

A CRM APPLICATION FOR LAPTOP RENTALS

by

Kondiparthi Suma Venkata Vasanta Lakshmi

21501a0595@pvpsit.ac.in

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:

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

2. Apex Development:

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

3. Report Creation:

- Design and customization of reports to display booking data.
- Use of bucket fields to categorize booking amounts into predefined ranges for better analysis.
- 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.
- **Enhanced Reporting:** Comprehensive reports providing detailed insights into booking trends, laptop usage, and financial metrics, aiding in strategic decision-making.

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

S no	Topics	page no
1	Introduction	5
2	Salesforce	6
3	Object Creation	10
4	Tabs	13
5	The Lightning App	16
6	Fields	19
7	Validation Rule	27
8	Profiles	28
9	Roles and Hierarchy	32
10	Users	34
11	Flows	36
12	Apex	39
13	Reports	48
14	Dashboards	50
15	Thankyou	53

*****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?

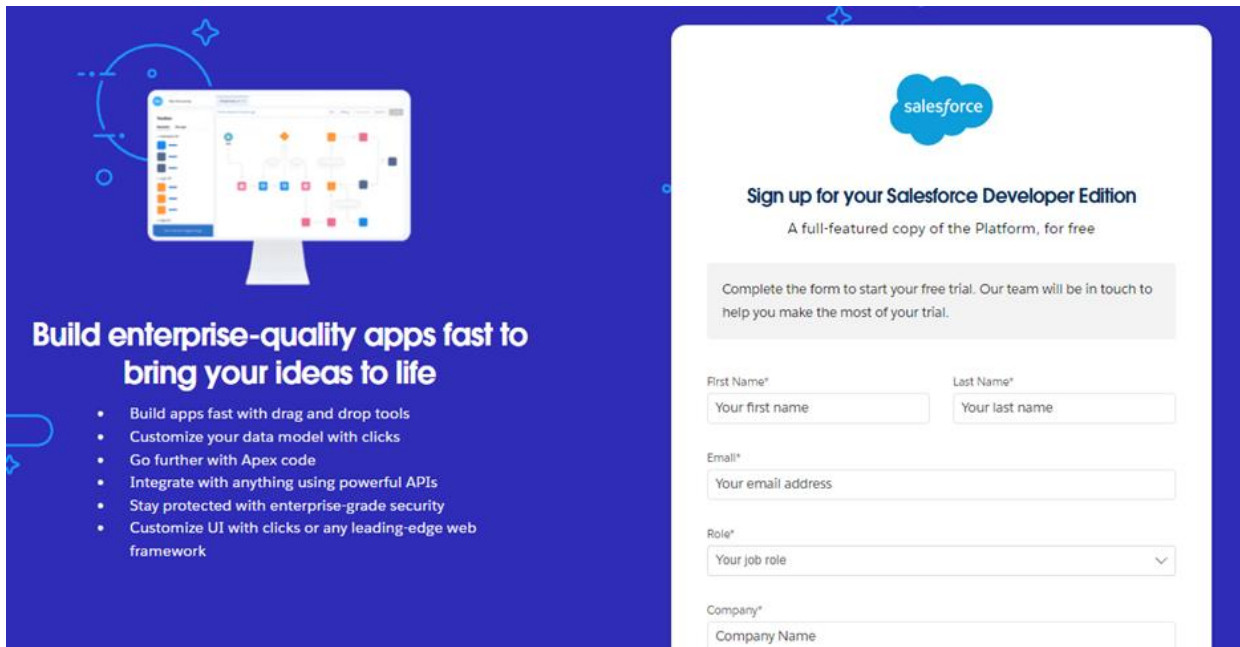
Salesforce serves as a comprehensive customer success platform, engineered to assist you in selling, servicing, marketing, analyzing, and connecting with your customers.

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>.



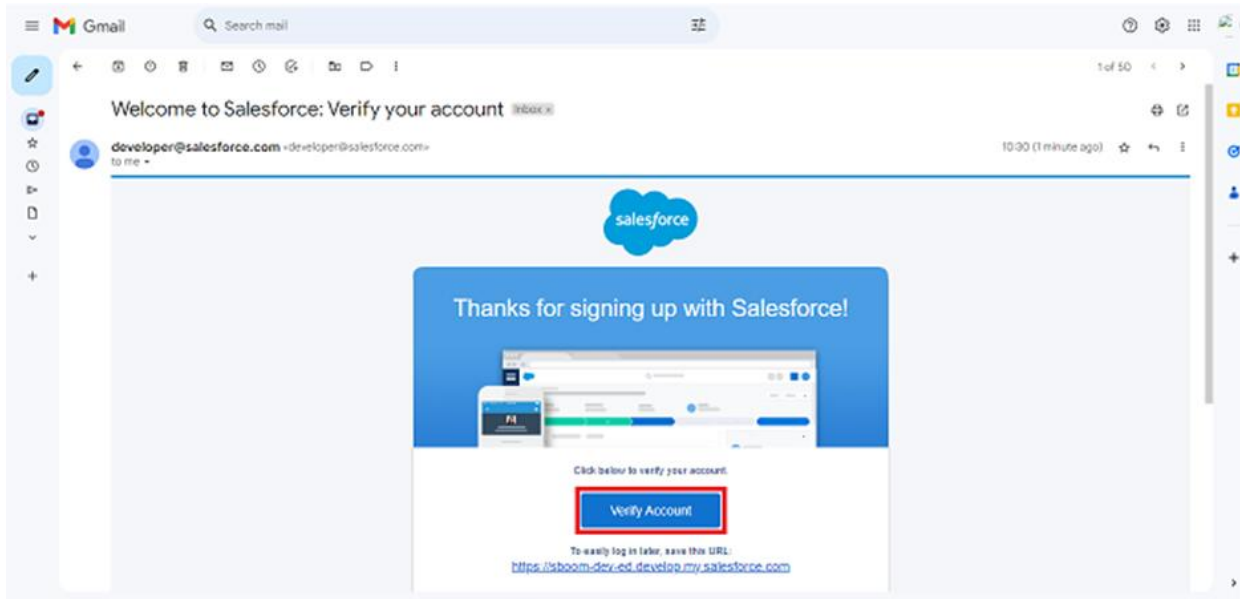
2. Complete the sign-up form with the following details:

- **First Name** and **Last Name**
- **Email Address**
- **Role**: Developer
- **Company**: College Name
- **Country**: India
- **Postal Code**: Enter your pin code
- **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

Change Your Password

Enter a new password for **lead@sb.oom**.
Make sure to include at least:

- ✓ 8 characters
- ✓ 1 letter
- ✓ 1 number

* New Password

.....
Good

* Confirm New Password

.....
Match

Security Question

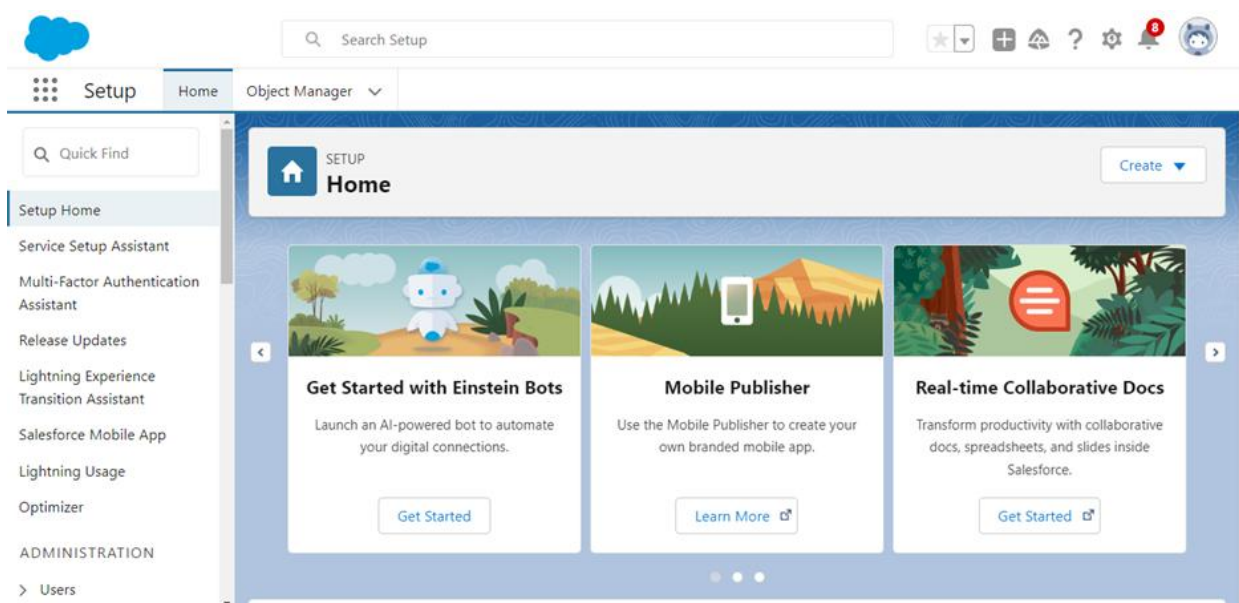
▼
In what city were you born?

* Answer

asdfghjkl

Change Password

.4.You will be redirected to your Salesforce setup page, where you can begin using your new developer account.



Creation in Salesforce

Object

What Is an Object?

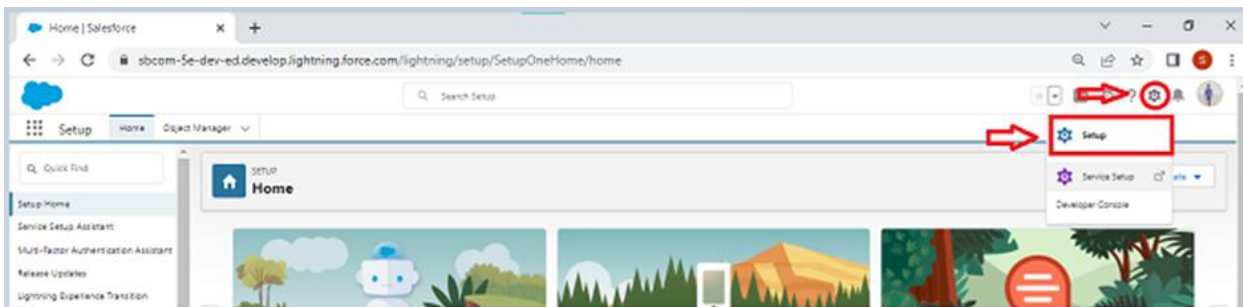
In Salesforce, objects are analogous to database s that allow you to store data pertinent to your organization. There are two primary types of Salesforce objects:

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.

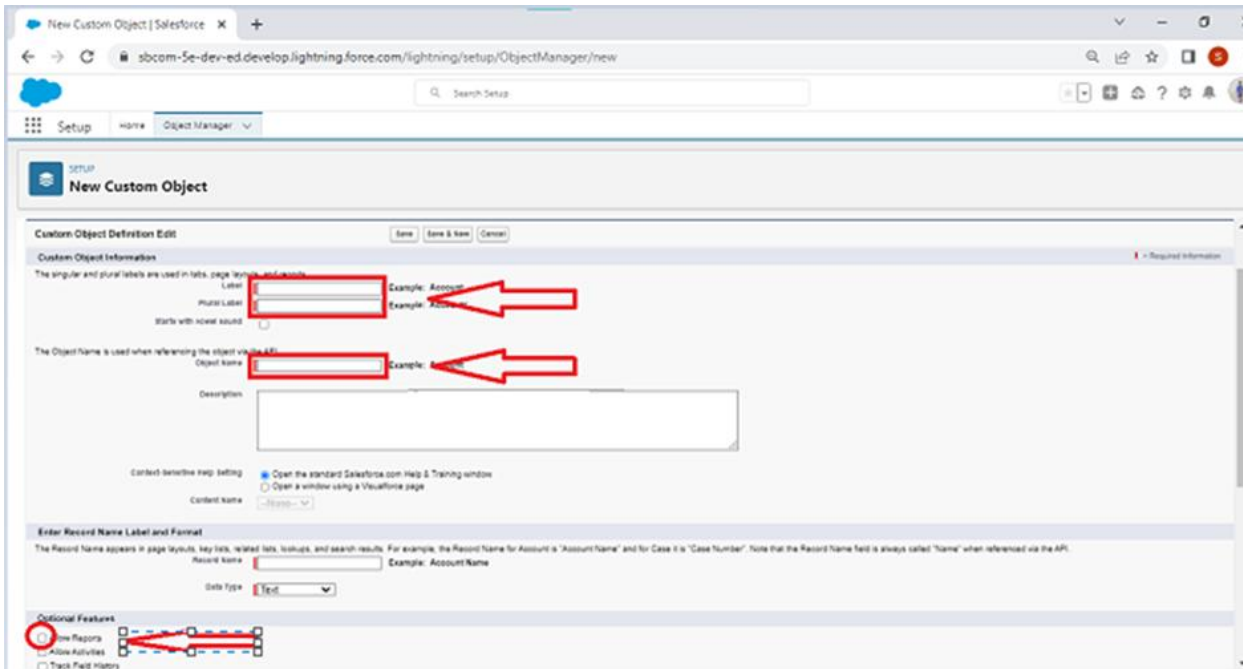


2. Click on Create and then select Custom Object.

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 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**:
 - **Record Name:** Total Laptops
 - **Data Type:** Text
4. Check the options for:
 - **Allow Reports**
 - **Allow Search**
 - **Track Field History**
5. Click **Save** to create the 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**:
 - **Record Name:** consumer_name
 - **Data Type:** Name
4. Check the options for:
 - **Allow Reports**
 - **Allow Search**
 - **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:**
 - **Record Name: Laptop Bookings**
 - **Data Type: Name**
4. Check the options for:
 - **Allow Reports**
 - **Allow Search**
 - **Track Field History**
5. Click **Save** to create the object.

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**
 - **Purpose:** Used for custom objects that you create in Salesforce. They function similarly to standard Salesforce tabs, such as Accounts, Contacts, and Opportunities.
 - **Functionality:** Provides a user interface for custom applications and objects.
2. **Web Tabs**
 - **Purpose:** Display web content or applications embedded within the Salesforce interface.
 - **Functionality:** Allows users to quickly access frequently used content and applications without leaving Salesforce.
3. **Visualforce Tabs**
 - **Purpose:** Display custom Visualforce pages.
 - **Functionality:** Visualforce tabs behave like standard Salesforce tabs but are designed to present custom Visualforce content.
4. **Lightning Component Tabs**
 - **Purpose:** Allow you to add Lightning components to the navigation menu in Lightning Experience and the Salesforce mobile app.
 - **Functionality:** Provides a way to incorporate custom Lightning components into the user interface.

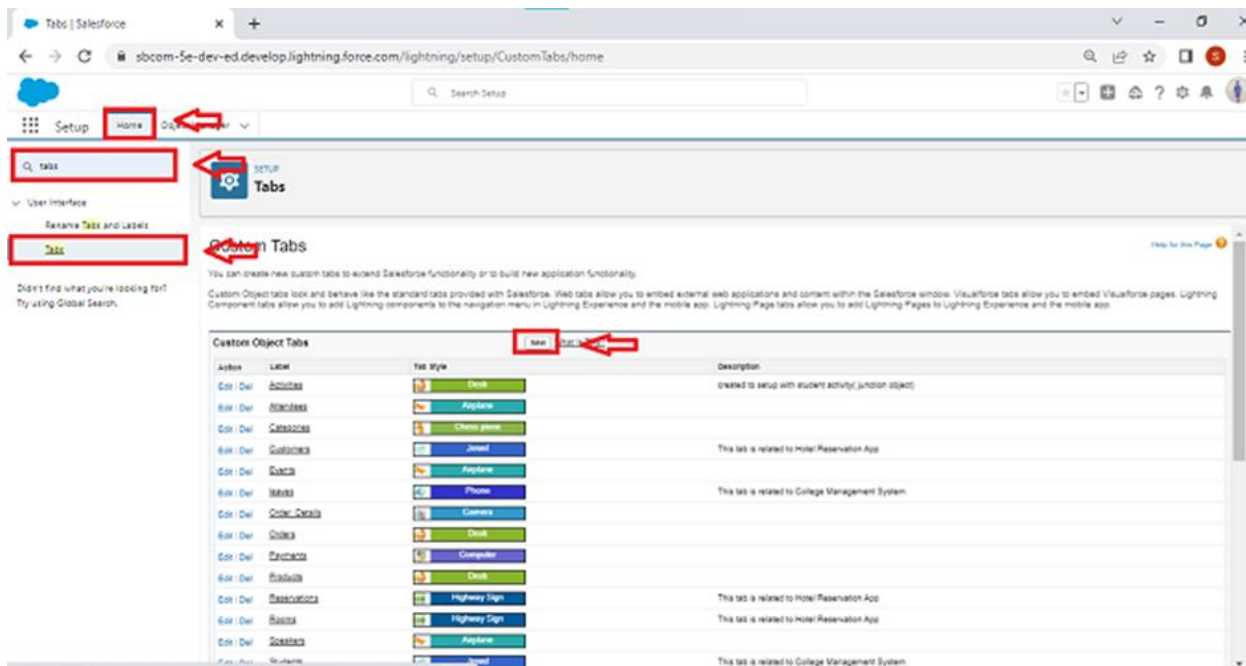
5. Lightning Page Tabs

- **Purpose:** Add Lightning Pages to the navigation menu in the mobile app.
- **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.



Step 3. Add to Custom Apps

Step 3 of 3

Choose the custom apps for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each Custom App.

Custom App	 <input type="checkbox"/> Include Tab
Platform (standard__Platform)	<input type="checkbox"/>
Sales (standard__Sales)	<input type="checkbox"/>
Service (standard__Service)	<input type="checkbox"/>
Marketing (standard__Marketing)	<input type="checkbox"/>
Sample Console (standard__ServiceConsole)	<input type="checkbox"/>
High Volume Customer Portal User	<input type="checkbox"/>
Authenticated Website User	<input type="checkbox"/>
App Launcher (standard__AppLauncher)	<input type="checkbox"/>
Analytics Studio (standard__Insights)	<input type="checkbox"/>
Sales Console (standard__LightningSalesConsole)	<input type="checkbox"/>
Service Console (standard__LightningService)	<input type="checkbox"/>
Sales (standard__LightningSales)	<input type="checkbox"/>
Lightning Usage App (standard__LightningInstrumentation)	<input type="checkbox"/>
Digital Experiences (standard__SalesforceCMS)	<input type="checkbox"/>
Queue Management (standard__QueueManagement)	<input type="checkbox"/>
Bolt Solutions (standard__LightningBolt)	<input type="checkbox"/>
Data Manager (standard__DataManager)	<input type="checkbox"/>
Salesforce Scheduler Setup (standard__LightningScheduler)	<input type="checkbox"/>

☒ Append tab to users' existing personal customizations



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

App Manager | Salesforce

sbcom-5e-der-ed.develop.lightning.force.com/lightning/setup/NavigationMenus/home

Search Setup

Setup Home Object Manager

Q: app manager

Lightning Experience App Manager

New Lightning App New Connected App

Clone App (Beta)

Didn't find what you're looking for? Try using Global Search.

Quickly create new Lightning apps by cloning existing apps. To use the beta feature, indicate that you've read all legal requirements and agree to participate by toggling Enable App Cloning. See additional details and terms in the [Winter 23 release notes](#).

Enable App Cloning ☐ Disabled

35 items • Sorted by App Name • Filtered by All app components • Select Type

App Name	Developer Name	Description	Last Modified	App Type	VL
1. All Tests	AllTests		04/12/2022, 10:13 am	Classic	
2. Analytics Studio	Analytics	Build CRM Analytics dashboards and apps	04/12/2022, 10:13 am	Classic	✓
3. App Launcher	AppLauncher	App Launcher tabs	04/12/2022, 10:13 am	Classic	✓
4. Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry	04/12/2022, 10:18 am	Lightning	✓
5. Charter Desktop	CharterDesktop	Charter Desktop is an Adobe AIR-based desktop application that lets Charter users stay connecte...	29/12/2022, 4:04 pm	Connected (Managed)	
6. Charter Mobile for BlackBerry	CharterforBlackBerry	The Salesforce.com Charter Mobile app lets you access Charter data on the go. Use it to view fee...	29/12/2022, 4:05 pm	Connected (Managed)	
7. College Management System	Nadearn	demo app	08/12/2022, 4:16 pm	Lightning	✓
8. Community	Community	Salesforce CRM Communities	04/12/2022, 10:13 am	Classic	✓
9. Content	Content	Salesforce CRM Content	04/12/2022, 10:13 am	Classic	✓
10. Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recode.	04/12/2022, 10:13 am	Lightning	✓

New Lightning App

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 [?]
Name your app...

*Developer Name [?]
Enter a developer name...

Description [?]
Enter a description...

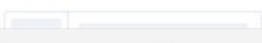
App Branding

Image [?]
 Upload

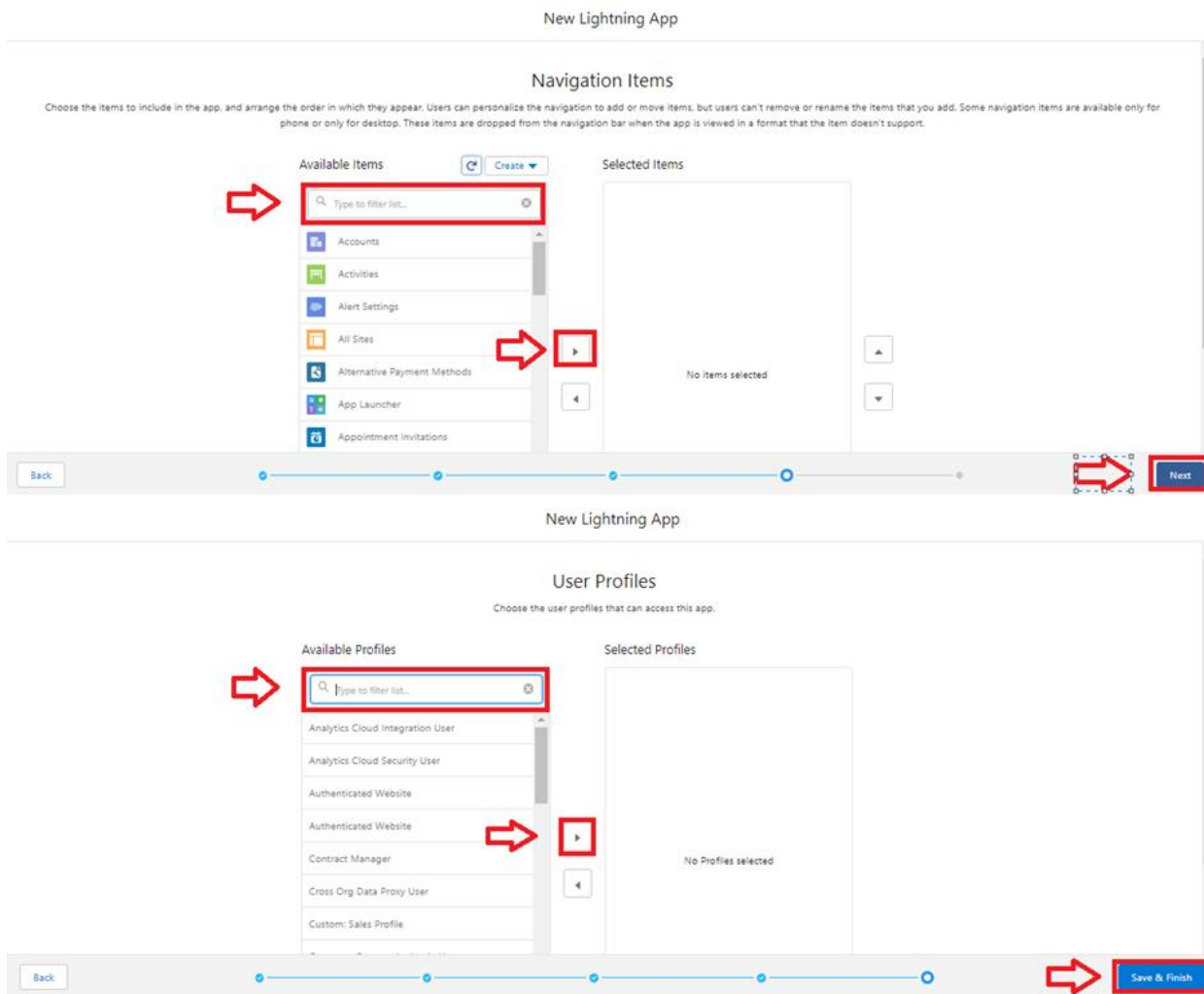
Primary Color Hex Value [?]
#0070D2

Org Theme Options
☐ Use the app's image and color instead of the org's custom theme

App Launcher Preview



Next



4o mini

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:

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

2. Email Field:

- Go to **Setup**.
- Click on **Object Manager**.
- Type **Consumer** in the search bar and click on the object.
- Click on **Fields & Relationships**.
- Click on **New**.

- Select **Data Type** as **Email** and click **Next**.
- Fill in the details:
 - **Field Label:** Email
 - **Field Name:** (auto-generated)
- Click **Next >> Next >> Save & New**.

3. Address Field:

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

4. Consumer Status Field:

- Go to **Setup**.
- Click on **Object Manager**.
- Type **Consumer** in the search bar and click on the object.
- Click on **Fields & Relationships**.
- Click on **New**.
- Select **Data Type** as **Picklist** and click **Next**.
- Fill in the details:
 - **Field Label:** Consumer Status
 - **Values:** Enter values with each value separated by a new line:
 - Student
 - Employee
 - Others
 - **Field Name:** (auto-generated)
- Check the **Required** option checkbox.
- Click **Next >> Next >> Save & New**.

<div><div>SETUP</div><div>Object Manager</div><div>103 Items, Sorted by Label</div></div> <div><input type="text" value="Quick Find"/> Schema Builder Create</div>			
Credential Stuffing Event Store	CredentialStuffingEventStore	Standard Object	
Credit Memo	CreditMemo	Standard Object	
Credit Memo Invoice Application	CreditMemoInvApplication	Standard Object	
Credit Memo Line	CreditMemoLine	Standard Object	
Customer	Customer	Standard Object	
Customer1	Customer__c	Custom Object	12/06/2023 ✓
D&B Company	DandBCompany	Standard Object	
Data Use Legal Basis	DataUseLegalBasis	Standard Object	
Data Use Purpose	DataUsePurpose	Standard Object	
Digital Wallet	DigitalWallet	Standard Object	
Duplicate Record Item	DuplicateRecordItem	Standard Object	
Duplicate Record Set	DuplicateRecordSet	Standard Object	
Email Message	EmailMessage	Standard Object	

SETUP > OBJECT MANAGER

consumer

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Step 2. Enter the details

Step 2 of 4

Previous Next Cancel

Field Label

phone number

Field Name

phone_number

Description

Help Text

Required

☒ Always require a value in this field in order to save a record

Auto add to custom report type

☒ Add this field to existing custom report types that contain this entity

Default Value

Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes ("the_text"). Include numbers without quotes (25), show percentages as decimals (0.10), and express date calculations in the standard format: {Today() + 7}. To reference a field from a Custom Metadata type record use \$CustomMetadata.Type__notRecordName.Field__c

Previous Next Cancel

SETUP > OBJECT MANAGER

consumer

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

consumer

New Custom Field

Help for this Page

Step 2. Enter the details

Step 2 of 4

Previous Next Cancel

Field Label

Email

Field Name

Email

Description

Help Text

Required

☐ Always require a value in this field in order to save a record

Unique

☐ Do not allow duplicate values

External ID

☐ Set this field as the unique record identifier from an external system

Auto add to custom report type

☒ Add this field to existing custom report types that contain this entity

Default Value

Show Formula Editor

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.**

SETUP > OBJECT MANAGER
Laptop Bookings

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Field Label

Values

☐ Use global picklist value set

☒ Enter values, with each value separated by a new line

☐ Display values alphabetically, not in the order entered

☐ Use first value as default value

☒ Restrict picklist to the values defined in the value set

Field Name

Description

Help Text

Previous Next Cancel

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.

SETUP > OBJECT MANAGER
Laptop Bookings

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Field Label

Values

☐ Use global picklist value set

☒ Enter values, with each value separated by a new line

☐ Display values alphabetically, not in the order entered

☐ Use first value as default value

☒ Restrict picklist to the values defined in the value set

Field Name

Description

Help Text

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

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

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.

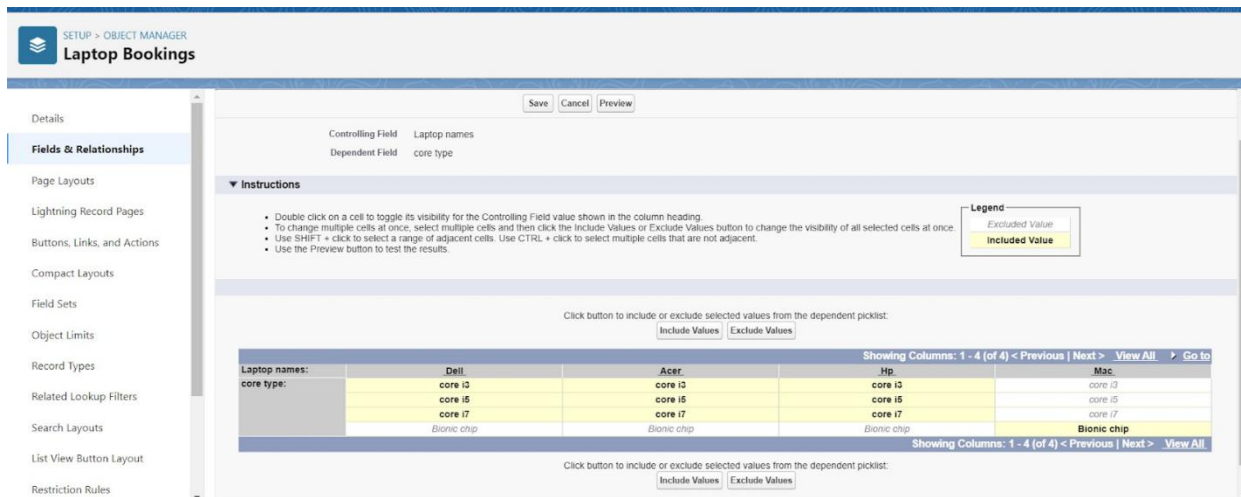
Creating Field Dependencies in the Laptop Bookings Object

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

1. **Go to Setup:**
 - Click on **Object Manager**.
 - Type **Laptop Booking** in the search bar and click on the object.
2. **Create Fields for Laptop Model and Processor Type:**
 - Follow the steps outlined previously to create the **Laptop Model** and **Processor Type** fields as picklists.
3. **Create Field Dependency:**
 - 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 **Field Dependencies**.
 - Click **New**.
 - Select the **Controlling Field** as **Laptop Model** and the **Dependent Field** as **Processor Type**.
 - Click **Continue**.
4. **Define the Field Dependency:**
 - You will see a matrix with the controlling field values on the left and the dependent field values on the top.
 - Select the values for each laptop model that should correspond to the processor types:
 - For **Dell**: check the boxes for **Core i3**, **Core i5**, **Core i7**.
 - For **Acer**: check the boxes for **Core i3**, **Core i5**, **Core i7**.
 - For **HP**: check the boxes for **Core i3**, **Core i5**, **Core i7**.
 - For **Mac**: check the box for **Bionic Chip**.
5. **Save the Field Dependency:**
 - 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.

40



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

To create a validation rule that ensures the phone number and email fields are not left blank in the Consumer object, follow these steps:

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:** PhonenumOrEmailBlankRule
 - **Description:** Phone number and email should not be blank.
4. **Enter the Formula:**
 - In the formula editor, enter the following formula:
 - plaintext
 - OR(ISBLANK(Phone_Number__c), ISBLANK(Email__c))
 - Click on **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. When a user tries to save a record without filling in either field, an error message will be displayed, guiding the user to provide the necessary information.

The screenshot shows the Salesforce Setup interface for creating a validation rule for the 'consumer' object. The left sidebar contains navigation links: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main content area is titled 'Display an error if Discount is more than 30%'. It includes a formula editor with the formula: `OR(ISBLANK(phone_number__c), ISBLANK(email__c))`. Below the formula editor is a 'Check Syntax' button with the message 'No errors found'. To the right of the formula editor is a 'FUNCTIONS' panel with a list of functions: ABS, ACOS, ADDMONTHS, AND, ASCII, and ASIN. Below the functions list is an 'Error Message' section. It includes an 'Example' message: 'Discount percent cannot exceed 30%'. Below the example is a text box for the 'Error Message' with the text: 'Please fill the phone number and email id'. Below the text box is a section for 'Error Location' with two radio buttons: 'Top of Page' (selected) and 'Field'.

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

By default, Salesforce provides the following 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:**
 - Click on **Profiles** in the Quick Find box.
2. **Clone the Standard User Profile:**
 - Find and click on the **Standard User** profile.
 - Click **Clone** to create a new profile based on the Standard User.
3. **Enter Profile Details:**
 - **Profile Name:** Enter Owner.
 - Click **Save**.
4. **Configure Custom Object Permissions:**
 - Scroll down to **Custom Object Permissions**.
 - Grant access permissions for the following objects:
 - **Total Laptops**
 - **Consumers**

- **Laptop Booking**
- **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:**

- 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.

4o mini

The screenshot displays the Salesforce Setup Profiles page for the 'owner' profile. It includes sections for Profile Detail, Page Layouts, Custom Object Permissions, Session Settings, and Password Policies.

Profile Detail:

Name	owner
User License	Salesforce
Description	
Created By	udayrushi.yelagandula, 10/07/2023, 10:56 am
Modified By	udayrushi.yelagandula, 10/07/2023, 10:56 am

Page Layouts:

Standard Object Layouts	Global	Email Application	Home Page Layout	Account	Alternative Payment Method	Appointment Invitation
Global	Global Layout [View Assignment]					
Email Application		Not Assigned [View Assignment]				
Home Page Layout			DE Default [View Assignment]			
Account				Account Layout [View Assignment]		
Alternative Payment Method					Alternative Payment Method Layout [View Assignment]	
Appointment Invitation						Appointment Invitation Layout [View Assignment]

Custom Object Permissions:

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

Session Settings:

Session Times Out After: 2 hours of inactivity

Session Security Level Required at Login: --None--

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:**
 - Type **Profiles** in the Quick Find box and select **Profiles**.
2. **Clone the Standard Platform User Profile:**

- Find and click on the **Standard Platform User** profile.
 - Click **Clone** to create a new profile based on the Standard Platform User.
3. **Enter Profile Details:**
- **Profile Name:** Enter Agent.
 - Click **Save**.
4. **Edit the Profile:**
- On the profile page, click **Edit**.
5. **Configure Custom Object Permissions:**
- Scroll down to **Custom Object Permissions**.
 - Grant access permissions for the following objects:
 - **Total Laptops**
 - **Consumer**
 - **Laptop Bookings**
 - **Billing Process**
6. Ensure the appropriate permissions (Read, Create, Edit, Delete) are checked for each object as required.
7. **Save the Profile:**
- 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.

The screenshot shows the Salesforce Setup Profiles page. The 'Profiles' section is active, displaying a table of permissions for the 'Agent' profile. The table is divided into two main sections: 'Custom Object Permissions' and 'Session Settings'.

Custom Object Permissions:

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

Session Settings:

Session Times Out After: 2 hours of inactivity

Session Security Level Required at Login: --None--

Password Policies:

User passwords expire in: 90 days

Enforce password history: 3 passwords remembered

Minimum password length: 8

Password complexity requirement: Must include alpha and numeric characters

Password question requirement: Cannot contain password

Maximum invalid login attempts: 10

Lockout effective period: 15 minutes

40 mini

Roles and Hierarchy

Creating Roles in Salesforce

Creating the Owner Role

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

This process sets up the hierarchy where the **Owner** role is at a higher level, and **Agent** roles are positioned beneath it, allowing for structured data visibility and access.

4o mini



SETUP
Roles

Role Edit
New Role
Help for this Page

Role Edit

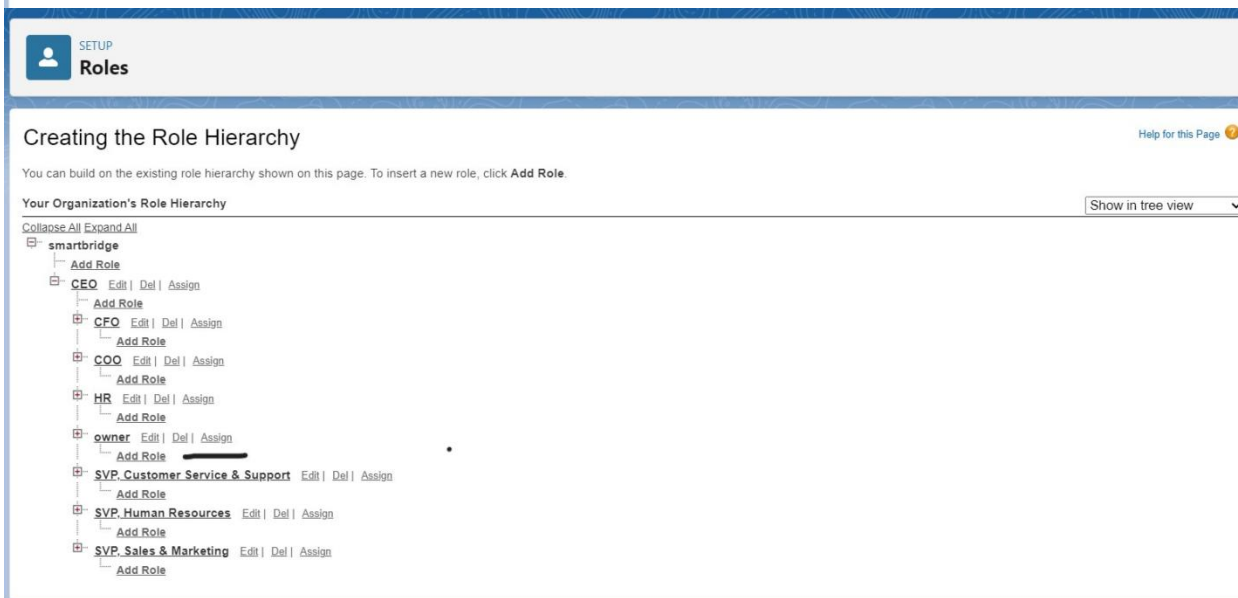
Label
owner

Role Name
owner

This role reports to
CEO

Role Name as displayed on reports

Save Save & New Cancel



Users

Creating Users in Salesforce

Creating a User with the Owner Role

- Go to Setup:**
 - In the Quick Find box, type **Users** and select **Users**.
- Create a New User:**
 - Click **New User**.
- Fill in the User Details:**

- **First Name:** Vicky
- **Last Name:** Y
- **Alias:** (Provide a suitable alias)
- **Email:** (Enter your personal email address)
- **Username:** (Format: text@text.text)
- **Nickname:** (Provide a suitable nickname)
- **Role:** Owner
- **User License:** Salesforce
- **Profile:** Owner

4. **Save:**

- Click **Save** to create the user.

Creating Another User with the Agent Role

1. **Go to Setup:**

- In the Quick Find box, type **Users** and select **Users**.

2. **Create a New User:**

- Click **New User**.

3. **Fill in the User Details:**

- **First Name:** Ram
- **Last Name:** Ram
- **Alias:** (Provide a suitable alias)
- **Email:** (Enter your personal email address)
- **Username:** (Format: text@text.text)
- **Nickname:** (Provide a suitable nickname)
- **Role:** Agent
- **User License:** Salesforce Platform
- **Profile:** Standard Platform User

4. **Save:**

- Click **Save** to create the user.

This will set up two users with their respective roles and profiles, enabling them to interact with Salesforce according to their assigned permissions.

4o mini

SETUP
Users

New User
Help for this Page

User Edit
Save Save & New Cancel

General Information
Required Information

First Name vicky
Last Name rushi
Alias vrush
Email udayrushi00@gmail.com
Username udayrushi00@456789gmail
Nickname vicky
Title
Company
Department
Division

Role owner
User License Salesforce
Profile Standard User
Active
Marketing User
Offline User
Knowledge User
Flow User
Service Cloud User
Site.com Contributor User
Site.com Publisher User
WDC User
Data.com User Type --None--
Data.com Monthly Addition Limit Default Limit (300)
Accessibility Mode (Classic Only)

SETUP
Users

New User
Help for this Page

User Edit
Save Save & New Cancel

General Information
Required Information

First Name ram
Last Name ramesh
Alias rrame
Email udayrushi@gmail.com
Username udayrushi@0145gmail.com
Nickname ram
Title
Company
Department
Division

Role Agent
User License Salesforce Platform
Profile Standard Platform User
Active
Marketing User
Offline User
Knowledge User
Flow User
Service Cloud User
Site.com Contributor User
Site.com Publisher User
WDC User
Data.com User Type --None--
Data.com Monthly Addition Limit Default Limit (300)
Accessibility Mode (Classic Only)

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.

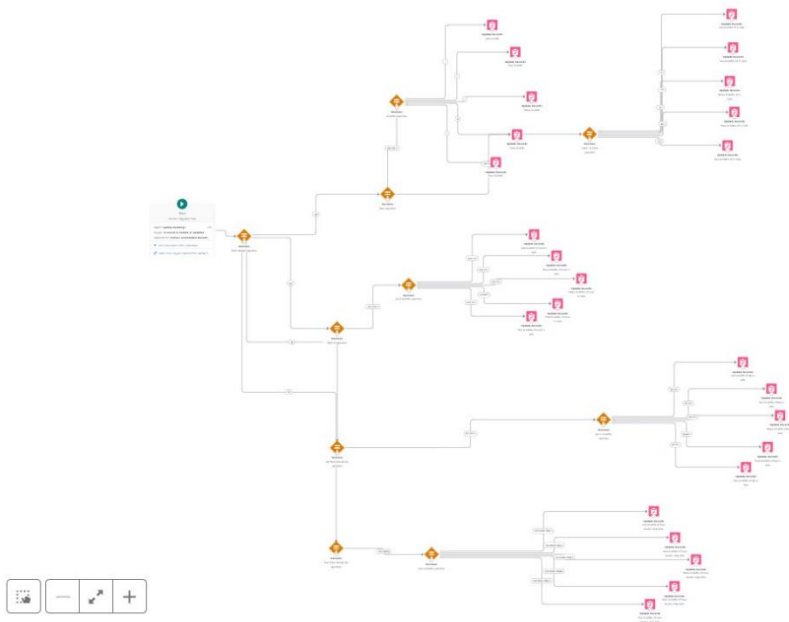
Flow Elements: Flow Builder offers various elements that you can use to create flows, such as variables, decisions, loops, and more. These elements allow you to build sophisticated logic into your flows.

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.



Click on save .

Label:- Laptop distributions, api name:- automatically filled

Save the flow and activate it.

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 :

While still in the trailhead account, navigate to the gear icon in the top right corner.

Click on developer console and you will be navigated to a new console window.

Click on the File menu in the toolbar, and click on new- Trigger.

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);

}

}

```

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});

}

}

}

```

Note:-

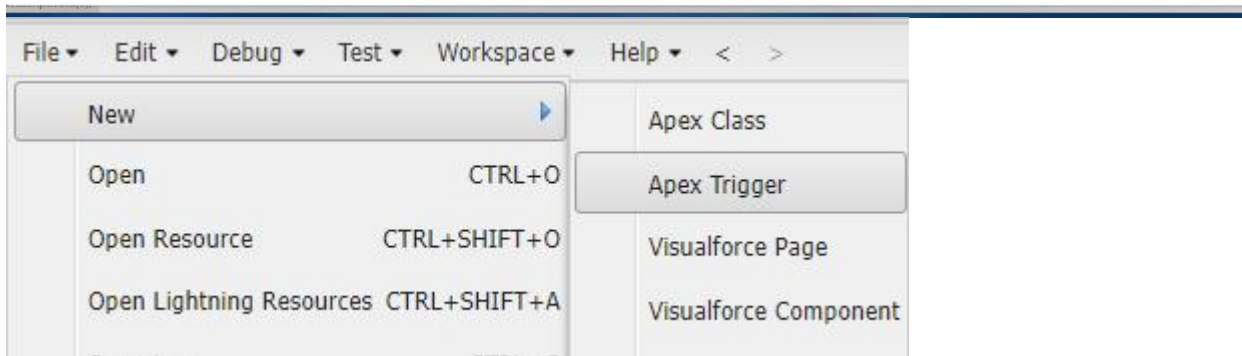
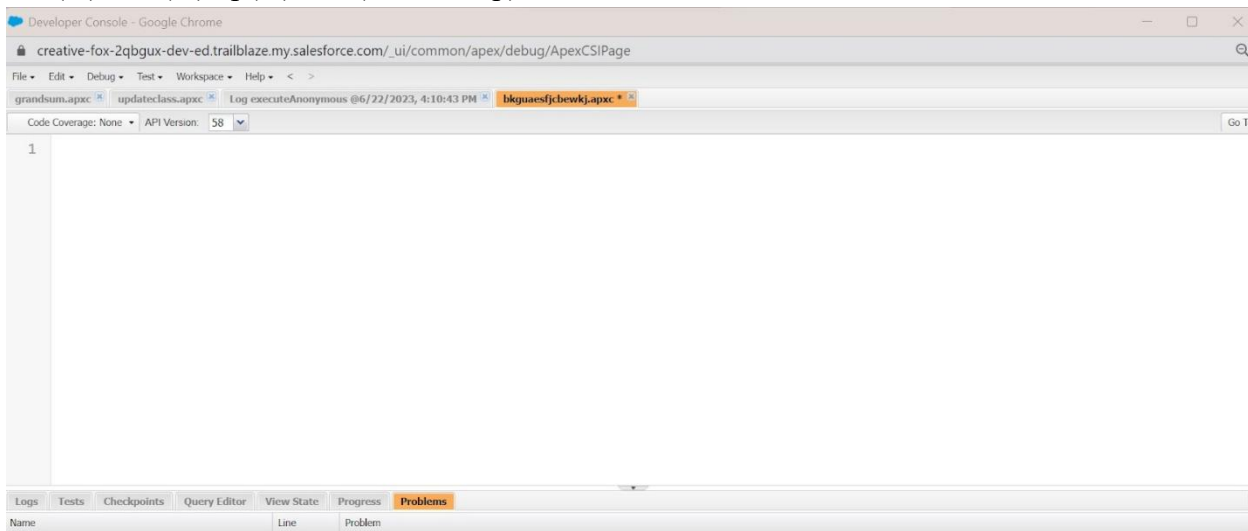
- 1.Class name:- LaptopBookingHandler
- 2.API Name:- Laptop_Bookings__c(as per your org go to laptop booking object and copy from that).
- 3.core__c (as per your org go to laptop booking object and copy from that).
- 4.Laptop_type__c.(as per your org go to laptop booking object and copy from that).

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.

Result:

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).



New Apex Trigger

Name:

sObject:

Submit

Developer Console - Google Chrome

google-7da-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

LaptopBookingHandler.apxc LaptopBooking.apxt

Code Coverage: None API Version: 58

```

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

```

LaptopBookingHandler.apxc LaptopBooking.apxt

Code Coverage: None API Version: 58 Go

```

1 public class LaptopBookingHandler {
2     public static void sendEmailNotification (List<Laptop_Bookings__c> lapList){
3         for(Laptop_Bookings__c lap:lapList)
4         {
5             Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
6             email.setToAddresses( new List<String>{lap.Email__c});
7             email.setSubject('Welcome to our company');
8             string body = 'Dear ' +lap.Name +', \n';
9             body += 'Welcome to Laptop Rentals! You have been seen as a valuable customer to us.\n Please continue your journey with us, while
10             email.setPlainTextBody(body);
11             Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
12
13         }
14     }
15 }
16

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem
------	------	---------

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

Tabular

Summary

Matrix

Joined Reports

1. Tabular 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.

Report: consumer with Laptop Bookings and Total Laptops consumer with Laptop Bookings and laptop									
Total Records	Total Amount								
10	₹39,100								
<input type="checkbox"/> type of versions ↑	consumer: consumer_name	Laptop Bookings: Laptop Bookings	Address	Phone number	Laptop Names	core type	Total No Of Laptops: Total Laptops	Amount	Email ↑
<input type="checkbox"/> basic (2)	sam	sam	street6	9875366705	Acer	Core i5	8	₹400	sam1@gmail.com
	srvs	srvs	street7	9857837982	Hp	Core i7	13	₹900	srv1a@gmail.com
Subtotal								₹1,300	
<input type="checkbox"/> intermediate (2)	suma	Set In Review to True	street1	7953790536	Hp	Core i7	4	₹1,200	suma2@gmail.com
	surekha	Set In Review to True	satanagar	9857837987	Dell	Core i5	6	₹1,500	surekha@gmail.com
Subtotal								₹2,700	
<input type="checkbox"/> high (6)	anju	anju	street9	7953790529	Dell	core i3	5	₹5,000	anju2@gmail.com
	google	google	street5	9857837989	Mac	Bionic chip	4	₹8,500	google12@gmail.com
	jyo	jyo	street8	7953790563	Mac	Bionic chip	8	₹6,800	jyo1@gmail.com
	nandu	nandu	street43	6875366706	Hp	Core i5	12	₹1,700	nandu1@gmail.com
	smartinlernz	smartinlernz	street4	7953790539	Mac	Bionic chip	11	₹5,100	smart1@gmail.com
	suma	suma	street3	7953790536	Acer	Core i7	16	₹8,000	suma2@gmail.com
Subtotal								₹35,100	
Total (10)								₹39,100	

Edit Subscription

Settings

Frequency

Daily

Weekly

Monthly

Time

8:00 am

Attachment

Attach File

Recipients

Send email to

Me

Edit Recipients

Run Report As

☐ Me

☒ Another Person

Cancel

Save

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

Click on the app launcher and search for the dashboard.

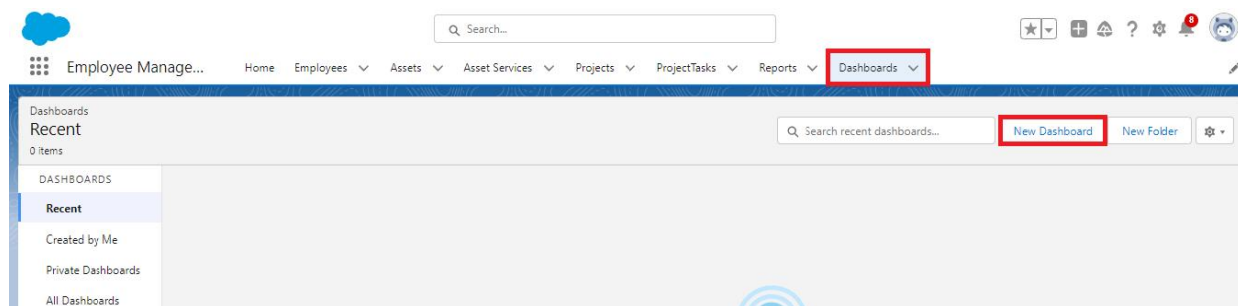
Click on the dashboard tab.

Click the new folder, give the folder label as “total rent amount”.

Folder unique names will be auto populated.

Click save.

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



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

New Dashboard

* Name

data analytics of laptops

Description

total amount of data in dashboards

Folder

total rents amount

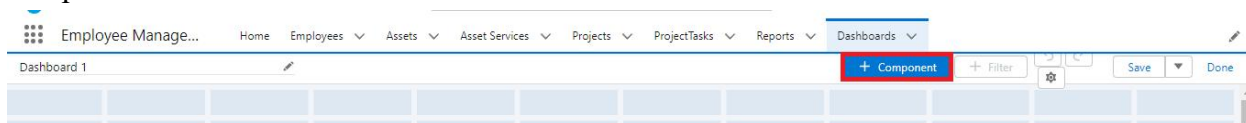
Select Folder

Cancel

Create

Select add

component.



Select a

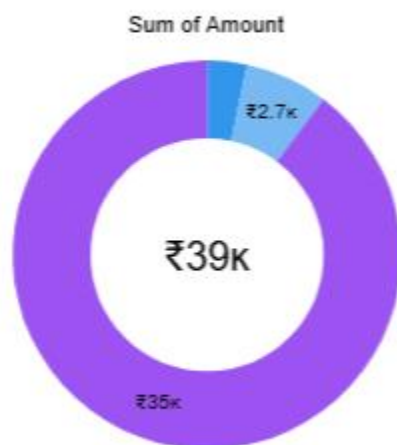
Report and click on select.

Select the dark component and add to the dashboards.

Save it.

Click done.

consumer with Laptop Bookings and laptop



type of versions

basic

intermediate

high

[View Report \(consumer with Laptop Bookings and laptop\)](#)

THANKYOU