscountScheduler implements Schedulable {
    public void execute(SchedulableContext sc) {
        // Logic for sending the email
        // if (System.now() == System.DayOfWeek.Sunday) {
        List<Customer_Detail__c> pz = new List<Customer_Detail__c>();
        String s='gmail.com';
        for(Customer_Detail__c c:pz)
        {
            if(c.Email__c.contains(s))
            {
                system.debug('haiiiii');
            }
        }

        Messaging.SingleEmailMessage email = new
        Messaging.SingleEmailMessage();
        email.setToAddresses(new
        List<String>{'user@example.com'});
        email.setSubject('Special Sunday Discount');
        email.setPlainTextBody('Enjoy a 20% discount on all
        pizzas today!');
        Messaging.sendEmail(new
        List<Messaging.SingleEmailMessage>{email});
    }
}
```

Activity-2

For Making the Schedule to send Mail To the Customer Follow the steps below:

1. Click on the **Gear Icon** >> Go to the **Home Tab** >> In the Quick Find Box >> Search for **Apex Class**

2. Click on the **Schedule Apex** >> Give Job Name As >> **Schedule Pizza**.

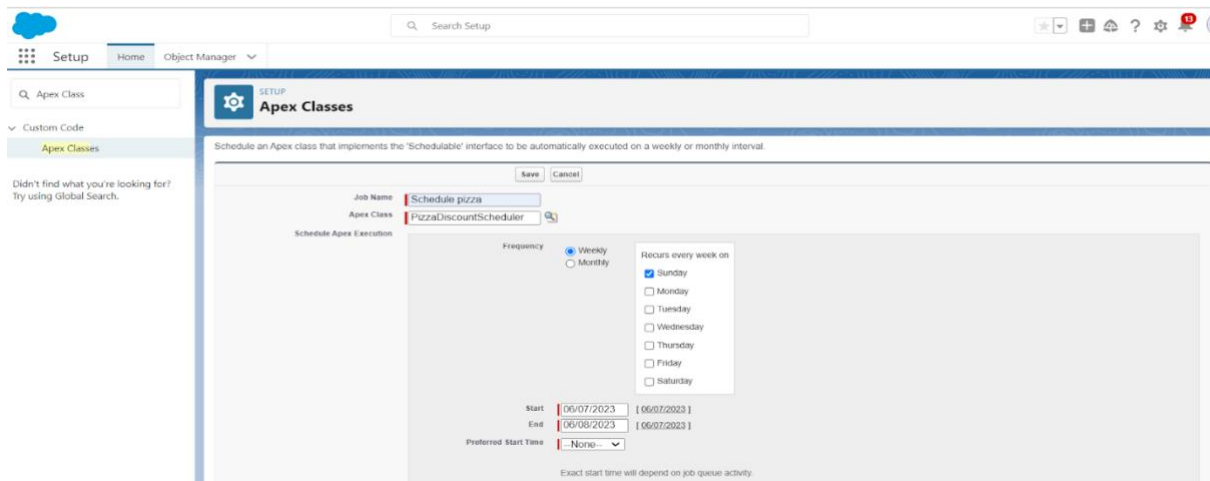
Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	AttendeeEventsService		56.0	Active	969	Shivam Upadhye: 05/05/2023, 12:25 pm	<input type="checkbox"/>
Edit Del Security	ChangePasswordController		57.0	Active	376	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>
Edit Del	ChangePasswordControllerTest		57.0	Active	480	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>
Edit Del Security	CommunitiesLandingController		57.0	Active	203	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>
Edit Del	CommunitiesLandingControllerTest		57.0	Active	502	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>
Edit Del Security	CommunitiesLoginController		57.0	Active	393	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>
Edit Del	CommunitiesLoginControllerTest		57.0	Active	320	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>
Edit Del Security	CommunitiesSelfRegConfirmController		57.0	Active	118	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>
Edit Del	CommunitiesSelfRegConfirmControllerTest		57.0	Active	282	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>
Edit Del Security	CommunitiesSelfRegController		57.0	Active	2,269	Shivam Upadhye: 08/05/2023, 10:08 am	<input type="checkbox"/>

3. Click on **Apex Class Lookup** >> Select **PizzaDiscountScheduler** In Recently Viewed Apex Class

Name	Namespace Prefix	Api Version
PizzaDiscountScheduler		58

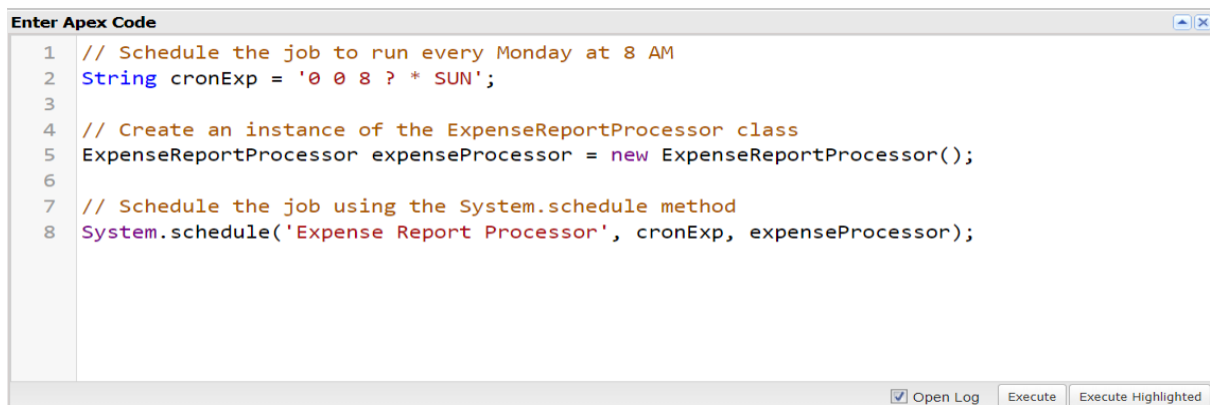
Copyright © 2000-2023 salesforce.com, inc. All rights reserved.

4. In Frequency Click on the **Weekly Radio** button You can Select the **Start Date** and **Enddate** As Per You Requirement and then Click on the **Save Button**.



Alternate Option

6. Click on the **debug** besides file >> Click on the **Open Execute Anonymous Windows** Execute the below code .



```
// Schedule the job to run every Monday at 8 AM
String cronExp = '0 0 8 ? * SUN';
```

```
// Create an instance of the ExpenseReportProcessor class
ExpenseReportProcessor expenseProcessor = new
ExpenseReportProcessor();
```

```
// Schedule the job using the System.schedule method
System.schedule('Expense Report Processor', cronExp, expenseProcessor);
```

6.REPORTS & DASHBOARD

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

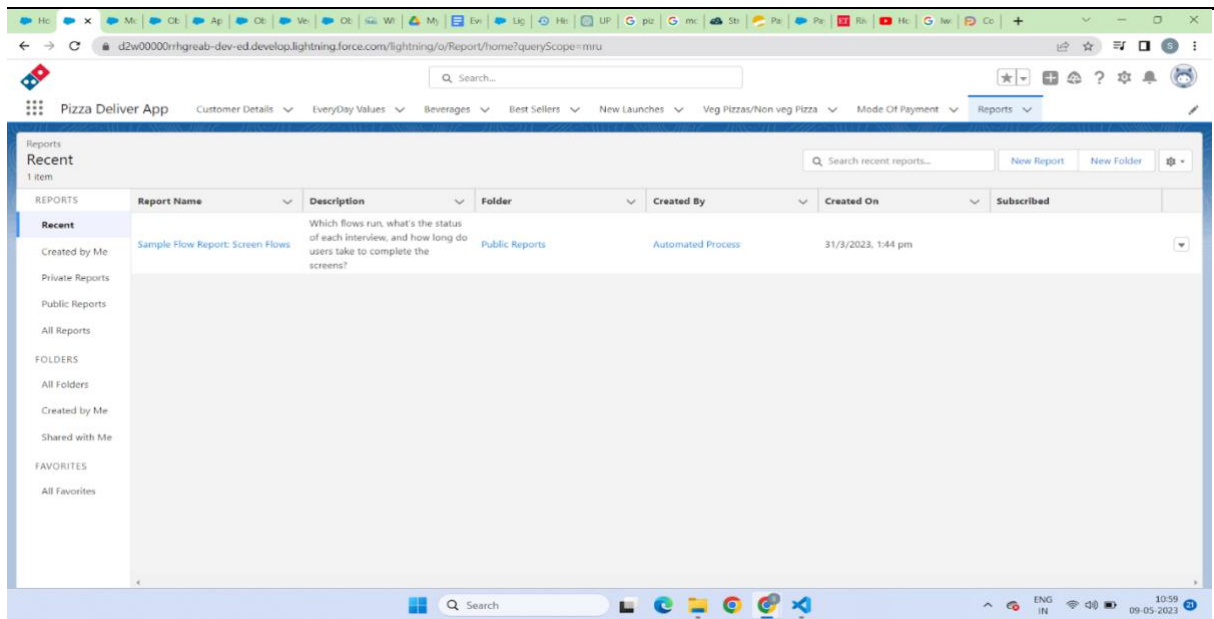
Types of Reports in Salesforce

- Tabular
- Summary
- Matrix
- Joined Reports

1)Create A Report

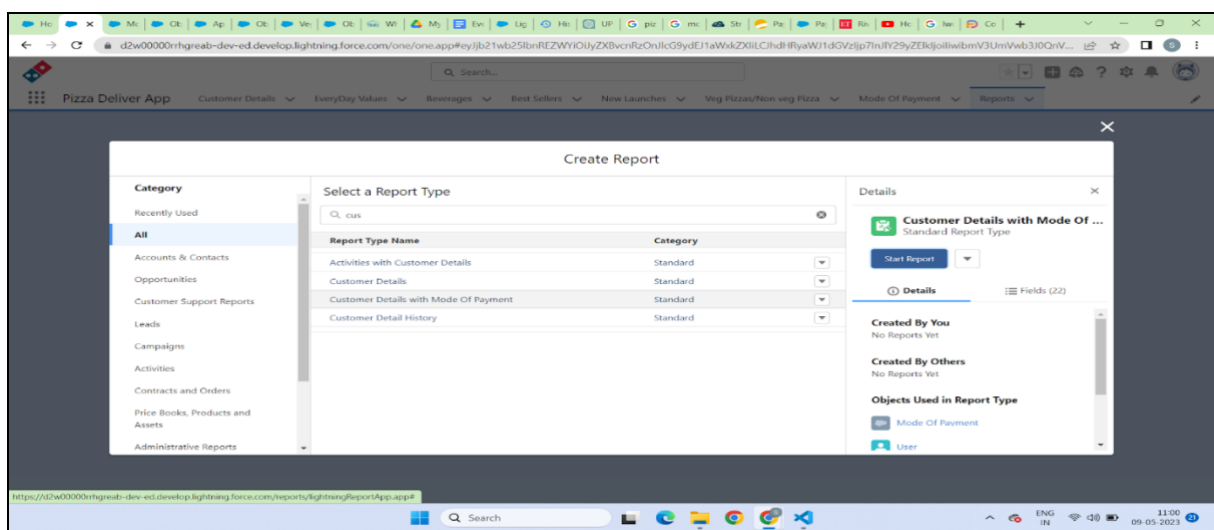
1. Go to the gear icon >> click on the setup >> click on the home button >> in the the quick fin box search for app manager select the App and go to the navigation >> There select the Report and include in the app >> than go to App There you will find the Tab

2. Click New Report? create folder >> Create the Report >> Select the Customer with Mode Of Payment Object Mode of Payment in the columns >> Customer detail:Created Date >> columns >> mode of payment:Id and in the rows >> Mode of payment >> Mode of payment

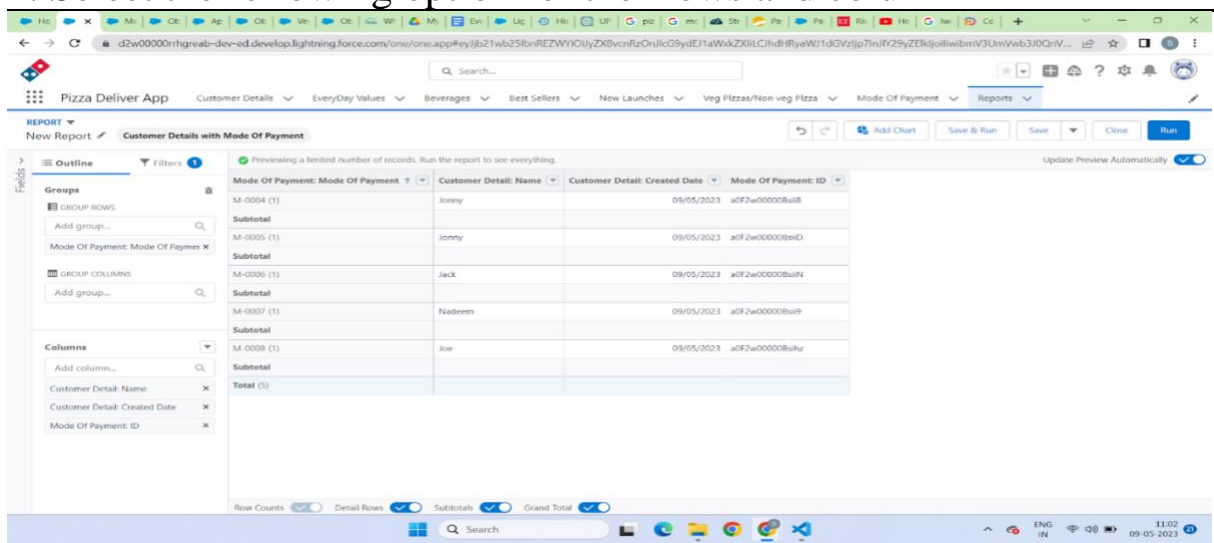


Select Start Report:

3. For to the reports and select the >> customer details with mode of payment



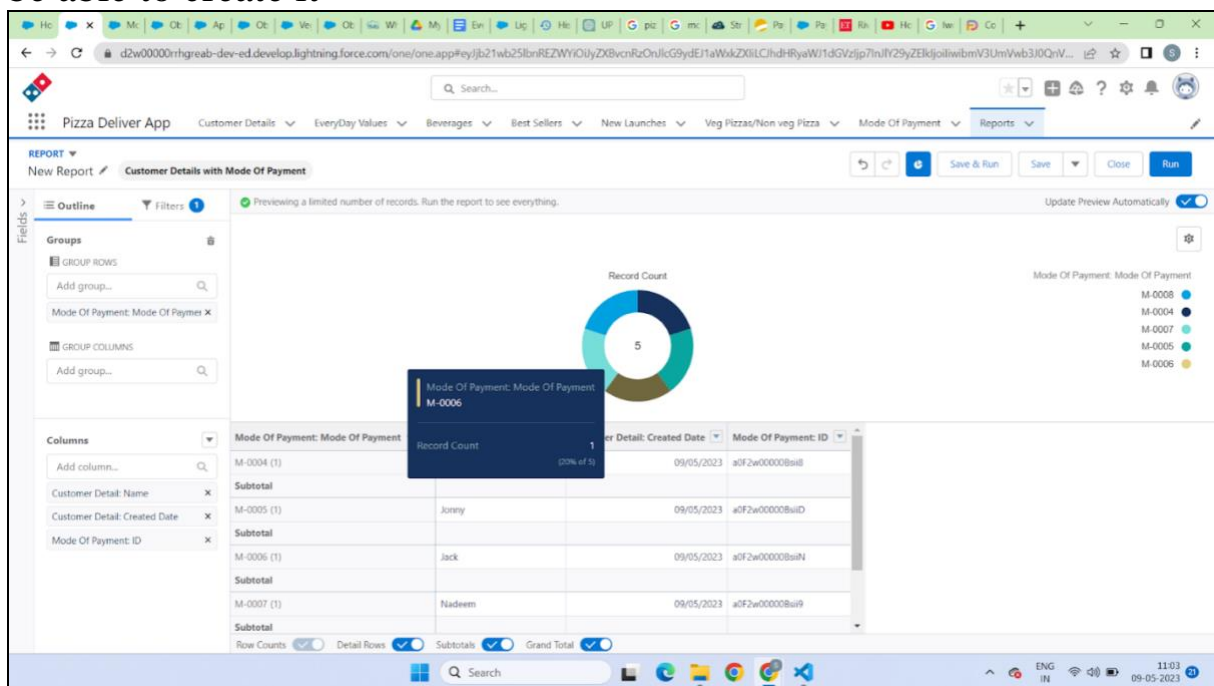
4. Select the following option for the Rows and column



The screenshot shows the 'Pizza Deliver App' interface. The 'Reports' menu is open, and a report titled 'Customer Details with Mode Of Payment' is selected. The report is displayed in a table format with the following columns: Mode Of Payment, Customer Detail: Name, Customer Detail: Created Date, and Mode Of Payment: ID. The table is grouped by Mode Of Payment, with subtotals for each group. The 'Mode Of Payment' column is set to 'Mode Of Payment' and the 'Customer Detail: Name' column is set to 'Customer Detail: Name'.

Mode Of Payment	Customer Detail: Name	Customer Detail: Created Date	Mode Of Payment: ID
M-0004 (1)	Jonny	09/05/2023	a0f2a000008u08
Subtotal			
M-0005 (1)	Jonny	09/05/2023	a0f2a000008u0D
Subtotal			
M-0006 (1)	Jack	09/05/2023	a0f2a000008u04
Subtotal			
M-0007 (1)	Nadeem	09/05/2023	a0f2a000008u09
Subtotal			
M-0008 (1)	Joe	09/05/2023	a0f2a000008u0z
Subtotal			
Total (5)			

5. You can create the chart as so just click on the chart and you will be able to create it



The screenshot shows the 'Pizza Deliver App' interface. The 'Reports' menu is open, and a report titled 'Customer Details with Mode Of Payment' is selected. The report is displayed in a table format with the following columns: Mode Of Payment, Customer Detail: Name, Customer Detail: Created Date, and Mode Of Payment: ID. A donut chart is overlaid on the table, showing the record count for each Mode Of Payment. The chart is titled 'Record Count' and shows a total of 5 records. The 'Mode Of Payment' column is set to 'Mode Of Payment' and the 'Customer Detail: Name' column is set to 'Customer Detail: Name'.

Mode Of Payment	Customer Detail: Name	Customer Detail: Created Date	Mode Of Payment: ID
M-0004 (1)	Jonny	09/05/2023	a0f2a000008u08
Subtotal			
M-0005 (1)	Jonny	09/05/2023	a0f2a000008u0D
Subtotal			
M-0006 (1)	Jack	09/05/2023	a0f2a000008u04
Subtotal			
M-0007 (1)	Nadeem	09/05/2023	a0f2a000008u09
Subtotal			
M-0008 (1)	Joe	09/05/2023	a0f2a000008u0z
Subtotal			
Total (5)			

GitHub & Project Video Demo Link

Project Video Link :

<https://drive.google.com/file/d/141SnbiG3JBrCiAfXIFCGtrkGWPgGFAM1/view?usp=drivesdk>

Github Link:

<https://github.com/KrishnarajS0010/NaanMudhalvan-Salesforce-Krishnaraj-S>