A CRM APPLICATION

For

Laptop Rentals

NAME: GUDUGUNTLA ASHWINI

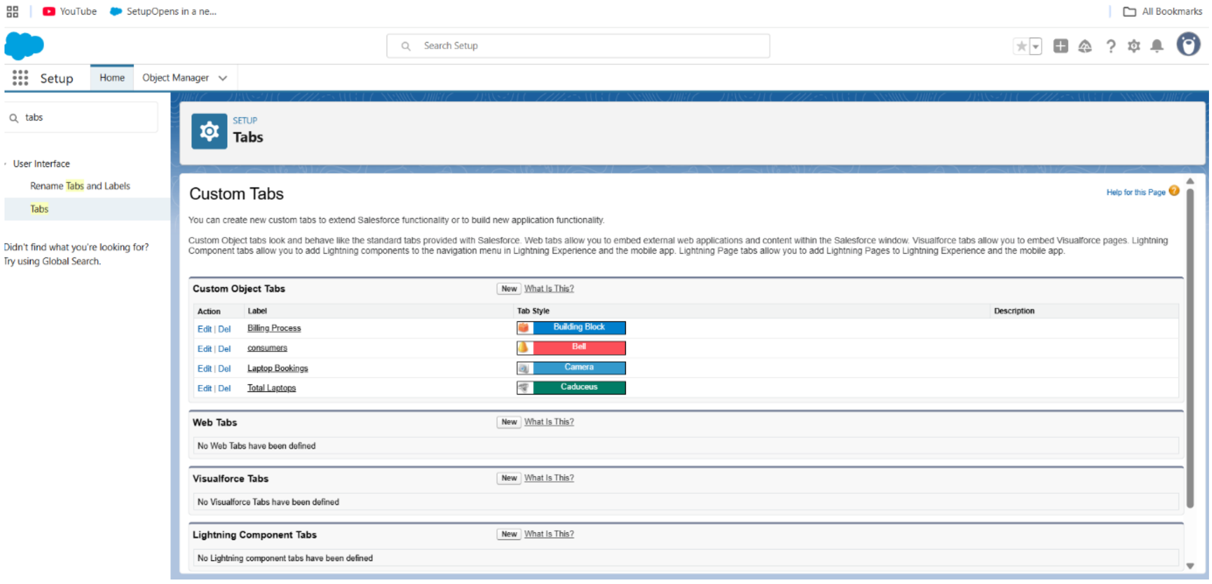
EMAIL :ashwiniguduguntla212@gmail.com

Phase 1: Problem Understanding & Industry Analysis

- **Requirement Gathering**
    - Manage laptop rentals for various brands and core types (e.g., I3, i5, i7).
    - Automate price updates based on rental periods using Salesforce Flows.
    - Send automated email notifications to customers with calculated prices.
    - Generate daily sales reports for business owners.
- **Stakeholder Analysis**
    - **Business Owners**: Require real-time insights into sales performance and customer behaviour for data-driven decision-making**.**
    - **Customers:** Need a seamless rental experience with timely notifications and transparent pricing**.**
    - **Administrators/Agents:** Need a streamlined system to manage inventory, bookings, and billing, reducing manual errors**.**
- **Business Process Mapping**
  The core business process was mapped as:
    1. **Inventory Management:** Adding laptops with specs (Core Type, RAM, Storage) and tracking availability.
    2. **Booking Process:** Customers book a laptop for a specific rental period, triggering an automated price calculation**.**
    3. **Notification**: An automated email is sent to the customer confirming the booking and price.
    4. **Billing & Reporting:** The billing process is initiated, and daily sales data is aggregated for reports.
- **Industry-specific Use Case Analysis**
    - Use cases focused on the equipment rental industry, emphasizing dynamic pricing models, inventory tracking, and customer communication loops.
- **AppExchange Exploration**
    - While the initial solution used custom development, AppExchange was explored for potential complementary solutions for advanced reporting (e.g., CRM Analytics) or mass email tools.

Phase 2: Org Setup & Configuration

- **User Setup & Licenses:** Standard Salesforce licenses were assigned based on user roles (Administrator, Standard User).
- **Profiles:** Custom profiles were created (e.g., "Laptop Rental Agent") with modified object and field-level permissions.

- **Roles:** A role hierarchy was established to mirror the organizational structure and control record visibility**.**
- **Permission Sets:** Used to grant additional permissions, such as API access, to specific users without modifying their base profile.
- **OWD (Organization-Wide Defaults)**: Set to Private for key custom objects to ensure strict data security, with access opened via sharing rules.
- **Sharing Rules:** Implemented to share record access based on criteria and roles within the hierarchy.
- **Salesforce Org**: Development was done in a Developer Edition org. A salesforce Org was used for testing before deployment**.**
- 

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

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	Building Block	
Edit | Del	consumers	Bell	
Edit | Del	Laptop Bookings	Camera	
Edit | Del	Total Laptops	Caduceus	

Web Tabs

No Web Tabs have been defined

Visualforce Tabs

No Visualforce Tabs have been defined

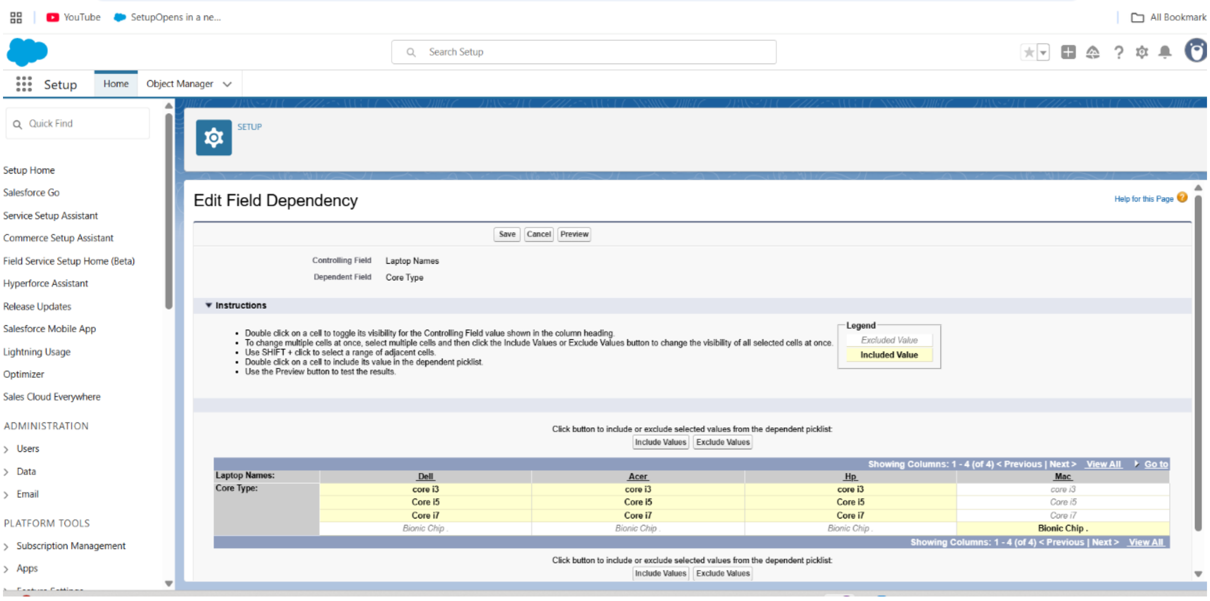
Lightning Component Tabs

No Lightning component tabs have been defined

Phase 3: Data Modelling & Relationships

- **Custom Objects: Three primary custom objects were created:**
  - **Laptop__c:** To store laptop inventory details.
  - **Laptop_Booking__c**: To manage customer rental transactions.
  - **Billing_Process__c**: To handle invoicing and payments**.**
- **Fields:**
  - **On Laptop__c:**
    - **Laptop_Name__c (Text)**
    - **Core_Type__c (Picklist: I3, i5, i7)**
    - **RAM__c (Number)**
    - **Storage__c (Number)**
    - **Laptops_Available__c (Formula: 50 - Total_no_of_laptops_r.Laptops_delivered__c) - Tracks real-time inventory.**

- **Relationships:**
  - A Lookup Relationship was created from Laptop_Booking__c to Laptop__c to link each booking to a specific laptop.
  - Field dependencies were configured on the Laptop_Booking__c object to dynamically show relevant options based on previous selections (e.g., available laptops based on brand).
- **Page Layouts & Compact Layouts:** Custom page layouts were designed for each object to optimize the user interface for data entry and review.



Edit Field Dependency

Controlling Field Laptop Names

Dependent Field Core Type

Instructions

- Double click on a cell to toggle its visibility for the Controlling Field value shown in the column heading.
- To change multiple cells at once, select multiple cells and then click the Include Values or Exclude Values button to change the visibility of all selected cells at once.
- Use SHIFT + click to select a range of adjacent cells.
- Double click on a cell to include its value in the dependent picklist.
- Use the Preview button to test the results.

Legend
Excluded Value
Included Value

Click button to include or exclude selected values from the dependent picklist:

Showing Columns: 1 - 4 (of 4) < Previous | Next > View All Go to

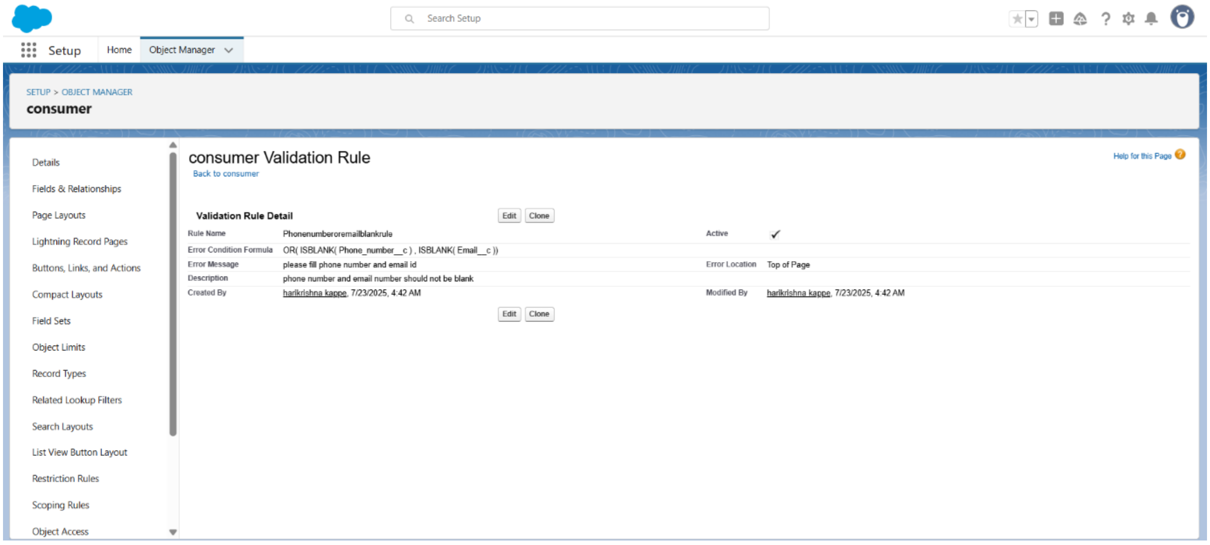
Laptop Names:	Dell	Acer	Hp	Mac
Core Type:	core i3	core i3	core i3	core i3
	Core i5	Core i5	Core i5	Core i5
	Core i7	Core i7	Core i7	Core i7
	Bionic Chip .	Bionic Chip .	Bionic Chip .	Bionic Chip .

Showing Columns: 1 - 4 (of 4) < Previous | Next > View All

Click button to include or exclude selected values from the dependent picklist:

Phase 4: Process Automation (Admin)

- **Validation Rules:**
  - **Rule Name:** Phone_number_or_email_blank_rule
  - **Object**: Laptop Booking (or related Consumer object)
  - **Error Condition:** OR(ISBLANK(Phone_number__c), ISBLANK(Email__c))
  - **Error Message: "**Phone number and email should not be blank.**"**



consumer Validation Rule

Validation Rule Detail

Rule Name	Phonenumberoremailblankrule	Active	✓
Error Condition Formula	OR(ISBLANK(Phone_number__c) , ISBLANK(Email__c))		
Error Message	please fill phone number and email id	Error Location	Top of Page
Description	phone number and email number should not be blank		
Created By	harikrishna kappe, 7/23/2025, 4:42 AM	Modified By	harikrishna kappe, 7/23/2025, 4:42 AM

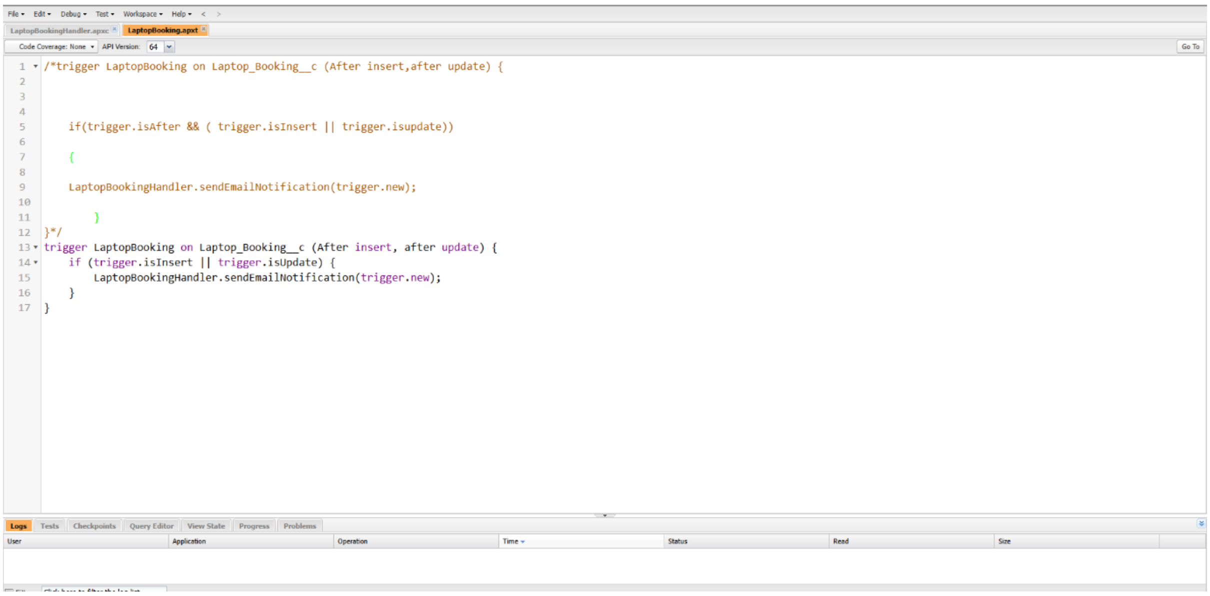
- **Flow Builder:** Automated, record-triggered flows were built to:
  - Calculate rental prices based on the selected laptop and rental period.
  - Update related records, such as decreasing available inventory upon a confirmed booking.

Phase 5: Apex Programming

- **Apex Trigger:**
  - **Name:** LaptopBookingTrigger
  - **Purpose:** To send a confirmation email to the customer after a booking is created or updated.

Apex:

```
trigger LaptopBookingTrigger on Laptop_Booking__c (after insert, after update) {
  if (Trigger.isAfter && (Trigger.isInsert || Trigger.isUpdate)) {
    LaptopBookingHandler.sendEmailNotification(Trigger.new);
  }
}
```

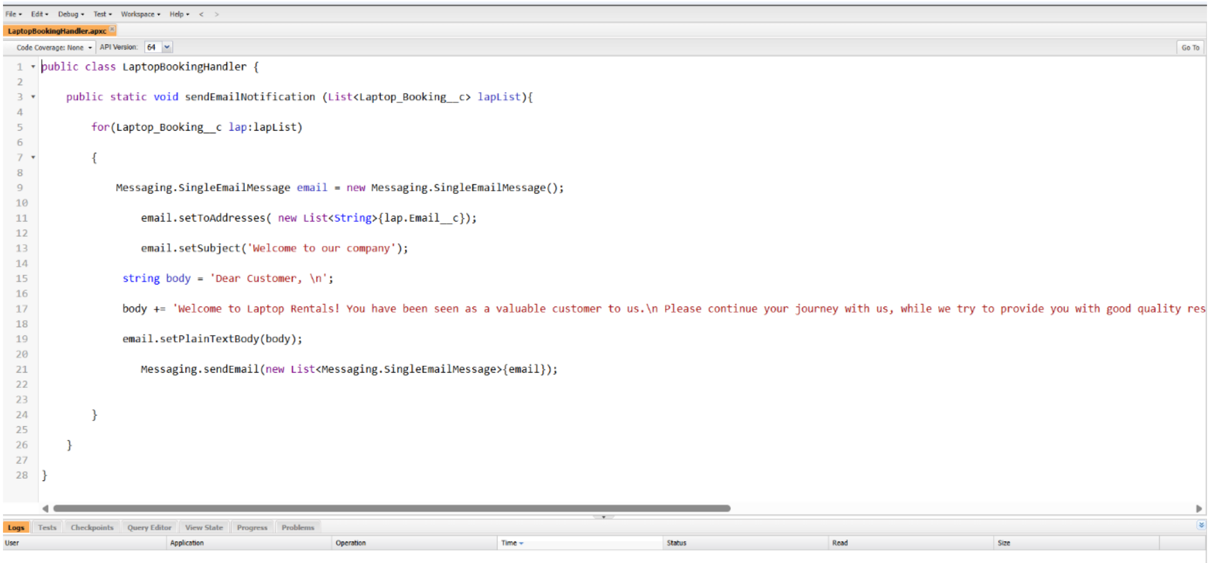


```
/*trigger LaptopBooking on Laptop_Booking__c (After insert,after update) {



    if(trigger.isAfter && ( trigger.isInsert || trigger.isupdate))

    {

    LaptopBookingHandler.sendEmailNotification(trigger.new);

        }
}*/
trigger LaptopBooking on Laptop_Booking__c (After insert, after update) {
    if (trigger.isInsert || trigger.isUpdate) {
        LaptopBookingHandler.sendEmailNotification(trigger.new);
    }
}
```

- **Apex Class (Handler):**
  - **Name:** LaptopBookingHandler
  - **Purpose:** Contains the business logic for constructing and sending the email. This follows best practices by keeping logic out of the trigger.

Handler class:

```
public class LaptopBookingHandler {
    public static void sendEmailNotification(List<Laptop_Booking__c> bookingList) {
        List<Messaging.SingleEmailMessage> emailList = new List<Messaging.SingleEmailMessage>();

        for (Laptop_Booking__c booking : bookingList) {
            if (booking.Email__c != null) {
                Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
                String[] toAddresses = new String[] {booking.Email__c};
                email.setToAddresses(toAddresses);
                email.setSubject('Welcome to Our Laptop Rental Service');

                String emailBody = 'Dear Customer, \n\n';
                emailBody += 'Welcome to Laptop Rentals! We value your business.\n';
                emailBody += 'Please continue your journey with us, while we try to provide you with good quality resources.\n\n';
                emailBody += 'Your Booking Details:\n';
        emailBody += '- Laptop Type: ' + booking.Laptop_Name__c + '\n';

        emailBody += '- Core Type: ' + booking.Core_Type__c + '\n';

        emailBody += '- Total Amount: ' + booking.Amount__c + '\n\n';

        emailBody += 'Thank you for choosing us!';


        email.setPlainTextBody(emailBody);

        emailList.add(email);

      }

    }


    if (!emailList.isEmpty()) {

      Messaging.sendEmail(emailList);

    }

  }

}
```

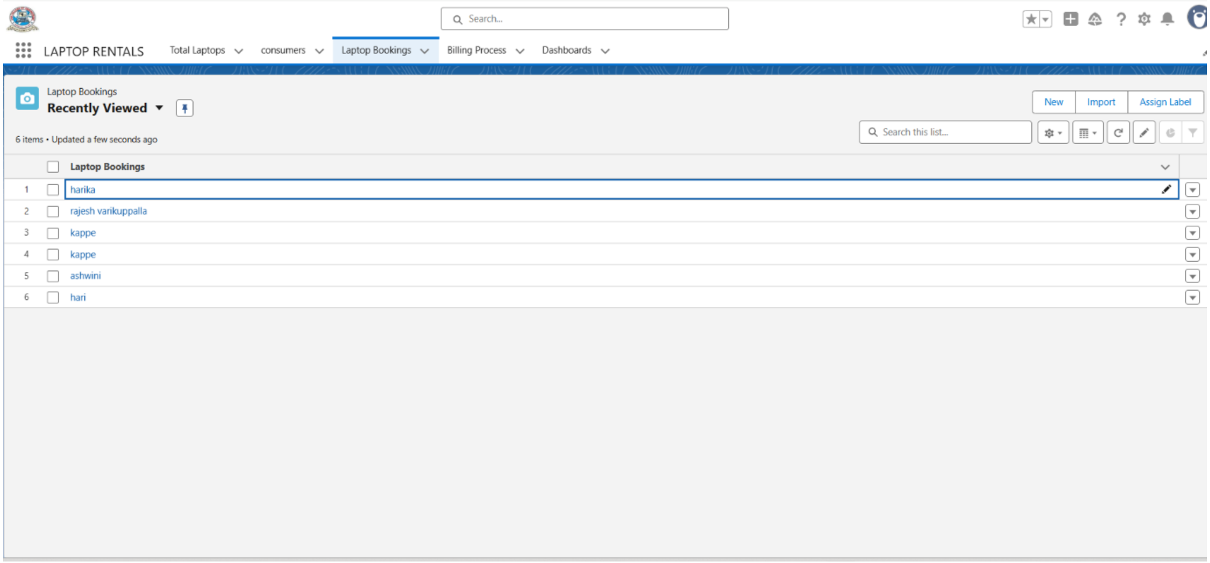


```
public class LaptopBookingHandler {

    public static void sendEmailNotification (List<Laptop_Booking__c> lapList){

        for(Laptop_Booking__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 Customer, \n';

              body += 'Welcome to Laptop Rentals! You have been seen as a valuable customer to us.\n Please continue your journey with us, while we try to provide you with good quality res

              email.setPlainTextBody(body);

                Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});


        }

    }

}
```

- **Test Classes:** Apex test classes were written to achieve over 75% code coverage, verifying that the trigger and handler function correctly and can be deployed to production.

Phase 6: User Interface Development

- **Lightning App Builder**: A custom Lightning app named "Laptop Rentals" was created to serve as the central workspace for users, containing all necessary tabs and components**.**
- **Record Pages**: The custom page layouts were applied to the Lightning Record Pages for a modern and efficient user experience.

- **Dynamic Forms:** Were used on the Laptop_Booking__c object to create a responsive and user-friendly form that guides the agent through the booking process.
- 

Laptop Bookings
Recently Viewed

6 items • Updated a few seconds ago

	Laptop Bookings
1	harika
2	rajesh varikuppalla
3	kappe
4	kappe
5	ashwini
6	hari

Phase 7: Integration & External Access

- **This phase was considered for future iterations. Potential integrations include:**
    - **Payment Gateways:** Using Named Credentials and REST callouts to integrate with services like Stripe or PayPal for payment processing.
    - **Email Services:** Configuring a more robust email service for larger notification volumes.

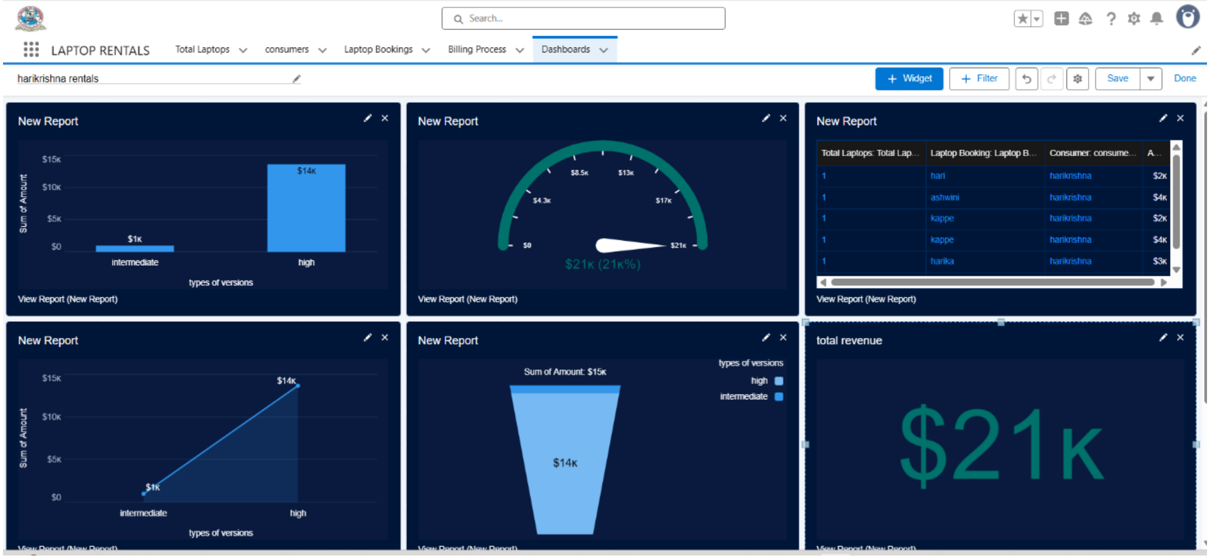
Phase 8: Data Management & Deployment

- **Data Import Wizard:** Used to migrate initial data (e.g., laptop inventory list) from spreadsheets into the Salesforce objects.
- **Change Sets:** The primary method for migrating customizations (objects, code, flows) from the sandbox environment to production.
- **Deployment Process: A rigorous deployment process was followed:**
    1. All Apex code was validated in the sandbox using the Developer Console.
    2. Test classes were run to ensure >75% coverage and all tests passed.
    3. Change Sets were created and validated.
    4. A final manual QA check was performed in production after deployment.

Phase 9: Reporting, Dashboards & Security Review

- **Reports:** Custom reports were built, including a "Daily Sales Report" for owners to track revenue and bookings.
- **Dashboards:** A manager dashboard was created to visualize key metrics from the reports, providing real-time business insights.
- **Security Review:**

- **Profiles & Permission Sets:** Reviewed to ensure users had the minimum access necessary (Principle of Least Privilege).
- **Field-Level Security**: Verified that sensitive fields (e.g., amount, cost) were hidden from inappropriate profiles.



New Report

Total Laptops: Total Lap...	Laptop Booking: Laptop B...	Consumer: consume...	A...
1	hari	harikrishna	$2K
1	ashwini	harikrishna	$4K
1	kappe	harikrishna	$2K
1	kappe	harikrishna	$4K
1	harika	harikrishna	$3K

View Report (New Report)

Phase 10: Final Presentation & Demo

- **Demo Walkthrough:** A live demonstration was conducted, showcasing:
  - The process of creating a new laptop booking.
  - The automatic price calculation.
  - The instant automated email notification received by the customer.
  - The daily sales report and dashboard for management.
- **Demo Link:** https://1drv.ms/v/c/c818c6e560802203/ES6Vfqn6RPtHuf98dCJx2ToBELwGoMX5n6XVwvNDUXAuDA?e=ecpfxe
- **Conclusion:**
  Our Laptop Rentals CRM project has successfully streamlined business processes, enhanced customer experience, and provided valuable insights for growth. The automated price updates, customer notifications, and daily sales reports have improved efficiency and decision-making. With proper documentation and maintenance, this CRM application will continue to support business growth and scalability.