

Salesforce CRM Solution for Laptop Rental Management

1. Project Overview:

This project is dedicated to developing a robust CRM application tailored for managing laptop rentals within a business environment using the Salesforce platform. The primary goal is to tackle operational challenges, including handling rental requests efficiently, monitoring inventory, and facilitating effective communication with customers. Leveraging Salesforce's strengths in automation, workflow optimization, and data storage, this CRM application aims to:

1. Optimize the Rental Workflow: Streamline the booking and return processes to reduce time and minimize manual intervention required for each rental.
2. Enhance Customer Satisfaction: Send timely email notifications to customers for booking confirmations and return reminders.
3. Increase Data Accuracy and Insightful Reporting: Provide real-time visibility into laptop inventory and rental status, empowering staff to make informed, quick decisions.

2. Objective Goals:

1. Streamline the laptop rental and return processes to minimize manual effort, enhancing efficiency and accuracy.
2. Strengthen customer management practices to improve service quality and foster lasting customer relationships.
3. Implement precise tracking and reporting for laptop inventory to ensure accurate availability and effective asset management.

Specific outcomes:

- A customized Salesforce CRM application designed to manage rental services with an intuitive interface, allowing users to easily view, manage, and track rental activity.
- Automated workflows streamline the handling of rental requests, provide real-time status updates, and send email notifications.
- Comprehensive reports and dashboards deliver real-time insights into inventory levels, rental activity, and customer engagement.

3. Salesforce Key Features and Concepts Utilized:

This section details the core Salesforce functionalities and tools used in the Laptop Rentals CRM application.

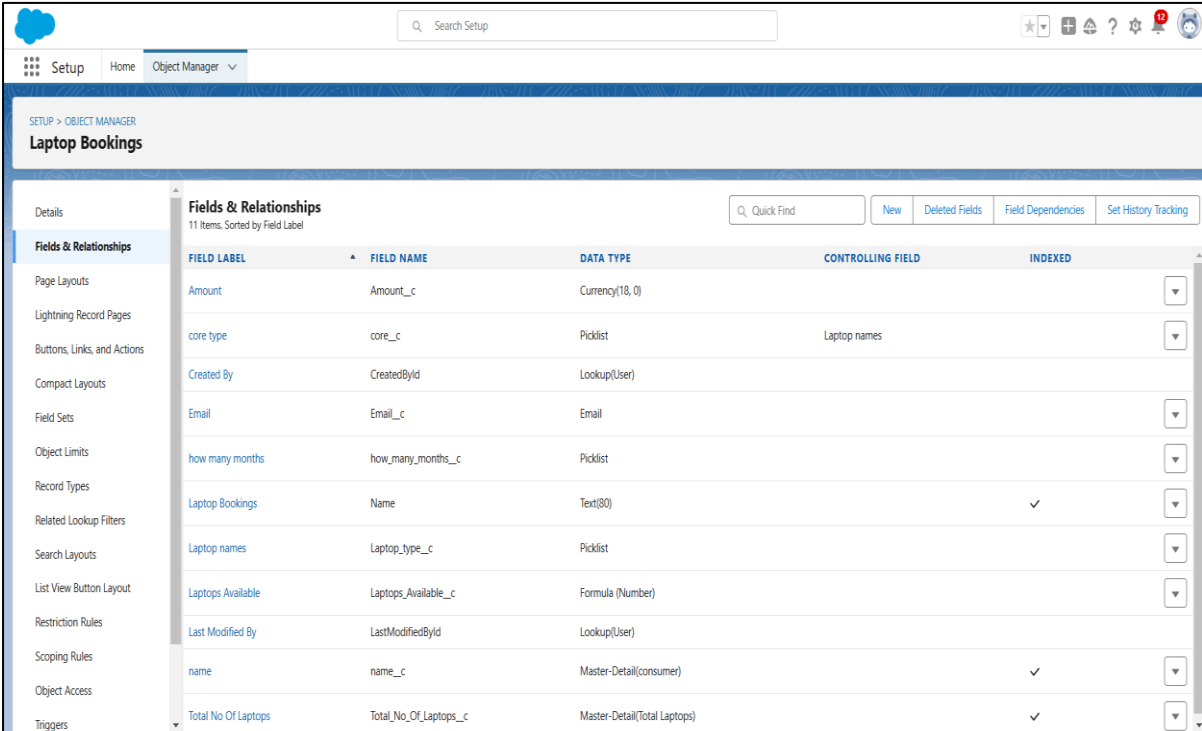
- **Custom Objects and Fields:**
 - Created a custom object, `Laptop_Bookings__c`, to manage rental bookings with fields for Customer Name, Email, Amount, Core, Laptop Type, and Status.
 - Set up a `Laptops__c` object to track inventory, capturing details like model, specifications, and availability.
- **Apex Triggers and Classes:**
 - Developed an `AfterInsert` **trigger** (`LaptopBooking`) to send automated confirmation emails when a booking is made. The `LaptopBookingHandler` class formats and personalizes these emails with customer details.
- **Process Automation with Process Builder and Flows:**
 - Designed a Process Builder flow to manage booking status changes (e.g., from "Booked" to "In Use" to "Returned") without manual updates.
- **Validation Rules for Data Accuracy:**
 - Added rules to enforce data quality, requiring key fields like Email and Amount for complete booking records.
- **Reports and Dashboards:**
 - Created dashboards and reports to provide insights on **inventory, rental volumes, popular laptop models**, and customer demographics, supporting strategic planning and resource allocation.

4. Detailed Steps to Solution design:

The solution was carefully designed in multiple steps, covering data models, user - interface, and business logic. Here's an overview:

- **Data Model Design:**
 - Created **Laptop_Bookings__c** and **Laptops__c** custom objects.
 - Established relationships between **Laptop_Bookings__c** and standard Salesforce objects (Account or Contact) to connect customers with their respective bookings.
 - Additional fields, such as **Email__c**, **Amount__c**, **Core__c**, and **Laptop_Type__c**, are used to store booking-specific data. Relationships between objects are established to link the **Laptop_Bookings__c** object with other standard or custom objects (e.g., Account, Contact) to enhance data integrity and support comprehensive reporting.

Fields Included in the **LaptopBookings** object:



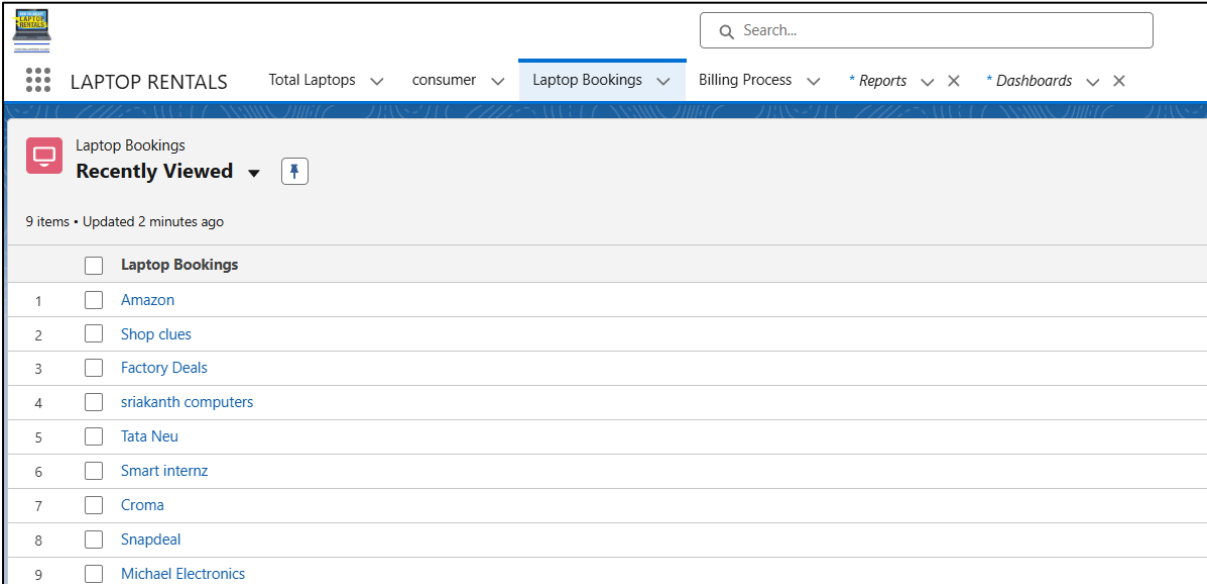
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Currency(18, 0)		
core type	core__c	Picklist	Laptop names	
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
how many months	how_many_months__c	Picklist		
Laptop Bookings	Name	Text(80)		✓
Laptop names	Laptop_type__c	Picklist		
Laptops Available	Laptops_Available__c	Formula (Number)		
Last Modified By	LastModifiedById	Lookup(User)		
name	name__c	Master-Detail(consumer)		✓
Total No Of Laptops	Total_No_Of_Laptops__c	Master-Detail(Total Laptops)		✓

- **User Interface (UI) Design:**

The custom Lightning app **Laptop Rentals** features easy navigation with tabs for key components, including **Bookings, Reports, and Dashboards**.

- Configured Page Layouts for **Laptop_Bookings__c** and **Laptops__c** with relevant fields, sections, and related lists for ease of access.
- Designed dashboards for quick visual insights on active rentals, availability and top-performing laptops.
- Custom Lightning components may be added to enhance specific sections, such as displaying a chart of most rented laptop models.

Navigation Items on Laptop Rental Application:



The screenshot displays the 'Laptop Rentals' Lightning App interface. At the top, there is a search bar and a navigation bar with tabs for 'Total Laptops', 'consumer', 'Laptop Bookings' (selected), 'Billing Process', '* Reports', and '* Dashboards'. Below the navigation bar, the 'Laptop Bookings' section is titled 'Recently Viewed' with a dropdown arrow and a share icon. It indicates '9 items • Updated 2 minutes ago'. The main content area is a table listing the recently viewed items.

	<input type="checkbox"/> Laptop Bookings
1	<input type="checkbox"/> Amazon
2	<input type="checkbox"/> Shop clues
3	<input type="checkbox"/> Factory Deals
4	<input type="checkbox"/> sriakanth computers
5	<input type="checkbox"/> Tata Neu
6	<input type="checkbox"/> Smart internz
7	<input type="checkbox"/> Croma
8	<input type="checkbox"/> Snapdeal
9	<input type="checkbox"/> Michael Electronics

- **Business Logic Design:**

- Developed the LaptopBookingHandler class to handle email notifications using **Messaging.SingleEmailMessage**.
- Added the **AfterInsert trigger** on **Laptop_Bookings__c** to initiate sendEmailNotification method, ensuring customers receive a welcome and confirmation email after booking.

Flow Automation:

Decision

* Label

field must updated

* API Name

field_must_updated

Description

the field must be updated

Outcomes

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.

OUTCOME ORDER

+

dell i3

dell i5

dell i7

Default Outcome

OUTCOME DETAILS

Delete Outcome

* Label

dell i3

* Outcome API Name

dell_i3

Condition Requirements to Execute Outcome

All Conditions Are Met (AND)

Resource

...ptop_Bookings__c > core type

Operator

Equals

Value

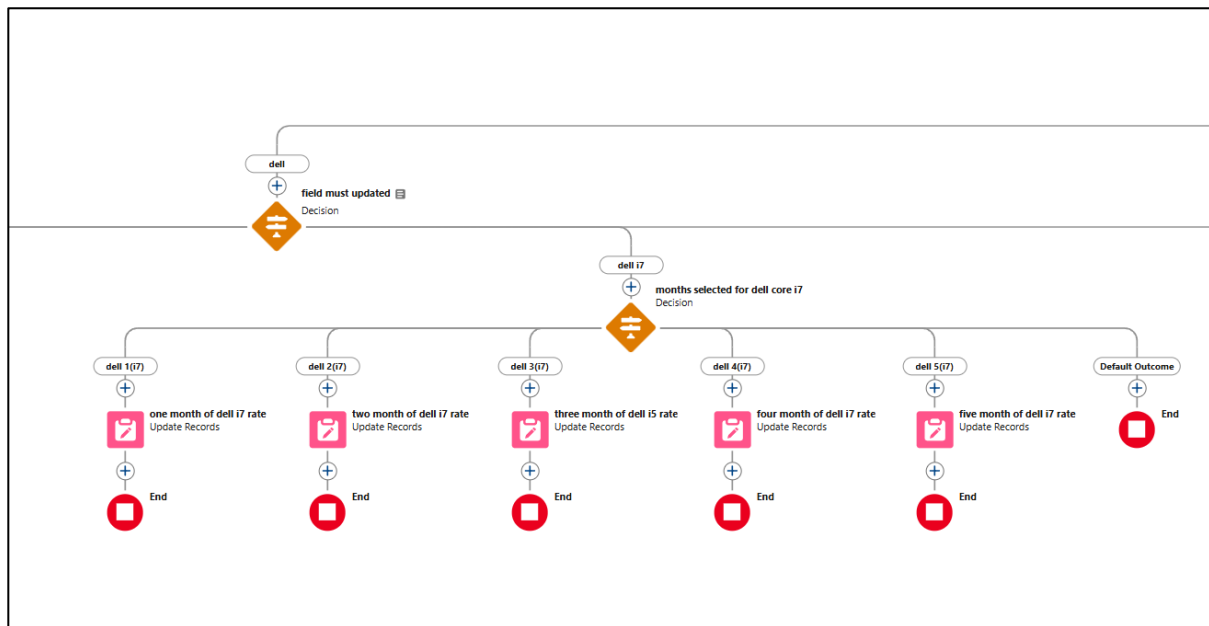
core i3

+ Add Condition

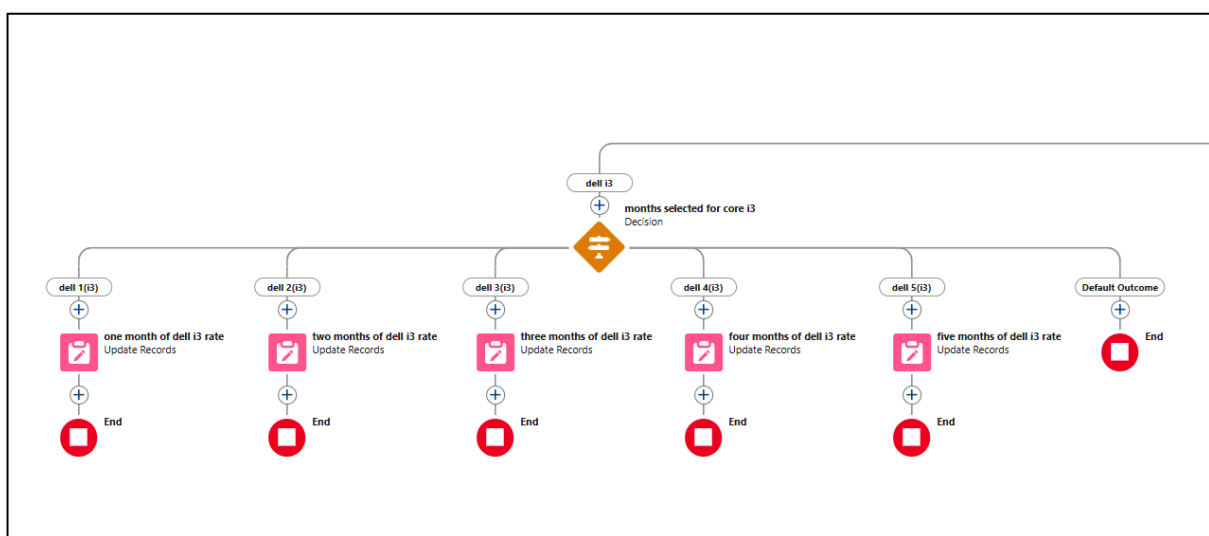
When to Execute Outcome

☒ If the condition requirements are met

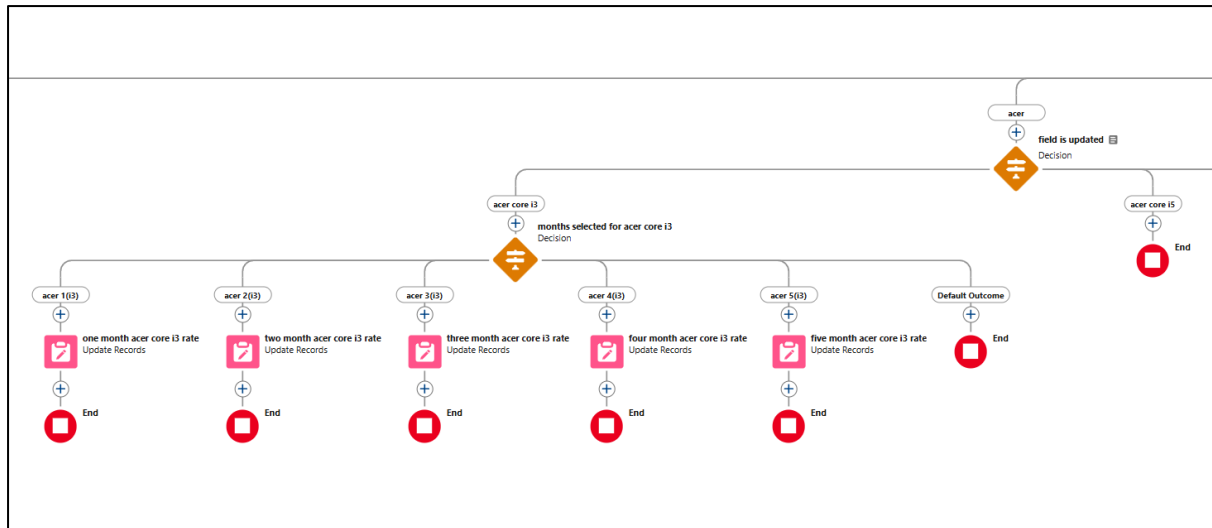
☐ Only if the record that triggered the flow to run is updated to meet the condition requirements



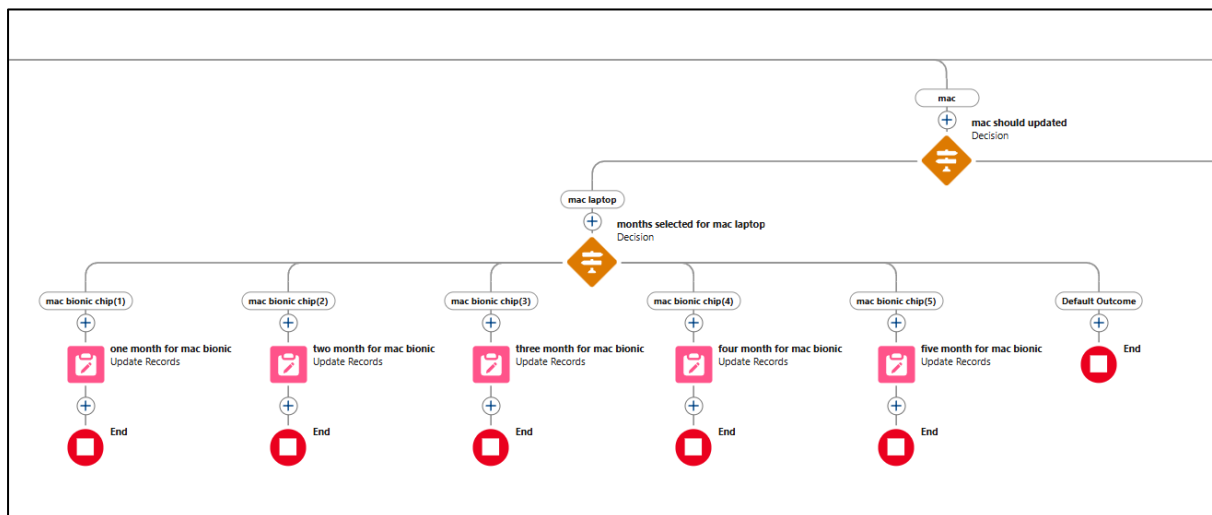
Dell Laptop Flow:



Acer Laptop Flow:



Mac Laptop Flow:



Validation Rules:

SETUP > OBJECT MANAGER

consumer

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Object Access


Triggers

Validation Rules

1 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Phonenumberoremailblankrule	Top of Page	please fill the phone number and email id	✓	LAKSHMI VENKATA SATYA GANESH JAVVADI, 26/10/2024, 12:03 pm

Conditional Formula:



consumer

Setup Home Object Manager

SETUP > OBJECT MANAGER

consumer

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

consumer Validation Rule

Back to consumer

Help for this Page

Validation Rule Detail

EditClone

Rule Name

Phonenumberoremailblankrule

Active

✓

Error Condition Formula

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

Error Message

please fill the phone number and email id

Error Location

Top of Page

Description

phone number and email number should not be blank

Created By

LAKSHMI VENKATA SATYA GANESH JAVVADI, 26/10/2024, 12:03 pm

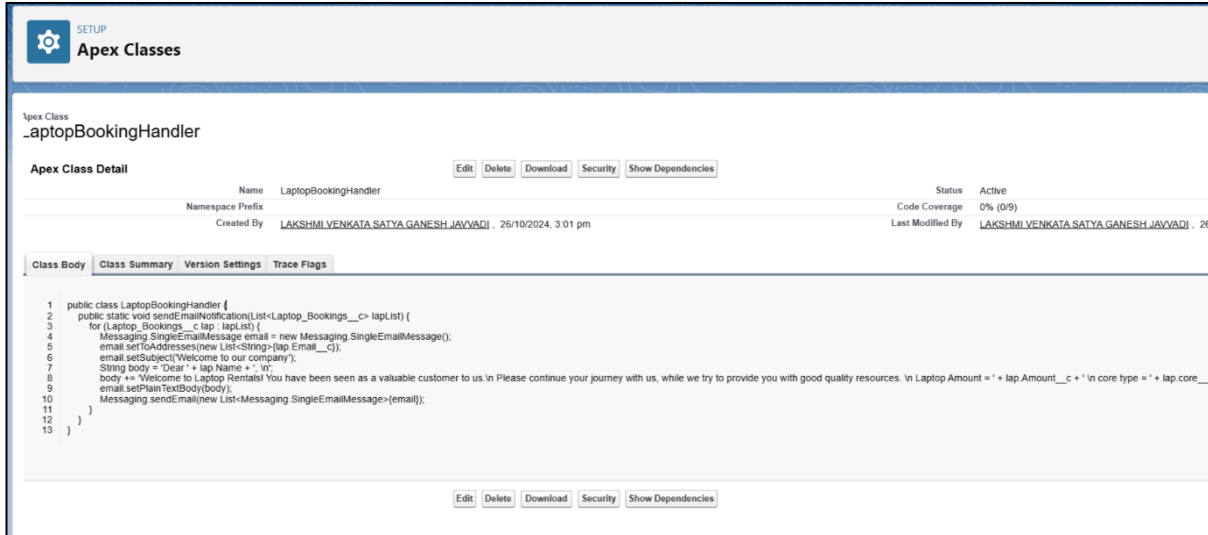
Modified By

LAKSHMI VENKATA SATYA GANESH JAVVADI, 26/10/2024, 12:03 pm

EditClone

APEX Classes:

Created Apex class with name of **LaptopBookingHandler**



The screenshot shows the Salesforce Apex Class Editor interface. At the top, there's a "SETUP" button and the title "Apex Classes". Below this, the class name "LaptopBookingHandler" is displayed. A table provides details about the class:

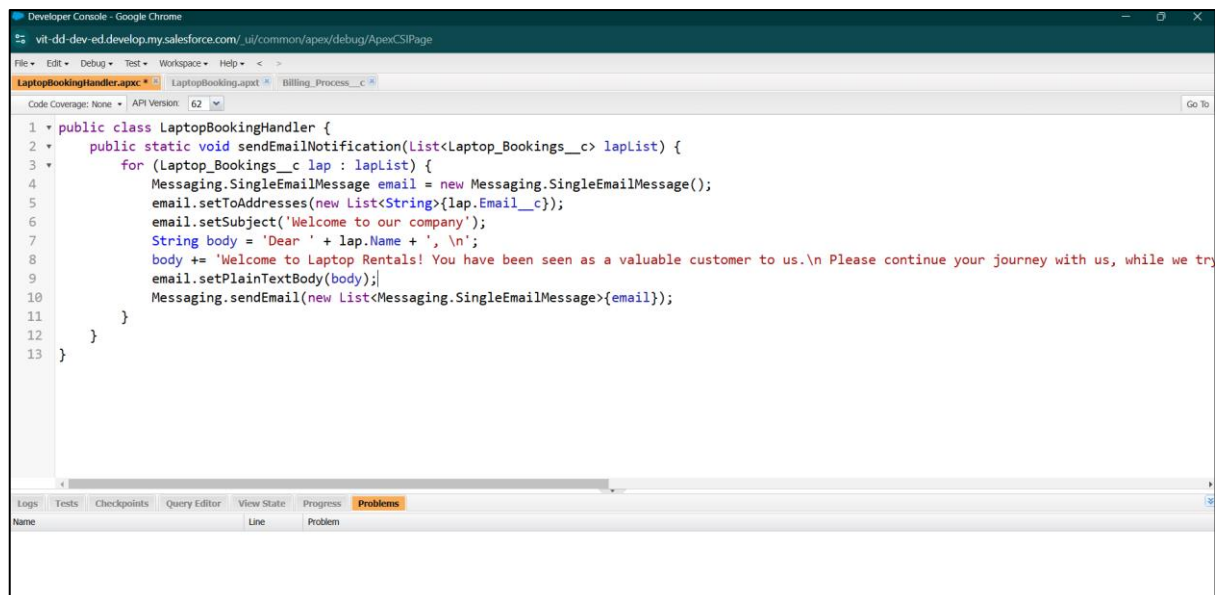
Name	Namespace Prefix	Status
LaptopBookingHandler		Active

Below the table, it shows the creator "LAKSHMI VENKATA SATYA GANESH JAYVADI" and the creation date "26/10/2024, 3:01 pm". The "Code Coverage" is listed as "0% (0/9)".

The "Class Body" tab is selected, showing 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             Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
5             email.setToAddresses(new List<String>{lap.Email__c});
6             email.setSubject('Welcome to our company');
7             String body = 'Dear ' + lap.Name + ', \n';
8             body += 'Welcome to Laptop Rentals! You have been seen as a valuable customer to us. \n Please continue your journey with us, while we try to provide you with good quality resources. \n Laptop Amount = ' + lap.Amount__c + ' \n core type = ' + lap.Core_Type__c;
9             email.setPlainTextBody(body);
10            Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
11        }
12    }
13 }
```

Developer Console:



The screenshot shows the Salesforce Developer Console interface. The top bar indicates the URL "vit-dd-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage". The "LaptopBookingHandler.apxc" file is open in the editor. The code is as follows:

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

APEX TRIGGERS:

Created an APEX Trigger with name of **LaptopBooking**

The screenshot shows the Salesforce 'Apex Triggers' setup page. The trigger is named 'LaptopBooking' and is associated with the 'Laptop Bookings' object. The status is 'Active'. The code coverage is 0% (0/2). The trigger was created by 'LAKSHMI VENKATA SATYA GANESH JAVVADI' on 26/10/2024 at 3:07 pm. The namespace prefix is empty. The trigger code is displayed in the 'Apex Trigger' tab, showing a trigger that calls 'LaptopBookingHandler.sendEmailNotification' when a new record is inserted or updated.

Apex Trigger Detail

Name	sObject Type	Status
LaptopBooking	Laptop Bookings	Active

Code Coverage: 0% (0/2)

Created By: LAKSHMI VENKATA SATYA GANESH JAVVADI, 26/10/2024, 3:07 pm

Last Modified By: LAKSHMI VENKATA SATYA GANESH JAVVADI, 26/10/2024, 3:07 pm

Namespace Prefix:

Apex Trigger

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

Developer Console:

The screenshot shows the Salesforce Developer Console. The URL is 'vit-dd-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage'. The console displays the Apex trigger code for 'LaptopBooking.apxt'.

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

Reports and Dashboards:


Reports and dashboards in Salesforce provide valuable insights into the Laptop Rentals CRM application, helping the business monitor bookings, revenue, and customer trends effectively.

Reports:

- **Tabular Reports:** Used for simple lists, such as a full list of active laptop bookings or consumer records, giving a quick snapshot of data.
- **Summary Reports:** Offers data with grouped subtotals, such as revenue grouped by laptop type, making it easy to identify high performing categories.

Dashboards:

- **Revenue Overview:** Shows total rental revenue over time, helping track financial performance.
- **Popular Laptops:** Displays the most rented laptop types, aiding in inventory and marketing decisions.
- **Customer Insights:** Provides data on customer types, helping target future marketing campaigns more effectively.



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

Total Records	Total Amount
9	₹18,500

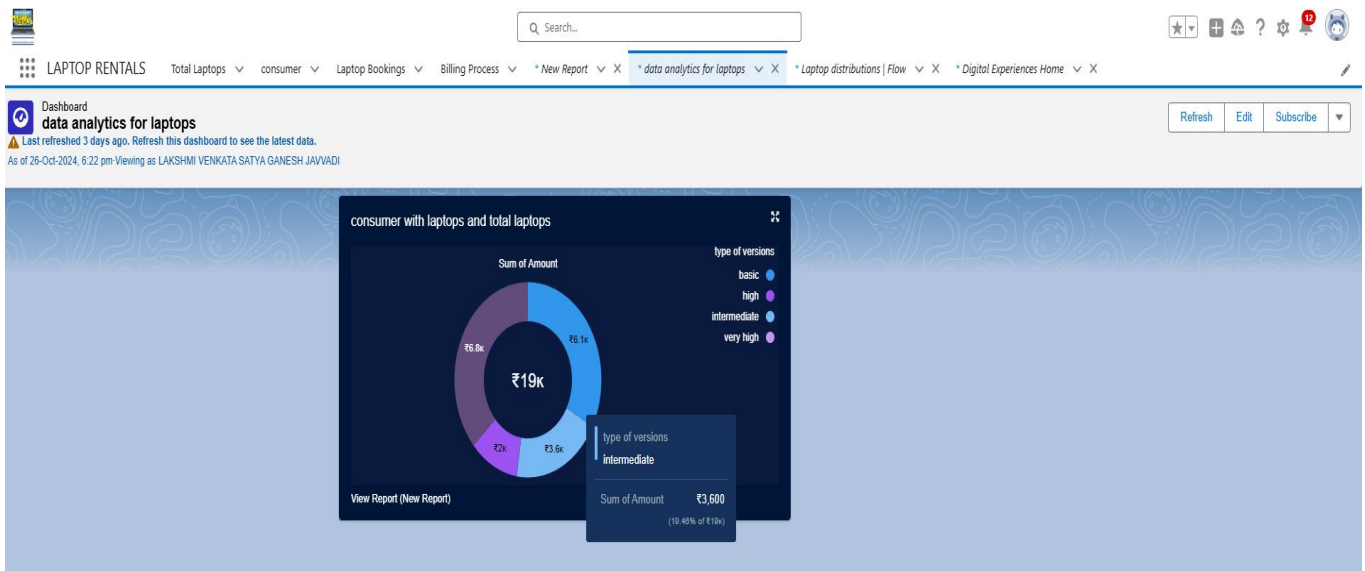
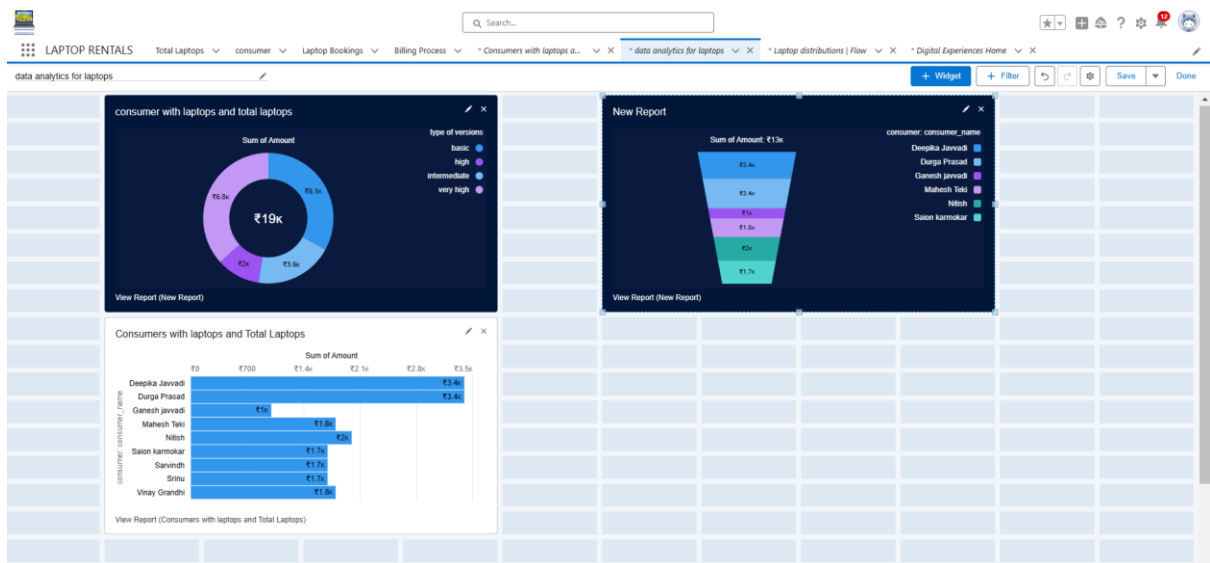
<input type="checkbox"/> consumer: consumer_name ↑	Laptop Bookings: Laptop Bookings	Laptop names	Address	Phone number	type of versions	Amount
<input type="checkbox"/> Deepika Javvadi (1)	Croma	Mac	Tanuku	9398341710	high	₹3,400
Subtotal						₹3,400
<input type="checkbox"/> Durga Prasad (1)	Tata Neu	Hp	kakinada	7396433224	high	₹3,400
Subtotal						₹3,400
<input type="checkbox"/> Ganesh javvadi (1)	Michael Electronics	Dell	Tanuku	7780617696	intermediate	₹1,000
Subtotal						₹1,000
<input type="checkbox"/> Mahesh Teki (1)	Amazon	Acer	undrajavaram , Tanuku	8074643716	high	₹1,800
Subtotal						₹1,800
<input type="checkbox"/> Nitish (1)	Shop clues	Dell	Rajahmundry	9912930336	high	₹2,000
Subtotal						₹2,000
<input type="checkbox"/> Saion karmokar (1)	Snapdeal	Mac	Swetha Hospital,Tanuku	7993258988	intermediate	₹1,700
Subtotal						₹1,700
<input type="checkbox"/> Sarvindh (1)	Factory Deals	Mac	jagayyapeta@vishnu.edu.in	8978109692	intermediate	₹1,700
Subtotal						₹1,700
<input type="checkbox"/> Srinu (1)	Smart internz	Hp	Tanuku	9848131118	intermediate	₹1,700
Subtotal						₹1,700
<input type="checkbox"/> Vinay Grandhi (1)	srikanth computers	Acer	Kakinada	9381691645	high	₹1,800
Subtotal						₹1,800
Total (9)						₹18,500

Row Counts

Detail Rows

Subtotals

Grand Total



5. Testing and Validation:

- **Unit Testing:**

Each Apex class and trigger was thoroughly tested, ensuring at least 75% code coverage. Tests were designed to validate that:

- Booking confirmation emails are sent correctly.
- Inventory statuses update accurately.

- **User Interface Testing:**

Verified that the UI elements display correctly on page layouts and that all required fields and related lists are accessible. Ensured users could access only the fields allowed by their profile permissions.

- **Test Scenarios:**

- **Scenario 1:**

Create a new booking and verify that an email is sent with the correct booking details.

- **Scenario 2:**


Check that validation rules enforce required fields like Email__c and Amount__c.

- **Scenario 3:**

Test that inventory adjust automatically when a booking status is changed from "Booked" to "Returned."

Apex class:

Create a Testing class **LaptopBookingTest.apxc**

 **Apex Classes**

Name	LaptopBookingHandlerTest	Status	Active
Namespace Prefix		Created By	LAKSHMI VENKATA SATYA GANESH JAVVADI
Last Modified By	LAKSHMI VENKATA SATYA GANESH JAVVADI		29/10/2024, 10:31 pm

Class Body

Class Summary

Version Settings

Trace Flags

```
1 // Test Class
2 @isTest
3 private class LaptopBookingHandlerTest {
4     @isTest
5     static void testSendEmailNotification() {
6         // Setup test data for consumer
7         consumer__c consumer = new consumer__c(
8             Name = 'John Doe',
9             Phone_number__c = '9887346134',
10            Address__c = 'Paris, France',
11            Email__c = 'ganeshjavvadi367@gmail.com',
12            consumer_Status__c = 'Student'
13        );
14        insert consumer;
15        System.debug('Consumer Id: ' + consumer.Id);
16
17        // Setup test data for total laptops
18        Total_Laptops__c totalLaptops = new Total_Laptops__c(
19            Name = '1'
20        );
21        insert totalLaptops;
22
23        // Setup test data for laptop booking
24        Laptop_Bookings__c testBooking = new Laptop_Bookings__c(
25            Name = 'Booking 1',
26            name__c = consumer.Id,
27            Email__c = consumer.Email__c,
28            Amount__c = 900,
29            core__c = 'Core i3',
30            Laptop_Type__c = 'Dell',
31            Total_No_Of_Laptops__c = totalLaptops.Id,
32            How_Many_Months__c = '1'
33        );
34        insert testBooking;
35
36        // Test email notification
37        Test.startTest();
38        // Insert to trigger the notification in the trigger itself
39        insert testBooking;
40        Test.stopTest();
41
42        // Assert email sent count
43        Integer sentEmailCount = Limits.getEmailInvocations();
44        System.assertEquals(1, sentEmailCount, 'One email should have been sent');
45    }
46 }
```

6. Key Scenarios Addressed by Salesforce in the Implementation Project:

This project addresses several critical scenarios to ensure a smooth rental experience for both customers and staff:

- **Automated_Notifications:**

Automatically send confirmation emails when a booking is created and provide status updates as the rental progresses. This keeps customers informed and reduces the workload for staff.

- **Real-Time_Inventory_Management:**

Track the availability of laptops in real time, updating inventory counts when laptops are rented and returned.

- **Workflow_Automation_for_Booking_Lifecycle:**

Manage booking statuses with automation, moving each booking from "Booked" to "In Use" and finally "Returned" without manual intervention.

- **Role-Based_Data_Access:**

Define roles and permissions, allowing staff to manage bookings and customers, while customers can only access their rental information.

7. Conclusion:

The **Laptop Rentals CRM** application demonstrates an effective use of Salesforce to automate and manage key rental business processes. Leveraging custom objects, Apex programming, and flow automation, the project streamlines data management and enhances customer interactions. Key features like email notifications, dynamic pricing, and streamlined booking processes have improved both customer experience and operational efficiency.

Salesforce's reporting and dashboard tools offer valuable insights into rental trends, revenue tracking, and decision-making. With validation rules and role-based access, the solution ensures data security and integrity, supporting sustainable business growth. This project not only meets immediate objectives but also establishes a scalable foundation for future enhancements, positioning Laptop Rentals for long-term success in customer satisfaction and operational efficiency.