

# A CRM APPLICATION FOR LAPTOP RENTALS

**BY :**

**HARISH KURMANA**

**harishkurmana1103@gmail.com**



# **PROJECT ABSTRACT**

## **Salesforce Laptop Bookings System**

### **Objective:**

The primary objective of this project is to develop a comprehensive laptop booking management system using Salesforce. This system aims to streamline the booking process for various laptop models, automate notifications, and generate insightful reports to assist in business decision-making.

### **Scope:**

The project encompasses the creation of custom objects and fields within Salesforce to track laptop bookings, including detailed records for different laptop types, core configurations, and booking amounts. It also involves the development of Apex triggers and handler classes to automate email notifications when certain conditions are met, such as exceeding a threshold amount. Additionally, the project includes the creation of custom reports to analyze booking data and track performance metrics.

### **Methodology:**

#### **1. Customization of Salesforce Objects:**

- a. Creation of a custom object, Laptop\_Bookings\_\_c, to capture booking details.
- b. Addition of fields such as Laptop Names, Amount, Total Number of Laptops, Email, Duration, and Core Types.

#### **2. Apex Development:**

- a. Implementation of an Apex trigger to monitor insertions and updates to the Laptop\_Bookings\_\_c object.
- b. Development of a handler class, LaptopBookingHandler, to calculate booking amounts and send email notifications based on predefined thresholds.

- c. Use of bucket fields to categorize booking amounts into predefined ranges for better analysis.
- d. Grouping and summarizing data to generate actionable insights.

### 3. Report Creation:

- a. Design and customization of reports to display booking data.
- b. Use of bucket fields to categorize booking amounts into predefined ranges for better analysis.
- c. Grouping and summarizing data to generate actionable insights.

### Outcomes:

- **Automated Notifications:** Successful implementation of email notifications triggered when booking amounts exceed specified thresholds, enhancing customer communication and engagement.

**Efficient Data Management:** Streamlined tracking of laptop bookings with real-time updates and accurate data representation.

### Conclusion:

This Salesforce project effectively integrates custom object management, automation through Apex triggers, and advanced reporting to improve the laptop booking process. By leveraging Salesforce's robust features, the project delivers a scalable solution that enhances operational efficiency and supports data-driven business strategies.

# **INDEX Page**

<b>S no</b>	<b>Topics</b>	<b>Page no</b>
1	Introduction	4
2	Salesforce	5
3	Object Creation	7
4	Tabs	10
5	The Lightning App	12
6	Fields	14
7	Validation Rule	21
8	Profiles	22
9	Roles and Hierarchy	25
10	Users	26
11	Flows	28
12	Apex	31
13	Reports	36
14	Dashboards	40

# Introduction:

Welcome to our **CRM Application for Laptop Rentals**—a sophisticated solution designed to transform the way you manage your rental operations. This application utilizes the power of customer relationship management (CRM) to provide an exceptional rental experience, streamline store operations, and elevate efficiency across the board.

Our CRM application not only ensures seamless delivery of laptops to your customers but also enhances communication through targeted email outreach. By integrating these capabilities, we aim to optimize customer interactions and operational workflows, ultimately driving a superior rental experience and fostering stronger customer relationships.

## SALESFORCE:

### Introduction:

Are you new to Salesforce and unsure about its purpose or how to utilize it effectively? If you have responded affirmatively to these inquiries, you have arrived at the appropriate resource. This module is designed specifically for you.

Welcome to **Salesforce**! Salesforce represents transformative technology equipped with a multitude of productivity-enhancing features intended to facilitate more intelligent and expedient sales processes. As you progress towards earning your badge for this module, we will guide you through these features and address the fundamental question, *“What is Salesforce?”*

### What Is Salesforce?

Salesforce serves as a comprehensive customer success platform, engineered to assist you in selling, servicing, marketing, analyzing, and connecting with your customers.

This platform provides all the tools necessary to operate your business from any location. Utilizing its standard products and features, you can effectively manage relationships with prospects and clients, collaborate with employees and partners, and securely store your data in the cloud.

### Creating a Developer Account in Salesforce

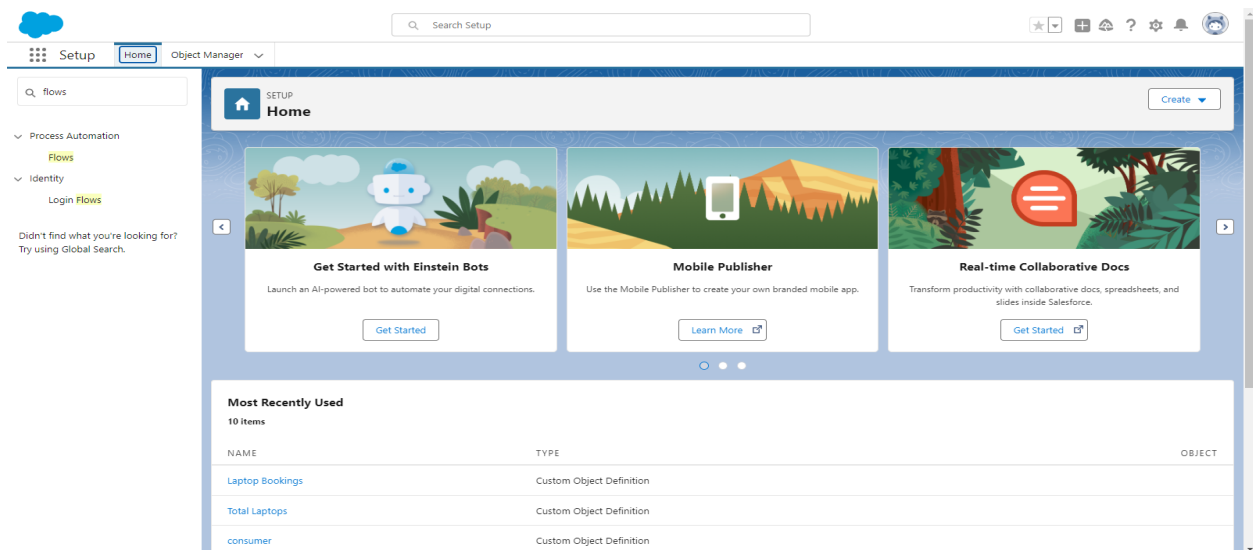
To create a developer organization in Salesforce, follow these steps:

1. Navigate to <https://developer.salesforce.com/signup>.

2. Complete the sign-up form with the following details:
  - a. **First Name** and **Last Name**
  - b. **Email Address**
  - c. **Role:** Developer
  - d. **Company:** College Name
  - e. **Country:** India
  - f. **Postal Code:** Enter your pin code
  - g. **Username:** This should be a combination of your name and company. It does not need to be a valid email address; you can use a format like `username@organization.com`.
3. After entering the required information, click on the "Sign Me Up" button to complete the registration process.

## Account Activation

1. Check the inbox of the email address you used during the sign-up process. Look for an email from Salesforce and click on the **Verify Account** link to activate your account. Note that the email might take 5-10 minutes to arrive.
2. Click on **Verify Account** within the email.
3. Set a password and answer a security question, then click on **Change Password** to finalize your setup.
4. You will be redirected to your Salesforce setup page, where you can begin using your new developer account.



# Object Creation in Salesforce

## What Is an Object?

In Salesforce, objects are analogous to databases that allow you to store data pertinent to your organization. There are two primary types of Salesforce objects:

- **Standard Objects:** These are pre-defined objects provided by Salesforce, such as Users, Contracts, Reports, and Dashboards.
- **Custom Objects:** These are objects created by users to capture information specific to their organization. Custom objects are essential for tailoring Salesforce to meet unique business needs and provide a structure for data sharing.

## Navigating to the Setup Page:

1. Click on the gear icon (⚙️) in the top right corner of the Salesforce interface.
2. Select **Setup** from the dropdown menu.

## To Create a Custom Object:

1. From the Setup page, navigate to **Object Manager**.
2. Click on **Create** and then select **Custom Object**.

## On the Custom Object Definition Page:

1. Enter the **Label Name** and **Plural Label Name**.
2. Check the options for **Allow Reports** and **Allow Search**.
3. Click **Save** to create the custom object.

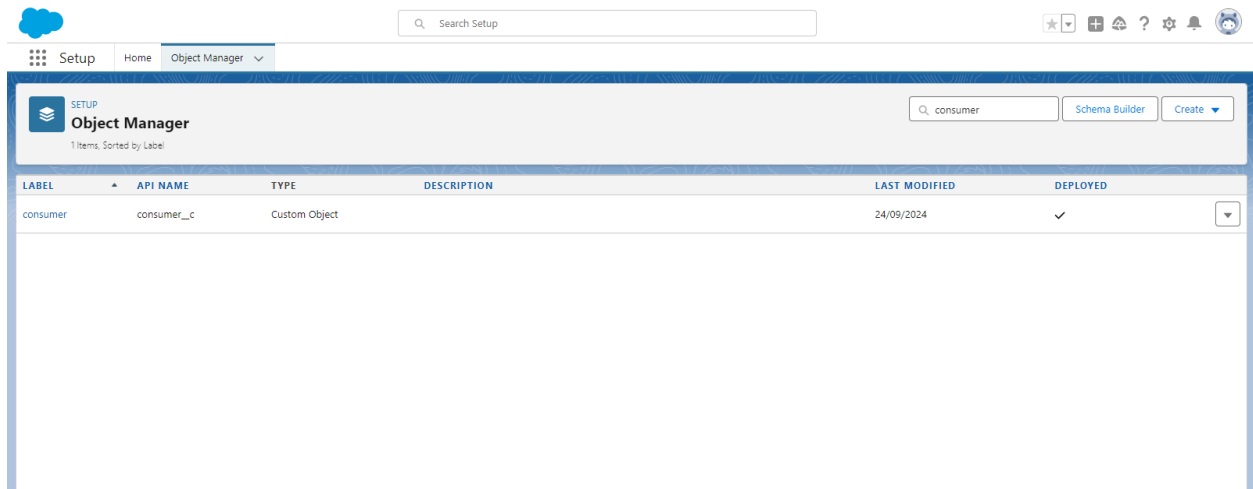
## Creating the Consumer Object:

1. From the Setup page, navigate to **Object Manager**.
2. Click on **Create** and select **Custom Object**.

## On the Custom Object Definition Page:

1. Enter the **Label Name:** Consumer.
2. Enter the **Plural Label Name:** Consumers.
3. Configure the **Record Name:**

- a. **Record Name:** consumer\_name
  - b. **Data Type:** Name
4. Check the options for:
  - a. **Allow Reports**
  - b. **Allow Search**
  - c. **Track Field History**
5. Click **Save** to create the object.



### Creating the Total Laptops Object:

1. From the Setup page, navigate to **Object Manager**.
2. Click on **Create** and select **Custom Object**.

### On the Custom Object Definition Page:

1. Enter the **Label Name:** Total Laptops.
2. Enter the **Plural Label Name:** Total Laptops.
3. Configure the **Record Name:**
  - a. **Record Name:** Total Laptops
  - b. **Data Type:** Text
4. Check the options for:
  - a. **Allow Reports**
  - b. **Allow Search**



c. **Track Field History**

5. Click **Save** to create the object

## Creating the Laptop Bookings Object:

1. From the Setup page, navigate to **Object Manager**.
2. Click on **Create** and select **Custom Object**.

## On the Custom Object Definition Page:

1. Enter the **Label Name**: Laptop Bookings.
2. Enter the **Plural Label Name**: Laptop Bookings.
3. Configure the **Record Name**:
  - a. **Record Name**: Laptop Bookings
  - b. **Data Type**: Name
4. Check the options for:
  - a. **Allow Reports**
  - b. **Allow Search**
  - c. **Track Field History**
5. Click **Save** to create the object.

The screenshot shows the Salesforce Setup page with the 'Object Manager' tab selected. The 'Object Manager' page displays a list of custom objects. The table below shows the objects 'Laptop Bookings' and 'Total Laptops'.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Laptop Bookings	Laptop_Bookings__c	Custom Object		26/09/2024	✓
Total Laptops	Total_Laptops__c	Custom Object		26/09/2024	✓

# Tabs in Salesforce

A tab in Salesforce serves as a user interface element that enables users to build and view records for objects. It provides a way to interact with different types of data and functionality within the Salesforce platform.

## Types of Tabs:

### 1. Custom Tabs

- a. **Purpose:** Used for custom objects that you create in Salesforce. They function similarly to standard Salesforce tabs, such as Accounts, Contacts, and Opportunities.
- b. **Functionality:** Provides a user interface for custom applications and objects.

### 2. Web Tabs

- a. **Purpose:** Display web content or applications embedded within the Salesforce interface.
- b. **Functionality:** Allows users to quickly access frequently used content and applications without leaving Salesforce.

### 3. Visualforce Tabs

- a. **Purpose:** Display custom Visualforce pages.
- b. **Functionality:** Visualforce tabs behave like standard Salesforce tabs but are designed to present custom Visualforce content.

### 4. Lightning Component Tabs

- a. **Purpose:** Allow you to add Lightning components to the navigation menu in Lightning Experience and the Salesforce mobile app.
- b. **Functionality:** Provides a way to incorporate custom Lightning components into the user interface.

### 5. Lightning Page Tabs

- a. **Purpose:** Add Lightning Pages to the navigation menu in the mobile app.
- b. **Functionality:** Unlike other tabs, Lightning Page tabs do not appear on the "All Tabs" page or in the "Available Tabs" list when customizing app tabs. They are used specifically for mobile navigation.

## Creating a Custom Tab

To create a custom tab for an object in Salesforce:

1. Go to the **Setup** page.
2. Type **Tabs** in the Quick Find bar and click on **Tabs**.
3. Click on **New** under the **Custom Object Tabs** section.
4. Select the object (e.g., **Total Laptops**).
5. Choose a **Tab Style**.
6. Click **Next**.
7. On the **Add to Profiles** page, keep the default settings and click **Next**.
8. On the **Add to Custom App** page, uncheck the option to include the tab in custom apps. Ensure that the option **Append tab to users' existing personal customizations** is checked.
9. Click **Save** to create the tab.

## Activity 2: Creating Remaining Tabs

Repeat the steps outlined above to create tabs for the remaining objects:

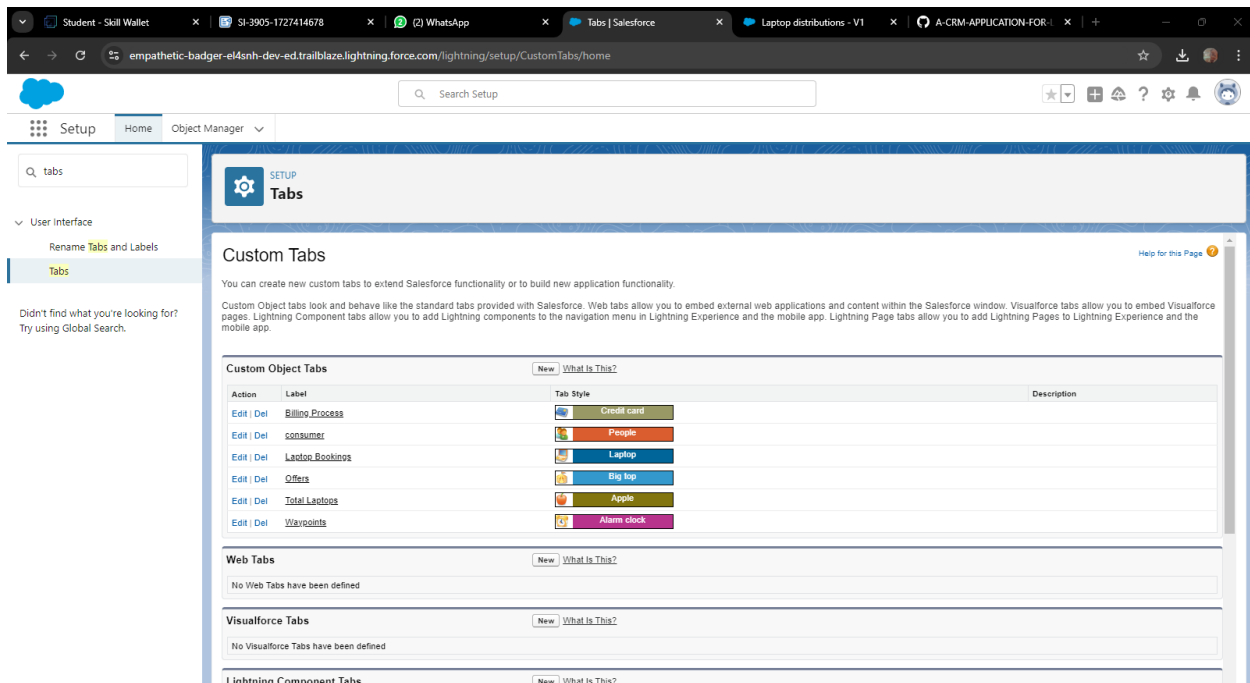
- **Consumer**
- **Laptop Bookings**
- **Billing Process**

Follow the same procedure:

1. Go to the **Setup** page.
2. Type **Tabs** in the Quick Find bar and click on **Tabs**.
3. Click on **New** under the **Custom Object Tabs** section.
4. Select the respective object.
5. Choose a **Tab Style**.
6. Click **Next**.
7. On the **Add to Profiles** page, keep the default settings and click **Next**.
8. On the **Add to Custom App** page, uncheck the option to include the tab in custom apps. Ensure that the option **Append tab to users' existing personal customizations** is

checked.

9. Click **Save** to create each tab.



## The Lightning App

An app in Salesforce is a collection of items that work together to serve a specific function. In Lightning Experience, Lightning apps provide users with access to sets of objects, tabs, and other components, all conveniently bundled in the navigation bar.

### Key Features of Lightning Apps:

1. **Unified Access:** Users can access a set of related objects, tabs, and other items in one place, improving efficiency and ease of use.
2. **Customization:** Lightning apps allow for branding with custom colors and logos, giving a unique identity to each app.
3. **Utility Bar:** You can include a utility bar in your Lightning app, offering quick access to useful tools and features.
4. **Lightning Page Tabs:** Incorporate Lightning Page tabs for a seamless navigation experience within the app.
5. **Efficient Switching:** Users can switch between different apps effortlessly, enabling them

to work more efficiently and manage various tasks within the organization.

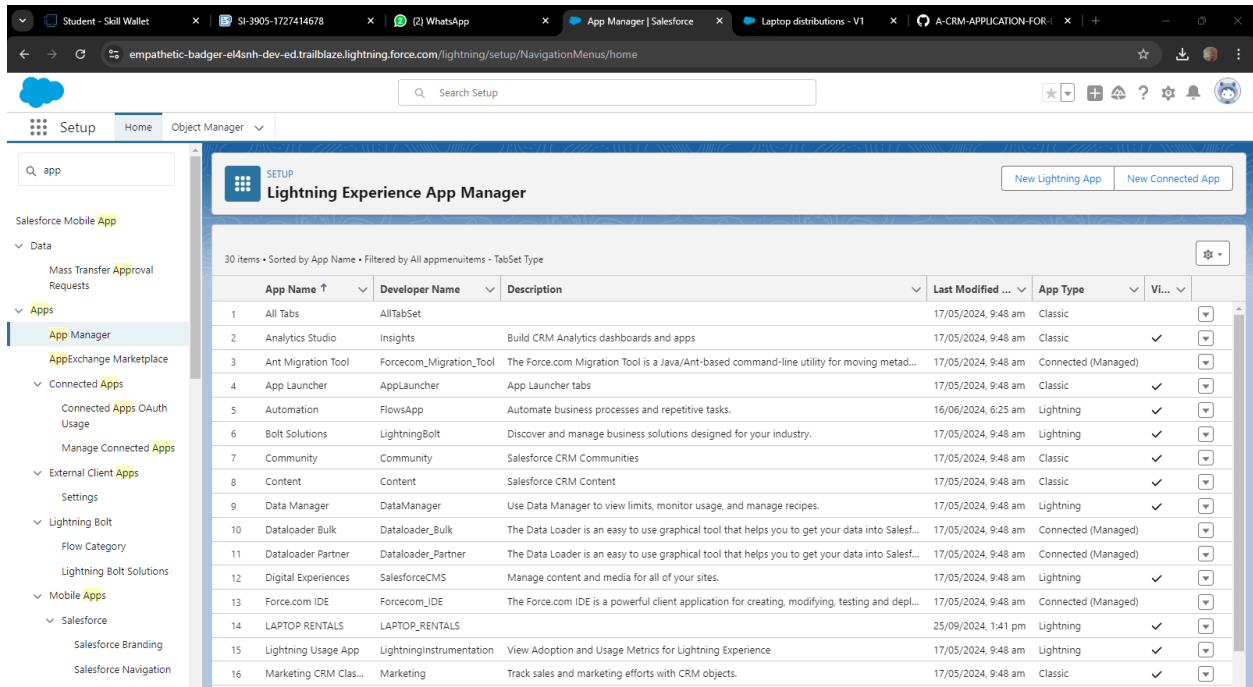
## Creating a Lightning App

To create a Lightning app page in Salesforce, follow these steps:

1. Go to the **Setup** page.
2. In the Quick Find bar, search for **App Manager** and select **App Manager**.
3. Click on **New Lightning App**.

### App Creation Steps:

1. **App Details:**
  - a. Enter the **App Name**: LAPTOP RENTALS.
  - b. Click **Next**.
2. **App Options Page:**
  - a. Keep the default settings.
  - b. Click **Next**.
3. **Utility Items:**
  - a. Keep the default settings.
  - b. Click **Next**.
4. **App Branding:**
  - a. Upload a photo related to your app for branding.
5. **Add Navigation Items:**
  - a. Select the items **Total Laptops**, **Consumer**, **Laptop Booking**, and **Billing Process** from the search bar.
  - b. Use the arrow button to move the selected items to the navigation.
  - c. Click **Next**.
6. **Add User Profiles:**
  - a. In the search bar, search for the profile **System Administrator**.
  - b. Click on the arrow button to add the profile to the app.
  - c. Click **Save & Finish**.



# Fields in Salesforce

In Salesforce, fields represent the data stored in the columns of a relational database. They hold valuable information required for specific objects, making the processes of searching, deleting, and editing records simpler and quicker.

## Types of Fields:

1. Standard Fields
2. Custom Fields

## Standard Fields:

Standard Fields are predefined fields in Salesforce that perform specific tasks. These fields cannot be deleted if they are required; however, non-required standard fields can be deleted by users. Common standard fields found in every Salesforce application include:

- Created By
- Owner
- Last Modified
- Field Made During Object Creation

## Custom Fields:

Custom Fields are highly flexible and can be modified according to user requirements. Unlike standard fields, custom fields are not mandatory and can be added or removed as needed by the organization. This flexibility allows each organization to tailor its data structure to its specific needs.

By understanding and utilizing both standard and custom fields, users can effectively manage and organize their data in Salesforce.

## Creating Fields in the Consumer Object

To create fields in the Consumer object, follow these steps:

### 1. Phone Number Field:

- a. Go to **Setup**.
- b. Click on **Object Manager**.
- c. Type **Consumer** in the search bar and click on the object.
- d. Click on **Fields & Relationships**.
- e. Click on **New**.
- f. Select **Data Type** as **Phone** and click **Next**.
- g. Fill in the details:
  - i. **Field Label:** Phone Number
  - ii. **Field Name:** (auto-generated)
- h. Check the **Required** option checkbox.
- i. Click **Next** >> **Next** >> **Save & New**.

### 2. Email Field:

- a. Go to **Setup**.
- b. Click on **Object Manager**.
- c. Type **Consumer** in the search bar and click on the object.
- d. Click on **Fields & Relationships**.
- e. Click on **New**.
- f. Select **Data Type** as **Email** and click **Next**.
- g. Fill in the details:

- i. **Field Label:** Email
  - ii. **Field Name:** (auto-generated)
- h. Click **Next** >> **Next** >> **Save & New**.

### 3. Address Field:

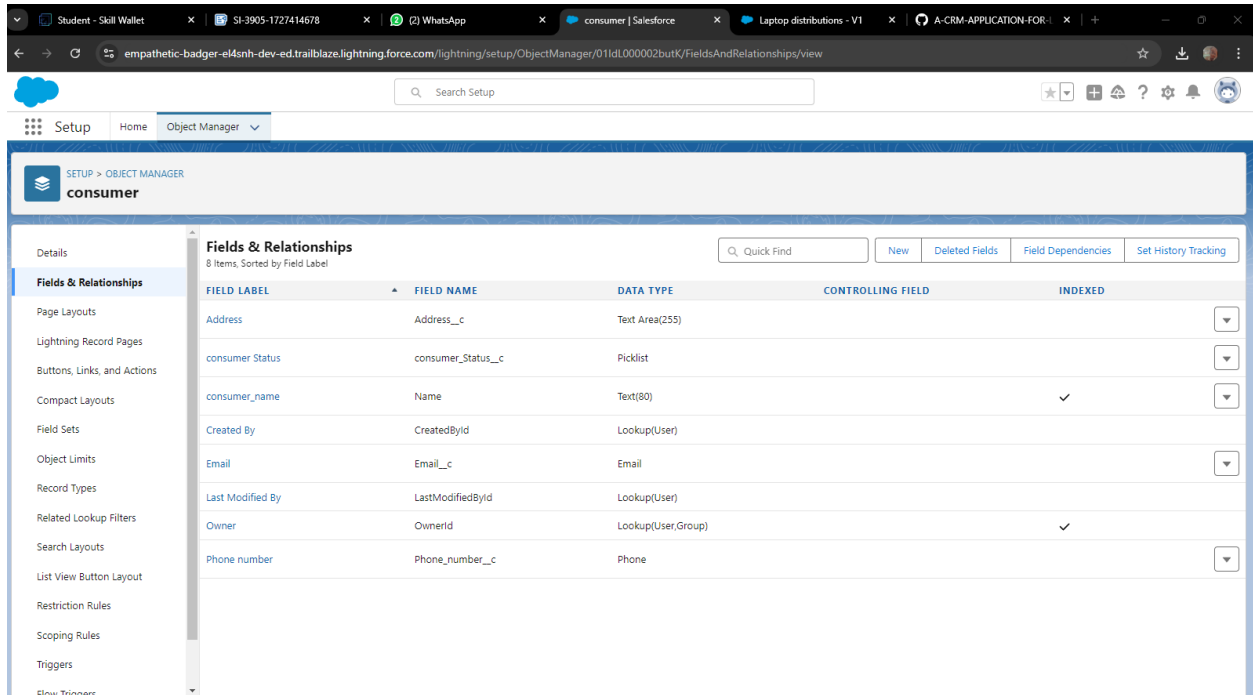
- a. Go to **Setup**.
- b. Click on **Object Manager**.
- c. Type **Consumer** in the search bar and click on the object.
- d. Click on **Fields & Relationships**.
- e. Click on **New**.
- f. Select **Data Type** as **Text Area** and click **Next**.
- g. Fill in the details:
  - i. **Field Label:** Address
  - ii. **Field Name:** (auto-generated)
- h. Check the **Required** option checkbox.
- i. Click **Next** >> **Next** >> **Save & New**.

### 4. Consumer Status Field:

- a. Go to **Setup**.
- b. Click on **Object Manager**.
- c. Type **Consumer** in the search bar and click on the object.
- d. Click on **Fields & Relationships**.
- e. Click on **New**.
- f. Select **Data Type** as **Picklist** and click **Next**.
- g. Fill in the details:
  - i. **Field Label:** Consumer Status
  - ii. **Values:** Enter values with each value separated by a new line:
    - 1. Student
    - 2. Employee
    - 3. Others



- iii. **Field Name:** (auto-generated)
- h. Check the **Required** option checkbox.
- i. Click **Next >> Next >> Save & New.**



## Creating Fields in the Laptop Bookings Object

To create fields in the Laptop Bookings object, follow these steps:

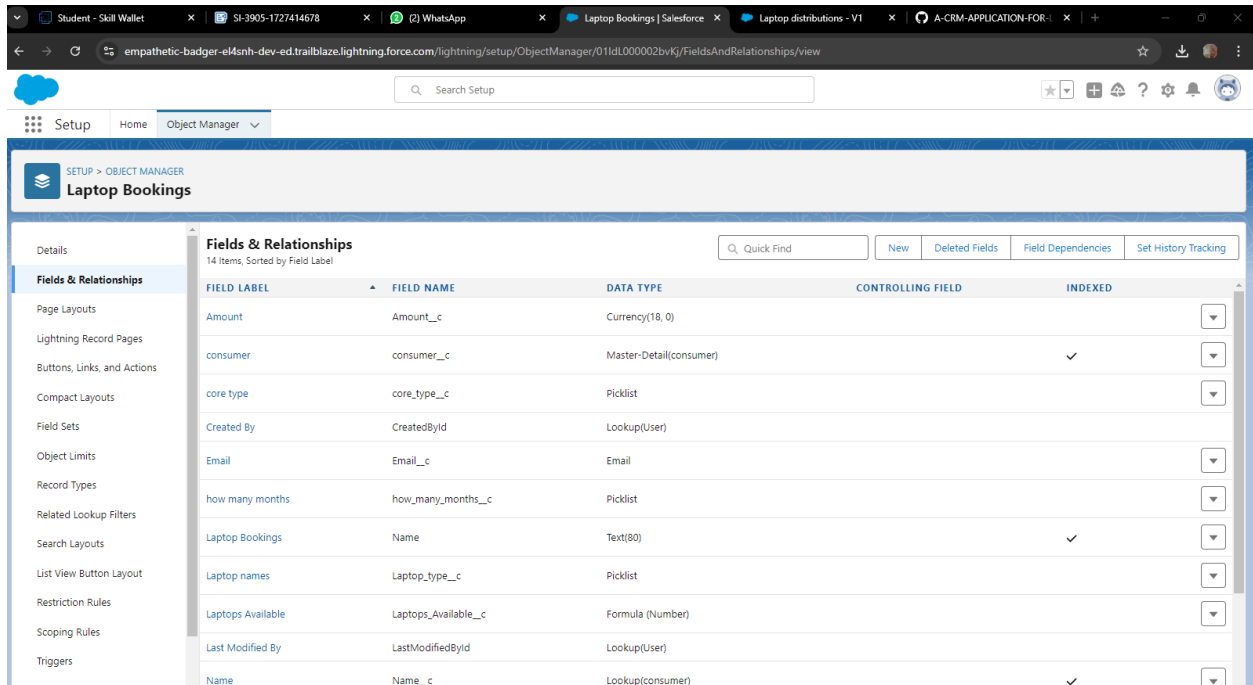
### 1. Laptop Model Field:

- Go to **Setup**.
- Click on **Object Manager**.
- Type **Laptop Booking** in the search bar and click on the object.
- Click on **Fields & Relationships**.
- Click on **New**.
- Select **Data Type** as **Picklist**.
- Fill in the details:
  - **Field Label:** Laptop Model
  - **Picklist Values:**

- Dell
  - Acer
  - HP
  - Mac
- Check the **Required** option checkbox.
- Click **Next** >> **Next** >> **Save & New**.

## 2. Processor Type Field:

- Go to **Setup**.
- Click on **Object Manager**.
- Type **Laptop Booking** in the search bar and click on the object.
- Click on **Fields & Relationships**.
- Click on **New**.
- Select **Data Type** as **Picklist**.
- Fill in the details:
  - **Field Label:** Processor Type
  - **Picklist Values:**
    - Core i3
    - Core i5
    - Core i7
  - Check the **Required** option checkbox.
- Click **Next** >> **Next** >> **Save & New**.



## Field Dependency:

A field dependency refers to a relationship between two fields on an object where the values of one field determine the available values for another field. This is commonly used for picklist fields, where the options in a dependent picklist are determined by the value selected in a controlling picklist.

### To create a field dependency:

1. Go to **Setup**.
2. Click on **Object Manager**.
3. Type **Laptop Booking** in the search bar and click on the object.
4. Click on **Fields & Relationships**.
5. Click on **Field Dependencies**.
6. Click **New**.
7. Select the **Controlling Field** (e.g., Laptop Model) and the **Dependent Field** (e.g., Processor Type).
8. Define the dependency by selecting the values that correspond between the controlling and dependent fields.
9. Click **Save**.

By using field dependencies, you can ensure that the values available in one picklist are dynamically filtered based on the selected value in another picklist. This enhances data accuracy and user experience.

## Creating Field Dependencies in the Laptop Bookings Object

To create fields and relationships in the Laptop Bookings object with field dependencies, follow these steps:

1. **Go to Setup:**
  - a. Click on **Object Manager**.
  - b. Type **Laptop Booking** in the search bar and click on the object.
2. **Create Fields for Laptop Model and Processor Type:**
  - a. Follow the steps outlined previously to create the **Laptop Model** and **Processor Type** fields as picklists.
3. **Create Field Dependency:**
  - a. Go to **Setup**.
  - b. Click on **Object Manager**.
  - c. Type **Laptop Booking** in the search bar and click on the object.
  - d. Click on **Fields & Relationships**.
  - e. Click on **Field Dependencies**.
  - f. Click **New**.
  - g. Select the **Controlling Field** as **Laptop Model** and the **Dependent Field** as **Processor Type**.
  - h. Click **Continue**.
4. **Define the Field Dependency:**
  - a. You will see a matrix with the controlling field values on the left and the dependent field values on the top.
  - b. Select the values for each laptop model that should correspond to the processor types:
    - i. **For Dell:** Check the boxes for Core i3, Core i5, Core i7.
    - ii. **For Acer:** Check the boxes for Core i3, Core i5, Core i7.

- iii. **For HP:** Check the boxes for Core i3, Core i5, Core i7.
- iv. **For Mac:** Check the box for Bionic Chip.

5. **Save the Field Dependency:**

- a. Click **Save** to create the field dependency.

This setup ensures that the available processor types are dynamically filtered based on the selected laptop model, improving data accuracy and user experience.

## **Creating a Validation Rule for the Phone Number Field in the Consumer Object**

1. **Go to Setup:**

- Click on **Object Manager**.
- Type **Consumer** in the search bar and click on the object.

2. **Create the Validation Rule:**

- Click on **Validation Rules**.
- Click **New**.

3. **Enter the Details for the Validation Rule:**

- **Rule Name:** `PhoneNumberOrEmailBlankRule`
- **Description:** Phone number and email should not be blank.

4. **Enter the Formula:**

- In the formula editor, enter the following formula:

```
plaintext
```

```
Copy code
```

```
OR ( ISBLANK ( Phone_Number__c ) , ISBLANK ( Email__c ) )
```

- Click **Check Syntax** to ensure there are no errors in the formula.

5. **Enter the Error Message:**

- **Error Message:** Either Phone Number or Email must be filled out.
- **Error Location:** Select **Field** and choose the `Phone_Number__c` field.

6. **Save the Validation Rule:**

- Click **Save** to create the validation rule.

This validation rule will ensure that users cannot save a Consumer record if both the phone number and email fields are left blank. An error message will guide users to fill out at least one of these fields before saving the record.

## **Profiles in Salesforce**

A profile in Salesforce is a collection of settings and permissions that determine what a user can do within the platform. Profiles control various aspects such as object permissions, field permissions, user permissions, tab settings, app settings, Apex class access, Visualforce page access, page layouts, record types, login hours, and login IP ranges. Profiles are typically defined based on the user's job function, such as System Administrator, Developer, or Sales Representative.

### **Types of Profiles in Salesforce**

#### **Standard Profiles**

- **Contract Manager**
- **Read Only**
- **Marketing User**
- **Solutions Manager**

- **Standard User**
- **System Administrator**

Standard profiles come with a predefined set of permissions for all the standard objects available on the platform. These profiles cannot be deleted.

**Custom Profiles** Custom profiles are defined by users to meet specific business requirements. Unlike standard profiles, custom profiles can be deleted if no users are assigned to them. Custom profiles offer the flexibility to tailor permissions and settings according to the unique needs of an organization.

**Summary** Profiles in Salesforce play a crucial role in managing user access and permissions. They help ensure that users have the appropriate level of access to perform their job functions while maintaining data security and integrity. Understanding and configuring profiles effectively is essential for successful Salesforce administration.

### **Creating a New Profile: Owner**

To create a new profile named "Owner" based on the Standard User profile, follow these steps:

1. **Go to Setup:**
  - a. Click on **Profiles** in the Quick Find box.
2. **Clone the Standard User Profile:**
  - a. Find and click on the **Standard User** profile.
  - b. Click **Clone** to create a new profile based on the Standard User.
3. **Enter Profile Details:**
  - a. **Profile Name:** Enter `Owner`.
  - b. Click **Save**.
4. **Configure Custom Object Permissions:**
  - a. Scroll down to **Custom Object Permissions**.
  - b. Grant access permissions for the following objects:
    - i. Total Laptops
    - ii. Consumers
    - iii. Laptop Booking
    - iv. Billing Process
5. **Ensure that the required access permissions (Read, Create, Edit, Delete) are checked for each object as per your requirements.**

## 6. **Save the Profile:**

- a. Click **Save** to apply the changes and finalize the creation of the Owner profile.

This new Owner profile will inherit the permissions of the Standard User profile but can be customized further to meet specific needs, including adjusting access to various custom objects.

## **Creating a New Profile: Agent**

To create a new profile named "Agent" based on the Standard Platform User profile, follow these steps:

### 1. **Go to Setup:**

- a. Type **Profiles** in the Quick Find box and select **Profiles**.

### 2. **Clone the Standard Platform User Profile:**

- a. Find and click on the **Standard Platform User** profile.
- b. Click **Clone** to create a new profile based on the Standard Platform User.

### 3. **Enter Profile Details:**

- a. **Profile Name:** Enter `Agent`.
- b. Click **Save**.

### 4. **Edit the Profile:**

- a. On the profile page, click **Edit**.

### 5. **Configure Custom Object Permissions:**

- a. Scroll down to **Custom Object Permissions**.
- b. Grant access permissions for the following objects:
  - i. Total Laptops
  - ii. Consumer
  - iii. Laptop Bookings
  - iv. Billing Process

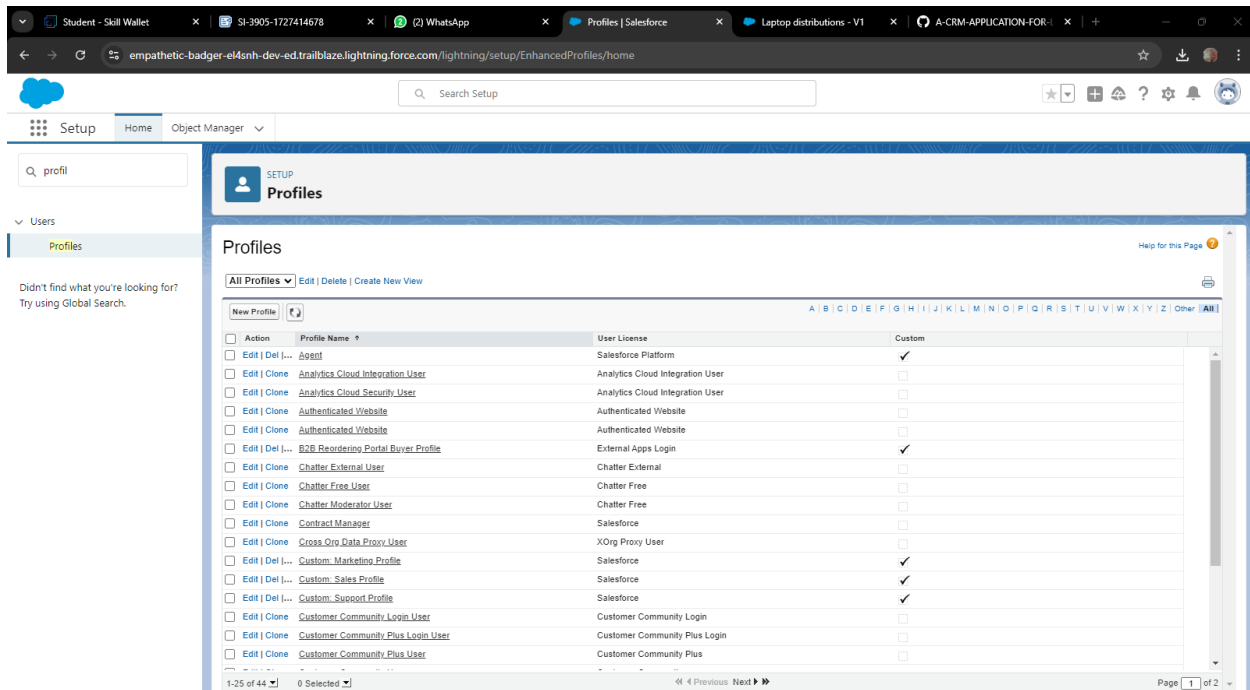
### 6. **Ensure the appropriate permissions (Read, Create, Edit, Delete) are checked for each object as required.**

### 7. **Save the Profile:**

- a. Click **Save** to apply the changes and finalize the creation of the Agent profile.

This new Agent profile will inherit the permissions of the Standard Platform User profile but can be customized further to provide specific access to the mentioned custom objects.





# Roles and Hierarchy

In Salesforce, roles help define a user's position within the organization's hierarchy, which impacts data visibility and access. Roles are used in conjunction with profiles to control the access and permissions a user has.

## Creating Roles in Salesforce

### Creating the Owner Role

1. **Go to Setup:**
  - a. In the Quick Find box, type **Roles** and select **Set Up Roles**.
2. **Expand All:**
  - a. Click **Expand All** to view the current role hierarchy.
3. **Add Role:**
  - a. Click **Add Role** under the appropriate parent role (e.g., **CEO**).
4. **Enter Role Details:**
  - a. **Label:** Enter `Owner`.
  - b. **Role Name:** This is auto-populated based on the Label.
5. **Save:**
  - a. Click **Save** to create the Owner role.

## Creating Agent Roles

### 1. Go to Setup:

- In the Quick Find box, type **Roles** and select **Set Up Roles**.

### 2. Add Role Under Owner:

- Click the plus icon next to the **Owner** role to add a new role under it.

### 3. Enter Role Details:

- Label:** Enter Agent.
- Role Name:** This is auto-populated based on the Label.

### 4. Save:

- Click **Save** to create the Agent role.

## Hierarchy Setup:

- The Owner role is positioned at a higher level in the hierarchy.

Agent roles are created beneath the Owner role, which allows for structured data visibility and access, ensuring that users with higher roles can view and manage data for users in lower roles.

The screenshot displays the Salesforce Setup interface for the 'Roles' section. The left sidebar shows the navigation menu with 'Setup' selected. The main content area is titled 'Understanding Roles' and provides instructions on setting up a role hierarchy. A 'Sample Role Hierarchy' is shown, starting with 'Executive Staff' (CEO, President, CFO, VP, Sales) at the top. Below this are three regional sales director roles: 'Western Sales Director', 'Eastern Sales Director', and 'International Sales Director'. Each director role has associated sales representative roles: 'CA Sales Rep', 'OR Sales Rep' for Western; 'NY Sales Rep', 'WA Sales Rep' for Eastern; and 'Asian Sales Rep', 'European Sales Rep' for International. Arrows indicate the reporting structure from reps to directors and from directors to executive staff. A 'Set Up Roles' button is visible at the bottom right of the diagram.

## Users

In Salesforce, users are individuals who can access and interact with the platform based on their roles, profiles, and permissions. Setting up users correctly ensures that they have the appropriate

access to perform their tasks efficiently.

## **Creating Users in Salesforce**

### **Creating a User with the Owner Role**

1. **Go to Setup:**
  - a. In the Quick Find box, type **Users** and select **Users**.
2. **Create a New User:**
  - a. Click **New User**.
3. **Fill in the User Details:**
  - a. **First Name:** Vicky
  - b. **Last Name:** Y
  - c. **Alias:** (Provide a suitable alias)
  - d. **Email:** (Enter your personal email address)
  - e. **Username:** (Format: text@text.text)
  - f. **Nickname:** (Provide a suitable nickname)
  - g. **Role:** Owner
  - h. **User License:** Salesforce
  - i. **Profile:** Owner
4. **Save:**
  - a. Click **Save** to create the user.

### **Creating Another User with the Agent Role**

1. **Go to Setup:**
  - a. In the Quick Find box, type **Users** and select **Users**.
2. **Create a New User:**
  - a. Click **New User**.
3. **Fill in the User Details:**
  - a. **First Name:** Ram
  - b. **Last Name:** Ram
  - c. **Alias:** (Provide a suitable alias)
  - d. **Email:** (Enter your personal email address)

- e. **Username:** (Format: text@text.text)
  - f. **Nickname:** (Provide a suitable nickname)
  - g. **Role:** Agent
  - h. **User License:** Salesforce Platform
  - i. **Profile:** Standard Platform User
4. **Save:**
- a. Click **Save** to create the user.

### Summary:

- **Vicky** is set up with the **Owner** role, **Salesforce** license, and **Owner** profile.
- **Ram** is set up with the **Agent** role, **Salesforce Platform** license, and **Standard Platform User** profile.

This configuration ensures that users have the necessary roles and permissions to access and manage data according to their responsibilities within the organization.

The screenshot shows the Salesforce Setup page for Users. The left sidebar contains navigation links for Setup, Home, Object Manager, and a search bar. The main content area is titled 'All Users' and includes a table of users. The table has columns for Action, Full Name, Alias, Username, Role, Active, and Profile. The users listed are:

Action	Full Name	Alias	Username	Role	Active	Profile
[Edit]	Brown, Courtney	cbrow	harishkurmana1103@usource@gmail.com	Manager	✓	Standard Platform User
[Edit]	Chatter Expert	Chatter	chath.00dd000001udaxuae.ds@usource@chatter.salesforce.com		✓	Chatter Free User
[Edit]	Kurmana, Harish	Hkum	harishkurmana1103@empathetic-badger-el4snh.com		✓	System Administrator
[Edit]	ram, ram	rm	harishkurmana1103@gmail.com	Agent	✓	Standard Platform User
[Edit]	User Integration	intero	integration@00dd000001udaxuae.com		✓	Analytics Cloud Integration User
[Edit]	User Security	sas	insightsec@00dd000001udaxuae.com		✓	Analytics Cloud Security User
[Edit]	Wheeler, Allison	aythex	harishkurmana1103@usource@gmail.com	VP, North American Sales	✓	Standard Platform User
[Edit]	y_vicky	yy	harishkurmana1103@gmail.com		✓	owner

## Flows in Salesforce

Salesforce Flows are a versatile automation tool that allows you to streamline and automate complex business processes. They are designed to work without code, using a visual interface to help you create, manage, and deploy processes efficiently.

## Types of Flows in Salesforce

### 1. Screen Flows:

- a. **Purpose:** Guide users through a series of screens to collect or display information.
- b. **Usage:** Often used for data entry, updates, and user interactions.

### 2. Autolaunched Flows:

- a. **Purpose:** Triggered by events like record creation or updates without user interaction.
- b. **Usage:** Ideal for background automation tasks.

### 3. Scheduled Flows:

- a. **Purpose:** Run at specific times or intervals.
- b. **Usage:** Automate recurring tasks and processes.

### 4. Record-Triggered Flows:

- a. **Purpose:** Triggered when records meet specified criteria.
- b. **Usage:** Automate record updates and related actions based on changes.

### 5. Flow Builder:

- a. **Purpose:** The visual interface for creating flows.
- b. **Usage:** Allows you to design flows with a drag-and-drop approach, adding elements like screens, logic, and actions.

### 6. Flow Templates:

- a. **Purpose:** Pre-built flow templates provided by Salesforce.
- b. **Usage:** Offer a starting point for creating flows, covering various use cases.

### 7. Flow Elements:

- a. **Purpose:** Components used to build flows.
- b. **Usage:** Include variables, decisions, loops, and other elements to build sophisticated logic.

### 8. Subflows:

- a. **Purpose:** Reusable flow elements.
- b. **Usage:** Simplify management and maintenance by incorporating reusable processes into multiple flows.

## Creating a Flow to Automatically Populate Amount Field

**Scenario:** You want to create a flow to automatically populate the `Amount` field based on the selected `Laptop Model` in the `Laptop Bookings` object.

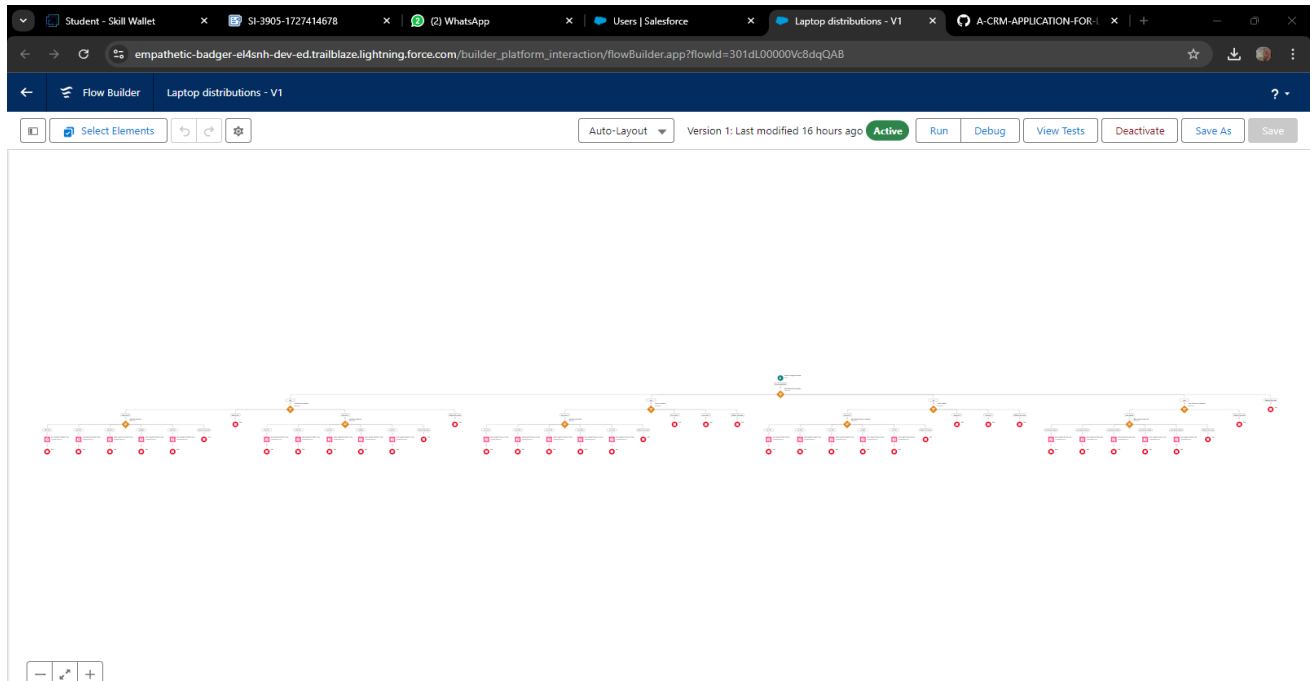
### Steps to Create the Flow:

1. **Go to Setup:**
  - a. In the Quick Find box, type **Flows** and select **Flows**.
2. **Create a New Flow:**
  - a. Click **New Flow**.
3. **Choose Flow Type:**
  - a. Select **Record-Triggered Flow** because you want the flow to run when a record is created or updated.
4. **Configure Trigger:**
  - a. **Object:** Choose `Laptop Bookings`.
  - b. **Trigger the Flow When:** Select **A record is created or updated**.
5. **Add Elements to Flow:**
  - a. **Get Records:** Retrieve the selected `Laptop Model` from the `Laptop Bookings` record.
  - b. **Decision Element:** Determine which `Laptop Model` is selected.
  - c. **Assignment Element:** Assign the corresponding amount based on the `Laptop Model` to the `Amount` field.
6. **Define Field Values:**
  - a. **For Dell:** Set the amount to \$X.
  - b. **For Acer:** Set the amount to \$Y.
  - c. **For HP:** Set the amount to \$Z.
  - d. **For Mac:** Set the amount to \$W.
7. **Update Records:**
  - a. Add an **Update Records** element to update the `Amount` field with the assigned value.
8. **Save and Activate the Flow:**
  - a. Click **Save** and provide a name for the flow.

b. Click **Activate** to deploy the flow.

**Summary:** This flow will automatically populate the Amount field in the Laptop Bookings object based on the selected Laptop Model, ensuring data accuracy and reducing manual input.

Feel free to adjust the flow elements and configurations based on your specific business needs.



## Apex

Apex is a powerful, strongly typed, object-oriented programming language used to add custom business logic to Salesforce applications. It operates similarly to Java and is designed for writing custom code that runs on the Salesforce platform.

### Key Features of Apex:

- **Object-Oriented:** Supports classes, objects, and methods.
- **Database Integration:** Executes DML operations (Insert, Update, Delete) and handles transactions.
- **Events:** Triggers can be executed before or after data manipulation.
- **Custom Logic:** Allows for automation and customization beyond standard Salesforce capabilities.

### Creating Apex Classes

## Apex Class:

- **Definition:** A class in Apex serves as a blueprint from which objects are created. It encapsulates data and methods.

## Steps to Create an Apex Class:

1. **Log In to Salesforce:**
  - a. Access your Salesforce account and open the Developer Console.
2. **Create a New Class:**
  - a. In the Developer Console, go to **File > New > Apex Class**.
3. **Enter Class Details:**
  - a. Provide a name for the class.
  - b. Write the class code as required.
4. **Save:**
  - a. Click **Save** to create the class.

## Example Apex Class Code:

apex

Copy code

```
public class MyClass {  
  
    public static void myMethod() {  
  
        // Method logic here  
  
    }  
  
}
```

## Access Specifiers in Apex

1. **Private:**
  - a. **Definition:** The default access level. Accessible only within the defining class.
  - b. **Usage:** Use for internal class methods and variables.
2. **Protected:**
  - a. **Definition:** Accessible by the class itself, inner classes, and subclasses.
  - b. **Usage:** Suitable for instance methods and member variables that need to be



accessed by subclasses.

3. **Public:**

- a. **Definition:** Accessible by any Apex code within the same package or namespace.
- b. **Usage:** Use for methods and variables that need to be accessed by other classes in the same namespace.

4. **Global:**

- a. **Definition:** Accessible by all Apex code in the Salesforce organization.
- b. **Usage:** Required for methods and variables accessed via API or from other namespaces.

## Creating Apex Triggers

### Trigger:

- **Definition:** Apex code that runs before or after DML operations (Insert, Update, Delete) on Salesforce objects.

### Types of Triggers:

- **Before Triggers:** Run before changes are saved to the database. Useful for validation or modification before committing.
- **After Triggers:** Run after changes are committed to the database. Useful for updating related records or sending notifications.

### Steps to Create a New Trigger:

1. **Log In to Salesforce:**
  - a. Open the Developer Console.
2. **Create a New Trigger:**
  - a. Go to **File > New > Trigger**.
3. **Enter Trigger Details:**
  - a. Provide a name for the trigger and select the object it will be associated with.
4. **Write Trigger Code:**
  - a. Define the trigger logic using Apex code.
5. **Save:**
  - a. Click **Save** to create the trigger.

### Example Trigger Code:

apex

Copy code

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

## Creating a Handler Class

### Handler Class:

- **Definition:** A class that contains logic executed by triggers. Keeps trigger logic organized and manageable.

### Example Handler Class Code:

apex

Copy code

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

```
body += 'Core Type = ' + lap.core__c + ' \n';

body += 'Laptop Type = ' + lap.Laptop_type__c;

email.setPlainTextBody(body);

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

}

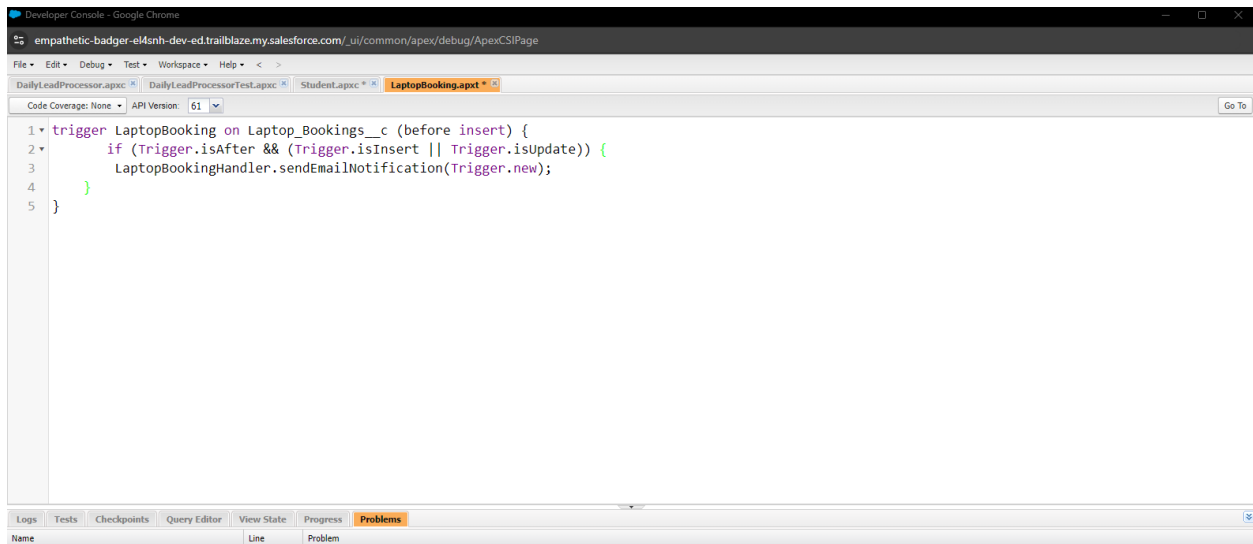
}

}
```

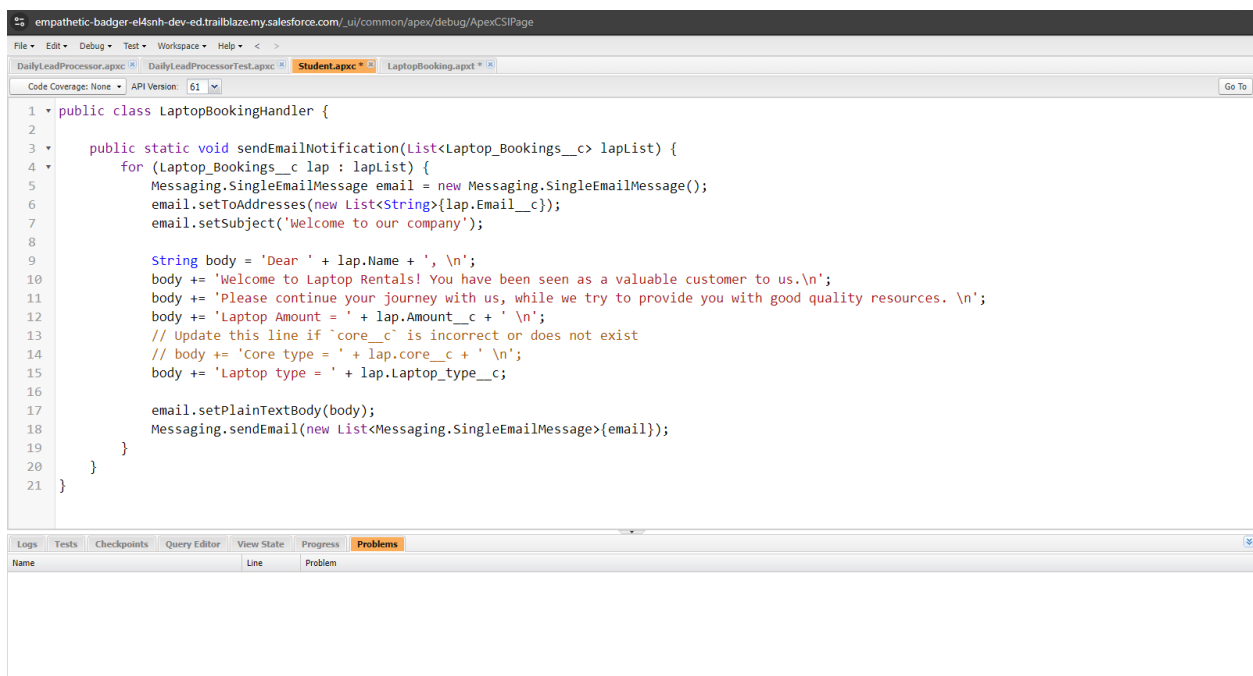
## Testing and Deploying

1. **Test:** Before deploying, ensure you have test records in the `Laptop Bookings` object to verify that triggers and classes function correctly.
2. **Deploy:** Once tested, deploy the code to your production environment or sandbox as needed.

This setup helps automate email notifications based on `Laptop Bookings` records, providing a streamlined approach to managing communications.



```
1 trigger LaptopBooking on Laptop_Bookings__c (before insert) {
2   if (Trigger.isAfter && (Trigger.isInsert || Trigger.isUpdate)) {
3     LaptopBookingHandler.sendEmailNotification(Trigger.new);
4   }
5 }
```



```
1 public class LaptopBookingHandler {
2
3   public static void sendEmailNotification(List<Laptop_Bookings__c> lapList) {
4     for (Laptop_Bookings__c lap : lapList) {
5       Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
6       email.setToAddresses(new List<String>{lap.Email__c});
7       email.setSubject('Welcome to our company');
8
9       String body = 'Dear ' + lap.Name + ', \n';
10      body += 'Welcome to Laptop Rentals! You have been seen as a valuable customer to us.\n';
11      body += 'Please continue your journey with us, while we try to provide you with good quality resources. \n';
12      body += 'Laptop Amount = ' + lap.Amount__c + ' \n';
13      // Update this line if 'core__c' is incorrect or does not exist
14      body += 'Core type = ' + lap.core__c + ' \n';
15      body += 'Laptop type = ' + lap.Laptop_type__c;
16
17      email.setPlainTextBody(body);
18      Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
19    }
20  }
21 }
```

# Reports

Reports in Salesforce are essential for analyzing and visualizing your data. They allow you to create customized views of your data, helping you gain insights and make informed decisions. Salesforce offers several types of reports, each suited to different needs.

## Types of Reports

### 1. **Tabular Reports:**

- a. **Description:** Basic listing of data with no groupings or subtotals.
- b. **Use Case:** Best for simple lists or when you need a straightforward view of data.
- c. **Example:** Listing all accounts, contacts, or opportunities.

#### **Key Features:**

- d. Simple, flat view of data.
- e. No grouping or aggregation.

### 2. **Summary Reports:**

- a. **Description:** Provides data with groupings and subtotals based on a specific field.
- b. **Use Case:** Ideal for hierarchically grouped data, such as sales figures broken down by stages or time periods.
- c. **Example:** Opportunities summarized by Sales Stage and Owner.

#### **Key Features:**

- d. Allows grouping of rows.
- e. Provides subtotals and summaries for each group.

### 3. **Matrix Reports:**

- a. **Description:** Allows grouping of records by rows and columns, providing a grid-like view.
- b. **Use Case:** Useful for comparing data across two dimensions, such as time periods and product categories.
- c. **Example:** Opportunities summarized by month (rows) and account (columns).

#### **Key Features:**

- d. Group data both vertically and horizontally.
- e. Provides cross-tabulation of data.

### 4. **Joined Reports:**

- a. **Description:** Combines blocks of related information into a single report.
- b. **Use Case:** Best for displaying data from multiple report types or objects in one view.
- c. **Example:** A report showing opportunity, case, and activity data for accounts in separate blocks.

### **Key Features:**

- d. Supports multiple report blocks with unique columns and filters.
- e. Allows for combining different types of data.

## **Creating Reports**

### **1. Access Reports:**

- a. Navigate to the **Reports** tab in Salesforce.

### **2. Create a New Report:**

- a. Click **New Report**.

### **3. Select Report Type:**

- a. Choose the type of report (Tabular, Summary, Matrix, or Joined) based on your needs.

### **4. Define Report Criteria:**

- a. Set up filters, groupings, and columns to display the desired data.

### **5. Run and Save Report:**

- a. Click **Run Report** to view the data.
- b. Save the report with a descriptive name and any necessary sharing settings.

### **6. Schedule Reports (Optional):**

- a. Set up report scheduling to automatically run and deliver reports at specified intervals.

## **Example Use Cases**

- **Tabular Report:** Generate a list of all leads with their contact information.
- **Summary Report:** Analyze sales performance by region and sales stage.
- **Matrix Report:** Compare opportunities by month and product type.
- **Joined Report:** Combine opportunities, cases, and activities for a comprehensive view of account interactions.

By understanding and utilizing these report types, you can effectively organize, view, and analyze your Salesforce data to meet various business needs.

Report: consumer with Laptop Bookings and Total Laptops  
consumer with laptops and total laptops

Total Records: 18, Total Amount: ₹6,06,450

types of version	consumer: consumer_name	Laptop Bookings: Laptop Bookings	Total Laptops: Total Laptops	Laptop names	core type	Amount	type of version
- (4)	Raju	bhavya	54	Hp	Core i5	₹5,100	high
	Harish	apple	50	Mac	Core i7	-	basic
	Harish	Google	50	Dell	Core i3	₹2,000	high
	Mahesh	Infosys	20	Acer	Core i5	₹20,000	very high
Subtotal						₹27,100	
apple (4)	kamesh	mac2	34	Hp	Core i3	₹20,000	very high
	ramu	apple	54	Dell	Core i5	₹35,000	very high
	ramu	mac	30	Mac	Bionic chip	₹6,800	high
	suresh	harsh	60	Acer	Core i7	₹55,000	very high
Subtotal						₹1,16,800	
Google (3)	suresh	fool	34	Dell	Core i7	₹8,000	high
	kamesh	hari	23	Acer	Core i5	₹20,000	very high
	Harish	Google	50	Dell	Core i3	₹2,000	high
Subtotal						₹30,000	
hexaware (2)	kamesh	range	23	Acer	Core i3	₹4,800	high
	Raju	mani	54	Mac	Core i5	₹85,650	very high
Subtotal						₹90,450	

# Dashboards

Dashboards in Salesforce are visual representations of your data, allowing you to monitor and analyze key metrics and trends. They provide a consolidated view of multiple reports, making it easier to make informed decisions based on real-time data.

## Steps to Create a Dashboard

### 1. Create a Dashboard Folder

#### 1. Access the Dashboard Tab:

- Click on the **App Launcher** (grid icon) and search for **Dashboards**.
- Click on the **Dashboards** tab.

#### 2. Create a New Folder:

- Click **New Folder**.
- Enter the **Folder Label** (e.g., "Total Rent Amount").
- The **Folder Unique Name** will be auto-populated.
- Click **Save**.

### 2. Create a New Dashboard

#### 1. Access the Dashboard Tab:

- a. Go to the **Dashboards** tab in the app.
2. **Create a New Dashboard:**
  - a. Click **New Dashboard**.
  - b. Enter a **Name** for the dashboard.
  - c. Select the folder you created earlier (e.g., "Total Rent Amount").
  - d. Click **Create**.
3. **Add Components to the Dashboard**
  1. **Add a Component:**
    - a. Click + **Add Component**.
    - b. Select a **Report** that you want to include in the dashboard.
    - c. Click **Select**.
  2. **Choose Component Type:**
    - a. Select the type of component you want to add (e.g., chart, gauge).
    - b. Configure the component settings (e.g., data range, display options).
  3. **Add Component to Dashboard:**
    - a. Choose a **Component Type** (e.g., chart) and add it to the dashboard.
    - b. Adjust the component settings as needed.
4. **Save and Finish**
  1. **Save the Dashboard:**
    - a. Click **Save** to save your changes.
  2. **Finalize the Dashboard:**
    - a. Click **Done** to complete the dashboard setup.

### Example Use Cases

- **Sales Overview:** Display charts and graphs summarizing sales performance, opportunities, and revenue.
- **Customer Metrics:** Show metrics related to customer interactions, support cases, and satisfaction.
- **Financial Summary:** Provide visual insights into financial data such as rent amounts, billing, and expenditures.



By creating and customizing dashboards, you can effectively visualize your data, track performance metrics, and gain insights to drive business decisions.

