

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

This **CRM application** is designed to streamline and enhance the process of **renting laptops** for short-term use. The main objective is to utilize Salesforce's **Customer Relationship Management** (CRM) capabilities to foster stronger customer relationships, ultimately leading to an improved customer experience. By managing **customer data, rental transactions, and inventory on a unified platform**, this project optimizes store operations, reduces manual effort, and boosts overall efficiency. Additionally, the CRM enables seamless communication with customers through automated email notifications, keeping them updated on bookings, billing, and other important information. The outcome is a comprehensive solution that supports efficient rental management, enhances customer satisfaction, and promotes operational excellence for the business.

## Salesforce:

### What Is Salesforce?

Centralizes customer data for personalized service and relationship management.

- Automates repetitive tasks to boost productivity and efficiency.
- Provides real-time insights with customizable reports and dashboards.
- Integrates easily with other tools, enhancing data flow and reducing silos.
- Offers mobile accessibility, enabling flexibility and responsiveness on the go.

## Major Objectives

- **Streamline Sales Processes:** Use Salesforce solutions to automate and enhance sales workflows, minimizing manual tasks and boosting lead management efficiency.
- **Improve Customer Engagement:** Take advantage of Salesforce tools to deliver a tailored experience for customers, enhancing communication and overall satisfaction.
- **Insights from Data:** Utilize Salesforce analytics to collect actionable insights and effectively monitor performance metrics.

## Key Features for Salesforce and Concepts Used

This Salesforce CRM project utilizes the following concepts and incorporates a variety of

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

features:

- **Custom Objects:** Defined unique custom objects such as **Consumer**, **Laptop Bookings**, **Total Laptops**, and **Billing Process** to handle specific requirements for managing customer data, rental transactions, and inventory. These custom objects allowed for precise data handling and organization beyond the standard Salesforce objects.
- **Custom Tabs:** Created custom tabs for each custom object to provide easy access to **Consumer**, **Laptop Bookings**, **Billing Process**, and **Total Laptops** information. This allowed the team to navigate and manage data directly from the Salesforce app interface.
- **Creating Lightning Apps:** Built custom Lightning apps for the **Laptop Rentals CRM** to provide a tailored user interface, streamline workflows, and improve user experience by organizing all necessary tabs, reports, and dashboards under one cohesive application.
- **Validation Rules:** Implemented validation rules to maintain data integrity and enforce business requirements. For example, a customer must specify their contact information (either email or phone number).
- **Profile Cloning and Custom Permissions:** Cloned the standard **User** profile to create **Owner** and **Agent** profiles, customizing permissions to restrict or allow access to specific objects and data fields based on user roles. This ensured proper access control and safeguarded sensitive information.
- **Role Hierarchy and Role Creation:** Established a role hierarchy by creating an **Owner** role under the **CEO** and an **Agent** role under the **Owner**.
- **User Creation:** Created two user accounts: **Owner** and **Agent**. The **Owner** has elevated permissions for full control, while the **Agent** can handle customer interactions and manage bookings within the limits set by their profile.
- **Flows for Laptop Models:** Created a **Record-Triggered Flow** for laptop models to automate the model selection, processor type selection, number of months to rent the laptop, and the charges based on the chosen inputs.
- **Apex Programming:** Developed **Apex classes** and **triggers** to execute complex business logic and trigger email notifications after booking a laptop for rent.
- **Reports and Report Sharing:** Created a custom report to provide insights into rental activity, filtering data according to version types and sharing the report with the **Owner**.
- **Dashboard Folder and Dashboard Creation:** Created a **Dashboard** to represent the created report in the form of a donut chart, enabling analysis of the rental amounts for each version. This is the final step in Application creation.

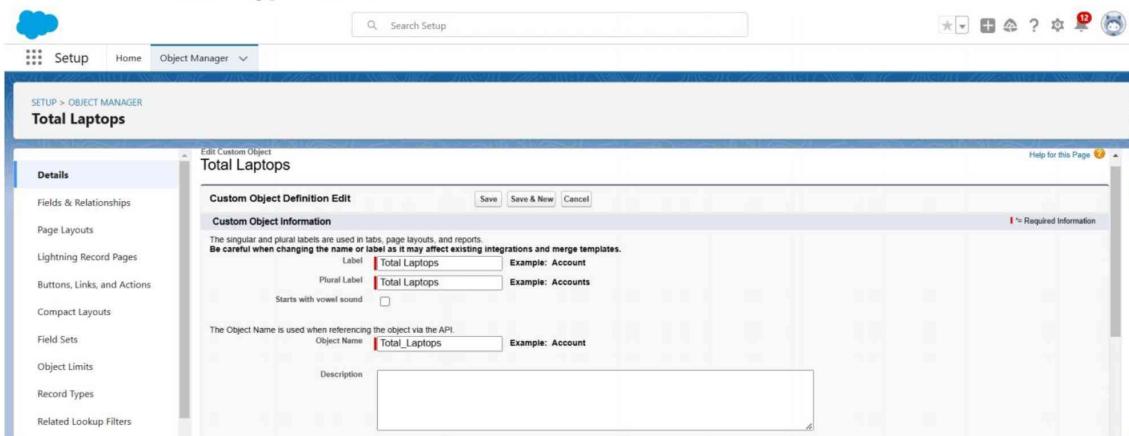
# Title: A CRM APPLICATION FOR LAPTOP RENTALS

## 1. Objects Creation

### Create Total Laptops Object

#### Fill in the Required Fields:

- Enter the following details:
  - **Label Name:** Total Laptop
  - **Plural Label Name:** Total Laptops
- Configure the **Record Name** field:
  - **Record Name Label:** Total Laptops
  - **Data Type:** Text



### Create consumer Object

#### Consumer Object: Simplifying Customer Interactions

The **Consumer** object enables the Laptop Rentals team to deliver exceptional service by keeping all customer information readily accessible. This allows team members to focus on solving customer needs quickly and effectively.

#### Why It Matters:

- **Phone\_number\_c:** Ensures quick customer contact for rental updates or queries.
- **Email\_c:** Keeps customers informed through timely reminders, confirmations, and follow-ups.
- **Address\_c:** Helps ensure prompt and accurate deliveries, avoiding delays.
- **Consumer\_Status\_c:** Helps identify customer segments (e.g., students, employees) to provide personalized service.

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text Area(255)		
consumer Name	Name	Text(80)		
consumer Status	consumer_Status__c	Picklist		
Created By	CreatedById	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone number	Phone_number_c	Phone		

## Create Laptop Bookings Object

### Laptop Bookings Object Design:

The **Laptop Bookings** custom object is engineered to integrate seamlessly with the CRM system, enabling data consistency and supporting scalability for future needs.

### Field Details:

- Name:** A mandatory text field used as the primary identifier for booking records.
- Laptop\_names\_\_c:** A picklist field to standardize the selection of available laptop models.
- Core\_type\_\_c:** Another picklist field to specify processor types, ensuring easy filtering and reporting.
- Amount\_\_c:** A currency field for storing the rental payment amount, supporting accurate financial tracking.
- Email\_\_c:** An email field to ensure consistent formatting and enable email-based notifications.
- Name\_\_c:** A lookup field to the **Consumer** object, linking bookings with customer records for relational data.
- Total\_No\_Of\_Laptops\_\_c:** A lookup field to the **Total Laptops** object to manage inventory counts and availability.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount_c	Currency(18, 0)		
consumer	consumer_c	Master-Detail(Consumer)		
Core Type	Core_Type__c	Picklist	Laptop Names	
Core Type	Core__c	Picklist		
Created By	CreatedById	Lookup(User)		
Email	Email_c	Email		
how many months	how_many_months__c	Picklist		
Laptop Bookings Name	Name	Text(80)		
Laptop Names	Laptop_Names_c	Picklist		
Laptops Available	Laptops_Available__c	Formula (Number)		
Last Modified By	LastModifiedById	Lookup(User)		
Total No Of Laptops	Total_No_Of_Laptops_c	Master-Detail(Total Laptops)		

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

## Create Billing Process Object

- **Amount\_c:** A formula field (Number) that automatically calculates the total amount for the billing process based on predefined criteria, ensuring accurate and consistent billing.
- **Name:** The standard text field used as the unique identifier for each billing record, which may be labeled as "Billing Process Name" to distinguish it from other records.
- **Laptop\_Booking\_c:** A lookup field that links each billing record to a specific laptop booking within the **Laptop\_Bookings\_c** object, providing context for the rental transaction.
- **Name\_c:** A master-detail relationship to the **Consumer** object, ensuring each billing record is directly associated with a specific consumer for clear customer tracking.
- **Payment\_Mode\_c:** A picklist field that specifies the payment method used for the transaction, offering options such as "Credit Card," "Bank Transfer," or "Cash."

Fields & Relationships				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount_c	Formula (Currency)		
Billing Process Name	Name	Text(80)		
Created By	CreatedById	Lookup(User)		
Laptop Booking	Laptop_Booking_c	Lookup(Laptop Booking)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name_c	Master-Detail(consumer)		
Payment Mode	Payment_Mode_c	Picklist		

## 2. Tabs

**What is Tab :** A tab is a user interface element used in Salesforce to organize and display records for various objects. It allows users to create, view, and manage records efficiently within the platform.

### Types of Tabs:

- **Custom Tabs:**
  - Custom object tabs serve as the user interface for custom applications you build in Salesforce. They resemble standard Salesforce tabs like Accounts, Contacts, and Opportunities but are tailored for custom objects.
- **Web Tabs:**
  - Web tabs are designed to display web content or applications directly within the

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

Salesforce interface. They allow users to access frequently used external content or applications without leaving Salesforce.

- **Visualforce Tabs:**

- Visualforce tabs display Visualforce pages within Salesforce. They behave like standard tabs and can be customized to present data or functionality unique to your business needs.

- **Lightning Component Tabs:**

- Lightning Component tabs enable the integration of Lightning components into the navigation menu in Lightning Experience and the Salesforce mobile app. These tabs offer a dynamic way to enhance user workflows.

- **Lightning Page Tabs:**

- Lightning Page tabs allow Lightning Pages to be included in the navigation menu for the Salesforce mobile app. However, unlike other custom tabs, Lightning Page tabs:

- Do not appear on the "All Tabs" page (accessible via the Plus icon).
- Are not listed under "Available Tabs" when customizing app tabs.

## Creating a Custom Tab

- **Total Laptops Tab:**

- Provides users with centralized access to the **Total Laptops** object.
- Facilitates efficient inventory management by consolidating data on laptops available for rental.

- **Consumer Tab:**

- Enables streamlined management of customer details.
- Supports effective customer relationship management for better service delivery.

- **Laptop Booking Tab:**

- Allows users to manage laptop rental transactions.
- Links bookings to specific customers and inventory for accurate tracking.

- **Billing Process Tab:**

- Handles billing and payment details seamlessly.
- Tracks financial transactions related to laptop rentals for transparency and accuracy.

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. The left sidebar has 'User Interface' expanded, with 'Tabs' selected. The main area is titled 'Custom Tabs' and contains sections for 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. Under 'Custom Object Tabs', there is a table listing four tabs: 'Billing Processes' (Diamond style), 'consumers' (Presenter style), 'Laptop Bookings' (Chalkboard style), and 'Total Laptop' (Laptop style). Each row includes 'Edit | Del' links.

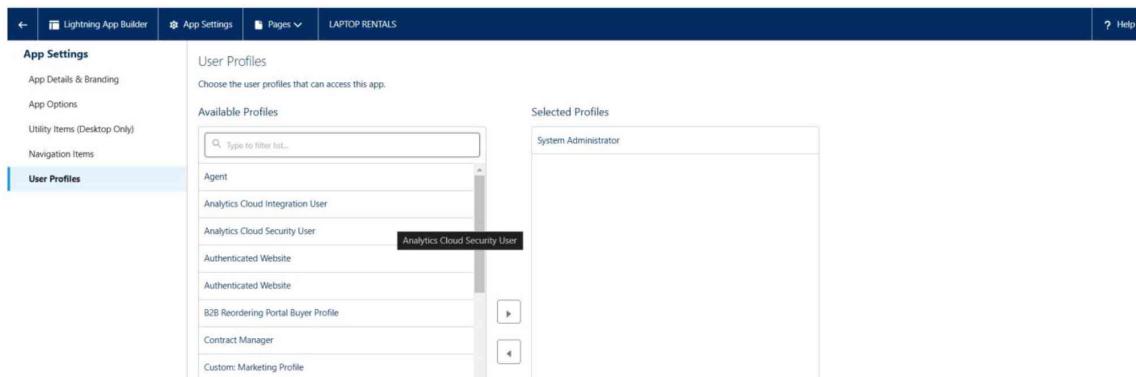
## 3. The Lightning App

A **Lightning App** is a collection of tools and features that work together to serve a specific purpose, providing users with access to objects, tabs, and other resources in a single, streamlined navigation bar in **Lightning Experience**. These apps allow for customization with custom colors, logos, utility bars, and Lightning page tabs, enabling organizations to brand their apps while enhancing functionality. By grouping related tools and resources, Lightning Apps improve efficiency, allowing users to switch between apps seamlessly and work more productively. They offer a centralized and flexible solution to optimize workflows and navigation within the organization.

The screenshot shows the Lightning App Builder interface with the 'App Details & Branding' tab selected. The left sidebar lists 'App Settings', 'App Details & Branding' (selected), 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles'. The main area has two sections: 'App Details' and 'App Branding'. In 'App Details', fields include 'App Name' (LAPTOP RENTALS), 'Developer Name' (LAPTOP\_RENTALS), and 'Description' (Enter a description...). In 'App Branding', fields include 'Image' (a thumbnail of a laptop), 'Primary Color Hex Value' (#0070D2), and 'Org Theme Options' (checkbox). Below these are 'App Launcher Preview' and a preview card showing the app icon and name.

To add user permissions:

# Title: A CRM APPLICATION FOR LAPTOP RENTALS



## Fields in Salesforce

In Salesforce, fields represent the data stored in the columns of a relational database. They hold valuable information specific to an object, making the searching, editing, and deletion of records simpler and more efficient.

### Types of Fields

#### 1. Standard Fields

Standard fields are predefined fields in Salesforce that perform essential functions.

These fields cannot be deleted unless they are non-required standard fields. Some standard fields are common across all Salesforce applications, including:

- **Created By:** Tracks the user who created the record.
- **Owner:** Identifies the user or group that owns the record.
- **Last Modified:** Indicates the last user to modify the record and the modification date.
- **Fields Created During Object Creation:** Fields automatically added when an object is created.

#### 2. Custom Fields

Custom fields are user-defined and highly flexible. They can be tailored to meet specific organizational needs. Unlike standard fields, custom fields are optional and can be added or removed as required. Users have complete control over custom fields, making them ideal for adapting Salesforce to unique business processes.

### Create Fields in Consumer Object

#### • Fields

- Phone number
- Email

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

- Address
- Consumer Status
- etc

The image contains two screenshots of the Salesforce Setup interface, specifically the Object Manager for the 'consumer' object.

**Screenshot 1: Edit consumer Custom Field - Phone number**

This screenshot shows the 'Custom Field Definition Edit' screen for a field named 'Phone number'. The field label is 'Phone number', the field name is 'Phone\_number', and the data type is 'Phone'. Other settings include 'Data Owner' set to 'User', 'Field Usage' and 'Data Sensitivity Level' both set to 'None', and 'Compliance Categorization' showing 'Available' categories: PII, HIPAA, GDPR, and PCI. The 'Chosen' category is empty.

**Screenshot 2: New Custom Field - Step 2. Enter the details**

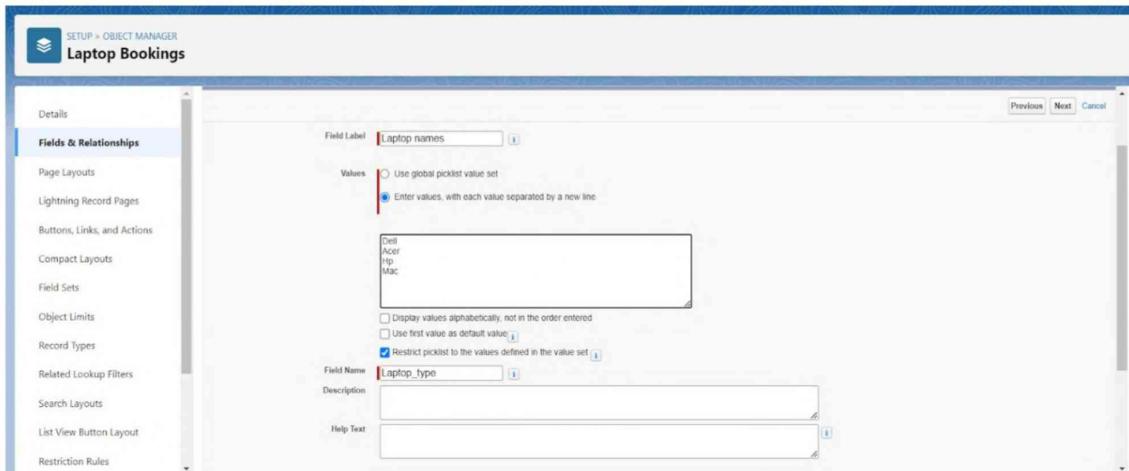
This screenshot shows the 'Step 2. Enter the details' screen for creating a new custom field named 'Email'. The field label is 'Email', and the field name is 'Email'. Other settings include 'Required' checked, 'Unique' unchecked, 'External ID' unchecked, and 'Auto add to custom report type' checked. The 'Chosen' category is empty.

## Creating the Field in Laptops Booking Object

- **Create Laptop Names:**
- **Field Label:** Laptop Names.
- **Picklist Values:**
  - Dell
  - Acer
  - HP

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

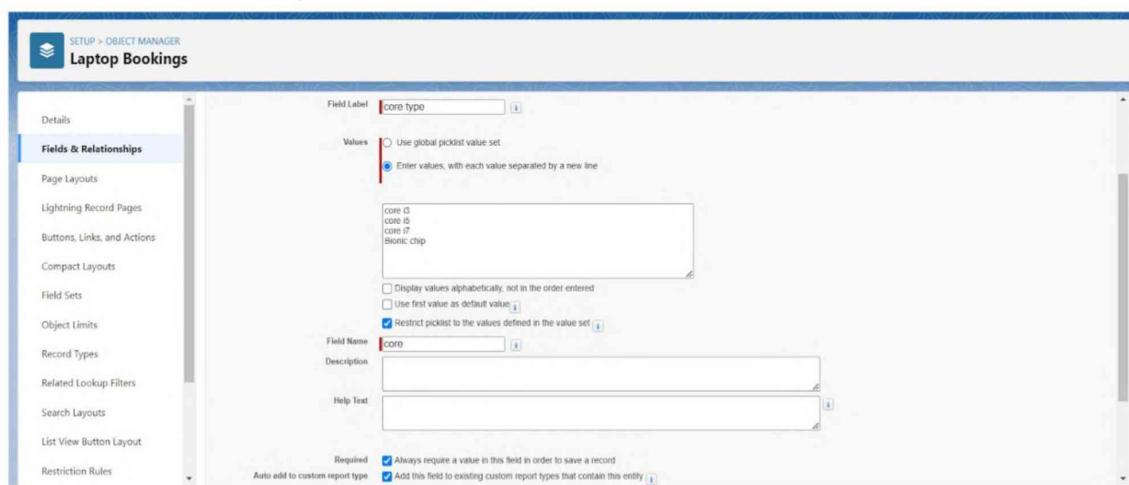
- Mac



## 2. Create Core Type:

This Core Type field is used for version splitting

- **Field Label:** Core Type.
- **Picklist Values:**
  - Core i3
  - Core i5
  - Core i7
  - Bionic Chip



Note: By using field dependencies, you can streamline data entry and ensure that users are only presented with relevant options based on their previous selections.

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

## Creating a Field Dependency in the Laptop Booking Object

To create a field dependency for an object:

- **Navigate to the Object:**
  - Setup > Object Manager >> object name (**Laptop Booking**)
- **Access Field Dependency:**
  - Field Dependencies >> Next.
- **Select Controlling and Dependent Fields:**
  - Controlling Field as Laptop Names >> Dependent Field as Core Type.
- **Define the Dependency Values:**
  - **Dell**,values: Core i3, Core i5, Core i7.
  - **Acer**,values: Core i3, Core i5, Core i7.
  - **HP**,values: Core i3, Core i5, Core i7.
  - **Mac**,value: Bionic Chip.
- 5. **Save it.**

The screenshot shows the Salesforce Object Manager interface for the 'Laptop Bookings' object. The 'Fields & Relationships' tab is selected. On the right, the 'Field Dependencies' section is open, showing the configuration for a new dependency. The 'Controlling Field' is set to 'Laptop names' and the 'Dependent Field' is set to 'core type'. Below this, there's an 'Instructions' section with a legend: 'Excluded Value' (grey) and 'Included Value' (yellow). A table displays the dependency values, where 'Dell' includes 'core i3', 'core i5', and 'core i7'; 'Acer' includes 'core i3', 'core i5', and 'core i7'; 'HP' includes 'core i3', 'core i5', and 'core i7'; and 'Mac' includes 'Bionic chip'. Buttons for 'Include Values' and 'Exclude Values' are available for each row.

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

- After creating the Master-Detail Relationship, you can create a Roll-Up Summary field.
- Fill in the details as follows:
  - i. **Field Label:** Laptops Delivered
  - ii. **Field Name:** Auto-generated based on the field label

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

The screenshot shows the 'Total laptops' object in the Salesforce Object Manager. A new custom field is being created with the following details:

- Field Label:** Laptops delivered
- Field Name:** laptops\_delivered
- Description:** (empty)
- Help Text:** (empty)

Step 2 of 5 is completed. Navigation buttons: Previous, Next, Cancel.

- **Next.**
- In the **Summarized Object**, select **Laptop Bookings**.
- Select the **Count** radio button in the **Roll-Up Type** section.

The screenshot shows the continuation of creating the 'Laptops delivered' custom field. Step 3 of 5 is shown:

- Select Object to Summarize:** Master Object: Total laptops; Summarized Object: Laptop Bookings (selected from a dropdown menu).
- Select Roll-Up Type:** COUNT (radio button selected).
- Field to Aggregate:** None (dropdown menu).
- Filter Criteria:** All records should be included in the calculation (radio button selected).

Step 3 of 5 is completed. Navigation buttons: Previous, Next, Cancel.

## To create fields Laptops Available:

- Fill in the details as follows:
  - **Field Label:** Laptops Available
- **Formula Return Type as Number >> Set Decimal Places to 0**
- **Advanced Formula** section

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

- **Insert Field**, and a popup window will appear. Under **Laptop Booking**, select **Total No Of Laptops** in the second column and **Laptops Delivered** in the third column.
- **Insert** to add:  
50 - Total\_no\_of\_laptops\_\_r.Laptops\_delivered\_\_c



2. Then What are the steps done in Consumer, Laptop Booking object we can do the same as to the remaining objects as well(Billing process, Total laptops).

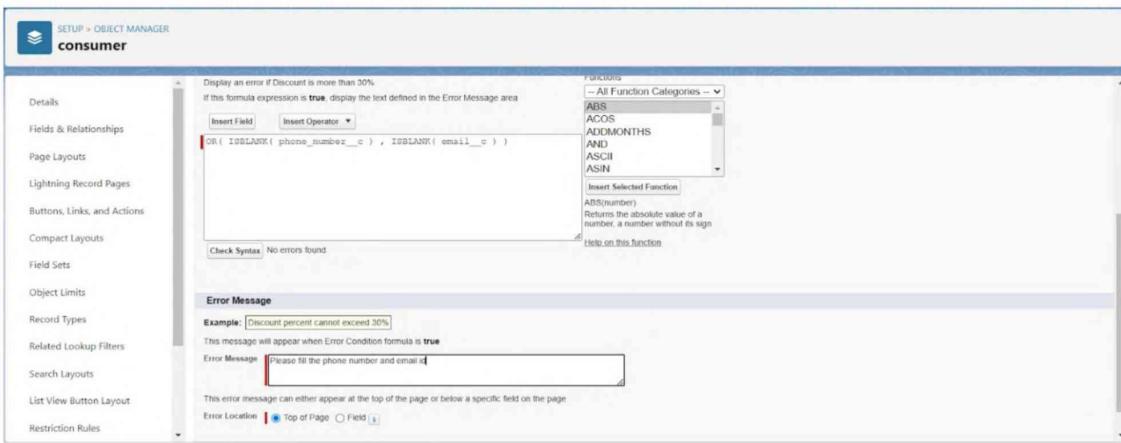
## 4. Creating the validation rule for phone number field in consumer object

To create a validation rule for the phone number field in the Consumer object:

A validation rule, **PhoneNumberOrEmailBlankRule**, was implemented in the **Consumer** object to ensure data completeness and reliability. This rule enforces that at least one of the fields, **Phone Number** or **Email**, must be populated in each consumer record. By preventing both fields from being left blank, this rule ensures accurate and essential contact information is always captured for every consumer.

1. **Rule Name:** Phonenumbereoremailblankrule.
2. **Description:** Phone number and email number should not be blank.
3. **Formula:** OR( ISBLANK( phone\_number\_\_c ), ISBLANK( email\_\_c ) )

# Title: A CRM APPLICATION FOR LAPTOP RENTALS



## 5. Profiles

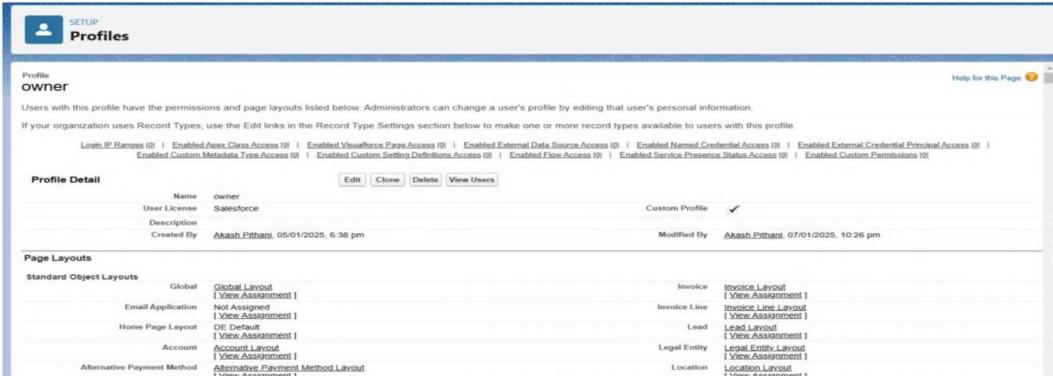
A **profile** in Salesforce is a collection of settings and permissions that define what a user can do. It controls object permissions, field permissions, user permissions, tab settings, app settings, Apex class access, Visualforce page access, page layouts, record types, and login hours/IP ranges. Profiles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

### owner Profile:

By cloning the **Standard User** Profile we can create this **Owner Profile**. And also giving the Custom Odject Permission this profile also.

#### To create a new profile:

- Profile Name: Owner



# Title: A CRM APPLICATION FOR LAPTOP RENTALS

The screenshot shows the 'Profiles' setup page. It displays 'Custom Object Permissions' for several objects. A blue arrow points to the 'Basic Access' row for the 'Billing Process' object, which includes columns for Read, Create, Edit, Delete, View All, and Modify All. Other objects shown include Individuals, Invoices, Leads, Work Types, Work Type Groups, Laptop Bookings, and Total Laptops. Below the permissions section are 'Session Settings' and 'Password Policies'.

- Give the appropriate access

## Agent Profile:

By cloning the **Standard Platform User** Profile we can create this **Agent Profile**. And also giving the Custom Odject Permission this profile also.

The screenshot shows the 'Profile Agent' setup page. It displays the 'Profile Detail' section for the 'Agent' profile. The profile has a name of 'Agent', a user license of 'Salesforce Platform', and is marked as a 'Custom Profile'. It was created by 'Akash Pithani' on '05/01/2025, 6:43 pm' and modified by 'Akash Pithani' on '07/01/2025, 10:26 pm'. There are buttons for 'Edit', 'Clone', 'Delete', and 'View Users'.

## 6. Roles and Hierarchy

A **role** in Salesforce defines a user's visibility and access at the record level. Roles are used to specify the types of access users can have to data within the Salesforce organization. In simple terms, a role determines what records a user can view or access based on their position in the Salesforce hierarchy.

The created **Owner Role** and the **Agent Role** are included in the Role heirarchy. **OwnerRole** is placed under the **CEO** Role and **Agent Role** under the **Owner Role**.

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

The screenshot shows the Salesforce Setup Roles page. At the top, there's a blue header bar with a user icon and the text "SETUP Roles". Below the header, the title "Creating the Role Hierarchy" is displayed. A sub-instruction below it says, "You can build on the existing role hierarchy shown on this page. To insert a new role, click Add Role." A horizontal line separates this from the main content area. The main area is titled "Your Organization's Role Hierarchy" and contains a collapsible tree view of roles. The root node is "no", which has several children: "Add Role", "CEO", "CFO", "COO", "owner", "Agent", "SVP, Customer Service & Support", "Customer Support, International", "Customer Support, North America", "Installation & Repair Services", and "SVP, Human Resources". Each node includes "Edit | Del | Assign" links and an "Add Role" link under its name.

## 7. Users

A user is anyone who logs into Salesforce. Users are typically employees of your company—such as sales representatives, managers, or IT specialists—who require access to company records. Each user in Salesforce has a unique user account. This account identifies the user and includes settings that define the features and records they can access.

### Create User:

Create Two User called **Vicky** and **Ram Ramesh** along with related details.

Purpose of users:

1. Vicky - Owner profile

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

User Detail for vicky y

**User Detail**

Name	vicky y	Role	owner
Alias	yvick	User License Profile	Salesforce
Email	pithanikash788@gmail.com [Verify]	Profile	owner
Username	pithanikash788@247gmail.com	Active	<input checked="" type="checkbox"/>
Nickname	vicky	Marketing User	<input type="checkbox"/>
Title		Offline User	<input type="checkbox"/>
Company		Knowledge User	<input type="checkbox"/>

## 2. Ram Ramesh - Agent profile

User Detail for ram ramesh

**User Detail**

Name	ram ramesh	Role	Agent
Alias	rame	User License Profile	Salesforce Platform
Email	pithanikash@gmail.com [Verify]	Profile	Agent
Username	pithanikash@00gmail.com	Active	<input checked="" type="checkbox"/>
Nickname	rame	Marketing User	<input type="checkbox"/>
Title		Offline User	<input type="checkbox"/>
Company		Knowledge User	<input type="checkbox"/>

## 8. Flows

In Salesforce, a flow is a versatile tool that helps automate business processes, gather and update data, and guide users through a series of steps or screens. Flows are created using a visual interface, making it easy to build them without any coding knowledge.

Created a Record-triggered flow to automate the laptop booking process that is initiated by the consumer. This flow automates what to do after booking the laptop like triggering email notifications.

**Create a Flow on dell laptop, Acer, Hp, Mac:**

**Activity:**

- **Each and every laptop type are having three Core Types.**
- **Every Core Type also having Five different months with separated amounts.**

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

**Decision**

* Label Field should be Update	* API Name Field_should_be_Update
Description	
<b>Outcomes</b> For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.	
<b>OUTCOME ORDER</b> <span style="border: 1px solid #ccc; padding: 2px;">dell</span> <span style="border: 1px solid #ccc; padding: 2px;">acer</span> <span style="border: 1px solid #ccc; padding: 2px;">hp</span> <span style="border: 1px solid #ccc; padding: 2px;">mac</span> <span style="border: 1px solid #ccc; padding: 2px;">False</span>	<b>OUTCOME DETAILS</b> <p>* Label: dell</p> <p>* Outcome API Name: dell</p> <p>Condition Requirements to Execute Outcome: All Conditions Are Met (AND)</p> <p>Resource: ...Bookings_c &gt; Laptop Names Operator: Equals Value: Dell</p> <p>+ Add Condition</p> <p>When to Execute Outcome: If the condition requirements are met</p>

**Decision**

* Label field updated	* API Name field_updated
Description	
<b>Outcomes</b> For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.	
<b>OUTCOME ORDER</b> <span style="border: 1px solid #ccc; padding: 2px;">dell core i3</span> <span style="border: 1px solid #ccc; padding: 2px;">dell core i5</span> <span style="border: 1px solid #ccc; padding: 2px;">dell core i7</span> <span style="border: 1px solid #ccc; padding: 2px;">Default Outcome</span>	<b>OUTCOME DETAILS</b> <p>* Label: dell core i3</p> <p>* Outcome API Name: dell_core_i3</p> <p>Condition Requirements to Execute Outcome: All Conditions Are Met (AND)</p> <p>Resource: ...top_Bookings_c &gt; Core Type Operator: Equals Value: Core i3</p> <p>+ Add Condition</p> <p>When to Execute Outcome: If the condition requirements are met</p>

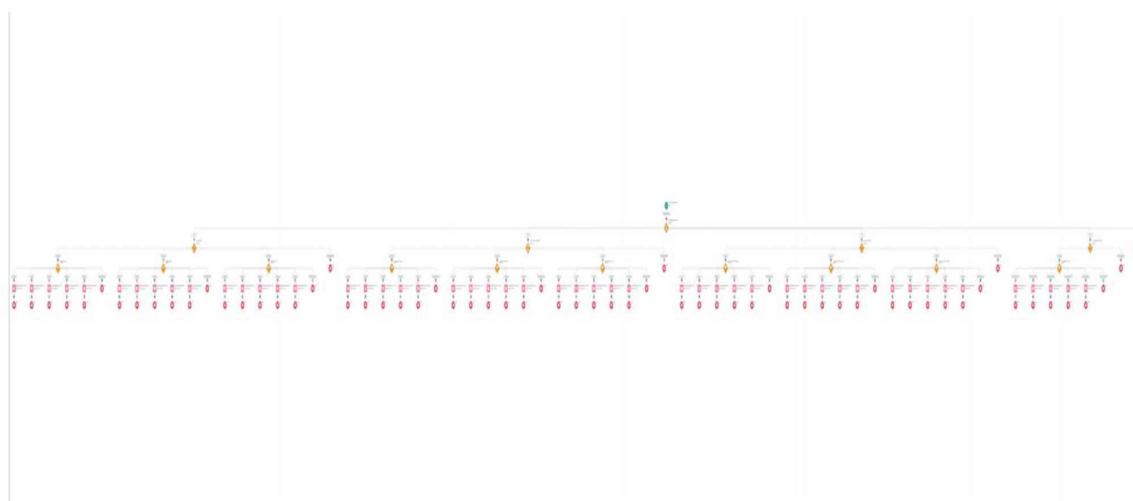
**Decision**

* Label months selected	* API Name months_selected
Description	
<b>Outcomes</b> For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.	
<b>OUTCOME ORDER</b> <span style="border: 1px solid #ccc; padding: 2px;">dell 1(i3)</span> <span style="border: 1px solid #ccc; padding: 2px;">dell 2(i3)</span> <span style="border: 1px solid #ccc; padding: 2px;">dell 3(i3)</span> <span style="border: 1px solid #ccc; padding: 2px;">dell 4(i3)</span> <span style="border: 1px solid #ccc; padding: 2px;">dell 5(i3)</span> <span style="border: 1px solid #ccc; padding: 2px;">Default Outcome</span>	<b>OUTCOME DETAILS</b> <p>* Label: dell 1(i3)</p> <p>* Outcome API Name: dell_1_i3</p> <p>Condition Requirements to Execute Outcome: All Conditions Are Met (AND)</p> <p>Resource: ...Bookings_c &gt; how many months Operator: Equals Value: 1</p> <p>+ Add Condition</p> <p>When to Execute Outcome: If the condition requirements are met</p>

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

## Final Outcome :

- All the four types:
  - Dell
  - Acer
  - HP
  - Mac



## 9: Apex Programming

**Apex** is a strongly typed, object-oriented programming language designed for the **Lightning Platform**, enabling developers to execute flow and transaction control statements directly on the platform server. It integrates seamlessly with calls to the Lightning Platform API. With a syntax similar to Java and functionality akin to database stored procedures, Apex allows developers to implement business logic in various contexts, such as button clicks, related record updates, and Visualforce pages.

Apex supports **Object-Oriented Programming (OOP)** concepts like classes, objects, and methods, making it intuitive for developers familiar with Java. Apex code can be triggered by web service requests or events like changes in database records, ensuring dynamic and responsive application behavior.

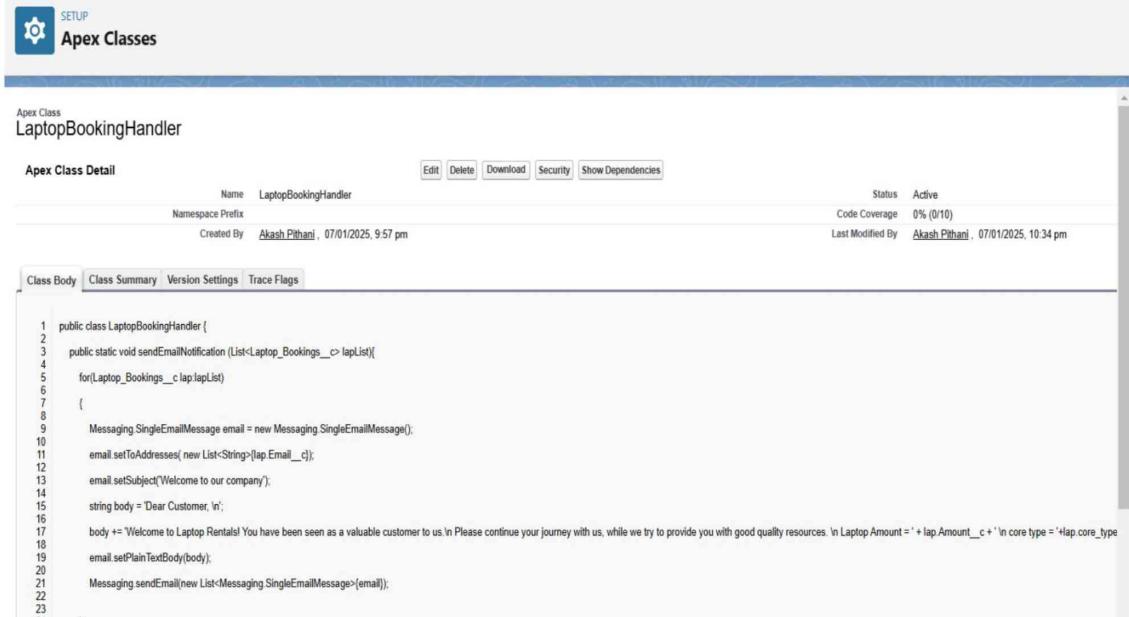
### Apex Classes:

**Apex Classes** are modeled after their Java counterparts, providing developers with a

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

familiar and robust framework for building applications. In Apex, you can define, instantiate, and extend classes, as well as implement interfaces. The language supports features like Apex class versions, properties, and other key concepts that align closely with object-oriented programming principles, offering flexibility and scalability for application development.

**Apex Class Name :** LaptopBookingHandler



The screenshot shows the Apex Class detail page for 'LaptopBookingHandler'. The page includes the class name, namespace prefix, creation date, status, code coverage, and last modified by user. The 'Class Body' tab is selected, displaying the following Apex code:

```
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 Customer, In';
9                 body += 'Welcome to Laptop Rentals! You have been seen as a valuable customer to us. Please continue your journey with us, while we try to provide you with good quality resources. \n Laptop Amount = ' + lap.Amount__c + '\n core type = ' + lap.core_type;
10                email.setPlainTextBody(body);
11                Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
12            }
13        }
14    }
```

**Apex Trigger:**

A **trigger** is a set of Apex code that executes automatically before or after **DML (Data Manipulation Language)** events, such as insert, update, or delete operations. Triggers allow developers to automate complex tasks that are challenging or impossible to perform through the Salesforce user interface alone. By creating custom scripts, developers can tailor automation to meet specific requirements, with the only constraint being their coding expertise. Triggers are powerful tools for enhancing business processes and ensuring data consistency within Salesforce.

**Apex Trigger Name:** LaptopBooking

LaptopBooking trigger will Automatically calls when(After Insert, After Update)

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

The screenshot shows the Apex Triggers page in Salesforce. The trigger is named 'LaptopBooking' and is of type 'Apex Trigger'. It was created by 'Akash Pithani' on 07/01/2025 at 9:55 pm. The code coverage is 0% (0/2). The trigger code is as follows:

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

## 10. Reports

**Reports** provide access to your Salesforce data, allowing you to analyze it in virtually limitless combinations. They enable you to display data in clear, easy-to-understand formats and share valuable insights with others.

I was created a **report** using the **Laptop Bookings** object to categorize rental data into buckets based on the **Amount\_\_c** field. Bookings are grouped into **Basic, Intermediate, High, and Very High** tiers, offering a clear and concise summary of booking distributions across different price ranges.

### Report Name: consumer with Laptop Bookings and total

The report displays data for 'consumer: consumer Name' and 'Laptop Booking: Laptop Bookings Name'. The report includes columns for 'Types of version', 'Consumer Name', 'Laptop Booking: Laptop Bookings Name', 'Laptops Available', 'Total No Of Laptops: Total Laptops Name', and 'Amount'. The data is categorized by 'basic (1)', 'Subtotal', 'intermediate (1)', 'Subtotal', and 'high (10)'. The report shows 12 total records, 560 total laptops available, and a total amount of ₹36,200.

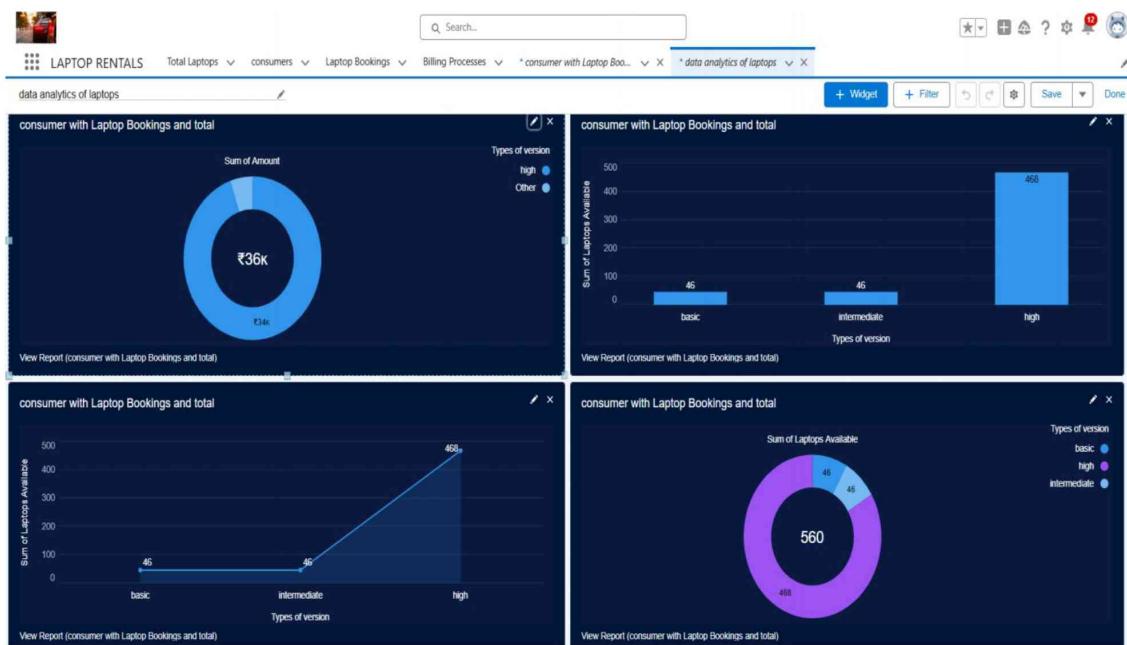
Types of version	Consumer Name	Laptop Booking: Laptop Bookings Name	Laptops Available	Total No Of Laptops: Total Laptops Name	Amount
basic (1)	Nitin	Acer	46	Acer	₹900
Subtotal			46		₹900
intermediate (1)	Sai	Amazon	46	Dell	₹1,000
Subtotal			46		₹1,000
high (10)	Naveen	Flipkart	46	Acer	₹1,800
	Murty	Moglix	48	hp	₹3,200
	Priya	Reliance	48	Mac	₹5,100
	Rama Krishna	Croma	46	Dell	₹2,000
	Akash	SnapDeal	46	Acer	₹3,000
	Madhu	HP	48	hp	₹5,300
	Jaswanth	Gadgets360	48	Mac	₹3,400
	Akash	LaptopMag	46	Acer	₹3,000
	Murty	PcMag	46	Dell	₹4,000
	Rama Krishna	Dell	46	Dell	₹3,500
Subtotal			468		₹34,300
Total (12)			560		₹36,200

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

## 11. Dashboards

**Dashboards** provide a visual representation of real-time business data, helping you monitor changing conditions and make informed decisions. Powered by reports, dashboards enable users to identify trends, analyze quantities, and assess the impact of their activities. Before creating, interpreting, and sharing dashboards, it's important to review the basics to maximize their effectiveness and usability.

In this Dash Board we can see **A donut chart** and some more charts were added to the dashboard to visually represent the **Laptop Bookings report**, segmented by the **pricing buckets—Basic, Intermediate, High, and Very High**.



The Above DashBoard Displays the Overall Information about the Laptop Bookings.

## 12. Testing

This Laptop rental Application was tested by various ways and provides accurate results.

# Title: A CRM APPLICATION FOR LAPTOP RENTALS

**Testing By Apex codes:** All Apex classes and triggers were tested using **unit tests**, which were designed to check if each part of the code works correctly. This testing achieved 100% code coverage, ensuring that every piece of the code was tested and works as intended.

**User Interface Testing:** The user interface, including how the laptop booking process flows, how users move between different tabs, and how forms handle input validations, was thoroughly tested both by hand and using test scripts to ensure everything works smoothly and without errors.

**Thorough testing** ensures the reliability and usability of the application, creating a seamless experience for users. By achieving 100% code coverage in **unit testing**, developers can confidently deploy the application, knowing that every part of the code has been checked for errors. Similarly, detailed **user interface testing** helps identify and fix issues related to navigation, form validations, and overall workflow, ensuring that users can complete tasks without confusion or interruption. This comprehensive approach to testing not only improves the application's quality but also builds user trust and satisfaction.

## **13. Key Use cases Addressed in the Implementation Project by Salesforce**

- **Improved Customer Engagement:**

Personalized and automated email notifications keep customers informed throughout their rental journey, including booking confirmations, reminders, and follow-ups.

- **Streamlined Customer Data Management**

Salesforce efficiently captures and organizes customer data, enabling easy access to detailed information such as booking history, contact details and preferences for effective customer management.

## **14. Conclusion**

In conclusion, the implementation of the Laptop Rentals CRM represents a modern solution for managing customer relationships, blending automation, tailored communication, and powerful data analytics. This approach not only streamlines operations but also significantly enhances the overall customer experience, helping the business remain competitive and positioned for continued growth and long-term success in the rental market.

**Pithani Akash**

**Vishnu Institute of Technology**