

# **KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY**



**Domain: Salesforce Developer**

**Branch: Information Technology**

**Batch:1**

Sekuri Manju Bhargavi    [218x1a1232@khitguntur.ac.in](mailto:218x1a1232@khitguntur.ac.in)    [manjubhargavi5579@gmail.com](mailto:manjubhargavi5579@gmail.com)

Bandaru Kesava Ram    [228x5a1202@khitguntur.ac.in](mailto:228x5a1202@khitguntur.ac.in)    [kesavaramwork@gmail.com](mailto:kesavaramwork@gmail.com)

Punati Anil Kumar    [208x1a1231@khitguntur.ac.in](mailto:208x1a1231@khitguntur.ac.in)    [anilkumarpunati1234@gmail.com](mailto:anilkumarpunati1234@gmail.com)

Shaik Balaji    [208x1a1263@khitguntur.ac.in](mailto:208x1a1263@khitguntur.ac.in)    [skbalaji795@gmail.com](mailto:skbalaji795@gmail.com)

# **Project Title: Pizza Delivery Application**

**1. Project Overview:** Pizza delivery is a service in which a pizzeria or pizza chain delivers a pizza to a customer. An order is typically made either by telephone, or over the internet, in which the customer can request pizza type and size, and other items to be delivered with it, commonly including soft drinks. Pizzas may be delivered in cardboard pizza boxes or delivery bags, and deliveries are made with either an automobile, motorized scooter, or bicycle.

## **2. Objectives:**

### **Business Goals:**

#### **1. Increase Sales:**

- Achieve a 20% increase in orders in the first 3 months.
- Generate \$250,000 in revenue in the first 6 months.

#### **2. Customer Retention:**

- Retain 75% of customers after their first order.
- Increase repeat orders by 30% using personalized marketing.

#### **3. Operational Efficiency:**

- Reduce delivery time by 15% within 6 months.
- Cut customer support response time by 20%.

### **Specific Outcomes:**

#### **1. Salesforce Integration:**

- Integrate the app with Salesforce to track customer data and orders.
- Sync customer information between the app and Salesforce.

#### **2. Order Tracking:**

- Use Salesforce to track real-time order updates and delivery status.
- Allow customers to track their orders in real-time via the app.

### **3. Customer Engagement:**

- Set up personalized marketing using Salesforce to send targeted promotions.
- Use Salesforce to send automated messages for new offers or discounts.

### **4. Customer Support:**

- Use Salesforce to manage customer service requests and resolve issues within 24 hours.
- Implement a chatbot for quick responses to common questions.

### **5. Analytics & Reporting:**

- Use Salesforce to track key metrics like sales, customer satisfaction, and delivery times.
- Generate monthly reports on app performance and customer trends.

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

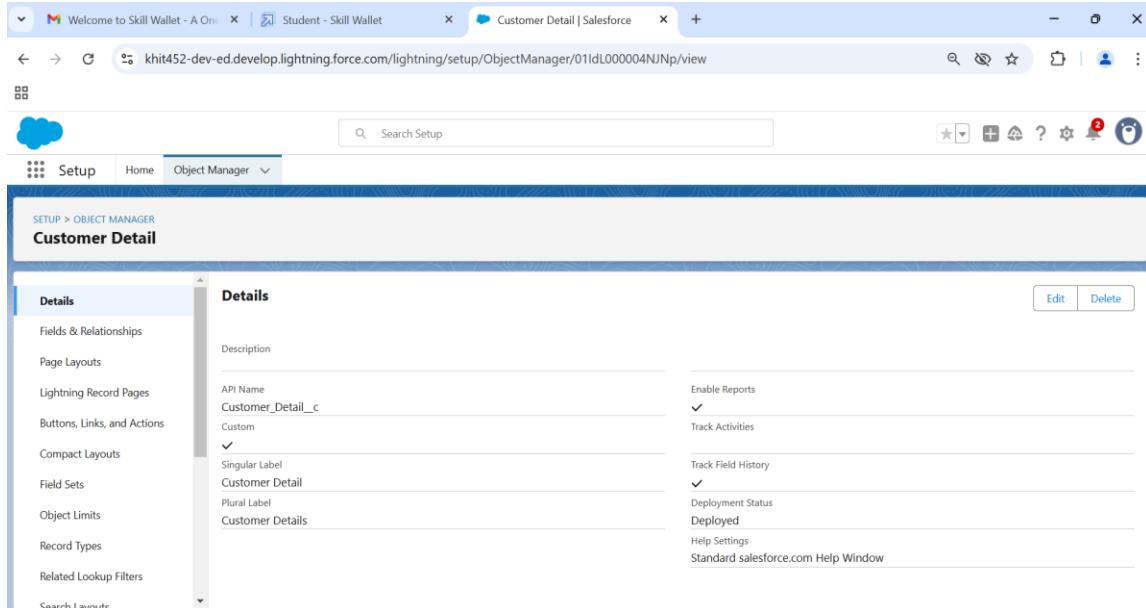
- ❖ Object
- ❖ Tabs
- ❖ Create App
- ❖ Fields & Relationships
- ❖ Record Types
- ❖ Profile
- ❖ User Adoption
- ❖ OWD (Organization Wide Default)
- ❖ Reports
- ❖ Apex Trigger
- ❖ Schedule Apex

## 4.Detailed Steps to Solution Design Objects

### Create a custom object for Customer Detail

#### To create an object follow

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



Similarly do the same process for other objects also

## Custom object for Every Day Value

The screenshot shows the Salesforce Object Manager interface for the 'EveryDay Value' custom object. The left sidebar lists various setup options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main details pane shows the following configuration:

Details	
Description	
API Name	EveryDay_Value__c
Custom	✓
Singular Label	EveryDay Value
Plural Label	EveryDay Values
Enable Reports	✓
Track Activities	
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

At the top right of the details pane are 'Edit' and 'Delete' buttons.

## Custom object for Best Seller

The screenshot shows the Salesforce Object Manager interface for the 'Best Seller' custom object. The left sidebar lists various setup options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main details pane shows the following configuration:

Details	
Description	
API Name	Best_Seller__c
Custom	✓
Singular Label	Best Seller
Plural Label	Best Sellers
Enable Reports	✓
Track Activities	
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

At the top right of the details pane are 'Edit' and 'Delete' buttons.

## Custom object for New Launches

The screenshot shows the Salesforce Setup interface for creating a new object. The left sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Lightning Record Pages. The main 'Details' section on the right shows the following configuration for the 'New\_Launch\_c' object:

- Description:** [empty]
- API Name:** `New_Launch_c`
- Custom:**
- Singular Label:** `New Launch`
- Plural Label:** `New Launches`
- Enable Reports:**
- Track Activities:** [empty]
- Track Field History:**
- Deployment Status:** `Deployed`
- Help Settings:** `Standard salesforce.com Help Window`

At the bottom right are 'Edit' and 'Delete' buttons.

## Custom object for Pizza

The screenshot shows the Salesforce Setup interface for creating a new object. The left sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Lightning Record Pages. The main 'Details' section on the right shows the following configuration for the 'Pizza\_c' object:

- Description:** [empty]
- API Name:** `Pizza_c`
- Custom:**
- Singular Label:** `Pizza`
- Plural Label:** `Pizzas`
- Enable Reports:**
- Track Activities:** [empty]
- Track Field History:**
- Deployment Status:** `Deployed`
- Help Settings:** `Standard salesforce.com Help Window`

At the bottom right are 'Edit' and 'Delete' buttons.

## Custom object for Beverages

The screenshot shows the Salesforce Setup interface for creating a new object named 'Beverage'. The left sidebar lists various configuration tabs: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main 'Details' tab displays the following settings:

- Description: (empty)
- API Name: Beverage\_c (Custom checked)
- Singular Label: Beverage
- Plural Label: Beverages
- Enable Reports: ✓
- Track Activities: (empty)
- Track Field History: ✓
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

The URL in the browser bar is <https://khit452-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lL000004Njdx/Details/view>.

## Custom object for Mode of Payment

The screenshot shows the Salesforce Setup interface for creating a new object named 'Mode Of Payment'. The left sidebar lists various configuration tabs: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main 'Details' tab displays the following settings:

- Description: (empty)
- API Name: Mode\_of\_Payment\_c (Custom checked)
- Singular Label: Mode Of Payment
- Plural Label: Mode Of Payments
- Enable Reports: ✓
- Track Activities: (empty)
- Track Field History: ✓
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

The URL in the browser bar is <https://khit452-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lL000004NjfZ/Details/view>.

# Tabs

## How to create a tab

1. To Create the tab? click on the gear Icon? then click Home button? In the quick find box search for the Tabs.
2. Then Click on the new button to create tab, in the Tab style you can select whatever you want to select
3. Than select the Customer Detail Object and select the Icon from the search button and click next and save
4. Repeat the same Process to create the Remaining Tabs such as,
  - Every Day Value
  - Best Seller
  - New Launches
  - Pizzas
  - Beverages
  - Mode of Payment

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page open. The URL in the browser is [khit452-dev.develop.lightning.force.com/lightning/setup/CustomTabs/home](https://khit452-dev.develop.lightning.force.com/lightning/setup/CustomTabs/home). The page title is 'Custom Tabs'. It includes a 'Custom Object Tabs' section and a 'Web Tabs' section. The 'Custom Object Tabs' section lists several tabs with their labels and icons:

Action	Label	Tab Style	Description
Edit   Del	Best Sellers	Building	
Edit   Del	Beverages	Dice	
Edit   Del	Customer Details	Bottle	
Edit   Del	EveryDay Values	Airplane	
Edit   Del	Mode Of Payments	Castle	
Edit   Del	New Launches	Books	
Edit   Del	Pizzas	Apple	

The 'Web Tabs' section is currently empty.

# Create App

## Create the Lightning App

1. Go to the Gear Icon? Click on setup? click on the Home Button? Go to the quick find box and select the App Manager
2. Fill the app name as Pizza Delivery in app details and branding? Next? App option page keep it as default? Next
3. Add Navigation Items add tabs Customer Details, Every Day Values, Beverages, Best Seller, New Launches, pizza and Mode of payment? Next? (Add User Profile) Add System Administrator, Salesforce platform user, Standard User? Next.
4. To Add Navigation Items: Select the items from the search bar and move it using the arrow button? Next. select all the tabs which you have created
5. To Add User Profiles: Search profiles in search bar? click on the arrow button & select Standard user, standard Platform user & System Admin Profile? save & finish

The screenshot shows the Salesforce App Manager interface. The left sidebar navigation includes sections like Data, Apps (with App Manager selected), Connected Apps, External Client Apps, and Lightning Bolt. The main content area displays a table titled "Lightning Experience App Manager" with 26 items. The columns are App Name, Developer Name, Description, Last Modified Date, App Type, and Visibility. The table lists various Salesforce applications such as All Tabs, Analytics Studio, App Launcher, Approvals, Automation, Bolt Solutions, Business Rules Engine, Community, Content, Data Manager, Digital Experiences, and Lightning Usage App. The "App Type" column indicates whether the app is Classic or Lightning.

App Name	Developer Name	Description	Last Modified Date	App Type	Visibility
All Tabs	AllTabSet	Build CRM Analytics dashboards and apps	24/02/2025, 8:18 pm	Classic	✓
Analytics Studio	Insights	Build CRM Analytics dashboards and apps	24/02/2025, 8:18 pm	Classic	✓
App Launcher	AppLauncher	App Launcher tabs	24/02/2025, 8:18 pm	Classic	✓
Approvals	Approvals	Manage approvals and approval flows	24/02/2025, 8:18 pm	Lightning	✓
Automation	FlowsApp	Automate business processes and repetitive tasks.	24/02/2025, 8:22 pm	Lightning	✓
Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	24/02/2025, 8:21 pm	Lightning	✓
Business Rules Engine	ExpressionSetConsole	Create and maintain business rules that perform complex lookups and calculations.	24/02/2025, 8:18 pm	Lightning	✓
Community	Community	Salesforce CRM Communities	24/02/2025, 8:18 pm	Classic	✓
Content	Content	Salesforce CRM Content	24/02/2025, 8:18 pm	Classic	✓
Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	24/02/2025, 8:18 pm	Lightning	✓
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	24/02/2025, 8:18 pm	Lightning	✓
Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	24/02/2025, 8:18 pm	Lightning	✓

# Fields & Relationships

## Create the Field for Every Day Value

1. Go to setup click on Object Manager type object name in search bar click on the object? Every Day Value? Field and Relationship? then click on new Field Data Type Checkbox
2. Go to setup click on Object Manager type object name in search bar click on the object Every Day Value Field and Relationship then click on new Field Data Type Checkbox
3. Follow the above steps and create the Remaining Fields.
  - Pickup Checkbox
  - Select City Picklist (New Delhi, Mumbai, Bangalore, Pune, Chennai, Hyderabad, Kolkata, Noida, Agra, Ahmad Nagar, Ahmedabad, Nasik)
  - Select Store Picklist
  - Delivery Checkbox
  - Pickup Checkbox

The screenshot shows the Salesforce Setup interface with the following details:

- Setup Path:** SETUP > OBJECT MANAGER > EveryDay Value
- Page Title:** Fields & Relationships
- Table Headers:** FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, INDEXED
- Table Data:** The table lists eight items, sorted by Field Label:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Delivery	Delivery__c	Checkbox		
EveryDay Value	Name	Text(80)		✓
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Pickup	Pickup__c	Checkbox		
Select City	Select_City__c	Picklist		
Select Store	Select_Store__c	Picklist		

## Create the Field for Best Seller Object

The screenshot shows the Salesforce Object Manager interface for the 'Best Seller' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Buttons, Links, and Actions. The main content area displays the 'Fields & Relationships' section with 5 items. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Best Seller	Name	Text(80)		✓
Created By	CreatedBy	Lookup(User)		
Crust	Crust_c	Picklist		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓

## Create the Field for New Launches

The screenshot shows the Salesforce Object Manager interface for the 'New Launch' object. The left sidebar lists various setup options. The main content area displays the 'Fields & Relationships' section with 5 items. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
New Beverages	New_Beverages_c	Picklist		
New Launch	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓

## Create the Field for Pizza

The screenshot shows the Salesforce Object Manager interface for the 'Pizza' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main content area is titled 'Fields & Relationships' and displays eight fields sorted by field label. The fields are:

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

## Create the Field for Beverages

The screenshot shows the Salesforce Object Manager interface for the 'Beverage' object. The left sidebar lists various setup options. The main content area is titled 'Fields & Relationships' and displays five fields sorted by field label. The fields are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Beverage	Name	Text(80)		✓
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Types of Cold Drinks	Types_of_Cold_Drinks__c	Picklist		

## Create the Field for Customer Detail

Types of Roasted Chicken picklist values:

1. Chicken taco pizza
2. Chicken fajita pizza
3. Chicken pesto pizza
4. Barbecue chicken pizza bread bowl

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Add To Cart	Add_To_Cart__c	Checkbox		
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		✓
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Types of Roasted Chicken	Types_of_Roasted_Chicken__c	Picklist		

## Create the Field for Mode of Payment

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Customer Detail	Customer_Detail__c	Master-Detail(Customer Detail)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Mode Of Payment	Mode_Of_Payment__c	Picklist		
Mode Of Payment	Name	Text(80)		✓

# Record Types

## Record Type for Pizza Object

1. Click on the Gear Icon? Then Click on the object Manager? Search for Pizza object?  
Record Types Click on new Veg Pizza

The screenshot shows the Salesforce Object Manager interface for the 'Pizza' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The 'Record Types' option is selected. In the main content area, a record named 'Veg Pizza' is displayed. The record type details include:

- Record Type Label: Veg Pizza
- Record Type Name: Veg\_Pizza
- Status: Active (checked)
- Created By: Manju Bhargavi Sekuri, 25/02/2025, 12:18 pm
- Modified By: Manju Bhargavi Sekuri, 25/02/2025, 12:32 pm

A section titled 'Picklists Available for Editing' lists two fields:

Action	Field	Modified Date
Edit	Amount	25/02/2025, 12:31 pm
Edit	Types of Pizzas	25/02/2025, 12:32 pm

The screenshot shows the Salesforce Object Manager interface for the 'Pizza' object, specifically editing the 'Types of Pizzas' record type. The 'Record Types' option is selected in the sidebar. The 'General Properties' section shows:

- Field Label: Types of Pizzas
- Record Type: Veg Pizza

In the 'Picklist Values' section, there are two lists:

- Available Values:** Paneer Chicken, Chicken pizza
- Selected Values:** Paneer Makhani, Margherita, Onion Pizza, Tomato pizza

Buttons for 'Add' and 'Remove' are visible between the two lists.

2. Similarly do for the Non-veg and just select the non-veg instead of veg

The screenshot shows the Salesforce Object Manager interface. The left sidebar is titled 'SETUP > OBJECT MANAGER' and lists various options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types (which is selected), Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules.

The main content area is titled 'Record Type Non-veg' and shows the following details:

- Record Type Label: Non-veg
- Record Type Name: Non\_veg
- Namespace Prefix:
- Description:
- Created By: Manu Bhargavi Sekuri, 25/02/2025, 12:20 pm
- Modified By: Manu Bhargavi Sekuri, 25/02/2025, 12:34 pm

Below this, there is a section titled 'Picklists Available for Editing' which lists two items:

Action	Field	Modified Date
Edit	Amount	25/02/2025, 12:31 pm
Edit	Types of Pizzas	25/02/2025, 12:34 pm

The screenshot shows the 'Types of Pizzas' picklist values for the Non-veg record type. The left sidebar is identical to the previous screenshot.

The main content area is titled 'Record Type Edit Types of Pizzas' and shows the 'General Properties' section with the following details:

- Field Label: Types of Pizzas
- Record Type: Non-veg

Below this is the 'Picklist Values' section, which contains the following information:

Select an item from the Available Values list and add it to the Selected Values list to include it as a picklist value for this Record Type. Note that removing an item from the picklist does not remove it from any existing records. Finally, select a default picklist value for this Record Type.

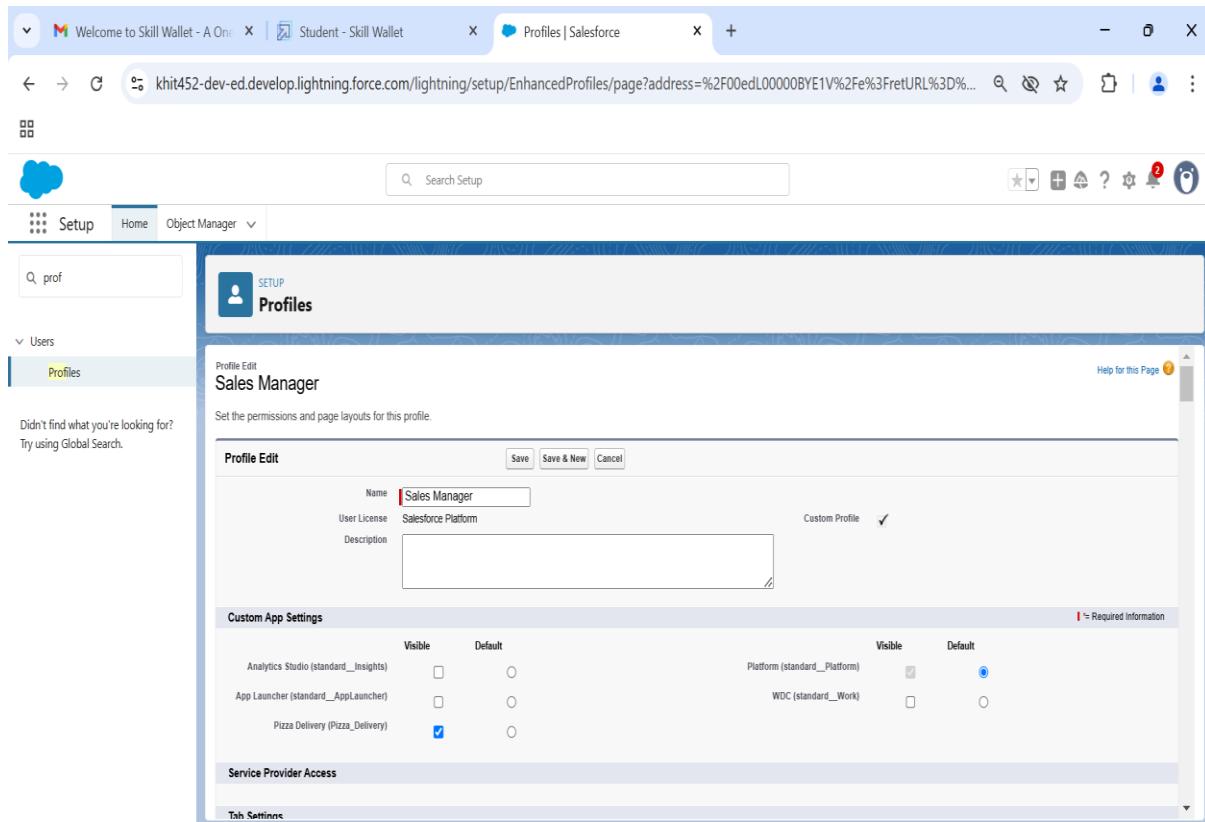
Available Values	Selected Values
Paneer Makhanai Margherita Onion Pizza Tomato pizza	Paneer Chicken Chicken pizza

Buttons between the lists are 'Add' and 'Remove'.

# Profile

## To create a new profile

1. Go to setup type profiles in quick find box click on profiles clone the desired profile (standard platform user is pref) and clone that profile
2. Enter a Profile Name Sales Manager and click on Save
3. Click on the new created profile
4. While still on the profile page, then click Edit
5. For the sales manager profile give the following access
6. Similarly clone the standard platform profile and give the name as Sales Executive and give the same access to the as of sales manager
7. Again clone the standard User profile and this time give all access for the objects, tabs and field and give the profile name as Delivery Person



The screenshot shows the Salesforce Setup interface with the following details:

- Profile Edit:** A modal window titled "Delivery Person" is open.
- Name:** Delivery Person
- User License:** Salesforce
- Description:** (Empty text area)
- Custom Profile:** Checked
- Custom App Settings:** A grid of app settings with checkboxes for "Visible" and "Default". Some apps have additional checkboxes for "Visible" and "Default".
- Buttons:** Save, Save & New, Cancel

## User Adoption

1. Go to the Gear Icon? Click the setup click on the home button and in the quick find box search for the user Click on the user than click on new fill the fields
2. Follow the above and create 2 more user? Marketing manager and Sales Rep

The screenshot shows the Salesforce Setup interface with the following details:

- Users:** A list of users including:
  - Chatter Expert (Alias: Chatter, Username: chatty\_00dd00000kny0jab, Profile: Chatter Free User)
  - M\_Tushar (Alias: tm, Username: 218x1at1232@khit.com, Profile: Sales Manager)
  - N\_Marketing\_manager (Alias: nn, Username: 218x1at1232@ac.in, Profile: Sales Manager)
  - Seku\_Manu\_Bhargav (Alias: MSeku, Username: batch@khit.com, Profile: System Administrator)
  - T\_Sales\_Rep (Alias: st, Username: 218x1at1232@ac.in, Profile: Sales Manager)
  - User\_Integration (Alias: integ, Username: integration@000dd00000kpy0jab.com, Profile: Analytics Cloud Integration User)
  - User\_Security (Alias: sec, Username: insightssecurity@000dd00000kpy0jab.com, Profile: Analytics Cloud Security User)
- Role:** A column indicating the user's role.
- Active:** A column indicating if the user is active.
- Profile:** A column showing the user's profile name.

# OWD (Organization Wide Default)

## Create OWD Setting

1. Setup, use the Quick Find box to find Sharing Settings.
2. Click Edit in the Organization-Wide Defaults area.
3. For each object, select the default access you want to give everyone.
4. For Every custom object give private as a record level security.

The screenshot shows the Salesforce Sharing Settings page. The URL in the browser is <https://khit452-dev-ed.lightning.force.com/lightning/setup/SecuritySharing/page?address=%2Fshare%2FOrgDefaultSharing%3Fid%3D00U0A0000000000>. The page title is "Sharing Settings".

The left sidebar shows the "Sharing Settings" section selected under "Sharing". Other sections like "Guest User Sharing Rule Access Report" and "Standard Report Visibility" are also visible.

The main content area displays a table of sharing settings for various objects:

Object	Default Access	
Web Cart Document	Private	
Work Order	Private	
Work Plan	Private	
Work Plan Template	Private	
Work Step Template	Private	
Work Type	Private	
Work Type Group	Private	
Address Books	Private	
Best Seller	Private	
Beverage	Private	
Customer Detail	Private	
EveryDay Value	Private	
New Launch	Private	
Pizza	Private	

At the bottom of the page, there are "Save" and "Cancel" buttons, and checkboxes for "Standard Report Visibility", "Manual User Record Sharing", "Manager Groups", "Secure guest user record access", and "Require permission to view record names in lookup fields".

# Reports

1. Go to the gear icon click on the setup click on the home button in the quick fin box search for app manager select the App and go to the navigation
2. There select the Report and include in the app than go to App There you will find the Tab
3. Click New Report create folder Create the Report Select the Customer with Mode of Payment Object Mode of Payment in the columns Customer detail Created Date columns mode of payment Id and in the rows Mode of payment Mode of payment

The screenshot shows the Salesforce Lightning interface for a Pizza Delivery account. The top navigation bar includes tabs for 'Welcome to Skill Wallet - A On...', 'Student - Skill Wallet', 'Home | Salesforce', and 'Recent | Reports | Salesforce'. The main content area is titled 'Reports' and shows a list of recent reports. The table has columns: Report Name, Description, Folder, Created By, Created On, and Subscribed. Two reports are listed:

Report Name	Description	Folder	Created By	Created On	Subscribed
New Report		Private Reports	Manju Bhargavi Sekuri	3/3/2025, 3:52 pm	
New Report Customer Mode of Payment		Private Reports	Manju Bhargavi Sekuri	3/3/2025, 3:13 pm	

The left sidebar contains navigation links for Reports, Folders, and Favorites.

4. For to the reports and select the customer details with mode of payment.
5. Select the following option for the Rows and column.
6. You can create the chart as so just click on the chart and you will be able to create it

Browser Tab: Welcome to Skill Wallet - A One

Browser Tab: Student - Skill Wallet

Browser Tab: Home | Salesforce

Browser Tab: New Report | Salesforce

URL: khit452-dev-ed.lightning.force.com/lightning/r/Report/00OdL00000AapojUAB/view?queryScope=userFolders

Report: Customer Details with Mode Of Payments

Customer Detail: Customer Name: Ram

Customer Detail: Created Date: 03/03/2025

Mode Of Payment: ID: a06dL000008GJvR

M-0001 (1) Subtotal: teju

M-0004 (1) Subtotal: brahma

M-0005 (1) Subtotal: tejaswini

M-0006 (1) Subtotal: manju

M-0007 (1) Subtotal: bhargavi

M-0008 (1) Subtotal: kesav

Total (7)

Row Counts: Detail Rows: Subtotals: Grand Total:

Browser Tab: Welcome to Skill Wallet - A One

Browser Tab: Student - Skill Wallet

Browser Tab: Home | Salesforce

Browser Tab: New Report | Salesforce

Browser Tab: pizza delivery application

URL: khit452-dev-ed.lightning.force.com/lightning/r/Report/00OdL00000AapojUAB/view?queryScope=userFolders

Report: Customer Details with Mode Of Payments

Total Records: 7

Record Count: 7

Mode Of Payment: Mode Of Payment

- M-0007
- M-0005
- M-0006
- M-0001
- M-0004
- M-0008
- M-0009

Mode Of Payment: Mode Of Payment: Customer Detail: Customer Name: Customer Detail: Created Date: Mode Of Payment: ID:

Row Counts: Detail Rows: Subtotals: Grand Total:

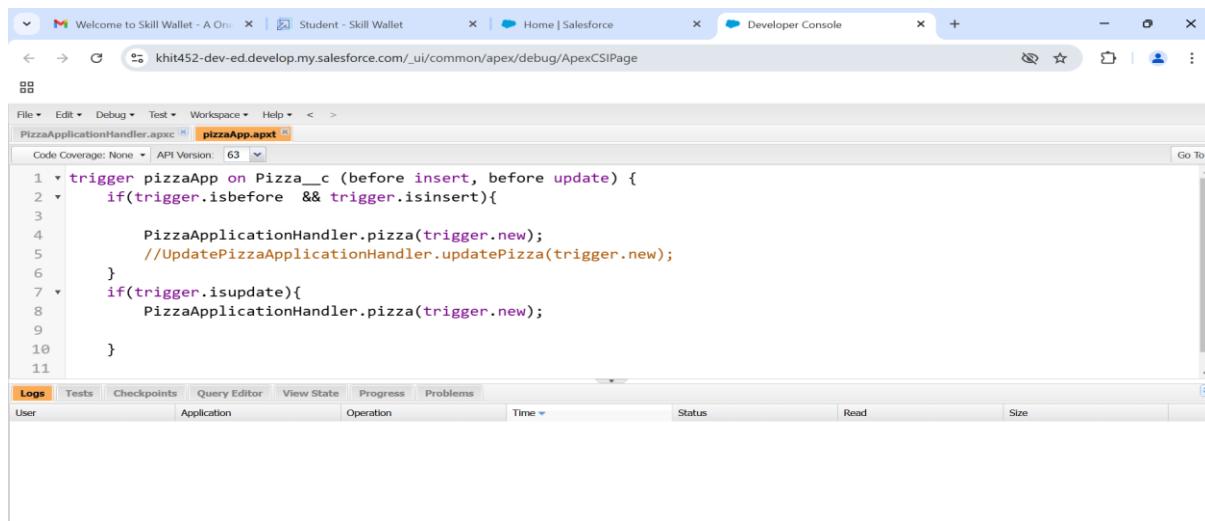
## 5. Testing and Validation

### Apex Trigger

#### Activity- 1

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

#### Trigger



The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for 'Welcome to Skill Wallet - A On...', 'Student - Skill Wallet', 'Home | Salesforce', and 'Developer Console'. The main area displays the code for the 'pizzaApp.apxt' trigger:

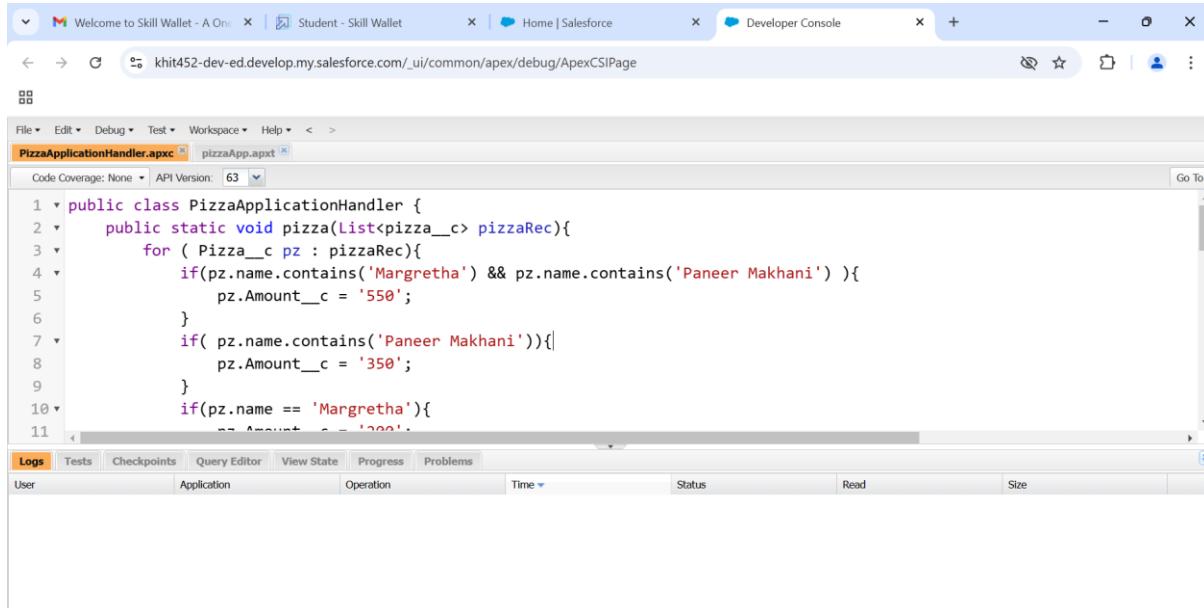
```
trigger pizzaApp on Pizza__c (before insert, before update) {
    if(trigger.isbefore && trigger.isinsert){
        PizzaApplicationHandler.pizza(trigger.new);
        //UpdatePizzaApplicationHandler.updatePizza(trigger.new);
    }
    if(trigger.isupdate){
        PizzaApplicationHandler.pizza(trigger.new);
    }
}
```

The code uses the `PizzaApplicationHandler` class to handle pizza operations. It checks if the trigger is before insert and if so, calls `pizza` and `updatePizza` methods. If the trigger is an update, it only calls `pizza`. The developer console also shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with the Logs tab currently selected.

#### Trigger Code:

```
trigger pizzaApp on Pizza__c (before insert, before update) {
    if(trigger.isbefore && trigger.isinsert){
        PizzaApplicationHandler.pizza(trigger.new);
        //UpdatePizzaApplicationHandler.updatePizza(trigger.new);
    }
    if(trigger.isupdate){
        PizzaApplicationHandler.pizza(trigger.new);
    }
}
```

## Trigger Handler



The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for 'Welcome to Skill Wallet - A One...', 'Student - Skill Wallet', 'Home | Salesforce', and 'Developer Console'. Below the navigation is a toolbar with icons for back, forward, search, and refresh. The main area is a code editor titled 'PizzaApplicationHandler.apc' with the file path 'pizzaApp.apxt'. The code is written in Apex and defines a class 'PizzaApplicationHandler' with a static method 'pizza'. The method iterates through a list of pizza records ('pizza\_\_c') and sets their amounts based on specific conditions related to pizza names like 'Margretha', 'Paneer Makhani', 'Tomato Pizza', and 'Onion Pizza'. The code editor has a status bar at the bottom showing 'Code Coverage: None', 'API Version: 63', and a 'Go To' button. Below the code editor is a log viewer with tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Logs' tab is selected, showing a table with columns for User, Application, Operation, Time, Status, Read, and Size.

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

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

## Trigger Handler code:

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

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

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

## Schedule Apex

## Schedule Apex for Frequently visited Customer

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for 'Welcome to Skill Wallet - A On...', 'Student - Skill Wallet', 'Home | Salesforce', and 'Developer Console'. The URL in the address bar is 'khit452-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage'. The main area displays the code for the 'PizzaDiscountScheduler' class:

```
1 public class PizzaDiscountScheduler implements Schedulable {
2     public void execute(SchedulableContext sc) {
3         // Logic for sending the email
4         // if (System.now() == System.DayOfWeek.Sunday) {
5             List<Customer_Detail__c> pz = new List<Customer_Detail__c>();
6             String s='gmail.com';
7             for(Customer_Detail__c c: pz)
8             {
9                 if(c.Email__c.contains(s))
10                 {
11                     system.debug('haiiiii');
```

The code is annotated with comments and logic for sending emails to customers whose email addresses contain 'gmail.com'. A debug statement is present at the end of the loop. The code editor shows tabs for 'PizzaApplicationHandler.apxc', 'pizzaApp.apxt', and 'PizzaDiscountScheduler.apxc', with 'PizzaDiscountScheduler.apxc' currently selected. The status bar at the bottom indicates 'Logs' is active.

```

public class PizzaDiscountScheduler implements Schedulable {
    public void execute(SchedulableContext sc) {
        // Logic for sending the email
        // if (System.now() == System.DayOfWeek.Sunday) {
            List<Customer_Detail__c> pz = new List<Customer_Detail__c>();
            String s='gmail.com';
            for(Customer_Detail__c c:pz)
            {
                if(c.Email__c.contains(s))
                {
                    system.debug('haiiiii');
                }
            }
            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
            email.setToAddresses(new List<String>{'user@example.com'});
            email.setSubject('Special Sunday Discount');
            email.setPlainTextBody('Enjoy a 20% discount on all pizzas today!');
            Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
        }
    }
}

```

## Activity-2

1. Click on the Gear Icon? Go to the Home Tab in the Quick Find Box Search for Apex Class.
2. Click on the Schedule Apex Give Job Name as Schedule Pizza.
3. Click on Apex Class Lookup Select Pizza Discount Scheduler In Recently Viewed Apex Class.
4. In Frequency Click on the Weekly Radio button You can Select the Start Date and End date As Per You Requirement and then Click on the Save Button.

The screenshot shows the Salesforce Setup Apex Classes page. The sidebar on the left includes sections for Email, Custom Code (with Apex Classes selected), Environments, Jobs, Apex Flex Queue, and Apex Jobs. The main content area displays the Apex Classes page with the following details:

- Apex Classes** header with a gear icon.
- A message: "Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform."
- A green box: "Percent of Apex Used: 0.05% You are currently using 3,006 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization."
- Buttons: Estimate your organization's code coverage, Compile all classes, View: All, Create New View.
- A table listing two Apex classes:
 

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit   Del   Security	PizzaApplicationHandler		63.0	Active	1,954	Manju Bhargavi Sekuri, 04/03/2025, 11:07 am	<input type="checkbox"/>
Edit   Del   Security	PizzaDiscountScheduler		63.0	Active	799	Manju Bhargavi Sekuri, 04/03/2025, 11:22 am	<input type="checkbox"/>
- Dynamic Apex Classes** section with the text: "Dynamic Apex extends your programming reach by interacting with Lightning Platform components." and a View: All button.

- Click on the debug besides file Click on the Open Execute Anonymous Windows
- Execute the below code

The screenshot shows the Salesforce Developer Console with the "Logs" tab selected. A modal window titled "Enter Apex Code" is open, displaying the following Apex code:

```

1 // Schedule the job to run every Monday at 8 AM
2 String cronExp = '0 0 8 ? * SUN';
3
4 // Create an instance of the ExpenseReportProcessor class
5 PizzaDiscountScheduler expenseProcessor = new PizzaDiscountScheduler();
6
7 // Schedule the job using the System.schedule method
8 System.schedule('Expense Report Processor', cronExp, expenseProcessor);
9

```

The developer console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, and View. At the bottom, there are buttons for Open Log, Execute, and Execute Highlighted.

```
// Schedule the job to run every Monday at 8 AM  
  
String cronExp = '0 0 8 ? * SUN';  
  
// Create an instance of the ExpenseReportProcessor class  
  
PizzaDiscountScheduler expenseProcessor = new PizzaDiscountScheduler();  
  
// Schedule the job using the System.schedule method  
  
System.schedule('Expense Report Processor', cronExp, expenseProcessor);
```

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

### **1. Order Management**

- Track customer orders from start to finish.

### **2. Customer Support**

- Handle customer complaints and issues (like wrong orders) quickly through chat, email, or phone.

### **3. Personalized Offers**

- Send special discounts or offers to customers based on their past orders.

### **4. Driver Management**

- Assign delivery drivers to orders and track their location in real-time.

### **5. Inventory Management**

- Keep track of ingredients and automatically reorder when stock is low.

## **7. Conclusion**

- A comprehensive Pizza Delivery Application has been successfully developed using Salesforce, offering a streamlined process for managing orders, deliveries, and inventory.
- Key features like real-time order tracking, automated inventory management, and delivery performance monitoring have been implemented.
- Salesforce's powerful platform has enabled the development of a scalable, automated, and efficient system that enhances the operational efficiency and customer satisfaction of the pizza delivery business.