

A CRM APPLICATION FOR LAPTOP RENTALS

1. Project Overview

The Laptop Rentals CRM Application is a comprehensive solution for managing and simplifying laptop rental operations. It enables daily tracking and reporting on available laptops, rental bookings, consumer information, and billing processes, which are then communicated to the management. This CRM leverages customer relationship management to enhance customer engagement, streamline operations, and improve efficiency in the laptop rental business. The project aims to deliver a user-friendly application that meets the specific operational needs of a laptop rental company.

2. Objectives

Business Goals: The Laptop Rentals CRM Application will automate rental operations and revenue reporting, offering insights into performance and customer preferences. It will help forecast demand, streamline resource allocation, and optimize inventory management.

Specific Outcomes: The CRM will track rental trends, improve operational efficiency, and support better decision-making through automated reporting and demand forecasting.

3. Salesforce Key Features and Concepts Utilized

- **Reporting and Dashboards:**
 - **Rental Activity Reports:** Generates insights on daily laptop rentals and returns.
 - **Revenue Reports:** Provides information on daily revenue from laptop rentals.
 - **Customer Analytics:** Tracks popular laptop models and top customers.
 - **Inventory Management:** Assists in planning and managing laptop inventory.

- **Rollup Summary Field:**
 - **Purpose:** Aggregates data from child to parent objects in a master-detail relationship and includes COUNT and SUM functions.
- **Cross-Object Formula Field:**
 - **Purpose:** References fields across objects.
 - **Function:** Calculates total rental fees by multiplying rental duration and price per day.
- **Validation Rules:**
 - **Purpose:** Ensures data accuracy through input validation.
 - **Is Blank Formula:** Checks if required fields are blank and displays an error if true.
- **Permission Sets:**
 - **Organization-Wide Defaults (OWD):** Sets baseline access levels for records.
 - **Roles and Access:**
 - **Owner:** Views records of all customers and rental agreements.
 - **Agent:** Accesses assigned rental agreements and customer information.

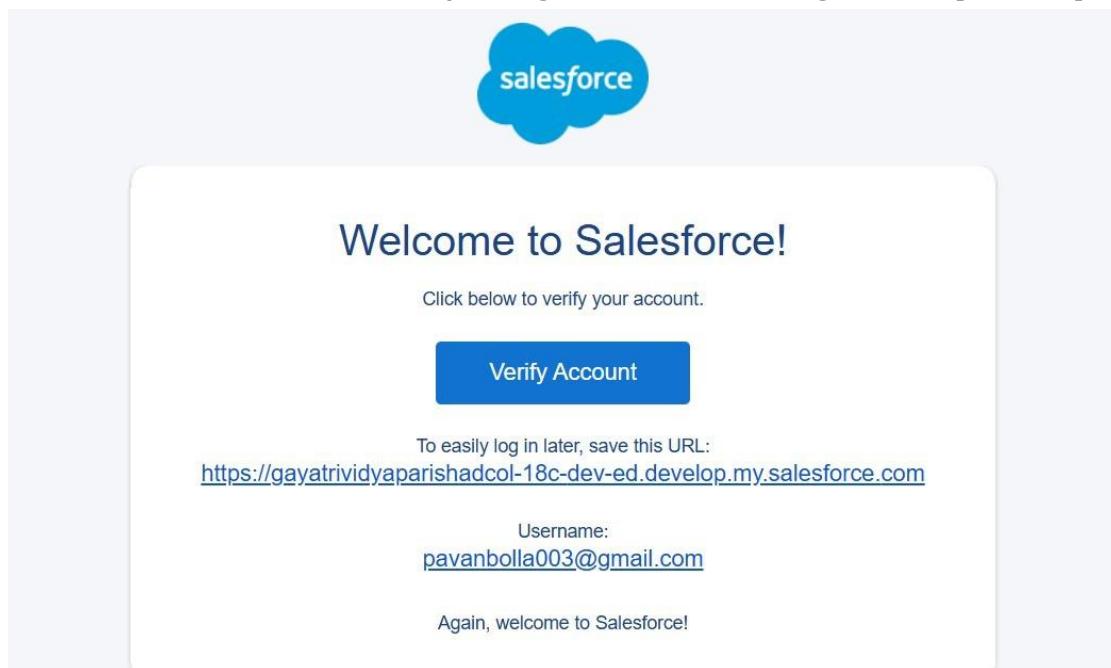
4. Detailed Steps to Solution Design

1. Salesforce Setup: Laying the Foundation for Your CRM

- **Purpose:** Establish a Salesforce environment as the foundation for your CRM.
- **Process:**
 - Create Your Developer Account: Start with the Salesforce Developer Sign-Up.
 - Go to <https://developer.salesforce.com/signup>.
 - On the sign-up form, enter the following details:
 1. **First name & Last name**
 2. **Email**
 3. **Role:** Developer
 4. **Company:** College Name
 5. **Country:** India
 6. **Postal Code:** pin code
 7. **Username:** Should be a combination of your name and company.

This does not need to be an actual email id; you can use anything in the format: username@organization.com.

- Click on **Sign me up** after filling in these details.
- Activate: Confirm your registration email and log in to complete setup.



2. Object Creation: Building the Data Structure

► **Purpose:** Salesforce objects are database tables that permit you to store data specific to your organization. There are two types of Salesforce objects:

- **Standard Objects:** Provided by Salesforce (e.g., Users, Contracts).
- **Custom Objects:** Created by users to store information unique to their organization.

► **Process:**

- Click on the gear icon (⚙️) and then click **Setup**.

Creating a Custom Object

- **Access Object Manager:** From the Setup page, click on **Object Manager**.

1. **Create Total Laptops Object:**

- a. Click on **Create** and select **Custom Object**.
- b. **Enter the following details:**
 - i. Label Name: Total Laptops
 - ii. Plural Label Name: Total Laptops

iii. Record Name Label: Total Laptops

iv. Data Type: Text

c. Click on **Allow Reports, Allow Search, and Track Field History.**

d. Click **Save.**

2. Create Consumer Object:

a. Click on **Create** and select **Custom Object**.

b. Enter the following details:

i. Label Name: Consumer

ii. Plural Label Name: Consumer

iii. Record Name Label: consumer_name

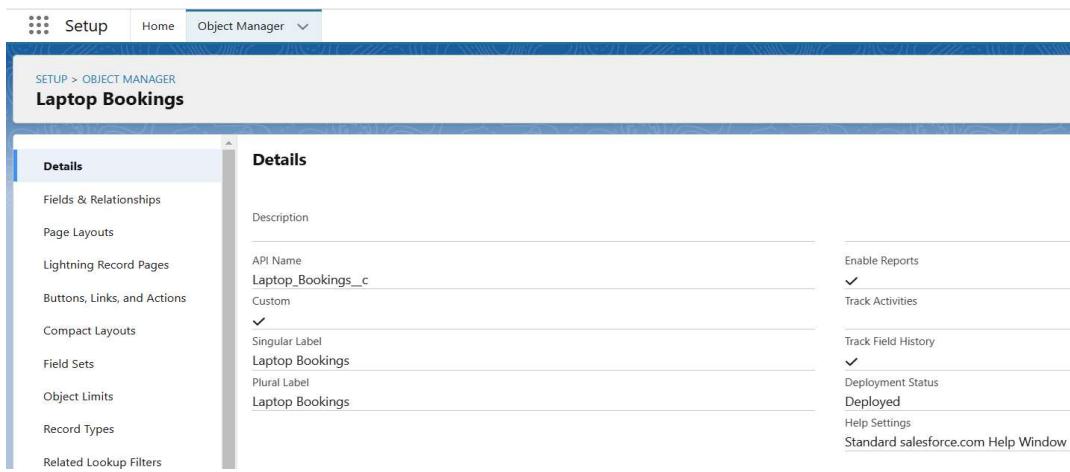
iv. Data Type: Name

c. Click on **Allow Reports, Allow Search, and Track Field History.**

d. Click **Save.**

3. Create Laptop Bookings Object:

- a. Click on **Create** and select **Custom Object**.
- b. Enter the following details:
 - i. Label Name: Laptop Bookings
 - ii. Plural Label Name: Laptop Bookings
 - iii. Record Name Label: Laptop Bookings
 - iv. Data Type: Name
- c. Click on **Allow Reports, Allow Search, and Track Field History**.
- d. Click **Save**.



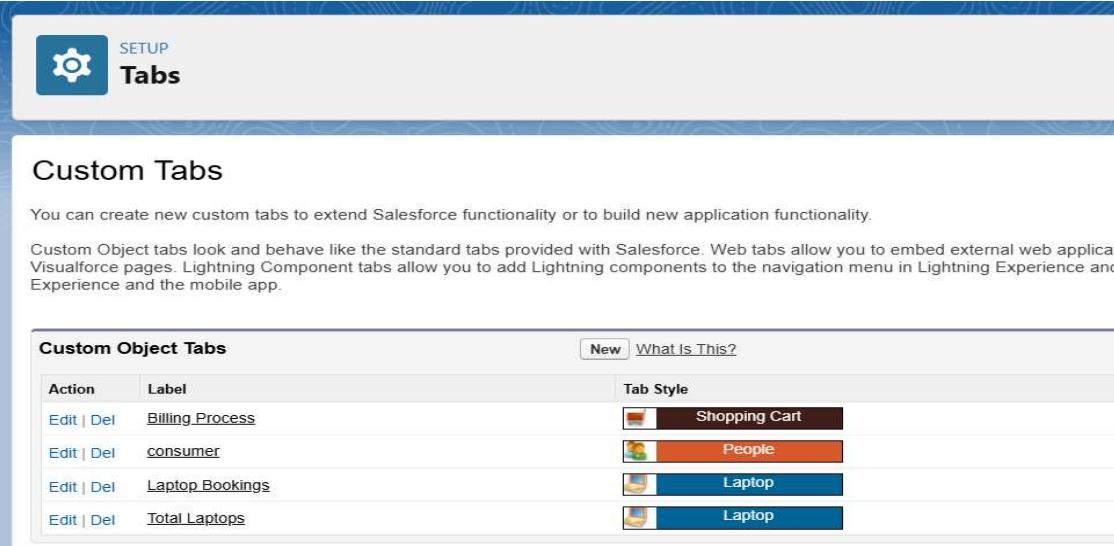
4. Create Billing Process Object:

- a. Click on **Create** and select **Custom Object**.
- b. Enter the following details:
 - i. Label Name: Billing Process
 - ii. Plural Label Name: Billing Process
 - iii. Record Name Label: Billing Process Name
 - iv. Data Type: Name
- c. Click on **Allow Reports, Allow Search, and Track Field History**.
- d. Click **Save**.



3. Tabs: Creating Easy Navigation

- **Purpose:** Organize custom objects with tabs to streamline navigation.
- **Process:**
 - Add Custom Tabs: Setup > Tabs > Custom Object Tabs > New.
 - Select Object(Total Laptops) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
 - Make sure that the Append tab to user's existing personal customizations is checked.
 - Click save.
 - Repeat: create the Tabs for the remaining Objects, they are “consumer, Laptop Booking, Billing process”. Follow the same steps as mentioned above for Total Laptops.



The screenshot shows the 'Custom Object Tabs' section of the Salesforce Setup. At the top, there are 'New' and 'What Is This?' buttons. Below is a table with columns for Action (Edit | Del), Label, and Tab Style. The table contains four rows corresponding to the objects listed in the process:

Action	Label	Tab Style
Edit Del	Billing_Process	Shopping Cart
Edit Del	consumer	People
Edit Del	Laptop Bookings	Laptop
Edit Del	Total Laptops	Laptop

4. The Lightning App: Centralizing the CRM Workspace

- **Purpose:** Bundle CRM components in one Lightning App for efficiency.
- **Process:**
 - Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.
 - Fill the app name in app details as LAPTOP RENTALS >>Next >> (App

option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.

- Upload a photo that is related to your app.
- To Add Navigation Items: Select the items (Total Laptops, consumers, Laptop Booking, Billing Process) from the search bar and move it using the arrow button >> Next.
- To Add User Profiles: Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

5. Fields & Relationships: Defining Data Points and Connections

- **Purpose:** Add essential data fields and relationships between objects.
- **Process:**
 - Consumer Object: In Object Manager > Consumer > Fields & Relationships, add fields like Name, Email, and Phone Number.

FIELD LABEL	FIELD NAME	DATA TYPE
Address	Address__c	Text Area(255)
consumer_Status	consumer_Status__c	Picklist
consumer_name	Name	Text(80)
Created By	CreatedById	Lookup(User)
Email	Email__c	Email
Last Modified By	LastModifiedById	Lookup(User)
Owner	OwnerId	Lookup(User,Group)
Phone number	Phone_number__c	Phone

- Laptop Bookings: Define fields like Amount, Core Type, email, how many months, laptop Bookings, Laptop Names, Laptops Available.

SETUP > OBJECT MANAGER
Laptop Bookings

Fields & Relationships			
11 Items, Sorted by Field Label			
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
Amount	Amount_c	Currency(18, 0)	
Core Type	Core_Type_c	Picklist	Laptop Names
Created By	CreatedById	Lookup(User)	
email	email_c	Email	
how many months	how_many_months_c	Picklist	
Laptop Bookings	Name	Text(80)	
Laptop Names	Laptop_Names_c	Picklist	
Laptops Available	Laptops_Available_c	Formula (Number)	
Last Modified By	LastModifiedById	Lookup(User)	

- Establish Relationships: Link "Laptop Bookings" to "Total Laptops" and "Consumer".
- Billing Process: Add fields for billing such as Billing ProcessName, Name, Payment Mode and relate to "Laptop Bookings".

SETUP > OBJECT MANAGER
Billing Process

Fields & Relationships			
7 Items, Sorted by Field Label			
FIELD LABEL	FIELD NAME	DATA TYPE	
Amount	Amount_c	Formula (Number)	
Billing ProcessName	Name	Text(80)	
Created By	CreatedById	Lookup(User)	
Laptop Booking	Laptop_Booking_c	Lookup(Laptop Bookings)	
Last Modified By	LastModifiedById	Lookup(User)	
Name	Name_c	Master-Detail(consumer)	
Payment Mode	Payment_Mode_c	Picklist	

6 . Validation Rules: Enforcing Data Quality Standards

- **Purpose:** Ensure data consistency with validation rules.
- **Process:**
 - Go to the setup page >> click on object manager >> From drop down click edit for consumer object.
 - Click on the validation rule >> click New.
 - Enter the Rule name as “Phonenumberoremailblankrule”.
 - Enter the description as “phone number and email number should not be blank”.
 - Enter the formula as “OR(ISBLANK(phone_number__c), ISBLANK(email__c))” and check the syntax.
 - Save the validation rule.

consumer Validation Rule

Validation Rule Detail	Edit	Clone
Rule Name: Phonenumberoremailblankrule	Active	✓
Error Condition Formula: OR(ISBLANK(Phone_number__c), ISBLANK(Email__c))		
Error Message: Please fill the phone number and email id	Error Location	Top of Page
Description: phone number and email number should not be blank		
Created By: Manasa Kuraku, 26/10/2024, 5:59 pm	Modified By	Manasa Kuraku, 26/10/2024, 5:59 pm
Edit Clone		

7. Profiles: Setting Access Permissions

- **Purpose:** Manage what users can see and do in the CRM.
- **Process:**
 - Owner Profile:**
 - Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (owner) >> Save.
 - Scroll down to Custom Object Permissions and Give access permissions for Total Laptops, consumers , Laptop Booking and Billing Process

objects as mentioned in the below diagram.

- Give Access and Save it.

The screenshot shows the 'Profiles' setup page. It includes sections for 'Individuals', 'Invoices', 'Leads', 'Work Types', and 'Custom Object Permissions'. Under 'Custom Object Permissions', there are two tables: one for 'Billing Process' and one for 'consumers'. A red arrow points to the 'Basic Access' row of the 'Billing Process' table, specifically highlighting the 'Read' column. Other columns include Create, Edit, Delete, View All, and Modify All. The 'consumers' table has similar columns. Below these tables are 'Session Settings' and 'Password Policies' sections.

Agent Profile:

- Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard Platform User) >> enter profile name (Agent) >>Save.
- While still on the profile page, then click Edit.
- Scroll down to Custom Object Permissions and Give access permissions for Total Laptops, consumer , Laptop Bookings and Billing Process objects as mentioned in the below diagram.
- Give access and save it.

This screenshot shows the 'Profiles' setup page with different sections compared to the previous one. It includes 'Contact Point Consents', 'Contact Point Emails', 'Selling Channels', and 'User External Credentials'. Below these are 'Custom Object Permissions' tables for 'Billing Process' and 'consumers', which mirror the structure and data from the first screenshot. At the bottom are 'Session Settings' and 'Password Policies' sections, with specific configuration details like password expiration and complexity requirements.

8 . Roles & Hierarchy: Managing Data Access

► **Purpose:** Define role-based record sharing and access levels.

► **Process:**

Owner Role

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

The screenshot shows a software interface for managing roles. At the top, there's a header with a user icon and the word 'SETUP'. Below it, a sub-header says 'Roles'. The main area is titled 'Role Edit' and has a sub-subtitle 'New Role'. There are four input fields: 'Label' (containing 'owner'), 'Role Name' (containing 'owner'), 'This role reports to' (set to 'CEO'), and 'Role Name as displayed on reports' (empty). At the bottom right are three buttons: 'Save', 'Save & New', and 'Cancel'.

Agent Role

- Go to quick find - Search for Roles - click on set up roles.
- Click plus on CEO role, and click add role under owner.
- Give Label as “Agent” and Role name gets auto populated. Then click on Save.

9. Users: Adding Team Members to the CRM

► **Purpose:** Add individual users to the system with assigned roles and permissions.

► **Process:**

- Go to setup - type users in quick find box - select users -click New user.
- Fill in the fields
 - First Name : vicky
 - Last Name : y

- Alias : Give a Alias Name
- Email id : Give your Personal Email id
- Username : Username should be in this form: text@text.text
- Nick Name : Give a Nickname
- Role : owner
- User license : Salesforce
- Profiles : owner.

- Save it.

The screenshot shows the 'User Edit' page for a user named 'vicky'. The page has a header with 'SETUP' and 'Users'. Below the header, the user's name 'vicky' is displayed. There are two tabs: 'General Information' (selected) and 'Advanced Settings'. The 'General Information' tab contains fields for First Name ('vicky'), Last Name ('y'), Alias ('vick'), Email ('kurakumanasa2003@gmail.com'), Username ('kurakumanasa@gmail.com'), Nickname ('vicky'), Title (empty), Company (empty), Department (empty), Division (empty), Role ('owner'), User License ('Salesforce'), Profile ('Standard User'), and Active status (checked). Other optional checkboxes include Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, and Data.com User Type ('--None--'). A note at the bottom right indicates that some fields are required.

Creating another users

- Go to setup -type users in quick find box - select users -click New user.
- Fill in the fields
 - First Name : ram
 - Last Name : ram
 - Alias : Give a Alias Name
 - Email id : Give your Personal Email id
 - Username : Username should be in this form: text@text.text
 - Nick Name : Give a Nickname
 - Role : Agent
 - User license : Salesforce platform

- Profiles : standard platform user.
- Save it.

User Edit
ram ramesh

User Edit

General Information

First Name	ram
Last Name	ramesh
Alias	rrame
Email	kurakumanasa2003@gmail.com
Username	manasakuraku@gmail.com
Nickname	ram
Title	
Company	
Department	
Division	

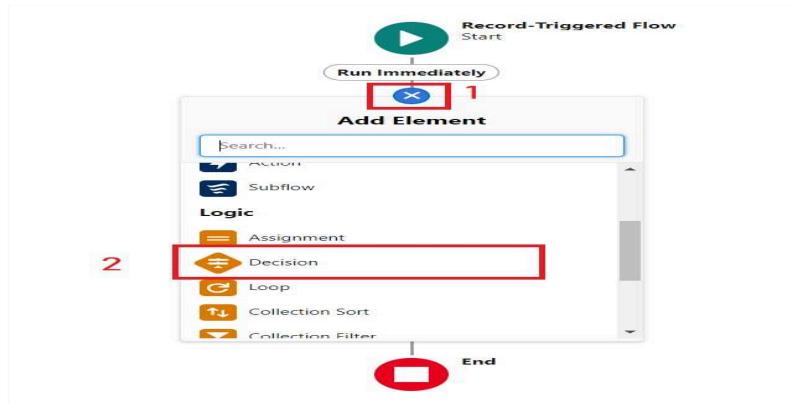
Role: Agent
User License: Salesforce Platform
Profile: Standard Platform User
Active:
Marketing User:
Offline User:
Knowledge User:
Flow User:
Service Cloud User:
Site.com Contributor User:
Site.com Publisher User:
WDC User:
Data.com User Type: None

10. Flows: Automating Workflow Processes

- Purpose: Automate CRM tasks with tailored flows for each laptop type.
- Process:

Create a Flow on dell laptop

- Go to setup >>type Flow in quick find box >> Click on the Flow and Select the New Flow.
- Select the Record-triggered flow and Click on Create.
- Select the Object as a Laptop Booking in the Drop down list.
- Select the Trigger Flow when: “A record is Created or Updated”.



- Select the Optimize the flow for: “Actions and Related Records” and Click on Done.
- Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Decision Element”.
- Enter the Details Label: Field should be Update, API name: Gets Automatically Generated.
- Enter the Outcome Details Label: dell , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.Laptop booking__c.
 - Operator: Select Equals.
 - Value: Select dell
 - Add the same outcome order to acer , hp,mac.
 - Click done.

Edit Decision

*Label	*API Name
field should updated	field_should_updated
Description	
the field should be automatically updated	
Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.	
OUTCOME ORDER ① +	OUTCOME DETAILS
dell	*Label dell *Outcome API Name dell
acer	
hp	Condition Requirements to Execute Outcome All Conditions Are Met (AND)
mac	
false	Resource: \$Record > Laptop names X Operator: Equals Value: Dell

Done

- Go to flow page.
- Beside dell there is a symbol ‘+’ click on that.
- Again select decision
- Enter the Details Label: Field should Update(any one u want), API name: Gets Automatically Generated.

- select the Outcome Details Label: dell core i3 , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.core type.
 - Operator: Select Equals.
 - Value: Select core i3.
 - Then again click the symbol ‘+’ outcome details

- select the Outcome ‘+’ Details Label: dell core i5 , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.core type.
 - Operator: Select Equals.
 - Value: Select core i5.
 - Then again click the symbol ‘+’ outcome details

- Enter the Outcome Details Label: dell core i7 , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.core type.
 - Operator: Select Equals.
 - Value: Select core i7.

- Click done.

Edit Decision

*Label	*API Name														
field updated	field_updated														
Description															
<pre>if \$Record > core type == "dell core i3" then field updated = true endif</pre>															
Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.															
OUTCOME ORDER 1 + <ul style="list-style-type: none"> dell core i3 dell core i5 dell core i7 	OUTCOME DETAILS <table border="1"> <tr> <td>*Label</td> <td>*Outcome API Name:</td> </tr> <tr> <td>dell core i3</td> <td>dellcore_i3</td> </tr> <tr> <td colspan="2">Condition Requirements to Execute Outcome</td> </tr> <tr> <td colspan="2">All Conditions Are Met (AND)</td> </tr> <tr> <td>Resource</td> <td>Operator</td> <td>Value</td> </tr> <tr> <td>\$Record > core type</td> <td>Equals</td> <td>core i3</td> </tr> </table>	*Label	*Outcome API Name:	dell core i3	dellcore_i3	Condition Requirements to Execute Outcome		All Conditions Are Met (AND)		Resource	Operator	Value	\$Record > core type	Equals	core i3
*Label	*Outcome API Name:														
dell core i3	dellcore_i3														
Condition Requirements to Execute Outcome															
All Conditions Are Met (AND)															
Resource	Operator	Value													
\$Record > core type	Equals	core i3													
Delete Outcome		Cancel Done													

- Go to the flow page select ‘+’ after core i3 then again select the decision.
- Enter the Details Label: months selected , API name: Gets Automatically Generated.
- Enter the Outcome Details Label: dell 1(i3) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: 1.
- Enter the Outcome Details Label: dell 2(i3) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: Select 2..
- Click ‘+’ outcome details
- Enter the Outcome Details Label: dell 3(i3) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: Select 3..
- Click ‘+’ outcome details
- Enter the Outcome Details Label: dell 4(i3) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: Select 4.
- Click ‘+’ outcome details
- Enter the Outcome Details Label: dell 5(i3) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: Select 5.
- After dell 1(i3) there is ‘+’ symbol like dell 2(i3),dell 3(i3),dell 4(i3),dell

5(i3).

- Click on ‘+’ then select update records
- Enter the Details Label: one month of dell i3 rate , API name: Gets Automatically Generated.
- Field:- Amount__c , value:- for dell 1(i3)-1000, dell 2(i3)-2000, dell 3(i3)-3000, dell 4(i3)-4000, dell 5(i3)-5000. Follow for all these finally
- Click done.
- Enter the Details Label: months selected , API name: Gets Automatically Generated.
- Enter the Outcome Details Label: dell 1(i7) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: 1.
- Enter the Outcome Details Label: dell 2(i7) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: Select 2.
- Click ‘+’ outcome details
- Enter the Outcome Details Label: dell 3(i7) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: Select 3.
- Click ‘+’ outcome details
- Enter the Outcome Details Label: dell 4(i7) , Outcome API name: Gets Automatically Generated.
 - Resource: Select Record.how many months.
 - Operator: Select Equals.
 - Value: Select 4.
- Click ‘+’ outcome details

- Enter the Outcome Details Label: dell 5(i7) , Outcome API name: Gets Automatically Generated.

- Resource: Select Record.how many months.
- Operator: Select Equals.
- Value: Select 5.

Edit Decision

* Label months selected	* API Name months_selected																		
Description																			
<p>Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.</p> <table border="1"> <thead> <tr> <th>OUTCOME ORDER</th> <th>OUTCOME DETAILS</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>* Label 1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)</td> <td>Delete Outcome</td> </tr> <tr> <td>2</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td></td> <td></td> </tr> <tr> <td>4</td> <td></td> <td></td> </tr> <tr> <td>5</td> <td></td> <td></td> </tr> </tbody> </table>		OUTCOME ORDER	OUTCOME DETAILS	Actions	1	* Label 1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)	Delete Outcome	2			3			4			5		
OUTCOME ORDER	OUTCOME DETAILS	Actions																	
1	* Label 1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)	Delete Outcome																	
2																			
3																			
4																			
5																			
<p>Resource Operator Value</p> <p>: \$Record > how many months X Equals 1</p>																			
<input type="button" value="Cancel"/> <input type="button" value="Done"/>																			

- Follow the above picture you will understand.
- After dell 1(i7) there is ‘+’ symbol like dell 2(i7),dell 3(i7),dell 4(i7),dell 5(i7).
- Click on ‘+’ then select update records
- Enter the Details Label: one month of dell i5 rate , API name: Gets Automatically Generated.
- Field:- Amount__c , value:- for dell 1(i7)-2000, dell 2(i7)-4000, dell 3(i7)-6000, dell 4(i7)-8000, dell 5(i7)-10000. Follow for all these finally
- Click done.

Create a Flow on Acer laptop

- Go to the flow page.
- Beside **Acer**, locate the ‘+’ symbol and click on it.
- Select **Decision**:
 - Details Label: Field is Update
 - API Name: Gets Automatically Generated
 - Outcome Details Label: Acer core i3
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.core type
 - Operator: Equals
 - Value: Select core i3
- Go to the flow page again.
- Beside **Dell**, locate the ‘+’ symbol and click on it.
- Select **Decision**:
 - Details Label: months selected
 - API Name: Gets Automatically Generated
- Enter Outcome Details:
 - Outcome Details Label: Acer 1(i3)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 1
 - Outcome Details Label: Acer 2(i3)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 2
 - Click ‘+’ to add another outcome.
 - Outcome Details Label: Acer 3(i3)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 3
 - Click ‘+’ to add another outcome.
 - Outcome Details Label: Acer 4(i3)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months

- Operator: Equals
- Value: 4
- Click '+' to add another outcome.
- Outcome Details Label: Acer 5(i3)
- Outcome API Name: Gets Automatically Generated
- Resource: Select Record.how many months
- Operator: Equals
- Value: 5
- Update records for each outcome (Acer 1(i3) through Acer 5(i3)):
 - Click '+' beside each outcome, then select **Update Records**.
 - Details Label: one month of Acer i3 rate
 - API Name: Gets Automatically Generated
 - Field: Amount__c
 - Values:
 - For Acer 1(i3): 900
 - For Acer 2(i3): 1800
 - For Acer 3(i3): 2700
 - For Acer 4(i3): 3600
 - For Acer 5(i3): 4500

The screenshot shows the 'Update Records' configuration screen. At the top, there are fields for 'Label' (set to 'five month of acer i3 rate') and 'API Name' (set to 'five_month_of_acer_i3_rate'). Below these are sections for 'Description' (empty), 'How to Find Records to Update and Set Their Values' (radio button selected for 'Use the laptop bookings record that triggered the flow'), 'Set Filter Conditions' (dropdown set to 'None—Always Update Record'), and 'Set Field Values for the Laptop Bookings Record' (table with a single row showing 'Amount__c' field with value '4500' and a delete icon). The entire form is enclosed in a light gray border.

Create a Flow on Hp Laptop

- Go to the Flow page.
- Beside HP, locate the ‘+’ symbol and click on it.
- Add Decision:
 - Details Label: Field is Update
 - API Name: Gets Automatically Generated
- Enter Outcome Details:
 - Outcome Details Label: HP core i5
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.core type
 - Operator: Equals
 - Value: HP i5
 - Go to the Flow page again.
 - Beside HP, locate the ‘+’ symbol and click on it.
 - Add Decision:
 - Details Label: HP field should be updated
 - API Name: Gets Automatically Generated
 - Enter Outcome Details:
 - Outcome Details Label: HP 1(i5)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 1
 - Outcome Details Label: HP 2(i5)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 2
 - Click ‘+’ to add another outcome.
 - Outcome Details Label: HP 3(i5)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 3
 - Click ‘+’ to add another outcome.
 - Outcome Details Label: HP 4(i5)
 - Outcome API Name: Gets Automatically Generated

- Resource: Select Record.how many months
- Operator: Equals
- Value: 4
- Click '+' to add another outcome.
- Outcome Details Label: HP 5(i5)
- Outcome API Name: Gets Automatically Generated
- Resource: Select Record.how many months
- Operator: Equals
- Value: 5
- Update Records for each outcome (HP 1(i5) through HP 5(i5)):
 - Click '+' beside each outcome, then select Update Records.
 - Details Label: one month of HP i5 rate
 - API Name: Gets Automatically Generated
 - Field: Amount__c
 - Values:
 - For HP 1(i5): 1700
 - For HP 2(i5): 3400
 - For HP 3(i5): 5100
 - For HP 4(i5): 6800
 - For HP 5(i5): 8500

The screenshot shows the 'Update Records' configuration screen. At the top, there are fields for 'Label' (containing 'five month of hp i5 rate') and 'API Name' (containing 'five_month_of_hp_i5_rate'). Below these are sections for 'Description' (empty), 'How to Find Records to Update and Set Their Values' (radio button selected for 'Use the laptop bookings record that triggered the flow'), 'Set Filter Conditions' (dropdown set to 'None—Always Update Record'), and 'Set Field Values for the Laptop Bookings Record' (table with a single row showing 'Amount__c' field with value '8500').

Field	Value
Amount__c	8500

Create a Flow on Mac Laptop

- Go to the Flow page.
- Beside **Mac**, locate the ‘+’ symbol and click on it.
- Select **Decision**:
 - Details Label: mac should be Updated
 - API Name: Gets Automatically Generated
 - Outcome Details Label: mac laptop
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.core type
 - Operator: Equals
 - Value: Bionic Chip
- Go to the Flow page again.
- Beside **Mac**, locate the ‘+’ symbol and click on it.
- Select **Decision**:
 - Details Label: Mac months selected
 - API Name: Gets Automatically Generated
- Enter Outcome Details:
 - Outcome Details Label: mac bionic chip(1)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 1
 - Outcome Details Label: mac bionic chip(2)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 2
 - Click ‘+’ to add another outcome.
 - Outcome Details Label: mac bionic chip(3)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 3
 - Click ‘+’ to add another outcome.
 - Outcome Details Label: mac bionic chip(4)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months

- Operator: Equals
 - Value: 4
 - Click ‘+’ to add another outcome.
 - Outcome Details Label: mac bionic chip(5)
 - Outcome API Name: Gets Automatically Generated
 - Resource: Select Record.how many months
 - Operator: Equals
 - Value: 5
 - **Update Records** for each outcome (mac bionic chip(1) through mac bionic chip(5)):
 - Click ‘+’ beside each outcome, then select **Update Records**.
 - Details Label: one month of mac rate
 - API Name: Gets Automatically Generated
 - Field: Amount__c
 - Values:
 - For one month of mac bionic chip rate: 1700
 - For two months of mac bionic chip rate: 3400
 - For three months of mac bionic chip rate: 5100
 - For four months of mac bionic chip rate: 6800
 - For five months of mac bionic chip rate: 8500

Update Records

* Label	* API Name				
five month of mac rate	five_month_of_mac_rate				
Description					
<p>* How to Find Records to Update and Set Their Values</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Use the laptop bookings record that triggered the flow <input type="radio"/> Update records related to the laptop bookings record that triggered the flow <input type="radio"/> Use the IDs and all field values from a record or record collection <input type="radio"/> Specify conditions to identify records, and set fields individually 					
<h3>Set Filter Conditions</h3> <p>Condition Requirements to Update Record</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;"> <input style="border: none; font-size: small; width: 100%;" type="button" value="None—Always Update Record"/> ▼ </div>					
<h3>Set Field Values for the Laptop Bookings Record</h3> <table border="1"> <thead> <tr> <th>Field</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Amount_c</td> <td>8500</td> </tr> </tbody> </table>		Field	Value	Amount_c	8500
Field	Value				
Amount_c	8500				

FLOW:

- Click on **Save**.
 - Label: Laptop distributions
 - API Name: Automatically Filled
- Save the flow and activate

11. APEX: Advanced Customization

► **Purpose:** Use APEX to add custom automation and business logic.

APEX LaptopBookingHandler class code :

```
public class LaptopBookingHandler {  
  
    public static void sendEmailNotification (List<Laptop_Bookings__c> lapList){  
  
        for(Laptop_Bookings__c lap:lapList)  
  
        {  
  
            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();  
  
            email.setToAddresses( new List<String>{lap.Email__c});  
  
            email.setSubject('Welcome to our company');  
  
            string body = 'Dear ' +lap.Name +', \n';  
  
            body += 'Welcome to Laptop Rentals! You have been seen as a valuable  
customer to us.\n Please continue your journey with us, while we try to provide you with  
good quality resources. \n Laptop Amount = ' + lap.Amount__c + '\n core type =  
' +lap.Core_Type__c +'\n Laptop type =' +lap.Laptop_Names__c;  
  
            email.setPlainTextBody(body);  
  
            Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});  
        }  
    }  
}
```

- Class name:- LaptopBookingHandler
- API Name:- Laptop_Bookings__c(as per your org go to laptop booking object and copy from that).
- core__c (as per your org go to laptop booking object and copy from that).
- Laptop_type__c.(as per your org go to laptop booking object and copy from that).

Trigger for Email Notification: Sends email to customers upon booking

Trigger Code:

```
trigger LaptopBooking on Laptop_Bookings__c (After insert,after update) {
```

```
    if(trigger.isAfter && ( trigger.isInsert || trigger.isupdate))
    {
        LaptopBookingHandler.sendEmailNotification(trigger.new);
    }
}
```

- LaptopBooking - trigger name
- Laptop_Bookings__c -as per your org(go to laptop bookings object and copy from that object api name).

Welcome to our company [Inbox](#)

vicky project via nhvsluct7j@seinh.rw9buyc.5-c9o7weakap27.onc.salesforce.com
to me *

Dear smartinternz,
Welcome to Laptop Rentals! 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.
Laptop Amount = 5100.0
core type = core i5
Laptop type = Hp

[Reply](#)

[Forward](#)

12. Reports: Data-Driven Insights at a Glance

- **Purpose:** Use reports to gain insights into data trends and performance.
- **Process:**

Creating the Report

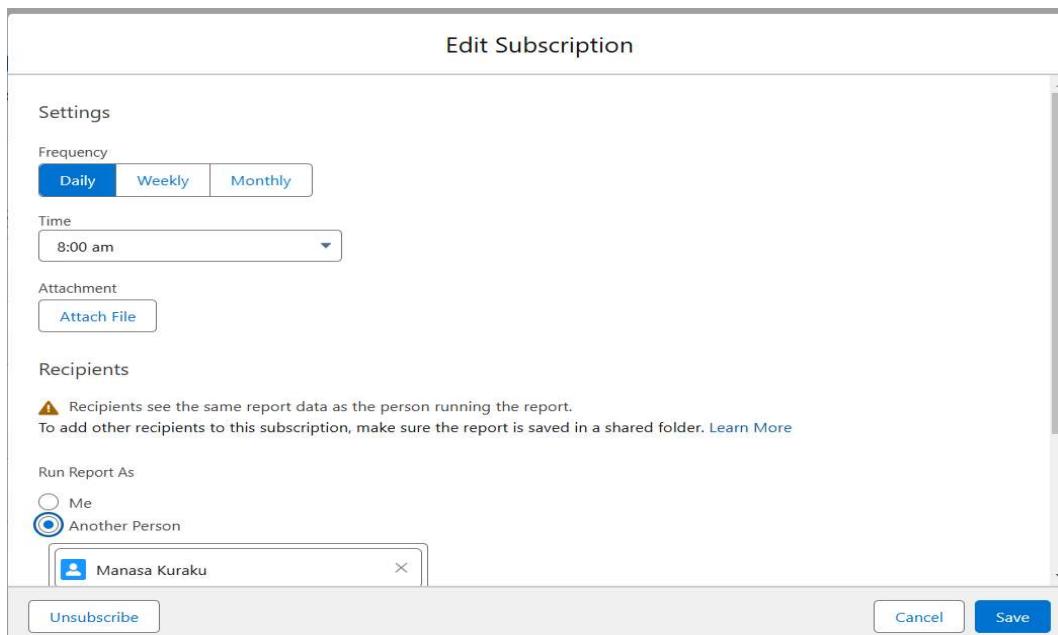
- Go to the App:
 - Click on the Reports tab.
- Click on New Report:Select Report Type:
 - Choose the report type from category or from the report type panel.
 - Search for Consumer with Laptop Bookings and Total Laptops.
 - Click on Start Report.
- Customize Your Report:
 - Add fields from the left pane as needed.
 - Organize the fields into rows and columns according to your requirements.
- Group Rows and Columns:
 - Follow the provided image or layout to group rows and columns.
- Select Bucket List:
 - Click the column drop-down and select Bucket List and click Apply.
- Save or Run the Report:
 - Follow the instructions or images to either save or run the report.

The screenshot shows a Zoho Analytics report interface. The title bar includes 'LAPTOP RENTALS', 'Total Laptops', 'consumer', 'Laptop Bookings', 'Billing Process', 'Analytics Report', and 'Data Analytics of Laptops'. The main area displays a table with the following data:

Total Records	Total Amount						
12	₹82,930						
Dell (2)	Bindhu (flash), Teja (Agency)	50, 50	core i7, core i3	₹500, ₹6,900	Basic, High		
Subtotal				₹7,400			
Acer (4)	Supriya (timesnew), Sravani (College), Sai (geeks), Chakri (School)	15, 50, 50, 50	core i3, core i5, core i5, Bionic chip	₹900, ₹1,200, ₹1,200, ₹9,000	Basic, Intermediate, Intermediate, High		
Subtotal				₹12,300			
Hp (4)	Vishnu (Personal), Manasa (google), Manasa (Company), Chakri (google)	50, 50, 50, 50	core i3, core i5, core i5, core i7	₹150, ₹9,000, ₹12,080, ₹15,000	Basic, High, Very High, Very High		
Subtotal				₹36,230			
Mac (2)	Sita (smartinternz), Ram (smartinternz)	50, 50	Bionic chip	₹12,000, ₹15,000	Very High, Very High		
Row Counts	<input checked="" type="checkbox"/>	Detail Rows	<input checked="" type="checkbox"/>	Subtotals	<input checked="" type="checkbox"/>	Grand Total	<input checked="" type="checkbox"/>

Sharing the Report with the Owner

- Edit Drop-Down:
 - Click the Edit drop-down menu.
 - Select the Subscribe option.
- Select Run Report As Another Person:
 - After selecting to run the report as “another person,” choose your personal account or the account of the person you want to send the email to.
- Save the Subscription



13. Dashboards: Real-Time Visualization

- **Purpose:** Provide visual snapshots of key metrics and trends.
- **Process:**

Creating a Dashboard Folder

- Open App Launcher: Search for Dashboard.
- Dashboard Tab: Click on the Dashboard tab.
- New Folder: Click New Folder.
- Label Folder: Name it Total Rent Amount (unique name auto-populates).
- Save Folder: Click Save.

Create folder

* Folder Label

* Folder Unique Name

Cancel
Save

Creating a Dashboard

- Dashboards Tab: Click on the Dashboards tab.
- New Dashboard: Name the dashboard and select the Total Rent Amount folder. Click Create.
- Add Component: Click Add Component.
- Select Report: Choose a report and click Select.
- Component Style: Choose the dark component style.
- Save Dashboard: Click Save.
- Finalize: Click Done.

The screenshot shows a Zoho Analytics dashboard titled "Data Analytics of Laptops". The dashboard has a header with navigation links: "Total Laptops", "consumer", "Laptop Bookings", "Billing Process", "Analytics Report", and a search bar. Below the header, there's a message: "Last refreshed 2 days ago. Refresh this dashboard to see the latest data." and "As of 27-Oct-2024, 1:38 am Viewing as Manasa Kuraku". The main area contains three reports:

- Amount based on Months:** A horizontal bar chart showing the sum of amount for each month. The data is as follows:

Month	Sum of Amount
1	₹1.2k
2	₹9k
3	₹12k
4	₹150
5	₹6.9k
- Analytics Report:** A donut chart showing the distribution of the sum of amount across different laptop brands. The data is as follows:

Laptop Names	Sum of Amount
Dell	₹6.9k
Acer	₹1.2k
Hp	₹9k
Mac	₹12k
- Versions based on Amount:** A bubble chart showing the sum of amount for different laptop versions. The data is as follows:

Sum of Amount
₹0
₹10k
₹20k
₹30k

5. Conclusion

The Salesforce CRM project for laptop rentals successfully established a structured environment with custom objects like Total Laptops, Consumer, Laptop Bookings, and Billing Process. Essential data fields were created, along with validation rules to ensure data integrity. User access was managed through profiles and roles, enhancing security and functionality. Automation through flows and insightful reports and dashboards enables data-driven decision-making. This setup positions the organization to efficiently manage rentals and improve customer engagement while remaining adaptable to future needs.