

Project Title

A CRM APPLICATION FOR LAPTOP RENTALS

1. Project Overview

This project is focused on developing a CRM Application for Laptop Rentals, designed to address the primary challenge of efficiently managing rental operations and customer interactions in the laptop rental industry. The goal is to deliver a comprehensive solution by leveraging Salesforce CRM technology to streamline customer management, track rental orders, and facilitate effective communication with potential customers. Through this project, we aim to enhance customer experience, operational efficiency, and communication effectiveness, supporting the long-term goals of maintaining high customer satisfaction and maximizing business efficiency in the rental services department.

2.Objectives

Business Goals:

- Streamline the laptop rental process by providing an automated, end-to-end management system within Salesforce.
- Improve booking accuracy and efficiency, reducing manual data entry and potential human errors.
- Enhance customer experience with quick and reliable booking, billing, and tracking of laptop rentals.
- Enable data-driven decision-making by providing real-time insights into inventory, booking trends, and revenue.

Specific Outcomes:

- **Automated Laptop Booking Process:** Implement a seamless booking system through custom objects and record-triggered flows, ensuring accurate laptop and processor selections.
- **Billing Automation:** Generate automated billing amounts based on laptop type, processor, and rental duration, minimizing manual calculations.
- **Real-time Inventory Management:** Track available and booked laptops using custom fields, roll-up summaries, and validation rules to maintain inventory accuracy.
- **User Management and Security:** Set up custom profiles, roles, and validation rules to ensure secure and role-based access to the system.
- **Enhanced User Interface:** Develop a user-friendly Lightning app, "Laptop Rentals," with

organized tabs for easy navigation through booking, consumer data, and billing processes.

- **Data Validation:** Apply validation rules for essential fields like phone number and email to improve data integrity and quality.

- **Conditional Process Automation:** Implement Record-Triggered Flow decision elements to categorize bookings by laptop and processor type, automating updates to billing amounts based on specific conditions.

3. Salesforce Key Features and Concepts Utilized

1. Custom Objects and Fields

- Created a custom object, Laptop_Bookings__c, to manage laptop rental bookings, with fields for storing details like laptop model, processor type, rental duration, customer information, and calculated rental amounts.
- Defined custom fields such as Total_Amount__c, Laptop_Type__c, Core_Type__c, and others to capture specific booking details.

2. Validation Rules

- Implemented validation rules to ensure data quality and integrity. For example, validations for fields like Email__c and Phone__c ensure correct formats, while others ensure that key fields are not left empty.
- Used conditional checks in validation rules to enforce business rules, such as ensuring that the rental duration is greater than zero.

3. Profiles and Roles

- Configured profiles and roles to control access and permissions. This ensures that users can only view, edit, or delete records according to their roles.
- Set up custom profiles to provide role-based access to specific Salesforce objects, fields, and records.

4. Record-Triggered Flows

- Automated the booking process using Record-Triggered Flows. Flows update or create records based on conditions defined in decision elements, such as the type of laptop or processor.
- Automated calculations for rental amounts, which adjust dynamically based on user inputs for laptop type, cores, and rental duration.

5. Apex Triggers and Handler Classes.

- Created Apex triggers to automate actions based on DML events (e.g., before insert, before update) on the Laptop_Bookings__c object.
- Used a handler class, LaptopBookingHandler, to separate business logic from the trigger, ensuring a cleaner, modular, and maintainable code structure.
- The handler class includes methods to automate tasks like sending email

notifications to customers upon booking.

6. Email Notifications

- Automated email notifications using Apex to send a personalized message to the customer with booking details, such as laptop type, amount, and other rental information.

7. Reports and Dashboards

- Created various reports (e.g., tabular, summary, matrix) to visualize data like total bookings, revenue by laptop type, and rental trends.
- Built dashboards to provide a graphical representation of key metrics, enabling stakeholders to make data-driven decisions based on real-time insights.

8. Custom Lightning App

- Developed a custom Lightning app named "Laptop Rentals" for easy navigation. The app includes organized tabs for managing bookings, consumers, and reports.
- Utilized Salesforce Lightning to enhance the user interface, making it more intuitive and user-friendly.

9. Access Specifiers in Apex

- Used access modifiers (private, protected, public, and global) in Apex classes and methods to control visibility and encapsulate business logic.
- Applied these concepts to ensure that the code adheres to best practices in object-oriented programming.

10. Data Manipulation Language (DML) Operations

- Performed DML operations (insert, update) in Apex to automate record updates based on business rules.
- Ensured that the system can handle bulk updates and processes records efficiently.
- These features and concepts work together to create a comprehensive laptop rental management system within Salesforce, automating key processes, ensuring data accuracy, and providing valuable insights for decision-making.

4. Detailed Steps to Solution Design

1. Data Models Description: The main custom object, Laptop_Bookings__c, captures details such as the laptop model, processor type, rental duration, customer information, and total amount. Additional fields like Email__c, Amount__c, Cores__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.

Screenshots: Fields included in Laptop Bookings object are:

Setup	Home	Object Manager
-------	------	----------------

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

Scoping Rules

Fields & Relationships

11 Items, Sorted by Field Label

Core type

Core_type__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_names__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)

✓

2. User Interface Designs Description: The user interface design aims to provide a streamlined and userfriendly experience for managing laptop bookings. The custom Lightning app "Laptop Rentals" features easy navigation with tabs for key components such as bookings, reports, and dashboards. Page layouts for the Laptop_Bookings__c object are designed to display relevant information at a glance, with important fields like laptop model, processor type, and rental details prominently shown. Custom Lightning components may be added to enhance specific sections, such as displaying a chart of most rented laptop models.

Screenshots: Laptop Rentals App along with navigation items

←	↺	https://shrivishnuengineeringcolle5-dev-ed.develop.lightning.force.com/lightning/o/consumer__c/list?filterName=__Recent
---	---	-------------------------------------------------------------------------------------------------------------------------

LAPTOP RENTALS

Total Laptops

consumer

Laptop Bookings

Billing Process

Reports

* Laptop Bookings

* Data Analytics of L

consumer

Recently Viewed

4 items • Updated a few seconds ago

Search this lis

consumer_name

1 Mamatha

2 Rama

3 Ramya

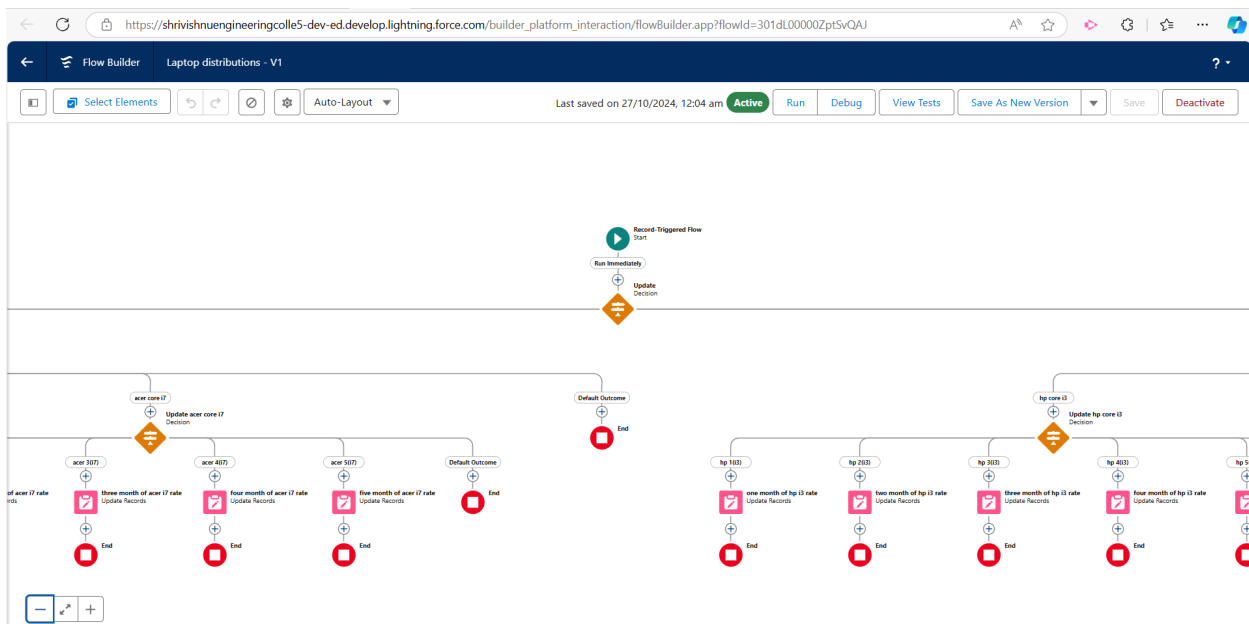
4 John Doe

	<input type="checkbox"/> Laptop Bookings	
1	<input type="checkbox"/> Dell	
2	<input type="checkbox"/> Dell	
3	<input type="checkbox"/> Dell	
4	<input type="checkbox"/> Mac	
5	<input type="checkbox"/> Dell	
6	<input type="checkbox"/> Hp	
7	<input type="checkbox"/> Acer	
8	<input type="checkbox"/> Dell XPS i5	
9	<input type="checkbox"/> Dell XPS 13	

3. Business Logic

Description: The business logic is implemented through a combination of validation rules, flows, Apex triggers, and handler classes. Validation rules ensure data quality by enforcing conditions, such as valid email format and minimum rental duration. RecordTriggered Flows automate key processes like updating total rental amount based on selected laptop type and processor. Apex triggers and handler classes are used to perform more complex logic, such as sending email notifications upon booking confirmation or recalculating amounts if certain fields are updated.

Screenshots: Flow Automation



https://shrivishnuengineeringcolle5-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301dL00000ZptSvQAJ

Flow Builder Laptop distributions - V1

Last saved on 27/10/2024, 12:04 am **Active** Run Debug View Tests Save As New Version

Update Records

* Label: five month of acer i7 rate * API Name: five_month_of_acer_i7_rate

Description:

How to Find Records to Update and Set Their Values

- ☒ Use the laptop bookings record that triggered the flow
- ☐ Update records related to the laptop bookings record that triggered the flow
- ☐ Use the IDs and all field values from a record or record collection
- ☐ Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record: None—Always Update Record

Set Field Values for the Laptop Bookings Record

Field: Amount__c Value: 6800

+ Add Field

Update acer core i7
Decision

acer 4 (i7) acer 5 (i7) Default Outcome

four month of acer i7 rate Update Records five month of acer i7 rate Update Records

End End End

Validation rules for consumer object:

https://shrivishnuengineeringcolle5-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01dL000002xmFZ/ValidationRules/03ddL0000044pwzQAA/view

Setup Home Object Manager

SETUP > OBJECT MANAGER
consumer

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
Flow Triggers
Validation Rules

consumer Validation Rule
[Back to consumer](#)

Validation Rule Detail

Rule Name	Phonenumberoremailblankrule	Active	<input checked="" type="checkbox"/>
Error Condition Formula	OR(ISBLANK(Phone_number__c), ISBLANK(Email__c))	Error Location	Top of Page
Error Message	please fill the phone number and email id	Modified By	Syamy Penumala, 26/10/2024, 9:54 pm
Description	phone number and email number should not be blank		
Created By	Syamy Penumala, 26/10/2024, 9:54 pm		

Creating the Apex class:

In this example, LaptopBookingHandler is the class name.

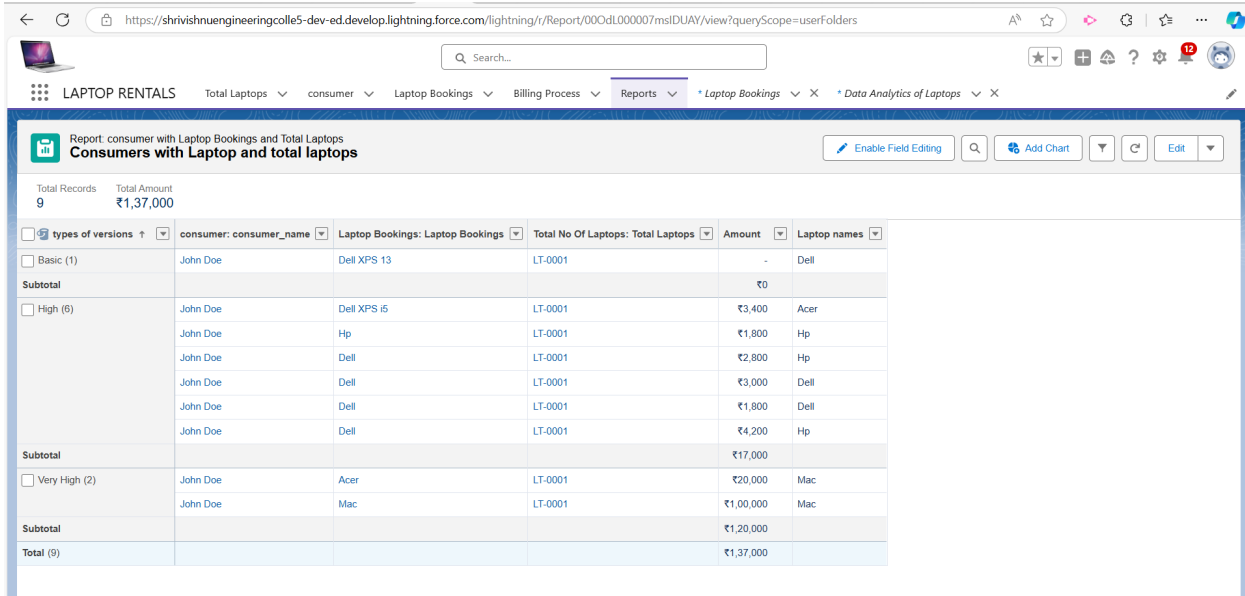
Creating apex trigger: Here is the LaptopBooking trigger.

4. Reports and Dashboards Description:

Reports and dashboards are created to provide insights into the rental data, allowing for monitoring of key metrics such as the total number of bookings, revenue generated, and popular laptop models. Reports are designed in various formats—tabular, summary, and matrix—to show data from different perspectives. Dashboards visualize this information in real-time, with components like charts, gauges, and tables. For example, a dashboard might display total rental revenue by laptop model or monthly booking trends.

Screenshots:

Reports:

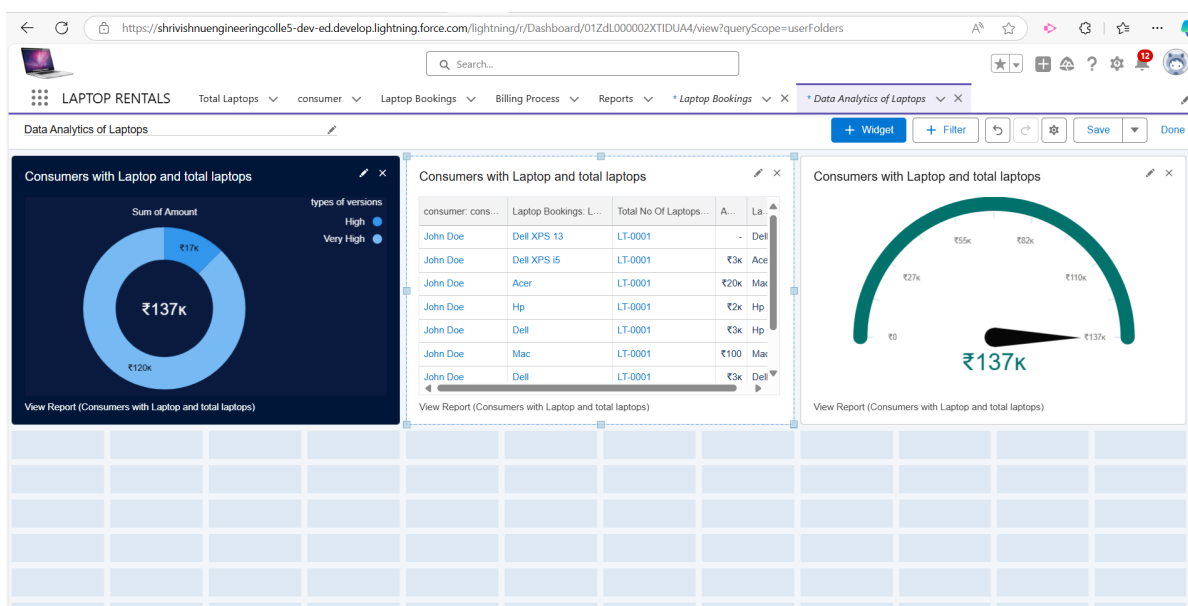


Report: consumer with Laptop Bookings and Total Laptops
Consumers with Laptop and total laptops

Total Records: 9
Total Amount: ₹1,37,000

types of versions	consumer: consumer_name	Laptop Bookings: Laptop Bookings	Total No Of Laptops: Total Laptops	Amount	Laptop names
Basic (1)	John Doe	Dell XPS 13	LT-0001	-	Dell
Subtotal				₹0	
High (6)	John Doe	Dell XPS i5	LT-0001	₹3,400	Acer
	John Doe	Hp	LT-0001	₹1,800	Hp
	John Doe	Dell	LT-0001	₹2,800	Hp
	John Doe	Dell	LT-0001	₹3,000	Dell
	John Doe	Dell	LT-0001	₹1,800	Dell
	John Doe	Dell	LT-0001	₹4,200	Hp
Subtotal				₹17,000	
Very High (2)	John Doe	Acer	LT-0001	₹20,000	Mac
	John Doe	Mac	LT-0001	₹1,00,000	Mac
Subtotal				₹1,20,000	
Total (9)				₹1,37,000	

Dashboard:

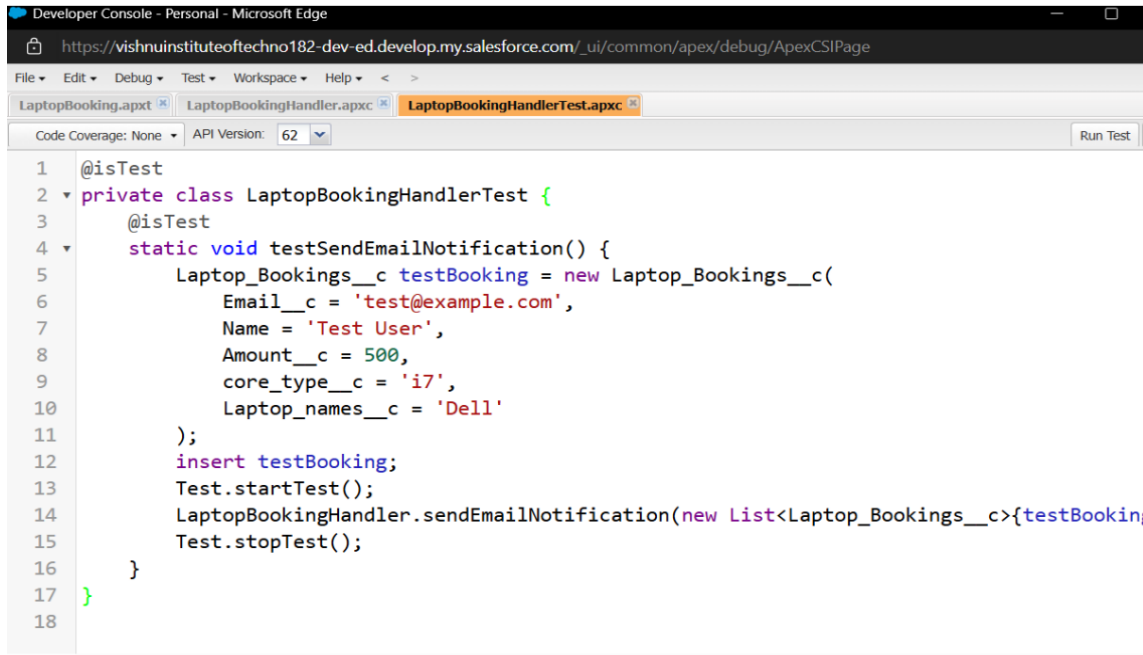


5. Testing and Validation Unit Testing

- **Approach:** Unit testing in Apex is performed using test classes annotated with

@isTest. Each class should cover various scenarios to ensure the functionality works correctly.

- **Coverage:** Aim for 100% code coverage, especially for critical business logic, ensuring that all methods and branches are tested.

The screenshot shows the Salesforce Developer Console interface. At the top, the browser address bar displays the URL: https://vishnuinstituteoftechno182-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. Below the browser, the console has a menu bar with 'File', 'Edit', 'Debug', 'Test', 'Workspace', and 'Help'. The workspace shows three tabs: 'LaptopBooking.apxt', 'LaptopBookingHandler.apxc', and 'LaptopBookingHandlerTest.apxc'. The 'LaptopBookingHandlerTest.apxc' tab is active, showing the following Apex code:

```
1 @isTest
2 private class LaptopBookingHandlerTest {
3     @isTest
4     static void testSendEmailNotification() {
5         Laptop_Bookings__c testBooking = new Laptop_Bookings__c(
6             Email__c = 'test@example.com',
7             Name = 'Test User',
8             Amount__c = 500,
9             core_type__c = 'i7',
10            Laptop_names__c = 'Dell'
11        );
12        insert testBooking;
13        Test.startTest();
14        LaptopBookingHandler.sendEmailNotification(new List<Laptop_Bookings__c>{testBooking});
15        Test.stopTest();
16    }
17 }
18
```

Above is LaptopBookingHandlerTest class

User Interface Testing

- **Approach:**
- Manual and functional testing of the UI components is crucial. This includes testing all navigation paths and user interactions in the Lightning app.
- **Examples:**
- Tested all record pages, forms, and custom UI components to ensure they function correctly.
- Verified that data entries trigger validation rules as expected.

6. Key Scenarios Addressed by Salesforce in the Implementation Project

Here are some key scenarios that your Salesforce implementation project addresses, particularly in the context of a Laptop Rental Management System:

1. Automated Booking Process

- **Scenario:** Users can quickly book laptops through a user-friendly interface without needing manual intervention.

- **Salesforce Solution:** Using custom objects for laptop bookings and record triggered flows, the system automates the booking process, ensuring that users receive immediate confirmations.

2. Email Notifications

- **Scenario:** Customers need timely updates regarding their laptop bookings and other relevant information.

- **Salesforce Solution:** Apex triggers are used to send automated email notifications upon successful bookings, keeping customers informed and engaged.

3. Dynamic Pricing Calculation

- **Scenario:** Different laptop models have different pricing structures based on specifications like processor type and rental duration.

- **Salesforce Solution:** Apex classes and triggers dynamically calculate the total rental amount based on model selection, cores, and duration. This ensures accurate billing and transparency for users.

4. Inventory Management

- **Scenario:** Managing available laptops and ensuring no double bookings occur is critical for operational efficiency.

- **Salesforce Solution:** Custom fields and validation rules track inventory levels, while workflows or flows ensure that bookings are only confirmed if laptops are available.

5. Validation of Data Integrity

- **Scenario:** Users must enter accurate data, such as email addresses and phone numbers, to ensure effective communication and record management.

- **Salesforce Solution:** Validation rules are implemented to check for proper data formats and required fields before records are saved, improving data quality.

7. Conclusion

The implementation of the Laptop Rental Management System using Salesforce has significantly improved operational efficiency and user experience. Key achievements include:

- **Automated Booking and Notifications:** Streamlined the booking process and implemented automated email notifications for users.

- **Dynamic Pricing and Accurate Billing:** Established a dynamic billing system that ensures accurate rental calculations based on laptop models and rental durations.

- **Effective Inventory Management:** Enhanced tracking of available laptops, preventing double bookings and maintaining data integrity.

- **Improved User Interface:** Developed a user-friendly Lightning app for seamless navigation and interaction.

Overall, this project has transformed the laptop rental management process, resulting in improved customer satisfaction and operational productivity, while providing valuable insights through robust reporting and analytics capabilities.