Salesforce VirtualInternship Program

SmartInternz

A CRM APPLICATION FOR WHOLESALE RICE MILL

NAME : MAHAPATHRO DEDEEPYA

E-MAIL : 322103312111@gvpce.ac.in

COLLEGE NAME : GAYATRI VIDYA PARISHAD

COLLEGE OF ENGINEERING

Project Title : A CRM APPLICATION FOR WHOLESALE RICE MILL

1. Project Overview

This project focuses on the development of a CRM Application for Wholesale Rice Mills using Salesforce to streamline and optimize daily operations, improve data accuracy, and provide actionable insights. Designed to meet the specific needs of rice mill owners, this application automates key processes such as tracking daily rice production, sales data, inventory management, and generating real-time reports.

The primary challenge addressed by this project is the manual handling of critical business operations, which often leads to inefficiencies and data inaccuracies. By leveraging Salesforce's robust CRM platform, the project delivers a comprehensive, user-friendly solution that ensures efficient resource management, enhanced customer relationships, and seamless reporting.

Through this project, the rice mill business aims to achieve:

- **Operational Excellence**: Automating routine processes to save time and reduce errors.
- **Data-Driven Decision Making**: Providing owners with detailed, real-time insights into production, sales, and customer trends.
- **Scalability and Efficiency**: Supporting long-term growth with a flexible, secure, and scalable solution.

2. Objectives

Business Goals:

1. **Streamlining Operations**: Automating daily processes such as rice production tracking, inventory updates, and sales recording.
2. **Improved Decision-Making**: Delivering detailed reports and dashboards for real-time analytics, enabling better resource allocation and strategic planning.
3. **Enhancing Customer Relationships**: Providing personalized insights into customer preferences and purchasing behavior.
4. **Ensuring Data Security**: Implementing role-based access controls to restrict sensitive information to authorized users.

Specific Outcomes:

- A centralized platform to monitor and manage rice mill operations effectively.
- Real-time automated reports on daily production, sales, and revenue trends.
- Reduction of manual errors in data entry and calculations.
- User-friendly dashboards to visualize performance metrics and insights.

3. Salesforce Key Features and Concepts Utilized

1. Reports and Dashboards:

■ Automated generation of daily, weekly, and monthly reports on rice production, sales, inventory levels, and revenue.

■ Dashboards display critical metrics such as most purchased rice types, top customers, and revenue trends.

2. Rollup Summary Fields:

■ Used to summarize data from child records to parent records in master-detail relationships.

■ Examples:

○ Total rice supplied by each supplier.

○ Total sales revenue generated from specific rice types.

3. Cross-Object Formula Fields:

■ Enables calculations across related objects.

■ Example: Total payment owed to suppliers calculated using Quantity of Rice x Price per Kilogram

4. Validation Rules:

■ Implements logic to ensure data accuracy and completeness.

■ Example: The ISBLANK formula prevents saving records with missing mandatory f ields, such as rice quantity or customer details, and displays error messages to guide users.

5. Permission Sets and Organization Wide Defaults (OWD):

■ Configures access levels based on roles:

○ Owner: Complete access to all records, including employees and workers.

○ Employer: Access restricted to worker-related records.

○ Worker: Limited access based on job-specific requirements.

■ Ensures sensitive data is protected while enabling collaboration.

4. Detailed Steps to Solution Design

- **Requirement Gathering:**
  - Conducted extensive discussions with stakeholders, including owners, employers, and workers, to understand operational pain points, reporting needs, and goals.

- **Data Model Design:**
  - Created custom objects for "Rice Inventory," "Supplier," "Sales," and "Customer."
  - Defined relationships:
    - Master-detail relationship between "Rice Inventory" and "Supplier."
    - Lookup relationship between "Sales" and "Customer."

- **User Interface (UI) Design:**
  - Developed intuitive Lightning Pages tailored to different user roles (e.g., Owner Dashboard, Sales Entry Form).
  - Included custom components to facilitate data entry and quick access to reports.

- **Business Logic Implementation:**
  - Automated workflows for low inventory alerts and sales notifications.
  - Developed Apex classes and triggers for advanced calculations and inventory updates.

- **Reports and Dashboards:**
  Configured reports to highlight:
    - Daily rice production and sales trends.
    - Inventory levels and popular rice types.
    - Revenue generated by different customer segments.
  - Dashboards provide real-time visualizations for quick decision-making.

- **Documentation and Screenshots:**
  - Detailed all components, configurations, and workflows with accompanying screenshots for clarity and reference.

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

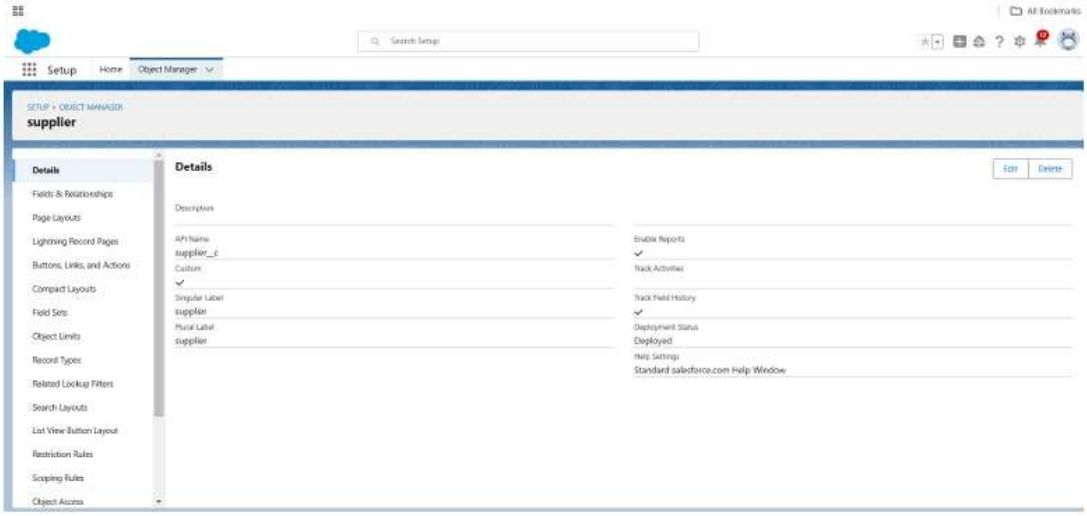
Creating required object

To create an object:

1. From the setup page -
Click on Object Manager -Click on Create - Click on Custom Object
2. On Custom object defining page:
3. Enter the label name, plural label name, click on Allow reports, Allow
4. Click onSave.

To create an object:

1. From the setup page >> Click on Object Manager>> Click on Create>>Click on Custom Object.
1. Enter the label name>>supplier
2. Plural label name>>supplier
3. Enter Record Name Label and Format
    - Record Name >> supplier Name
    - Data Type>>Text 2. Click on Allow reports and Track Field History and allow search
3. Allow search >> save

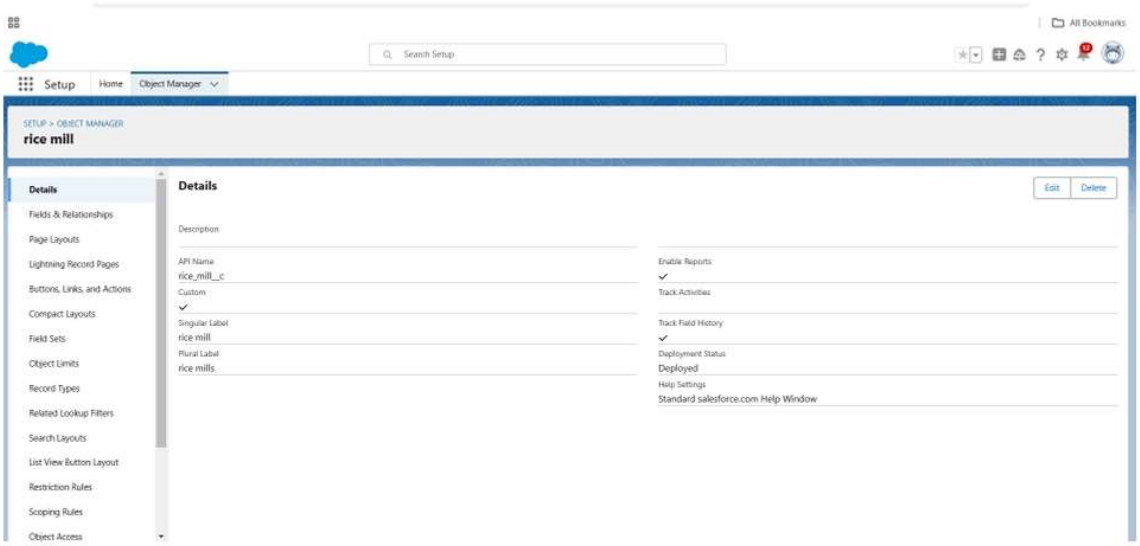


To create an object:

1. From the setup page >> Click on Object Manager>>Click on Create >> Click on Custom Object.
    1. Enter the label name>>rice mill
    2. Plural label name>> rice mills
    3. Enter Record Name Label and Formt
        - Record Name >>
        - Data Type >> Auto Number

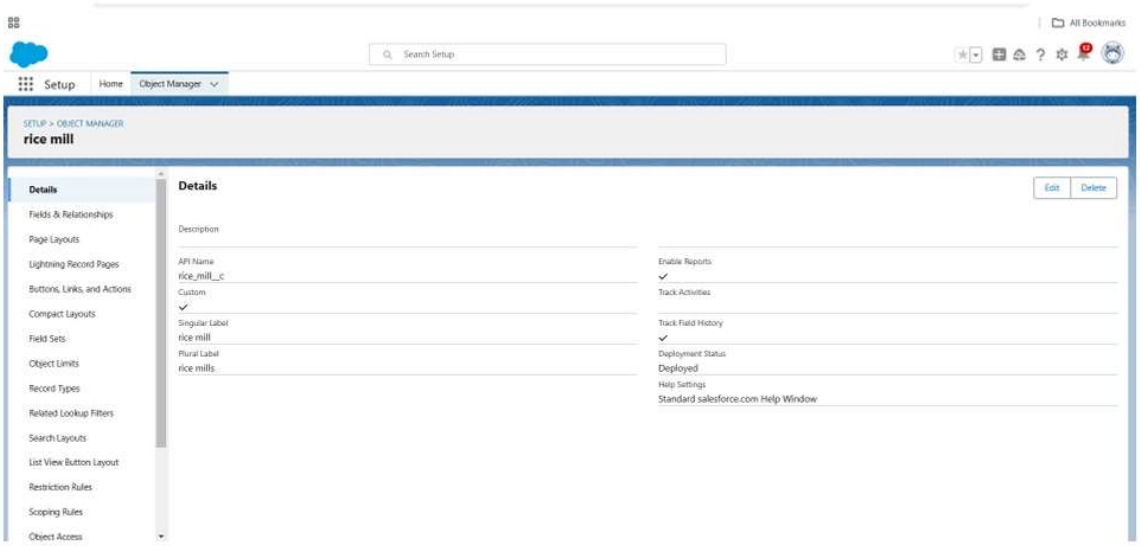
- Display Format >> rice-{000}
- Starting number >> 1

2. Click on Allow reports and Track Field History, Allow Search and Save.

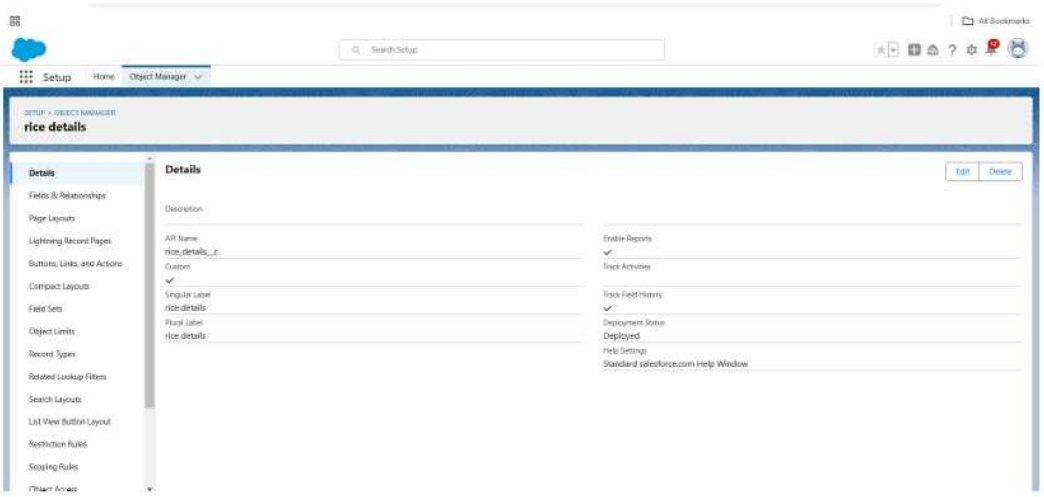


Follow the same steps as mentioned above for the and Receipt objects

1. Use these display format for the consumer
   - label name >> consumer
   - Plural label name >> consumers
   - Display Format >> consumers-{000}



1. Use these display format for the rice details
    - label name >> rice details
    - Plural label name >> rice details
    - Display Format >> rice-{000}
    - Starting Number >>1



Tabs

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:

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.

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.

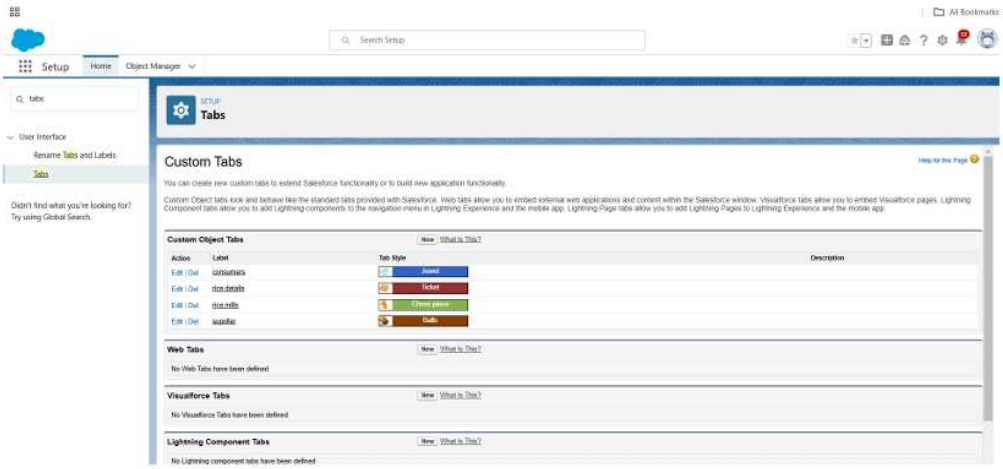
Visualforce Tabs Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

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

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.

To create a Tab:(supplier)

1. Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)
2. Select Object(supplier) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
3. Make sure that the Append tab to users' existing personal customizations is checked.
4. Click save

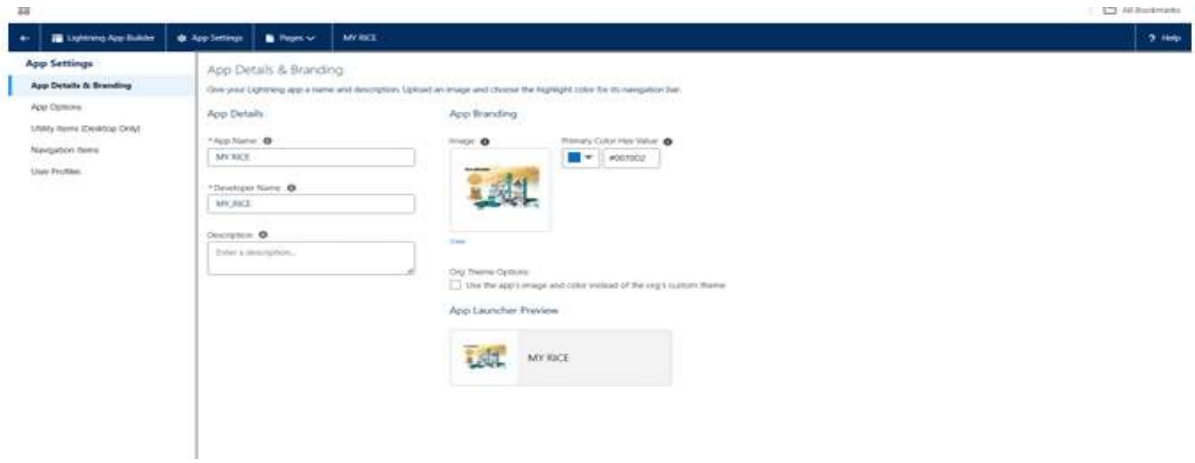


The Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your 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. To create a lightning app page:
1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.
2. Fill the app name in app details as MY RICE >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.
3. Upload a photo that is related to your app
4. To add Navigation Item:
● Select the items (supplier, rice mill, consumer, Rice details) from the search bar and using the arrow button >> Next.

● To Add User Profiles:
● Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.



Fields
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
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 anon 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 everySalesforce application. They are,
1. Created by

2. Owner
3. Last Modified
4. 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. Itmeans 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.

Creating the number field in rice details object

1. Go to the setup page >> click on object manager >> From drop down click edit for rice details object
2. Click on fields & relationship >> click on New.
3. Select Data type as “Number” and click Next.
4. Given the Field Label as “ rice distributed ” and length as “ 5 ”.
5. Field Name will be auto populated, and click on Next- Next >> Save.

A Junction object is a custom object that serves as a bridge between two related objects in a many-to-many relationship. It allows you to create a relationship between records of two different objects by creating a many-to-many relationship model.

Creating junction object as rice details with supplier& rice mill

To create junction objec

1.Go to the setup page >> click on object manager >> From drop down click edit for rice details object
2. Click on fields & relationship - click on New.
3. Select “Master-Detail relationship” as data type and click Next.
4. Select the related object “ supplier ” and click next.

5. Give Field Label as “supplier Name” and click Next.
6. Next >> Next >> Save & New.
7. Follow the same steps from 1 to 3.
8. Select the related object “ rice mill ” and click Next.
9. Give Field Label as “rice mill 1(one)” and click Next.

Creating Fields in rice mill Objects

1. Select Data type as “Number” and click Next.
2. Given the Field Label as “ rice price/kg ” and length as “ 5 ”

Creating Fields in consumer Objects

<table>
<tr><th>S.no</th><th>Object name</th><th>Fields data type</th></tr>
<tr><td>1.</td><td>consumer</td><td>
<table>
<tr><td>First name</td><td>Text</td></tr>
<tr><td>Last name</td><td>Text</td></tr>
<tr><td>Phone number</td><td>phone</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>email</td><td>email</td></tr>
<tr><td>Rice taken by shops</td><td>Number (length=5)</td></tr>
</table>
<table>
<tr><td>Rice type</td><td>(Picklist values)
1.basmati
2.normal rice</td></tr>
</table>
</td></tr>
</table>

		Mode of payment: Picklist values • Credit card • Debit card • Net banking • UPI • Cash

Creating Cross Object Formula Field in consumer Object

A cross-object formula field is a formula field that references fields from another object in Salesforce. This type of formula allows users to calculate and display data from multiple objects on a single record.

1. Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “Amount Paid ” and select formula return type as “Number” and click next.
5. Insert fields formula should be : rice_taken_by_shops__c * rice_mill_name__r.rice_price_kg__c
6. Under Advanced Formula write down the formula and click “Check Syntax” and Save. 1. Creating the Formula field in consumer Object

2. Go to setup >> click on Object Manager >> type object name(consumer)in search bar >>

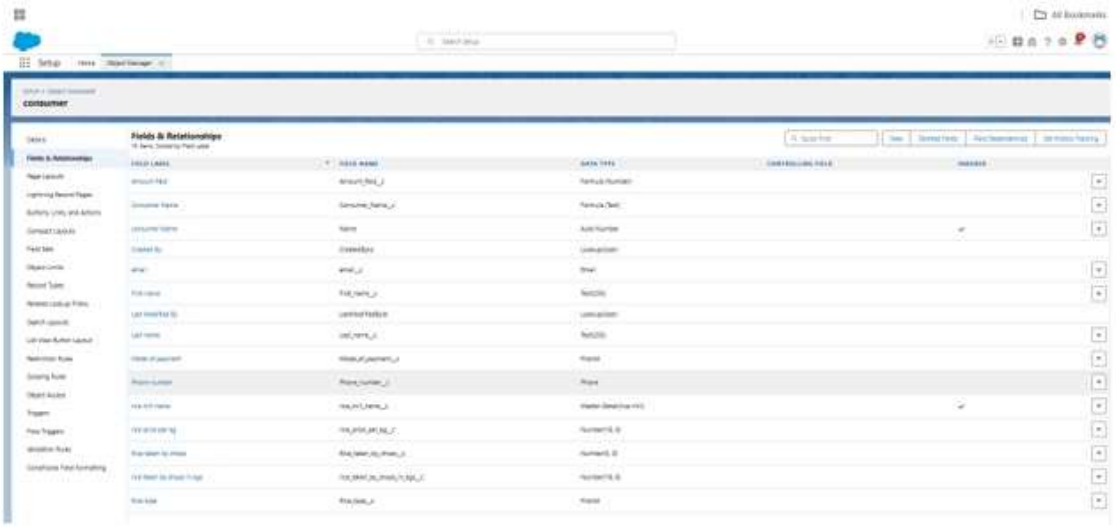
click on the object.
3. Click on fields & relationship >> click on New.
4. Select Data type as “Formula” and click Next.
5. Give Field Label and Field Name as “Consumer Name” and select formula return type as “TEXT” and click next.
6. Insert field formula should be : First_Name__c + ' ' + Last_Name__c
7. click “Check Syntax” and Save.

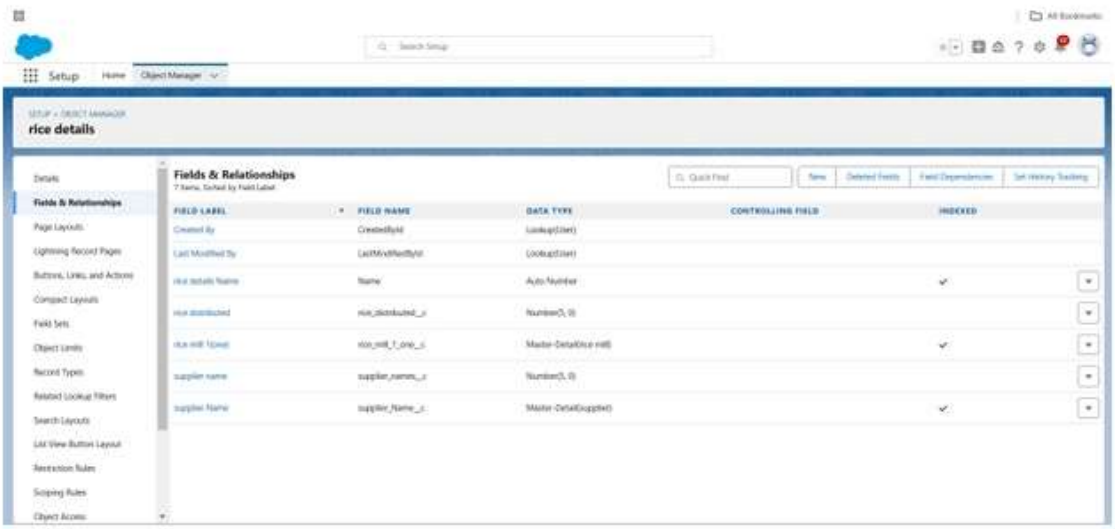
Creating the validation rule

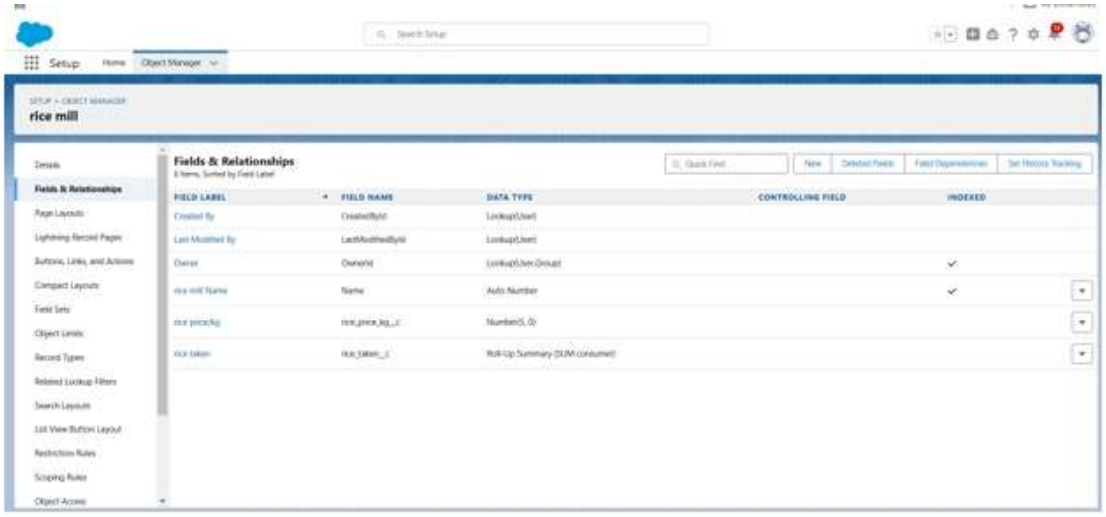
Improve the quality of your data using validation rules. Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value.

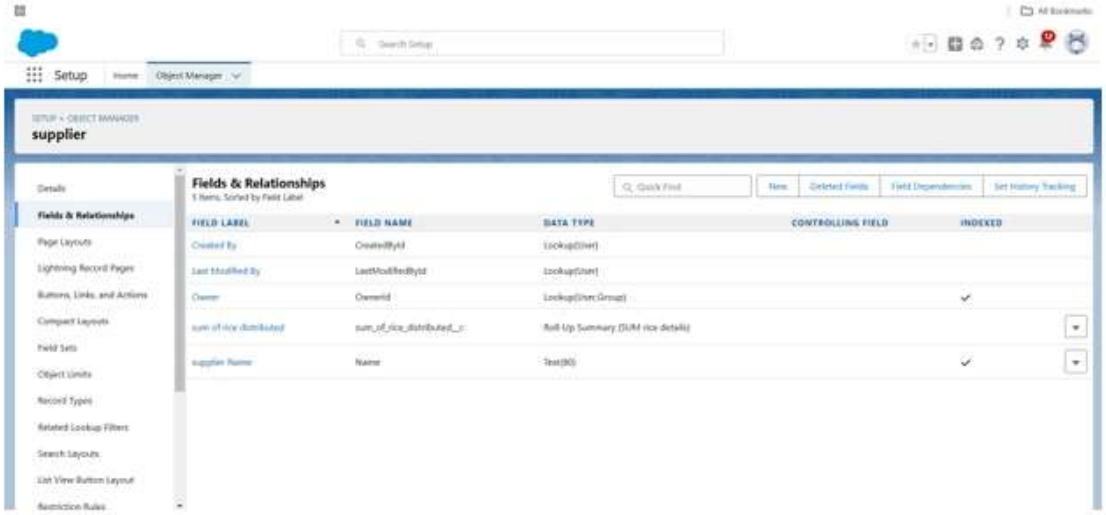
Creating the validation rule for phone number field in consumer object

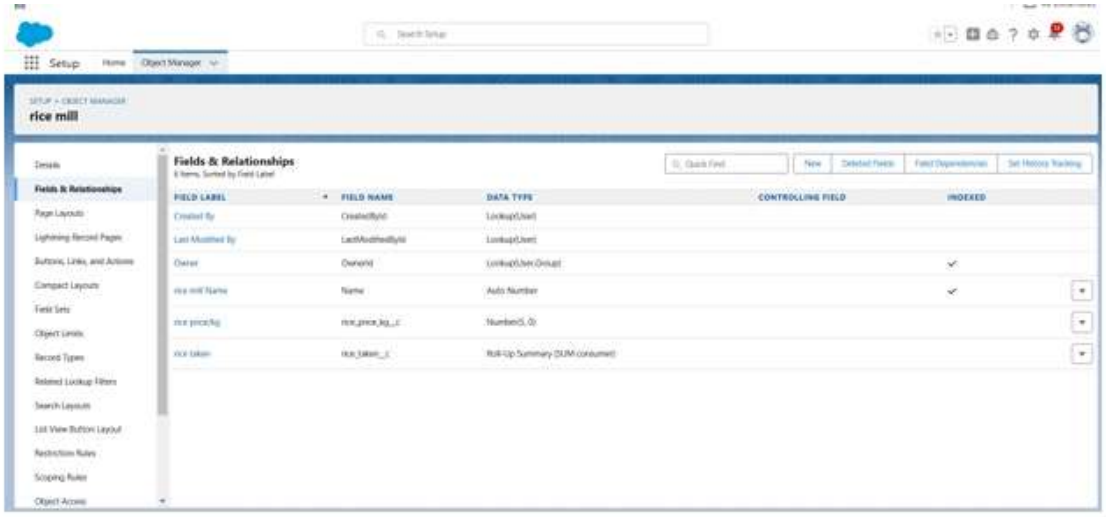
1. Go to the setup page >>click on object manager >> From drop down click edit for consumer object.
2. Click on the validation rule >> click New
3. Enter the Rule name as “Phonenumberoremailblankrule ”.
4. Enter the description as “phone number and email number should not be blank”.
5. Enter the formula as “OR(ISBLANK(phone_number__c) , ISBLANK(email__c))” and check the syntax.
6. Under the error message write as”please fill in your phone number.”
7. Select error location “top of page”.
8. Save the validation rule.

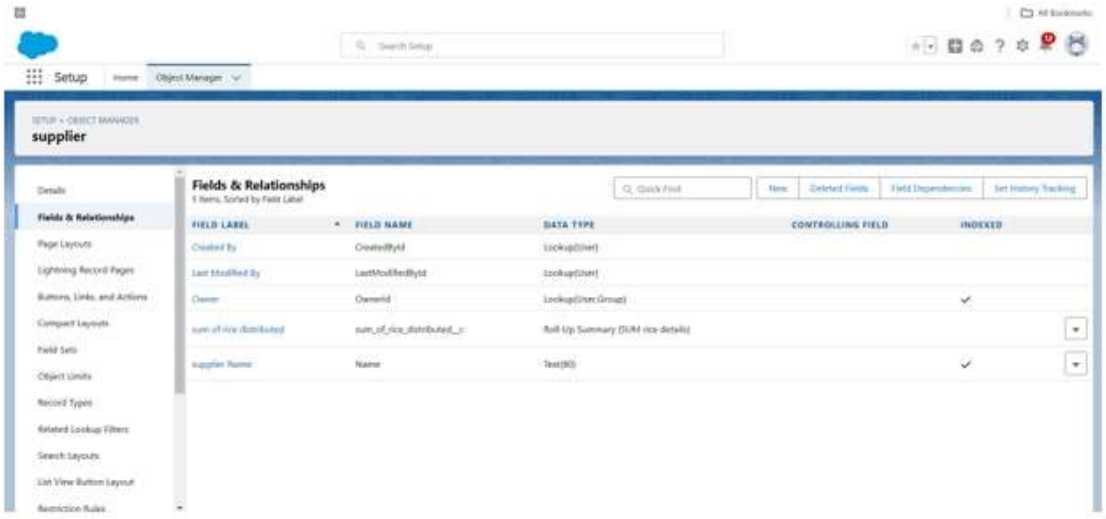

consumer
Fields & Relationships


Setup
Home
Object Manager
rice details
Fields & Relationships
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED


Setup
Home
Object Manager
rice mill
Fields & Relationships
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED


Setup
Home
Object Manager
supplier
Fields & Relationships
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED





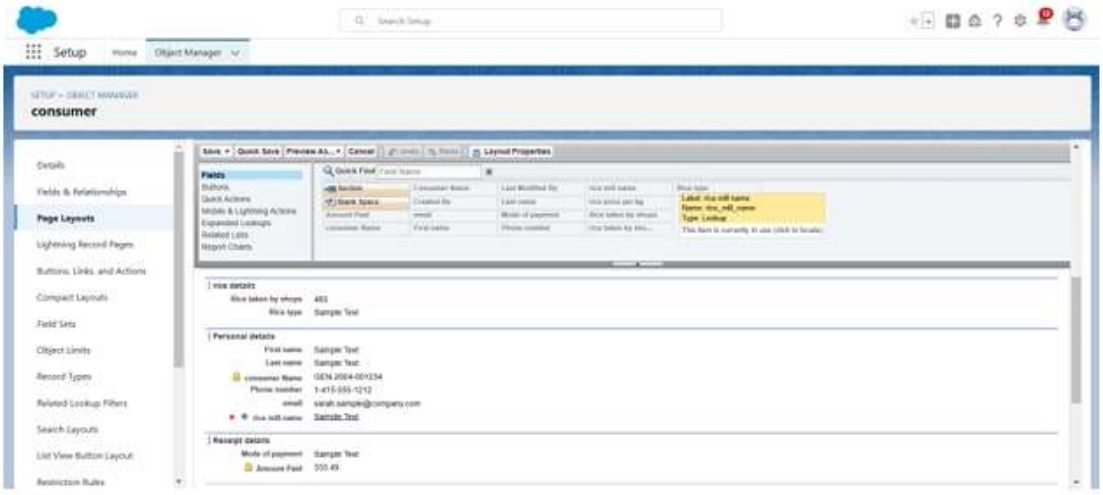
Page layouts

Page Layout in Salesforce allows us to customize the design and organize detail and edit pages of records in Salesforce. Page layouts can be used to control the appearance of fields, related lists, and custom links on standard and custom objects' detail and edit pages. Creating page Layout To Create a Page layout:

1. Go to Setup >> Click on Object Manager >>Search for the object (consumer) >> From

drop down select the object and click on it.

2. Click on Page layout >> Click on New.
3. Select the existing page layout, and give the page layout name as “consumer layout”, and click save.
4. Drag and drop the section field to consumer details and create the section.
5. Enter the section name as “Personal details”, - click Ok.
6. Now drag the fields to this section that mentioned , they are
   - First name , last name , consumer name , phone number, email, rice mill name.
7. Follow the same process for another two sections as shown above , they are
8. One section is “ rice details ” , drag the fields that are
   - Rice taken by shop, rice type.
9. Another section is “Receipt details ”, and drag the fields that are
   - Mode of payment , Amount paid.



10. Then , Click save.

Profiles

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, Visual force 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

1. Standard profiles:
By default salesforce provides below standard profiles.
- Contract Manager
- Read Only
- Marketing User
- Solutions Manager
- Standard User
- System Administrator. We cannot deleted standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

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

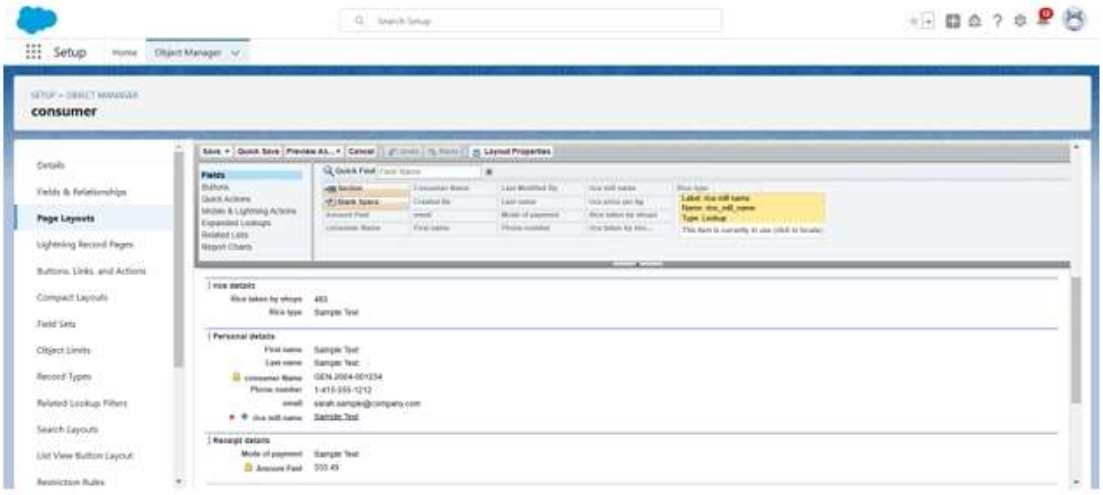
Owner Profile

To create a new profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >>enter profile name (owner) >> Save.

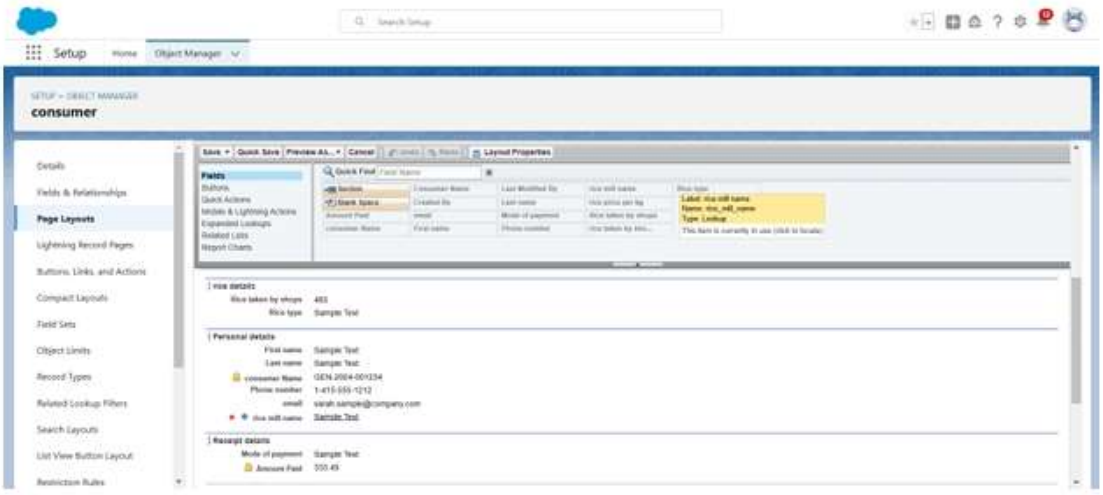
2. Scroll down to Custom Object Permissions and Give access permissions for consumers, rice details , rice mill and suppliers objects as mentioned in the below diagram.

3. Give access and save it.



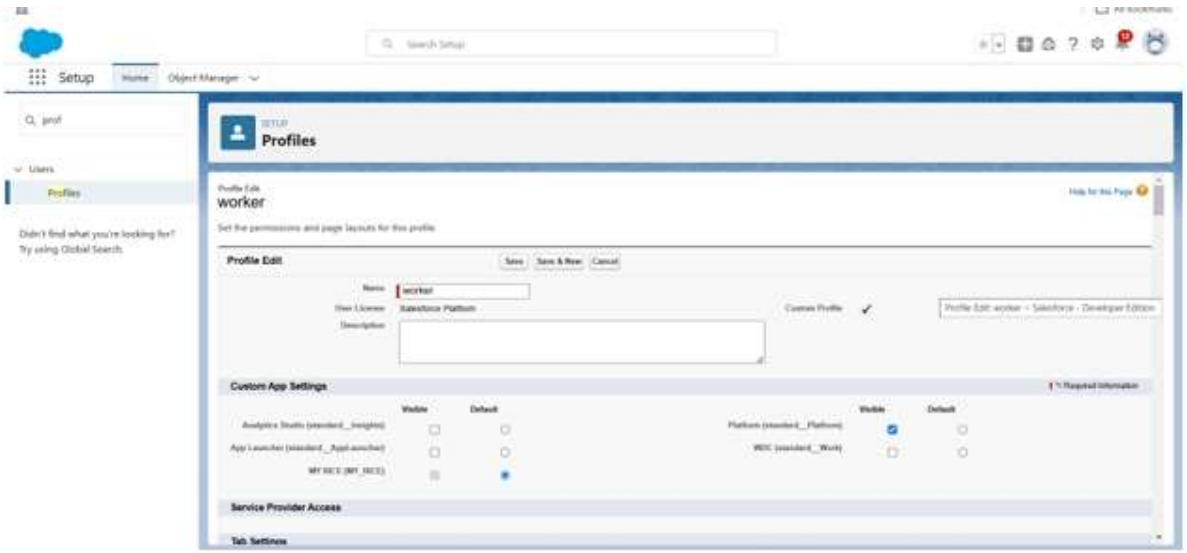
Employer Profile

1. Go to setup >> type profiles in quick find box >>click on profiles >> clone the desired profile (Standard Platform User) >> enter profile name (employer) >> Save
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the rice mill.
4. Scroll down to Custom Object Permissions and Give access permissions for consumer, rice details , rice mill and suppliers objects as mentioned in the below diagram.
5. And click save.



Worker Profile

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard Platform User) >> enter profile name (worker) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the rice mill.
4. Scroll down to Custom Object Permissions and Give access permissions for consumer, rice details , rice mill and suppliers objects as mentioned in the below diagram
5. And click save.



Role & Role Hierarchy

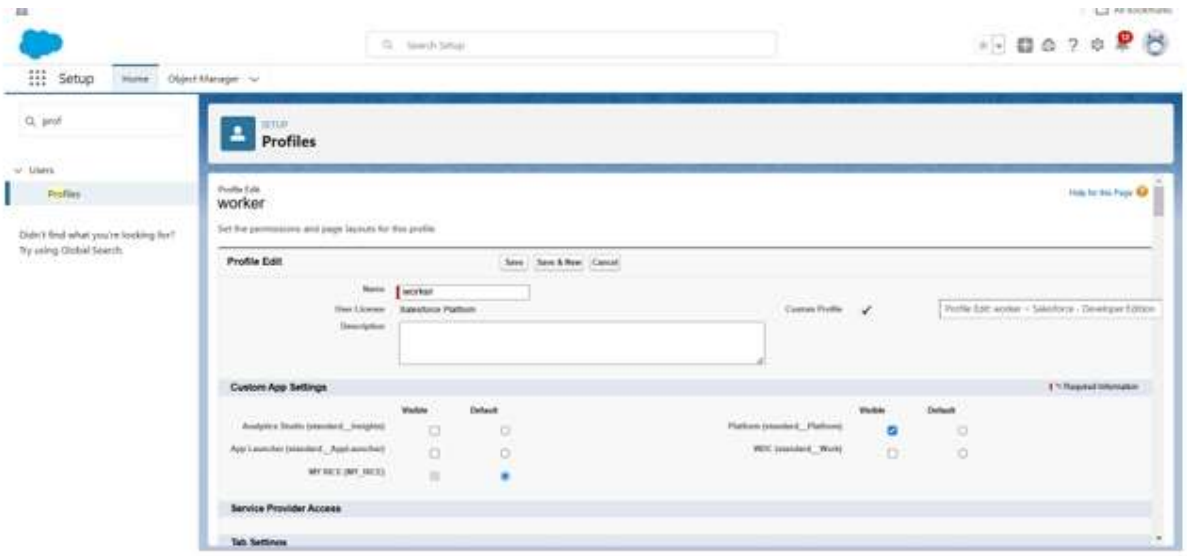
A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Creating owner Role:

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Go to quick find >> Search for Roles >> click on set up roles.
3. Click on Expand All and click on add role under whom this role works.
4. Give Label as “owner” and Role name gets auto populated. Then click on Save.
5. Click and save it.

Creating another two roles under manager

1. Go to quick find >>Search for Roles >>click on set up roles.
2. Click plus on CEO role, and click add role under owner.
3. Give Label as “employer” and Role name gets auto populated. Then click on Save.
4. Repeat the same steps, for another role.
5. Click plus on CEO role, and click plus on owner, and click add role under employer. '
6. give Label as “worker” and Role name gets auto populated. Then click on Save.



Users

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.

Create User

1. Go to setup >> type users in quick find box >> select users >> click New user. 2. Fill in the fields

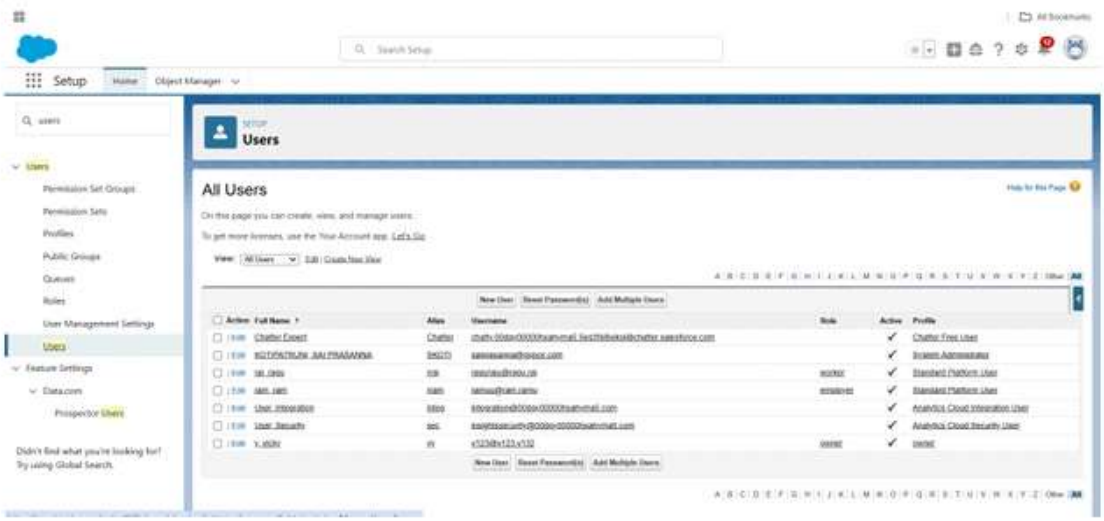
3. First Name: vicky
4. Last Name: y
5. Alias: Give a Alias Name
6. Email id: Give your Personal Email id
7. Username: Username should be in this form: text@text.text
8. Nick Name: Give a Nickname
9. Role: owner 10. User license: Salesforce
11. Profiles:owner
12. Save it.

Creating another users

1. Go to setup ? type users in quick find box ? select users ? click New user.
2. Fill in the fields
3. First Name: ram
4. Last Name: ram
5. Alias: Give a Alias Name
6. Email id: Give your Personal Email id
7. Username: Username should be in this form: text@text.text
8. Nick Name: Give a Nickname
9. Role: employer
10. User license: Salesforce platform
11. Profiles: standard platform user.

Create Another User

1. Go to setup ? type users in quick find box ? select users ? click New user.
2. Fill in the fields
3. First Name: ragu
4. Last Name: raj
5. Alias : Give a Alias Name
6. Email id : Give your Personal Email id
7. Username : Username should be in this form: text@text.text
8. Nick Name: Give a Nickname
9. Role : worker
10. User license
11. Profiles : Salesforce platform user

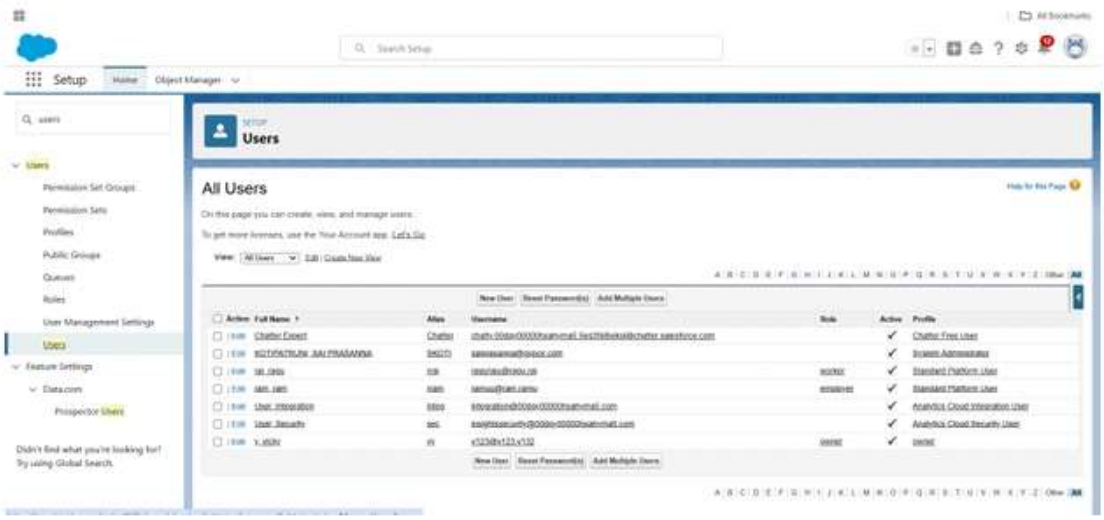


Permission sets

A permission set is a collection of settings and permissions that give users access to various tools and functions. Permission sets extend users' functional access without changing their profiles and are the recommended way to manage your users' permissions.

Creating OWD setting.

1. Go to setup >> type "sharing settings " in quick search >> Click edit
2. Scroll down, change the default internal access to " public read-only" for rice mill and supplier object.
3. Click save.
4. Extra information, By these every profile has their own access, according to their profile.
5. But in our case we created roles and given the roles in such a way that the owner can see employer and worker records , and the employer can see the worker records.



Report

Create Report

1. Go to the app >>click on the reports tab

2. Click New Report.

3. select for report type, search for “rice mill with consumers” click on it. And click on start report.

1. Their outline pane is opened already, select the fields that are mentioned below in the column section.

1.consumer name

2.rice type

3.rice price/kg

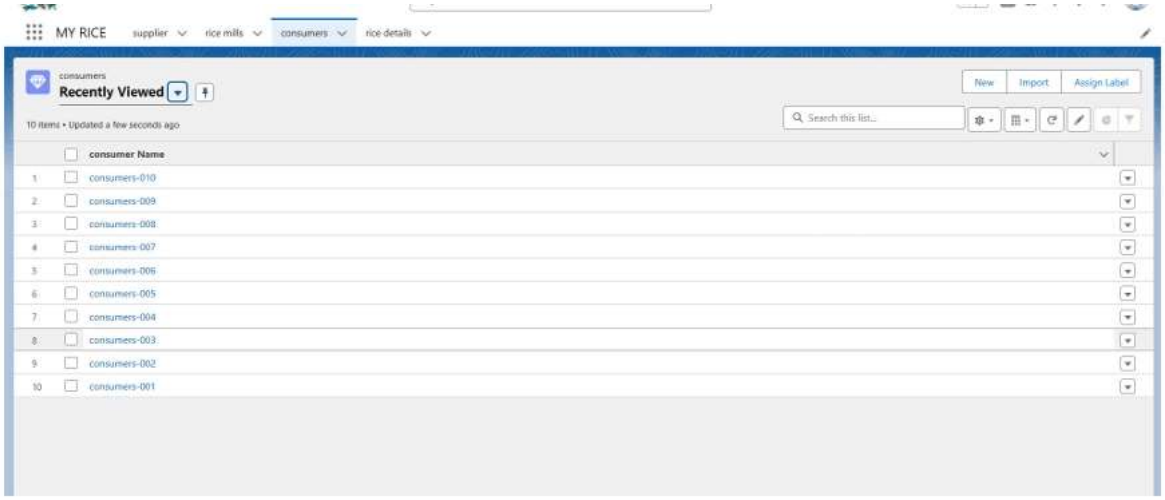
4.mode of payments

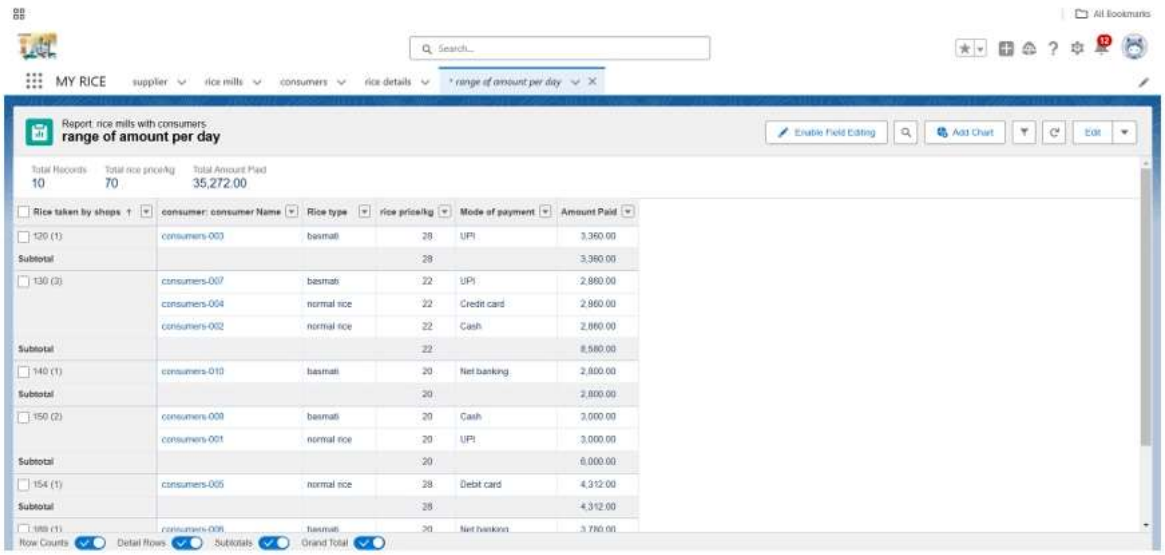
5.amount paid

2. Remove the unnecessary fields.

3. Select the fields that are mentioned below in the GROUP ROWS section.

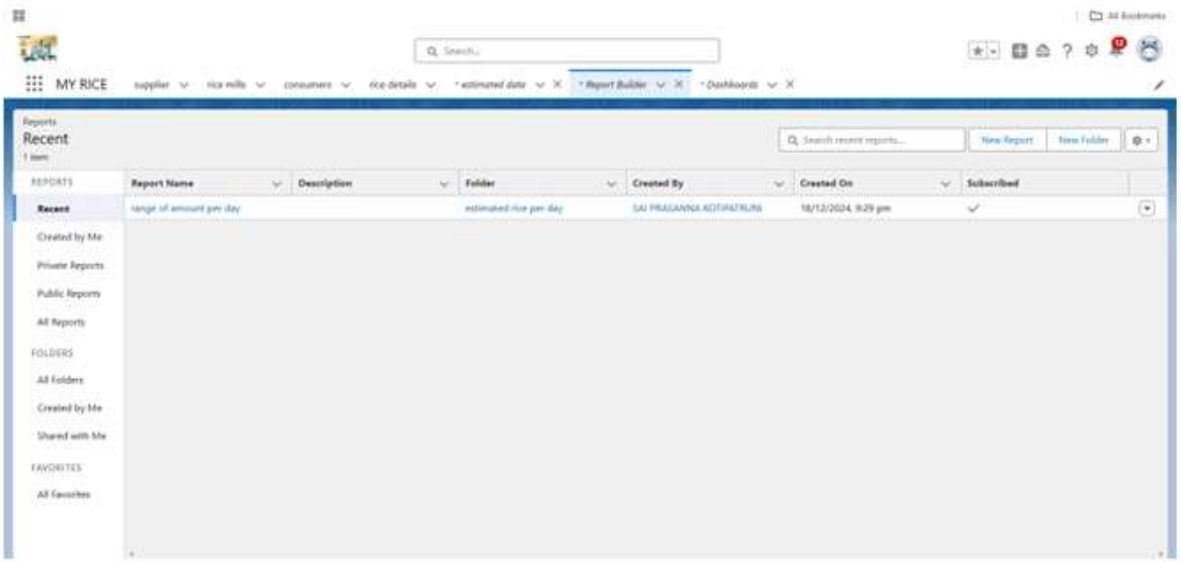
1. Rice taken by shops Click save and run and save the report as “range of amount per day”.and save it





Sharing report to owner

1. Click edit drop down and select subscribe option
2. Follow as per below image.
3. After selecting the run report as a “another person” select your personal account or whom you want to send that mail to.
4. Click save.



Create a report folder

1. Click on the app launcher and search for reports.
2. Double click on the report, " reports tab" will be auto populated in the navigation bar.
3. Click on the report tab, click on the new folder.
4. Give the Folder label as "estimated rice per day ", Folder unique name will be auto populated.

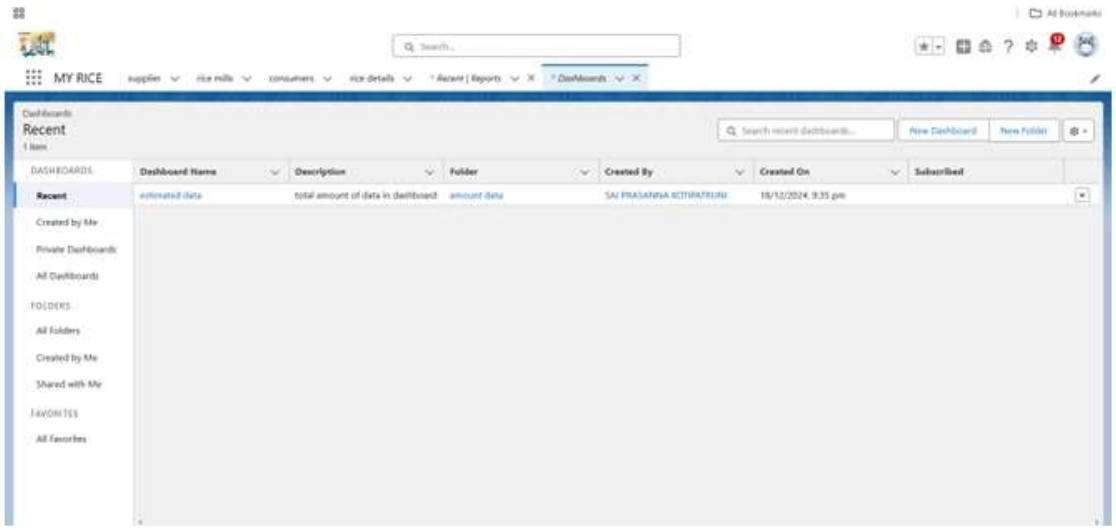
5. Click save.
   1.navigate to app launcher and click reports on that.
   2.click all reports.
3. Select the range of amount per day drop down in that click move.
4. Select estimated rice per day folder and select folder.

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.

Create Dashboard Folder

1. Click on the app launcher and search for the dashboard.
2. Click on the dashboard tab.
3. Click the new folder, give the folder label as " amount data dashboard".
4. Folder unique names will be auto populated.
5. Click save.



Create Dashboard

1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that was created, and click on create.
3. Select add component.
4. Select a Report and click on select.
   Display as>> vertical bar chart
   X-axis >> rice taken by shops
   Y-axis >> sum of amount
   Y-axis range >> automatic
   Sort by >> rice taken by shops
   Component theme >> dark.

5..Add the component

Again select add component with above same steps

1.display as donut chart

2.sort by >> sum of amount

3.title>>range of amount per day

4.component theme dark

5.Click add.

6.Click save and done.

APEX

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Lightning platform server in conjunction with calls to the Lightning Platform? API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visual force pages. Apex code can be initiated by Web service requests and from triggers on objects.

It is as similar as java i.e, it also supports OOP(Object oriented programming) like Classes, objects, methods.

Creating Classes :

Apex classes are modeled on their counterparts in Java. You'll define, instantiate, and extend classes, and you'll work with interfaces, Apex class versions, properties, and other related class concepts.

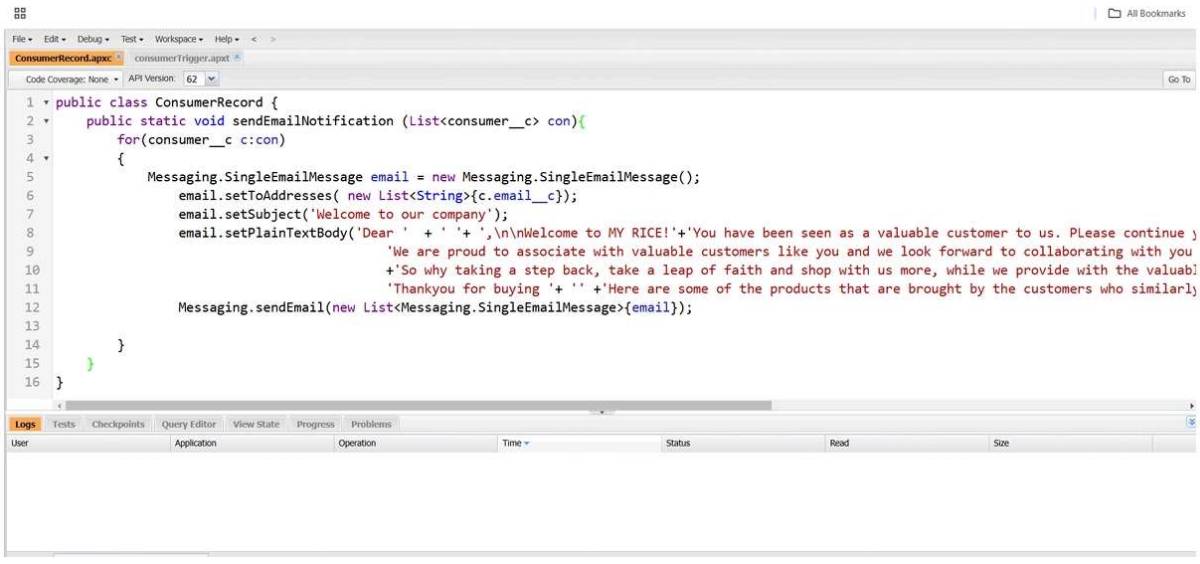
- **Class:**
  As in Java, you can create classes in Apex. A class is a template or blueprint from which objects are created. An object is an instance of a class.
- **Object**
  Object is an instance of a class, where it can access all the properties that are present in a class i.e, variables and methods.

Creating an Apex Class(ConsumerRecord)

1. Login to the Salesforce account and navigate to the gear account in the top right corner.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. Then you can see many tools in the Toolbar of the new console window. Click on File, New and Apex Class.
4. Enter the name of the class(ConsumerRecord) to create a new class file.

Code Snippet :

```
class ConsumerRecord {
    public static void sendEmailNotification (List con){
    for(consumer__c c:con)        {
        Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
email.setToAddresses( new List{c.email__c});              email.setSubject('Welcome to our
company');              email.setPlainTextBody('Dear '  + ' '+ ',\n\nWelcome to MY RICE!'+'You have
been seen  as a valuable customer to us. PLease continue your journey with us, while we try to
provide you  with good quality resources.'+'\n'+
    "We are proud to     associate with valuable customers like you and we  look forward to
collaborating with you by providing more and more exciting discounts or even  product offers
too.' + '\n'
        +'So why taking a step back, take a leap of faith and shop with us more,  while we provide
with the valuable products and offers'+'\n'+'\n'+'\n'+
                'Thankyou for buying '+ " +'Here are some of the products that are  brought by the
customers who similarly bought products like this'+'\n\n');              Messaging.sendEmail(new
List{email});          }
 }
 }
```

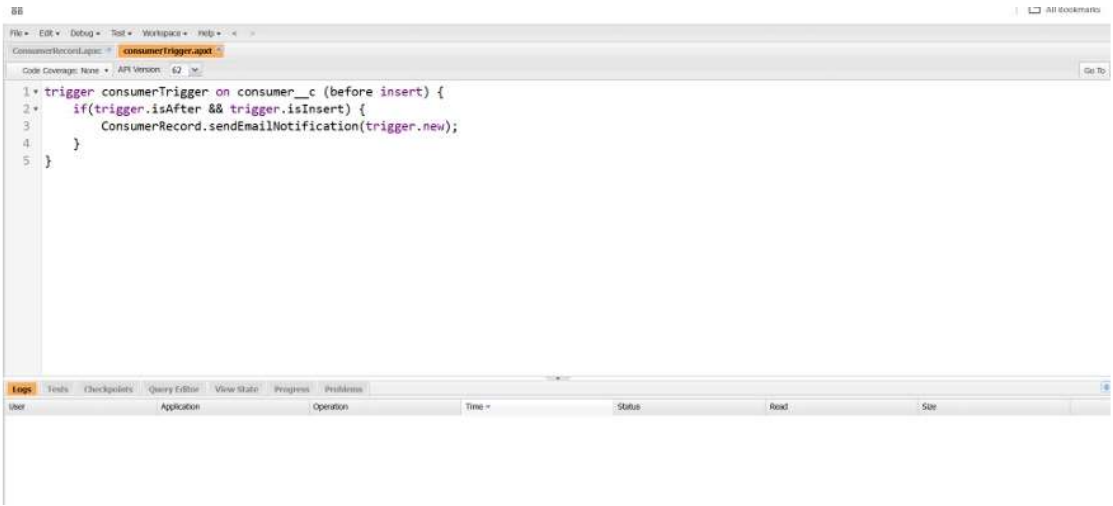


Creating an Apex Trigger
How to create a new trigger :

While still in the trailhead account, navigate to the gear icon in the top right corner. Click on developer console and you will be navigated to a new console window. Click on the File menu in the toolbar, and click on new? Trigger. Enter the trigger name and the object to be triggered.
Syntax
For creating trigger :
The syntax for creating trigger is :
Trigger [trigger name] on [object name](Before/After event)
//Trigger Logic
}

Trigger code:

```
trigger consumerTrigger on consumer__c (After insert) {
 if(trigger.isAfter && trigger.isInsert) {
ConsumerRecord.sendEmailNotification(trigger.new);
}
 }
```



5. Testing and Validation

- Unit Testing:

■ Apex classes and triggers tested to ensure they handle edge cases and return accurate results.

■ Achieved >90% code coverage to meet Salesforce standards.
● User Interface Testing:
■ Validated all forms and pages across different browsers and devices.
■ Ensured consistent user experience and accurate data presentation.
● End-to-End Testing:
■ Simulated real-world scenarios, such as entering daily production data, generating reports, and managing inventory, to confirm seamless functionality

Code Snippet :

```
class ConsumerRecord {
    public static void sendEmailNotification (List con){
   for(consumer__c c:con)       {
        Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
email.setToAddresses( new List{c.email__c});              email.setSubject('Welcome to our
company');              email.setPlainTextBody('Dear '  + ' '+ ',\n\nWelcome to MY RICE!'+'You have
been seen  as a valuable customer to us. PLease continue your journey with us, while we try to
provide you  with good quality resources.'+'\n'+
    "We are proud to    associate with valuable customers like you and we  look forward to
collaborating with you by providing more and more exciting discounts or even  product offers
too.' + '\n'
       +'So why taking a step back, take a leap of faith and shop with us more,  while we provide
with the valuable products and offers'+'\n'+'\n'+'\n'+
                'Thankyou for buying '+ " +'Here are some of the products that are  brought by the
customers who similarly bought products like this'+'\n\n');              Messaging.sendEmail(new
List{email});         }
 }
 }
```

Trigger code:

```
trigger consumerTrigger on consumer__c (After insert) {
 if(trigger.isAfter && trigger.isInsert) {
ConsumerRecord.sendEmailNotification(trigger.new);
}
 }
```

conclusion