Project Title
A CRM APPLICATION FOR LAPTOP RENTALS

1.Project Overview

This project focuses on developing a CRM application for laptop rentals, designed to streamline and enhance the management of rental operations. The primary goal is to create a user-friendly platform that simplifies customer interactions, rental tracking, and inventory management. By leveraging a CRM system tailored to the rental business, we aim to improve operational efficiency, customer service quality, and data accuracy. This project will support the long-term goal of building a seamless, integrated experience for both the rental business and its clients, ensuring reliable service and better resource management.

2.Objectives

Object Creation

Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects
Salesforce objects are of two types:

1.Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.

2.Custom Objects: Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.
Click on gear icon >> click setup.

To create an object:

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

2.On Custom object defining page:

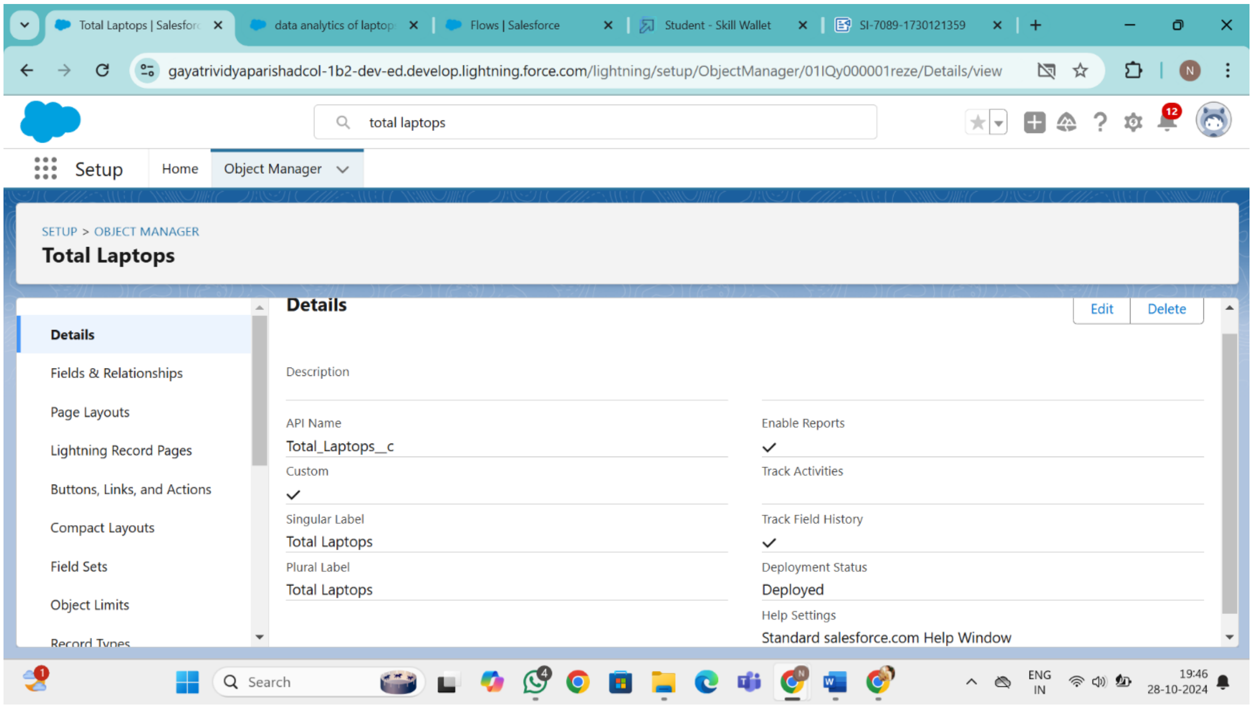
3.Enter the label name, plural label name, click on Allow reports, Allow search.

4.Click on Save.

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 Total Laptops Object



gayatrividyaparishadcol-1b2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IQy000001reze/Details/view

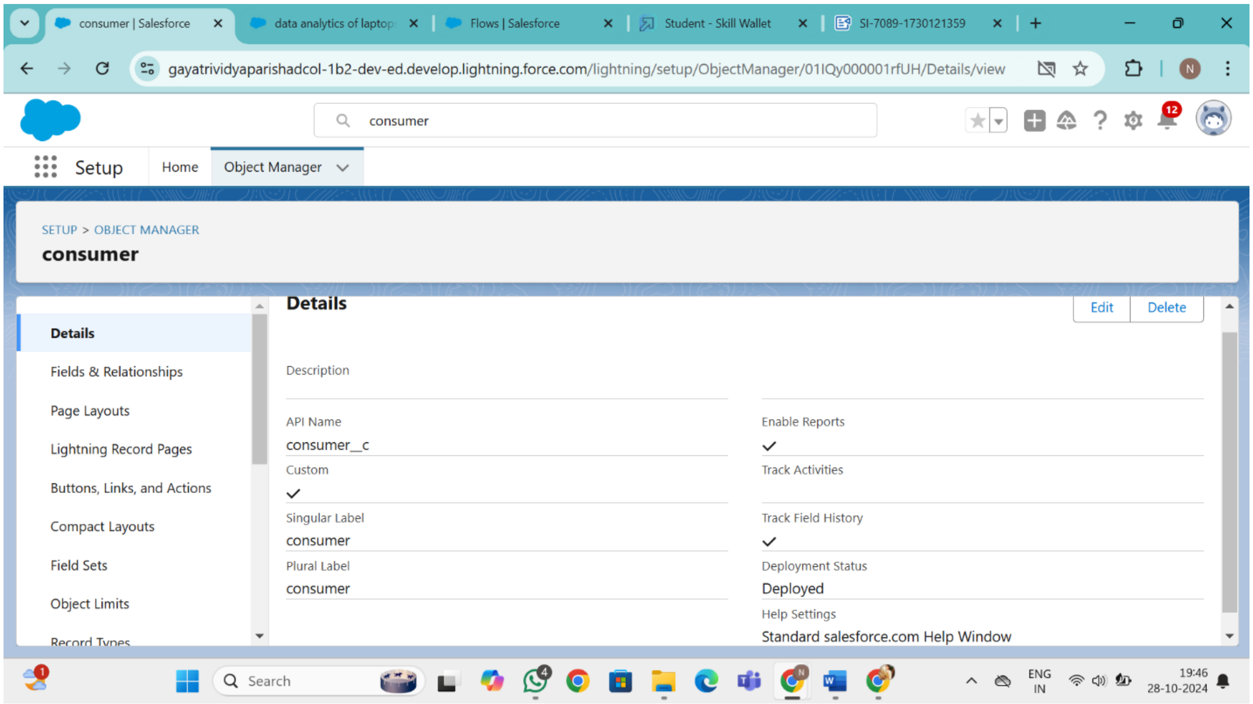
SETUP > OBJECT MANAGER
Total Laptops

Details

Description	
API Name Total_Laptops__c	Enable Reports ✓
Custom ✓	Track Activities
Singular Label Total Laptops	Track Field History ✓
Plural Label Total Laptops	Deployment Status Deployed
	Help Settings Standard salesforce.com Help Window

Similarly the custom objects are created Consumer, Laptop Bookings and Billing Process

Create consumer Object



gayatrividyaparishadcol-1b2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IQy000001rfUH/Details/view

SETUP > OBJECT MANAGER

consumer

Details

Description

API Name: consumer__c

Custom: ✓

Singular Label: consumer

Plural Label: consumer

Enable Reports: ✓

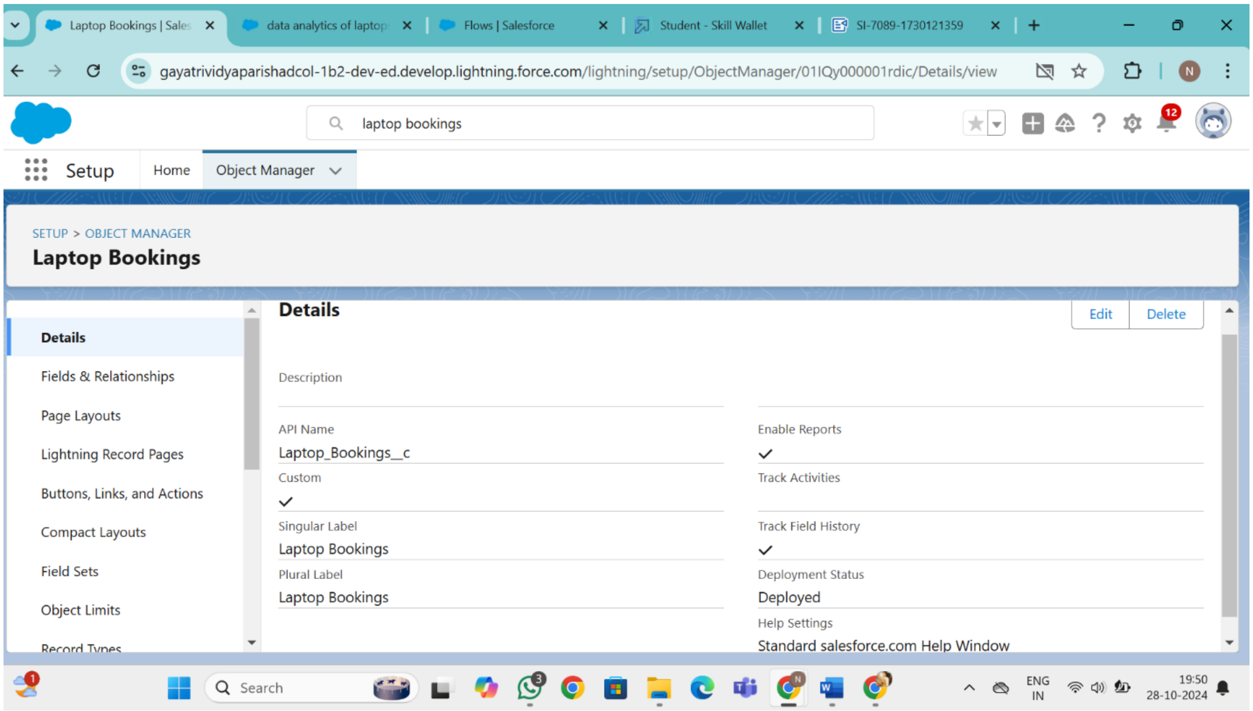
Track Activities

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Create Laptop Bookings Object



gayatrividyaparishadcol-1b2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IQy000001rdic/Details/view

SETUP > OBJECT MANAGER

Laptop Bookings

Details

Description

API Name: Laptop_Bookings__c

Custom: ✓

Singular Label: Laptop Bookings

Plural Label: Laptop Bookings

Enable Reports: ✓

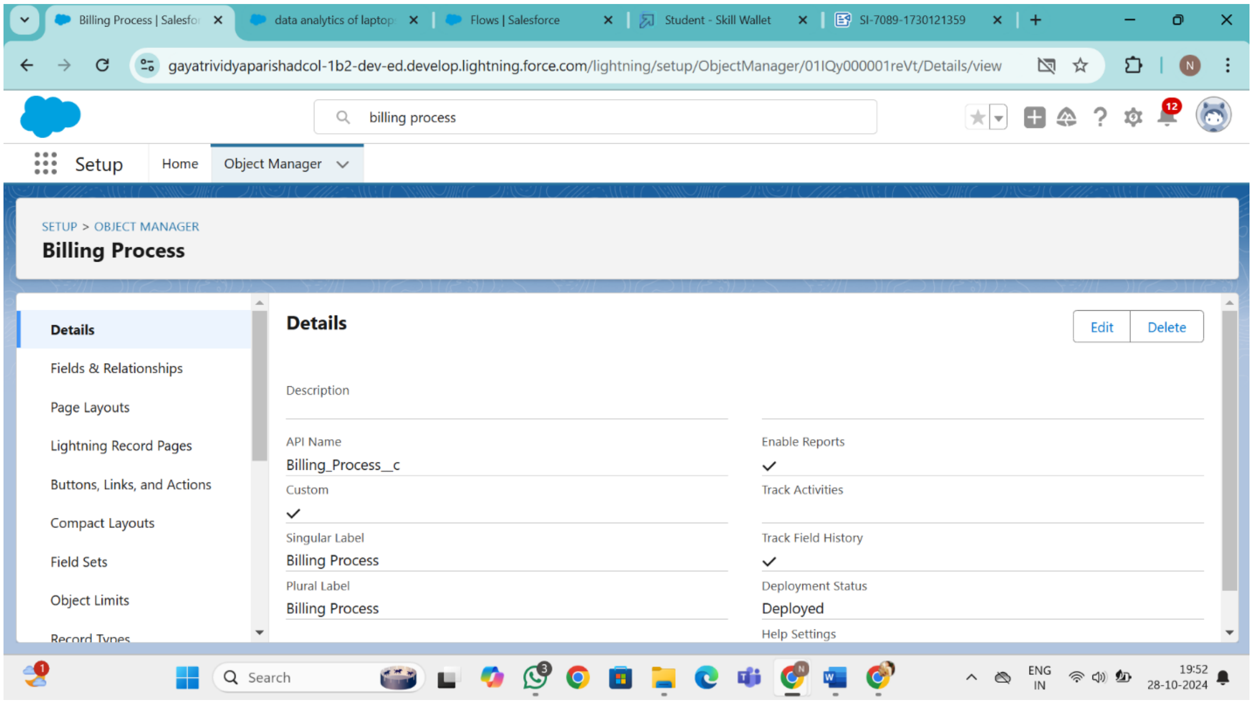
Track Activities

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Create Billing Process Object



Details

Description

API Name Billing_Process__c	Enable Reports ✓
Custom ✓	Track Activities
Singular Label Billing Process	Track Field History ✓
Plural Label Billing Process	Deployment Status Deployed
	Help Settings

Tabs

A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1.Custom Tabs:

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

2.Web Tabs:

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

3.Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

4.Lightning Component Tabs

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

5.Lightning Page Tabs

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu. Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

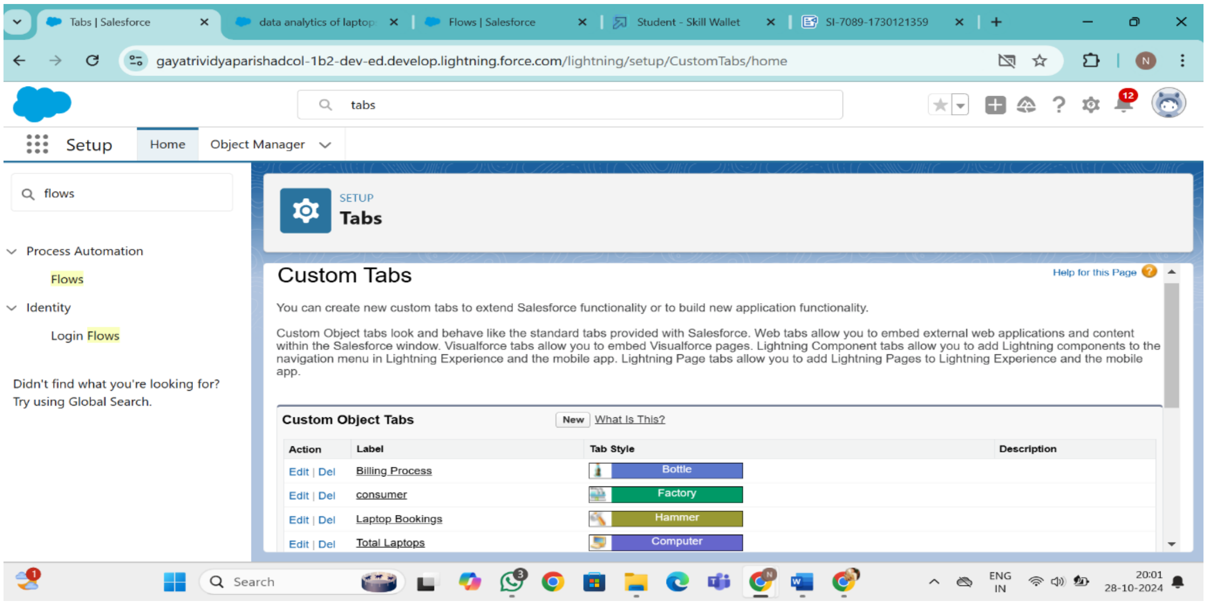
Creating a Custom Tab

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.

Activity 2: Creating Remaining Tabs

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



Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs New What Is This?

Action	Label	Tab Style	Description
Edit | Del	Billing Process	Bottle	
Edit | Del	consumer	Factory	
Edit | Del	Laptop Bookings	Hammer	
Edit | Del	Total Laptops	Computer	

3.Salesforce Key Features and Concepts Utilized

The Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.
Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

Create a Lightning App

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.

3.Upload a photo that is related to your app.

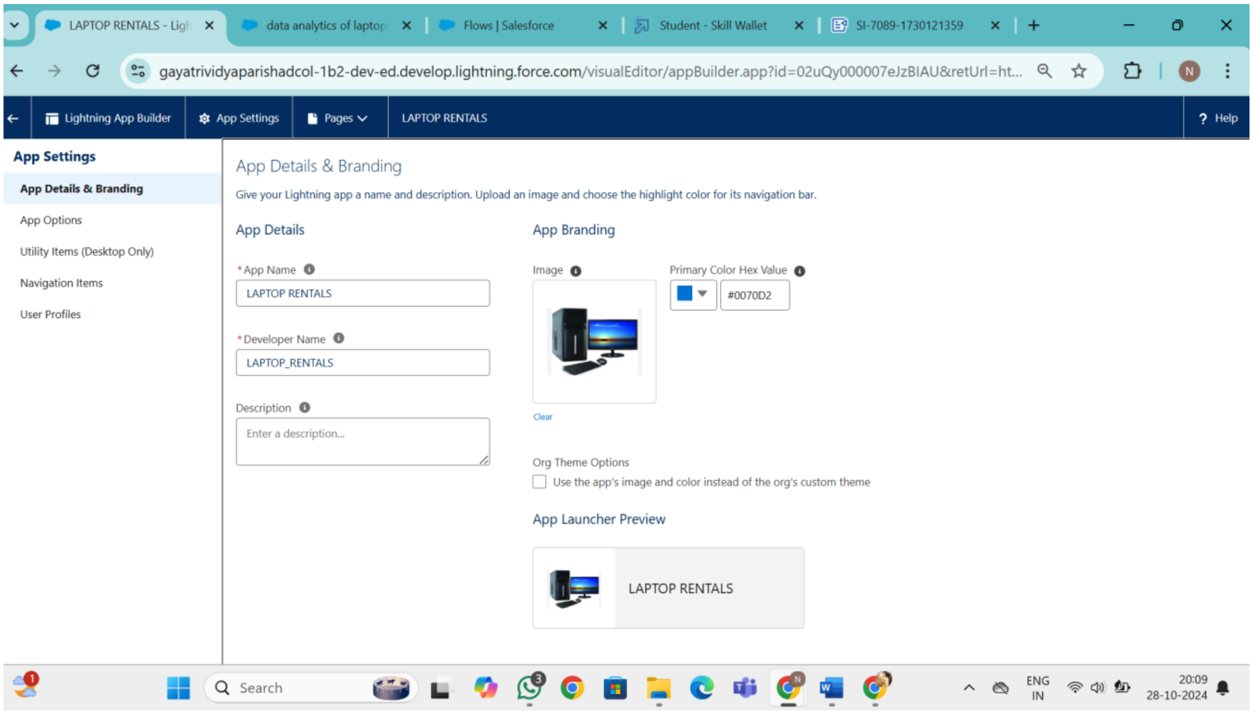
To Add Navigation Items:

Select the items (Total Laptops,consumer,Laptop Booking,Billing Process) from the search bar and move it using the arrow button >> Next.

To Add User Profiles:

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

Creating Laptop Rentals in Lightning App:



gayatrividyaparishadcol-1b2-dev-ed.develop.lightning.force.com/visualEditor/appBuilder.app?id=02uQy000007eJzBIAU&retUrl=ht...

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: LAPTOP RENTALS

*Developer Name: LAPTOP_RENTALS

Description: Enter a description...

App Branding

Image

Primary Color Hex Value: #0070D2

Org Theme Options

□ Use the app's image and color instead of the org's custom theme

App Launcher Preview

LAPTOP RENTALS

4.Detailed Steps to Solution Design

Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Types of Fields:

1.Standard Fields

2.Custom Fields

Standard Fields:

As the name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard task. The main point is that you can't simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salesforce application. They are,

>>Created By

>>Owner

>> Last Modified
>> Field Made During object Creation

Custom Fields:
On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organizer or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.

Creating the field in consumer object

To create fields in an object:

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

Now click on “Fields & Relationships” >> New
Select Data Type as a “Phone”
Click on next
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.

To create another fields in an object:

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

Now click on “Fields & Relationships” >> New
Select Data type as a “Email” and Click on Next
Fill the Above as following:
Field Label: Email
Field Name :It’s gets auto generated
Click on Next >> Next >> Save and new.

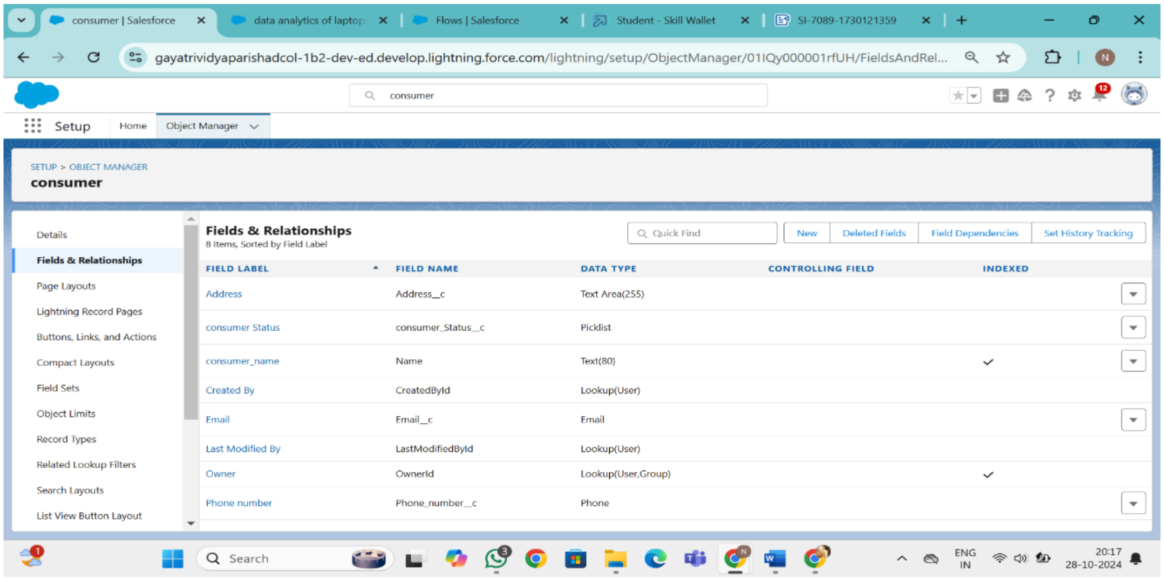
To create another fields in an object:

Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.
Now click on “Fields & Relationships” >> New
Select Data type as a “Text Area” and Click on Next
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:

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

Now click on “Fields & Relationships” >> New
Select Data type as a “Picklist” and Click on Next
Fill the Above as following:
Field Label: consumer Status
Value - Select enter values with each value separated by a new line
Student
Employee
Others
Select required
Field Name :It’s gets auto generated
Click on Next >> Next >> Save and new.



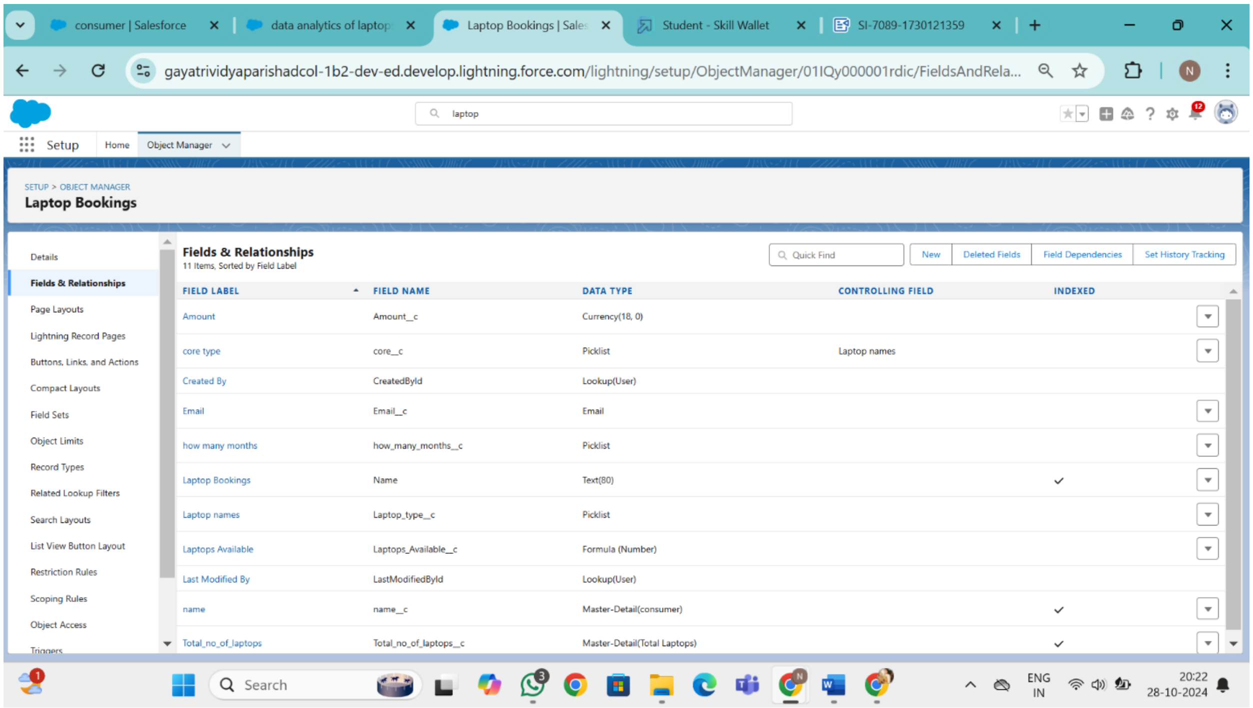
consumer

Fields & Relationships
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
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		

Similarly,

Creating the field in Laptops Bookings object



SETUP > OBJECT MANAGER

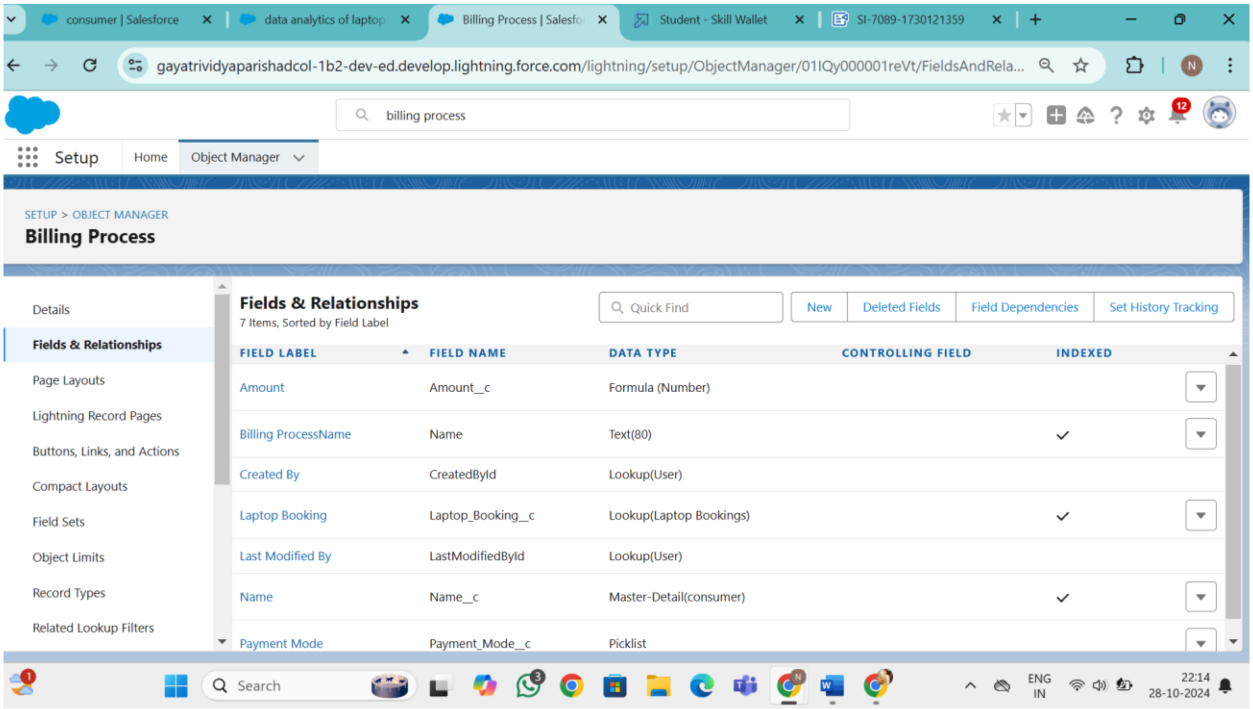
Laptop Bookings

Fields & Relationships

11 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Currency(18, 0)		
core type	core__c	Picklist	Laptop names	
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
how many months	how_many_months__c	Picklist		
Laptop Bookings	Name	Text(80)		✓
Laptop names	Laptop_type__c	Picklist		
Laptops Available	Laptops_Available__c	Formula (Number)		
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)		✓

Creation of Fields & Relationship for Billing Process Object



SETUP > OBJECT MANAGER

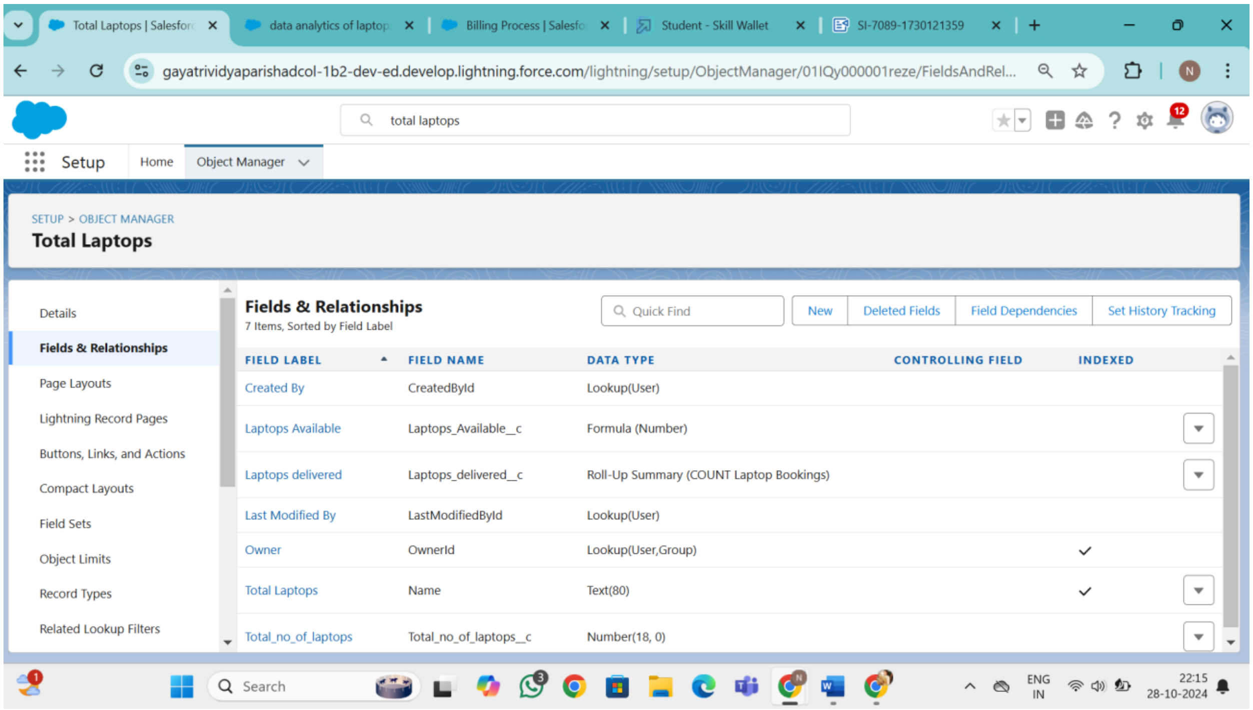
Billing Process

Fields & Relationships

7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Formula (Number)		
Billing ProcessName	Name	Text(80)		✓
Created By	CreatedById	Lookup(User)		
Laptop Booking	Laptop_Booking__c	Lookup(Laptop Bookings)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Master-Detail(consumer)		✓
Payment Mode	Payment_Mode__c	Picklist		

Creating the field in Total Laptops object



Total Laptops

Fields & Relationships

7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
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)		✓
Total_no_of_laptops	Total_no_of_laptops__c	Number(18, 0)		

5.Testing and Validation

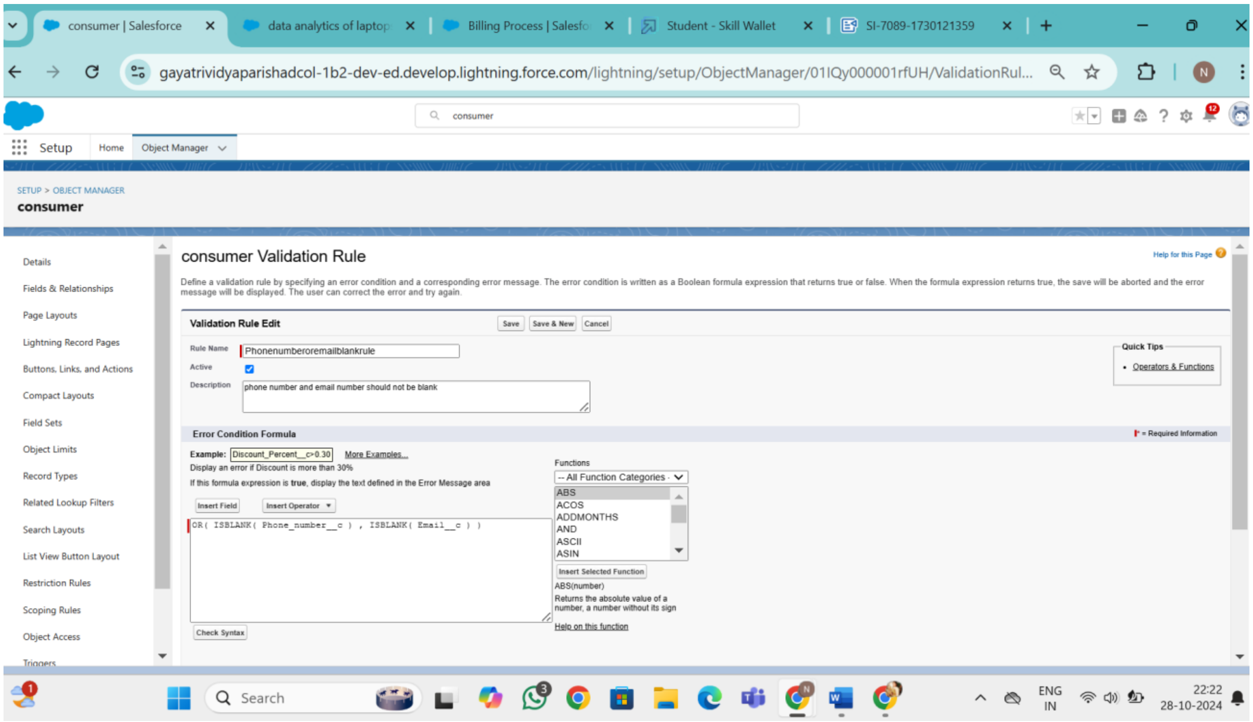
Validation rule

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 phone number field in consumer object

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



consumer Validation Rule

Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.

Validation Rule Edit

Rule Name: Phonenumberoremailblankrule
Active: ☑
Description: phone number and email number should not be blank

Error Condition Formula

Example: Discount_Percent__c>0.30 More Examples...
Display an error if Discount is more than 30%
If this formula expression is **true**, display the text defined in the Error Message area

```
OR( ISBLANK( Phone_number__c ) , ISBLANK( Email__c ) )
```

ABS(number)
Returns the absolute value of a number, a number without its sign

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.

Autolaunched 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.

Subflows: 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.

Why do we need to create a flow:

To get the Amount Field automatic by the selection of laptop types the Amount is generated Automatically in the amount field.

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.

   Select the Object as a Laptop Booking in the Drop down list.

Select the Trigger Flow when: “A record is Created or Updated”.
Select the Optimize the flow for: “Actions and Related Records” and Click on Done.

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

Enter the Details Label: Field should be Update, API name: Gets Automatically Generated.
Enter the Outcome Details Label: dell , Outcome API name: Gets Automatically Generated.
Resource: Select Record.Laptop booking__c.
Operator: Select Equals.
Value: Select dell
Add the same outcome order to acer , hp,mac.
Click done.

Go to flow page

Beside dell there is a symbol   ‘+’   click on that.
Again select decision
Enter the Details Label: Field should Update(any one u want), API name: Gets Automatically Generated.
select the Outcome Details Label: dell core i3 , Outcome API name: Gets Automatically Generated.
Resource: Select Record.core type.
Operator: Select Equals.
Value: Select core i3.
Then again click the symbol   ‘+’ outcome details

select the Outcome ‘+’   Details Label: dell core i5 , Outcome API name: Gets Automatically Generated.
Resource: Select Record.core type.
Operator: Select Equals.
Value: Select core i5.
Then again click the symbol’+’ outcome details
Enter the Outcome Details Label: dell core i7 , Outcome API name: Gets Automatically Generated.
Resource: Select Record.core type.
Operator: Select Equals.
Value: Select core i7.

Click done.

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.
Resource: Select Record.how many months.
Operator: Select Equals.
Value: Select 2..
Click ‘+’ outcome details
Enter the Outcome Details Label: dell 3(i3) , Outcome API name: Gets Automatically Generated.
Resource: Select Record.how many months.
Operator: Select Equals.
Value: Select 3..
Click ‘+’ outcome details
Enter the Outcome Details Label: dell 4(i3) , Outcome API name: Gets Automatically Generated.
Resource: Select Record.how many months.
Operator: Select Equals.
Value: Select 4..
Click ‘+’ outcome details
Enter the Outcome Details Label: dell 5(i3) , Outcome API name: Gets Automatically Generated.
Resource: Select Record.how many months.
Operator: Select Equals.
Value: Select 4.

Follow the above picture 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.

Enter the Details Label: months selected , API name: Gets Automatically Generated.
Enter the Outcome Details Label: dell 1(i7) , Outcome API name: Gets Automatically Generated.

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

Enter the Outcome Details Label: dell 2(i7) , Outcome API name: Gets Automatically Generated.
Resource: Select Record.how many months.
Operator: Select Equals.
Value: Select 2..
Click '+' outcome details
Enter the Outcome Details Label: dell 3(i7) , Outcome API name: Gets Automatically Generated.
Resource: Select Record.how many months.
Operator: Select Equals.
Value: Select 3..
Click '+' outcome details
Enter the Outcome Details Label: dell 4(i7) , Outcome API name: Gets Automatically Generated.
Resource: Select Record.how many months.
Operator: Select Equals.
Value: Select 4.
Click '+' outcome details
Enter the Outcome Details Label: dell 5(i7) , Outcome API name: Gets Automatically Generated.
Resource: Select Record.how many months.
Operator: Select Equals.
Value: Select 5.

Follow the above picture you will understand.
After dell 1(i7) there is '+' symbol like dell 2(i7),dell 3(i7),dell 4(i7),dell 5(i7).
Click on '+' then select update records

Enter the Details Label: one month of dell i5 rate , API name: Gets Automatically Generated.
Field:- Amount__c , value:- for dell 1(i7)-2000, dell 2(i7)-4000, dell 3(i7)-6000, dell 4(i7)-8000, dell 5(i7)-10000. Follow for all these finally
Click done.

creating flow on acer laptop

Go to flow page

Beside acer there is a symbol '+' click on that.
Again select decision
Enter the Details Label: Field is Update, API name: Gets Automatically Generated.
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.

Click done.
Go to flow page
Beside dell there is a symbol '+' click on that.
Again select decision

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

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

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..
Click '+' outcome details

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..
Click '+' outcome details
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.
Click '+' outcome details
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.

Click done.
After acer 1(i3) there is '+' symbol like acer 2(i3),acer 3(i3),acer 4(i3),acer 5(i3).
Click on '+' then select update records
Enter the Details Label: one month of acer i3 rate , API name: Gets Automatically Generated.
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

Click done.

creating a flow on hp laptop

Go to flow page

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

Go to flow page

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

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

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..
Click '+' outcome details
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..
Click '+' outcome details
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.
Click '+' outcome details
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.
Click on done.
After hp 1(i5) there is '+' symbol like hp 2(i5), hp 3(i5), hp 4(i5),hp 5(i5).
Click on '+' then select update records
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, hp 5(i5)-8500. Follow for all these finally

Click done.

creating a flow on mac laptop

Go to flow page
Beside mac there is a symbol '+' click on that.
Again select decision
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.

Click done.
Go to flow page
Beside Mac there is a symbol '+' click on that.
Again select decision

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

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: Select 2..
Click '+' outcome details
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: Select 3..
Click '+' outcome details
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: Select 4.
Click '+' outcome details
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: Select 5.
Click done.
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).
Click on '+' then select update records
Enter the Details Label: one month of mac rate , API name: Gets Automatically Generated.
month of mac bionic chip rate-6800,   five month of mac bionic chip rate-8500.
Follow for all these finally
Click done.

FLOW:

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 Click on save .
Label:- Laptop distributions, api name:- automatically filled
Save the flow and activate it.

Final Flow Builder:

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. This is how a new class is created :

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.

There are two Salesforce Apex trigger types:

Before triggers. These are helpful in cases that require a validation process before accepting a change. They run before any database changes. After triggers. These are helpful in cases where you need to modify your database records and when the necessary value is stored in other records. They run after any database changes. Both types will help you perform custom tasks and manage records effectively. They can help you perform bulk actions as they can handle several records simultaneously.

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.

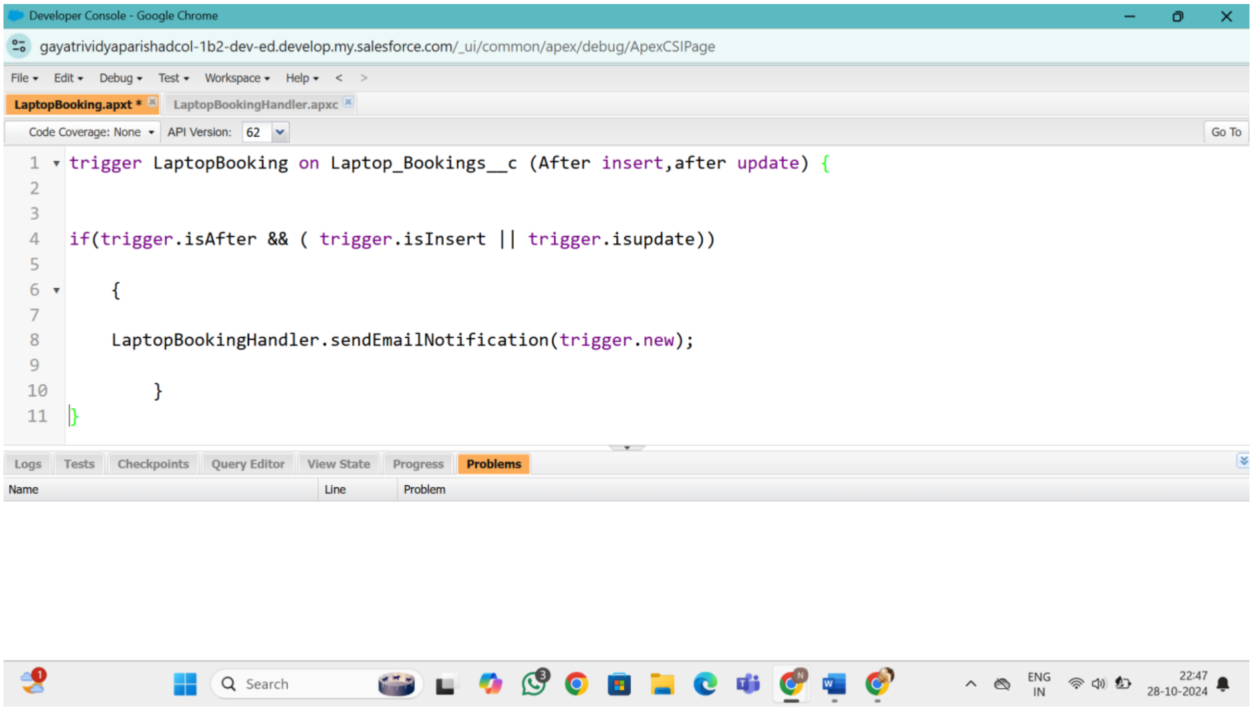
Syntax For creating trigger :

The syntax for creating trigger is :

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

{

}
```

Trigger code:

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

  if(trigger.isAfter && ( trigger.isInsert || trigger.isupdate))

    {

    LaptopBookingHandler.sendEmailNotification(trigger.new);

        }

}
```



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


if(trigger.isAfter && ( trigger.isInsert || trigger.isupdate))

    {

    LaptopBookingHandler.sendEmailNotification(trigger.new);

        }
}
```

Note:- copy the API names

1.LaptopBooking - trigger name

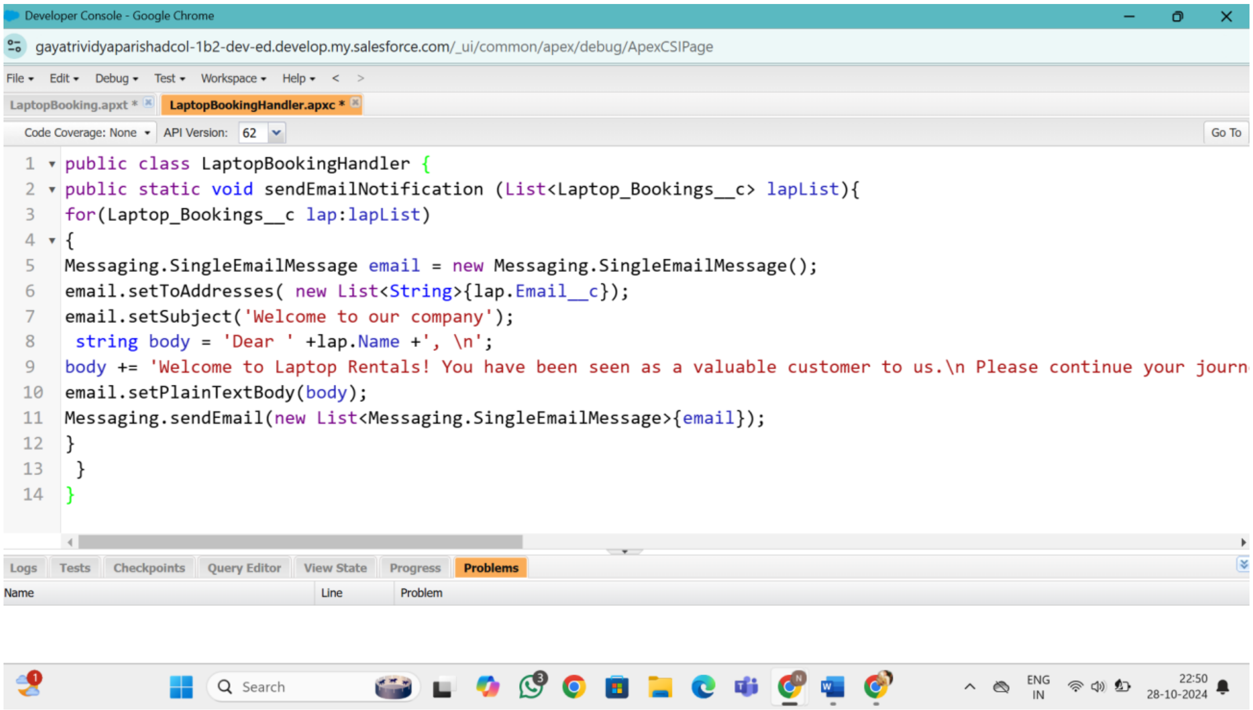
2.Laptop_Bookings__c -as per your org(go to laptop bookings object and copy from that object api name).

Handler Class:

Code Snippet :

```
public class LaptopBookingHandler {

    public static void sendEmailNotification (List<Laptop_Bookings__c> lapList){

        for(Laptop_Bookings__c lap:lapList)

        {

            Messaging.SingleEmailMessage email = new
Messaging.SingleEmailMessage();

                email.setToAddresses( new List<String>{lap.Email__c});
```

```
              email.setSubject('Welcome to our company');

            string body = 'Dear ' +lap.Name +', \n';

            body += 'Welcome to Laptop Rentals! You have been seen as a valuable customer to us.\n Please continue your journey with us, while we try to provide you with good quality resources. \n Laptop Amount = ' + lap.Amount__c + ' \n core type = '+lap.core__c +' \n Laptop type = '+lap.Laptop_type__c;

            email.setPlainTextBody(body);

              Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});



        }

    }

}
```



Developer Console - Google Chrome
gayatrividyaparishadcol-1b2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
LaptopBooking.apxt *
LaptopBookingHandler.apxc *
Code Coverage: None ▾
API Version: 62
Go To

```
public class LaptopBookingHandler {
public static void sendEmailNotification (List<Laptop_Bookings__c> lapList){
for(Laptop_Bookings__c lap:lapList)
{
Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
email.setToAddresses( new List<String>{lap.Email__c});
email.setSubject('Welcome to our company');
 string body = 'Dear ' +lap.Name +', \n';
body += 'Welcome to Laptop Rentals! You have been seen as a valuable customer to us.\n Please continue your journ
email.setPlainTextBody(body);
Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
}
 }
}
```

Note:-

1.Class name:- LaptopBookingHandler

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.

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).

6.Key Scenarios Addressed by Salesforce in the Implementation Project

Reports

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

In Salesforce.com we can easily generate reports in different styles. And can create reports in a very short time and also schedule the reports. Salesforce provides a powerful suit of analytic tools to help you organize, view and analyze your data.

Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix

4. Joined Reports

1. Tabula Reports: Simple listing of data without any subtotals. This type of reports provide you most basically to look at your data. Use tabular reports when you want a simple list or a list of items with a grand total.

Example: This type of reports are used to list all accounts, List of contacts, List of opportunities…..etc.….

2. Summary Reports: This type of reports provide a listing of data with groupings and sub totals. Use summary reports when you want subtotals based on the value of a particular field or when you want to create a hierarchically grouped report, such as sales organized by year and then by quarter.

Example: All opportunities for your team sub totaled by Sales Stage and Owner.

3. Matrix Reports: This type of reports allow you to group records both by row and by column. A comparison of related totals, with totals by both row and column. Use matrix reports when you want to see data by two different dimensions that aren't related, such as date and product.

Example: Summarize opportunities by month vertically and by account horizontally.

4. Joined Reports: Blocks of related information in a single report. This type of reports enable you to adopt five different blocks to display different types of related data. Each block can own unique columns, summary fields, formulas, filters and sort order. Use joined reports to group and show data from multiple report types in different views.

Example: You can build a report to show opportunity, case and activity data for your accounts.

Create Report

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.Customize your report

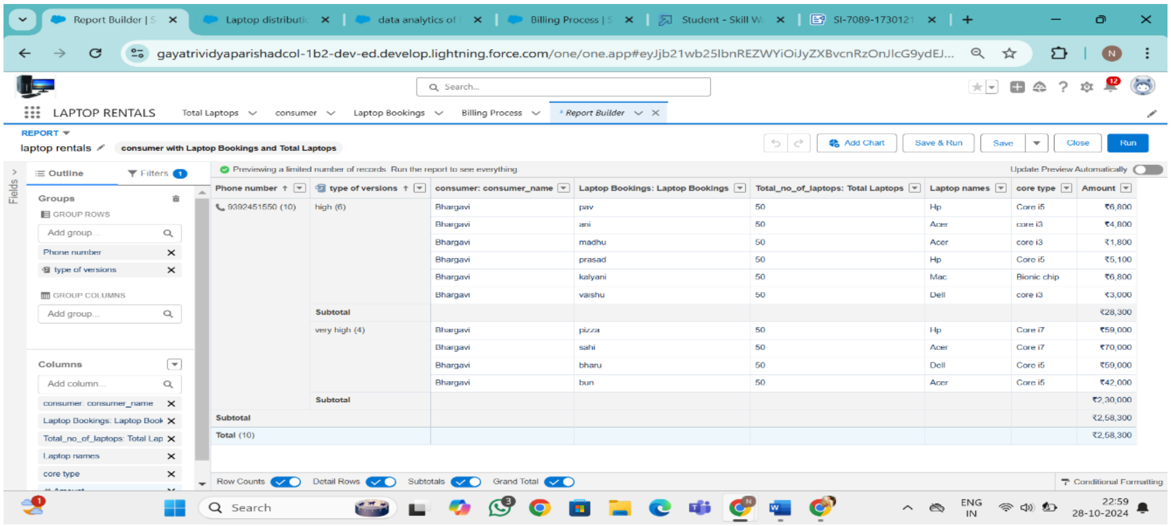
5.Add fields from left pane as shown below

Follow the above image 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.



Phone number ↑	type of versions ↑	consumer: consumer_name	Laptop Bookings: Laptop Bookings	Total_no_of_laptops: Total Laptops	Laptop names	core type	Amount
9392451550 (10)	high (6)	Bhargavi	pav	50	Hp	Core i5	₹6,800
		Bhargavi	ani	50	Acer	core i3	₹4,800
		Bhargavi	madhu	50	Acer	core i3	₹1,800
		Bhargavi	prasad	50	Hp	Core i5	₹5,100
		Bhargavi	kalyani	50	Mac	Bionic chip	₹6,800
		Bhargavi	vaishu	50	Dell	core i3	₹3,000
	Subtotal						₹28,300
	very high (4)	Bhargavi	pizza	50	Hp	Core i7	₹59,000
		Bhargavi	sahi	50	Acer	Core i7	₹70,000
		Bhargavi	bharu	50	Dell	Core i5	₹59,000
		Bhargavi	bun	50	Acer	Core i5	₹42,000
	Subtotal						₹2,30,000
Subtotal							₹2,58,300
Total (10)							₹2,58,300

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.

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

Go to the app >> click on the Dashboards tabs.

Give a Name and select the folder that was created, and click on create. Select add component.

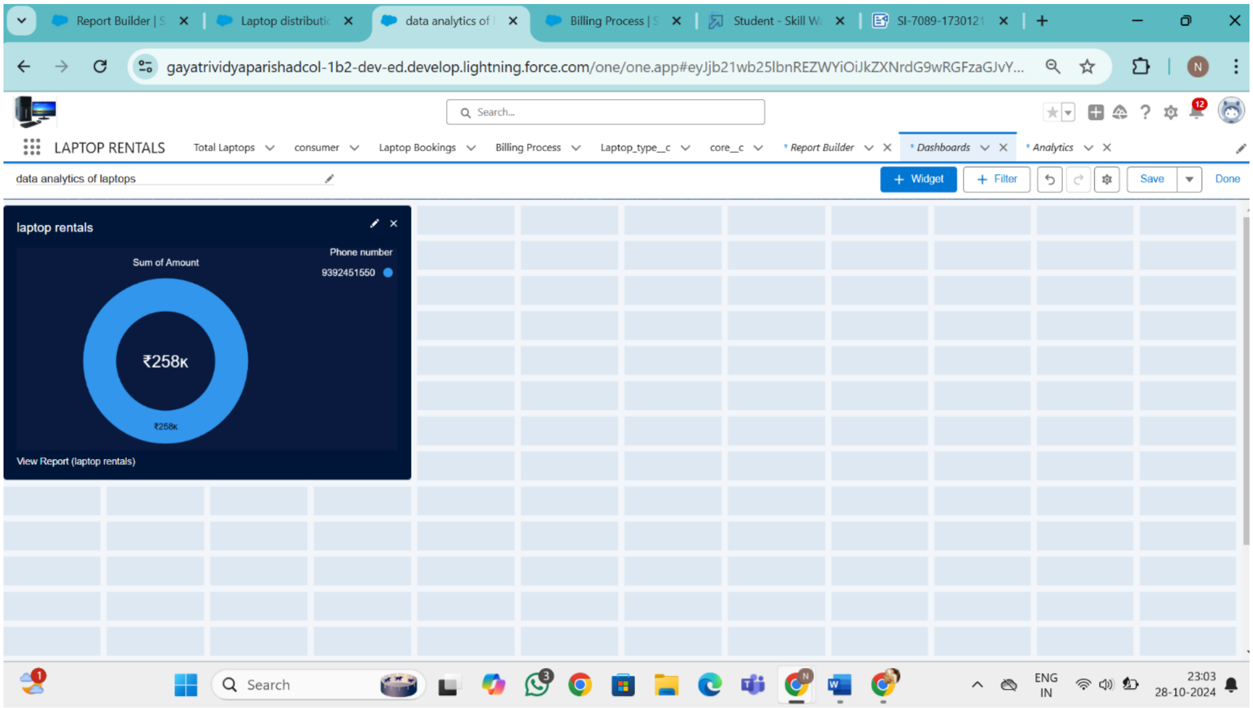
Select a Report and click on select.

Select the dark component and add to the dashboards.

Save it.

Click done.

Output:



Conclusion:

In conclusion, the CRM application for laptop rentals will transform the way rental operations are managed, making it easier to track inventory, manage customer interactions, and streamline daily tasks. By implementing this system, the rental business can improve efficiency, deliver better customer service, and maintain accurate records, ultimately supporting growth and reliability. This project sets a solid foundation for future expansion and adapts to evolving customer needs, helping the business run smoothly and meet its long-term goals.