

A PROJECT REPORT

ON

CRM APPLICATION

FOR

LAPTOP RENTALS

Submitted By:

Name: Kaushik Koganti

**College: Prasad V. Potluri Siddhartha Institute of
Technology**

Roll Number: 21501A0589

Email: 21501A0589@pvpsit.ac.in

Skill Wallet ID: SWUID20240039482

INDEX

S.No	Title	Page Nos
1.	Introduction	3
2.	Salesforce and Developer Account Creation	4-6
3.	Object Creation	7-11
4.	Tabs	8-15
5.	The Lightning App	16-18
6.	Fields	19-33
7.	Validation Rule	34-35
8.	Profiles	35-37
9.	Roles and Hierarchy	38
10.	Users	39-40
11.	Flows	41-45
12.	APEX	46-50
13.	Reports	51-53
14.	Dashboards	54-55
15.	Conclusion	56

A CRM APPLICATION FOR LAPTOP RENTALS

Introduction:

The CRM (Customer Relationship Management) application for laptop rentals is designed to streamline the process of delivering rental laptops to customers while enhancing the overall customer experience. It empowers businesses by leveraging the core principles of CRM to build strong relationships with customers, optimize operations, and ensure seamless service delivery.

One of the primary objectives of this CRM application is to enhance customer experiences by understanding their preferences and ensuring timely deliveries. Through detailed customer profiles and rental history tracking, businesses can personalize their offerings and provide services that cater to specific customer needs. This personalized approach not only fosters customer satisfaction but also helps in retaining loyal clients.

In addition to improving customer satisfaction, the CRM application focuses on operational efficiency. By automating key processes such as order tracking, inventory management, and delivery scheduling, the application ensures that businesses operate with minimal delays and errors. This optimization helps save time and resources while delivering consistent and reliable services.

A crucial aspect of this CRM application is effective communication with potential and existing customers. It integrates email marketing strategies to reach out to identified potential customers, keeping them informed about rental offers, promotions, and updates. This targeted communication helps nurture leads, convert them into loyal customers, and maintain long-term relationships.

By integrating customer data, communication tools, and analytics, the CRM application provides businesses with actionable insights. These insights help in decision-making, identifying trends, and improving services based on customer feedback.

Overall, the CRM application for laptop rentals is not just a tool for managing customer relationships; it is a comprehensive solution for improving customer engagement, optimizing store operations, and driving business growth through effective communication and streamlined processes.

Salesforce:

Salesforce is a powerful and innovative technology platform that has revolutionized the way businesses operate. It serves as a comprehensive customer success platform, enabling companies to sell, service, market, analyze, and connect with their customers more effectively.

Designed with flexibility and productivity in mind, Salesforce provides businesses with a range of tools and features to streamline operations and enhance customer relationships. It allows organizations to manage prospects and customers, collaborate with employees and partners, and securely store data in the cloud—all accessible from anywhere.

With Salesforce, businesses can:

- **Sell Smarter:** Streamline sales processes and close deals faster.
- **Provide Exceptional Service:** Deliver outstanding customer support experiences.
- **Market Effectively:** Engage with prospects through targeted marketing campaigns.
- **Analyze Data:** Gain valuable insights through robust analytics.
- **Collaborate and Connect:** Foster better communication with employees, partners, and customers.

At its core, Salesforce empowers businesses to work smarter and faster. It helps sales teams close deals efficiently, customer service teams deliver exceptional support, and marketing teams run targeted campaigns to engage with audiences. Additionally, its powerful analytics tools provide deep insights into data, enabling informed decision-making and fostering business growth.

Beyond functionality, Salesforce enhances collaboration by connecting employees, partners, and customers on a unified platform. Its versatility and cloud-based accessibility make it an ideal solution for businesses of all sizes, driving success and productivity across industries.

In essence, Salesforce is more than just a software platform—it's a game-changing solution that helps businesses achieve their goals and build lasting customer relationships in today's dynamic world.

Creating a Developer Account:

Steps to Create a Developer Org in Salesforce

1. Visit the Salesforce Developer Sign-Up Page
 - Go to <https://developer.salesforce.com/signup>.

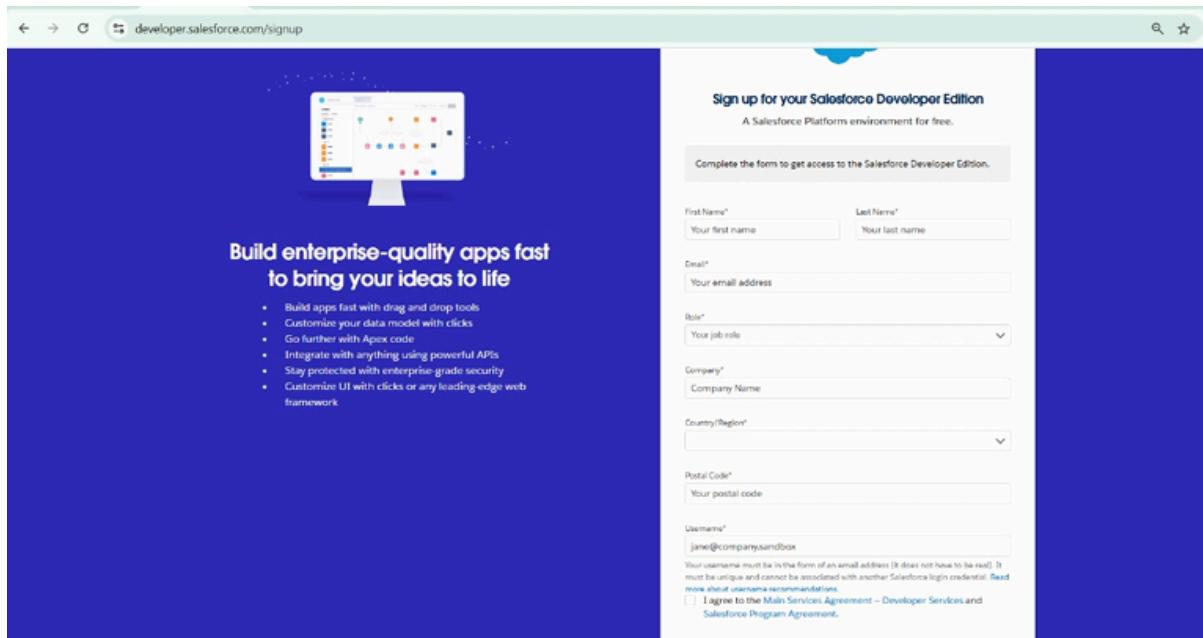
2. Fill Out the Sign-Up Form

Provide the following details in the form:

- **First Name & Last Name:** Enter your full name.
- **Email:** Provide a valid email address.
- **Role:** Select *Developer* from the dropdown.
- **Company:** Enter your college name.
- **Country:** Choose *India*.
- **Postal Code:** Enter your pin code.
- **Username:** Create a username combining your name and company in the format:
username@organization.com
(Note: This need not be an actual email ID.)

3. Submit the Form

- After filling in all the details, click on the **Sign Me Up** button.



The screenshot shows the 'Sign up for your Salesforce Developer Edition' page. The page has a blue header with the text 'Sign up for your Salesforce Developer Edition' and 'A Salesforce Platform environment for free.' Below the header, there's a sub-header 'Complete the form to get access to the Salesforce Developer Edition.' The form contains the following fields:

- First Name* (text input field)
- Last Name* (text input field)
- Email* (text input field)
- Role* (dropdown menu)
- Company* (text input field)
- Country/Region* (dropdown menu)
- Postal Code* (text input field)
- Username* (text input field)
Jane@company.sandbox
Your username must be in the form of an email address (it does not have to be real). It must be unique and cannot be associated with another Salesforce login credential. Read more about username recommendations.
- I agree to the [Main Services Agreement – Developer Services](#) and [Salesforce Program Agreement](#).

You will receive a confirmation email with further instructions to activate your developer org.

Account Activation:

Steps for Account Activation in Salesforce

1. Check Your Email Inbox

- Open the inbox of the email address you used during the sign-up process.
 - Look for an email from Salesforce with the subject related to account verification.
 - Note: The email may take 5–10 minutes to arrive.

2. Verify Your Account

- Click on the Verify Account link in the email to begin the activation process.

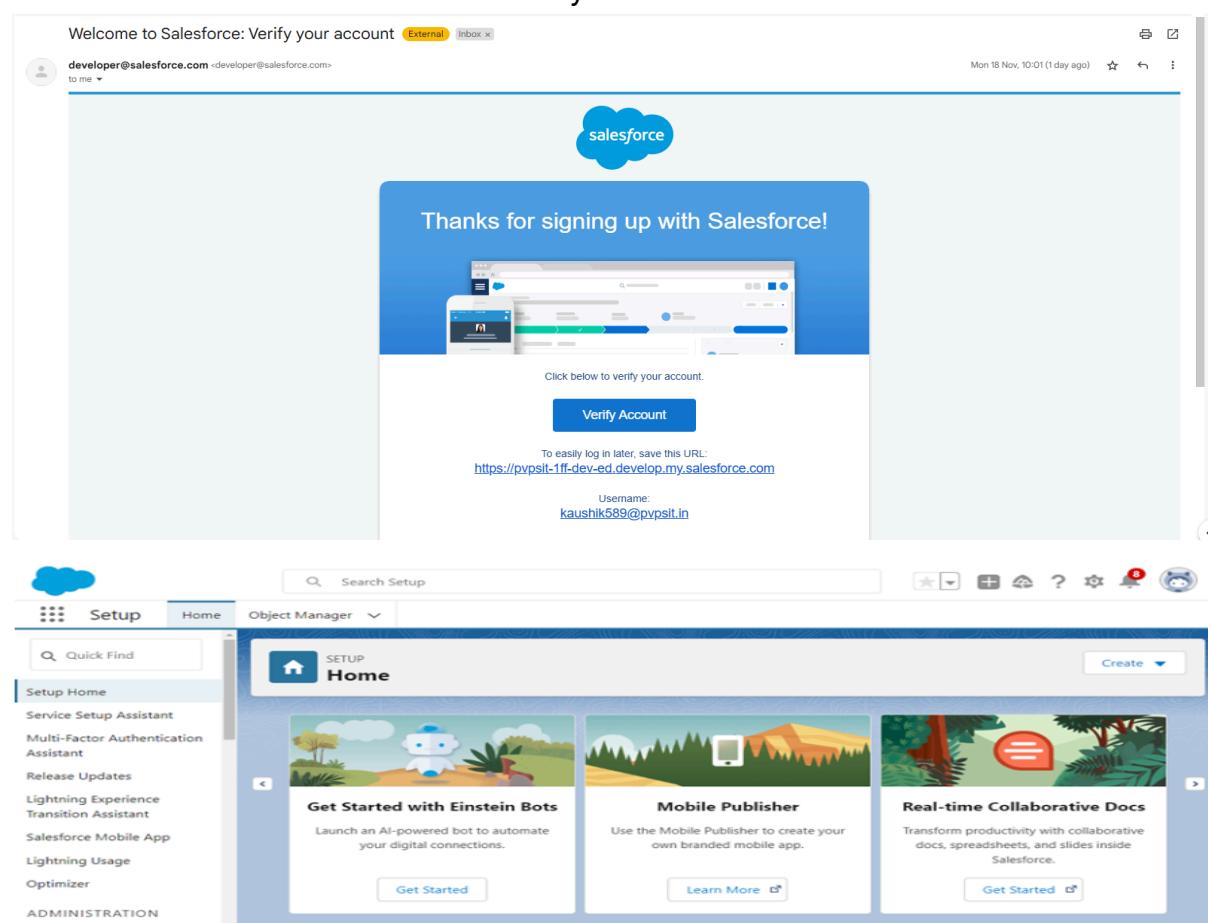
3. Set Your Password

- Create a password for your Salesforce account.
 - Choose and answer a security question for account recovery.
 - Click on Change Password to save your settings.

4. Access Salesforce Setup

- Once the password is set, you will be redirected to your Salesforce setup page.

Your account is now activated and ready for use.



Object Creation:

Understanding Salesforce Objects

Salesforce objects are database tables that allow you to store data specific to your organization. They are classified into two types:

1. Standard Objects

- Provided by Salesforce, these include objects such as Users, Contracts, Reports, Dashboards, etc.

2. Custom Objects

- Created by users, these objects store unique and essential information specific to an organization. They form the foundation of applications and provide a structure for data sharing.

Steps to Navigate to the Setup Page

1. Click on the **Gear Icon** in the Salesforce interface.
2. Select **Setup** from the dropdown menu.

Steps to Create a Custom Object in Salesforce

1. From the **Setup Page**, navigate to **Object Manager**.
2. Click on **Create** and then select **Custom Object**.
3. On the Custom Object definition page, do the following:
 - Enter the **Label Name** (e.g., "Project").
 - Enter the **Plural Label Name** (e.g., "Projects").
 - Select options such as **Allow Reports** and **Allow Search**.
4. Click on **Save** to create the custom object.

Your custom object is now ready for use in Salesforce.

The screenshot shows the Salesforce Setup interface. At the top, there is a navigation bar with 'Setup' and 'Object Manager' selected. An arrow points to the 'Object Manager' button. Below the navigation bar, the 'Object Manager' page is displayed, showing a list of objects. A circled 'Create' button is highlighted. A second arrow points to the 'Create' button on the right side of the page. The main content area shows a table with columns: LABEL, API NAME, TYPE, DESCRIPTION, and LAST MC. A new row is being edited, with 'Label' set to 'Project' and 'Plural Label' set to 'Projects'. The 'Custom Object' button is also circled. At the bottom, the 'New Custom Object' page is shown, where the 'Label' field is set to 'Project' and 'Plural Label' is set to 'Projects'. The 'Object Name' field is set to 'Account' and 'Example' is 'Accounts'. There is a note about permissions and a 'Save' button at the top of this page.

Create Total Laptops Object:

1. Navigate to the Setup Page

- Click on the Gear Icon and select **Setup**.

2. Open Object Manager

- From the Setup page, click on **Object Manager**.

3. Create a New Custom Object

- Click on **Create** and select **Custom Object**.

4. Define the Custom Object

- **Label Name:** Enter *Total Laptops*.
- **Plural Label Name:** Enter *Total Laptops*.
- **Record Name Label:** Enter *Total Laptops*.
- **Record Name Data Type:** Select **Text**.

5. Configure Additional Options

- Select the following options:
 - **Allow Reports**
 - **Allow Search**
 - **Track Field History**

6. Save the Object

- Click on **Save** to finalize the creation of the "Total Laptops" custom object.

The top screenshot shows the 'Custom Object Definition Edit' screen under 'SETUP'. It includes fields for 'Label' (Total Laptops), 'Plural Label' (Total Laptops), 'Object Name' (Total_Laptops), and a 'Description' text area. Context-sensitive help settings are also shown. The bottom screenshot shows the 'Object Manager' screen for the 'Total Laptops' object, displaying its details and settings like 'Enable Reports' and 'Track Field History'.

Create Consumer Object:

1. Navigate to the Setup Page

- Click on the Gear Icon and select **Setup**.

2. Open Object Manager

- From the Setup page, click on **Object Manager**.

3. Create a New Custom Object

- Click on **Create** and select **Custom Object**.

4. Define the Custom Object

- **Label Name:** Enter *consumer*.
- **Plural Label Name:** Enter *consumer*.
- **Record Name Label:** Enter *consumer_name*.
- **Record Name Data Type:** Select *Name*.

5. Configure Additional Options

- Select the following options:
 - **Allow Reports**
 - **Allow Search**
 - **Track Field History**

6. Save the Object

- Click on **Save** to finalize the creation of the "Consumer" custom object.

The screenshot shows the 'New Custom Object' setup page. In the 'Custom Object Information' section, the 'Label' is set to 'consumer' and the 'Plural Label' is also set to 'consumer'. The 'Object Name' field is also populated with 'consumer'. Under 'Description', there is a large text input field. In the 'Content Name' dropdown, the value '--None--' is selected. At the bottom, the 'Content Name' dropdown is expanded to show the option 'consumer'. The right side of the screen displays the 'Details' tab for the 'consumer' object, listing various configuration settings like API Name, Singular Label, Plural Label, and various checkboxes for reports, activities, and history tracking.

Create Laptop Bookings Object:

1. Navigate to the Setup Page
 - Click on the Gear Icon and select Setup.
2. Open Object Manager
 - From the Setup page, click on Object Manager.
3. Create a New Custom Object
 - Click on Create and select Custom Object.
4. Define the Custom Object
 - Label Name: Enter Laptop Bookings.
 - Plural Label Name: Enter Laptop Bookings.
 - Record Name Label: Enter Laptop Bookings.
 - Record Name Data Type: Select Name.
5. Configure Additional Options
 - Select the following options:
 - Allow Reports
 - Allow Search
 - Track Field History
6. Save the Object
 - Click on Save to finalize the creation of the "Laptop Bookings" custom object.

The screenshot shows the 'New Custom Object' setup page in Salesforce. The top navigation bar includes 'SETUP' and the 'New Custom Object' title. Below the title is a 'Custom Object Information' section with fields for 'Label' (Laptop Bookings), 'Plural Label' (Laptop Bookings), and 'Object Name' (Laptop_Bookings). There are also checkboxes for 'Starts with vowel sound' and 'Allows API access'. The 'Description' field is empty. Under 'Context-Sensitive Help Setting', the 'Open the standard Salesforce.com Help & Training window' option is selected. The 'Content Name' dropdown is set to 'None'. At the bottom right are 'Edit' and 'Delete' buttons. The main body of the page is titled 'Details' and contains sections for 'Description', 'API Name' (Laptop_Bookings_c), 'Custom' (checked), 'Singular Label' (Laptop Bookings), 'Plural Label' (Laptop Bookings), 'Enable Reports' (checked), 'Track Activities', 'Track Field History' (checked), 'Deployment Status' (Deployed), 'Help Settings' (Standard salesforce.com Help Window), and 'Edit' and 'Delete' buttons.

Create Billing Process Object:

1. Navigate to the Setup Page
 - Click on the Gear Icon and select **Setup**.
2. Open Object Manager
 - From the Setup page, click on **Object Manager**.
3. Create a New Custom Object
 - Click on **Create** and select **Custom Object**.
4. Define the Custom Object
 - **Label Name:** Enter *Billing Process*.
 - **Plural Label Name:** Enter *Billing Process*.
 - **Record Name Label:** Enter *Billing ProcessName*.
 - **Record Name Data Type:** Select *Name*.
5. Configure Additional Options
 - Select the following options:
 - **Allow Reports**
 - **Allow Search**
 - **Track Field History**
6. Save the Object
 - Click on **Save** to finalize the creation of the "Billing Process" custom object.

The screenshot shows the Salesforce Setup interface for creating a new custom object. The top navigation bar includes 'SETUP' and 'New Custom Object'. The main window title is 'New Custom Object'. A message bar at the top states: 'Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more](#) | [Don't show this message again](#)'. Below this is a 'Custom Object Definition Edit' section with tabs for 'Save', 'Save & New', and 'Cancel'. The 'Custom Object Information' tab is selected, showing fields for 'Label' (set to 'Billing Process'), 'Plural Label' (set to 'Billing Process'), and 'Example' fields for 'Account' and 'Accounts'. There is also a checkbox for 'Starts with vowel sound'. The 'Object Name' field is set to 'Billing Process' with an example of 'Account'. A 'Description' field is present but empty. At the bottom of the window, there is a breadcrumb trail 'SETUP > OBJECT MANAGER' and the object name 'Billing Process'. The right side of the screen shows a sidebar with 'Details' and various configuration sections like 'Fields & Relationships', 'Page Layouts', etc., along with their corresponding settings.

Tabs:

In Salesforce, **tabs** serve as the user interface elements that allow you to build, view, and manage records for various objects. They provide a convenient way for users to access and interact with records in a structured and organized manner.

Types of Tabs in Salesforce

1. Custom Tabs

Custom tabs are designed for custom objects and applications that you build in Salesforce. These tabs provide a similar interface to standard Salesforce tabs (like Accounts, Contacts, and Opportunities), enabling users to interact with the data specific to your organization's needs. Custom tabs allow for more flexibility and personalization within your Salesforce instance.

2. Web Tabs

Web tabs are custom tabs that display external web content or applications directly within the Salesforce interface. This feature allows users to quickly access frequently used web applications or content without leaving the Salesforce environment. Web tabs are ideal for integrating external resources that need to be accessed regularly by users, making them highly functional and seamless.

3. Visualforce Tabs

Visualforce tabs are used to display Visualforce pages within Salesforce. Visualforce is a framework that allows developers to create custom pages that can be embedded within Salesforce. These tabs offer a highly customizable interface and can provide tailored content, forms, and views to users, offering enhanced flexibility compared to standard Salesforce tabs.

4. Lightning Component Tabs

Lightning Component tabs allow you to integrate custom Lightning components into the Salesforce navigation menu. These tabs are useful for adding reusable components to both Lightning Experience and the mobile app. With Lightning Component tabs, you can create dynamic and responsive user interfaces that interact seamlessly with your Salesforce data.

5. Lightning Page Tabs

Lightning Page tabs enable you to add custom Lightning Pages to the Salesforce mobile app navigation menu. Unlike other types of tabs, Lightning Page tabs do not appear on the "All Tabs" page, nor do they show up in the "Available Tabs" list. Instead, once created, they serve to bring custom Lightning Pages into the mobile experience, enhancing user navigation on mobile devices.

Creating a Custom Tab:

Steps to Create a Custom Tab for "Total Laptops" Object

1. **Navigate to the Setup Page**
 - Click on the Gear Icon and select **Setup**.
2. **Find the Tabs Section**
 - In the **Quick Find** search bar, type **Tabs**.
 - Click on **Tabs** in the results.
3. **Create a New Tab**
 - Under the **Custom Object Tabs** section, click on **New**.
4. **Select the Object**
 - Choose **Total Laptops** from the dropdown list of objects.
5. **Select Tab Style**
 - Choose a suitable **Tab Style** from the available options.
 - Click **Next**.
6. **Add to Profiles**
 - In the **Profiles Page**, keep the default settings and click **Next**.
7. **Add to Custom App**
 - For the **Custom App**, uncheck the option **Include Tab**.
8. **Append Tab to Users' Customizations**
 - Ensure that **Append tab to users' existing personal customizations** is checked.
9. **Save the Tab**
 - Click **Save** to create the tab for "Total Laptops".

The screenshot shows the Salesforce Setup interface for creating a new custom object tab. At the top, there's a blue header bar with the word 'SETUP' and a 'Tabs' button. Below the header, the title 'New Custom Object Tab' is displayed. A blue navigation bar at the top of the main content area says 'Step 1. Enter the Details'. The main body contains the following fields:

- A note: 'Choose the custom object for this new custom tab. Fill in other details.'
- A dropdown labeled 'Object' set to 'Total Laptops'.
- A 'Tab Style' section with a 'Laptop' icon selected.
- An optional field for a 'Splash Page Custom Link' with a dropdown set to '--None--'.
- A text input field for a 'Description'.

Activity 2: Creating Tabs for Remaining Objects ("Consumer", "Laptop Booking", "Billing Process")

Follow the same steps as outlined in Activity 1 to create tabs for the following objects:

1. Consumer

- Go to Setup > Tabs > New
- Select **Consumer** as the object.
- Choose a tab style and click **Next**.
- Keep the default settings for profiles and click **Next**.
- Uncheck **Include Tab** for custom apps, and ensure **Append tab to users' existing personal customizations** is checked.
- Click **Save**.

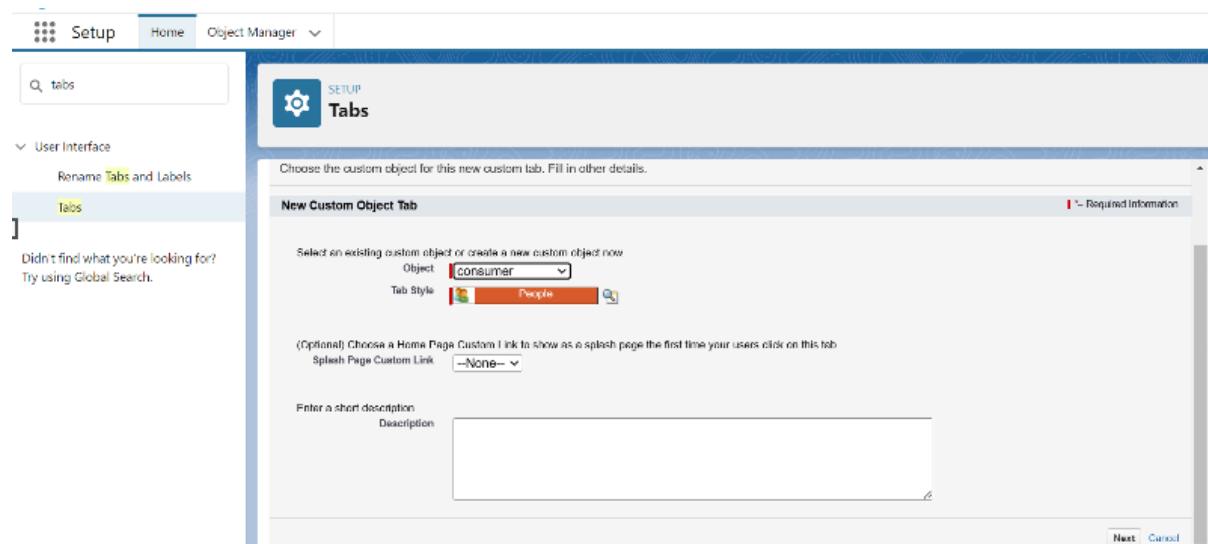
2. Laptop Booking

- Repeat the steps for the **Laptop Booking** object.

3. Billing Process

- Repeat the steps for the **Billing Process** object.

This will create the custom tabs for the **Consumer**, **Laptop Booking**, and **Billing Process** objects.



SETUP

Tabs

Choose the custom object for this new custom tab. Fill in other details.

New Custom Object Tab

Select an existing custom object or [create a new custom object now](#).

Object: Billing Process

Tab Style: Shopping Cart

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link: —None—

Enter a short description.

Description:

Next **Cancel**

SETUP

Tabs

Choose the custom object for this new custom tab. Fill in other details.

New Custom Object Tab

Select an existing custom object or [create a new custom object now](#).

Object: Laptop Bookings

Tab Style: Presenter

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link: —None—

Enter a short description.

Description:

Next **Cancel**

SETUP

Tabs

Custom Tabs

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

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

Custom Object Tabs

Action	Label	Tab Style	Description
Edit Del	Billing Process	Shopping Cart	
Edit Del	Consumer	People	
Edit Del	Laptop Bookings	Presenter	
Edit Del	Total Laptops	Laptop	

The Lightning App:

A **Lightning App** is a collection of components and tools bundled together to serve a specific function within Salesforce. In **Lightning Experience**, these apps offer a streamlined user interface, allowing users to access sets of objects, tabs, and other resources conveniently from the navigation bar.

Key Features:

1. **Customization:** Lightning apps can be personalized with custom colors and logos, helping align the app with your organization's branding.
2. **Enhanced Efficiency:** Users can switch easily between apps, improving workflow and productivity.
3. **Additional Features:** You can include features like a utility bar and Lightning page tabs, adding more functionality to the app.

In essence, Lightning apps provide a flexible and efficient way for users to manage tasks and access tools within Salesforce, all from a unified interface.

Create A Lightning App:

Steps to Create a Lightning App for "Laptop Rentals"

1. **Navigate to the Setup Page**
 - Click on the **Gear Icon** and select **Setup**.
2. **Search for App Manager**
 - In the **Quick Find** search bar, type **App Manager**.
 - Click on **App Manager** in the search results.
3. **Create a New Lightning App**
 - Click on **New Lightning App** to start the creation process.
4. **Fill in App Details**
 - **App Name:** Enter *LAPTOP RENTALS*.
 - Click **Next** to proceed.
5. **App Option Page**
 - Leave the settings as default and click **Next**.
6. **Utility Items**
 - Keep the default utility items and click **Next**.
7. **Upload a Photo**
 - Upload a relevant photo that represents your app.
8. **Add Navigation Items**
 - From the search bar, select the following items:
 - **Total Laptops**

- Consumer
- Laptop Booking
- Billing Process

- Use the arrow button to add these items to your app.
- Click **Next** to proceed.

9. Add User Profiles

- In the search bar, type **System Administrator**.
- Select the profile and move it using the arrow button.
- Click **Save & Finish** to complete the creation of the Lightning app.

Your **Laptop Rentals** Lightning app is now successfully created.

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

* Developer Name

Description

App Branding

Image

Primary Color Hex Value

Next

New Lightning App

App Options

Navigation and Form Factor

* Navigation Style Standard navigation Console navigation

* Supported Form Factors Desktop and phone Desktop Phone

Setup and Personalization

Setup Experience Setup (full set of Setup options) Service Setup

App Personalization Settings

Disable end user personalization of nav items in this app

Disable temporary tabs for items outside of this app

Use Omni-Channel sidebar

Next

New Lightning App

Utility Items (Desktop Only)

Give your users quick access to productivity tools and add background utility items to your app.

Add Utility Item

Utility Bar Alignment

The utility bar is a fixed footer that opens components in docked panels. Available only when the app is viewed in Lightning Experience on a desktop.

Next

New Lightning App

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Selected Items

Back Next

New Lightning App

User Profiles

Choose the user profiles that can access this app.

Available Profiles

Selected Profiles

Back Save & Finish

SETUP

Lightning Experience App Manager

New Lightning App New Connected App

24 items • Sorted by App Name • Filtered by All appmenuitems • Tabset type: App type

App Name	Developer Name	Description	Last Modified	Ap...	Vi...
Analytics Studio	Insights	Build CRM Analytics dashboards and apps	18/11/2024, 9:49 am	Classic	✓
App Launcher	AppLauncher	App Launcher tabs	18/11/2024, 9:49 am	Classic	✓
Automation	FlowsApp	Automate business processes and repetitive tas...	18/11/2024, 9:55 am	Lightning	✓
Bolt Solutions	LightningBolt	Discover and manage business solutions design...	18/11/2024, 9:52 am	Lightning	✓
Business Rules Engi...	ExpressionSetConsole	Create and maintain business rules that perform...	18/11/2024, 9:49 am	Lightning	✓
Community	Community	Salesforce CRM Communities	18/11/2024, 9:49 am	Classic	✓
Content	Content	Salesforce CRM Content	18/11/2024, 9:49 am	Classic	✓
Data Manager	DataManager	Use Data Manager to view limits, monitor usage...	18/11/2024, 9:49 am	Lightning	✓
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	18/11/2024, 9:49 am	Lightning	✓
LAPTOP RENTALS	LAPTOP_RENTALS		18/11/2024, 11:06 am	Lightning	✓

Fields:

In Salesforce, **Fields** represent the individual pieces of data stored in the columns of a relational database. They hold valuable information specific to an object, which makes searching, deleting, and editing records more efficient and faster.

Fields in Salesforce can be broadly categorized into two types:

Standard Fields and **Custom Fields**.

Standard Fields

Standard Fields are predefined fields provided by Salesforce for common data requirements. These fields are included in every Salesforce application and serve standard functions. Some of the most commonly used standard fields include:

- **Created By:** Tracks who created the record.
- **Owner:** Identifies the owner of the record.
- **Last Modified:** Shows the last date and time the record was modified.
- **Field Made During Object Creation:** Refers to fields automatically created during the setup of the object.

One important feature of Standard Fields is that they cannot be deleted unless they are non-required fields. Users may remove these fields if they are not critical for the application, but deletion is restricted for certain required fields.

Custom Fields

Unlike Standard Fields, **Custom Fields** are highly flexible and can be created by users to cater to specific business requirements. Custom Fields are not predefined by Salesforce, and their usage depends on the needs of the organization. These fields allow for customization of records to store additional information that is relevant to the business but not covered by standard fields.

The main advantages of Custom Fields are:

- They can be added or removed at any time based on the user's needs.
- Custom Fields provide businesses with the flexibility to tailor Salesforce to their unique processes.

In summary, Salesforce offers a balance between **Standard Fields** for general functionality and **Custom Fields** for specific, customizable data requirements, allowing businesses to manage and customize their data storage effectively.

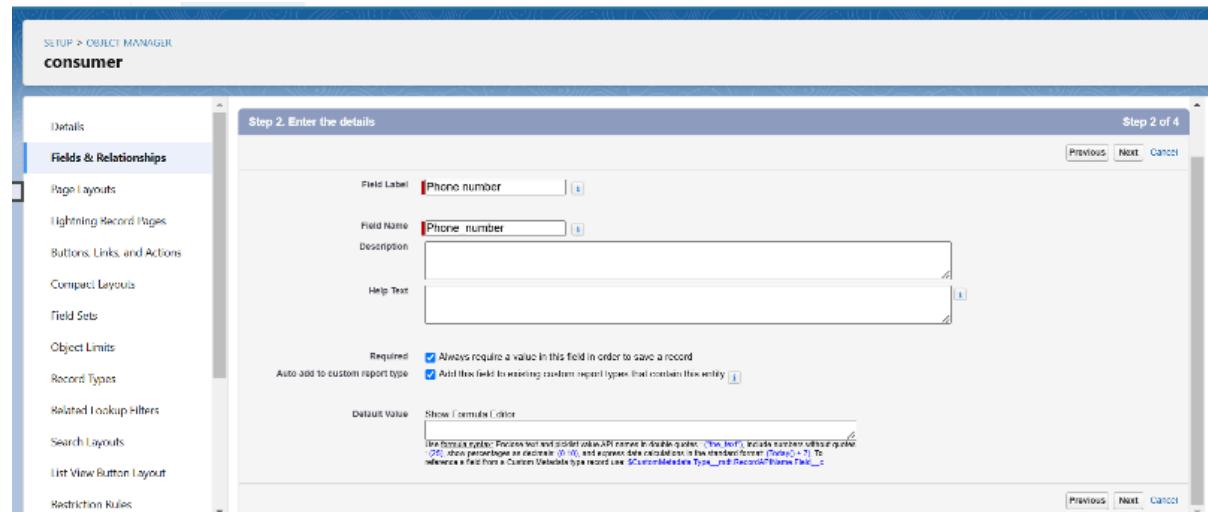
Creating the Fields in Consumer Object:



The screenshot shows the Salesforce Setup - Object Manager interface. At the top, there are tabs for 'Setup', 'Home', and 'Object Manager'. A search bar contains the text 'consumer'. Below the search bar, there is a 'Schema Builder' button and a 'Create' dropdown menu. The main area displays a table for the 'consumer' object. The columns are labeled 'LABEL', 'API NAME', 'TYPE', 'DESCRIPTION', 'LAST MODIFIED', and 'DEPLOYED'. One row is shown, with 'consumer' as the label, 'consumer_c' as the API name, 'Custom Object' as the type, and the last modified date as '18/11/2024'.

1. Create Phone Number Field

1. **Navigate to Setup**
 - Click on the Gear Icon and select **Setup**.
2. **Search for Consumer Object**
 - In the **Quick Find** search bar, type **Object Manager**.
 - Click on **Object Manager**.
 - Search for and select the **Consumer** object.
3. **Add a New Field**
 - In the **Fields & Relationships** section, click **New**.
4. **Select Data Type**
 - Choose **Phone** as the data type and click **Next**.
5. **Fill in Field Details**
 - **Field Label:** Enter *Phone number*.
 - **Field Name:** This will auto-generate.
 - Check the **Required** option box.
6. **Save the Field**
 - Click **Next**, then **Next**, and finally **Save and New**.



The screenshot shows the 'Step 2: Enter the details' screen for creating a new field. The left sidebar lists various object settings like 'Details', 'Fields & Relationships', 'Page Layouts', etc. The main form has the following fields:

- Field Label:** Phone number
- Field Name:** Phone_number
- Description:** (empty)
- Help Text:** (empty)
- 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 entry
- Default Value:** (empty)
- Show Formula Editor:** (checkbox)

A note at the bottom states: 'Use formula editor: Allows you to edit default values API names in double quotes: "Phone_number". Include numbers without quotes. e.g., always require a value in field "Phone_number" and exclude the value in field "Phone_number" To include a formula in a custom update type, replace the "Phone_number" placeholder value with "Phone_number".'

2. Create Email Field

1. Navigate to Setup

- Go to **Setup** and search for the **Consumer** object in the **Object Manager**.

2. Add a New Field

- Click on **Fields & Relationships** and then **New**.

3. Select Data Type

- Choose **Email** and click **Next**.

4. Fill in Field Details

- **Field Label:** Enter *Email*.
- **Field Name:** This will auto-generate.

5. Save the Field

- Click **Next**, then **Next**, and finally **Save and New**.

The screenshot shows the 'Step 2. Enter the details' page for creating a new custom field. The field label is set to 'Email'. The field name is also 'Email'. There are input fields for 'Description' and 'Help Text'. Under settings, 'Required' is checked, while 'Unique' and 'External ID' are unchecked. A checkbox for 'Auto add to custom report type' is checked, with a note about adding to existing report types. At the bottom, there are buttons for 'Default Value' and 'Show Formula Editor'.

3. Create Address Field

1. Navigate to Setup

- Go to **Setup** and search for the **Consumer** object in the **Object Manager**.

2. Add a New Field

- Click on **Fields & Relationships** and then **New**.

3. Select Data Type

- Choose **Text Area** and click **Next**.

4. Fill in Field Details

- **Field Label:** Enter *Address*.
- **Field Name:** This will auto-generate.
- Check the **Required** option box.

5. Save the Field

- Click **Next**, then **Next**, and finally **Save and New**.

The screenshot shows the 'Step 2. Enter the details' screen for creating a new field. The 'Field Label' is set to 'Address'. The 'Field Name' is also 'Address'. The 'Required' checkbox is checked. Other fields like 'Description', 'Help Text', and 'Default Value' are present but empty. A note at the bottom explains the use of formula editor for default values.

4. Create Consumer Status Field

1. Navigate to Setup

- Go to **Setup** and search for the **Consumer** object in the **Object Manager**.

2. Add a New Field

- Click on **Fields & Relationships** and then **New**.

3. Select Data Type

- Choose **Picklist** and click **Next**.

4. Fill in Field Details

- **Field Label:** Enter *Consumer Status*.
- **Value:** Enter the following values, each separated by a new line:
 - Student
 - Employee
 - Others
- **Field Name:** This will auto-generate.
- Check the **Required** option box.

5. Save the Field

- Click **Next**, then **Next**, and finally **Save and New**.

The screenshot shows the 'Step 2. Enter the details' screen for the consumer status field. The 'Field Label' is 'consumer Status'. The 'Values' section shows three entries: 'Student', 'Employee', and 'Others'. The 'Field Name' is 'consumer_Status'. The 'Required' checkbox is checked. Other fields like 'Description', 'Help Text', and 'Default Value' are empty.

Creating Fields in Laptop Booking Object:

Object Manager		Search Bar: Laptop		Schema Builder	Create
Label	API Name	Type	Description	Last Modified	Deployed
Laptop Bookings	Laptop_Bookings_c	Custom Object		10/11/2024	✓

1. Create Laptop Brand Field (Picklist)

1. Go to Setup
 - Click on the **Gear Icon** and select **Setup**.
2. Search for the Laptop Bookings Object
 - In the **Quick Find** search bar, type **Object Manager**.
 - Click on **Object Manager**.
 - Search for and select the **Laptop Booking** object.
3. Add a New Field
 - In the **Fields & Relationships** section, click **New**.
4. Select Data Type
 - Choose **Picklist** as the data type and click **Next**.
5. Define Picklist Values
 - Enter the following values:
 - Dell
 - Acer
 - HP
 - Mac
 - Check the **Required** option box.
6. Save the Field
 - Click **Next**, then **Next**, and finally **Save and New**.

The screenshot shows the Salesforce Object Manager interface for the 'Laptop Bookings' object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The 'Fields & Relationships' tab is selected. In the main pane, a new field is being created. The 'Field Label' is set to 'laptop names'. The 'Values' section has a radio button selected for 'Enter values, with each value separated by a new line', and below it, a text area contains the values 'Dell', 'Acer', 'Hp', and 'Mac'. There are also checkboxes for 'Display values alphabetically, not in the order entered' and 'Use first value as default value'. At the bottom, the 'Field Name' is 'Laptop_type', 'Description' is empty, and 'Help Text' is empty. Navigation buttons 'Previous', 'Next', and 'Cancel' are visible at the top right of the form.

2. Create Processor Type Field (Picklist)

1. Go to Setup
 - Go to **Setup** and search for the **Laptop Booking** object in the **Object Manager**.
2. Add a New Field
 - Click on **Fields & Relationships** and then **New**.
3. Select Data Type
 - Choose **Picklist** and click **Next**.
4. Define Picklist Values
 - Enter the following values:
 - Core i3
 - Core i5
 - Core i7
5. Check the **Required** box and click **Next**, then **Next** again.
6. Save the Field
 - Click **Save and New**.

The screenshot shows the 'Object Manager' interface in Salesforce. A new picklist field is being created for the 'Laptop Bookings' object. The 'Field Label' is 'core type'. In the 'Values' section, the 'Enter values, with each value separated by a new line' option is selected, and the values listed are 'core i3', 'core i5', 'core i7', and 'Bionic chip'. The 'Field Name' is set to 'Core'. Other settings include 'Description' and 'Help Text' fields, and checkboxes for 'Required', 'Always require a value in this field in order to save a record', and 'Add this field to existing custom report types that contain this entity'.

Note:

Field Dependency

- A **field dependency** refers to a relationship between two fields, where the values in one field determine the available values in another field.
- This is commonly used for picklist relationships, where the dependent picklist's options change based on the selection made in the controlling picklist.
- Using field dependencies allows for dynamic picklist values based on other field selections.

To Create Fields and Relationships to a Laptop Booking Object:

1. Go to Setup

- Click on the **Gear Icon** in the top-right corner and select **Setup**.

2. Search for Laptop Booking Object

- In the **Quick Find** search bar, type **Object Manager**.
- Click on **Object Manager**.
- Search for and select the **Laptop Booking** object.

3. Add Field Dependency

- In the **Fields & Relationships** section, click on **Field Dependencies**.
- Click **New** to create a new field dependency.

4. Select Controlling and Dependent Fields

- In the **Controlling Field** dropdown, select the **Laptop Brand** field (Picklist).
- In the **Dependent Field** dropdown, select the **Processor Type** field (Picklist).
- Click **Next**.

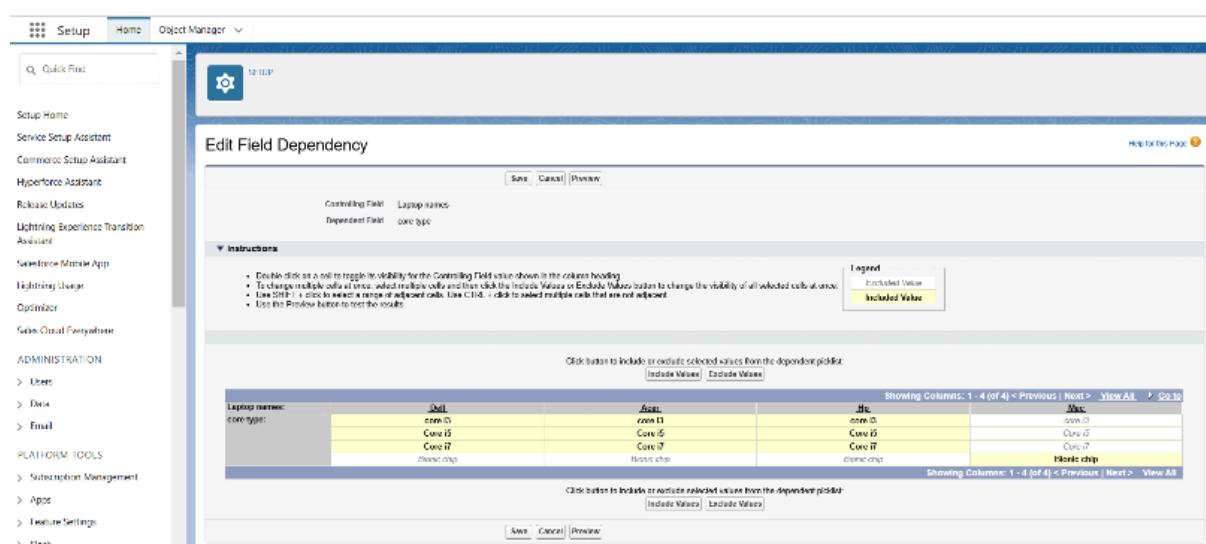
5. Define Field Dependency Values

- For each laptop brand, define the corresponding processor values:
 - Dell**: Core i3, Core i5, Core i7
 - Acer**: Core i3, Core i4, Core i5
 - HP**: Core i3, Core i4, Core i5
 - Mac**: Bionic Chip (Include values for it)

6. Save the Field Dependency

- Click **Save** to save the field dependency.

This sets up the relationship between the **Laptop Brand** and **Processor Type** fields, controlling the available processor options based on the laptop brand selected.



1. Create Lookup Relationship with Consumer Object

- **Go to Setup:** Click on the **Gear Icon** in the top-right corner and select **Setup**.
- **Search for Object:** Type **Laptop Booking** in the **Quick Find** search bar and click on the object.
- **Click on "Fields & Relationships":** In the **Fields & Relationships** section, click on **New**.
- **Select Data Type:** Choose **Lookup Relationship** and click **Next**.
- **Select Related Object:** In the **Related To** dropdown, select **Consumer** and click **Next**.
- **Field Label and Name:**
 - **Field Label:** Name
 - **Field Name:** Auto-generated
- **Click Next:** Click **Next** and **Save**.

Laptop Bookings
New Relationship

Step 3 of 6

Field Label: Name

Field Name: Name

Description:

Help Text:

Child Relationship Name: Laptop_Bookings

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

What to do if the lookup record is deleted? Clear the value of this field. You can't choose this option if you make this field required.

Don't allow deletion of the lookup record that's part of a lookup relationship.

Auto add to currency report types: Add this field to existing currency report types from entities in this entity.

Lookup Filter

2. Create Currency Field for Amount

- **Go to Setup:** Click on **Setup** and search for **Laptop Booking**.
- **Click on "Fields & Relationships":** Click on **New**.
- **Select Data Type:** Choose **Currency** and click **Next**.
- **Field Label and Name:**
 - **Field Label:** Amount
 - **Length:** 18,0
 - **Field Name:** Auto-generated
- **Click Next:** Click **Next** and **Save**.

Laptop Bookings
New Custom Field

Step 2 of 4

Field Label: [Help](#)

Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".

Length: Number of digits to the left of the decimal point

Decimal Places: Number of digits to the right of the decimal point

Field Name: [Help](#)

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. [Help](#)

3. Create Lookup Relationship with Total Laptops Object

- Go to Setup:** Click on **Setup** and search for **Laptop Booking**.
- Click on "Fields & Relationships":** Click on **New**.
- Select Data Type:** Choose **Lookup Relationship** and click **Next**.
- Select Related Object:** In the **Related To** dropdown, select **Total Laptops** and click **Next**.
- Field Label and Name:**
 - Field Label:** Total No of Laptops
 - Field Name:** Auto-generated
- Click Next:** Click **Next** and **Save**.

Laptop Bookings
New Relationship

Step 3 of 8

Field Label: [Help](#)

Field Name: [Help](#)

Description:

Help Text:

Child Relationship Name: [Help](#)

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

What to do if the lookup record is deleted?

- Clear the value of this field. You can't choose this option if you make this field required.
- Don't allow deletion of the lookup record that's part of a lookup relationship.

Auto add to custom report type: Add this field to existing custom report types that contain this entity. [Help](#)

4. Create Email Field

- Go to Setup:** Click on **Setup** and search for **Laptop Booking**.
- Click on "Fields & Relationships":** Click on **New**.
- Select Data Type:** Choose **Email** and click **Next**.
- Field Label and Name:**
 - Field Label:** Email
 - Field Name:** Auto-generated
- Click Save:** Click **Save**.

Laptop Bookings
New Custom Field

Step 2: Enter the details

Field Label: email

Field Name: email

Description:

Help Text:

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

Unique: Do not allow duplicate values

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

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

Default Value: None

Use formula editor. Previous field and field value must be double quotes ("), or you can replace it with a variable. To use auto-generate value, click on the formula editor icon ("), or reference a field from a custom database record using \$C{variableName}. See Help for more details.

5. Populate Records and Set Lookup Relationship

- After saving records in **Consumer** and **Laptop Booking**, go to **Laptop Booking** and edit the **Lookup** field to establish a detailed relationship.

6. Create Roll-up Summary Field in Total Laptops Object

- Go to Setup:** Click on **Setup** and search for **Total Laptops**.
- Click on "Fields & Relationships":** Click on **New**.
- Select Data Type:** Choose **Roll-up Summary** and click **Next**.
- Field Label and Name:**
 - Field Label:** Laptops Delivered
 - Field Name:** Auto-generated
- Summarized Object:** Choose **Laptop Bookings**.
- Roll-up Type:** Select **Count**.
- Click Next:** Click **Next** and **Save**.

SETUP > OBJECT MANAGER
Total Laptops

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Custom Fields

Total Laptops
New Custom Field

Step 2: Enter the details

Field Label: Laptops delivered

Field Name: Laptops_delivered

Description:

Help Text:

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

7. Create Formula Field for Laptops Available

- Go to Setup:** Click on **Setup** and search for **Laptop Booking**.
- Click on "Fields & Relationships":** Click on **New**.

- **Select Data Type:** Choose **Formula** and click **Next**.
- **Field Label and Name:**
 - **Field Label:** Laptops Available
 - **Field Name:** Auto-generated
- **Formula Return Type:** Choose **Number**.
- **Decimal Places:** Select **0**.
- **Advanced Formula:**
Enter the formula: `50 - Total_no_of_laptops__r.Laptops_delivered__c`
- Click **Insert Field** and select **Total No Of Laptops** and **Laptops Delivered**.
- **Click Next:** Click **Next** and **Save**.

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

Insert Field: Laptop available (Number) =
50 - Total_no_of_laptops__r.laptops_delivered__c

Insert Operator:

Functions: ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN

Check Syntax: No syntax errors in merge fields or functions (Compiled size: 36 characters)

8. Create Picklist Field

- **Go to Setup:** Click on **Setup** and search for **Laptop Booking**.
- **Click on "Fields & Relationships":** Click on **New**.
- **Select Data Type:** Choose **Picklist** and click **Next**.
- **Picklist Values:** Enter values like 1, 2, 3, 4, 5.
- **Click Save:** Click **Save**.

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

Values: Use global picklist value set Enter values, with each value separated by a new line

1
2
3
4
5

Display values alphabetically, not in the order entered
Use first value as default value

Field Name: how_many_months

Description:

Help Text:

Required: Always require a value in this field in order to save a record
 Add this field to existing custom report types that contain this entity

Creation of Fields and Relationships for Billing Process Object:

1. Create Master-detail Relationship with Consumer Object

- **Go to Setup:** Click on **Setup** and search for **Billing Process** in the **Quick Find** search bar.
- **Click on Object Manager:** Click on **Billing Process**.
- **Click on "Fields & Relationships":** Click **New**.
- **Select Data Type:** Choose **Master-detail Relationship** and click **Next**.
- **Select Related Object:** In the **Related To** dropdown, select **Consumer** and click **Next**.
- **Field Label and Name:**
 - **Field Label:** Name
 - **Field Name:** Auto-generated
- **Click Next:** Click **Next** and **Save**.

The image consists of three screenshots from the Salesforce setup interface, illustrating the process of creating a master-detail relationship between the Billing Process object and the Consumer object.

- Screenshot 1: Step 2. Choose the related object**

This screenshot shows the "Step 2. Choose the related object" screen. The "Related To" dropdown menu is open, and "consumer" is selected. Other options visible in the dropdown include: Calendar Model, Campaign, Cart, Cart Item, Case, Case Related Issue, Change Request, Change Request Related Issue, Change Request Related Item, Communication Subscription, Communication Subscription Channel Type, Communication Subscription Consent, Communication Subscription Timing, Contact, Contact Point Consent, and Contact Point Type Consent. The "Billing Process New Relationship" page title is at the top, and the left sidebar shows the "Fields & Relationships" tab is selected.
- Screenshot 2: Step 3. Enter the label and name for the lookup field**

This screenshot shows the "Step 3. Enter the label and name for the lookup field" screen. It includes fields for "Field Label" (set to "Name"), "Field Name" (set to "Name"), "Description" (empty), and "Help Text" (empty). Below these, under "Sharing Settings", the "Read/Write" checkbox is selected. At the bottom, there are checkboxes for "Allow re-parenting" and "Auto add to parent record type".
- Screenshot 3: Step 4 of 6**

This screenshot shows the final step of the wizard, "Step 4 of 6". It displays a summary of the settings chosen in the previous steps. The "Field Label" is set to "Name", "Field Name" is set to "Name", and the "Child Relationship Name" is set to "Billing_Process". The "Sharing Settings" section shows "Read/Write" is selected. The "Allow re-parenting" and "Auto add to parent record type" checkboxes are also present.

2. Create Lookup Relationship with Laptop Booking Object

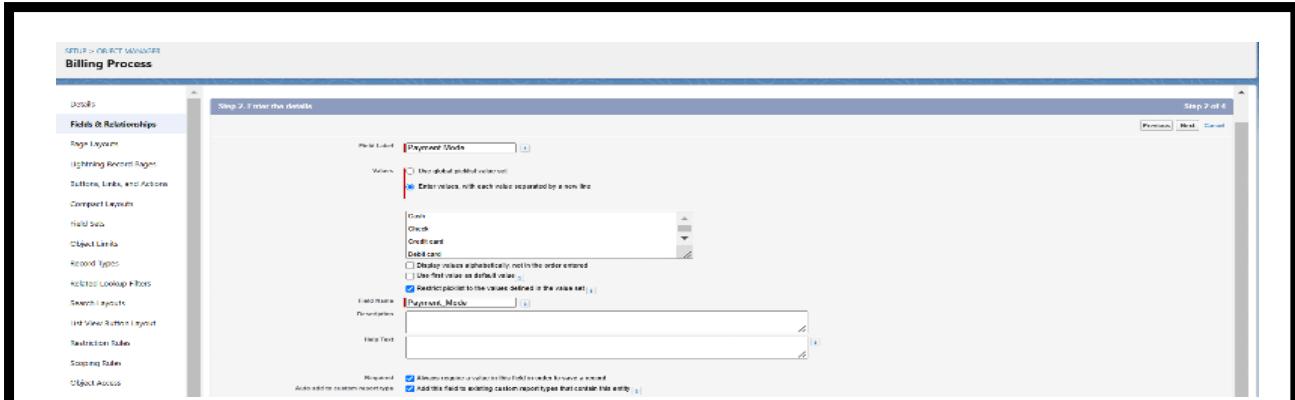
- **Go to Setup:** Click on **Setup** and search for **Billing Process**.
- **Click on "Fields & Relationships":** Click **New**.
- **Select Data Type:** Choose **Lookup Relationship** and click **Next**.
- **Select Related Object:** In the **Related To** dropdown, select **Laptop Booking** and click **Next**.
- **Field Label and Name:**
 - **Field Label:** Laptop Booking
 - **Field Name:** Auto-generated
- **Click Next:** Click **Next** and **Save**.

The screenshot shows the 'New Relationship' setup page in Salesforce. It's Step 3 of 8. The 'Field Label' is set to 'Laptop Booking'. The 'Field Name' is 'Laptop_Booking'. There is a 'Description' field and a 'Help Text' field, both of which are currently empty. Under 'Child Relationship Name', it says 'Billing_Process'. Below this, there are several checkboxes:

- Always require a value in this field in order to save a record.
- Clear the value of this field. You can't choose this option if you make this field required.
- Don't allow deletion of the lookup record that's part of a lookup relationship.
- Add this field to existing custom report types that contain this entity.

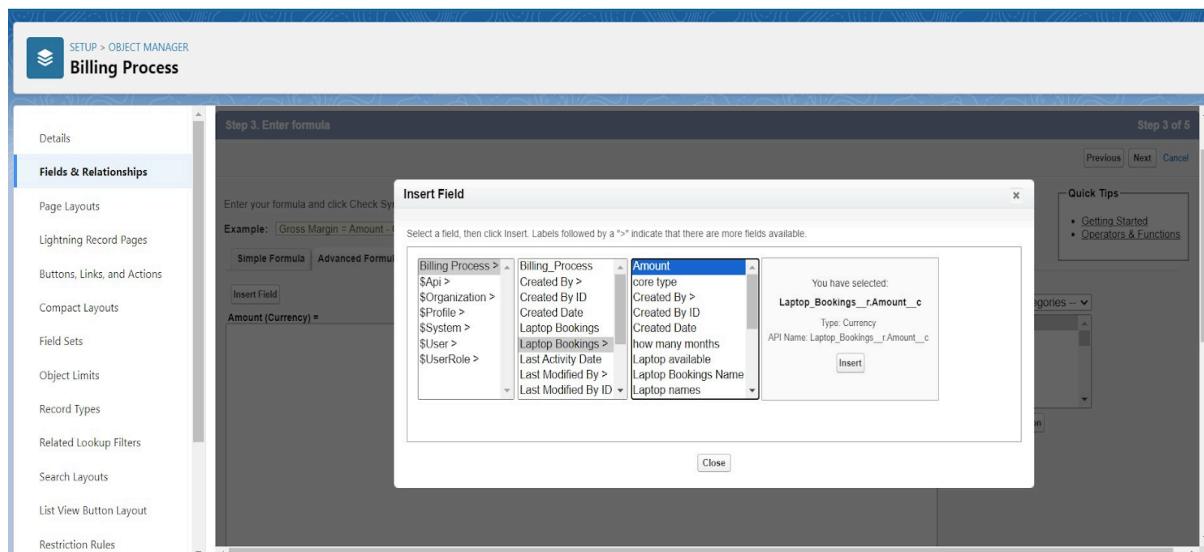
3. Create Picklist Field for Payment Mode

- **Go to Setup:** Click on **Setup** and search for **Billing Process**.
- **Click on "Fields & Relationships":** Click **New**.
- **Select Data Type:** Choose **Picklist** and click **Next**.
- **Field Label and Name:**
 - **Field Label:** Payment Mode
 - **Values:**
 - Cash
 - Check
 - Credit Card
 - Debit Card
 - UPI
 - PhonePe
 - GPay
 - Paytm
- **Click Next:** Click **Next** and **Save**.



4. Create Cross Object Formula Field in Billing Process Object

- **Go to Setup:** Click on **Setup** and search for **Billing Process**.
- **Click on "Fields & Relationships":** Click **New**.
- **Select Data Type:** Choose **Formula** and click **Next**.
- **Field Label and Name:**
 - **Field Label:** Amount
 - **Field Name:** Auto-generated
 - **Formula Return Type:** Number
- **Advanced Formula:**
 - Click **Insert Field** and select **Billing Process** from the first dropdown.
 - In the second dropdown, select **Laptop Booking**.
 - In the third dropdown, select **Amount** field and click **Insert**.
 - The formula: `Laptop_Booking__r.Amount__c`
- **Check Syntax:** Verify there are no syntax errors.
- **Click Next:** Click **Next** and **Save**.



Creating Fields in Total Laptops Object:

1. Go to Setup and Locate Total Laptops Object

- **Go to Setup:** Click on **Setup** and search for **Total Laptops** in the **Quick Find** search bar.
- **Click on Object Manager:** Click on **Total Laptops**.

2. Create Formula Field for Laptops Available

- **Click on "Fields & Relationships":** In the **Total Laptops** object, click on **Fields & Relationships** and then click **New**.
- **Select Data Type:** Choose **Formula** and click **Next**.

3. Fill in Field Details

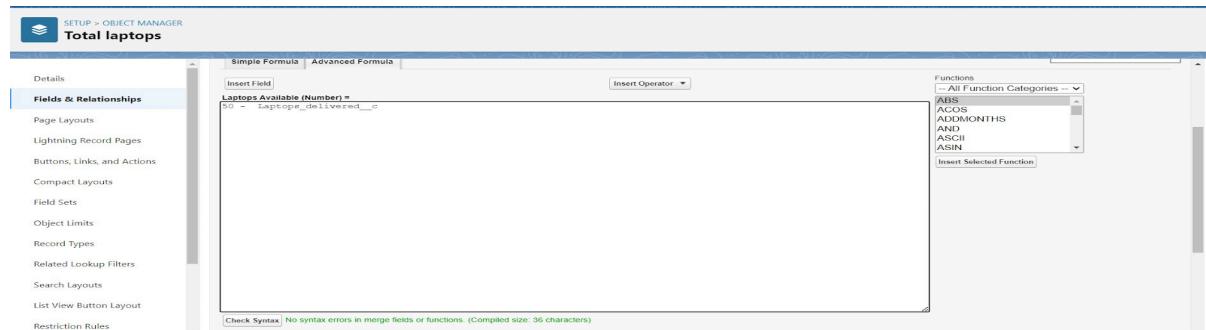
- **Field Label:** **Laptops Available**
- **Field Name:** Auto-generated
- **Formula Return Type:** **Number**
- **Decimal Places:** Set as **0**

4. Enter Formula

- **Advanced Formula:** Click on **Advanced Formula** and enter the following formula:
50 - Laptops_delivered__c
- **Check Syntax:** Click **Check Syntax** to ensure there are no errors.

5. Save the Field

- **Click Next:** Click **Next**.
- **Click Save and New:** Finally, click **Save and New** to create another field or complete the process.



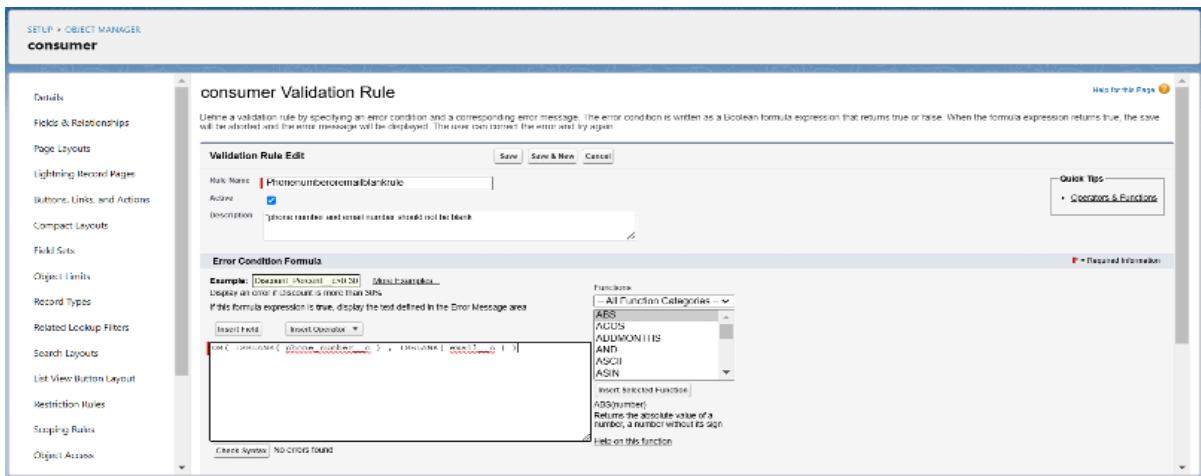
Validation Rule:

A **Validation Rule** in Salesforce is a mechanism used to enforce data quality by ensuring that users enter data in a record that meets specific criteria. When a user attempts to save a record, the rule evaluates the data entered in one or more fields based on a formula or expression. If the data doesn't meet the criteria, the rule triggers an error message and prevents the record from being saved until the issue is resolved. The formula used in the validation rule returns a **True** or **False** value, and if the result is **True**, the error message is displayed to the user.

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

1. **Go to Setup**
 - Navigate to **Setup** in Salesforce.
2. **Click on Object Manager**
 - In the Setup page, click on **Object Manager**.
3. **Select the Consumer Object**
 - From the Object Manager, locate and click on the **Consumer** object.
4. **Edit the Consumer Object**
 - In the **Consumer** object, click on **Edit**.
5. **Access Validation Rules**
 - On the Consumer object page, click on **Validation Rules**.
6. **Create New Validation Rule**
 - Click on **New** to create a new validation rule.
7. **Enter Rule Name**
 - **Rule Name:** Enter **Phonenumberoremailblankrule**.
8. **Enter Description**
 - **Description:** Enter **Phone number and email should not be blank**.
9. **Enter Formula**
 - **Formula:** Enter the following formula:
`OR(ISBLANK(phone_number__c), ISBLANK(email__c))`
10. **Check Syntax**
 - Click on **Check Syntax** to validate the formula.
11. **Save the Validation Rule**
 - Once the syntax is correct, click **Save** to create the validation rule.

This validation rule ensures that neither the **Phone Number** nor the **Email** field in the **Consumer** object can be left blank when creating or editing a record.



Profiles:

A **Profile** in Salesforce is a collection of settings and permissions that determine what a user can see, access, and do in Salesforce. It controls user interactions with **objects**, **fields**, **tabs**, **apps**, and other Salesforce features. Profiles are essential for managing access based on roles and responsibilities, ensuring appropriate data security and functionality for each user.

Types of Profiles in Salesforce

1. Standard Profiles

- **Definition:** Predefined profiles provided by Salesforce by default.
- **Key Characteristics:**
 - Cannot be deleted or modified significantly.
 - Includes a default set of permissions for standard objects.
- **Examples:**
 - **Contract Manager**
 - **Read Only**
 - **Marketing User**
 - **Solutions Manager**
 - **Standard User**
 - **System Administrator**

2. Custom Profiles

- **Definition:** Profiles created by users to meet specific business requirements.
- **Key Characteristics:**
 - Fully customizable to control access for custom objects, apps, and tabs.
 - Can be deleted if no users are assigned to them.

Owner Profile:

- **Navigate to Profiles**
 - Go to **Setup**.
 - In the **Quick Find Box**, type **Profiles** and click on it.
- **Clone an Existing Profile**
 - Locate the desired profile to clone (e.g., **Standard User**).
 - Click on the **Clone** button next to the profile.
 - Enter the **Profile Name** (e.g., **Owner**) and click **Save**.
- **Edit Custom Object Permissions**
 - Scroll down to the **Custom Object Permissions** section.
 - Find the objects you want to configure permissions for (e.g., **Total Laptops, Consumers, Laptop Booking, Billing Process**).
 - Select the required permissions (e.g., **Read, Write, Create, Delete**, etc.).
- **Save Changes**
 - After configuring the permissions, click **Save** to apply the changes.

The screenshot shows the Salesforce Setup interface. At the top, there's a navigation bar with a user icon, 'SETUP', and 'Profiles'. Below this is a modal window titled 'Clone Profile' with the sub-header 'Enter the name of the new profile.' It contains a message 'You must select an existing profile to clone from.' and fields for 'Existing Profile' (set to 'Standard User'), 'User License' (set to 'Salesforce'), and 'Profile Name' (set to 'owner'). At the bottom of the modal are 'Save' and 'Cancel' buttons. The background shows the 'Profiles' page with sections for 'Custom Object Permissions' (listing various objects like Ideas, Images, Incidents, Individuals, Inventory Reservations, Work Plans, etc.) and 'Session Settings' and 'Password Policies'.

Agent Profile:

- **Navigate to Profiles**
 - Go to **Setup**.
 - In the **Quick Find Box**, type **Profiles** and click on it.
- **Clone an Existing Profile**
 - Locate the desired profile to clone (e.g., **Standard Platform User**).
 - Click on the **Clone** button next to the profile.
 - Enter the **Profile Name** (e.g., **Agent**) and click **Save**.
- **Edit the Profile**
 - While still on the **Profile Details** page, click **Edit**.
- **Configure Custom Object Permissions**
 - Scroll down to the **Custom Object Permissions** section.
 - Locate the objects (e.g., **Total Laptops**, **Consumers**, **Laptop Bookings**, **Billing Process**).
 - Assign the appropriate permissions (e.g., **Read**, **Write**, **Create**, **Delete**, etc.) based on requirements.
- **Save Changes**
 - After configuring the permissions, click **Save** to finalize the settings.

The top screenshot shows the 'Clone Profile' dialog. It has a message 'You must select an existing profile to clone from.' Below it, 'Existing Profile' is set to 'Standard Platform User', 'User License' is 'Salesforce Platform', and 'Profile Name' is 'Agent'. At the bottom are 'Save' and 'Cancel' buttons.

The bottom screenshot shows the 'Profiles' page under 'SETUP'. The 'Custom Object Permissions' section is active, showing checkboxes for various objects. The 'Basic Access' and 'Data Administration' columns are visible for objects like 'Billing Process' and 'consumer'. Other sections like 'Session Settings' and 'Password Policies' are also present at the bottom.

Roles and Hierarchy:

A role in Salesforce defines a user's record-level visibility and access, specifying what data they can view and interact with based on their position within the organization. Roles work in conjunction with the role hierarchy, enabling users in higher roles to access data owned by those in lower roles. This structure ensures data security and aligns with the organization's reporting and operational needs.

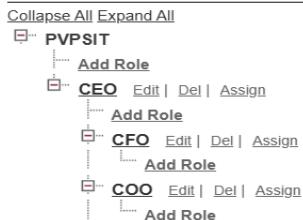
Creating Owner Role:

- **Access Roles Setup:**
 - Go to the **Quick Find** search bar in Setup.
 - Type **Roles** and click on **Set Up Roles**.
- **Expand Roles Hierarchy:**
 - Click on the **Plus (+)** icon next to the **CEO** role to expand the hierarchy.
- **Add New Roles Under Owner:**
 - Click **Add Role** under the **Owner** role.
- **Define Role Details for Agents:**
 - For the first Agent role:
 - Enter the **Label** as **Agent 1** (or **Agent**, as applicable).
 - The **Role Name** will be auto-populated.
 - Click **Save**.
 - Repeat the process for additional Agent roles (e.g., Agent 2).
- **Save Each Role:**
 - Click **Save** after defining the details for each role.

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy



Role Edit New Role

Role Edit	
Label	owner
Role Name	owner
This role reports to	CEO
Role Name as displayed on reports	

Save **Save & New** **Cancel**

Role Edit New Role

Role Edit	
Label	Agent
Role Name	Agent
This role reports to	owner
Role Name as displayed on reports	

Save **Save & New** **Cancel**

User:

A user in Salesforce is an individual who logs into the system to access company records and features. Users can include employees like sales representatives, managers, and IT specialists. Each user has a unique account that identifies them and defines their access permissions, determining what data and functionalities they can interact with based on their assigned profile and role.

Create Users:

Activity 1: Creating the "Owner" User

1. Go to **Setup**.
2. Type **Users** in the Quick Find box and select **Users**.
3. Click on **New User**.
4. Fill in the fields as follows:
 - **First Name:** Vicky
 - **Last Name:** Y
 - **Alias:** Enter an alias name.
 - **Email ID:** Enter your personal email ID.
 - **Username:** Enter in the format text@text.text.
 - **Nickname:** Enter a nickname.
 - **Role:** Select **Owner**.
 - **User License:** Salesforce.
 - **Profile:** Owner.
5. Click **Save**.

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

Field	Value
First Name	Vicky
Last Name	Y
Alias	vy
Email	21501a0667@ovpsit.ac.in
Username	21501a0667@ovpsit.ac.in
Nickname	User173191779234690861
Title	
Company	
Department	
Division	

The 'Role' dropdown is set to 'owner'. The 'User License' dropdown is set to 'Salesforce'. The 'Profile' dropdown is set to 'Owner'. The 'Active' checkbox is checked. The 'Marketing User' checkbox is unchecked. The 'Offline User' checkbox is unchecked. The 'Knowledge User' checkbox is unchecked. The 'Flow User' checkbox is unchecked. The 'Service Cloud User' checkbox is unchecked. The 'Site.com Contributor User' checkbox is unchecked. The 'Site.com Publisher User' checkbox is unchecked. The 'WDC User' checkbox is unchecked. The 'Data.com User Type' dropdown is set to '-None-'.

Activity 2: Creating the "Agent" User

1. Go to **Setup**.
2. Type **Users** in the Quick Find box and select **Users**.
3. Click on **New User**.
4. Fill in the fields as follows:
 - **First Name:** Ram
 - **Last Name:** Ram
 - **Alias:** Enter an alias name.
 - **Email ID:** Enter your personal email ID.
 - **Username:** Enter in the format text@text.text.
 - **Nickname:** Enter a nickname.
 - **Role:** Select **Agent**.
 - **User License:** Salesforce Platform.
 - **Profile:** Standard Platform User.
5. Click **Save**.

The screenshot shows the 'User Edit' page in the Salesforce setup interface. The top navigation bar includes 'SETUP' and 'Users'. Below it, the main title is 'User Edit' with buttons for 'Save', 'Save & New', and 'Cancel'. A note '1 Required Information' is present. The 'General Information' section contains fields for First Name (ram), Last Name (ram), Alias (rram), Email (21501a0567@pvpit.ac.in), Username (21501a0567@pvpit.ac.in), Nickname (User173191797396622279), Title, Company, Department, and Division. To the right, the 'Role' is set to 'Agent', 'User License' to 'Salesforce Platform', and 'Profile' to 'Standard Platform User'. The 'Active' checkbox is checked. A large grid of checkboxes for various user types like Marketing User, Offline User, etc., has most boxes unchecked except for 'Active'. At the bottom, there are dropdowns for 'Data.com User Type' (set to 'None'), 'Data.com Monthly Addition Limit' (set to 'Default Limit (300)'), and checkboxes for 'Accessibility Mode (Classic Only)' and 'High-Contrast Pallete on Charts'.

Flows:

Salesforce Flows are a powerful automation tool that simplifies and streamlines business processes, enabling organizations to automate tasks, collect and manage data, and guide users through interactive steps without requiring coding knowledge. Flows are built using a visual drag-and-drop interface called Flow Builder, which allows you to design custom workflows tailored to your needs.

Types of Flows in Salesforce

- **Screen Flows:** Designed for user interaction, they guide users through a series of screens to collect or display data, often used for data entry or updates.
- **Autolaunched Flows:** Triggered automatically by events like record creation or updates, they handle background processes without user involvement.
- **Scheduled Flows:** Used to automate recurring tasks, these flows run at specific times or intervals.
- **Record-Triggered Flows:** Activate when records meet defined criteria, automating updates or related actions.
- **Subflows:** Reusable components within flows that simplify the management of complex processes.

Key Components of Flows

- **Flow Elements:** Include variables, decisions, loops, and actions to create robust workflows.
- **Flow Templates:** Pre-built templates offered by Salesforce to accelerate flow creation for common use cases.

Use Case for Creating Flows

Flows can automate processes such as calculating amounts based on selected laptop types. For instance, if a user selects a specific laptop type in a record, a flow can be configured to automatically populate the associated amount in the Amount field. This eliminates manual data entry, improves efficiency, and reduces errors.

Salesforce Flows empower businesses to enhance productivity, ensure data accuracy, and streamline operations through powerful automation capabilities.

Create A Flow On Dell Laptop:

1. Setup the Flow:

- Go to **Setup** → Type **Flow** in Quick Find → Click **Flow** → Select **New Flow**.
- Choose **Record-Triggered Flow** → Click **Create**.
- Select **Object**: Laptop Booking.
- Trigger Flow when: A record is Created or Updated.
- Optimize for: Actions and Related Records → Click **Done**.

2. Add Decision Element for Laptop Type:

- Click **+** → Select **Decision**.
- **Label**: Field should be Update.
- **Outcome Details**: Add outcomes for dell, acer, hp, mac.
- Resource: Record.Laptop booking_c → Operator: Equals → Value: dell.

3. Add Sub-Decisions for Core Types:

- For dell, click **+** → Select **Decision**.
- Add outcomes: dell core i3, dell core i5, dell core i7.
- Resource: Record.core type → Operator: Equals → Values: core i3, core i5, core i7.

4. Add Decisions for Months:

- For each core type (i3, i5, i7), click **+** → Select **Decision**.
- Add outcomes: dell 1, dell 2, dell 3, dell 4, dell 5 based on months.
- Resource: Record.how many months → Operator: Equals → Value: (1-5).

5. Update Records for Amounts:

- For each month, click **+** → Select **Update Records**.
- Field: Amount_c → Values:
 - For dell i3: 1-1000, 2-2000, 3-3000, 4-4000, 5-5000.
 - For dell i7: 1-2000, 2-4000, 3-6000, 4-8000, 5-10000.

6. Finalize and Save:

- Repeat for all branches → Test the flow.
- Click **Save** → Activate the flow.

Create A Flow On Acer Laptop:

1. Setup Decision for Core Type:

- Go to **Flow Page**.
- Beside **Acer**, click the **+** symbol.
- Select **Decision**.
- **Label**: Field is Update.
- **Outcome Label**: acer core i3, **API Name**: Automatically generated.
- **Resource**: Record.core type → **Operator**: Equals → **Value**: core i3.
- Click **Done**.

2. Add Decision for Months:

- After **Acer core i3**, click **+** symbol → Select **Decision**.
- **Label**: months selected.
- **Outcome Labels**:
 - acer 1(i3), acer 2(i3), acer 3(i3), acer 4(i3), acer 5(i3)
 - **Resource**: Record.how many months → **Operator**: Equals → **Value**: 1, 2, 3, 4, 5.
- Click **Done**.

3. Update Records for Amounts:

- For each outcome (acer 1(i3) to acer 5(i3)), click **+** → Select **Update Records**.
- **Field**: Amount_c → **Values**:
 - acer 1(i3) = 900,
 - acer 2(i3) = 1800,
 - acer 3(i3) = 2700,
 - acer 4(i3) = 3600,
 - acer 5(i3) = 4800.
- Click **Done**.

4. Save and Activate Flow:

- Click **Save** → **Activate** the flow.

Creating A Flow On HP Laptop:

1. Setup Decision for Core Type:

- Go to **Flow Page**.
- Beside **HP**, click the **+** symbol.
- Select **Decision**.
- **Label**: Field is Update.
- **Outcome Label**: hp core i5, **API Name**: Automatically generated.
- **Resource**: Record.core type → **Operator**: Equals → **Value**: hp i5.
- Click **Done**.

2. Add Decision for Months:

- After **HP core i5**, click + symbol → Select **Decision**.
- **Label**: hp field should be updated.
- **Outcome Labels**:
 - hp 1(i5), hp 2(i5), hp 3(i5), hp 4(i5), hp 5(i5)
 - **Resource**: Record.how many months → **Operator**: Equals → **Value**: 1, 2, 3, 4, 5.
- Click **Done**.

3. Update Records for Amounts:

- For each outcome (hp 1(i5) to hp 5(i5)), click + → Select **Update Records**.
- **Field**: Amount__c → **Values**:
 - hp 1(i5) = 1700,
 - hp 2(i5) = 3400,
 - hp 3(i5) = 5100,
 - hp 4(i5) = 6800,
 - hp 5(i5) = 8500.
- Click **Done**.

4. Save and Activate Flow:

- Click **Save** → **Activate** the flow.

Create A Flow On Mac Laptop:

1. Setup Decision for Core Type:

- Go to **Flow Page**.
- Beside **Mac**, click the + symbol.
- Select **Decision**.
- **Label**: mac should be Updated.
- **Outcome Label**: mac laptop, **API Name**: Automatically generated.
- **Resource**: Record.core type → **Operator**: Equals → **Value**: Bionic Chip.
- Click **Done**.

2. Add Decision for Months:

- After **mac laptop**, click + symbol → Select **Decision**.
- **Label**: Mac months selected.
- **Outcome Labels**:
 - mac bionic chip(1), mac bionic chip(2), mac bionic chip(3), mac bionic chip(4), mac bionic chip(5)
 - **Resource**: Record.how many months → **Operator**: Equals → **Value**: 1, 2, 3, 4, 5.
- Click **Done**.

3. Update Records for Amounts:

- For each outcome (mac bionic chip(1) to mac bionic chip(5)), click **+** → Select **Update Records**.
- **Field:** Amount_c → **Values:**
 - mac bionic chip(1) = 1700,
 - mac bionic chip(2) = 3400,
 - mac bionic chip(3) = 5100,
 - mac bionic chip(4) = 6800,
 - mac bionic chip(5) = 8500.

■ Click **Done**.

4. Save and Activate Flow:

- Click **Save** → **Activate** the flow.

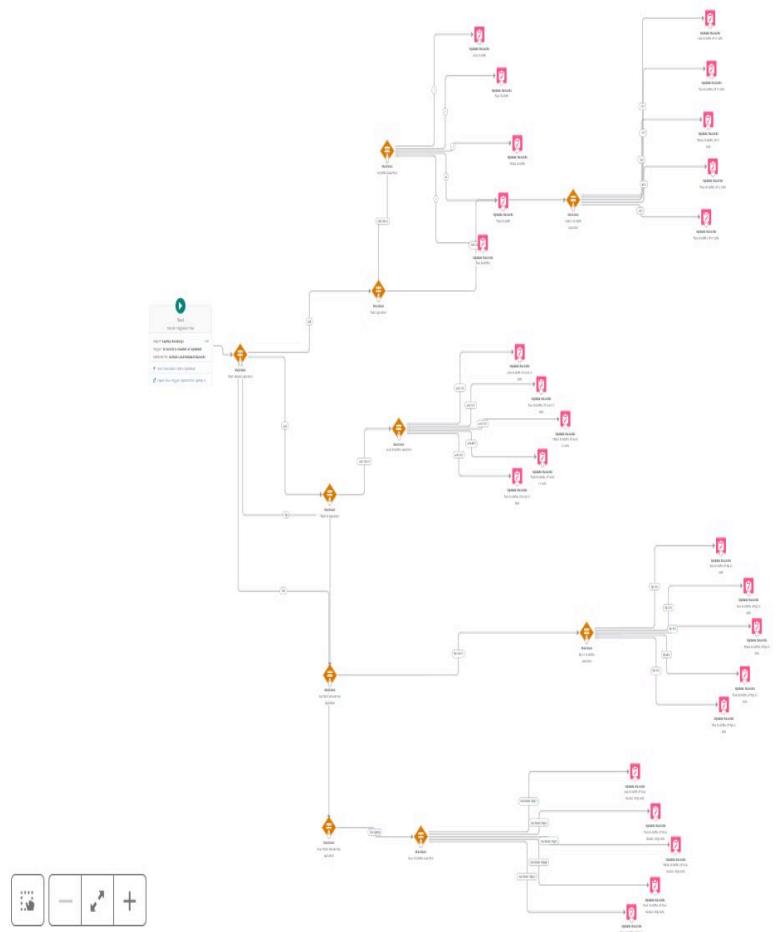
Saving the Flow:

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 Salesforce Lightning Platform server in conjunction with calls to the Salesforce Platform API. Apex is designed to look and behave like Java, offering developers the capability to add custom business logic to system events such as button clicks, record updates, Visualforce pages, and more.

Apex can be invoked by Web service requests, or it can be triggered by Salesforce platform events such as updates to Salesforce objects (via triggers). It supports object-oriented principles such as classes, methods, and inheritance, allowing for reusable code structures and modular programming. Apex is primarily used for server-side programming in Salesforce, allowing businesses to automate workflows, validate data, and integrate with other systems.

Creating Classes in Apex

In Apex, classes are used as templates to create objects, similar to how classes work in Java. A class encapsulates both data (attributes or variables) and behavior (methods or functions) that can operate on the data.

- **Class:** A class defines the structure and behavior of objects. It serves as a blueprint for creating instances (objects) in Salesforce. You can define variables, methods, and constructors within a class.
- **Object:** An object is an instance of a class. Once a class is defined, you can create multiple objects from that class. Objects store data and can invoke methods defined in the class.

Steps to Create a Class in Apex

1. **Log in to Salesforce:** Sign in to your Salesforce account and navigate to the Developer Console. Click on the gear icon in the top right corner and select the Developer Console option.
2. **Create New Class:** In the Developer Console window, navigate to the **File** menu, select **New**, and then choose **Apex Class**. This will open a new window where you can define your class.
3. **Define the Class:** Enter a name for the class in the pop-up window, ensuring it follows naming conventions (e.g., MyClass).
4. **Write Class Code:** After creating the class, write the required logic, variables,

and methods inside the class definition.

Access Specifiers in Apex

Access specifiers define the level of access that is allowed for the members (variables and methods) of a class. Apex supports four primary access modifiers: **private**, **protected**, **public**, and **global**. Each has different visibility rules.

1. Private:

- **Definition:** The private modifier is the default access level in Apex. It restricts the visibility of methods and variables to the class in which they are defined.
- **Usage:** If no access modifier is provided, Apex treats the method or variable as private.

2. Protected:

- **Definition:** The protected access modifier allows a method or variable to be accessed by the defining class and any class that inherits from it (i.e., subclasses).
- **Usage:** It's typically used when creating a class intended to be extended, allowing subclasses to have access to certain methods or variables.

3. Public:

- **Definition:** The public modifier allows methods and variables to be accessed by any Apex code within the same namespace. This makes the method or variable visible to all classes in the same Salesforce organization or package.
- **Usage:** It is used for methods or variables that need to be accessed outside of the class but within the same package.

4. Global:

- **Definition:** The global modifier provides the broadest access. A method or variable declared as global can be accessed by any other Apex code, even in external packages or in API calls.
- **Usage:** It is typically used for methods or variables that need to be referenced by external Salesforce organizations or applications, such as in SOAP or REST API calls.

Triggers in Apex

A **trigger** is a set of Apex code that is executed automatically in response to specific events related to Salesforce data, such as insertions, updates, and deletions (DML events). Triggers help automate business processes, including data validation, sending notifications, and more.

Triggers can be defined on Salesforce objects, and they are executed when certain events occur. Triggers can either run **before** or **after** the DML operation.

Types of Triggers in Salesforce

1. Before Triggers:

- **Definition:** A "before" trigger runs before any DML operation occurs. It is often used to validate data or modify record values before they are saved to the database.
- **Use Case:** Data validation, setting default values, preventing invalid data from being saved.
- **Example:** A trigger that checks if a required field is populated before an insert operation.

2. After Triggers:

- **Definition:** An "after" trigger runs after the DML operation has been completed. It is typically used when you need to work with data that has already been committed to the database, such as sending email notifications or updating related records.
- **Use Case:** Sending email alerts, updating related records, logging actions.
- **Example:** A trigger that sends an email after a new Account record is inserted.

How to Create a New Trigger in Apex

To create a new trigger:

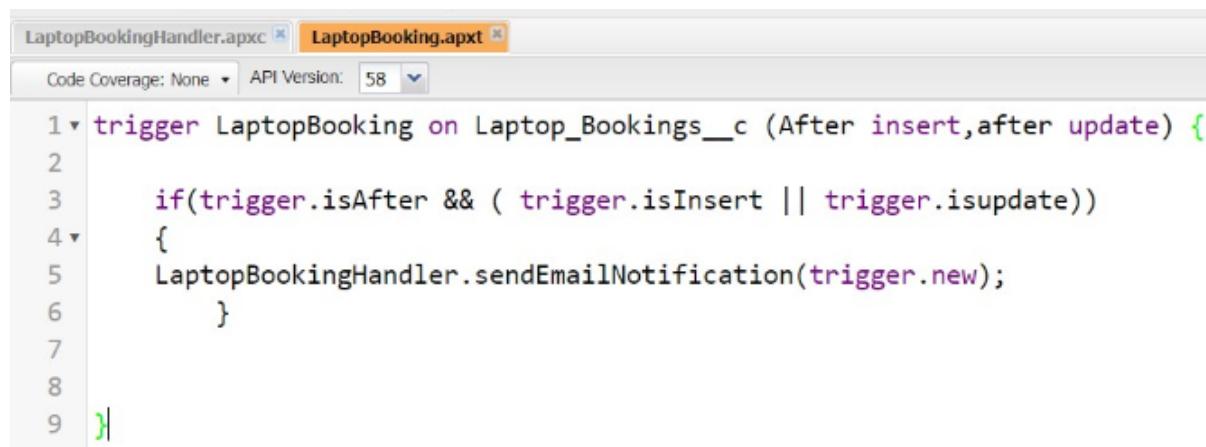
1. **Log in to Developer Console:** As with creating a class, log in to Salesforce and open the Developer Console.
2. **Create Trigger:** From the **File** menu, select **New**, then choose **Trigger**.
3. **Define Trigger:** Provide a trigger name and choose the object that will activate the trigger (e.g., Account or Contact).
4. **Write Trigger Logic:** Write the logic inside the trigger to specify what happens during the before or after DML event.

Syntax for Creating a Trigger

The general syntax for a trigger is as follows:

```
Trigger TriggerName on ObjectName (Before/After insert, update,  
delete) {  
    // Trigger logic goes here  
}
```

For example, a trigger to send email notifications after a record is inserted or updated:



A screenshot of a code editor window titled "LaptopBookingHandler.apxc" with a tab "LaptopBooking.apxt" highlighted. The code coverage is set to "None" and the API version is "58". The code itself is a trigger definition:

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

Code for Trigger Handler Class

A **handler class** is used to encapsulate the business logic that is executed in response to a trigger. This promotes modularity and reusability. The trigger itself should be kept as lightweight as possible, delegating the complex logic to the handler class.

Example of a handler class (LaptopBookingHandler):

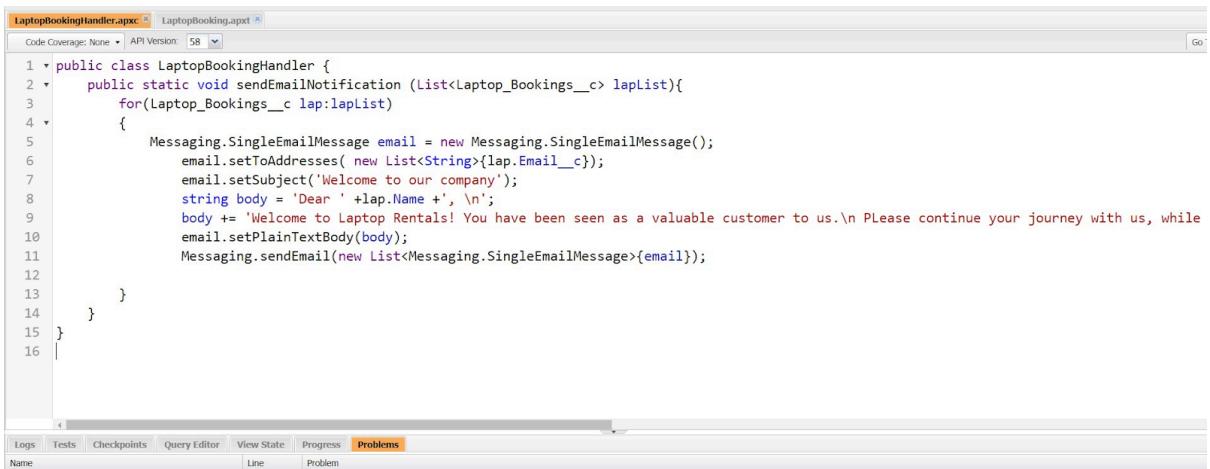
```
public class LaptopBookingHandler {  
    public static void  
sendEmailNotification(List<Laptop_Bookings__c> lapList) {  
        for (Laptop_Bookings__c lap : lapList) {  
            Messaging.SingleEmailMessage email = new  
            Messaging.SingleEmailMessage();  
            email.setToAddresses(new 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 += '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});
    }
}
}

```

In this example, the handler class is responsible for sending an email notification to the customer after their laptop booking record is inserted or updated. The class takes a list of `Laptop_Bookings__c` objects and processes each one, sending an email to the associated email address.



```

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

```

Summary

- Apex Overview:** Apex allows developers to add business logic and automation to Salesforce, similar to Java, with support for object-oriented programming and server-side execution.
- Classes & Objects:** Classes define the structure and behavior of objects. An object is an instance of a class, and methods can be used to manipulate the data of these objects.
- Access Specifiers:** Apex supports four access modifiers (private, protected, public, global) to control visibility and access to methods and variables within classes.
- Triggers:** Triggers in Apex run before or after DML operations, enabling developers to automate business processes.
- Handler Class:** A handler class separates the business logic from the trigger, promoting modularity and better organization of code.

Reports:

Salesforce provides robust tools for reporting and analyzing your data through various types of reports. These reports allow you to examine your data in various ways, with different formats and flexibility in display. You can also schedule and automate reports for regular use.

Types of Reports in Salesforce

1. Tabular Reports:

- **Description:** Simple listing of data without any subtotals. Provides the most basic view of your data, often with just a list or a grand total.
- **Use Cases:** Ideal for when you need to view data in a straightforward, non-hierarchical manner.
- **Example:** A list of all accounts, contacts, or opportunities without any grouping.

2. Summary Reports:

- **Description:** Lists data with groupings and subtotals. Useful for organizing data in a hierarchical manner.
- **Use Cases:** Best for when you want to group and summarize data based on a particular field.
- **Example:** All opportunities for a team, subtotalized by Sales Stage and Owner.

3. Matrix Reports:

- **Description:** Allows grouping records by both rows and columns. Provides a two-dimensional analysis, with totals by both row and column.
- **Use Cases:** Useful for comparing totals across two dimensions that are not necessarily related.
- **Example:** Summarize opportunities by month (vertical) and by account (horizontal).

4. Joined Reports:

- **Description:** Combines blocks of related data into one report. Each block can display different types of data with unique columns, summaries, filters, and sort orders.
- **Use Cases:** Ideal for grouping different report types or displaying various related data in a single view.
- **Example:** A report showing opportunities, cases, and activities for each account in separate blocks.

Create Report:

- **Navigate to the Reports Tab:**
 - Open your Salesforce application.
 - Click on the **Reports** tab in the navigation bar.
- **Create a New Report:**
 - Click on the **New Report** button.
- **Select a Report Type:**
 - From the available categories or the search panel, select the report type.
 - For example: "**Consumers with Laptop Bookings and Total Laptops**".
 - Click **Start Report** to proceed.
- **Customize the Report:**
 - Use the **Fields Pane** on the left to drag and drop fields into the report canvas.
 - Add relevant fields such as consumer name, booking date, and total laptops.
- **Group Rows and Columns:**
 - Group rows by dragging a field (e.g., Consumer Name) into the **Group Rows** section.
 - Similarly, group columns by dragging a field (e.g., Laptop Type) into the **Group Columns** section.
- **Add a Bucket Field (if needed):**
 - Click on the column drop-down menu.
 - Select **Bucket This Field**.
 - Define bucket criteria (e.g., grouping total laptops into ranges like 1-5, 6-10, etc.).
 - Click **Apply** to save the bucket configuration.
- **Save or Run the Report:**
 - Click **Save & Run** to save the report.
 - Provide a name and description for the report and select a folder to save it.
 - Click **Run** to generate the final report view.

The screenshot shows a Salesforce report interface. At the top, there's a navigation bar with tabs like 'LAPTOP RENTALS', 'Total Laptops', 'consumer', 'Laptop Bookings', 'Billing Process', 'Dashboards', and others. Below the navigation is a search bar and a toolbar with various icons. The main area displays a report titled 'Report: consumer with Laptop Bookings and Total Laptops consumer with laptops and total laptops'. The report includes a summary table with columns: 'types of versions', 'consumer: consumer_name', 'Laptop Bookings: Laptop Bookings', 'Total No Of Laptops: Total Laptops', 'Amount', and 'core type'. The summary table shows data for three rows: 'intermediate (1)' with consumer 'kaushik' and amount ₹1,000; 'very high (2)' with consumers 'shanks' and 'pavan' and amounts ₹50,000 and ₹60,000 respectively; and a subtotal row for ₹1,00,000. A final 'Total (3)' row shows a total of ₹1,11,000. At the bottom of the report area, there are checkboxes for 'Row Counts', 'Detail Rows', 'Subtotals', and 'Grand Total'.

Total Records	Total Amount
3	₹1,11,000
<input type="checkbox"/> types of versions ↑ ↓	
consumer: consumer_name	Laptop Bookings: Laptop Bookings
Total No Of Laptops: Total Laptops	Amount
core type	
Subtotal	
intermediate (1)	kaushik
	Update Status to Approved
	1
	₹1,000
Subtotal	₹1,00,000
very high (2)	shanks
	Update Status to Approved
	2
	₹50,000
	Core i5
	pavan
	Update Status to Approved
	3
	₹60,000
	Core i7
Total (3)	₹1,11,000

Sharing Report To Owner:

- **Open the Report:**
 - Navigate to the **Reports** tab in Salesforce.
 - Select the desired report you want to share.
- **Edit the Report:**
 - Click on the **Edit** drop-down menu next to the report name.
- **Subscribe to the Report:**
 - From the options, select **Subscribe**.
- **Set the Subscription Options:**
 - In the subscription panel, configure the following:
 - **Run Report As:** Select **Another Person**.
 - Choose your personal account or the account of the owner who should receive the email notification.
- **Save the Subscription:**
 - Confirm the frequency and time of the email notification (e.g., daily).
 - Click **Save** to finalize the subscription.

The screenshot shows the 'Edit Subscription' dialog box. It has several sections: 'Settings' (Frequency: Daily, Weekly, Monthly), 'Time' (set to 8:00 am), 'Attachment' (button to 'Attach File'), 'Recipients' (Send email to Me, button to 'Edit Recipients'), and 'Run Report As' (radio buttons for 'Me' and 'Another Person'). At the bottom right are 'Cancel' and 'Save' buttons.

Dashboards:

Dashboards in Salesforce provide a visual representation of your data, allowing you to monitor key metrics and business trends in real time. By leveraging data from reports, dashboards enable users to analyze quantities, track performance, and measure the impact of their activities effectively. They are customizable and interactive, helping teams make informed decisions quickly and adapt to changing business conditions.

Create a Dashboard Folder:

- 1. Access App Launcher**
 - Click on the **App Launcher** icon in the top-left corner of the Salesforce interface.
 - Search for and select **Dashboards**.
- 2. Open Dashboards Tab**
 - Click on the **Dashboards** tab to access the dashboard section.
- 3. Create a New Folder**
 - In the dashboard section, click on the **New Folder** button.
- 4. Set Folder Label**
 - In the pop-up window, enter the folder label as "**Total Rent Amount**".
 - The **Folder Unique Name** field will automatically populate based on the label you provide.
- 5. Save the Folder**
 - Click the **Save** button to create the folder.

The screenshot shows a 'Create folder' dialog box. At the top, it says 'Create folder'. Below that, there is a field labeled '* Folder Label' containing the text 'total rent amount'. Underneath it is a field labeled '* Folder Unique Name' containing the text 'totalrentamount'. At the bottom right of the dialog box are two buttons: 'Cancel' and 'Save'.

Create a Dashboard:

- **Access Dashboards Tab**
 - Go to the app and click on the **Dashboards** tab.
- **Create a Dashboard**
 - Click on **New Dashboard**.
 - Enter a **Name** for the dashboard.
 - Select the folder you created earlier (e.g., "Total Rent Amount").
 - Click **Create**.
- **Add a Component**
 - On the dashboard page, click **Add Component**.
- **Select a Report**
 - Choose a report from the list (e.g., "Laptop Bookings and Total Laptops").
 - Click **Select**.
- **Customize Component**
 - Select a **dark-themed component** to enhance visibility.
 - Add it to the dashboard.
- **Save the Dashboard**
 - Click the **Save** button.
- **Finalize**
 - Click **Done** to complete the process.

The screenshot shows a software interface for managing dashboards and reports. At the top, there's a navigation bar with various icons and dropdown menus. Below it, the 'Dashboards' tab is selected. A search bar and a 'Recent' section are visible. The main area displays a table of dashboards, with one entry highlighted: 'data analytics of laptops'. This entry has a detailed description: 'total amount of data in dashboards', 'total rent amount', 'Kaushik Koganti', and a date '18/11/2024, 4:26 pm'. Below this, another dashboard titled 'data analytics of laptops' is shown, featuring a large blue donut chart with the value '₹111k' and a note 'types of versions Other very high'. A link 'View Report (consumer with laptops and total laptops)' is at the bottom of this card.

Conclusion:

The project "**A CRM Application for Laptop Rentals**" showcases the effective utilization of Salesforce to streamline operations, enhance customer experience, and provide actionable business insights. By leveraging Apex classes and triggers, critical tasks like automated email notifications are seamlessly executed, reducing manual effort and ensuring efficiency.

The use of Salesforce reports—Tabular, Summary, Matrix, and Joined Reports—enables the organization to analyze rental trends, revenue, and customer behavior effectively. Dashboards further enhance real-time visualization, presenting key performance metrics in an accessible format for stakeholders.

This application adopts a customer-centric approach by automating communication and tracking preferences, fostering loyalty and repeat business. Its scalability ensures that the system grows alongside business demands, while scheduled reports and subscription features simplify data sharing and ensure accessibility.

In conclusion, "**A CRM Application for Laptop Rentals**" successfully integrates automation, data organization, and visualization to optimize business processes, making it a robust, flexible, and future-ready solution for the laptop rental industry.