

A CRM APPLICATION FOR LAPTOP RENTALS

CRM Application on Laptop rentals is about delivering the items to the customers by rental purpose. It leverages the power of customer relationship management (CRM) to enhance customer experiences, optimize store operations, and improve overall efficiency. Additionally to these, we also need to do an effective CRM i.e via communicating through email with the potential customers identified.

Below is a breakdown down how the CRM system functions in the context of laptop rentals, and how it can improve both the rental process and customer relationship management (CRM).

1. Customer Relationship Management (CRM) Basics

CRM stands for Customer Relationship Management, and its goal is to understand and anticipate customer needs. It uses data analytics about customers' history with the company to improve business relationships, assist in customer retention, and drive sales growth. In the context of a laptop rental business, CRM can be leveraged in several key areas:

Customer Interaction Tracking: Track and manage every interaction a customer has with the company. This includes inquiries, complaints, orders, payments, and feedback.

Personalized Communication: Based on customer history, send tailored offers, reminders, promotions, and updates. This can be done through email, SMS, or app notifications.

Sales and Marketing: Identify high-value customers, leads, and untapped markets, allowing for targeted marketing campaigns.

Customer Segmentation: Group customers based on various characteristics, such as rental frequency, type of laptop rented, or geographical location, to deliver more relevant and personalized offers.

2. Key Features of a Laptop Rental CRM Application

a. Customer Profiles

Data Collection: Store and manage detailed profiles of customers, including personal details (name, contact information), rental history, payment methods, preferences, and feedback.

Customer Segmentation: Organize customers into groups (e.g., business customers, students, occasional renters) to provide tailored services and communication.

Behaviour Tracking: Track when customers last rented, what models they rented, their frequency of rentals, etc. This data can be used to personalize offers and ensure timely communication.

b. Rental Management

Inventory Management: Track available laptops in real-time, noting which ones are available for rent, in use, or under maintenance.

Rental Duration & Scheduling: Customers can select rental periods (daily, weekly, or monthly), and the system ensures proper billing based on the duration.

Availability Alerts: Customers receive automatic notifications if a laptop they're interested in is available again after being rented out.

c. Order and Payment Management

Order Creation: Customers can book laptops through the application by selecting models, rental period, and other preferences (e.g., accessories).

Invoicing & Payments: Automatically generate invoices based on rental duration, model, and any additional services. The system should support multiple payment methods, including credit/debit cards, bank transfers, and digital wallets.

Late Fees: Implement automatic late fee calculations if customers return laptops past the rental period.

d. Communication & Engagement

Email Automation: Send personalized emails at various stages of the rental process: order confirmation, rental reminders, maintenance updates, return reminders, etc.

Customer Support: Integrated help desk for handling queries or issues related to rentals, laptop issues, or payments.

Feedback Collection: After a rental period, ask customers for feedback to improve services. Automated surveys or ratings can be sent via email or the app.

e. Reporting and Analytics

Sales Analytics: Track rental volumes, popular models, customer demographics, and rental trends. This helps in inventory management and decision-making for future acquisitions.

Customer Lifetime Value (CLV): Calculate the long-term value of each customer based on repeat rentals and their engagement with special offers or marketing campaigns.

Revenue and Expense Tracking: Maintain detailed financial records, such as income from rentals, payment processing fees, and laptop maintenance costs.

3. CRM for Effective Communication with Customers

A significant part of CRM in a laptop rental application is the ability to effectively communicate with customers, especially potential ones who may not have rented yet. Here's how the system can enhance communication and engagement:

a. Automated Email Campaigns

New Customer Onboarding: When a new customer registers or shows interest in renting a laptop, send a welcome email explaining the rental process, policies, and special offers for first-time renters.

Personalized Rental Reminders: Send emails or notifications when it's time to return rented laptops or renew the rental period, helping customers avoid late fees.

Promotional Offers: Target customers based on their rental history with personalized email promotions or discounts (e.g., "20% off your next rental if you rent for 7 days or more").

Re-engagement Campaigns: For customers who haven't rented in a while, send re-engagement

emails offering them a special discount or reminding them of your services.

b. Customer Support via Email

Issue Resolution: Use the CRM system to automatically generate support tickets when customers email regarding issues with their rentals, such as laptop malfunctions or damage. This ensures the issue is tracked and resolved promptly.

Post-Rental Communication: After a rental period ends, you can ask for feedback via email to gauge customer satisfaction and improve your services.

c. Segmentation for Targeted Marketing

By analysing customer behaviour data, CRM tools allow you to segment customers into various categories (e.g., frequent renters, long-term renters, seasonal renters). Based on this segmentation, you can craft highly-targeted marketing campaigns:

Frequent Renters: Send loyalty discounts, offers for long-term rentals, or personalized emails about new laptop models in stock.

First-Time Renters: Offer a discount or incentive for their second rental to build repeat business.

Seasonal Renters: For instance, students renting during exam time, or business professionals during travel periods. Send emails with promotions tailored to their rental needs.

4. Enhancing Customer Experience

To foster better relationships and increase customer loyalty, the application should focus on customer experience. Here are ways CRM can enhance it:

Easy Booking Process: Make it simple and quick for customers to reserve laptops, view available models, and choose their rental period.

Real-Time Updates: Send real-time notifications or emails regarding the status of the rental (e.g., dispatch confirmation, expected delivery time).

Loyalty Programs: Integrate loyalty programs, rewarding customers with discounts, free accessories, or extended rental periods based on their repeat business.

Proactive Maintenance: Keep customers informed about any maintenance needs on rented laptops. If a laptop is unavailable for any reason, offer an alternative or compensation.

5. Technical Integration

A well-integrated CRM system will connect with other business systems such as:

Inventory Management Software: Sync rental data with inventory, so you always know which laptops are in use and which are available for rent.

Payment Gateways: Seamlessly integrate with payment platforms like PayPal, Stripe, or bank accounts to handle payments efficiently.

Customer Support Systems: Integrate with help desk tools like Zendesk or Freshdesk to ensure that customer issues are logged, tracked, and resolved.

Steps to Create a Salesforce Developer Account:

To create a Salesforce Developer Account by signing up for a Salesforce Developer Organization

(org), follow the steps outlined below:

Go to the Signup Page: On the sign-up page, you'll need to fill in the following details:

First Name: Enter your first name.

Last Name: Enter your last name.

Email: Enter your email address (this email will be used for notifications related to your Salesforce Developer org).

Role: Select Developer from the drop-down list.

Company: Enter your college name (as you are using it for academic or learning purposes).

Country: Choose India from the drop-down list.

Postal Code: Enter the postal code (PIN code) of your location in India.

Username: This should be a unique Salesforce username. It does not need to be a real email address,

but it must be in the following format:

username@organization.com

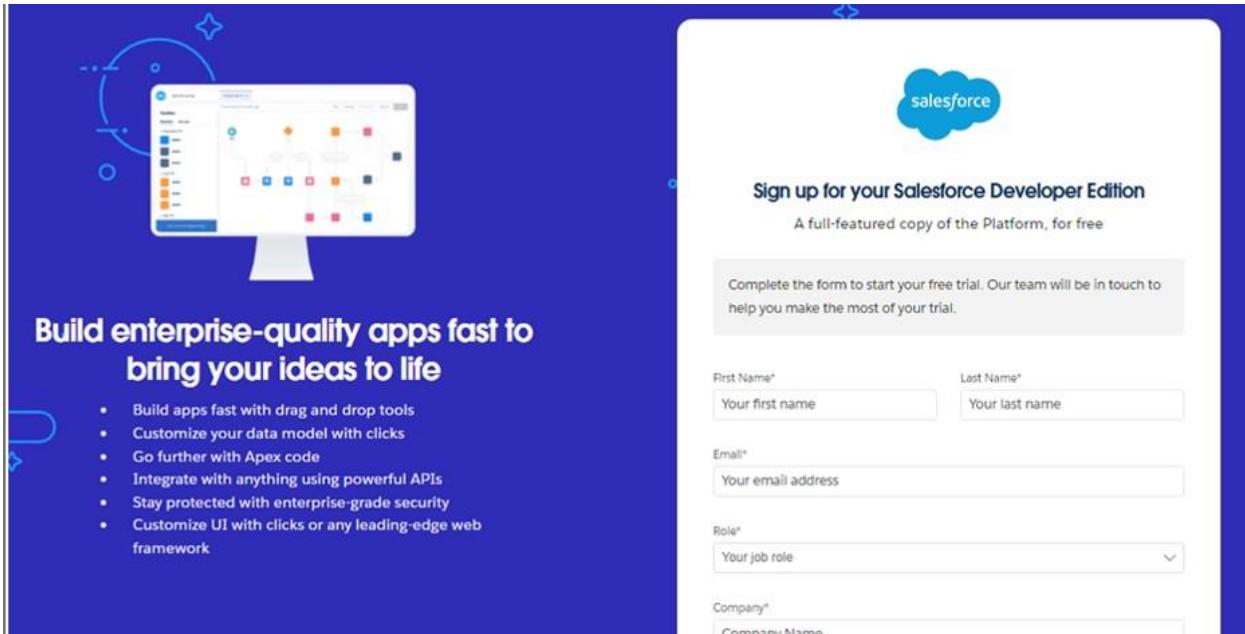
For example: johnsmith@mycollege.com. The username must be unique across Salesforce, so ensure you use a combination of your name and college name or any unique identifier.

Agree to the Terms: After filling out all the details, read and agree to Salesforce's terms of service.

Complete the Sign-Up: Once the form is complete, click on Sign Me Up.

Check Your Email: After submitting the form, Salesforce will send you a verification email to the address you provided. Follow the instructions in the email to verify your account.

Access Your Developer Organization: After verifying your email, you can log in to your newly created Salesforce Developer Organization using the username and password you set during registration.



Object Creation

Create Total Laptops Object

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

Click on Allow reports,Allow search, and Track Field History,

Allow search >> Save.

6 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
Created By	CreatedById	Lookup(User)	
Laptops Available	Laptops_Available__c	Formula (Number)	
Laptops delivered	Laptops_delivered__c	Roll-Up Summary (COUNT Laptop Bookings)	
Last Modified By	LastModifiedById	Lookup(User)	
Owner	OwnerId	Lookup(User,Group)	
Total Laptops	Name	Text(80)	

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

Click on Allow reports,Allow search, and Track Field History,
Allow search >> Save.

SETUP > OBJECT MANAGER
consumer

Details

Fields & Relationships

8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
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	

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

Click on Allow reports,Allow search, and Track Field History,

Allow search >> Save.

The screenshot shows the 'Fields & Relationships' section of the Salesforce Object Manager for the 'Laptop Bookings' object. The sidebar on the left lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main area displays a table of fields:

Fields & Relationships		
11 Items, Sorted by Field Label		
Amount	Amount_c	Currency(18, 0)
core type	core_type_c	Picklist
Created By	CreatedBy	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	LastModifiedBy	Lookup(User)
Name	Name_c	Master-Detail(consumer)

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 ProcessName

Data Type >> Name

Click on Allow reports,Allow search, and Track Field History,

Allow search >> Save.

Fields & Relationships				
7 Items, Sorted by Field Label				
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLER
Page Layouts	Amount	Amount_c	Formula (Number)	
Lightning Record Pages	Billing ProcessName	Name	Text(80)	
Buttons, Links, and Actions	Created By	CreatedById	Lookup(User)	
Compact Layouts	Laptop Bookings	Laptop_Bookings_c	Lookup(Laptop Bookings)	
Field Sets	Last Modified By	LastModifiedById	Lookup(User)	
Object Limits	Name	Name_c	Master-Detail(consumer)	
Record Types	Payment Mode	Pa_c	Picklist	
Related Lookup Filters				
Search Layouts				

Steps to Create a Lightning App in Salesforce:

1. Go to Setup:

Log in to your Salesforce org.

Click the Gear Icon (⚙️) in the top-right corner of the screen and select Setup from the dropdown.

2. Access the App Manager:

In the Quick Find box (on the left side), type "App Manager" and press Enter.

Under the App Setup section, select App Manager from the search results.

3. Create a New Lightning App:

In the App Manager, click the New Lightning App button in the top-right corner.

4. Enter App Details:

App Name: In the App Details section, enter the name of your app as LAPTOP RENTALS.

Developer Name: This is auto-filled based on your app name. You can leave it as default.

Click Next.

5. App Options Page:

In the App Options section, leave everything as default. You can keep the default settings for "App Page" and click Next.

6. Utility Items:

In the Utility Items section, leave everything as default as well. This step can be skipped or you can add utility items later.

Click Next.

7. Upload a Photo for Your App:

In the App Branding section, you will be prompted to upload a photo related to your app.

Recommended Image Size: 128 x 128 pixels.

Choose a logo or image that reflects your app's purpose (for example, an image of a laptop).

After uploading the photo, click Next.

8. Add Navigation Items:

In this step, you will add the navigation items that will appear in your app.

Search for Navigation Items:

In the Search Bar, type and search for the following items:

Total Laptops

Consumer

Laptop Booking

Billing Process

For each item, select it from the list and click the right arrow to add it to your app's navigation.

Once all items are added, click Next.

9. Add User Profiles:

In the Assign Profiles section, search for System Administrator in the search bar. This profile will have full access to the app.

Select System Administrator and click the right arrow to add it to the list of profiles with access to the app.

Once the profile is added, click Save & Finish.

10. App Creation Complete:

Your LAPTOP RENTALS Lightning App has now been created! You can click on View to preview the app or start customising the pages, objects, and components further.

You can access the app from the App Launcher (the 9-dot grid in the top left corner of Salesforce) and search for LAPTOP RENTALS.

Creating the field in consumer object

1. To create fields in an object:
2. Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.
3. Now click on “Fields & Relationships” >> New
4. Select Data Type as a “Phone”
5. Click on next
6. Fill the Above as following:
7. Field Label: Phone number
8. Field Name : gets auto generated
9. Click the required option checkbox.
10. Click on Next >> Next >> Save and new.
11. To create another field in an object:

12. Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.
13. Now click on “Fields & Relationships” >> New
14. Select Data type as “Email” and Click on Next
15. Fill the Above as following:
16. Field Label: Email
17. Field Name :It's gets auto generated

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

The screenshot shows the 'New Custom Field' setup page for the 'consumer' object. The 'Field Label' is set to 'Email' and the 'Field Name' is also 'Email'. The 'Required' checkbox is checked. Other settings like 'Unique', 'External ID', and 'Auto add to custom report type' are unchecked. The 'Default Value' is set to 'Some Formula Value'.

To create another field 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:
5. Field Label: Address
6. Field Name : It's gets auto generated
7. Select Required field.
8. Click on Next >> Next >> Save and new.

The screenshot shows the 'Object Manager' interface for the 'consumer' object. On the left, a sidebar lists various setup options like Page Layouts, Lightning Record Pages, etc. The main area is titled 'Step 2. Enter the details' and contains fields for 'Field Label' (Address), 'Field Name' (Address), 'Description', 'Help Text', and 'Required'. A note indicates that 'Required' means 'Always require a value in this field in order to save a record'. Below these are checkboxes for 'Auto add to custom report type' and 'Add this field to existing custom report types that contain this entity'. A 'Default Value' section includes a 'Show Formula Editor' link. At the bottom right are 'Previous', 'Next', and 'Cancel' buttons.

To create another field 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:
5. Field Label: consumer Status
6. Value - Select enter values with each value separated by a new line

This screenshot shows the configuration of a picklist field for the 'consumer' object. The 'Values' section is set to 'Enter values, with each value separated by a new line', and the values listed are 'student', 'employee', and 'others'. Other configuration options include 'Use global picklist value set' (unchecked), 'Use first value as default value' (unchecked), and 'Restrict picklist to the values defined in the value set' (checked). The field is labeled 'consumer status' and has a description. The 'Required' checkbox is checked. At the bottom, there are checkboxes for 'Auto add to custom report type' and 'Add this field to existing custom report types that contain this entity'. Navigation buttons 'Previous', 'Next', and 'Cancel' are also present.

Creating the field in Laptops Bookings object

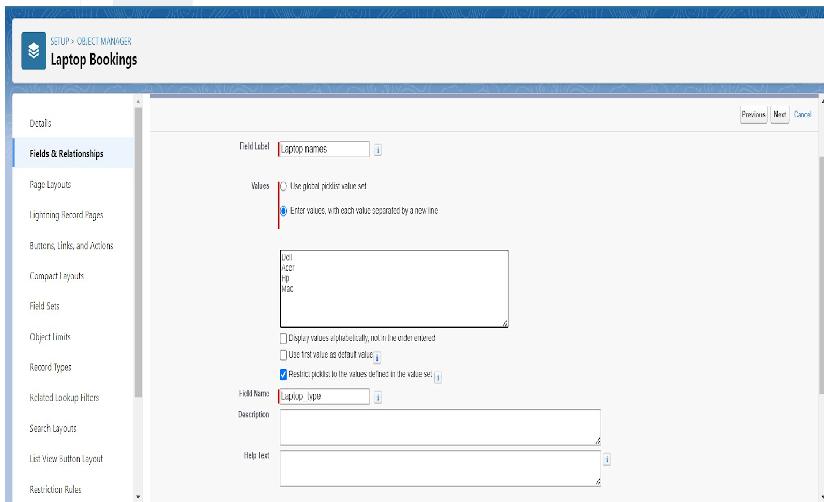
To create fields in an object:

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”

Picklist values are:-1.Dell 2. Acer 3.Hp 4.Mac



Select required

Click on Next >> Next >> Save and new

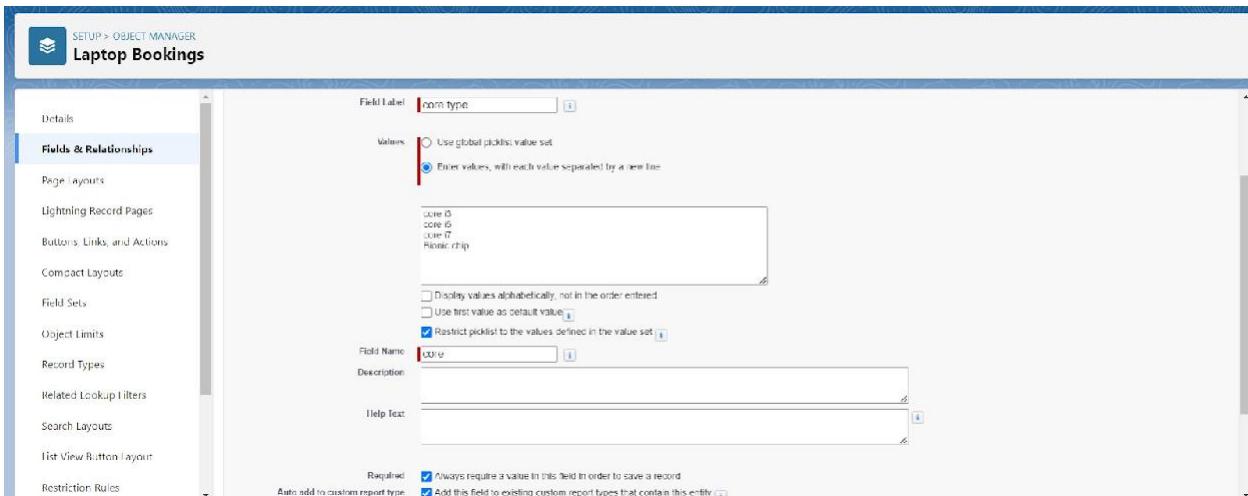
2. To Create a Fields & Relationship to an Laptop Booking Object
To create fields & relationships to an object:

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”

Picklist



Select required

Click on Next >> Next >> Save and new

To Create a Fields & Relationship to an Laptop Booking Object

To create fields & relationships 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. Click the include value for dell-core i3,i5,i7 and for acer i3,i4,i5 and for hip i3,i4,i5 and also for mac bionic chip include the values for it.

The screenshot shows the 'Fields & Relationships' section for the 'Laptop Bookings' object. The 'Controlling Field' is set to 'Laptop names' and the 'Dependent Field' is set to 'core type'. The grid displays the following data:

Laptop names:	Dell	Acer	HP	Mac
core type:	core i3	core i3	core i3	core i3
	core i5	core i5	core i5	core i5
	core i7	core i7	core i7	core i7
	Bionic chip	Bionic chip	Bionic chip	Bionic chip

Click save.

To Create a Fields & Relationship to an Laptop Booking Object

To create fields & relationships to 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 “Lookup Relationship”

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.

To create fields in an object:

1.Go to setup >> click on Object Manager >> type object name(Laptop Booking) in the 2.search bar >> click on the object.

3.Now click on “Fields & Relationships” >> New

4.Select Data Type as a “Currency”

5.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

The screenshot shows the 'Step 2. Enter the details' page for creating a new field. The 'Field Label' is set to 'Amount'. The 'Length' is specified as '18' and 'Decimal Places' as '0'. A note below says: '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 "12345678.00"'.

Below the field settings, there are several checkboxes:

- Always require a value in the field in order to save a record.
- Add this field to existing custom report types. Until contain this entity.

At the bottom, there is a 'Default Value' field containing '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 Select Data Type as a “Lookup Relationship”

Click on Next

The screenshot shows the 'Step 2. Choose the related object' page. The 'Related to' dropdown menu is open, showing 'Total Laptops' as the selected option.

3. 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 & relationships to 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 an “Email”

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	Picklist	Laptop names	
Created By	CreatedById	Lookup(List)		
Laptop Bookings Name	Name	Text(80)		✓
Laptop names	Laptop_type__c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Master-Detail[Customer]		✓
Total no of laptops	Total_no_of_laptops__c	Master-Detail[Total laptops]		✓

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

After Creating the Lookup Relationship Than Only you can create the Rollup Summary

Go to setup >> click on Object Manager >> type object name(Total Laptops) in the search bar >> click on the object.

Now click on “Fields & Relationships” >> New

Step 1. Choose the field type

Specify the type of information that the custom field will contain.

Data Type

- None Selected
- Auto Number
- Formula
- Roll Up Summary
- Lookup Relationship
- Master-Detail Relationship

Selections of the data types below:

- A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.
- A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.
- A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.
- Create a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a pop-up list. The other object is the source of the values in the list.
 - The relationship field is required on all detail records.
 - The ownership and sharing of a detail record are determined by the master record.
 - When a user deletes the master record, all detail records are deleted.
 - You can create rollup summary fields on the master record to summarize the detail records.

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

The screenshot shows the 'Total laptops' object in the Object Manager. A new custom field is being created, specifically 'Laptops delivered'. The 'Field Label' is set to 'Laptops delivered' and the 'Field Name' is also 'Laptops_delivered'. There is no description or help text provided. A note at the bottom indicates that the field can be added to existing custom report types.

Click on Next

Select the Laptop Bookings in the Summarized Object

Select the count Radio button in the select Roll-up Type

The screenshot shows the 'Step 3 of 5' screen for defining the summary calculation. The 'Master Object' is set to 'Total laptops' and the 'Summarized Object' is set to 'Laptop Bookings'. Under 'Select Roll-Up Type', the radio button for 'COUNT' is selected. The 'Field to Aggregate' dropdown is set to 'None'. Below this, there is a 'Filter Criteria' section with two options: 'All records should be included in the calculation' and 'Only records meeting certain criteria should be included in the calculation'. The 'Laptop Bookings' option is highlighted in blue.

8. 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 “Formula” and Click on Next

Fill the Above as following:

Field Label: Laptops Available

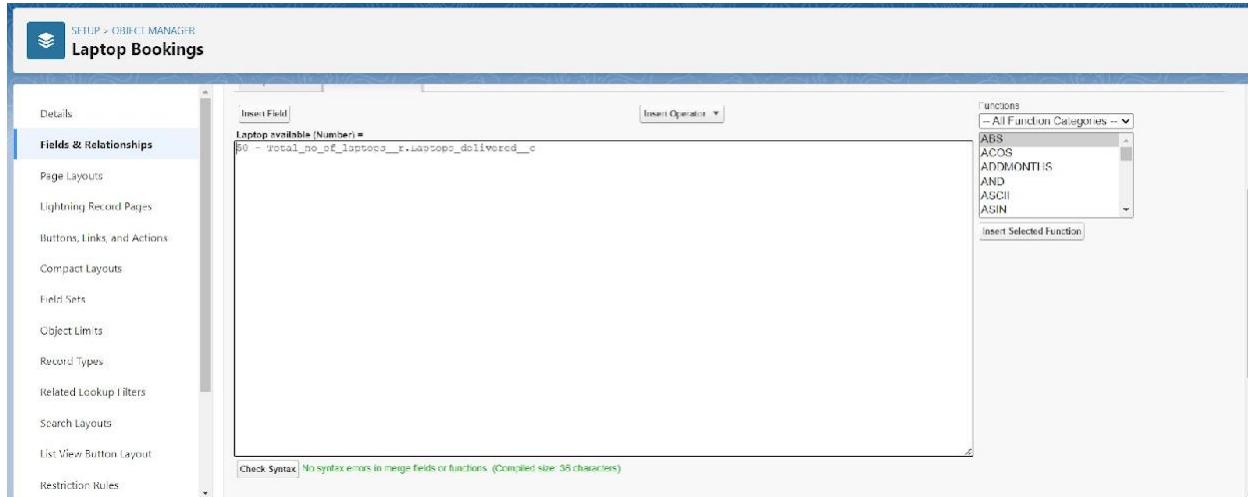
Field Name : It's gets auto generated

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 the formula box “ 50 - ” and Click on the insert field; then 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



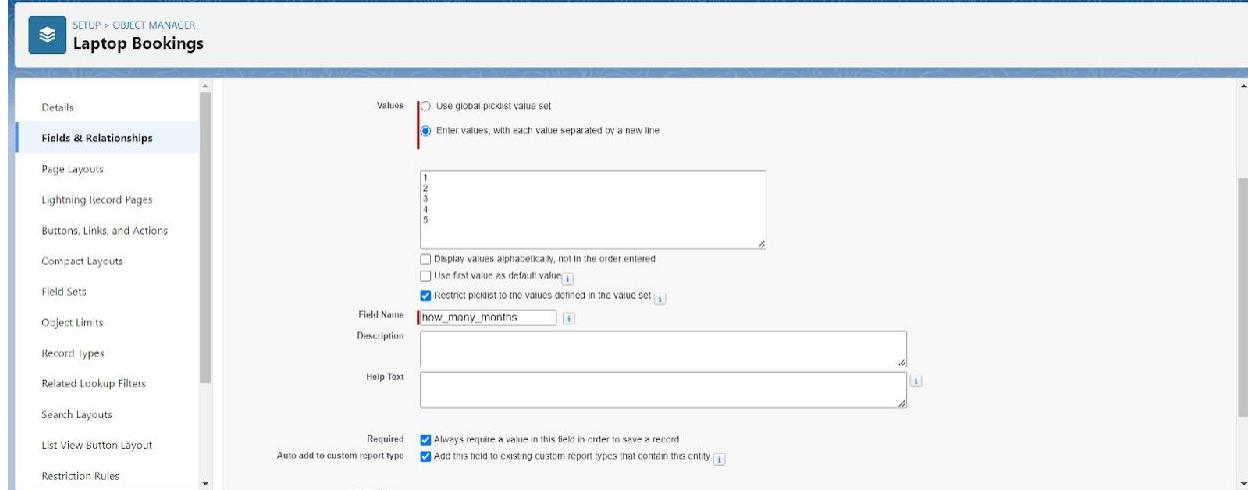
Click on Next >> Next >> Save and new

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”

Picklist values are 1.2.3.4.5

Click and save it.



Creation of Fields & Relationships for Billing Process Object

1. To create fields & relationships to an object:

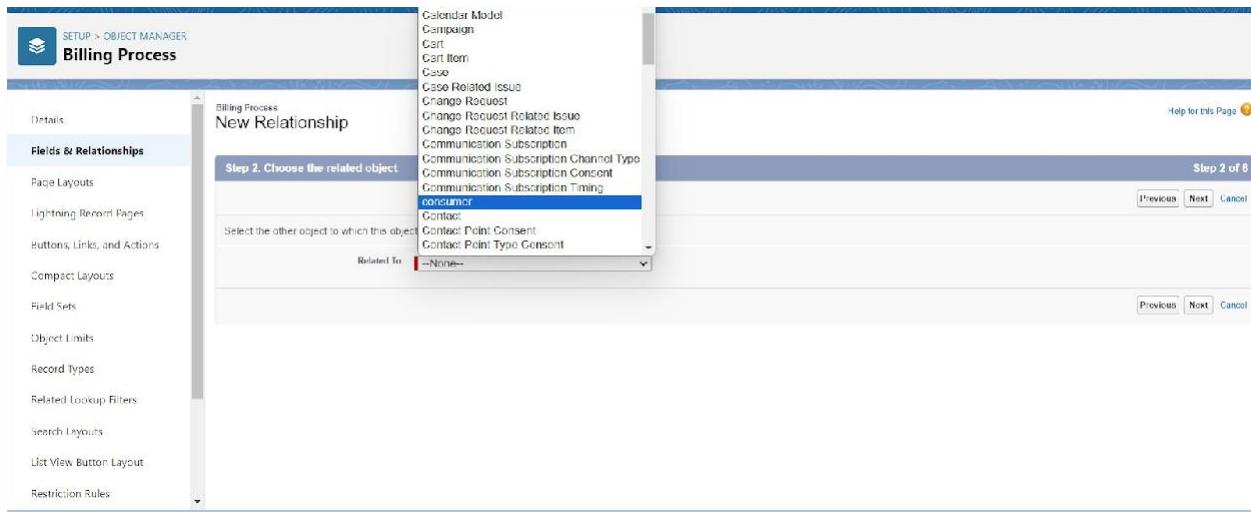
Go to setup >> click on Object Manager >> type object name(Billing Process) in the search bar >> click on the object.

Now click on “Fields & Relationships” >> New

Select Data Type as a “Master-detail Relationship”

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 field & relationship to an object:

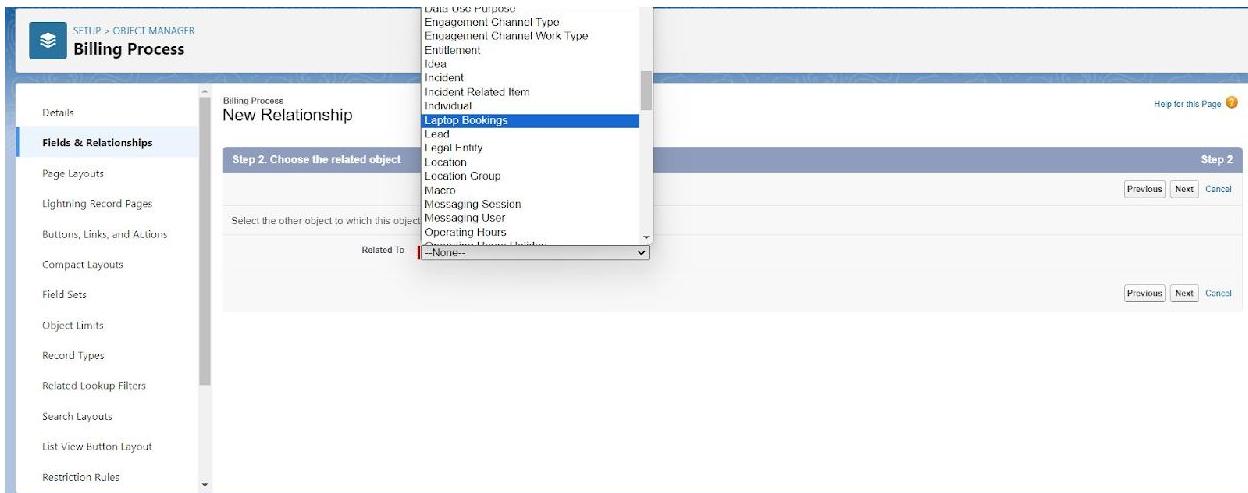
Go to setup >> click on Object Manager >> type object name(Billing Process) in the search bar >> click on the object.

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



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:

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 “Picklist”

Fill the Above as following:

Field Label: Payment Mode

Value >> Select enter values with each value separated by a new line

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.

4. Create a Cross object formula Field in billing process 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 “Formula”

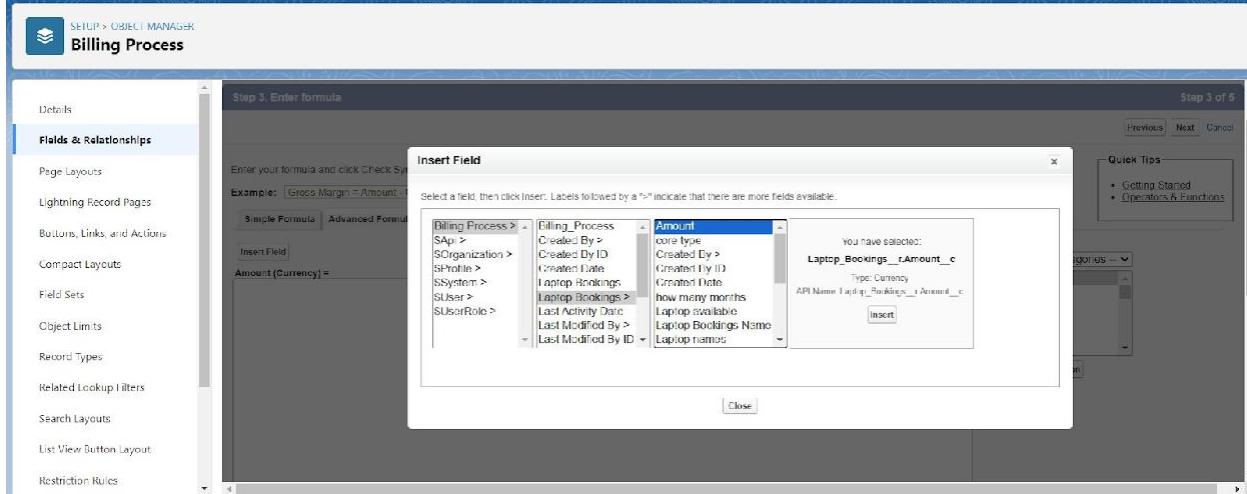
Click on Next

Enter the Field label: Amount, the Field name gets auto generated, and click on Next.(Formula return type Number).

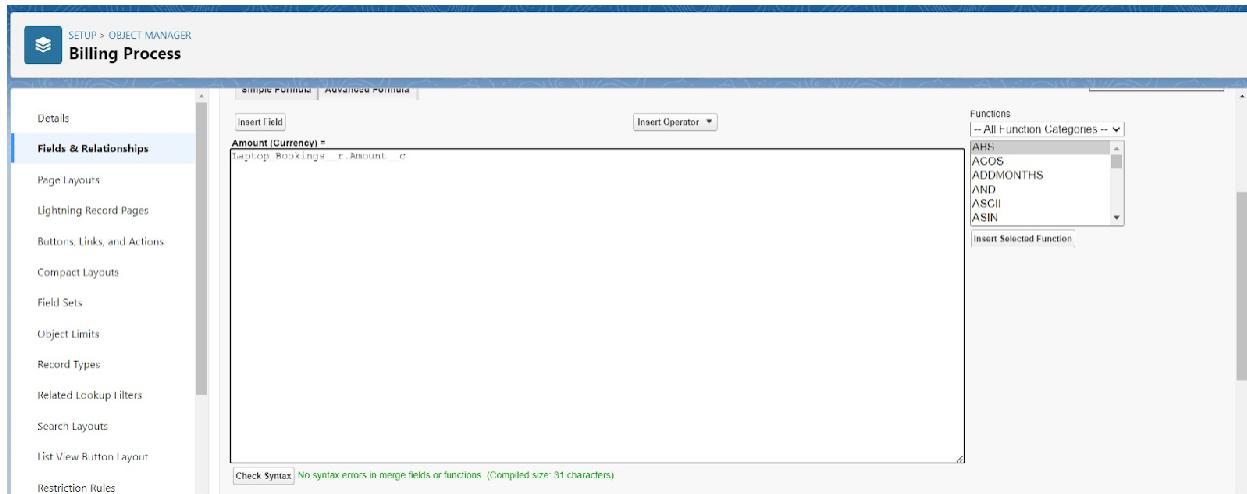
In the Advanced Formula, Click on the Insert field in the popup Screen Select the Billing

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 “Laptop_Booking_r.Amount__c”.

Click on the Check syntax: No syntax errors in merge fields



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



Creating the field in the Total Laptops object

1. To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Total Laptops) in search bar >> click on the object.

Now click on “Fields & Relationships” >> New

Select Data type as a “Formula” and Click on Next

Fill the Above as following:

Field Label: Laptops Available

Field Name : It's gets auto generated

Select the Formula Return Type as “Number”

Field Label: Laptops Available

Auto add to custom report type: Add this field to existing custom report types that contain this entry [E]

Formula Return Type:

- None Selected
- Checkbox
- Currency
- Date
- Date/Time
- Number
- Percent
- Text
- Time

Select one of the data types below.

Number Description: Calculate a numeric value.
Example: [LaptopsDelivered_c] * 2
Calculate a dollar or other currency amount and automatically format the field as a currency amount.
Example: [Gross Margin] = Amount - Cost_LC
Calculate a date, for example, by adding or subtracting days to other dates.
Example: [Remaining Days] = CloseDate - 7
Calculate a datetime, for example, by adding a number of hours or days to another datetime.
Example: [Next : NOWVH] + 1
Calculate a numeric value.
Example: [TaxRate] = 1.0 * Credits_c + .02
Calculate a percent and automatically add the percent sign to the number.
Example: [Discount] = (Amount - Discounted_Amount_C) / Amount
Create a text string, for example, by concatenating other text fields.
Example: [Full Name] = LastName & ', ' & FirstName
Calculate a time, for example, by adding a number of hours to another time.
Example: [Next] = TIMEVAL(IF(NOW)) + 1

Options: Decimal Places: 0 Example: 399

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

Click on the Advanced

Formula “ 50 - Laptops_delivered_c ” and Check Syntax

Simple Formula | Advanced Formula

Insert Field: Laptops Available (Number) =
50 - Laptops_delivered_c

Insert Operator:

Functions: All Function Categories

- ABS
- ACCS
- ADDMONTHS
- AND
- ASCII
- ASIN

Check Syntax: No syntax errors in merge fields or functions. (Compiled size: 35 characters)

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

Validation rule:

Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

Improve the quality of your data using validation rules. Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value.

Creating the validation rule for the phone number field in the consumer object

Creating the validation rule for the phone number field in the consumer object

Go to the setup page >> click on the object manager >> From the drop down, click edit for consumer object.

Click on the validation rule >> click New.

Enter the Rule name as “Phone number or email blank rule”.

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.

Profiles:

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. Profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visual-force page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example, System Administrator, Developer, Sales Representative.

Types of profiles in salesforce

Standard profiles:

By default, Salesforce provides below-standard profiles.

Contract Manager

Read Only

Marketing User

Solutions Manager

Standard User

System Administrator.

We cannot delete standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects

available on the platform.

Custom Profiles:

Custom ones defined by us.

They can be deleted if there are no users assigned to that particular one.

owner Profile

To create a new profile:

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

3. 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.

Agent Profile

Go to setup >> type profiles in the 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.

Roles and Hierarchy:

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Creating owner Role

Creating owner Role:

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.

Give Label as "owner" and Role name gets auto populated. Then click on Save.

Role Edit
New Role

Role Edit

Label: owner

Role Name: owner

This role reports to: CEO

Role Name as displayed on reports:

Save Save & New Cancel

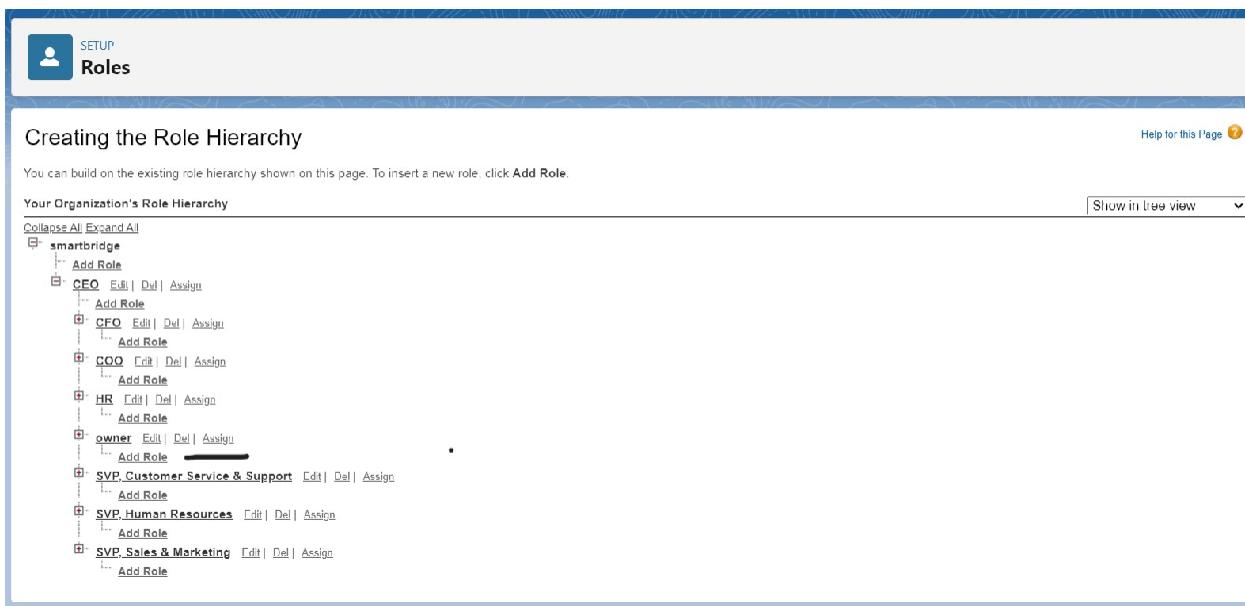
Click and save it.

Activity 2: Creating Agent roles

Creating another two roles under the manager

Go to quick find - Search for Roles - click on set up roles.

Click plus on the CEO role, and click add role under owner.



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

Users:

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings

determine what features and records the user can access.

Create User

Go to setup - type users in the quick find box - select users -click New user.

Fill in the fields

First Name : Vicky

Last Name : y

Alias : Give an 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.

New User

User Edit

General Information

First Name	vicky
Last Name	rushi
Alias	vrush
Email	udayrushi00@gmail.com
Username	udayrushi00@456.us@gmail
Nickname	vicky
Title	
Company	
Department	
Division	

Role	owner
User License	Salesforce
Profile	Standard User
Active	<input checked="" type="checkbox"/>
Marketing User	<input type="checkbox"/>
Offline User	<input type="checkbox"/>
Knowledge User	<input type="checkbox"/>
Flow User	<input type="checkbox"/>
Service Cloud User	<input type="checkbox"/>
Site.com Contributor User	<input type="checkbox"/>
Site.com Publisher User	<input type="checkbox"/>
WDC User	<input type="checkbox"/>
Data.com User Type	--None--
Data.com Monthly Addition Limit	Default Limit (300)
Accessibility Mode (Classic Only)	<input type="checkbox"/>

Save it.

Activity 2: creating another user

Go to setup -type users in the quick find box - select users -click New user.

Fill in the fields

First Name : ram

Last Name : ram

Alias : Give an 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.

The screenshot shows the 'User Edit' screen for a new user named 'ram'. The 'General Information' section includes fields for First Name, Last Name, Alias, Email, Username, Nickname, Title, Company, Department, and Division. The 'Profile' dropdown is set to 'Standard Platform User'. The 'Role' dropdown is set to 'Agent'. Other optional checkboxes like Marketing User, Offline User, Knowledge User, etc., are available. A note at the bottom indicates 'Data.com Monthly Addition Limit' and 'Accessibility Mode (Classic Only)'.

Save it.

Flows:

In Salesforce, a flow is a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps. Flows are built using a visual interface and can be created without any coding knowledge.

In Salesforce, "flows" typically refer to Salesforce Flow, which is a powerful automation tool that allows you to create custom, automated processes in your Salesforce org without writing code. Salesforce Flow is a point-and-click tool that enables you to design and automate complex business processes, collect data, and interact with users in a visual interface. There are different types of flows in Salesforce, including:

Screen Flows: These are used to guide users through a series of screens to collect or display information. Screen Flows are often used for data entry and updates.

Auto launched Flows: These are flows that are triggered by events, such as when a record is created or updated. They don't require user interaction and can be used for background automation.

Flow Builder: Flow Builder is the visual interface used to create flows. It allows you to design flows by adding elements, like screens, logic, and actions, using a drag-and-drop approach.

Flow Templates: Salesforce provides a library of pre-built flow templates that you can use as a starting point for your own flows. These templates cover a variety of use cases, from simple to complex.

Scheduled Flows: These are flows that you can schedule to run at specific times or intervals.

They are often used for automating recurring tasks.

Flow Elements: Flow Builder offers various elements that you can use to create flows, such as variables, decisions, loops, and more. These elements allow you to build sophisticated logic into your flows.

Sub flows: Subflows are reusable flow elements that you can incorporate into multiple flows, making it easier to manage and maintain complex processes.

Record-Triggered Flows: These are flows that are triggered when records meet specified criteria. They are often used for automating record updates and related actions.

Activity -

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

The screenshot shows the Salesforce Setup interface. In the left sidebar, under 'Process Automation', the 'Flows' item is selected and highlighted with a red box. Step 1 is indicated by a red box around the 'Setup' button. Step 2 is indicated by a red box around the 'Flows' link. Step 3 is indicated by a red box around the 'New Flow' button at the top right of the main content area. The main content area displays a table of 'Flow Definitions' with the following data:

Flow Label	Process Type	Ac...	To...	Package State	Pa...	Last Modified By	Last Modified ...
At A hour update	Autolaunched Flow	<input type="checkbox"/>	<input type="checkbox"/>	Unmanaged	Vive Venkata Varaprasad Androthu	07/06/2023, 11:35 a.m.	
Book Appointment from invitation	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Cancel Item Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Change Case Owner to incident Owner	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Close Change Request & Related Issues	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			

- 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.

The screenshot shows the 'Configure Start' screen in Flow Builder. It includes the following sections:

- Select Object:** A dropdown menu labeled 'Object' with 'Laptop Bookings' selected.
- Configure Trigger:** A section titled 'Trigger the Flow When:' with the option 'A record is created or updated' selected (indicated by a blue dot).
- Set Entry Conditions:** A note stating: '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.' Below this is a note: '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.'
- Buttons:** 'Cancel' and 'Done' buttons at the bottom right.

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 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

- Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “DecisionElement”.



- Enter the Details Label: Field should be Updated, 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 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 (1) (+) <ul style="list-style-type: none"> dell acer hp mac false 	OUTCOME DETAILS <table border="1" style="width: 100%;"> <tr> <td>* Label dell</td> <td>* Outcome API Name dell</td> </tr> <tr> <td colspan="2">Condition Requirements to Execute Outcome All Conditions Are Met (AND)</td> </tr> <tr> <td>Resource \$Record > Laptop names</td> <td>Operator Equals</td> <td>Value Dell</td> </tr> </table>	* Label dell	* Outcome API Name dell	Condition Requirements to Execute Outcome All Conditions Are Met (AND)		Resource \$Record > Laptop names	Operator Equals	Value Dell
* Label dell	* Outcome API Name dell							
Condition Requirements to Execute Outcome All Conditions Are Met (AND)								
Resource \$Record > Laptop names	Operator Equals	Value Dell						
Delete Outcome								
Cancel Done								

Edit Decision

* Label field updated	* API Name field_updated							
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.								
OUTCOME ORDER (1) (+) <ul style="list-style-type: none"> dell core i3 dell core i5 dell core i7 	OUTCOME DETAILS <table border="1" style="width: 100%;"> <tr> <td>* Label del core i3</td> <td>* Outcome API Name dellcore_i3</td> </tr> <tr> <td colspan="2">Condition Requirements to Execute Outcome All Conditions Are Met (AND)</td> </tr> <tr> <td>Resource \$Record > core type</td> <td>Operator Equals</td> <td>Value core i3</td> </tr> </table>	* Label del core i3	* Outcome API Name dellcore_i3	Condition Requirements to Execute Outcome All Conditions Are Met (AND)		Resource \$Record > core type	Operator Equals	Value core i3
* Label del core i3	* Outcome API Name dellcore_i3							
Condition Requirements to Execute Outcome All Conditions Are Met (AND)								
Resource \$Record > core type	Operator Equals	Value core i3						
Delete Outcome								
Cancel Done								

- So 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.

Edit Decision

*Label: months selected *API Name: 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.

OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome
1	*Label: 1 *Outcome API Name: X1 Condition Requirements to execute Outcome: All Conditions Are Met (AND) Resource: \$Record > how many months Operator: Equals Value: 1	Cancel Done
2		
3		
4		
5		

- Follow the above picture, and you will understand.
- 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.

Edit Decision

*Label: months selected *API Name: 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.

OUTCOME ORDER	OUTCOME DETAILS	Delete Outcome
1	*Label: 1 *Outcome API Name: X1 Condition Requirements to Execute Outcome: All Conditions Are Met (AND) Resource: \$Record > how many months Operator: Equals Value: 1	Cancel Done
2		
3		
4		
5		

- Follow the above picture, and 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.

creating flow on Acer laptop

1. Go to the flow page
2. Beside acer there is a symbol ‘+’ click on that.
3. Again, select decision
4. Enter the Details Label: Field is Update, API name: Gets Automatically Generated.
5. select the Outcome Details Label: acer core i3, Outcome API name: Gets Automatically Generated.
6. Resource: Select Record.core type.
7. Operator: Select Equals.

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										
# acer core i3	<input style="width: 100%;" type="text" value="acer core i3"/> <div style="display: flex; justify-content: space-between; width: 100%;"> *Label *Outcome API Name </div>	<input style="width: 100%;" type="button" value="Delete Outcome"/>									
# acer core i5											
# acer core i7											
Condition Requirements to Execute Outcome <div style="display: flex; align-items: center;"> All Conditions Are Met (AND) ▼ </div>											
Default Outcome											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Resource</td> <td style="width: 30%;">Operator</td> <td style="width: 40%;">Value</td> </tr> <tr> <td style="text-align: right; padding-right: 5px;">#Record > core type</td> <td style="text-align: right; padding-right: 5px;">Equals</td> <td style="text-align: right; padding-right: 5px;">core i3</td> </tr> <tr> <td colspan="3" style="text-align: right; padding-top: 5px;"> <input style="width: 100%;" type="button" value="Add Condition"/> </td> </tr> </table>			Resource	Operator	Value	#Record > core type	Equals	core i3	<input style="width: 100%;" type="button" value="Add Condition"/>		
Resource	Operator	Value									
#Record > core type	Equals	core i3									
<input style="width: 100%;" type="button" value="Add Condition"/>											
When to Execute Outcome <div style="display: flex; align-items: center;"> <input checked="" type="radio"/> If the condition requirements are met <input type="radio"/> Only if the record that triggered the flow to run is updated to meet the condition requirements </div>											

- 8.
9. Beside dell there is a symbol ‘+’ click on that.
10. Again, select decision
11. Enter the Details Label: months selected, API name: Gets Automatically Generated.
12. Enter the Outcome Details Label: acer 1(i3) , Outcome API name: Gets Automatically Generated.
13. Resource: Select Record.how many months.
14. Operator: Select Equals.
15. Value: 1.

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.																
OUTCOME ORDER 1 + <ul style="list-style-type: none"> # acer 1(i3) # acer 2(i3) # acer 3(i3) # acer 4(i3) # acer 5(i3) 	OUTCOME DETAILS <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">* Label</td> <td style="width: 30%;">* Outcome API Name</td> </tr> <tr> <td>acer 1(i3)</td> <td>acer_1_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 style="padding-top: 10px;">Resource</td> <td>Operator</td> <td>Value</td> </tr> <tr> <td>\$Record > how many months</td> <td>Equals</td> <td>1</td> </tr> </table>	* Label	* Outcome API Name	acer 1(i3)	acer_1_i3	Condition Requirements to Execute Outcome		All Conditions Are Met: (AND)		Resource	Operator	Value	\$Record > how many months	Equals	1	Delete Outcome
* Label	* Outcome API Name															
acer 1(i3)	acer_1_i3															
Condition Requirements to Execute Outcome																
All Conditions Are Met: (AND)																
Resource	Operator	Value														
\$Record > how many months	Equals	1														

16.

17. 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

Edit Update Records

one month of acer i3 rate (one_month_of_acer_i3_rate) edit

*** 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	Delete

Cancel Done

Click done.

creating a flow on hp laptop:

1. Go to the flow page

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

3.Again, select decision

4.Enter the Details Label: Field is Update, API name: Gets Automatically Generated. select the Outcome Details Label: hp core i5 , Outcome API name: Gets Automatically Generated.

Resource: Select Record.core type.

Operator: Select Equals.

Value: Select hp i5.

6.Beside hp there is a symbol '+' click on that.Again select decision

7.Enter the Details Label: hp field should be updated , API name: Gets Automatically Generated.

8.Enter the Outcome Details Label: hp 1(i5) , Outcome API name: Gets Automatically Generated.

Resource: Select Record.how many months.

Operator: Select Equals.

Value: 1.

Edit Decision

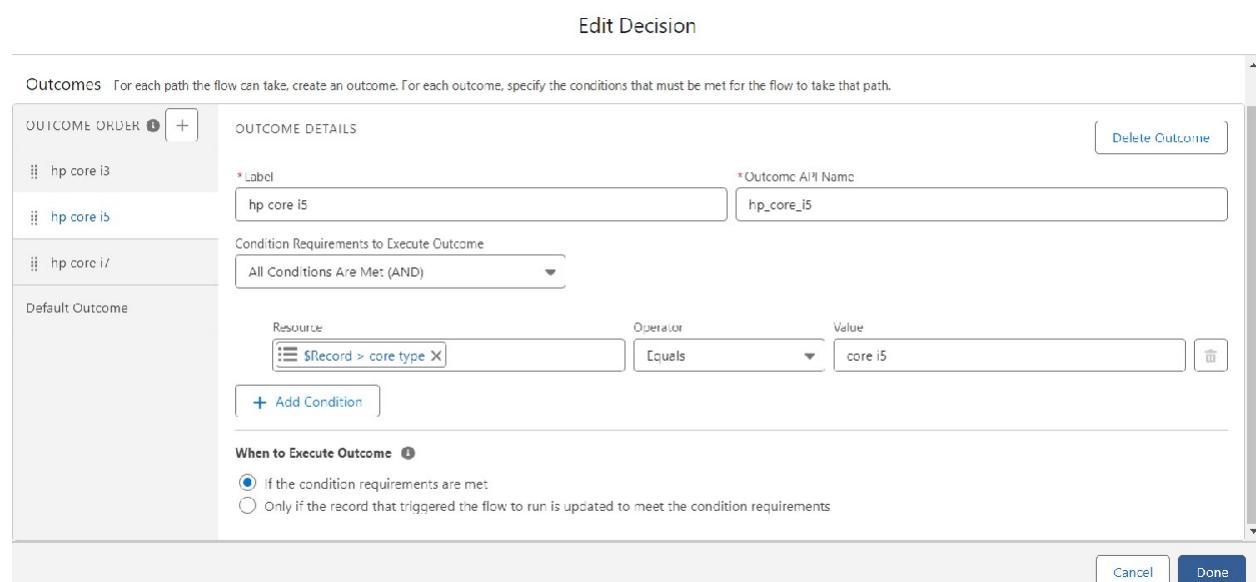
Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER	OUTCOME DETAILS	Actions
hp core i3	* Label: hp core i5 * Outcome API Name: hp_core_i5	Delete Outcome
hp core i5		
hp core i7	Condition Requirements to Execute Outcome All Conditions Are Met (AND)	
Default Outcome	Resource: \$Record > core type Operator: Equals Value: core i5	+ 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](#)



9.After hp 1(i5) there is '+' symbol like hp 2(i5), hp 3(i5), hp 4(i5),hp 5(i5).

10.Click on '+' then select update records

11.Enter the Details Label: one month of hp i5 rate , API name: Gets Automatically Generated.

Field:- Amount__c , value:- for hp 1(i5)-1700, hp 2(i5)-3400, hp 3(i5)-5100, hp 4(i5)-6800, hp5(i5)-8500.

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	Remove
Amount_c	1700	

[+ Add Field](#)

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

12. Click done.

creating a flow on mac laptop

1. Go to the 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.
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

5. Beside Mac there is a symbol '+' click on that. Again select decision

6. Enter the Details Label: Mac months selected, API name: Gets Automatically Generated.
7. Enter the Outcome Details Label: mac bionic chip(1), Outcome API name: Gets Automatically Generated.

Resource: Select Record.how many months.

Operator: Select Equals.

Value: 1.

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	
mac bionic chip(1)	*Label mac bionic chip(1) *Outcome API Name mac_bionic_chip_1	
mac bionic chip(2)		
mac bionic chip(3)		
mac bionic chip(4)		
mac bionic chip(5)		
Default Outcome		

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

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

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

8.Click done.

9.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).

10.Click on ‘+’ then select update records

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

12.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 month of mac bionic chip rate-8500. Follow for all these finally

Edit Update Records

* 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

13.Click done.

APEX

Apex OverView

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Lightning platform server in conjunction with calls to the Lightning Platform API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex code can be initiated by Web service requests and from triggers on objects.

It is as similar as java i.e, it also supports OOP(Object oriented programming) like Classes, objects, methods.

Creating Classes :

Apex classes are modeled on their counterparts in Java. You'll define, instantiate, and extend classes, and you'll work with interfaces, Apex class versions, properties, and other related class concepts.

Class:

As in Java, you can create classes in Apex. A class is a template or blueprint from which objects are created. An object is an instance of a class.

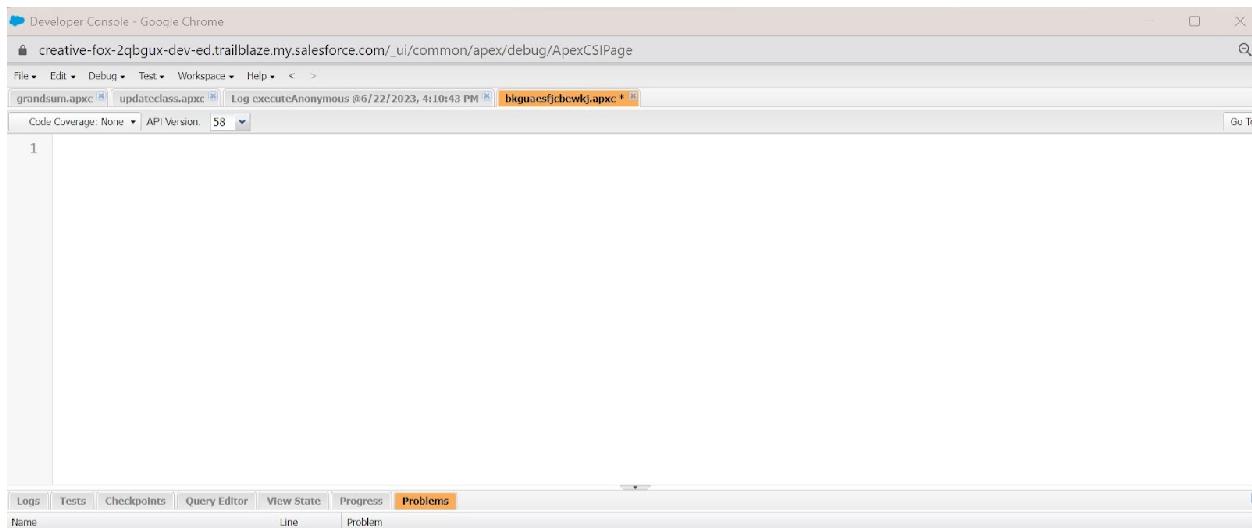
Object

Object is an instance of a class, where it can access all the properties that are present in a class i.e, variables and methods.

Steps to create a class in APEX:

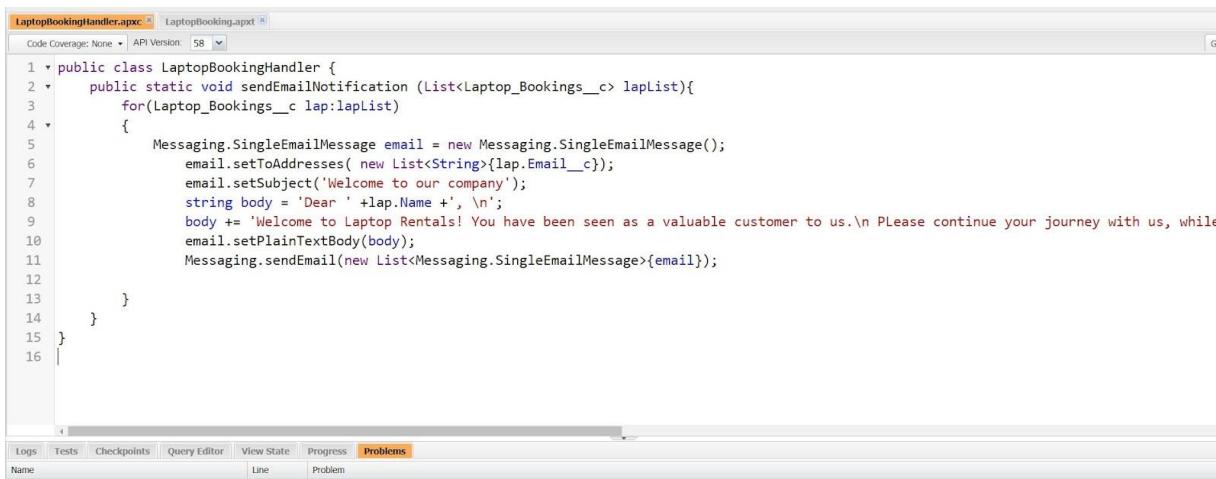
Login to the trailhead account and navigate to the gear account in the top right corner.

Then we can see the Developer console. Click on the developer console, and you will navigate to a new console window.



Then you can see many tools in the Toolbar of the new console window. Click on File, New, and Apex Class.

Enter the name of the class to create a new class file.



Access specifiers in Apex :

Apex allows you to use the private, protected, public, and global access modifiers when defining methods and variables.

While triggers and anonymous blocks can also use these access modifiers, they aren't as useful in smaller portions of Apex. For example, declaring a method as global in an anonymous block doesn't enable you to call it from outside of that code.

Private:

This access modifier is the default, and means that the method or variable is accessible only within the Apex class in which it's defined. If you don't specify an access modifier, the method or variable is private.

Protected:

This means that the method or variable is visible to any inner classes in the defining Apex class and to the classes that extend the defining Apex class. You can only use this access modifier for instance methods and member variables. This setting is strictly more permissive than the default (private) setting, just like Java.

Public :

This means that the method or variable is accessible by all Apex within a specific package. For accessibility by all second-generation (2GP) managed packages that share a namespace, use public with the @NamespaceAccessible annotation. Using the public access modifier in no-namespace packages implicitly renders the Apex code as @NamespaceAccessible.

Global:

This means the method or variable can be used by any Apex code that has access to the class, not just the Apex code in the same application. This access modifier must be used for any method that must be referenced outside of the application, either in SOAP API or by other Apex code. If you declare a method or variable as global, you must also declare the class that contains it as global.

Triggers :

A trigger is a set of Apex code that runs before or after DML(Data Manipulation Language) events. A DML event could be a variety of data processing tasks that include the standard insert, update, and delete commands.

With Apex triggers, you can automate tasks that would otherwise be nearly impossible to accomplish using only the Salesforce user interface. Triggers enable you to create custom scripts that you can implement according to your needs, and the only limitation is your coding skills.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is google-7da-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tabs at the top are 'LaptopBookingHandler.apxc' and 'LaptopBooking.apxt'. The code editor displays the following Apex trigger:

```
1 trigger LaptopBooking on Laptop_Bookings__c (After insert,after update) {
2
3     if(trigger.isAfter && ( trigger.isInsert || trigger.isupdate))
4     {
5         LaptopBookingHandler.sendEmailNotification(trigger.new);
6     }
7
8 }
9 }
```

Create Report

Go to the app -click on the reports tab

Click New Report.

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

4.Customize your report

5.Add fields from the left pane as shown below

Follow the above image to group rows and columns.

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

Click apply it.

Follow the picture and save or run it.

The screenshot shows a Salesforce Lightning Report titled "consumer with laptops and total laptops". The report displays a table with the following columns: types of version, consumer: consumer_name, Laptop Bookings: Laptop Bookings Name, Total No Of Laptops: Total Laptops, Laptop names, core type, Amount, Phone number, and Address. The table contains data for 5 records, totaling ₹11,400.

types of version	consumer: consumer_name	Laptop Bookings: Laptop Bookings Name	Total No Of Laptops: Total Laptops	Laptop names	core type	Amount	Phone number	Address
basic (3)	likitha Amrutha Sakshi	likitha Clear Discount Percentage Update Status to Deferred	16 15 14	Acer Hp Hp	core i5 core i5 core i7	₹500 ₹600 ₹800	9505886540 9912340575 9505886540	vij vij vij
Subtotal						₹1,900		
Intermediate (1)	kavya	Approval Status to Approved	12	Dell	core i3	₹1,000	6301678008	vij
Subtotal						₹1,000		
high (1)	Sarvani	Update Approval Status Submitted	16	Mac	Bionic chip	₹8,500	9912340575	vijayawada
Subtotal						₹8,500		
Total (5)						₹11,400		

Row Counts: Detail Rows: Subtotals: Grand Total:

Dashboards

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.



Thank You