

Mill Pro

"Streamline Rice Mill Management from Harvest to Delivery"

A CRM Application for Wholesale Rice Mill Management

Leveraging Salesforce to Enhance Efficiency and Analytics in Rice Mill Management

Presented by: Yogitha Rajulapati

Date: 18-01-2025

Demo Video: [Video Link](#)

Document Link: [Drive Link](#)

GitHub Repository: <https://github.com/yogithaR26/Mill-Pro.git>

Abstract

This document provides an overview of a Salesforce CRM application built to improve the operational efficiency of rice mills. The application utilizes Salesforce features like custom objects, roll-up summary fields, cross-object formulas, validation rules, and role-based permissions to manage rice production, supplier relationships, consumer data, and detailed reporting.

Project Overview

This project focuses on creating a Salesforce CRM Application for managing wholesale rice mill operations. Designed to centralize and simplify the management of inventory, suppliers, consumers, and detailed reporting, the project aims to streamline rice mill processes by leveraging the Salesforce platform. By implementing Salesforce's robust data management and reporting capabilities, this CRM solution improves operational efficiency, data accuracy, and resource allocation, supporting the long-term productivity and business goals of the rice mill industry.

Objectives

Business Goals:

- Develop an intuitive and accessible CRM platform to manage rice mill operations efficiently
- Streamline daily processes for inventory, supplier, and consumer management
- Optimize reporting and provide insightful analytics for strategic decision-making

Specific Outcomes:

- A user-friendly interface for tracking rice production, sales, and supplier interactions
 - Comprehensive dashboards and reports for real-time data insights
 - Validation rules, permission sets, and roll-up fields to ensure data integrity and secure access
-


Salesforce Key Features and Concepts Utilized

- **Custom Objects:** Supplier, Rice Mill, Consumer, Rice Details
 - **Field Types:** Custom fields, roll-up summary fields, cross-object formula fields
 - **Relationships:** Master-detail relationships to link related data.
 - **User Interface:** Custom tabs, page layouts, and a Lightning App for organized navigation
 - **Security and Permissions:** Validation rules, permission sets, and role hierarchy for controlled access
 - **Reports and Dashboards:** Interactive and detailed analytics for monitoring business performance
-

Detailed Steps to Solution Design


Step 1: Setting Up Salesforce Environment

- **Create a Salesforce Developer Account:**
 - Go to Salesforce Developer Sign-Up and sign up for a free account
 - Verify your email and log in to the Salesforce Developer Console, where you will build and configure the CRM application
- **Configure Salesforce Settings:**
 - Navigate to the Setup menu to configure your Salesforce environment
 - Familiarize yourself with the Object Manager and App Manager sections, where you will create custom objects, fields, and Lightning apps for this project



Build enterprise-quality apps fast to bring your ideas to life

- Build apps fast with drag and drop tools
- Customize your data model with clicks
- Go further with Apex code
- Integrate with anything using powerful APIs
- Stay protected with enterprise-grade security
- Customize UI with clicks or any leading-edge web framework



Sign up for your Salesforce Developer Edition

A Salesforce Platform environment for free.

Complete the form to get access to the Salesforce Developer Edition.

First Name*
Yogitha

Last Name*
Rajulapati


Email*
21pa1a5494@vishnu.edu.in

Role*
Developer

Company*
Vishnu Institute of Technology

Country/Region*
India

State/Province*
Andhra Pradesh



Please check your username and password. If you still can't log in, contact your Salesforce administrator.

Username
yogitha2025@vit.com

Password

Log In

☐ Remember me


[Forgot Your Password?](#)

© 2025 Salesforce, Inc. All rights reserved.

Why Slack is the choice for pioneers of change

How businesses are leading the way with automation, productivity and connection.

[Download the playbook](#)



Step 2: Designing Custom Objects

- Create Custom Objects:

- In Setup, go to Object Manager > Create > Custom Object
- Create the following objects:
 - **Supplier:** Store details for each rice supplier, including their contact information and supply history
 - **Rice Mill:** Capture operational data for the rice mill
 - **Consumer:** Track consumer information and their purchase details
 - **Rice Details:** Record rice production metrics, pricing, and transaction details
- **Define Object Properties:**
 - Enable options such as Allow Reports and Allow Activities for each object to ensure that these objects can be included in reports and analytics
 - Add custom icons for each object to improve the user interface

SETUP > OBJECT MANAGER

Supplier

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

Details

Description

API Name
Supplier__c

Custom
✓

Singular Label
Supplier

Plural Label
Suppliers

Enable Reports
✓

Track Activities

Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

Edit Delete

SETUP > OBJECT MANAGER

Rice Mill

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

Details

Description

API Name
Rice_Mill__c

Custom
✓

Singular Label
Rice Mill

Plural Label
Rice Mill

Enable Reports
✓

Track Activities

Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

Edit Delete

SETUP > OBJECT MANAGER

Consumer

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

Details

Edit Delete

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

<https://vishnuinstituteoftechno186-dev-ed.develop.lightning.force.com/one/one.app#/setup/ObjectManager/01dM000002x1N7/FieldsAndRelationships/view>

SETUP > OBJECT MANAGER

Rice Details

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

Details

Edit Delete

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

<https://vishnuinstituteoftechno186-dev-ed.develop.lightning.force.com/one/one.app#/setup/ObjectManager/01dM000002x0fa/FieldsAndRelationships/view>

Step 3: Establishing Relationships

- **Master-Detail Relationships:**

- In Object Manager, open each object and go to Fields & Relationships > New
 - Select Master-Detail Relationship and choose the related object for each primary object
- For example, create a master-detail relationship between Rice Mill and Rice Details to show rice production details for each mill

- **Lookup Relationships:**

- Use Lookup Relationships for fields that need optional relationships, such as linking Consumer to Supplier if required

SETUP > OBJECT MANAGER

Consumer

Details

Fields & Relationships

14 Items, Sorted by Field Label

Q, Quick Find

New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount Paid	Amount_Paid__c	Formula (Number)		
Consumer Name	Consumer_Name__c	Formula (Text)		
Consumer Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
First Name	First_Name__c	Text(255)		
Last Modified By	LastModifiedById	Lookup(User)		
Last Name	Last_Name__c	Text(255)		

SETUP > OBJECT MANAGER

Supplier

Details

Fields & Relationships

5 Items, Sorted by Field Label

Q, Quick Find

New Deleted Fields Field Dependencies Set History Tracking

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)		✓

Step 4: Creating Fields

- **Create Custom Fields:**

- In Fields & Relationships within each object, select New and choose field types like Number, Currency, or Text to store data
- For example, in Rice Details, create fields such as Quantity, Price per KG, and Total Cost

- **Implement Roll-Up Summary Fields:**

- Use roll-up summary fields in Supplier to aggregate data from related records, like the total quantity of rice supplied
- For each roll-up field, choose a function such as SUM to calculate totals across records
- **Cross-Object Formula Fields:**
 - Create formula fields that pull data from related objects to perform calculations

For instance, add a formula to Rice Details to calculate,

$$\text{Total Payment} = \text{Quantity} * \text{Price per KG.}$$
- Formula fields provide real-time calculations, so no additional updates are required when underlying fields change

SETUP > OBJECT MANAGER

Rice Mill

Details

Fields & Relationships
7 Items, Sorted by Field Label

Q Quick Find

New Deleted Fields Field Dependencies Set History Tracking

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)		✓
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)		▼

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

Setup Home Object Manager

Search Setup

SETUP > OBJECT MANAGER

Rice Details

Details

Fields & Relationships
6 Items, Sorted by Field Label

Q Quick Find

New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Rice Details Name	Name	Auto Number		✓ ▼
Rice Distributed	Rice_Distributed__c	Number(5, 0)		▼
Rice Mill 1	Rice_Mill_1__c	Master-Detail(Rice Mill)		✓ ▼
Supplier Name	Supplier__c	Master-Detail(Supplier)		✓ ▼

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

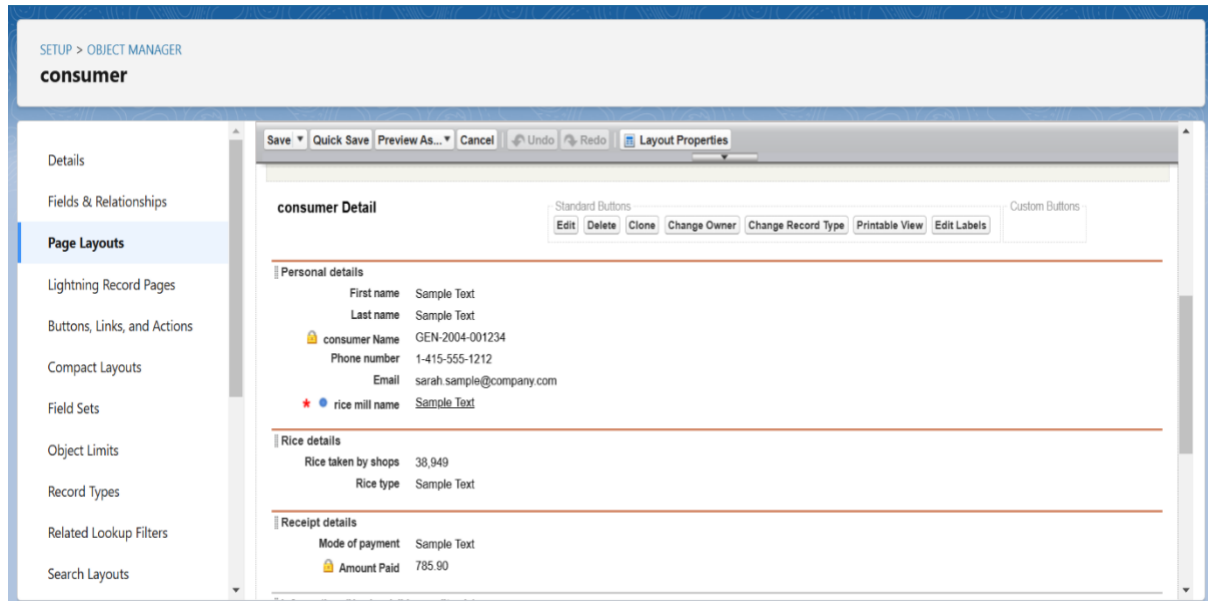
Search Layouts

List View Button Layout

Step 5: User Interface Customization

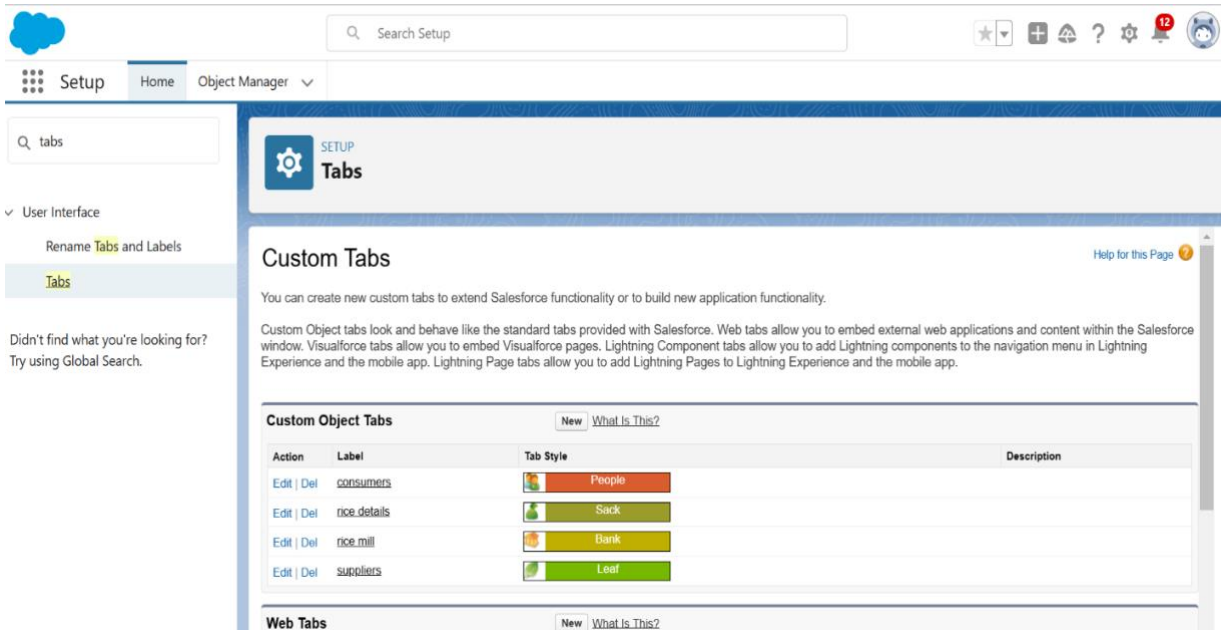
- **Develop Page Layouts:**

- Customize the page layouts for each object to display essential fields first and create sections for organized data viewing
- Adjust the layout by dragging and arranging fields in a logical order. Add sections like Supplier Information, Contact Details, and Supply History



- **Create Custom Tabs:**

- In Setup, go to Tabs > New and select the objects you have created
- Name each tab to represent the object, like Rice Mill or Supplier, and choose icons for easy identification



- **Build a Lightning App:**

- In App Manager, create a new Lightning App for this CRM
- Add the custom tabs and adjust navigation to simplify access to each object

The screenshot shows the 'App Settings' page for a Lightning App named 'MY RICE'. The page is divided into two main sections: 'App Details & Branding' and 'App Branding'.

App Details & Branding:

- App Name:** MY RICE
- Developer Name:** MY_RICE
- Description:** Streamline Rice Mill Management from Harvest to Delivery

App Branding:

- Image:** A logo featuring three stalks of rice and a green banner.
- Primary Color Hex Value:** #0070D2
- Org Theme Options:** ☐ Use the app's image and color instead of the org's custom theme
- App Launcher Preview:** A preview of the app launcher showing the app icon and name 'MY RICE' with the description 'Streamline Rice Mill Management from Harvest to Delivery'.

Step 6: Implementing Security Controls

- **Define Validation Rules:**

- In each object's Validation Rules section, create rules to ensure correct data input. For example, create a rule in Rice Details to prevent blank values in fields like Quantity
- Use error messages in each rule to notify users of invalid entries, which ensures data consistency

- **Configure Permission Sets:**

- Go to Permission Sets in Setup to define permissions for different users
- For instance, create a *Supplier Manager* permission set that allows only specific access to supplier-related data


- **Set Up Role Hierarchies and Profiles:**

- Define a Role Hierarchy in Setup > Roles to manage data visibility between different levels.
- Create profiles and assign them to roles, such as an Admin profile with access to all records or a Sales Rep profile with limited permissions

Step 7: Reporting and Dashboards

- **Create Reports:**

- In Reports, create custom reports to monitor rice production and supplier contributions
- Use filters to narrow down metrics, such as sales performance by month or by supplier. Include fields like Total Quantity Sold and Revenue Generated

 Report: rice mill with consumers
range of amount per day

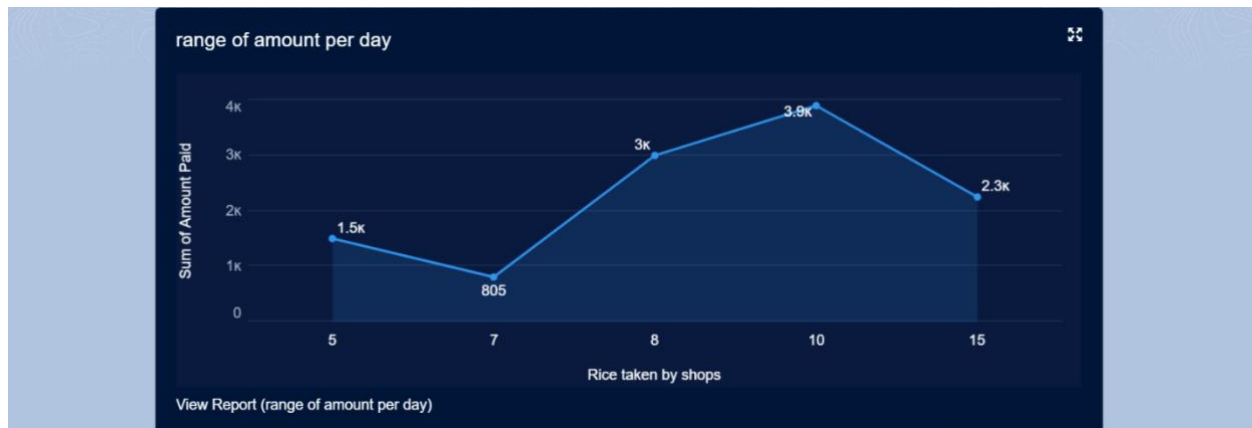
Enable Field Editing

Total Records	Total rice price/kg	Total Amount Paid				
10	640	11,455.00				
<input type="checkbox"/> Rice taken by shops ↑	consumer: consumer Name	Rice type	rice price/kg	Mode of payment	Amount Paid	
<input type="checkbox"/> 5 (2)	consumers-001	basmati	150	UPI	750.00	
	consumers-006	basmati	150	UPI	750.00	
Subtotal			300		1,500.00	
<input type="checkbox"/> 7 (1)	consumers-008	basmati	115	Debit card	805.00	
			115		805.00	
<input type="checkbox"/> 8 (3)	consumers-003	basmati	150	Net banking	1,200.00	
	consumers-005	basmati	135	Debit card	1,080.00	
	consumers-010	normal rice	90	UPI	720.00	
Subtotal			375		3,000.00	
<input type="checkbox"/> 10 (3)	consumers-002	normal rice	90	Cash	900.00	
	consumers-004	normal rice	150	Credit card	1,500.00	
	consumers-009	normal rice	150	Cash	1,500.00	
Subtotal			390		3,900.00	
<input type="checkbox"/> 15 (1)	consumers-007	normal rice	150	Net banking	2,250.00	
			150		2,250.00	
Total (10)			640		11,455.00	

Row Counts ☒ Detail Rows ☒ Subtotals ☒ Grand Total ☒

- **Design Dashboards:**

- Go to Dashboards and select New Dashboard. Customize dashboard components to display KPIs
- Add widgets like charts and graphs to represent key metrics visually. For instance, a bar chart for monthly sales volume or a pie chart for sales distribution by consumer type



Step 8: Testing and Validation

- **Unit Testing:**
 - Test all Apex Classes and Triggers for the CRM's business logic, like automated calculations or error notifications
 - Run tests within the Developer Console to ensure error-free operations
- **User Interface Testing:**
 - Test page layouts, Lightning app navigation, and validation rules to confirm accuracy and functionality
 - Perform final checks in multiple browsers to ensure compatibility

Developer Console - Google Chrome

vishnuinstituteoftechno186-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

DashboardDataController.apxc ConsumerRecord.apxc

Code Coverage: None API Version: 62 Go To

```
1 public class ConsumerRecord {
2     public static void sendEmailNotification(List<consumer__c> consumers) {
3         List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();
4
5         for (consumer__c consumer : consumers) {
6             Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
7             email.setToAddresses(new List<String>{consumer.email__c});
8             email.setSubject('Welcome to our company');
9
10            // Use the correct field names for first and last names
11            String emailBody = 'Dear ' + consumer.First_Name__c + ' ' + consumer.Last_Name__c + ',\n\n' +
12                               'Welcome to MY RICE! You have been seen as a valuable customer to us. Please continue your journey wit
13                               'We are proud to associate with valuable customers like you and look forward to collaborating with you
14                               'So why take a step back? Take a leap of faith and shop with us more, as we provide valuable products
15                               'Thank you for being a customer! Here are some of the products that are popular among our customers:\n
16
17            email.setPlainTextBody(emailBody);
18            emails.add(email); // Add email to the list
19        }
20    }
21 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem
------	------	---------

Developer Console - Google Chrome

vishnuinstituteoftechno186-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

DashboardDataController.apxc ConsumerRecord.apxc consumerTrigger.apxt

Code Coverage: None API Version: 62 Go To

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

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem
------	------	---------

Lightning Web Component (LWC) Implementation

Overview

As part of the **Mill Suite** application, a Lightning Web Component (LWC) has been implemented to enhance the user interface and provide dynamic functionalities. This component allows users to interact seamlessly with the Salesforce data related to rice mill operations.

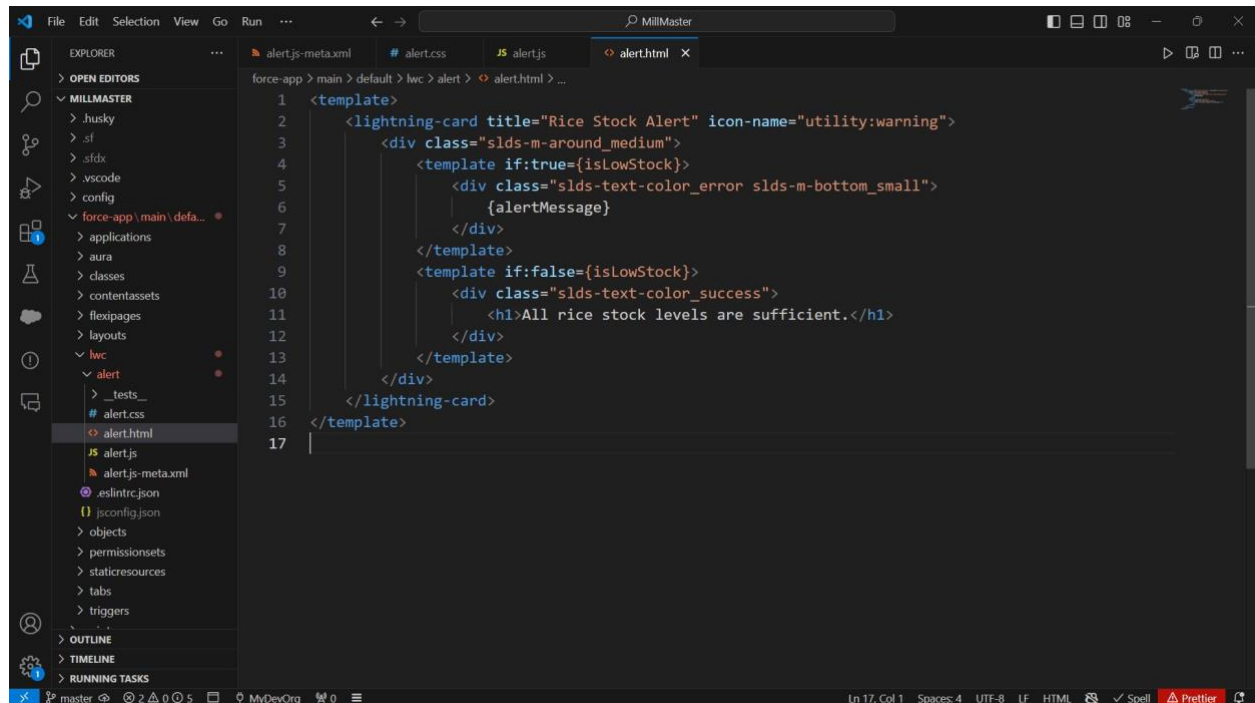
Component Purpose

The LWC serves the following purposes:

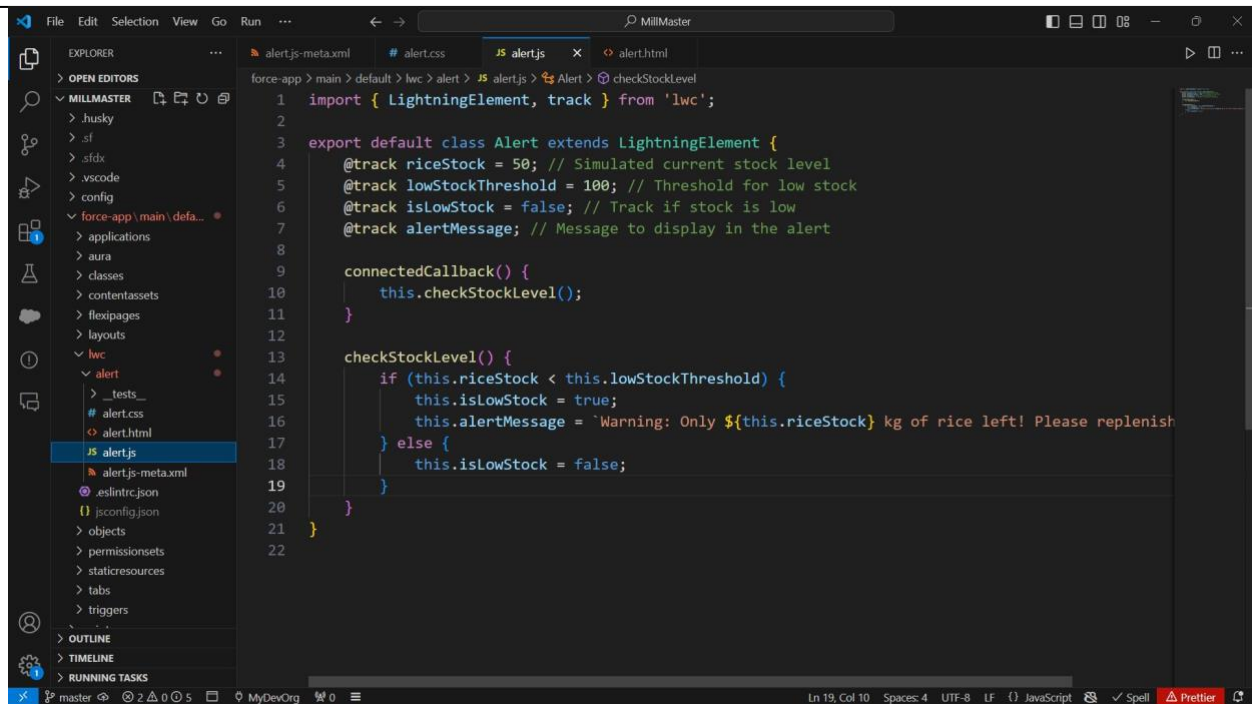
- **User Interaction:** Facilitates easy navigation and interaction with rice mill data
- **Dynamic Data Display:** Shows real-time information about suppliers, rice details, and consumer data
- **Enhanced User Experience:** Provides a modern interface that improves usability and accessibility for users with different technical backgrounds

Component Code

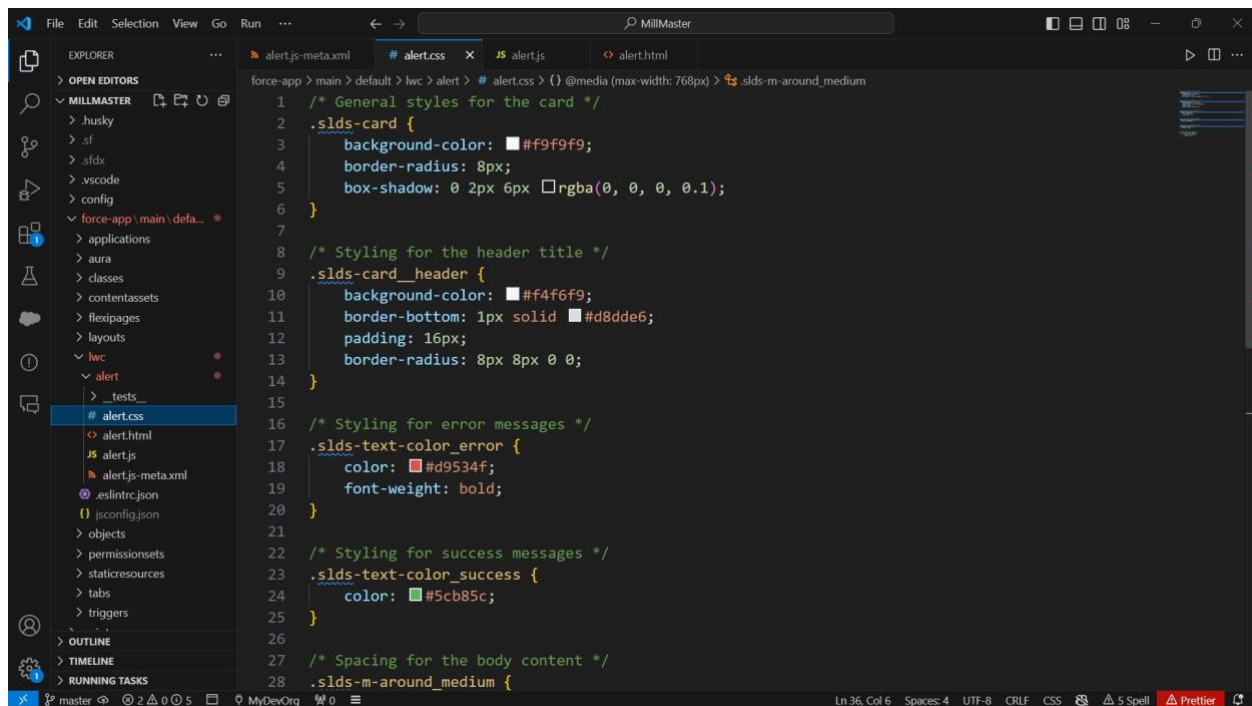
Here is the code for the implemented Lightning Web Component:

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project structure for 'MILLMASTER' with folders like 'husky', 'sf', 'sfdx', 'vscode', 'config', and 'force-app'. The 'force-app' folder is expanded, showing 'main' > 'default' > 'lwc' > 'alert'. The 'alert' folder contains files: '_tests_', 'alert.css', 'alert.html' (selected), 'alert.js', 'alert.js-meta.xml', '.eslintrc.json', 'jsconfig.json', 'objects', 'permissionsets', 'staticresources', 'tabs', and 'triggers'. The main editor area displays the content of 'alert.html'. The code is an LWC template using the 'lightning-card' component. It has a title 'Rice Stock Alert' and a warning icon. It uses a template to conditionally display either an error message (if 'isLowStock' is true) or a success message (if 'isLowStock' is false). The success message is 'All rice stock levels are sufficient.'.

```
1 <template>
2   <lightning-card title="Rice Stock Alert" icon-name="utility:warning">
3     <div class="slds-m-around_medium">
4       <template if:true={isLowStock}>
5         <div class="slds-text-color_error slds-m-bottom_small">
6           {alertMessage}
7         </div>
8       </template>
9       <template if:false={isLowStock}>
10        <div class="slds-text-color_success">
11          <h1>All rice stock levels are sufficient.</h1>
12        </div>
13      </template>
14    </div>
15  </lightning-card>
16 </template>
17
```

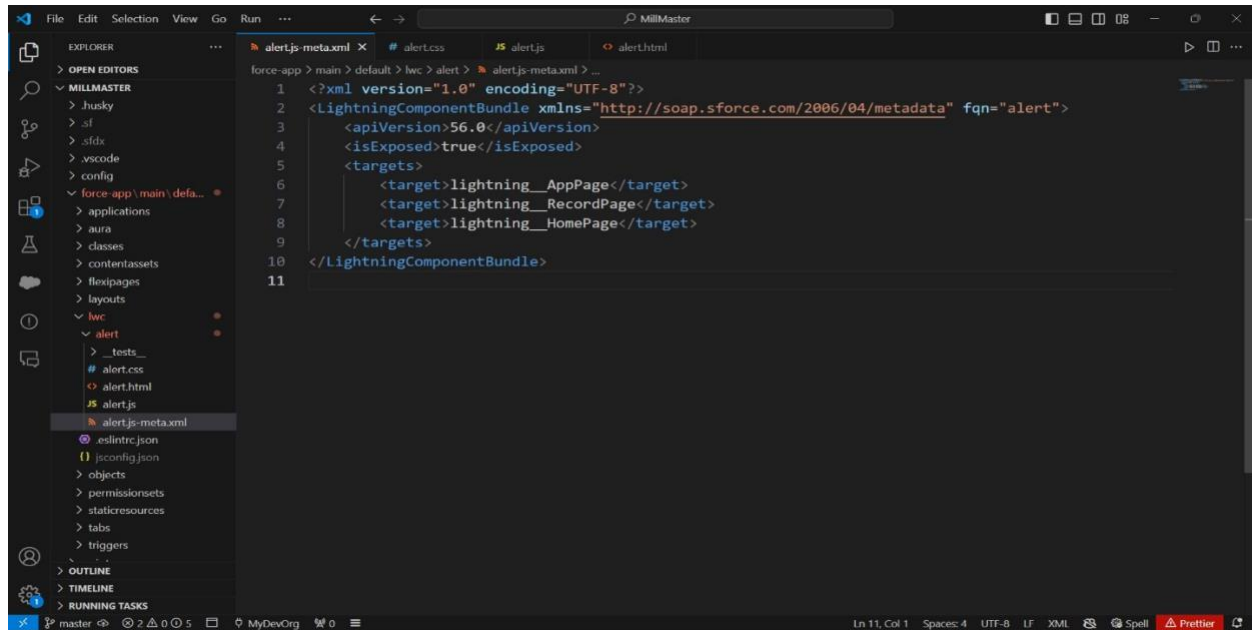


```
1 import { LightningElement, track } from 'lwc';
2
3 export default class Alert extends LightningElement {
4   @track riceStock = 50; // Simulated current stock level
5   @track lowStockThreshold = 100; // Threshold for low stock
6   @track isLowStock = false; // Track if stock is low
7   @track alertMessage; // Message to display in the alert
8
9   connectedCallback() {
10     this.checkStockLevel();
11   }
12
13   checkStockLevel() {
14     if (this.riceStock < this.lowStockThreshold) {
15       this.isLowStock = true;
16       this.alertMessage = `Warning: Only ${this.riceStock} kg of rice left! Please replenish`;
17     } else {
18       this.isLowStock = false;
19     }
20   }
21 }
22
```



```
1 /* General styles for the card */
2 .slds-card {
3   background-color: #f9f9f9;
4   border-radius: 8px;
5   box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
6 }
7
8 /* Styling for the header title */
9 .slds-card_header {
10   background-color: #f4f6f9;
11   border-bottom: 1px solid #d8dde6;
12   padding: 16px;
13   border-radius: 8px 8px 0 0;
14 }
15
16 /* Styling for error messages */
17 .slds-text-color_error {
18   color: #d9534f;
19   font-weight: bold;
20 }
21
22 /* Styling for success messages */
23 .slds-text-color_success {
24   color: #5cb85c;
25 }
26
27 /* Spacing for the body content */
28 .slds-m-around_medium {

```



Key Scenarios Addressed by Salesforce in the Implementation Project:

- **Supplier Management:** Maintain comprehensive supplier profiles and track the quantity of rice supplied
- **Inventory Tracking:** Monitor rice inventory levels, distribution, and sales patterns
- **Consumer Engagement:** Store consumer purchasing details and calculate billing based on quantity and rate
- **Reporting & Analytics:** Generate dynamic reports on rice production, distribution, and financial data for strategic planning
- **Dynamic User Interface with LWC Component:**
 - The implementation of the Lightning Web Component (LWC) enhances user interaction by providing a modern, dynamic interface for accessing and managing rice details
 - It allows users to view real-time data on rice types, quantities, and pricing, improving data visibility and usability
 - The LWC efficiently integrates with Salesforce data, ensuring that users have instant access to the latest information while maintaining a seamless experience across the application

Conclusion

The Salesforce CRM application successfully optimizes rice mill operations by centralizing supplier, inventory, and consumer data. This comprehensive solution supports efficient resource allocation and enhances decision-making with real-time analytics and reporting.

References

- Salesforce Documentation
- Salesforce Community