

A CRM APPLICATION FOR LAPTOP RENTALS

Project Overview:

CRM Application for Laptop Rentals:

The CRM Application for Laptop Rentals is a cloud-based system designed to streamline and automate the laptop rental process for businesses and service providers. This application helps manage customer interactions, track inventory, handle bookings, and monitor return-all within a single platform. By integrating key features like customer management, rental scheduling, and real-time reporting, the CRM enhances operational efficiency and improves customer satisfaction. Built on platforms like Salesforce, it ensures data security, scalability, and easy access, making it an ideal solution for managing large-scale laptop rentals.

Key features include:

- **Customer Management:** Centralized database for customer details (name, contact info, rental history, preferences). Enables quick access and personalized service for each customer.
- **Laptop Inventory Tracking:** Real-time updates of available, rented and under-maintenance laptops. Categorization by brand, specifications, and condition.
- **Rental Booking and Scheduling:** Easy interface to book, extend, or cancel rentals. Calendar view for tracking rental durations and upcoming returns.
- **Automated Notifications & Reminders:** Sends alerts for due dates, late returns, and promotional offers via email/SMS. Reduces missed deadlines and improves customer communication.
- **Payment and Billing Management:** Generates invoices, tracks payment status, and manages different payment modes. Includes tax calculation and downloadable billing reports.
- The CRM system manages customer details and rental history efficiently. It tracks laptop inventory in real time and simplifies the booking process. Automated reminders help with due dates and promotions. It also handles billing and payment tracking smoothly.

Objectives:

The CRM application for laptop rentals is designed to simplify and automate the entire rental process. It helps manage customer details, rental bookings, laptop issuance, and returns efficiently. One of its key goals is to improve customer satisfaction by offering personalized services and timely communication. It ensures smooth tracking of laptop inventory, including availability, damage status, and maintenance. The system reduces manual work and helps avoid errors or delays. It also generates reports that provide valuable insights into rental trends and customer behaviour. With features like payment tracking and digital records, it adds professionalism to the process. The CRM enables faster response to customer queries and enhances staff coordination. It aims to save time, reduce costs, and improve overall

productivity. Ultimately, it supports business growth by providing a reliable and scalable solution.

Phase 1: Requirement Analysis & Planning

Understanding Business Requirements: It is a crucial step in building an effective CRM application for laptop rentals. It involves identifying the main goals of the business, such as improving customer service, managing laptop inventory, increasing rental efficiency, and tracking payments. This step includes meetings with stakeholders like business owners, managers, and employees to understand their work flow, pain points, and expectations. It helps define what the system must achieve—like quick laptop booking, easy access to customer history, real-time availability of laptops, and automatic reminders. The better the business requirements are understood, the more accurately the CRM can be designed to meet actual business needs and support future growth.

Defining Project Scope and Objectives:

- To define the project scope and objectives of a CRM application for laptop rentals, we start by understanding what the business wants to achieve.
- Goals may include automating the rental process, improving customer service, or tracking inventory effectively.
- Next, we list all the expected features and functions of the system.
- These may include managing customer details, tracking laptop availability, handling bookings and returns, processing payments, and generating reports.
- It is also important to set boundaries by clearly stating what the system will not include in the current phase.
- This helps prevent scope creep and keeps the development focused.
- After this, we define specific, measurable objectives such as reducing processing time and reducing processing time and providing real-time inventory updates.
- Objectives can also include automating customer notifications and involving.
- Finally, we document the scope and objectives in a clear and organized way.
- All stakeholders must agree and understand what will be delivered before moving forward with design and development.

Design Data Model and Security Model:

The CRM application for laptop rentals includes a structured data model with key entities like customers, Laptops, Rentals, Payments, and Support tickets. Each rental links a customer to a laptop with payment and rental status details. The security model uses role-based access control, where admins have full access, sales staff manage rentals and payments, and technicians handle support tickets. Customers can only view their own records. Field and record-level security ensure data privacy and proper access control within the system.

Phase 2: Salesforce Development - Backend & Configurations

Setup environment & DevOps workflow: The CRM application for laptop rentals uses a simple DevOps workflow. Developers write and test code in the development environment. The code is saved in Git and tested automatically. After testing, it is moved to the staging environment for final checks. Once everything is fine, the code is deployed to the live production system. Monitoring tools help check performance and fix any issues.

Object creation:

Create Total Laptops Object

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

Create consumer Object

From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

- 1) Enter the label name >> consumer
 - 2) Plural label name >> consumer
 - 3) Enter Record Name Label and Format
Record Name >> consumer_name
Data Type >> Name
2. Click on Allow reports, Allow search and Track Field History,
3. Allow search >> Save.

Create Laptop Bookings Object

From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

- 1) Enter the label name >> Laptop Bookings
 - 2) Plural label name >> Laptop Bookings
 - 3) Enter Record Name Label and Format
 - Record Name >> Laptop Bookings
 - Data Type >> Name
2. Click on Allow reports, Allow search and Track Field History,
3. Allow search >> Save.

Create Billing Process Object

From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

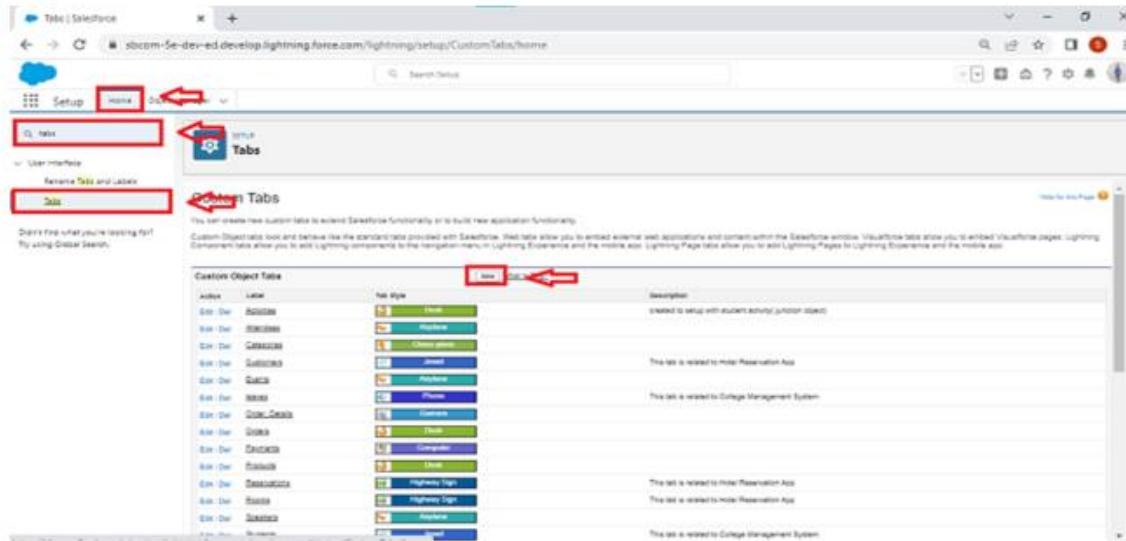
- 1) Enter the label name >> Billing Process
 - 2) Plural label name >> Billing Process
 - 3) Enter Record Name Label and Format
 - Record Name >> Billing Process Name
 - Data Type >> Name
2. Click on Allow reports, Allow search and Track Field History,
3. Allow search >> Save.

Tabs creating:

Creating a Custom Tab

To create a Tab:()

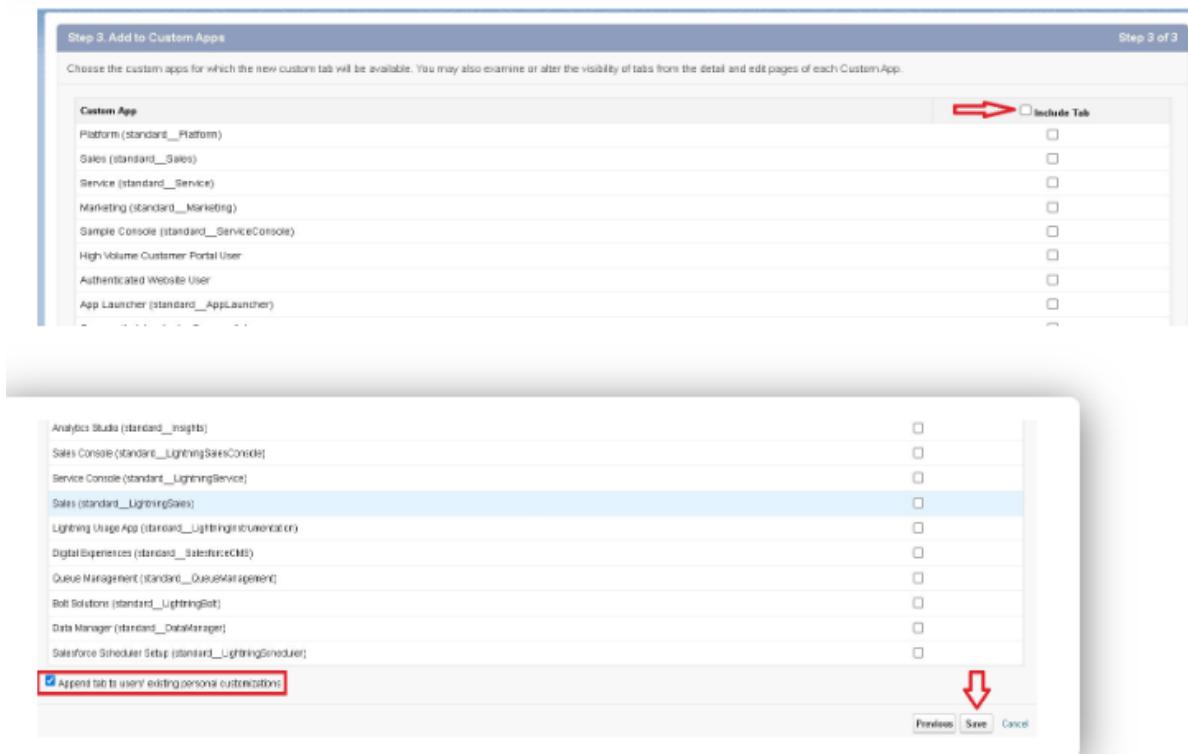
1. Go to setup page >> Type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)



2. 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.
3. Make sure that the Append tab to users' existing personal customizations is checked.
4. Click save.



5.



6.

Activity 2: Creating Remaining Tabs

1. Now create the Tabs for the remaining Objects, they are “consumer, Laptop Booking, Billing process”.
2. Follow the same steps as mentioned in Activity -1.

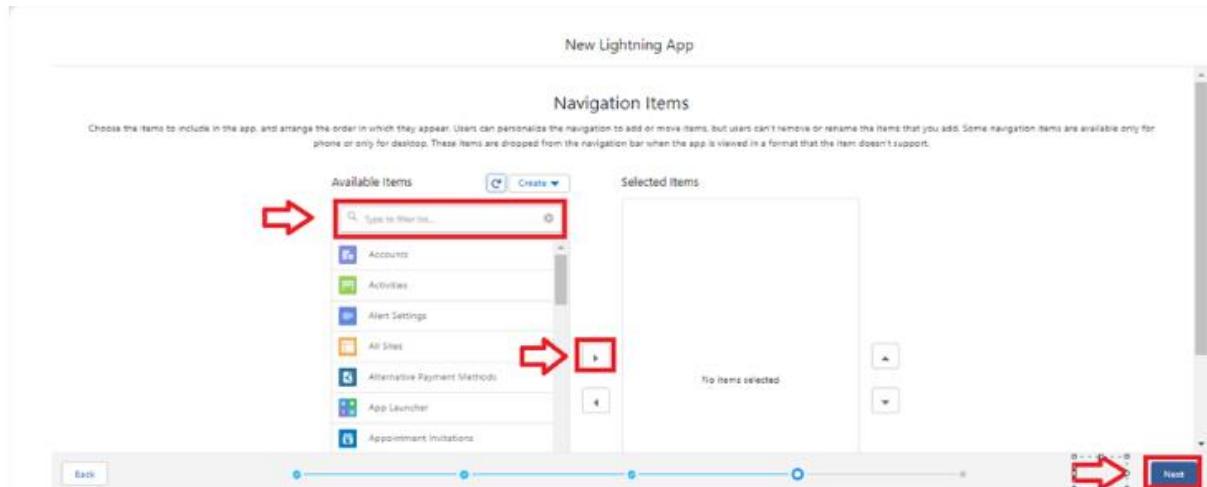
Lightning app creation:

To create a lightning app page:

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on new lightning App.

2. Fill the app name in app details as LAPTOP RENTALS >>Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next

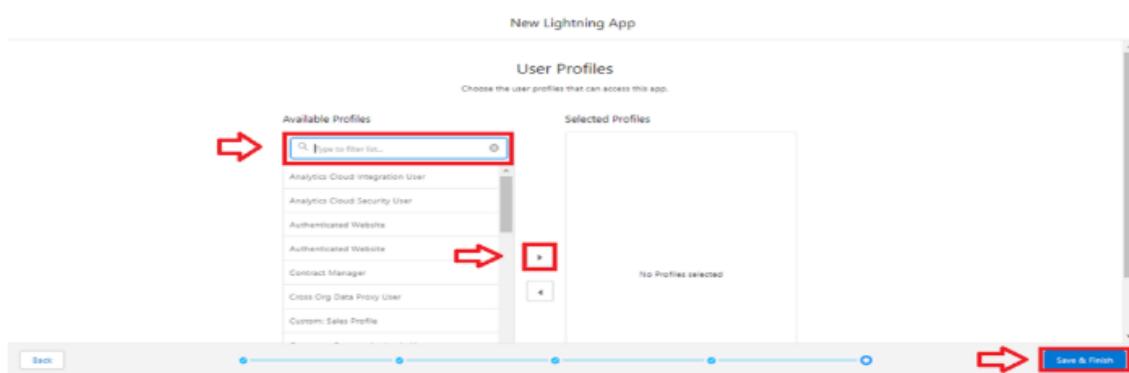
2. Upload a photo that is related to your app.
 3. To Add Navigation Items:
 4. Select the items (Total Laptops, consumer, Laptop Booking, Billing Process) from the search bar and move it using the arrow button >> Next.



5.

To Add User Profiles:

Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.



Customization of fields:

Creating the field in consumer object

- **1. To create fields in an object:**

1. Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Phone”
4. Click on next
5. Fill the Above as following:
 - Field Label: Phone number
 - Field Name: gets auto generated

- Click the required option checkbox.
- Click on Next >> Next >> Save and new.

Step 2. Enter the details

Field Label: Phone number

Field Name: phone_number

Description:

Help Text:

Required: Always require a value in this field in order to save a record

Auto-add to custom report type: Add this field to existing custom report types that contain this entity

Default Value: (Show Formula Editor)

Help Text: This formula editor provides a simple way to define formulas in Visualforce pages. It includes functions for date calculations, string manipulation, and numeric operations. To reference a field from a Custom Metadata type record use: @CustomMetadataType__Name__Field__Value__

To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Email” and Click on Next
4. Fill the Above as following:
 - Field Label: Email
 - Field Name: It’s gets auto generated
 - Click on Next >> Next >> Save and new.

Step 2. Enter the details

Field Label: Email

Field Name: Email

Description:

Help Text:

Required: Always require a value in this field in order to save a record

Unique: Do not allow duplicate values

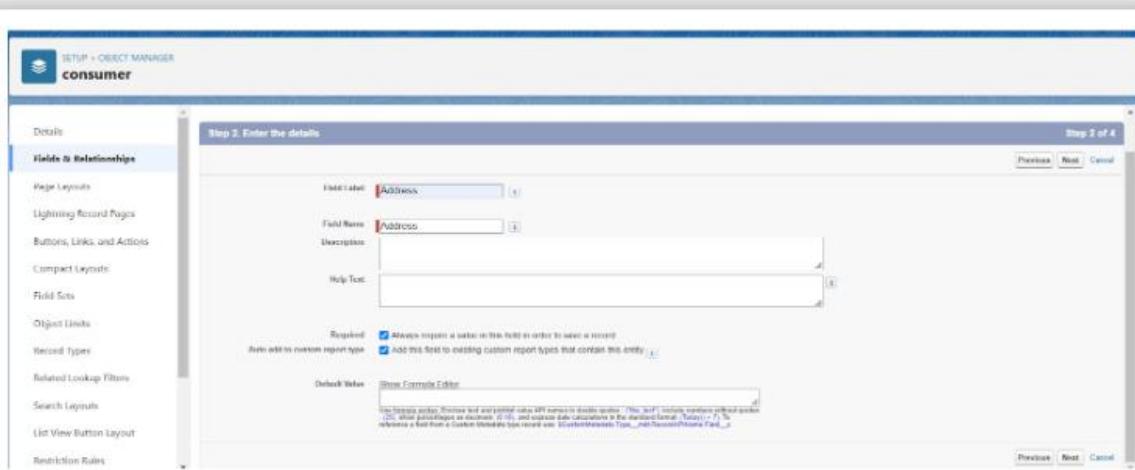
External ID: Set this field as the unique record identifier from an external system

Auto-add to custom report type: Add this field to existing custom report types that contain this entity

Default Value: (Show Formula Editor)

To create another fields in an object:

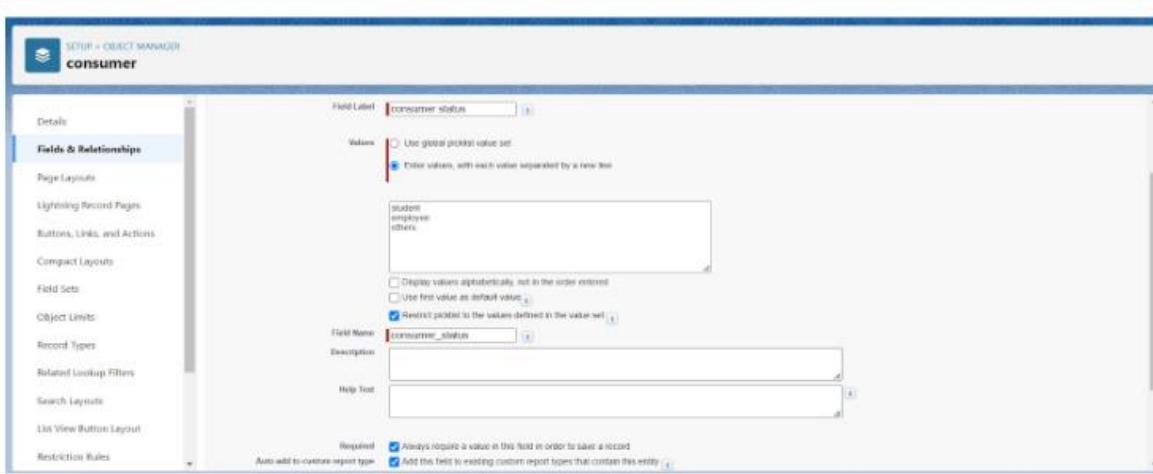
1. Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Text Area” and Click on Next
4. Fill the Above as following:
 - Field Label: Address
 - Field Name: It’s gets auto generated
 - Select Required field.
 - Click on Next >> Next >> Save and new.



To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Picklist” and Click on Next
4. Fill the Above as following:
 - Field Label: consumer Status
 - Value - Select enter values with each value separated by a new line
 1. Student
 2. Employee
 3. Others
 - Select required
 - Field Name: It’s gets auto generated

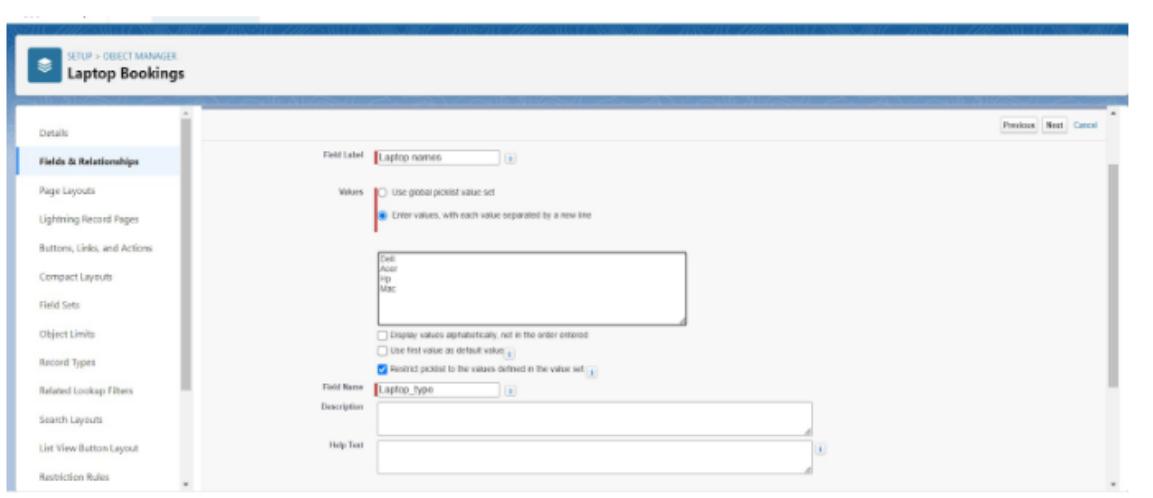
- Click on Next >> Next >> Save and new.



Creating the field in Laptops Bookings object

1. To create fields in an object:

- Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
- Now click on “Fields & Relationships” >> New
- Select Data Type as a “Picklist”
- Label: Laptop Names
- Picklist values are: 1. Dell 2. Acer 3. Hp 4. Mac

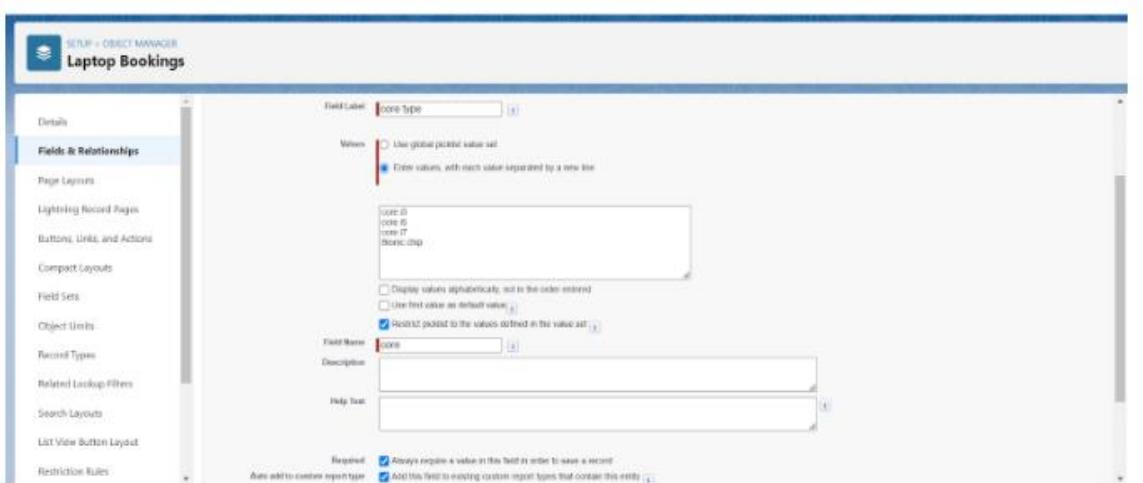


- Select required
- Click on Next >> Next >> Save and new

2. To Create a Fields & Relationship to An Laptop Booking Object

To create fields & relationship to an object:

- 1.Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
- 2.Now click on “Fields & Relationships” >> New
- 3.Select Data Type as a “Picklist” and Label: Core Type
- 4.Picklist values are:1. core i3 2. Core i5 3. Core i7 4. Bionic Chip.



5.Select required

6.Click on Next >> Next >> Save and new

3. To Create a Field Dependency in the Laptop Booking Object

To create field dependency to an object:

- 1.Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
- 2.click field dependency and next
3. Select **Controlling Field** as Laptop Names and **Dependent Field** as Core Type
4. Click the include value for dell-core i3, i5, i7 and for acer i3, i5, i7 and for hp i3, i5, i7 and also for mac bionic chip include the values for it.
5. Click save.

6.

To Create a Fields & Relationship to An Laptop Booking and Total Laptops Objects

To create fields & relationship to an object:

1. Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Master-Detail Relationship”
4. Click on Next
5. Click on the Related to drop down and select the “consumer” object and click on Next
6. Fill the Above as following:
 - Change the Field Label: Consumer
 - Field Name: It’s gets auto generated
7. Click on Next >> Next >> Save and new.

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
 2. Now click on “Fields & Relationships” >> New
 3. Select Data Type as a “Currency”
 4. Click on Next
- Fill the Above as following:

- Field Label: Amount
- Length: (18,0)
- Field Name: It's gets auto generated
- Click on Next >> Next >> Save and new

SETUP > OBJECT MANAGER
Laptop Bookings

Step 2. Enter the details Step 2 of 4

Field Label	Amount
Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".	
Length	18
Field Name	Amount
Description	Amount
Help Text	Amount
Required	<input checked="" type="checkbox"/> Always require a value in this field in order to save a record
Auto add to custom report type	<input checked="" type="checkbox"/> Add this field to existing custom report types that contain this entity
Default Value	Show Formula Editor

To Create a Fields & Relationship to an Object

1. Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Lookup Relationship”
4. Click on Next
- 5.

SETUP > OBJECT MANAGER
Laptop Bookings

Laptop Bookings New Relationship Step 2 of 6

Step 2. Choose the related object

Select the other object to which this object is related

Related to: Total laptops

6. Click on the Related to drop down and select the “Total Laptops” object and click on Next

- Fill the Above as following:
- Change the Field Label: Total No of Laptops
- Field Name: It's gets auto generated
- Click on Next >> Next >> Save and new.

4. To Create a Fields & Relationship to An Laptop Booking Object

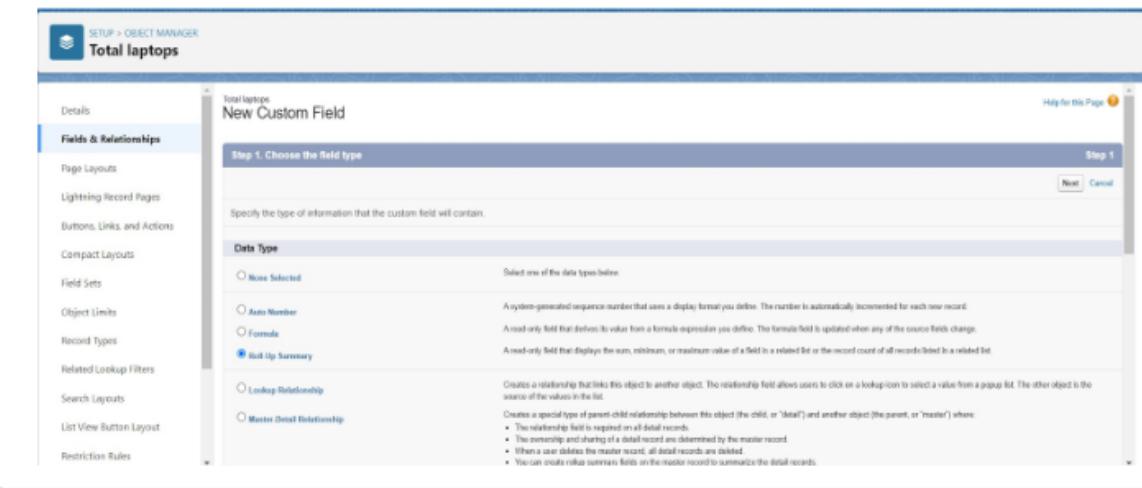
To create fields & relationship to an object:

8. Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
9. Now click on “Fields & Relationships” >> New
10. Select Data Type as a “Email”
11. Click on Next and save it.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount_c	Currency(10, 0)		
core type	core_c	PlainText	Laptop names	
Created By	CreatedById	Lookup(User)		
Laptop Bookings Name	Name	Text(80)		
Laptop names	Laptop_type_c	PlainText		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name_c	Master-Detail(consumer)		
Total no of laptops	Total_no_of_laptops_c	Master-Detail(total laptops)		

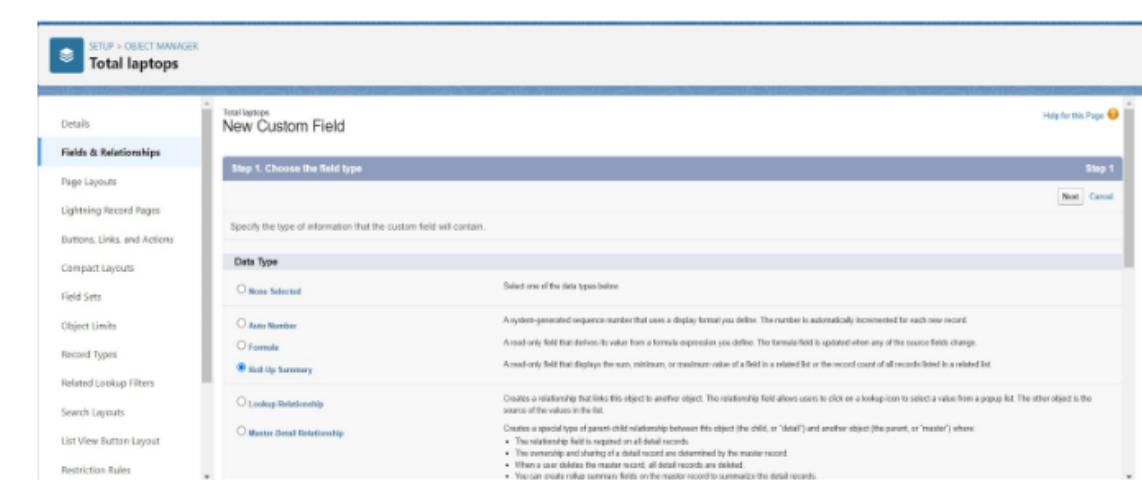
To Create a Rollup Summary Field in “Total Laptops Object”

1. After Creating the Master-Detail Relationship Than Only you can create the Rollup Summary
2. Go to setup >> click on Object Manager >> type object name (Total Laptops) in the search bar >> click on the object.
3. Now click on “Fields & Relationships” >> New



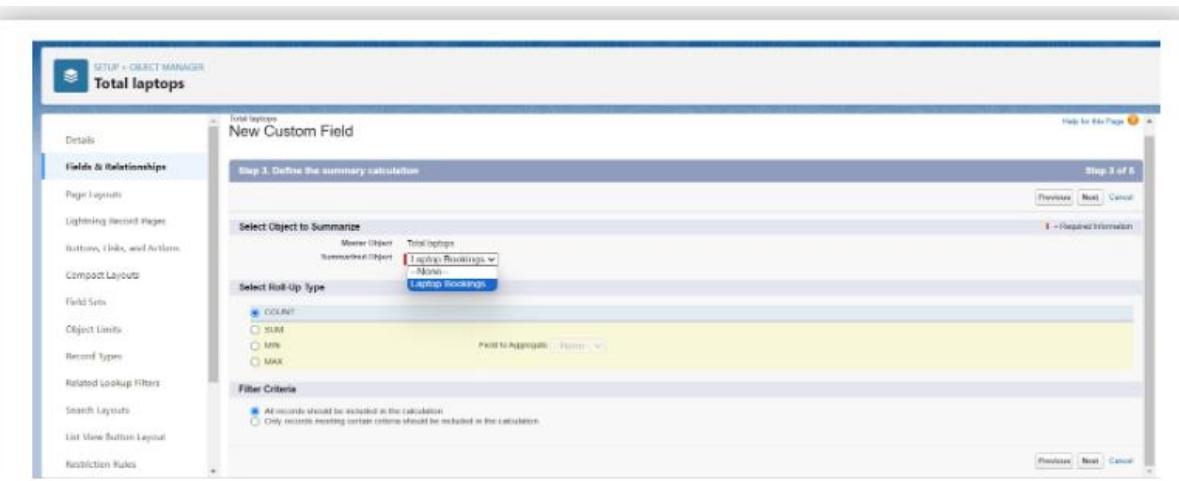
4. Select Data type as a “Roll-up Summary” and Click on Next

- Fill the Above as following:
- Field Label: Laptops delivered
- Field Name: It's gets auto generated



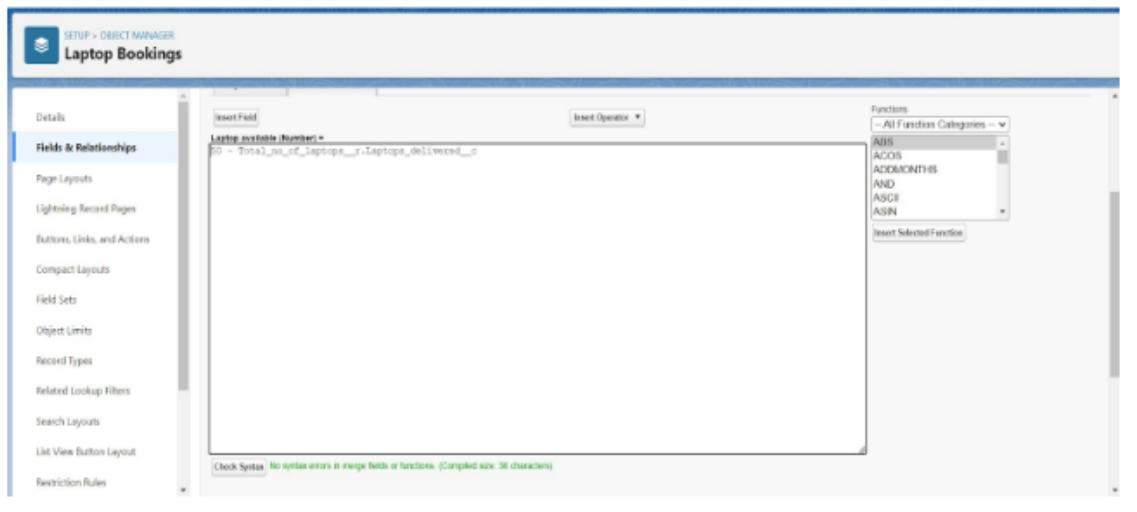
- Click on Next

6. Select the Laptop Bookings in the Summarized Object
7. Select the count Radio button in the select Roll-up Type



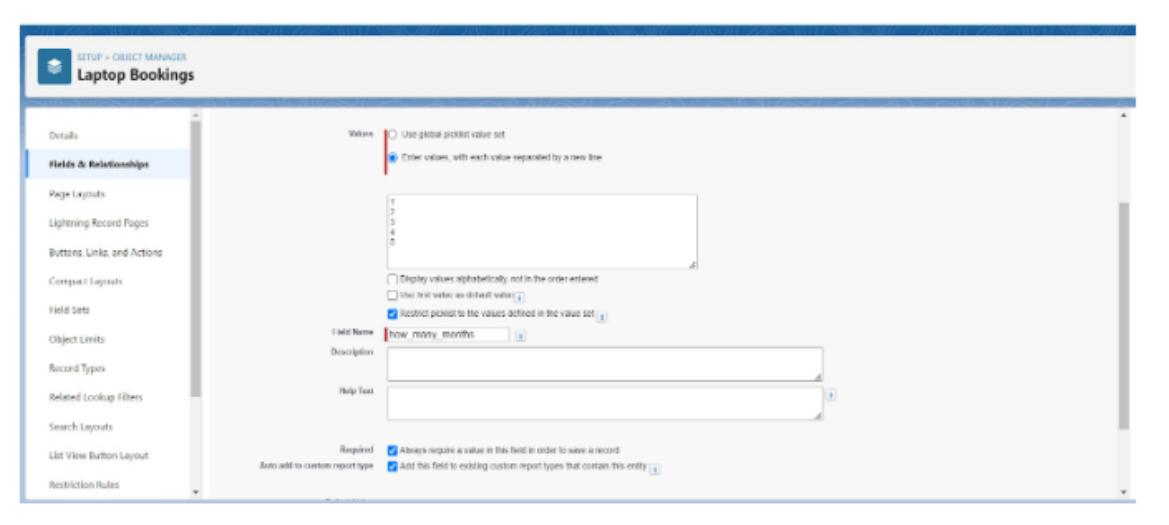
To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Formula” and Click on Next
4. Fill the Above as following:
 - Field Label: Laptops Available
 - Field Name: It's gets auto generated
 - Select the Formula Return Type as “Number”
 - Select the Decimal places as “0” and Click on Next
 - Click on the Advanced Formula and Enter the value in formula box “50” and click on insert field than you will find a pop window under the Laptop Booking select the Total No of Laptops in the second Column and select the laptops delivered in the third column and click on insert
 - “50 - Total_no_of_laptops __ r. Laptops_delivered __ c” and Check Syntax
 - Click on Next >> Next >> Save and new



To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Laptop Booking) in the search bar >> click on the object.
3. Now click on “Fields & Relationships” >> New
4. Select Data Type as a “picklist” and Label: how many months
5. Picklist values are 1.2.3.4.5
6. Click and save it.



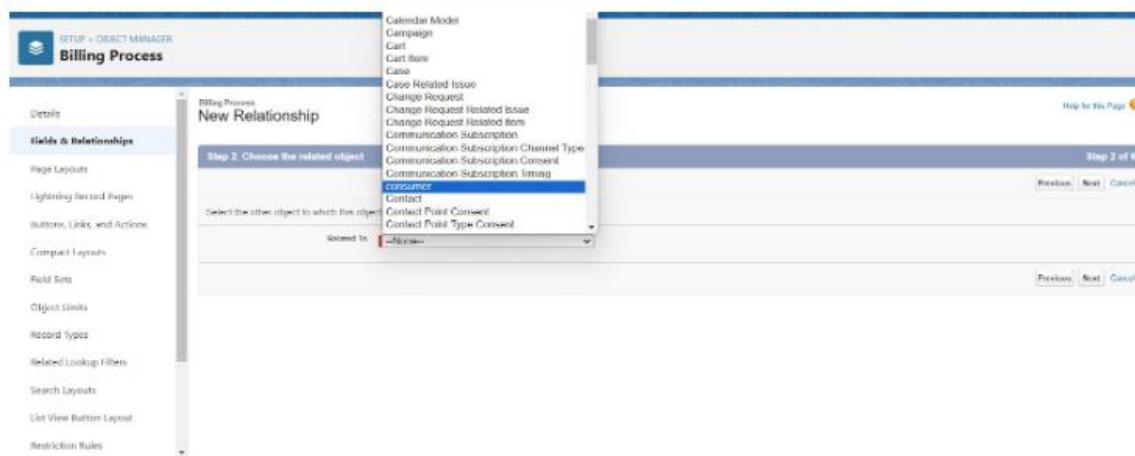
Creation of Fields & Relationship for Billing Process Object

1.

1. To create fields & relationship to an object:

1. Go to setup >> click on Object Manager >> type object name (Billing Process) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Master-detail Relationship”
4. Click on Next

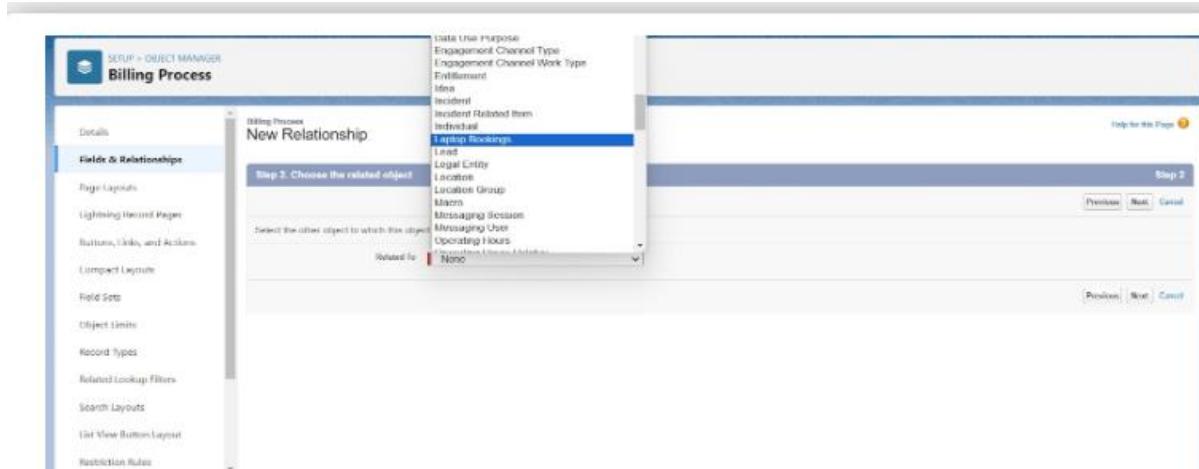
- Click on the Related to drop down and select the consumer object and click on Next



- Fill the Above as following:
 - Change the Field Label: Name
 - Field Name: It's gets auto generated
 - Click on Next >> Next >> Save and new.

2. To create another fields & relationship to an object:

- Go to setup >> click on Object Manager >> type object name (Billing Process) in the search bar >> click on the object.
- Now click on “Fields & Relationships” >> New
- Select Data Type as a “Lookup Relationship”
- Click on Next
- Click on the Related to drop down and Select the Laptop Booking object and click on Next



1. Fill the Above as following:
 - Change the Field Label: Laptop Booking
 - Field Name: It's gets auto generated
 - Click on Next >> Next >> Save and new.
3. Creation of another fields for the billing process object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Billing Process) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Picklist”
4. Fill the Above as following:
 - Field Label: Payment Mode
 - Value >> Select enter values with each value separated by a new line
 1. Cash
 2. Check
 3. Credit card
 4. Debit card
 5. UPI
 6. Phone pay
 7. G pay
 8. Paytm
 - Select required
 - Click on Next >> Next >> Save and new.

Cross Object Formula Field:

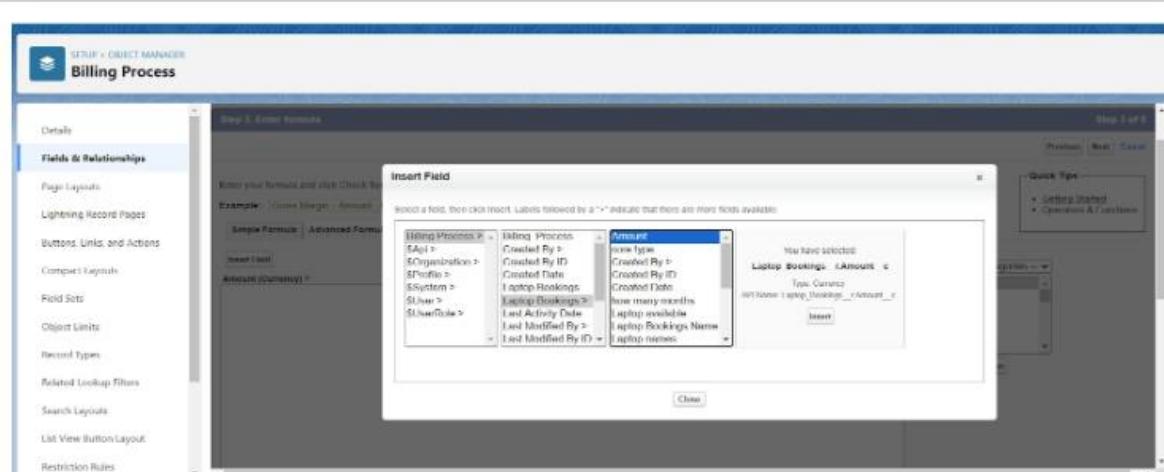
In Salesforce, a cross-object formula field allows you to create a formula that references fields from related objects. It enables you to perform calculations or display data from related records without the need for custom code or complex workflows.

Why do we need to create the Cross Object Formula Field:

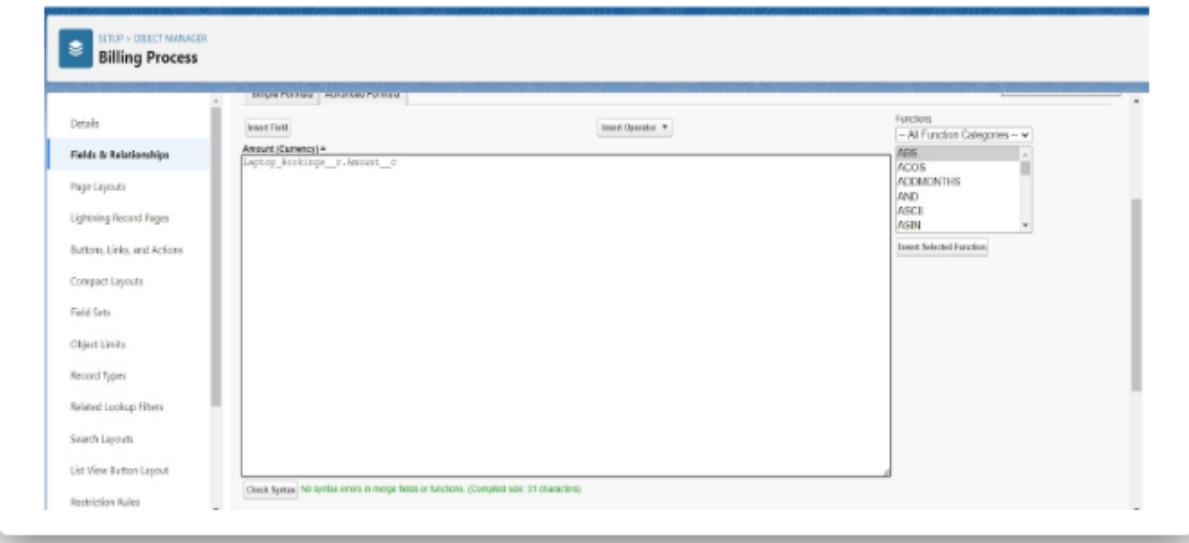
If we want to get the Particular field from another object in that case we will use the Cross object Formula field. For that First we need to create the relationship b/w two objects and relate the field with formula data type.

4. Create a Cross Object formula Field in billing process Object

1. Go to setup >> click on Object Manager >> type object name (Billing Process) in the search bar >> click on the object.
 2. Now click on “Fields & Relationships” >> New
 3. Select Data Type as a “Formula”
 4. Click on Next
 5. Enter the Field label: Amount, the Field name gets auto generated and click on Next. (Formula return type Currency).
 6. In the Advanced Formula Click on the Insert field in the popup Screen Select the Billing Process and in the second drop down select the Laptop Booking and in the three drop down select the Amount field and click on Insert
 7. “Laptop _ Booking __ r. Amount __ c”.
 8. Click on the Check syntax: No syntax errors in merge fields



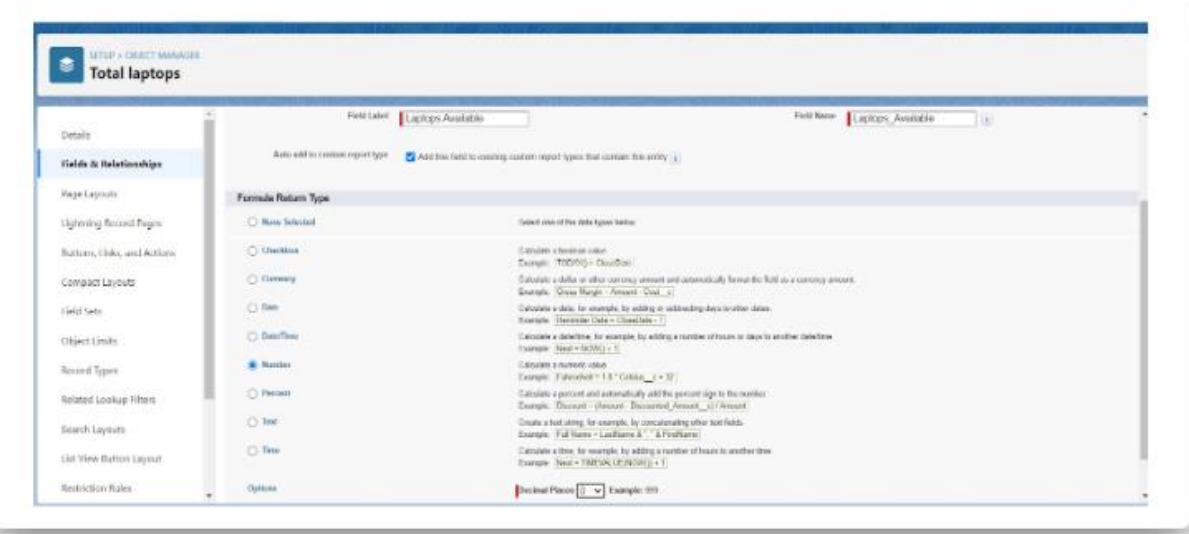
9.Click on Next >> Next >> Save and new.



Creating the field in Total Laptops object

To create fields in an object:

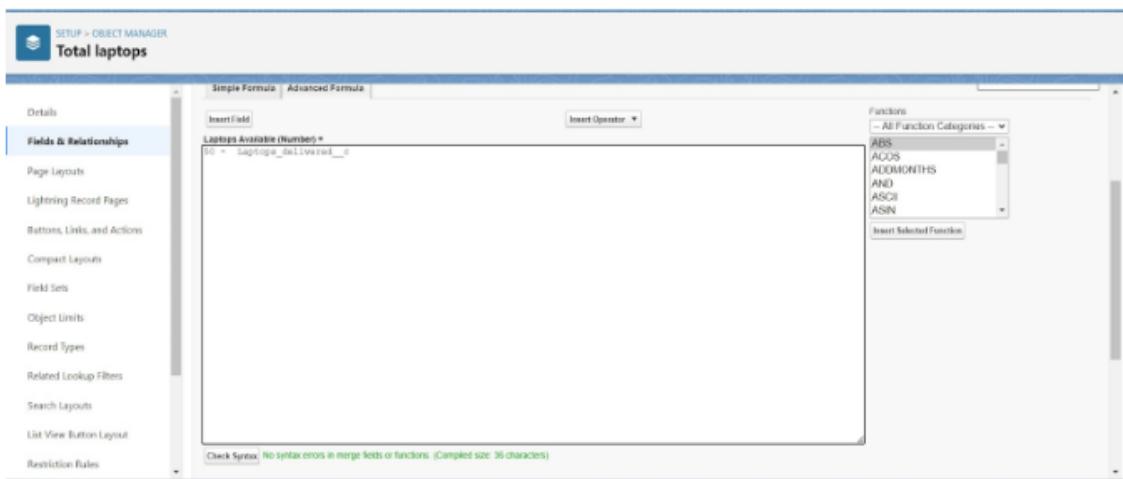
1. Go to setup >> click on Object Manager >> type object name (Total Laptops) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Formula” and Click on Next
4. Fill the Above as following:
5. Field Label: Laptops Available
6. Field Name: It’s gets auto generated
7. Select the Formula Return Type as “Number”



8. Select the Decimal places as “0” and Click on Next

Note: I am Considering “Total No Of Laptops = 50” While creating a new record in Total Laptops Object.

1. Click on the Advanced
2. Formula “50 – Laptops _ delivered __c” and Check Syntax



3. Click on Next >>Next >>Save and new.

Customization of validation rule:

Creating the validation rule for phone number field in consumer object

Creating the validation rule for phone number field in consumer object

1. Go to the setup page >> click on object manager >> From drop down click edit for consumer object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “Phone number or email blank rule”.
4. Enter the description as “phone number and email number should not be blank”.
5. Enter the formula as “OR (ISBLANK (phone _ number __c), ISBLANK (email __ c))” and check the syntax.



6. Save the validation rule.

Customization of profile:

owner Profile

To create a new profile:

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

The screenshot shows the 'Profiles' page in Salesforce Setup. The 'owner' profile is selected. Key details include:

- Name:** owner
- User License:** Salesforce
- Description:** (empty)
- Created By:** [Administrator](#) [10/01/2023, 10:56 am]
- Modified By:** [Administrator](#) [10/01/2023, 10:56 am]
- Custom Profile:** ✓

Page Layouts

Standard Object Layouts	Global	Object Layout	Object Milestone Layout
Lead	Global Layout	[View Assignment]	[View Assignment]
Opportunity	Not Assigned	[View Assignment]	Operating Hours Layout
Order	Not Assigned	[View Assignment]	Opportunity Layout
Order Product	Not Assigned	[View Assignment]	Opportunity Product Layout
Product2	Not Assigned	[View Assignment]	Order Layout
Task	Not Assigned	[View Assignment]	Order Product Layout
Work Order	Not Assigned	[View Assignment]	[View Assignment]

2. 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.

The screenshot shows the 'Custom Object Permissions' section of the profile configuration. A blue arrow points to the 'Billing Process' row under 'Custom Object Permissions'.

Object	Basic Access						Data Administration					
	Read	Create	Edit	Delete	View All	Modify All	Read	Create	Edit	Delete	View All	Modify All
Billing Process	<input checked="" type="checkbox"/>	Laptop Bookings	<input checked="" type="checkbox"/>									
consumers	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Total Laptops	<input checked="" type="checkbox"/>				

3. Give Access and Save it.

Agent Profile

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard Platform User) >> enter profile name (Agent) >> Save.
2. While still on the profile page, then click Edit.
3. 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.

Creating a Role and hierarchy:

Creating owner Role

Creating owner Role:

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click on Expand All and click on add role under whom this role works.

```

graph TD
    Root["Nick Enterprises"] --> CEO["CEO"]
    CEO --> HR["HR"]
    CEO --> Manager["Manager"]
    CEO --> OnSite["On Site Emp"]
    CEO --> Remote["Remote Emp"]
    HR --> AddRole1["Add Role"]
    Manager --> AddRole2["Add Role"]
    OnSite --> AddRole3["Add Role"]
    Remote --> AddRole4["Add Role"]
  
```

1. Give Label as “owner” and Role name gets auto populated. Then click on Save.

2. Click and save it.

Activity 2: Creating Agent roles

Creating another two roles under Owner

1. Go to quick find - Search for Roles - click on set up roles.
2. Click plus on CEO role, and click add role under owner.

The screenshot shows the 'Creating the Role Hierarchy' page in the Salesforce Setup. The title bar says 'SETUP Roles'. Below it, a message says 'You can build on the existing role hierarchy shown on this page. To insert a new role, click Add Role.' A 'Your Organization's Role Hierarchy' tree is displayed. The root node is 'smartbridge'. Under 'smartbridge', there are several nodes: 'CEO', 'CFO', 'COO', 'HR', 'owner', 'SVP_Customer Service & Support', 'SVP_Human Resources', and 'SVP_Sales & Marketing'. Each node has 'Edit | Del | Assign' options. Under each node, there is an 'Add Role' button. The 'owner' node is currently selected, indicated by a red box around its 'Edit | Del | Assign' options. The right side of the screen has a 'Help for this Page' link and a 'Show in tree view' dropdown menu.

3. Give Label as “Agent” and Role name gets auto populated. Then click on Save.

Create User

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

The screenshot shows the 'User Edit' screen for a new user. The 'General Information' section contains the following data:

Field	Value
First Name	vicky
Last Name	rushti
Alias	rushti
Email	vickyru@00715673@gmail.com
Username	vickyru@00715673@gmail.com
Nickname	vicky
Title	
Company	
Department	
Division	

The 'Role' dropdown is set to 'OWNER'. Under 'User License', 'Salesforce' is selected. The 'Profile' dropdown is set to 'Standard User'. The 'Active' checkbox is checked. Other sections include 'Marketing User', 'Office User', 'Knowledge User', 'Flow User', 'Service Cloud User', 'Site.com Contributor User', 'Site.com Publisher User', 'WDC User', 'Data.com User Type' (set to 'None'), and 'Data.com Monthly Addition Limit' (set to 'Default Limit (300)').

Save it.

Activity 2: creating another user

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

12. Save it.

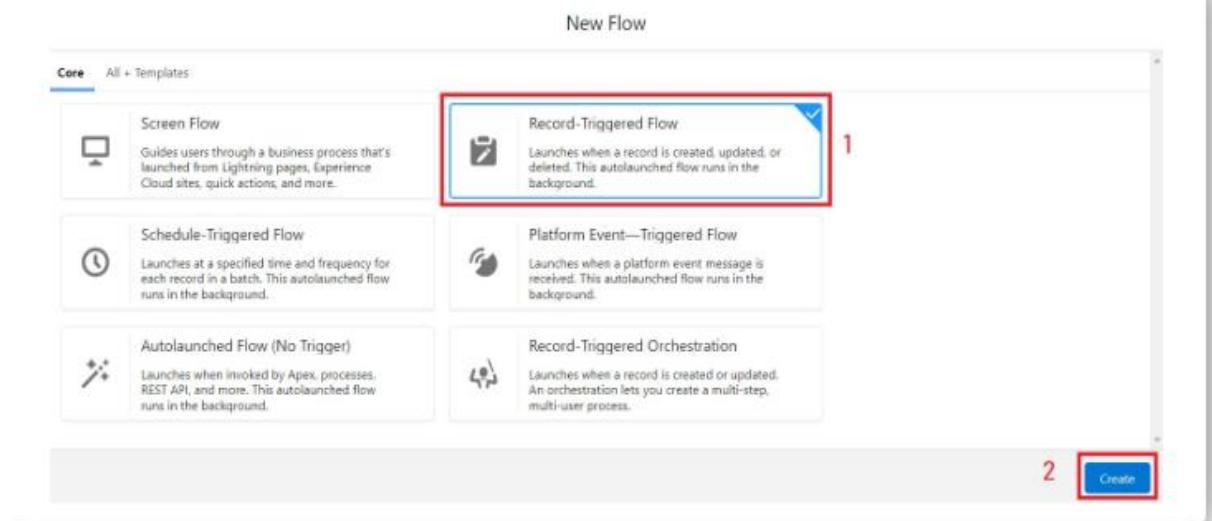
Customization of flows:

Create a Flow on dell laptop

Activity -

1. Go to setup >>type Flow in quick find box >> Click on the Flow and Select the New Flow.

2. Select the Record-triggered flow and Click on Create.



3. Select the Object as a Laptop Booking in the Dropdown list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimize the flow for: “Actions and Related Records” and Click on Done.

The screenshot shows the 'Configure Start' step of the flow creation wizard. The title is 'Configure Start'. The first section is 'Select Object', which asks to select the object whose records trigger the flow. A note says: 'Select the object whose records trigger the flow when they're created, updated, or deleted.' A dropdown menu labeled 'Object' contains 'Laptop Bookings', which is selected. The second section is 'Configure Trigger', which asks to trigger the flow when a record is created or updated. A note says: 'Trigger the flow when:' followed by radio buttons for 'A record is created', 'A record is updated', 'A record is created or updated' (which is selected), and 'A record is deleted'. The third section is 'Set Entry Conditions', which asks to specify entry conditions to reduce unnecessary flow executions. A note says: 'Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.' A note at the bottom says: 'If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the Only when a record is updated to meet the condition requirements option for When to Run the Flow for Updated Records.' At the bottom right are 'Cancel' and 'Done' buttons.

Configure Start

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements:

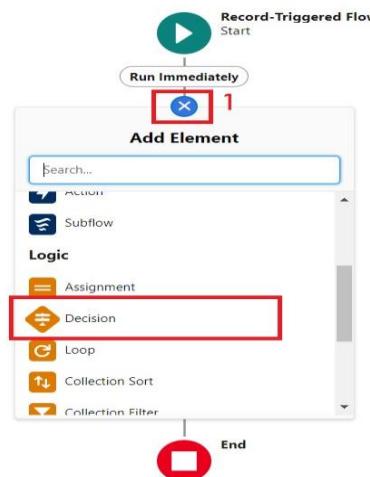
***Optimize the Flow for:**

Fast Field Updates
Update fields on the record that triggers the flow to run. This high-performance flow runs before the record is saved to the database.

Actions and Related Records
Update any record and perform actions, like send an email. This more flexible flow runs after the record is saved to the database.

Include a Run Asynchronously path to access an external system after the original transaction for the triggering record is successfully committed

6. Under the Record-triggered Flow Click on “+” Symbol and In the Dropdown List select the “Decision Element”.



7. Enter the Details Label: Field should be Update, API name: Gets Automatically Generated.

8. Enter the Outcome Details Label: dell, Outcome API name: Gets Automatically Generated.

- Resource: Select \$Record. Laptop_name __c.
- Operator: Select Equals.
- Value: Select dell
- Add the same outcome order to acer, hp, mac.
- Rename Default outcome as False
- Click done.

Edit Decision

* Label field should updated	* API Name 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 + OUTCOME DETAILS							
<ul style="list-style-type: none"> dell acer hp mac false 	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> * Label dell * Outcome API Name dell <small>Condition Requirements to Execute Outcome</small> All Conditions Are Met (AND) </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Resource</th> <th style="text-align: left; padding: 2px;">Operator</th> <th style="text-align: left; padding: 2px;">Value</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">\$Record > Laptop names X</td> <td style="padding: 2px;">Equals</td> <td style="padding: 2px;">Dell</td> </tr> </tbody> </table>	Resource	Operator	Value	\$Record > Laptop names X	Equals	Dell
Resource	Operator	Value					
\$Record > Laptop names X	Equals	Dell					
Delete Outcome							
Cancel Done							

9. Go to flow page
11. Beside dell there is a symbol ‘+’ click on that.
12. Again select decision.
13. Enter the Details Label: Field should Update (any one u want), API name: Gets Automatically Generated.
14. select the Outcome Details Label: dell core i3, Outcome API name: Gets Automatically Generated.
 - Resource: Select { !\$ Record. Core _ type__ c }.
 - Operator: Select Equals.
 - Value: Select core i3.
 - Then again click the symbol ‘+’ outcome details
15. 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
16. 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.
17. Click done.
-
- 18.
19. So go to the flow page select ‘+’ after core i3 then again select the decision.
20. Enter the Details Label: months selected, API name: Gets Automatically Generated.
21. Enter the Outcome Details Label: dell 1(i3), Outcome API name: Gets Automatically Generated.
- 22.
- Resource: Select Record. how many months.
 - Operator: Select Equals.
 - Value: 1.
23. 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.
24. Click ‘+’ outcome details
25. 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.
26. Click ‘+’ outcome details
27. 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.
28. Click ‘+’ outcome details
29. 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.

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>Delete Outcome</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>*Label 1 *Outcome API Name X1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)</td> <td></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>Resource \$Record > how many months X Operator Equals Value 1</td> <td></td> </tr> </tbody> </table>		OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome	1	*Label 1 *Outcome API Name X1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)		2			3			4			5	Resource \$Record > how many months X Operator Equals Value 1	
OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome																	
1	*Label 1 *Outcome API Name X1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)																		
2																			
3																			
4																			
5	Resource \$Record > how many months X Operator Equals Value 1																		

- 30.
31. Follow the above picture you will understand.
32. After dell 1(i3) there is ‘+’ symbol like dell 2(i3), dell 3(i3), dell 4(i3), dell 5(i3).
33. Click on ‘+’ then select update records for each and every outcome.
34. Enter the Details Label: one month of dell i3 rate, API name: Gets Automatically Generated. Enter label name in a similar way for other update records.
35. 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 all these finally and update the Amount field value based on the number of months.

36. Click done.
37. Enter the Details Label: months selected, API name: Gets Automatically Generated.
38. Enter the Outcome Details Label: dell 1(i7), Outcome API name: Gets Automatically Generated.
39.
 - Resource: Select Record. how many months.
 - Operator: Select Equals.
 - Value: 1.
40. 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.
41. Click ‘+’ outcome details
42. 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.
43. Click ‘+’ outcome details
44. 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.
45. Click ‘+’ outcome details
46. 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>Delete Outcome</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>*Label 1 *Outcome API Name X1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)</td> <td></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>Resource \$Record > how many months X Operator Equals Value 1</td> <td></td> </tr> </tbody> </table>		OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome	1	*Label 1 *Outcome API Name X1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)		2			3			4			5	Resource \$Record > how many months X Operator Equals Value 1	
OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome																	
1	*Label 1 *Outcome API Name X1 Condition Requirements to Execute Outcome All Conditions Are Met (AND)																		
2																			
3																			
4																			
5	Resource \$Record > how many months X Operator Equals Value 1																		

47.

48. Follow the above picture you will understand.
49. After dell 1(i7) there is ‘+’ symbol like dell 2(i7), dell 3(i7), dell 4(i7), dell 5(i7).
50. Click on ‘+’ then select update records
51. Enter the Details Label: one month of dell i5 rate, API name: Gets Automatically Generated.
52. 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
53. Click done.
54. **Follow the steps from 37 to 53 for Dell i5 and update the Amount for each month (1,2,3,4,5) as 1500, 2500, 3500, 4500, 5500 respectively.**

creating flow on acer laptop

1.
 1. Go to flow page
2. Beside acer there is a symbol ‘+’ click on that.
3. Again select decision
4. Enter the Details Label: Acer core type selection, API name: Gets Automatically Generated.

5. Select the Outcome Details Label: acer core i3, Outcome API name: Gets Automatically Generated.

- Resource: Select Record. core type.
- Operator: Select Equals.
- Value: Select core i3.

6. Similarly create outcomes for acer core i5 and acer core i7 also.

Edit Decision

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	OUTCOME DETAILS	Actions
acer core i3	*Label: acer core i3 *Outcome API Name: acer_core_i3	Delete Outcome
acer core i5		
acer core i7		
Default Outcome		

Condition Requirements to Execute Outcome: All Conditions Are Met (AND)

Resource: \$Record > core type, Operator: Equals, Value: core i3

+ Add Condition

When to Execute Outcome:

If the condition requirements are met
 Only if the record that triggered the flow to run is updated to meet the condition requirements

Cancel Done

Click done.

7. Go to flow page

8. Beside dell there is a symbol '+' click on that.

9. Again select decision

10. Enter the Details Label: Acer months selected, API name: Gets Automatically Generated.

11. Enter the Outcome Details Label: acer 1(i3), Outcome API name: Gets Automatically Generated.

12.

- Resource: Select Record. how many months.
- Operator: Select Equals.
- Value: 1.

16. Enter the Outcome Details Label: acer 2(i3), Outcome API name: Gets Automatically Generated.

- Resource: Select Record. how many months.
- Operator: Select Equals.
- Value: Select 2.

20. Click ‘+’ outcome details

21. Enter the Outcome Details Label: acer 3(i3), Outcome API name: Gets Automatically Generated.

- Resource: Select Record. how many months.
- Operator: Select Equals.
- Value: Select 3.

25. Click ‘+’ outcome details

26. Enter the Outcome Details Label: acer 4(i3), Outcome API name: Gets Automatically Generated.

- Resource: Select Record. how many months.
- Operator: Select Equals.
- Value: Select 4.

30. Click ‘+’ outcome details

31. Enter the Outcome Details Label: acer 5(i3), Outcome API name: Gets Automatically Generated.

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

Edit Decision

* Label acer months selected	* API Name acer_months_selected																			
Description 																				
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.																				
<table border="1"> <thead> <tr> <th>OUTCOME ORDER</th> <th>OUTCOME DETAILS</th> <th>Delete Outcome</th> </tr> </thead> <tbody> <tr> <td>acer 1(i3)</td> <td>* Label acer 1(i3)</td> <td>* Outcome API Name acer_1_i3</td> </tr> <tr> <td>acer 2(i3)</td> <td colspan="2">Condition Requirements to Execute Outcome All Conditions Are Met (AND)</td> </tr> <tr> <td>acer 3(i3)</td> <td colspan="2"></td> </tr> <tr> <td>acer 4(i3)</td> <td colspan="2"></td> </tr> <tr> <td>acer 5(i3)</td> <td>Resource \$Record > how many months</td> <td>Operator Equals</td> <td>Value 1</td> </tr> </tbody> </table>		OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome	acer 1(i3)	* Label acer 1(i3)	* Outcome API Name acer_1_i3	acer 2(i3)	Condition Requirements to Execute Outcome All Conditions Are Met (AND)		acer 3(i3)			acer 4(i3)			acer 5(i3)	Resource \$Record > how many months	Operator Equals	Value 1
OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome																		
acer 1(i3)	* Label acer 1(i3)	* Outcome API Name acer_1_i3																		
acer 2(i3)	Condition Requirements to Execute Outcome All Conditions Are Met (AND)																			
acer 3(i3)																				
acer 4(i3)																				
acer 5(i3)	Resource \$Record > how many months	Operator Equals	Value 1																	
Cancel Done																				

Click done.

35. After acer 1(i3) there is ‘+’ symbol like acer 2(i3), acer 3(i3), acer 4(i3), acer 5(i3).
36. Click on ‘+’ then select update records
37. Enter the Details Label: one month of acer i3 rate, API name: Gets Automatically Generated.
38. Field: Amount__ c, value: for acer 1(i3)-900, acer 2(i3)-1800, acer 3(i3)-2700, acer 4(i3)-3600, acer 5(i3)-4800. Follow for all these finally
39. **Follow the steps from 8 to 22 and create decision and Update record elements for Acer core i5 and Acer core i7. But enter different Amount values.**

Edit Update Records

one month of acer i3 rate (one_month_of_acer_i3_rate) 

* How to Find Records to Update and Set Their Values

Use the laptop bookings record that triggered the flow

Update records related to the laptop bookings record that triggered the flow

Use the IDs and all field values from a record or record collection

Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record

None—Always Update Record 

Set Field Values for the Laptop Bookings Record

Field		Value	
Amount_c		900	
+ Add Field			

Cancel
Done

40.

41. Click done.

creating a flow on hp laptop

1.

1. Go to flow page

2. Beside hp there is a symbol ‘+’ click on that.

3. Again select decision

4. Enter the Details Label: HP core selection, API name: Gets Automatically Generated.

5. select the Outcome Details Label: hp core i3, Outcome API name: Gets Automatically Generated.

- Resource: Select Record. core type.
- Operator: Select Equals.
- Value: Select hp i3.

9. Similarly create outcomes for hp core i5 and hp core i7 also.

10. Go to flow page

11. Beside hp there is a symbol ‘+’ click on that.
 12. Again select decision
-
13. Enter the Details Label: hp core i5 month selection, API name: Gets Automatically Generated.
 14. Enter the Outcome Details Label: hp 1(i5), Outcome API name: Gets Automatically Generated.
 15.
 - Resource: Select Record. how many months.
 - Operator: Select Equals.
 - Value: 1.
-
19. Enter the Outcome Details Label: hp 2(i5), Outcome API name: Gets Automatically Generated.
 - Resource: Select Record. how many months.
 - Operator: Select Equals.
 - Value: Select 2.
 23. Click ‘+’ outcome details
 24. Enter the Outcome Details Label: hp 3(i5), Outcome API name: Gets Automatically Generated.
 - Resource: Select Record. how many months.
 - Operator: Select Equals.
 - Value: Select 3.
 28. Click ‘+’ outcome details
 29. Enter the Outcome Details Label: hp 4(i5), Outcome API name: Gets Automatically Generated.
 - Resource: Select Record. how many months.
 - Operator: Select Equals.
 - Value: Select 4.
 33. Click ‘+’ outcome details
 34. Enter the Outcome Details Label: hp 5(i5), Outcome API name: Gets Automatically Generated.

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

Edit Decision

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	OUTCOME DETAILS
hp core i3 hp core i5 hp core i7 Default Outcome	<div style="display: flex; justify-content: space-between;"> * Label * Outcome API Name </div> <div style="display: flex; justify-content: space-between;"> <input style="width: 45%; border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="text" value="hp core i5"/> <input style="width: 45%; border: 1px solid #ccc; padding: 2px;" type="text" value="hp_core_i5"/> </div> <div style="margin-top: 10px;"> Condition Requirements to Execute Outcome <div style="border: 1px solid #ccc; padding: 2px; width: fit-content; margin-bottom: 5px;">All Conditions Are Met (AND)</div> <div style="display: flex; justify-content: space-between; align-items: center;"> Resource <div style="display: flex; align-items: center;"> <input style="width: 15%; border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="text" value="\$Record > core type"/> Operator <div style="display: flex; align-items: center;"> Equals Value <input style="width: 15%; border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="text" value="core i5"/> Delete </div> </div> </div> <div style="margin-top: 10px;"> + Add Condition </div> <div style="margin-top: 10px;"> When to Execute Outcome <div style="display: flex; justify-content: space-between; align-items: center;"> If the condition requirements are met <input checked="" type="radio"/> Only if the record that triggered the flow to run is updated to meet the condition requirements <input type="radio"/> </div> </div> </div>
Cancel Done	

Click on done.

38. After hp 1(i5) there is '+' symbol like hp 2(i5), hp 3(i5), hp 4(i5), hp 5(i5).
39. Click on '+' then select update records
40. Enter the Details Label: one month of hp i5 rate, API name: Gets Automatically Generated.
41. Field: Amount__c, value: for hp 1(i5)-1700, hp 2(i5)-3400, hp 3(i5)-5100, hp 4(i5)-6800, hp 5(i5)-8500. Follow for all these finally
42. **Follow the steps from 8 to 22 and create decision and Update record elements for Hp core i3 and Hp core i7. But enter different Amount values.**

Edit Update Records

one month of hp i5 rate (one_month_of_hp_i5_rate) 

* How to Find Records to Update and Set Their Values

- Use the laptop bookings record that triggered the flow
- Update records related to the laptop bookings record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record

None—Always Update Record

Set Field Values for the Laptop Bookings Record

Field	Value	Actions
Amount_c	1700	

[+ Add Field](#)

[Cancel](#) [Done](#)

Click done.

creating a flow on mac laptop

1. Go to flow page
2. Beside mac there is a symbol ‘+’ click on that.
3. Again select decision
4. Enter the Details Label: mac should be Updated, API name: Gets Automatically Generated.
5. select the Outcome Details Label: mac laptop, Outcome API name: Gets Automatically Generated.
 - Resource: Select Record. core type.
 - Operator: Select Equals.
 - Value: Select Bionic Chip.

Edit Decision

* Label	* API Name										
mac field should be updated	mac_field_should_be_updated										
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"> <tr> <td>OUTCOME ORDER ① +</td> <td>OUTCOME DETAILS</td> </tr> <tr> <td>mac laptop</td> <td>* Label mac laptop * Outcome API Name mac_laptop</td> </tr> <tr> <td>Default Outcome</td> <td>Condition Requirements to Execute Outcome All Conditions Are Met (AND)</td> </tr> <tr> <td></td> <td>Resource \$Record > core type Equals Value Bionic chip</td> </tr> <tr> <td></td> <td>+ Add Condition</td> </tr> </table>		OUTCOME ORDER ① +	OUTCOME DETAILS	mac laptop	* Label mac laptop * Outcome API Name mac_laptop	Default Outcome	Condition Requirements to Execute Outcome All Conditions Are Met (AND)		Resource \$Record > core type Equals Value Bionic chip		+ Add Condition
OUTCOME ORDER ① +	OUTCOME DETAILS										
mac laptop	* Label mac laptop * Outcome API Name mac_laptop										
Default Outcome	Condition Requirements to Execute Outcome All Conditions Are Met (AND)										
	Resource \$Record > core type Equals Value Bionic chip										
	+ Add Condition										
<input type="button" value="Cancel"/> <input type="button" value="Done"/>											

Click done.

9. Go to flow page

10. Beside Mac there is a symbol '+' click on that.

11. Again select decision

12. Enter the Details Label: Mac months selected, API name: Gets Automatically Generated.

13. Enter the Outcome Details Label: mac bionic chip (1), Outcome API name: Gets Automatically Generated.

14.

- Resource: Select Record. how many months.
- Operator: Select Equals.
- Value: 1.

18. Enter the Outcome Details Label: mac bionic chip (2), Outcome API name: Gets Automatically Generated.

- Resource: Select Record. how many months.
- Operator: Select Equals.
- Value: Select 2.

22. Click ‘+’ outcome details

23. Enter the Outcome Details Label: mac bionic chip (3), Outcome API name: Gets Automatically Generated.

- Resource: Select Record. how many months.
- Operator: Select Equals.
- Value: Select 3.

27. Click ‘+’ outcome details

28. Enter the Outcome Details Label: mac bionic chip (4), Outcome API name: Gets Automatically Generated.

- Resource: Select Record. how many months.
- Operator: Select Equals.
- Value: Select 4.

32. Click ‘+’ outcome details

33. Enter the Outcome Details Label: mac bionic chip (5), Outcome API name: Gets Automatically Generated.

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

Edit Decision

mac months selected (mac_months_selected)

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	OUTCOME DETAILS	Condition Requirements to Execute Outcome	Action
mac bionic chip(1)	*Label: mac bionic chip(1) *Outcome API Name: mac_bionic_chip_1	All Conditions Are Met (AND)	
mac bionic chip(2)			
mac bionic chip(3)			
mac bionic chip(4)			
mac bionic chip(5)			
Default Outcome			

Resource: \$Record > how many months Operator: Equals Value: 1

+ Add Condition

When to Execute Outcome

If the condition requirements are met
 Only if the record that triggered the flow to run is updated to meet the condition requirements

Click done.

37. After mac bionic chip (1) there is ‘+’ symbol like mac bionic chip (2), mac bionic chip (3), mac bionic chip (4), mac bionic chip (5).

38. Click on ‘+’ then select update records

39. Enter the Details Label: one month of mac rate, API name: Gets Automatically Generated.

40. Field: Amount_c, value: for one month of mac bionic chip rate-1700, two month of mac bionic chip rate-3400, three month of mac bionic chip rate-5100, four month of mac bionic chip rate-6800, five mouth of mac bionic chip rate-8500. Follow for all these finally

Edit Update Records

The screenshot shows the 'Edit Update Records' interface. It includes sections for 'How to Find Records to Update and Set Their Values', 'Set Filter Conditions', and 'Set Field Values for the Laptop Bookings Record'. The 'Field' column contains 'Amount_c' and the 'Value' column contains '2000'. Buttons for 'Cancel' and 'Done' are at the bottom right.

* How to Find Records to Update and Set Their Values

- Use the laptop bookings record that triggered the flow
- Update records related to the laptop bookings record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record

None—Always Update Record ▾

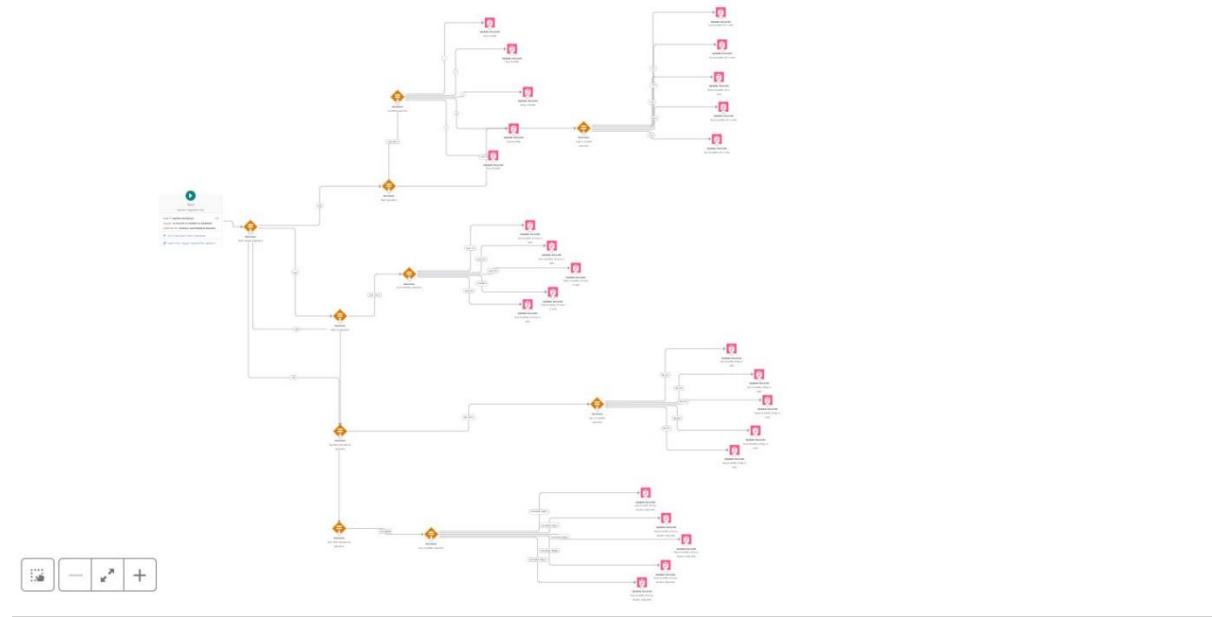
Set Field Values for the Laptop Bookings Record

Field	Value
Amount_c	2000

Cancel Done

Click done.

FLOW:



Click on save.

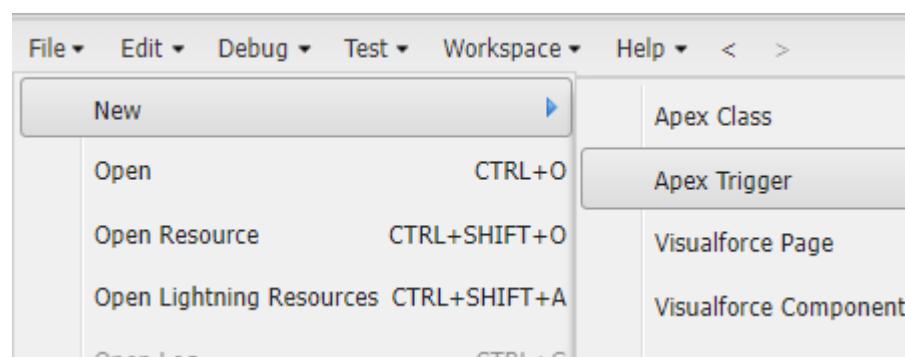
Label: Laptop distributions, API name: automatically filled

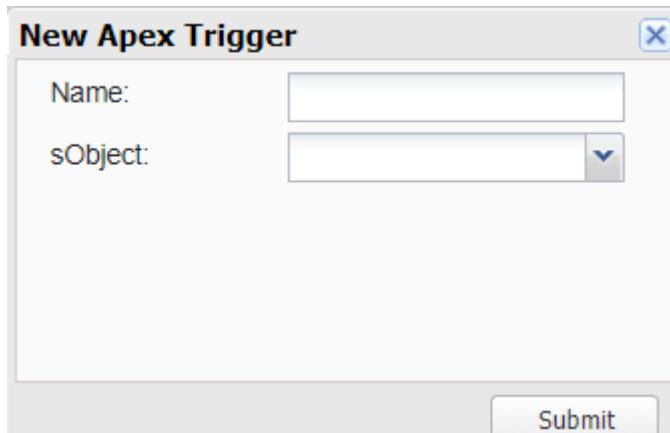
Save the flow and activate it.

Apex Trigger and Handler Class

How to create a new trigger:

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on the File menu in the toolbar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.





Syntax For creating trigger:

The syntax for creating trigger is:

Trigger [trigger name] on [object name] (Before/After event)

```
{  
}
```

```
trigger LaptopBooking on Laptop_Bookings__c (After insert,after update) {  
    if(trigger.isAfter && ( trigger.isInsert || trigger.isupdate))  
    {  
        LaptopBookingHandler.sendEmailNotification(trigger.new);  
    }  
}
```

Trigger code:

```

trigger Laptop Booking on Laptop_ Bookings__c (After insert, after update) {

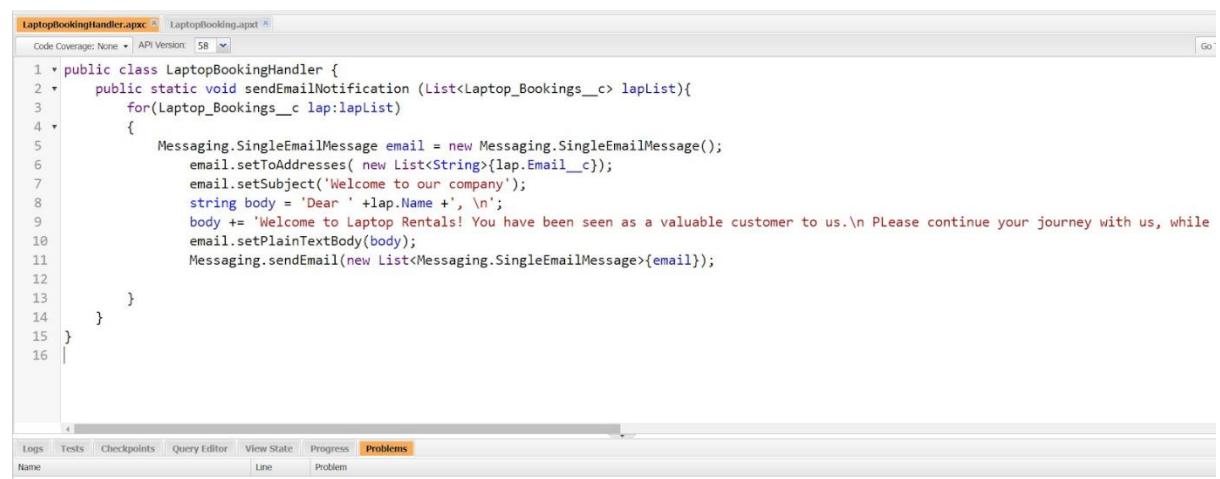
    if(trigger.is After && (trigger. Is Insert || trigger. Is update))
    {
        Laptop Booking Handler. Send Email Notification (trigger. new);
    }
}

```

Note: copy the API names

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

Handler Class:



```

LaptopBookingHandler.apxc  LaptopBooking.apxt
Code Coverage: None  API Version: 58  Go...
1 * public class LaptopBookingHandler {
2     public static void sendEmailNotification (List<Laptop_Bookings__c> laplist){
3         for(Laptop_Bookings__c lap:laplist)
4         {
5             Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
6             email.setToAddresses( new List<String>{lap.Email__c});
7             email.setSubject('Welcome to our company');
8             string body = 'Dear ' +lap.Name +', \n';
9             body += 'Welcome to Laptop Rentals! You have been seen as a valuable customer to us.\n Please continue your journey with us, while
10             email.setPlainTextBody(body);
11             Messaging.sendEmail(new List<Messaging.SingleEmailMessage>(email));
12         }
13     }
14 }
15
16
Logs  Tests  Checkpoints  Query Editor  View State  Progress  Problems
Name  Line  Problem

```

Code Snippet:

```
public class Laptop Booking Handler {  
    public static void Send Email Notification (List<Laptop_ Bookings__ c> lap List) {  
        for (Laptop_ Bookings__ c lap: lap List)  
        {  
            Messaging. Single Email Message email = new Messaging. Single Email Message  
            ();  
            email. set To Addresses (new List<String> {lap. Email__ c});  
            email. Set Subject ('Welcome to our company');  
            string body = 'Dear Customer, \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_ name __ c;  
            email. Set Plain Text Body (body);  
            Messaging. Send Email (new List<Messaging. Single Email Message>{email});  
        }  
    }  
}
```

Note:

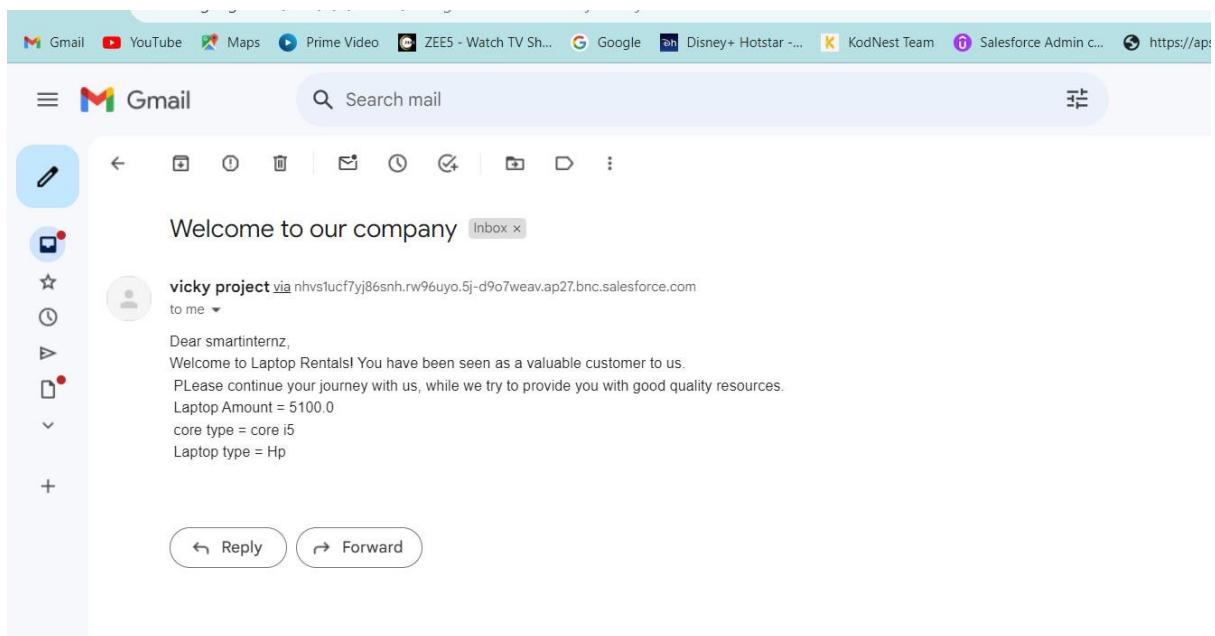
- 1.Class name: Laptop Booking Handler
- 2.API Name: Laptop _ Bookings __ c(as per your org go to laptop booking object and copy from that).

3.core__c (as per your org go to laptop booking object and copy from that).

4.Laptop_type__c. (as per your org go to laptop booking object and copy from that).

In this project, trigger is called whenever the particular record's sum exceeds the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

Result:



Create Report

1. **Note: Before creating reports just fill the 10-12 records in the Laptop Bookings object.**

Create records for each one you have to create at least 2 different records i.e dell(i3), dell(i7), acer (i3), hp (i5), mac (bionic chip).

1.

1. Go to the app? click on the reports tab
2. Click New Report.

3. Select report type from category or from report type panel or from search panel “consumer with Laptop Bookings and total laptops”? click on start report.

4. Create a simple tabular report

5. Add fields from left pane, make sure that Amount field will be selected.

6. Click the Amount column drop down and select bucket list.

Edit Bucket Column

* Field * Bucket Name

	Range	Bucket	
<input type="button" value="Add ►"/>	<= <input type="text" value="900"/>	basic	<input type="button" value="X"/>
<input type="button" value="Add ►"/>	> 900 to <input type="text" value="1500"/>	intermediate	<input type="button" value="X"/>
<input type="button" value="Add ►"/>	> 1,500 to <input type="text" value="10000"/>	high	<input type="button" value="X"/>
<input type="button" value="Add ►"/>	> <input type="text" value="10,000"/>	very high	<input type="button" value="X"/>

Treat empty Amount values in the report as zeros.

Click apply it.

8. Select Types of version in Group By Rows to create a **summary report**. Follow the image for other fields.

The screenshot shows a Business Central report interface. The top navigation bar includes 'LAPTOP RENTALS', 'Total Laptops', 'consumer', 'Laptop Bookings', 'Billing Process', 'Reports', 'Dashboards', and various system icons. Below the navigation is a 'REPORT' section with a 'Laptop Analytics' card. The main area displays a report titled 'Total Laptops with Laptop Bookings and Consumer'. On the left, a 'Fields' sidebar contains sections for 'Outline', 'Filters', 'Groups', 'Columns', and specific fields like 'Laptop Bookings', 'Consumer', '# Amount', and '# Laptops Available'. The main grid shows data for laptop bookings, grouped by consumer name and amount, with subtotals and grand totals. A red box highlights the 'Types of Versions' filter in the sidebar.

	Laptop Bookings: Laptop Bookings	Consumer: consumer Name	Amount	Laptops Available	Total Laptops: Total Laptops
Basic (1)	Dell i3	Shruthi	₹1,000	48	Dell core i3 50
Subtotal			₹1,000	48	
Intermediate (1)	Acer	Swetha	₹1,500	46	50
Subtotal			₹1,500	46	
High (4)	Acer	Swetha	₹4,800	46	50
	Acer	Swetha	₹3,800	46	50
	Acer	Swetha	₹3,800	46	50
	Dell i3 Booking	Shruthi	₹5,000	48	Dell core i3 50
Subtotal			₹17,400	94	
Total (6)			₹19,900	94	

Click on Save & run it.

Report: Total Laptops with Laptop Bookings and Consumer Laptop Analytics					
Total Records	Total Amount	Total Laptops Available			
6	₹19,900	94			
<input type="checkbox"/> Types of Versions	Laptop Bookings: Laptop Bookings	Consumer: consumer Name	Amount	Laptops Available	Total Laptops: Total Laptops
<input type="checkbox"/> Basic (1)	Dell i3	Shruthi	₹1,000	48	Dell core i3 50
Subtotal			₹1,000	48	
<input type="checkbox"/> Intermediate (1)	Acer	Swetha	₹1,500	46	50
Subtotal			₹1,500	46	
<input type="checkbox"/> High (4)	Acer	Swetha	₹4,800	46	50
	Acer	Swetha	₹3,800	46	50
	Acer	Swetha	₹3,800	46	50
	Dell i3 Booking	Shruthi	₹5,000	48	Dell core i3 50
Subtotal			₹17,400	94	
Total (6)			₹19,900	94	

Row Counts Detail Rows Subtotals Grand Total

Sharing report to owner

1.

1. Click edit drop down and select subscribe option

Report: Total Laptops with Laptop Bookings and Consumer Laptop Analytics					
Total Records	Total Amount	Total Laptops Available			
6	₹19,900	94			
<input type="checkbox"/> Types of Versions	Laptop Bookings: Laptop Bookings	Consumer: consumer Name	Amount	Laptops Available	Total Laptops: Total Laptops
<input type="checkbox"/> Basic (1)	Dell i3	Shruthi	₹1,000	48	Dell core i3 50
Subtotal			₹1,000	48	
<input type="checkbox"/> Intermediate (1)	Acer	Swetha	₹1,500	46	50
Subtotal			₹1,500	46	
<input type="checkbox"/> High (4)	Acer	Swetha	₹4,800	46	50
	Acer	Swetha	₹3,800	46	50
	Acer	Swetha	₹3,800	46	50
	Dell i3 Booking	Shruthi	₹5,000	48	Dell core i3 50
Subtotal			₹17,400	94	
Total (6)			₹19,900	94	

Row Counts Detail Rows Subtotals Grand Total

Save As
Save
Subscribe
Export
Delete
Add to Dashboard

2. Follow as per below image.

Edit Subscription

Settings

Frequency

Daily Weekly Monthly

Time
8:00 am ▾

Attachment
 Attach File

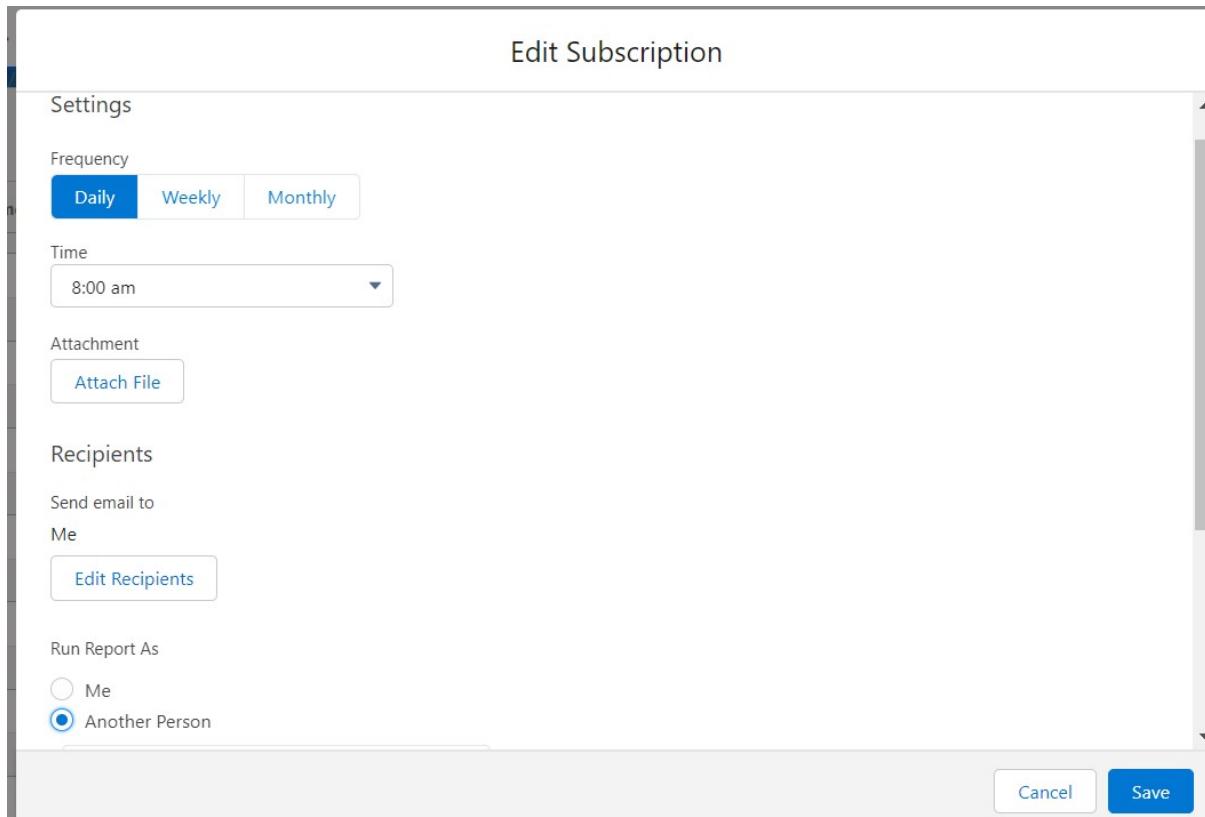
Recipients

Send email to
Me
 Edit Recipients

Run Report As

Me
 Another Person

Cancel Save

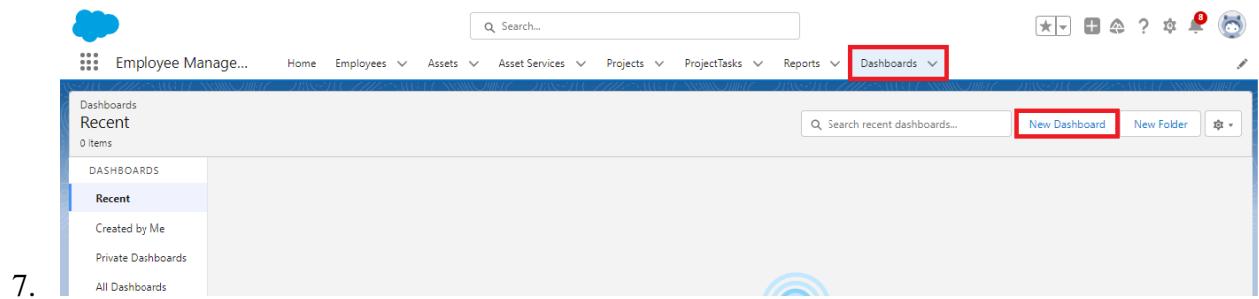


3. After selecting the run report as a “another person” select your personal account or whom you want to send that mail to.
4. Click save.

NOTE: The owner gets daily email notification of that laptop booking report.so that he can see all data remotely.

Create Dashboard Folder

1. Click on the app launcher and search for the dashboard.
2. Click on the dashboard tab.
3. Click the new folder, give the folder label as “total rent amount”.
4. Folder unique names will be auto populated.
5. Click save. **Create Dashboard**
6. Go to the app >> click on the Dashboards tabs.



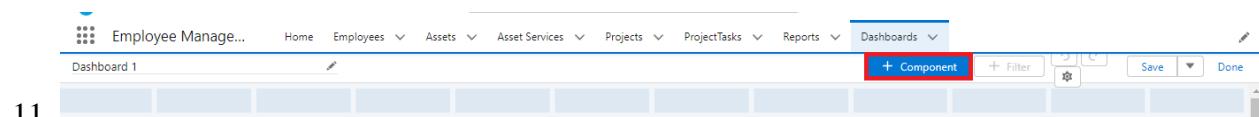
7.

8. Give a Name and select the folder that was created, and click on create.

The dialog has a title 'New Dashboard'. It contains fields for 'Name' (with placeholder 'data analytics of laptops'), 'Description' (with placeholder 'total amount of data in dashboards'), and 'Folder' (with placeholder 'total rents amount' and a 'Select Folder' button). At the bottom are 'Cancel' and 'Create' buttons.

9.

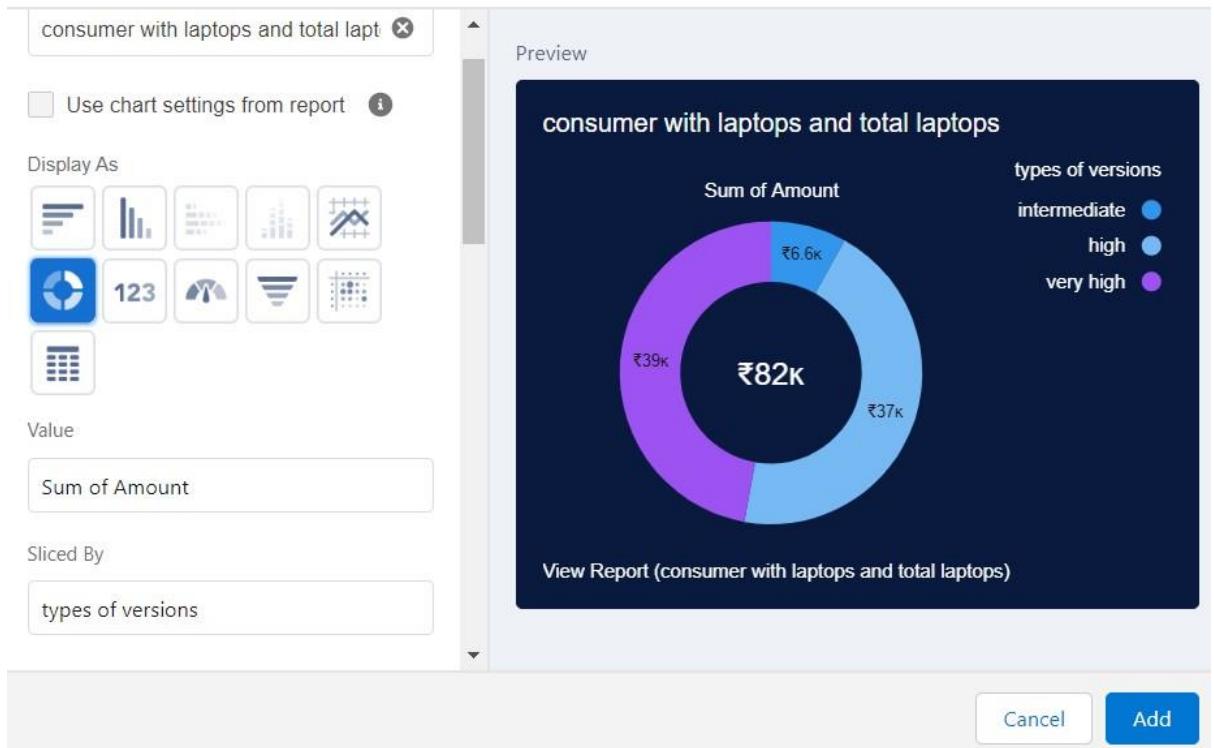
10. Select add component.



11.

12. Select a Report and click on select.

Add Component



13. Select the dark component and add to the dashboards.

14. Save it.

15. Click done.

Troubleshooting Approach:

To troubleshooting issues in a CRM application for laptop rentals, the first step is to clearly understand the problem by gathering information from the user and checking for any error messages or unusual behaviour. Next, replace the issue in a test environment to confirm its cause. Check user permissions to ensure they have the necessary access to objects, fields, and records. Review validation rules, automation like Flows or Process Builders, and custom code to identify any logic errors or conflicts. It's also important to verify the integrity of the data involved, ensuring all required fields are properly filled. Examine recent changes or departments that might have introduced the issue. Debug logs and audit trails can provide deeper insights into what went wrong. If integrations are involved, ensure API connections are functioning as expected. Additionally, browser or device-related issues can be resolved by clearing the cache or trying another device. If the issues persist after these steps, contacting Salesforce support may be necessary for further assistance.

Conclusion:

In conclusion, the CRM application for laptop rentals plays a vital role in streamlining rental operations, improving customer service, and managing inventory efficiently. It

centralizes customer interactions, tracks laptop availability, and automates rental processes, making the entire workflow smoother and more reliable. By providing real-time access to data and analytics, it empowers the business to make informed decisions and respond quickly to customer needs. Overall, the application enhances operational efficiency, ensures accurate record-keeping and supports business growth through better customer relationship management.