

**SalesforceVirtual Internship Program**

**SmartInternz**

**A CRM APPLICATION FOR  
WHOLESALE RICE MILL**

**NAME : K J V LASYA PRIYA**

**E-MAIL: 322103310119@gvpce.ac.in**

**COLLEGE NAME : GAYATRI  
VIDYAPARISHAD 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:

- a. **Operational Excellence:** Automating routine processes to save time and reduce errors.
- b. **Data-Driven Decision Making:** Providing owners with detailed, real-time insights into production, sales, and customer trends.
- c. **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. A centralized platform to monitor and manage rice mill operations effectively.
- b. Real-time automated reports on daily production, sales, and revenue trends.
- c. Reduction of manual errors in data entry and calculations.
- d. User-friendly dashboards to visualize performance metrics and insights.

## **3 Salesforce Key Features and Concepts Utilized**

### **1 Reports and Dashboards:**

- a. Automated generation of daily, weekly, and monthly reports on rice production, sales, inventory levels, and revenue.
- b. Dashboards display critical metrics such as most purchased rice types, top customers, and revenue trends.

### **2 Rollup Summary Fields:**

- c. Used to summarize data from child records to parent records in master-detail relationships.
- d. Examples:
  - i. Total rice supplied by each supplier.
  - ii. Total sales revenue generated from specific rice types.

### **3 Cross-Object Formula Fields:**

- e. Enables calculations across related objects.
- f. Example: Total payment owed to suppliers calculated using **Quantity of Rice x Price per Kilogram**.

### **4 Validation Rules:**

- g. Implements logic to ensure data accuracy and completeness.
- h. Example: The **ISBLANK** formula prevents saving records with missing mandatory fields, such as rice quantity or customer details, and displays error messages to guide users.

### **Permission Sets and Organization Wide Defaults (OWD):**

- i. Configures access levels based on roles:
  - i. **Owner:** Complete access to all records, including employees and workers.
  - ii. **Employer:** Access restricted to worker-related records.
  - iii. **Worker:** Limited access based on job-specific requirements.

- j. Ensures sensitive data is protected while enabling collaboration.

## Detailed Steps to SolutionDesign

### 1 Requirement Gathering:

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

### 2 Data Model Design:

- ii. Created custom objects for "Rice Inventory," "Supplier," "Sales," and "Customer."
- iii. Defined relationships:
  1. Master-detail relationship between "Rice Inventory" and "Supplier."
  2. Lookup relationship between "Sales" and "Customer."

### 3 User Interface (UI) Design:

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

### 4 Business Logic Implementation:

- vi. Automated workflows for low inventory alerts and sales notifications.
- vii. Developed Apex classes and triggers for advanced calculations and inventory updates.

### 5 Reports and Dashboards:

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

### 6 Documentation and Screenshots:

- x. 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 objects :**

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 Click onSave.

To create an object:

1 From the setup page >> Click on ObjectManager>> Click on Create>>Click on Custom Object.

1. Enter the label name>>supplier
2. Plural label name>>supplier
3. Enter RecordName Label and Format
  - a. Record Name >> supplierName
  - b. Data Type>>Text

Click on Allowreports and TrackField History and allow search  
Allow search>> Save.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. A search bar at the top right contains the text 'Search Setup'. Below the header, the breadcrumb navigation shows 'SETUP > OBJECT MANAGER' and the object name 'supplier'. The main area is divided into two columns: 'Details' on the left and 'Fields & Relationships' on the right. The 'Details' column contains fields for 'Description', 'API Name' (set to 'supplier\_c'), 'Custom' (checked), 'Singular Label' (set to 'supplier'), and 'Plural Label' (set to 'supplier'). The 'Fields & Relationships' column includes sections for 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window'). At the bottom right of the main area are 'Edit' and 'Delete' buttons.

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 RecordName Label and Format
  - Record Name >>
  - Data Type >> Auto Number
  - Display Format >> rice-{000}
  - Starting number >> 1
4. Click on Allow reports and Track Field History, Allow Search and Save.

The screenshot shows the Salesforce Object Manager interface. At the top, there's a navigation bar with icons for Home, Object Manager, and a search bar labeled 'Search Setup'. Below the navigation is a breadcrumb trail 'SETUP > OBJECT MANAGER'. The main area displays the 'rice mill' object details. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules, and Object Access. The 'Details' tab is selected. The main content area shows the following fields:

Field	Value
Description	
API Name	rice_mill_c
Custom	✓
Singular Label	rice mill
Plural Label	rice mills
Enable Reports	✓
Track Activities	
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

At the bottom right of the main content area are 'Edit' and 'Delete' buttons.

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

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

SETUP > OBJECT MANAGER  
**consumer**

**Details**

Description

API Name: consumer\_c

Custom: ✓

Singular Label: consumer

Plural Label: consumers

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Edit | Delete

## 1. Use these displayformat for the rice details

- a. label name >>rice details
- b. Plural label name >> rice details
- c. Display Format >> rice-{000}
- d. Starting Number >>1

SETUP > OBJECT MANAGER  
**rice details**

**Details**

Description

API Name: rice\_details\_c

Custom: ✓

Singular Label: rice details

Plural Label: rice details

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Edit | Delete

## 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 Appendtab to users' existing personalcustomizations is checked.
4. Click save

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. The left sidebar has 'Setup' selected. The main area has a search bar at the top. Below it, 'Custom Tabs' is displayed with a note: 'You can create new custom tabs to extend Salesforce functionality or to build new application functionality.' There are four sections: 'Custom Object Tabs', 'Web Tabs', 'Visualforce Tabs', and 'Lightning Component Tabs'. The 'Custom Object Tabs' section contains a table:

Action	Label	Tab Style	Description
Edit   Del	consumers	Car	
Edit   Del	rice details	Ticket	
Edit   Del	rice mills	Mail	
Edit   Del	supplier	Box	

## The Lightning App

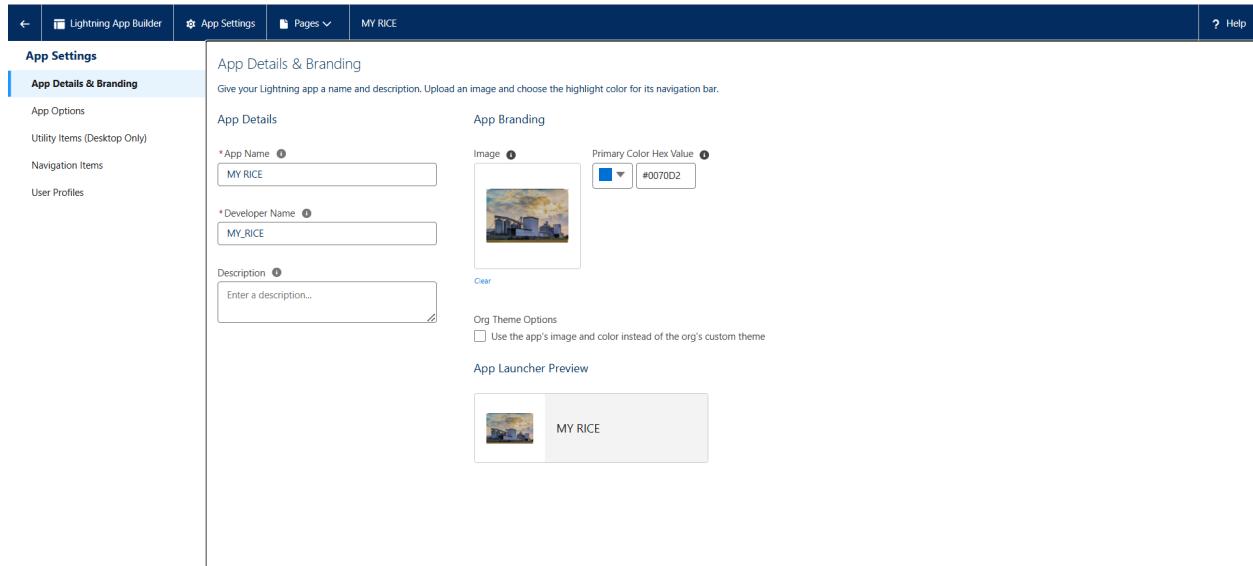
An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

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:
  - a. Select the items (supplier, rice mill, consumer, Rice details ) from the searchbar and move it using the arrow button >> Next.
  - b. To Add User Profiles:
  - c. Search profiles (Systemadministrator) 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. StandardFields
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 StandardField until it is an 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,

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. 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 CustomFields of any given form.

#### **Creating the number field in rice details object**

- a. Go to the setup page >> click on object manager >> From drop down click edit for rice details object
- b. Click on fields & relationship >> click on New.
- c. Select Data type as "Number" and click Next.
- d. Given the Field Label as "rice distributed" and length as "5".
- e. 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 object

Go to the setup page >> click on object manager >> From drop down click edit for rice details object

1. Click on fields & relationship - click on New.
2. Select "Master-Detailrelationship" as datatype and click Next.
3. Select the related object " supplier" and click next.
4. Give Field Label as "supplier Name" and click Next.
5. Next >> Next >> Save & New.
6. Follow the same steps from 1 to 3.
7. Select the related object " rice mill " and click Next.
8. Give Field Label as "rice mill 1(one)" and click Next.
9. Next >> Next >> Save.

Master-detail relationship is a type of relationship between two objects where the master object controls certain behaviors and settings of the detail object. Here are a few use cases that demonstrate the use of master-detail relationships

### **Creating Master-Detail Relationship between consumer & rice mill Object**

To Create a Master-Detail relationship

1. Go to the setup page >> click on object manager >> From drop down click edit for consumer object.
2. Click on fields & relationship >> click on New.
3. Select "Master-Detailrelationship" as datatype and click Next.
4. Select the related object " rice mill".
5. Give Field Label as "rice mill name" and click Next.
6. Next >> Next >> Save.

A rollupsummary field is a field that summarizes data from a child object to a parent object that shares a master-detail relationship. Rollup summary fields can use the COUNT, SUM, MIN, and MAX functions. For example, you could use a rollup summary field to display the total value (amount of rice supplied) from rice details on a related supplier.

Creating the Roll-up summary field on supplier & rice mill Objects.

1. Go to setup >> click on Object Manager >> type object name(supplier) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New

#### **Creating the number field in rice detailsobject**

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 “master detail” and click Next.
4. Given the Field Label as “ supplier name ” and length as “ 5 ”
5. Field Name will be auto populated, and click on Next>> Next >>Save.

#### **Creating Fields in rice mil 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**

S.n o	Object name	Fields	data type
	consumer	First name Last name Phone number email	text text phone email

S.no	Object name	Fields	data type

	consumer	<p>Rice taken by shops      Number (length=5)</p> <p>Rice type      (Picklist values)            1 basmati            2 normal rice</p> <p>Mode of payment            Picklist            values</p> <ul style="list-style-type: none"> <li>o Credit card</li> <li>o Debit card</li> <li>o Net banking</li> <li>o UPI</li> <li>o Cash</li> </ul>
--	----------	--

### **Creating Cross ObjectFormula 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 objectname(consumer) in searchbar >> click on the object.
  2. Click on fields & relationship >>click on New.
  3. Select Data type as “Formula” and click Next.
  4. Give Field Labeland Field Name as “AmountPaid ” 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 searchbar >> 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 "ConsumerName" and select formula return type as "TEXT" and click next.
6. Insert field formulashould 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 phonenumbers 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

Setup > Object Manager consumer

Fields & Relationships		
15 Items. Sorted by Field Label		
Created By	CreatedBy	Lookup(Page)
email	email_c	Email
First name	First_name_c	Text(255)
Last Modified By	LastModifiedBy	Lookup(User)
Last name	Last_name_c	Text(255)
Mode of payment	Mode_of_payment_c	Picklist
Phone number	Phone_number_c	Phone
Record Type	RecordTypeId	Record Type
rice mill name	rice_mill_c	Master-Detail(rice mill)
Rice taken by shops	Rice_taken_by_shops_c	Number(5, 0)
rice taken by shops in kgs	rice_taken_by_shops_in_kgs_c	Number(18, 0)

Setup > Object Manager rice details

Fields & Relationships		
7 Items. Sorted by Field Label		
Created By	CreatedBy	Lookup(User)
Last Modified By	LastModifiedBy	Lookup(User)
rice details Name	Name	Auto Number
rice distributed	rice_distributed_c	Number(5, 0)
rice mill (one)	rice_mill_c	Master-Detail(rice mill)
supplier Name	supplier_c	Master-Detail(supplier)
supplier name	supplier_name_c	Number(5, 0)

Setup > Object Manager rice mill

Fields & Relationships		
7 Items. Sorted by Field Label		
Created By	CreatedBy	Lookup(User)
Last Modified By	LastModifiedBy	Lookup(User)
Owner	OwnerId	Lookup(User,Group)
rice distributed to shops	rice_distributed_to_shops_c	Roll-Up Summary (SUM rice details)
rice mill Name	Name	Auto Number
rice price/kg	rice_price_kg_c	Number(5, 0)
rice taken	rice_taken_c	Roll-Up Summary (SUM consumer)

**Setup > Object Manager**

### supplier

**Fields & Relationships**

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
sum of rice distributed	sum_of_rice_distributed_c	Roll-Up Summary (SUM rice details)		
supplier Name	Name	Text(80)		

**Validation Rule Edit**

Validation Rule Edit

Rule Name: **Phone number or email blank**

Active:

Description: phone number and email number should not be blank

Error Condition Formula:

Example: `ISBLANK(Phone_number__c) OR ISBLANK(email__c)`

If this formula expression is true, display the text defined in the Error Message area.

Insert Field: Insert Operator: `OR`

Functions: ABS, ACOS, ADDMILLISECONDS, AND, ACOS, ASIN

Insert Selected Function: ABS(number), ACOS(number), ADDMILLISECONDS(value, number), a number without its sign

Check Syntax

Error Message:

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is true

Error Message: Please fill in your phone number.

Save | Save & New | Cancel | Help for this Page

**consumer Validation Rule**

Validation Rule Edit

Rule Name: **Phone number or email blank**

Active:

Description: phone number and email number should not be blank

Error Condition Formula:

Example: `ISBLANK(Phone_number__c) OR ISBLANK(email__c)`

If this formula expression is true, display the text defined in the Error Message area.

Insert Field: Insert Operator: `OR`

Functions: ABS, ACOS, ADDMILLISECONDS, AND, ACOS, ASIN

Insert Selected Function: ABS(number), ACOS(number), ADDMILLISECONDS(value, number), a number without its sign

Check Syntax

Error Message:

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is true

Error Message: Please fill in your phone number.

Save | Save & New | Cancel | Help for this Page

**MY RICE**

supplier > rice mills >

Recently Viewed

10 items • Updated a few seconds ago

consumer Name

- 1 consumer-010
- 2 consumer-009
- 3 consumer-008
- 4 consumer-007
- 5 consumer-006
- 6 consumer-005
- 7 consumer-004
- 8 consumer-003
- 9 consumer-002
- 10 consumer-001

First name: Srinu

Last name: Raju

Phone number:

email:

\*rice mill name: rice-003

Receipt details

Mode of payment: Credit card

We hit a snag.

Review the errors on this page.

please fill in your phone number.

New | Import | Assign Label |

## 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 "consumerlayout", and click save

1. Drag and drop the section field to consumer details and create the section.

2. Enter the section name as "Personal details", - click Ok.

3. Now drag the fields to this section that mentioned, they are

a. First name, last name, consumer name, phone number, email, rice mill name.

4. Follow the same process for another two sections as shown above, they are

5. One section is "rice details", drag the fields that are

a. Rice taken by shop, rice type.

6. Another section is "Receipt details", and drag the fields that are

a. Mode of payment, Amount paid.

7. 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.

- a. Contract Manager
- b. Read Only
- c. Marketing User
- d. Solutions Manager
- e. StandardUser
- f. 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.

The screenshot shows the Salesforce Setup interface with the 'Profiles' tab selected. A search bar at the top contains 'prof'. On the left, a sidebar shows 'Users' and 'Profiles' under 'Setup'. The main area displays the 'Profile Edit' screen for 'OWNER'. The 'Name' field is set to 'owner'. Under 'Custom App Settings', the 'Visible' column has checkboxes for 'All Tabs (standard\_\_AllTabSet)', 'Analytics Studio (standard\_\_Insights)', 'App Launcher (standard\_\_AppLauncher)', 'Automation (standard\_\_FlowApp)', and 'Dolt Solutions (standard\_\_LightningDolt)'. The 'Default' column has radio buttons next to each row. In the 'Object Permissions' section, the 'Visible' column has checkboxes for 'Queue Management (standard\_\_QueueManagement)', 'Sales (standard\_\_LightningSales)', 'Sales (standard\_\_Sales)', 'Sales Console (standard\_\_LightningSalesConsole)', and 'Salesforce Chatter (standard\_\_Chatter)'. The 'Default' column has radio buttons next to each row. Buttons at the bottom include 'Save', 'Save & New', and 'Cancel'.

### Employer Profile

1. Go to setup>> type profilesin 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.

The screenshot shows the Salesforce Setup interface with the 'Profiles' tab selected. A search bar at the top has 'prof' typed into it. On the left, a sidebar shows 'Users' and 'Profiles' under 'Users'. The main area displays a 'Profile Edit' screen for a profile named 'employer'. The profile is set to 'Custom Profile'. Under 'Custom App Settings', the 'MY RICE (MY\_RICE)' app is selected as the default. Other apps like Analytics Studio and App Launcher are listed with their visibility and default settings. The 'Service Provider Access' and 'Tab Settings' tabs are also visible at the bottom.

## 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 profilepage, 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 quickfind >> Searchfor Roles >>click on set up roles.
2. Go to quick find >> Searchfor 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 quickfind >>Search for Roles >>click on set up roles.
2. Click plus on CEO role, and click add role under owner.
1. Give Label as "employer" and Role name gets auto populated. Then click on Save.
1. Repeat the same steps, for another role.
2. Click plus on CEO role, and click plus on owner, and click add role under employer.
3. 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: Salesforce platform

## 11. Profiles: standard platform user.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chatter Expert	Chatter	chatty.00day00000hry3umaf.stywozvifazn@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	LASYA PRIYA KAVI	KLASY	kassyle@salesforceproject.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	raj.ragu	raj	raguraj75@gmail.com	worker	<input checked="" type="checkbox"/>	Standard Platform User
<input type="checkbox"/>	ram.ram	rram	siranm78@gmail.com	employer	<input checked="" type="checkbox"/>	Standard Platform User
<input type="checkbox"/>	User Integration	integ	integration@00day00000hry3umaf.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>	User Security	sec	insightssecurity@00day00000hry3umaf.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User
<input type="checkbox"/>	y.vicky	xy	vickyv@71@gmail.com	owner	<input checked="" type="checkbox"/>	owner

## 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.

The screenshot shows the Salesforce Sharing Settings page. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar. On the left, there's a sidebar with 'Sharing Settings' selected under 'Security'. A message says 'Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Sharing Settings' and lists various objects with their sharing permissions:

Service Territory	Public Read/Write	Private	
Shift	Private	Private	✓
Shipment	Private	Private	✓
Shipping Carrier	Public Read Only	Private	✓
Shipping Carrier Method	Public Read Only	Private	✓
Shipping Configuration Set	Public Read Only	Private	✓
Streaming Channel	Public Read/Write	Private	✓
Tableau Host Mapping	Public Read Only	Private	✓
User Provisioning Request	Private	Private	✓
Walllist	Private	Private	✓
Web Cart Document	Private	Private	✓
Work Order	Private	Private	✓
Work Plan	Private	Private	✓
Work Plan Template	Private	Private	✓
Work Step Template	Private	Private	✓
Work Type	Private	Private	✓
Work Type Group	Public Read/Write	Private	✓
consumer	Controlled by Parent	Controlled by Parent	
rice details	Controlled by Parent	Controlled by Parent	
rice mill	Public Read Only	Public Read Only	✓

## Report

### Create Report

1. Go to the app >>click on the reportstab
2. Click New Report.
3. select for report type, search for “rice mill with consumers” click on it. And clickon start report.
1. Their outline pane is opened already, select the fields that are mentioned below in the column section.
  - a. consumer name
  - b. rice type
  - c. rice price/kg
  - d. mode of payments
  - e. amount paid
2. Remove the unnecessary fields.
3. Select the fields that are mentioned below in the GROUP ROWS section.
  - a. Rice taken by shops

Click save and run and save the report as "range of amount per day".and save it.

Rank	Consumer Name
1	consumers-011
2	consumers-013
3	consumers-012
4	consumers-010
5	consumers-009
6	consumers-008
7	consumers-006
8	consumers-005
9	consumers-004
10	consumers-003
11	consumers-002

Total Records	Total rice price/kg	Total Amount Paid			
11	500	17,000.00			
Report: rice mill with consumers range of amount per day					
Rice taken by shops	consumer Name	Rice type	rice price/kg	Mode of payment	Amount Paid
8 (1)	consumers-002	basmati	50	Cash	400.00
Subtotal			50		400.00
10 (1)	consumers-003	normal rice	50	Cash	500.00
Subtotal			50		500.00
16 (1)	consumers-009	basmati	50	Cash	800.00
Subtotal			50		800.00
20 (1)	consumers-006	normal rice	50	Cash	1,000.00
Subtotal			50		1,000.00
23 (1)	consumers-010	basmati	50	Cash	1,150.00
Subtotal			50		1,150.00
25 (1)	consumers-004	basmati	50	Cash	1,250.00
Subtotal			50		1,250.00
29 (1)	consumers-012	basmati	50	Net banking	1,450.00
Subtotal			50		1,450.00
35 (1)	consumers-013	normal rice	50	Net banking	1,750.00
Subtotal			50		1,750.00
Row Counts	Detail Rows	Subtotals	Grand Total		

## 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

The screenshot shows a reporting application interface. At the top, there is a navigation bar with items like "MY RICE", "supplier", "rice mills", "consumers", "rice details", and a search bar. Below the navigation bar is a sidebar with sections for "REPORTS", "FOLDERS", and "FAVORITES". The main area displays a table titled "range of amount per day" with columns for Report Name, Description, Folder, Created By, Created On, and Subscribed. One row is visible in the table.

The screenshot shows a Gmail inbox. The left sidebar includes "Compose", "Inbox" (46), "Starred", "Snoozed", "Sent", "Drafts", and "More". The main area shows an email from "KJV LASYA PRIYA" with the subject "Report results (range of amount per day)". The email body contains the title "range of amount per day", a timestamp "As of 29/12/24 at 8:00 AM · Viewing as KJV LASYA PRIYA", and a blue button "OPEN IN SALESFORCE". Below the email are sections for "Details" and "Filters" with options like "My rice mills" and "Created Date: Current FQ".

## 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

Name	Created By	Created On	Last Modified By	Last Modified Date
Einstein Bot Reports	Automated Process	25/12/2024, 6:52 am	Automated Process	25/12/2024, 6:52 am
Einstein Bot Reports Spring '23	Automated Process	25/12/2024, 6:52 am	Automated Process	25/12/2024, 6:52 am
Einstein Bot Reports Summer '23	Automated Process	25/12/2024, 6:52 am	Automated Process	25/12/2024, 6:52 am
Einstein Bot Reports Summer '22	Automated Process	25/12/2024, 6:52 am	Automated Process	25/12/2024, 6:52 am
Einstein Bot Reports Winter '23	Automated Process	25/12/2024, 6:52 am	Automated Process	25/12/2024, 6:52 am
Enablement Dashboard Reports Spring '24	Automated Process	25/12/2024, 6:52 am	Automated Process	25/12/2024, 6:52 am
Enablement Dashboard Reports Summer '24	Automated Process	25/12/2024, 6:52 am	Automated Process	25/12/2024, 6:52 am
estimated rice per day	KJV LASYA PRIYA	28/12/2024, 1:30 pm	KJV LASYA PRIYA	28/12/2024, 1:30 pm

## 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

- a. Click on the app launcher and search for the dashboard.
- b. Click on the dashboard tab.
- c. Click the new folder, give the folder label as "amount data dashboard".
- d. Folder unique names will be auto populated.
- e. Click save.

The screenshot shows a dashboard application interface. At the top, there is a header with a logo, the text "MY RICE", and several navigation tabs: "supplier", "rice mills", "consumers", "rice details", "range of amount per day", and "Dashboards". Below the header is a search bar labeled "Search...". To the right of the search bar are various icons for filtering, sorting, and saving. The main area is titled "Dashboards" and "Recent". It shows a table with one item: "estimated data" (Dashboard Name), "total amount of data in dashboards" (Description), "amount data dashboard" (Folder), "KIV LASYA PRIYA" (Created By), "28/12/2024, 1:53 pm" (Created On), and "Subscribed" (Status). On the left side, there is a sidebar with categories: "DASHBOARDS" (Recent, Created by Me, Private Dashboards, All Dashboards), "FOLDERS" (All Folders, Created by Me, Shared with Me), and "FAVORITES" (All Favorites).

## Create Dashboard

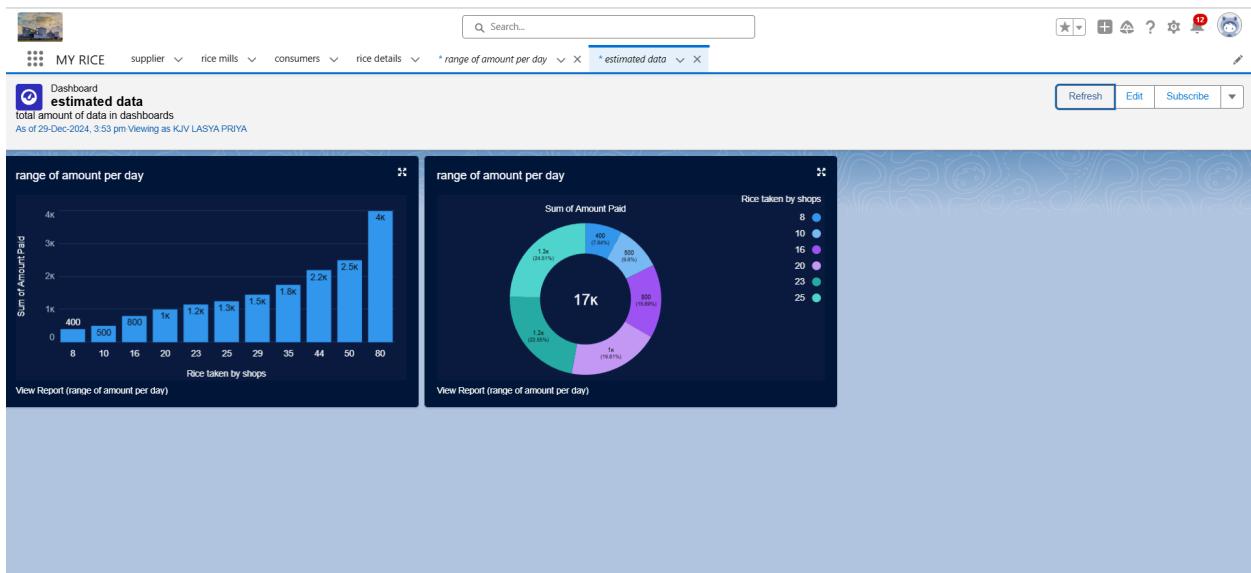
1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that was created, and click on create.
3. Select add component.
4. Select a Report and click  
on select. Display as>>  
vertical bar chart  
  
X-axis >> rice  
taken by shops Y-  
axis >> sum of  
amount  
Y-axis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

3. title>>range of amount per day
  - 4.Component theme dark
1. Click add.
  2. 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 otherrelated class concepts.

#### **a. Class:**

As in Java, you can createclasses in Apex.A class is a templateor blueprint from which objects are created. An object is an instance of a class.

#### **b. Object**

Object is an instanceof 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 Developerconsole. Click on the developerconsole 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.

Enter the name of the class(ConsumerRecord) to create a new class file.

#### **Code Snippet :**

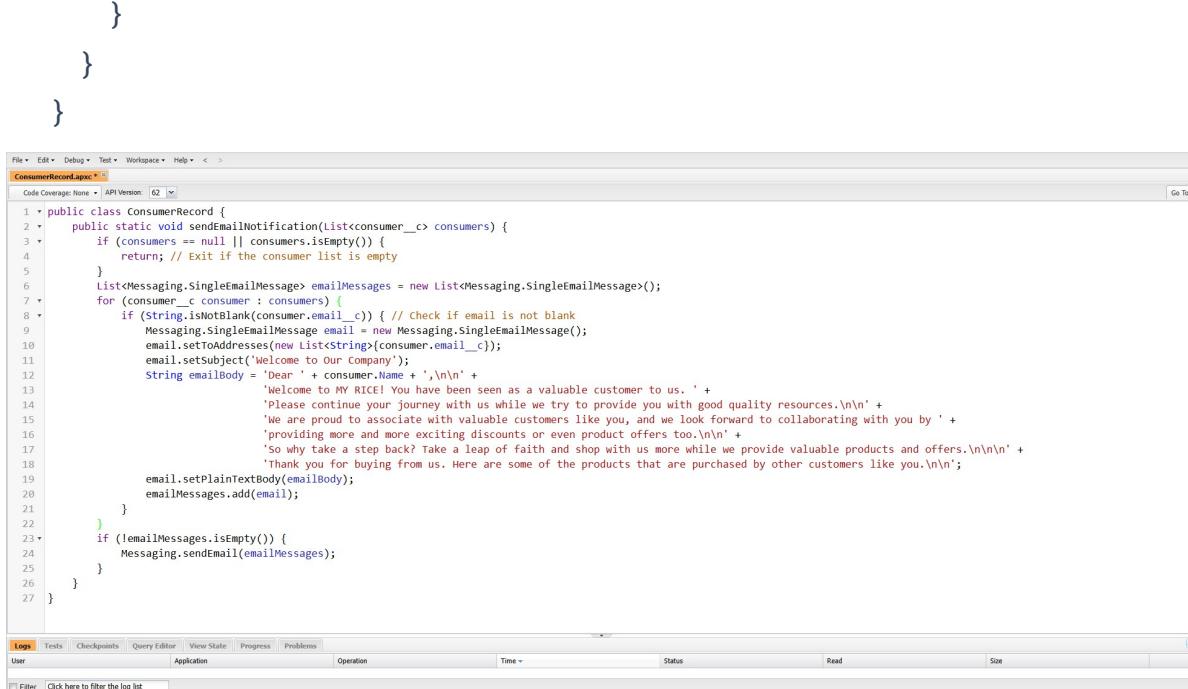
```
class ConsumerRecord {  
    public static void sendEmailNotification (List<consumer_  
        c> con){ for(consumer_c c:con)  
    {  
        Messaging.SingleEmailMessage email = new  
        Messaging.SingleEmailMessage(); email.setToAddresses( new  
        List<String>{c.email_c}); email.setSubject('Welcome to our  
        company');  
        email.setPlainTextBody('Dear ' + ''+ ',\n\nWelcome to MY RICE!'+ 'You have  
        been seen as a valuable customer to us. PLease continueyour journey with us, while  
        we try to provideyou 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<Messaging.SingleEmailMessage>{email});
```



```

1 public class ConsumerRecord {
2     public static void sendMailNotification(List<Consumer__c> consumers) {
3         if (consumers == null || consumers.isEmpty()) {
4             return; // Exit if the consumer list is empty
5         }
6         List<Messaging.SingleEmailMessage> emailMessages = new List<Messaging.SingleEmailMessage>();
7         for (Consumer__c consumer : consumers) {
8             if (!String.isBlank(consumer.email__c)) { // Check if email is not blank
9                 Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
10                email.setToAddresses(new List<String>{consumer.email__c});
11                email.setSubject('Welcome to Our Company');
12                String emailBody = 'Dear ' + consumer.Name + '\n\n' +
13                    'Welcome to MY RICE! You have been seen as a valuable customer to us. ' +
14                    'Please continue your journey with us while we try to provide you with good quality resources.\n\n' +
15                    'We are proud to associate with valuable customers like you, and we look forward to collaborating with you by ' +
16                    'providing more and more exciting discounts or even product offers too.\n\n' +
17                    'So why take a step back? Take a leap of faith and shop with us more while we provide valuable products and offers.\n\n' +
18                    'Thank you for buying from us. Here are some of the products that are purchased by other customers like you.\n\n';
19                email.setPlainTextBody(emailBody);
20                emailMessages.add(email);
21            }
22        }
23        if (!emailMessages.isEmpty()) {
24            Messaging.sendEmail(emailMessages);
25        }
26    }
27 }

```

## 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

### a. Unit Testing:

- i. Apex classes and triggers tested to ensure they handle edge cases and return accurate results.
  
- ii. Achieved >90% code coverage to meet Salesforce standards.

### b. User Interface Testing:

- i. Validated all forms and pages across different browsers and devices.
- ii. Ensured consistent user experience and accurate data presentation.

### c. 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<consumer_c>
con){ for(consumer_c c:con)
{
    Messaging.SingleEmailMessage email = new
    Messaging.SingleEmailMessage(); email.setToAddresses( new
    List<String>{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'+
    'Thank you 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<Messaging.SingleEmailMessage>{email});

}
}
}

Trigger code:
trigger consumerTrigger on consumer_c (After insert) {
    if(trigger.isAfter && trigger.isInsert) {
        ConsumerRecord.sendEmailNotification(trigger.new);
    }
}

```

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < ▾

ConsumerRecord.apc \* 62

Code Coverage: None API Version: 62 Go To

```
1 * public class ConsumerRecord {
2 *     public static void sendEmailNotification(List<consumer__c> consumers) {
3 *         if (consumers == null || consumers.isEmpty()) {
4 *             return; // Exit if the consumer list is empty
5 *         }
6 *         List<Messaging.SingleEmailMessage> emailMessages = new List<Messaging.SingleEmailMessage>();
7 *         for (consumer__c consumer : consumers) {
8 *             if (String.isNotBlank(consumer.email__c)) { // Check if email is not blank
9 *                 Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
10 *                 email.setToAddresses(new List<String>{consumer.email__c});
11 *                 email.setSubject('Welcome to Our Company');
12 *                 String emailBody = 'Dear ' + consumer.Name + ',\n\n' +
13 *                     'Welcome to MY RICE! You have been seen as a valuable customer to us. ' +
14 *                     'Please continue your journey with us while we try to provide you with good quality resources.\n\n' +
15 *                     'We are proud to associate with valuable customers like you, and we look forward to collaborating with you by ' +
16 *                     'providing more and more exciting discounts or even product offers too.\n\n' +
17 *                     'So why take a step back? Take a leap of faith and shop with us more while we provide valuable products and offers.\n\n' +
18 *                     'Thank you for buying from us. Here are some of the products that are purchased by other customers like you.\n\n';
19 *                 email.setPlainTextBody(emailBody);
20 *                 emailMessages.add(email);
21 *             }
22 *         }
23 *         if (!emailMessages.isEmpty()) {
24 *             Messaging.sendEmail(emailMessages);
25 *         }
26 *     }
27 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
Filter	Click here to filter the log list					

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < ▾

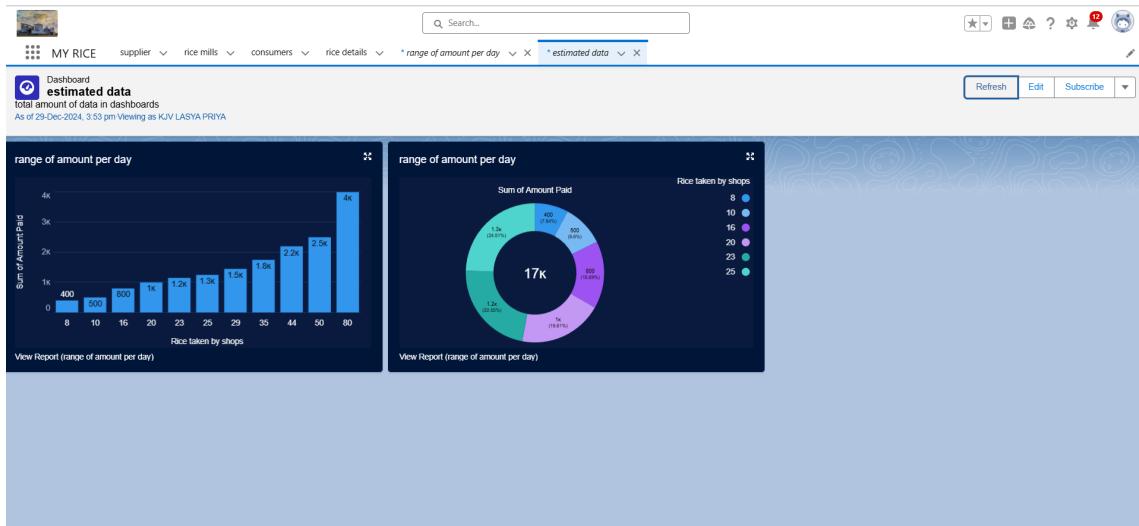
ConsumerRecord.apc consumerTrigger.apc

Code Coverage: None API Version: 62 Go To

```
1 * trigger consumerTrigger on consumer__c (before insert) {
2 *     if(trigger.isAfter && trigger.isInsert) {
3 *         ConsumerRecord.sendEmailNotification(trigger.new);
4 *     }
5 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
Filter	Click here to filter the log list					



**Report: rice mill with consumers**  
**range of amount per day**

Total Records: 11 Total rice price/kg: 500 Total Amount Paid: 17,000.00

Rice taken by shops	consumer Name	Rice type	rice price/kg	Mode of payment	Amount Paid
8 (1)	consumers-002	basmati	50	Cash	400.00
Subtotal					400.00
10 (1)	consumers-003	normal rice	50	Cash	500.00
Subtotal					500.00
15 (1)	consumers-009	basmati	50	Cash	800.00
Subtotal					800.00
20 (1)	consumers-006	normal rice	50	Cash	1,000.00
Subtotal					1,000.00
23 (1)	consumers-010	basmati	50	Cash	1,150.00
Subtotal					1,150.00
25 (1)	consumers-004	basmati	50	Cash	1,250.00
Subtotal					1,250.00
29 (1)	consumers-012	basmati	50	Net banking	1,450.00
Subtotal					1,450.00
35 (1)	consumers-013	normal rice	50	Net banking	1,750.00
Subtotal					1,750.00

Row Counts: Detail Rows: Subtotals: Grand Total:

**Gmail**

**Inbox** 46

Compose

Report results (range of amount per day)

KJV LASYA PRIYA via jd2w3b1c8p0uhg.qy-hrv3umaf.swe42.bnc.salesforce.com to me 8:00 AM (7 hours ago)

**range of amount per day**

As of 29/12/24 at 8:00 AM · Viewing as KJV LASYA PRIYA

[OPEN IN SALESFORCE](#)

**Details**

**Filters**

My rice mills Created Date: Current FQ

**Summary**

The screenshot shows a Gmail inbox with 46 unread messages. The main area displays a table of rice purchase details. The columns are: Rice taken by shops ↑, consumer Name, Rice type, rice price/kg Sum, Mode of payment, and Amount Paid Sum. The data includes various consumers (e.g., consumers-002, consumers-003) purchasing different rice types (e.g., basmati, normal rice) at different prices (e.g., 50) and using different modes of payment (e.g., Cash, Net banking).

Rice taken by shops ↑	consumer Name	Rice type	rice price/kg Sum	Mode of payment	Amount Paid Sum
8 (1 record)	consumers-002	basmati	50	Cash	400.00
			50		400.00
10 (1 record)	consumers-003	normal rice	50	Cash	500.00
			50		500.00
16 (1 record)	consumers-009	basmati	50	Cash	800.00
			50		800.00
20 (1 record)	consumers-006	normal rice	50	Cash	1,000.00
			50		1,000.00
23 (1 record)	consumers-010	basmati	50	Cash	1,150.00
			50		1,150.00
25 (1 record)	consumers-004	basmati	50	Cash	1,250.00
			50		1,250.00
29 (1 record)	consumers-012	basmati	50	Net banking	1,450.00
			50		1,450.00
35 (1 record)	consumers-013	normal rice	50	Net banking	1,750.00
			50		1,750.00
44	consumers-011	normal rice	50	Cash	2,200.00

## 6 Key Scenarios Addressed by Salesforce in the Implementation Project

- Scenario 1:** Daily automated reports sent to owners, summarizing rice production, sales, and revenue trends.
- Scenario 2:** Suppliers' payment details, calculated automatically based on rice supplied and price per kilogram.
- Scenario 3:** Role-based dashboards enabling owners, employers, and workers to view relevant data while maintaining security.
- Scenario 4:** Alerts for low inventory levels to ensure timely restocking and prevent production delays.
- Scenario 5:** Insights into customer preferences and purchasing behavior for targeted sales strategies.

## 7 Conclusion

### Summary of Achievements:

The **CRM Application for Wholesale Rice Mill** successfully addresses key challenges by providing an integrated, automated, and user-friendly solution. Key accomplishments include:

- Automated workflows and reports, saving time and reducing manual errors.
- Enhanced decision-making through real-time dashboards and analytics.
- Streamlined data management with rollup summary and cross-object formula fields.

#### 4. Role-based access controls ensuring secure and efficient data handling.

This project demonstrates how Salesforce CRM can be customized to meet specific industry needs, setting the stage for scalable growth and operational excellence in the rice mill industry. It exemplifies the power of technology in transforming traditional business practices into efficient, data-driven processes.