Project title: BIKE RENTALS

1. Project Overview

The "Bike Rentals" is designed for businesses, such as bike rental shops or companies, that provide bike rental services. The objective of this project is to create a user-friendly, efficient system that allows consumers to rent bikes based on specific criteria such as the rental company, bike model, or mileage over a period of 1, 2, or 3 months. By leveraging the Salesforce platform, the solution enhances operational efficiency, ensures seamless customer experiences, and provides powerful data analysis tools for bike shop owners to track and manage their rental operations.

Key features include real-time reporting of bike rentals, easy data modification, automated email notifications for customers, and monthly dashboard reports delivered to shop owners.

2. Objectives

Business Goals:

- Streamline the bike rental process, making it easier for consumers to rent bikes and for shop owners to manage rental operations.
- Enhance customer engagement with timely email notifications and updates.
- Provide shop owners with detailed reports and analytics on their bike rental business, improving decision-making and operational efficiency.
- Increase overall revenue by improving the visibility and management of bike inventory.

Specific Outcomes:

- A fully functional CRM application for bike rental companies that includes an intuitive interface for rental management.
- Automated email notifications sent to consumers regarding their rental status.
- Monthly reports with dashboard analysis sent to bike shop owners, summarizing key metrics like revenue, rentals by period, and popular bike models.

- Simple, one-click data updates for rental companies to manage bike availability, pricing, and rental terms.

3. Salesforce Key Features and Concepts Utilized

Custom Objects: Created for managing bikes, rental transactions, and customers.

Process Automation:

- **Email Alerts:** Automated notifications to customers confirming their rental details.
- **Scheduled Reports and Dashboards:** Monthly dashboards sent via email to shop owners with metrics such as total rentals, mileage, and most popular bikes.

Lightning Pages and Components: To provide a smooth user experience for rental management.

Reports and Dashboards: Shop owners can view real-time data regarding rentals, revenue, and bike performance metrics.

Workflows and Approvals: For handling requests such as rental cancellations or extensions.

Security and Sharing Settings: Ensuring that data is secure and visible only to relevant users like shop owners and managers.

4. Detailed Steps to Solution Design

Step 1. Creating a developer org in salesforce.

Step 2. Account Acivation.

Step 3. Create four Object:

- Total Bikes
- Consumer
- Bike Bookings
- Billing Process

Creating Object steps: From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

Creating Total Bikes Object:

From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

1) Enter the label name>> Total Bikes

2) Plural label name>> Total Bikes

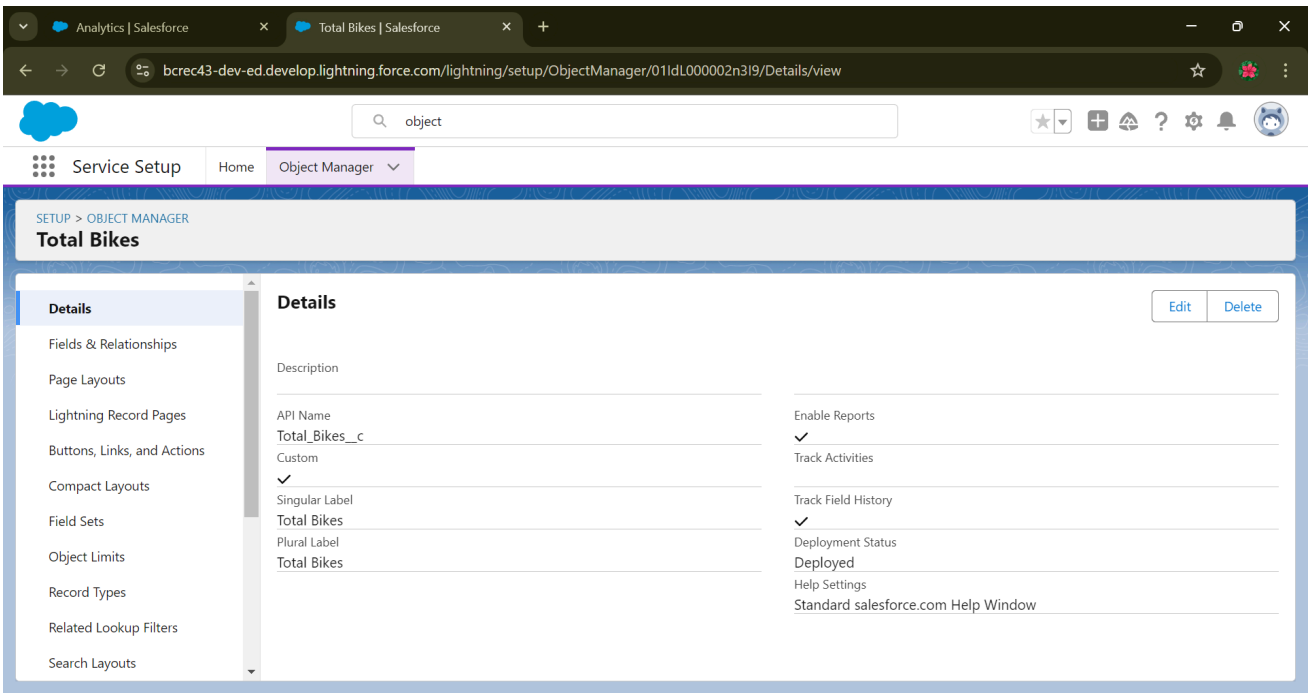
3) Enter Record Name Label and Format

Record Name >>Total Bikes

Data Type >> Text

Click on Allow reports,Allow search and Track Field History, Allow search >> Save.

Screenshot of **Total Bikes Object Created:**



Follow this step to create another Object.

- ➢ Consumer Object
- ➢ Bike Booking Object
- ➢ Billing Process Object

Step 4. Creating Custom tabs:

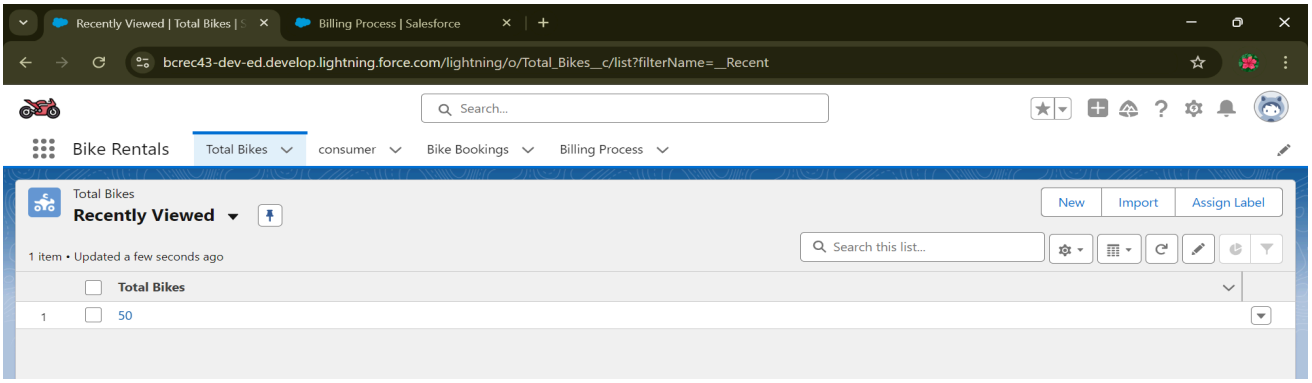
Creating custom tab for all objects: total bikes, consumer, bike booking and billing process.

Steps for creating custom tabs: Go to setup page >> Type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)

Creating Total Bikes Custom tab:

Go to setup page >> Type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab) Select Object(Total Bikes) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab. >> Make sure that the Append tab to users' existing personal customizations is checked. >> Click save.

Screenshot of Total Bikes Tab created:



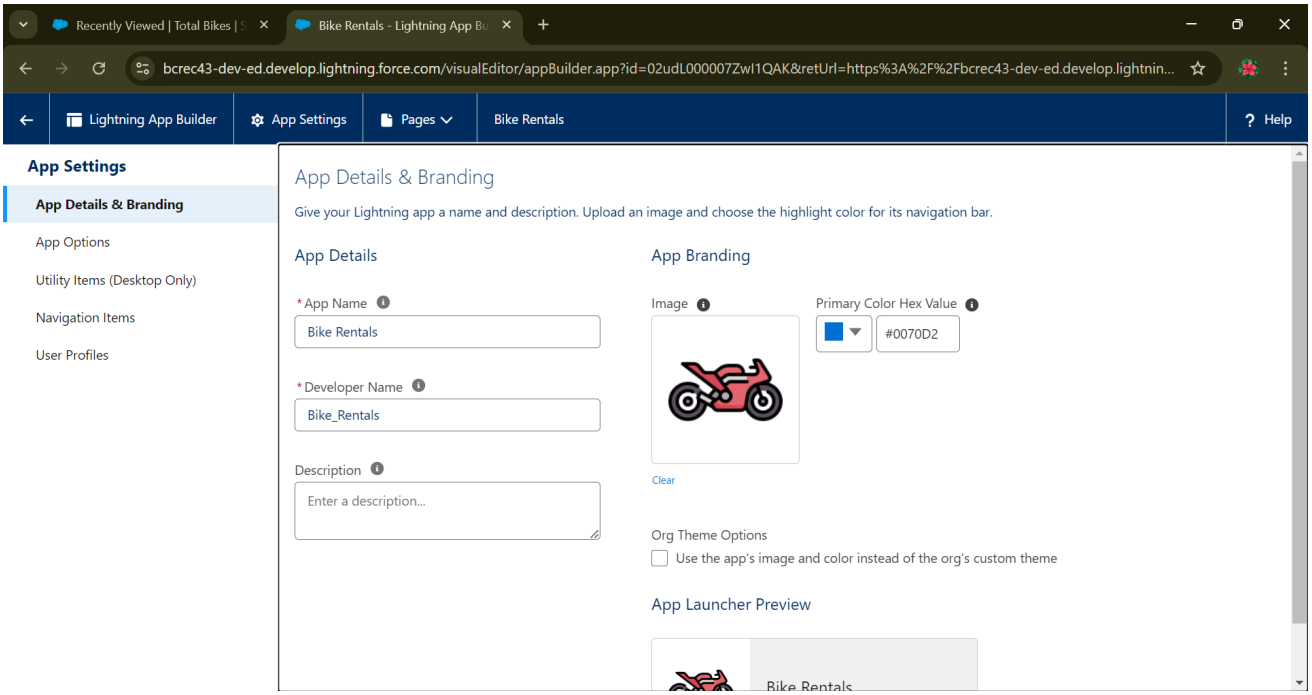
Follow this step to create another Custom tabs: Consumer, Bike Bookings and Billing Proces.

Step 5. Creating a lighting app:

Step for creating app: Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.

Fill all the necessary details like Name of app and upload image for app.

Screenshot of created lightning App:



Step 6. Creating Fields:

Steps for creating field: Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object >> Now click on “Fields & Relationships” >> New >> Select Data Type >> Click on next

Creating field and relationship in consumer object:
Phone, Email, Address and Consumer Status.

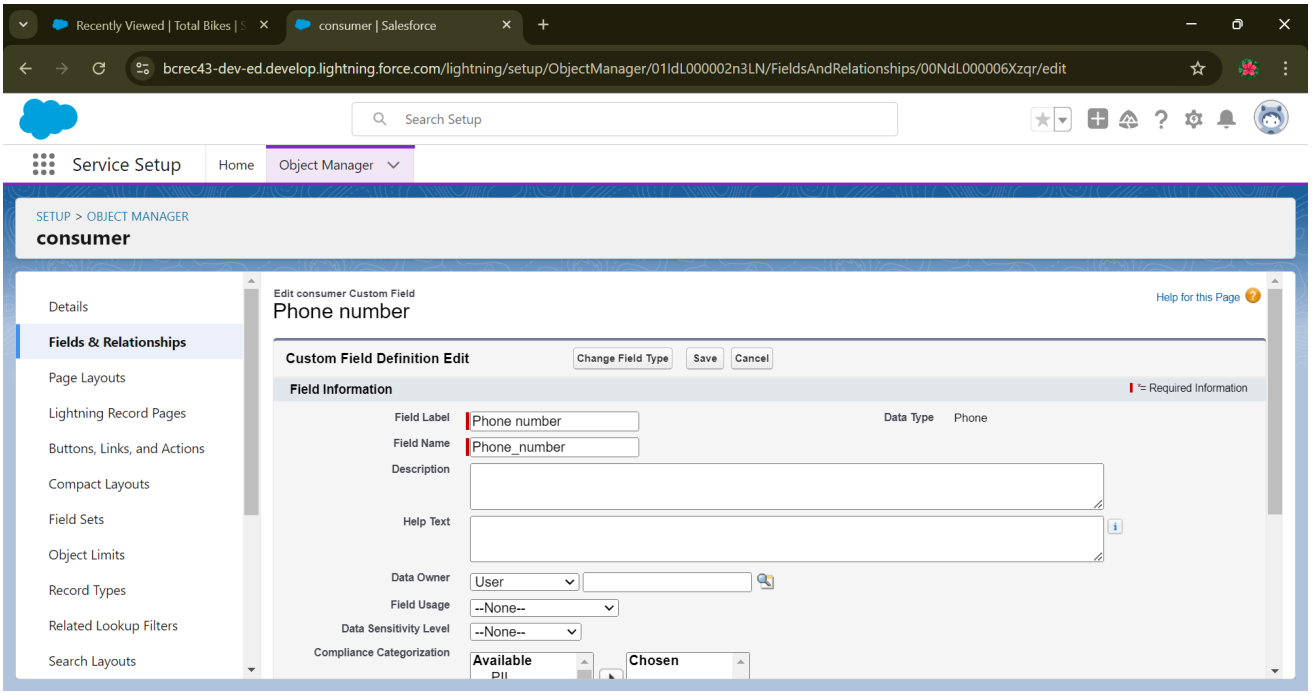
Creating Phone field and relationship:

Go to setup >> click on Object Manager >> type object name(consumer) in search bar >> click on the object.Now click on “Fields & Relationships” >> New >> Select Data Type as a “Phone” >> Click on next

Fill in the following:

- Field Label: Phone number
- Field Name : gets auto generated
- Click the required option checkbox.
- Click on Next >> Next >> Save and new.

Screenshot of Phone field and relationship.



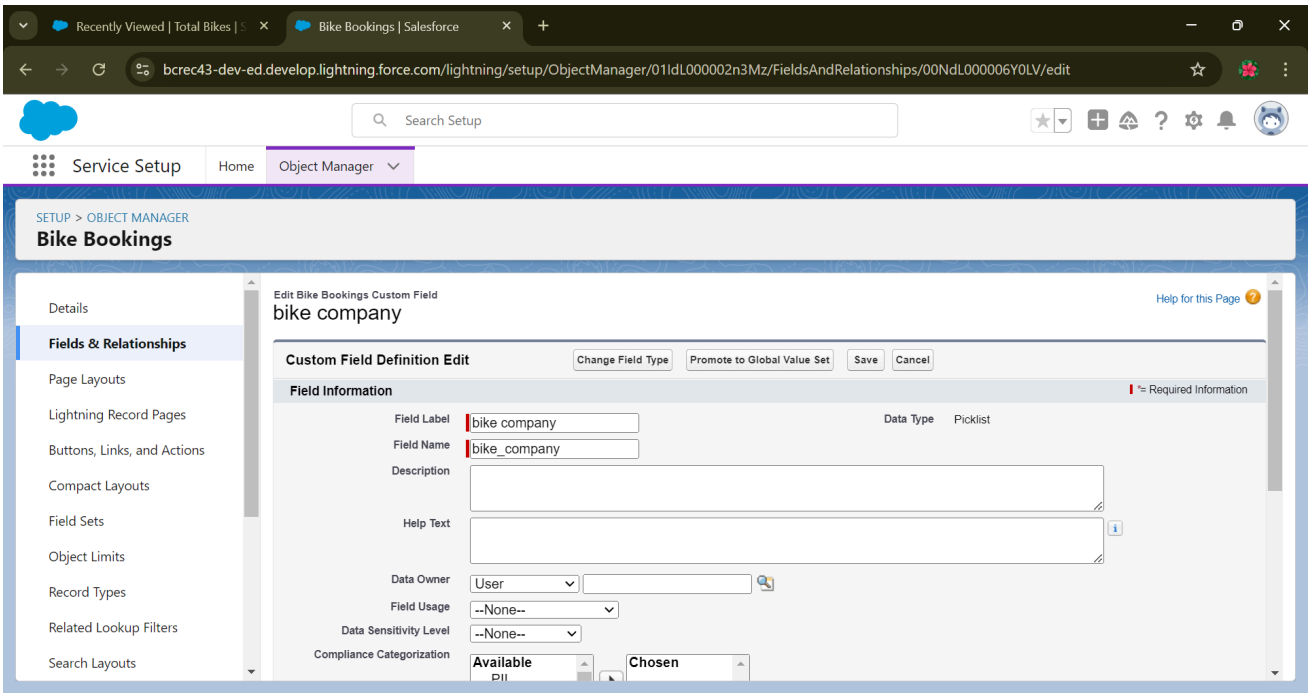
Same as all another field and relationships created in consumer object.

Creating field and relationship in Bike Booking object: Bike Company and Bike mileage.

Creating Bike Company field and relationship:

1. Go to setup >> click on Object Manager >> type object name(Bike Booking) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Picklist”
4. Picklist values are:-1.Tvs 2. Hero 3.Honda 4.Super Bike.
5. Select required
6. Click on Next >> Next >> Save

Screenshot of Bike Company field and relationship.

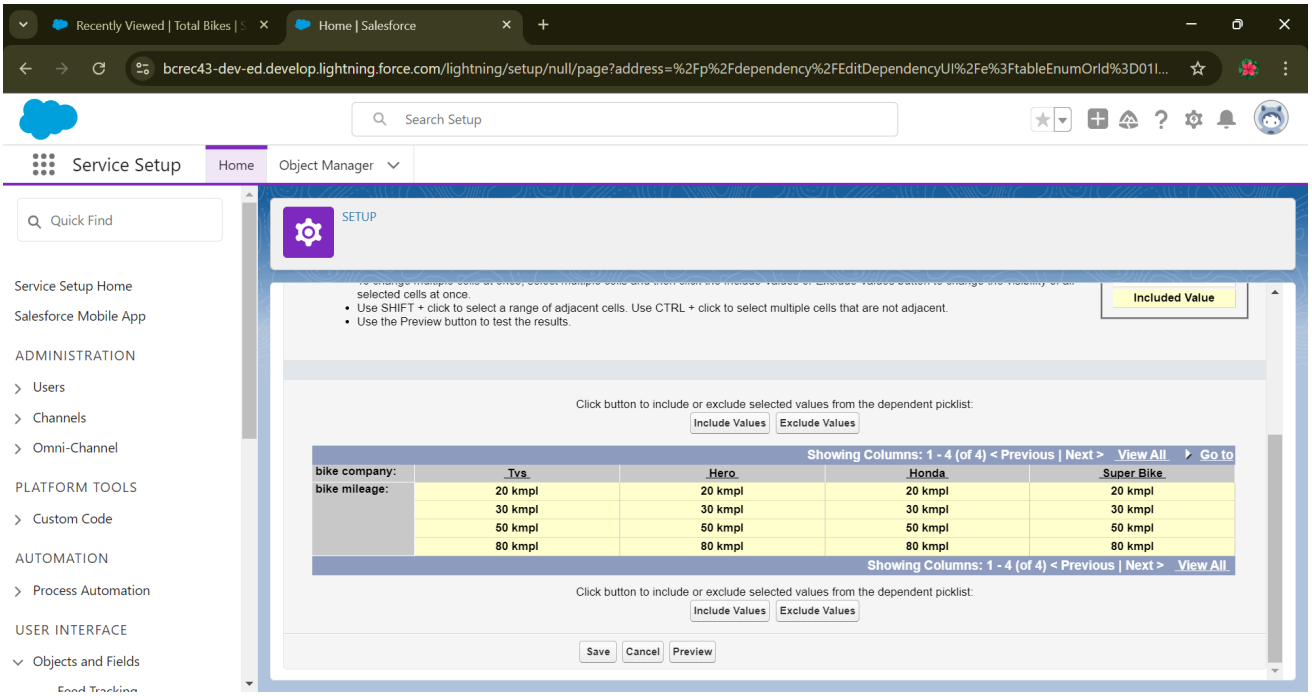


Same as another field and relationships created of Bike Booking object.

Creating field dependency of Bike Company and Bike mileage:

Step for creating field dependency: Go to setup >> click on Object Manager >> type object name(Bike Booking) in the search bar >> click on the object >> click fields and relationships >> click field dependency and next >> Click the include value.

Screenshot of field and dependency:



Creating Lookup Relationship of Bike Booking object: Name, Amount, Total no. of Bikes and Email.

Step for creating Lookup Relationship: Go to setup >> click on Object Manager >> type object name(Bike Booking) in the search bar >> click on the object >> Now click on “Fields & Relationships” >> New >> Select Data

Type as a “Lookup Relationship” >> Click on Next >> Click on the Related to drop down and Select the object and click on Next.

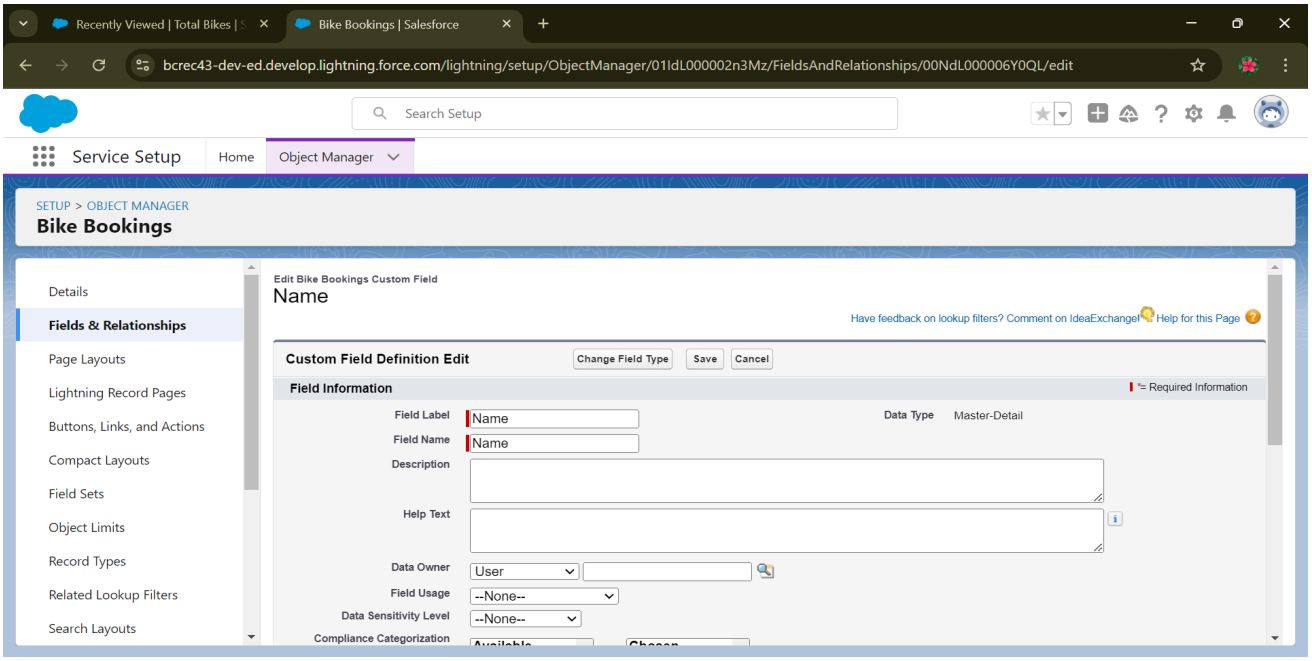
Creating Name Lookup Relationship:

Go to setup >> click on Object Manager >> type object name(Bike Booking) in the >> search bar >> click on the object >> Now click on “Fields & Relationships” >> New >> Select Data Type as a “Currency” >> Click on Next

Fill in the following:

- Field Label: Amount
- Length: (18,0)
- Field Name :It’s gets auto generated
- Click on Next >> Next >> Save and new

Screenshot of Name Lookup Relationship:



Same as Email Lookup Relationship created.

Creating Total no. of Bikes Lookup Relationship:

Go to setup >> click on Object Manager >> type object name(Bike Booking) in the search bar >> click on the object >> Now click on “Fields & Relationships” >> New >> Select Data Type as a “Lookup Relationship” >> Click on Next >> Click on the Related to drop down and Select the “Total Laptops” object and click on Next

Fill in the following:

- Change the Field Label: Total No Of Laptops
- Field Name :It’s gets auto generated
- Click on Next >> Next >> Save and new

Creating a Rollup Summary Field in “Total Bikes Object”:

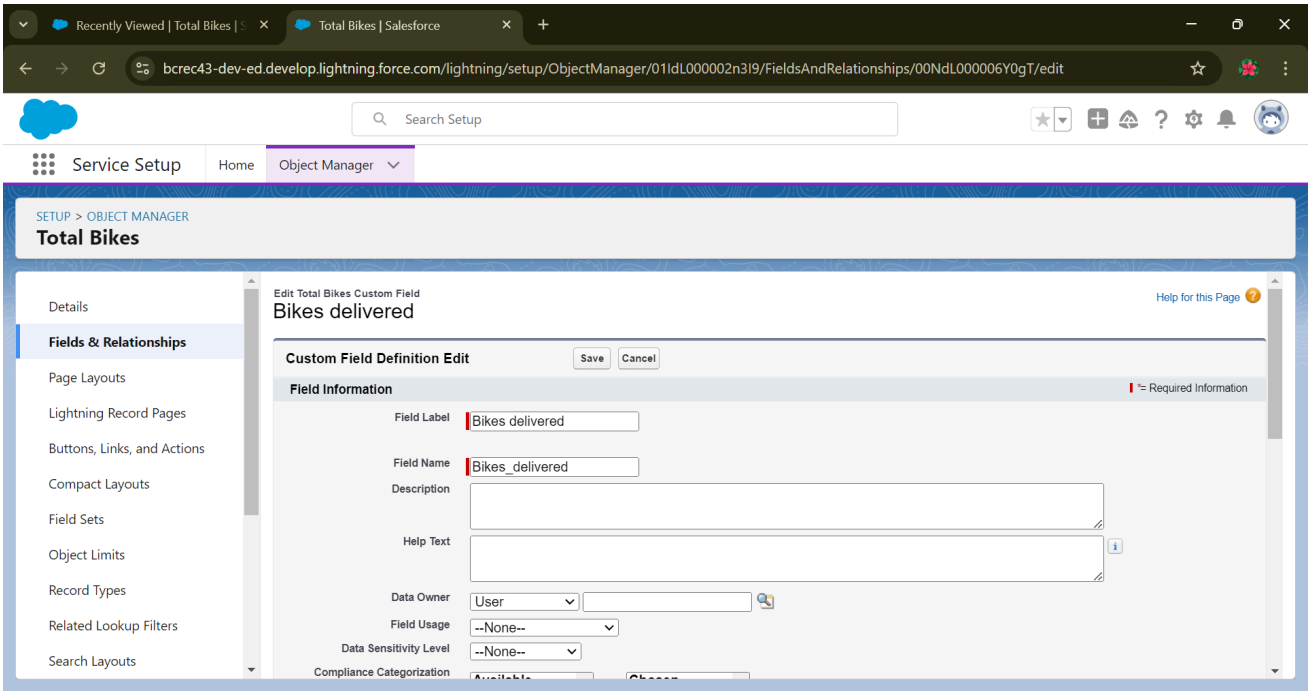
Go to setup >> click on Object Manager >> type object name(Total Bikes) in the search bar >> click on the object >> Now click on “Fields & Relationships” >> New >> Select Data type as a “Roll-up Summary” and Click on Next

Fill in the following:

- Field Label: Bikes delivered
- Field Name :It’s gets auto generated

Click on Next >> Select the Bike Bookings in the Summarized Object >> Select the count Radio button in the select Roll-up Type.

Screenshot of bikes delivered rollup in total bikes:



To creating fields in an object:

Go to setup >> click on Object Manager >> type object name(Bike Booking) in the search bar >> click on the object >> Now click on “Fields & Relationships” >> New >> Select Data type as a “Formula” and Click on Next

Fill in the following:

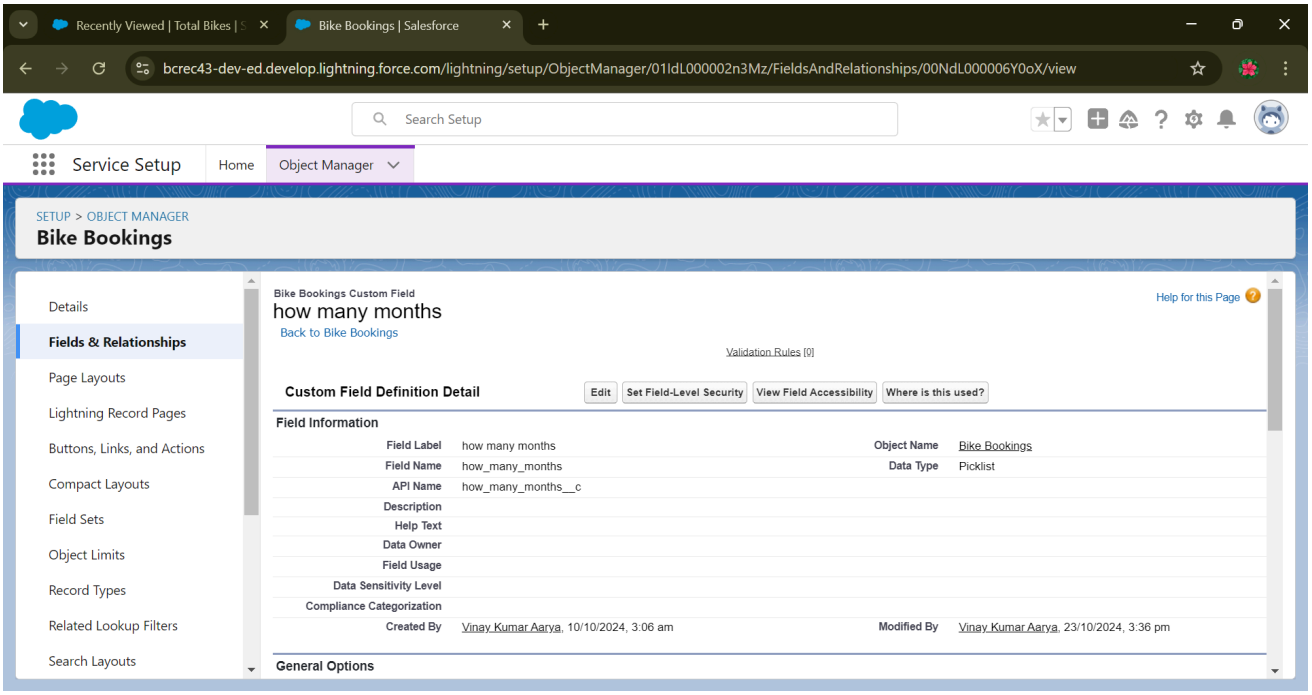
- Field Label: Bike Available
- Field Name : It’s gets auto generated
- Select the Formula Return Type as “Number”
- Select the Decimal places as “0” and Click on Next
- Click on the Advanced Formula and Enter the value in formula box “ 50 - ” and Click on insert field than you will find a pop window under the Bike Booking select the Total No Of Bikes in the second Column and select the bikes delivered in the third column and click on insert “ 50 - Total_no_of_bikes__r.Bikes_delivered__c ” and Check Syntax.

Click on Next >> Next >> Save.

Creating fields in an object:

Go to setup >> click on Object Manager >> type object name(Bike Booking) in the 2.search bar >> click on the object >> Now click on “Fields & Relationships” >> New >> Select Data Type as a “picklist” >> Field label: how many months >> Picklist values are 1,2,3 >> Click and save it.

Screenshot of Months selecting for rent created:



Creating of fields & Relationship for Billing Process Object:

Creating fields & relationship to an object:

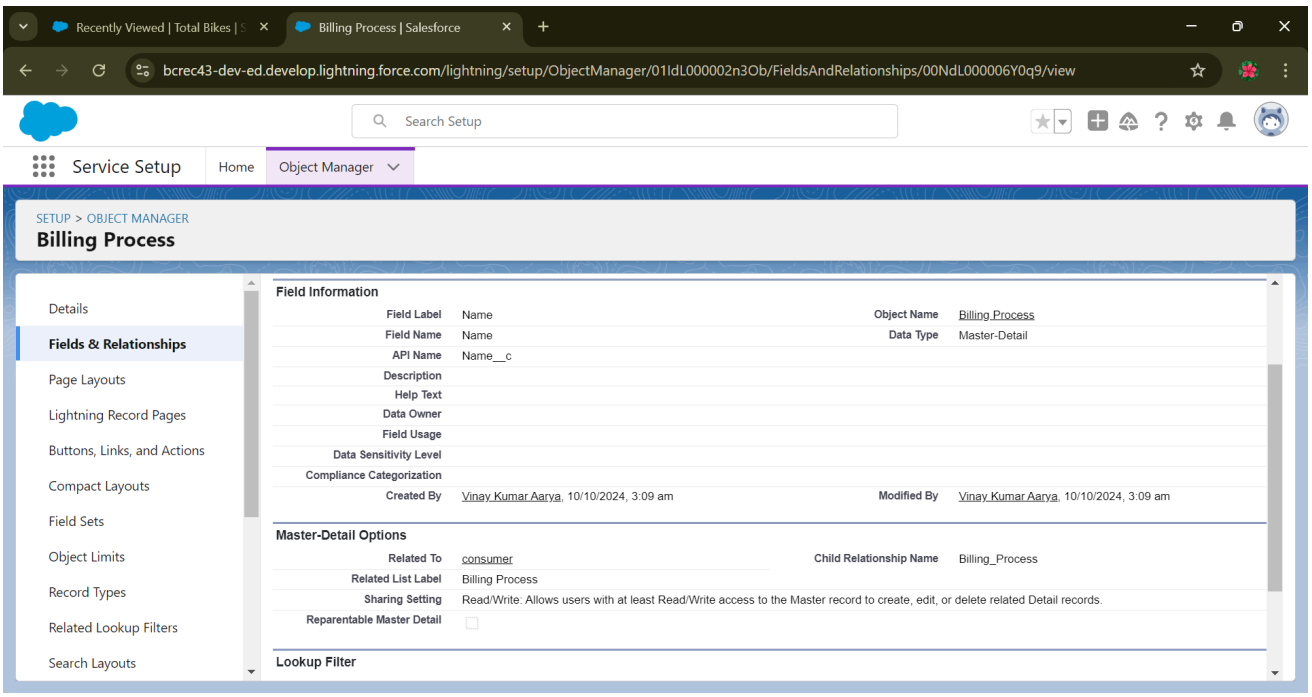
1. Go to setup >> click on Object Manager >> type object name(Billing Process) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New

3. Select Data Type as a “Master-detail Relationship”
4. Click on Next
5. Click on the Related to drop down and Select the consumer object and click on Next

   Fill the Above as following:

   - Change the Field Label: Name
   - Field Name :It’s gets auto generated
   - Click on Next >> Next >> Save.

Screenshot of above field & relationship:



Creating another fields & relationship to object:

1. Go to setup >> click on Object Manager >> type object name(Billing Process) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Lookup Relationship”
4. Click on Next

5. Click on the Related to drop down and Select the Laptop Booking object and click on Next
   Fill the Above as following:
   - Change the Field Label: Laptop Booking
   - Field Name :It's gets auto generated
   - Click on Next >> Next >>Save and new.

Creating another fields & relationships for billing process object :

1. Go to setup >> click on Object Manager >> type object name(Billing Process) in the search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New
3. Select Data Type as a "Picklist"
4. Fill the Above as following:
   - Field Label: Payment Mode
   - Value >> Select enter values with each value separated by a new line
     1. Cash
     2. Check
     3. Credit card
     4. Debit card
     5. UPI
     6. Phonepe
     7. Gpay
     8. Paytm
   - Select required
   - Click on Next >> Next >> Save and new.

Create a cross object formula field in billing process object:

1. Go to setup >> click on Object Manager >> type object name(Billing Process) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Formula”
4. Click on Next
5. Enter the Field label: Amount, the Field name gets auto generated and click on Next.(Formula return type Number).
6. In the Advanced Formula Click on the Insert field in the popup Screen Select the Billing Process and in the second drop down select the Bike Booking and in the three drop down select the Amount field and click on Insert
7. “ Bike_Booking__r.Amount__c ”.
8. Click on the Check syntax: No syntax errors in merge fields
9. Click on Next >> Next >> Save.

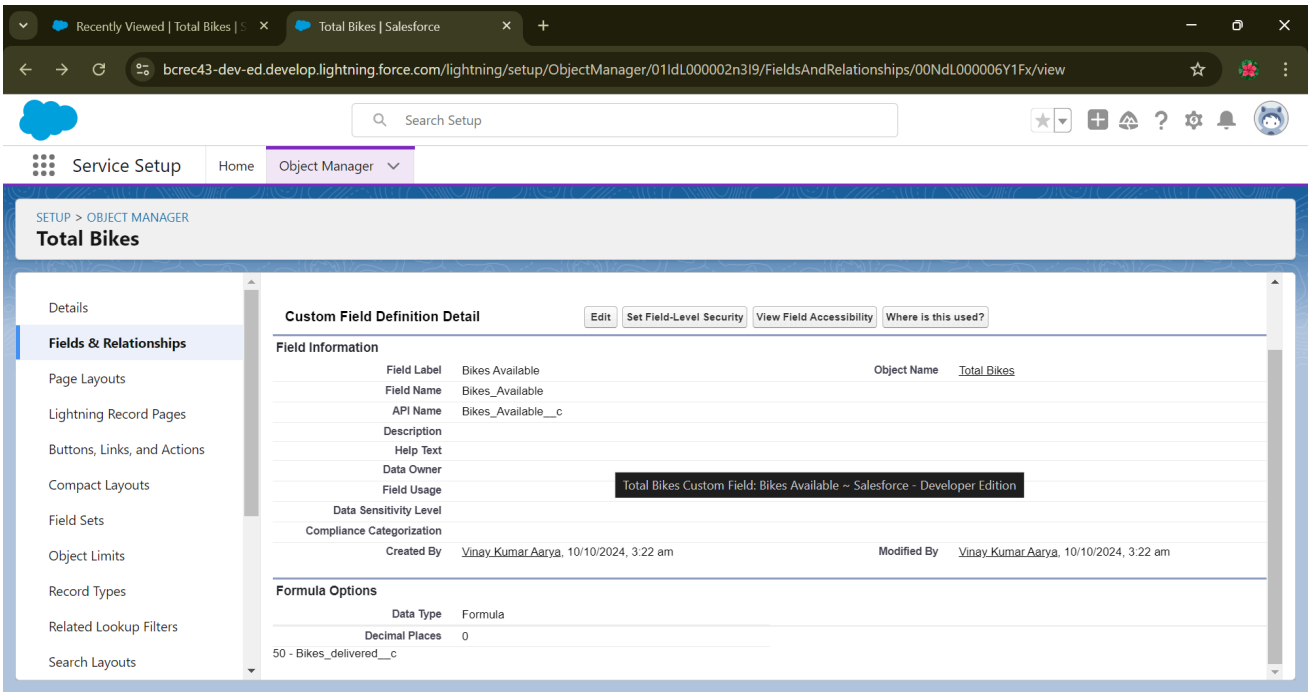
Creating the field in Total Laptops Object:

Create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Total Bikes) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Formula” and Click on Next
4. Fill the Above as following:
5. Field Label: Bikes Available

6. Field Name : It’s gets auto generated
7. Select the Formula Return Type as “Number”
8. Select the Decimal places as “0” and Click on Next
9. Click on the Advanced
10. Formula “ 50 - Bikes_delivered__c ” and Check Syntax
11. Click on Next >>Next >>Save and new.

Screenshot of above created field: Bikes Available



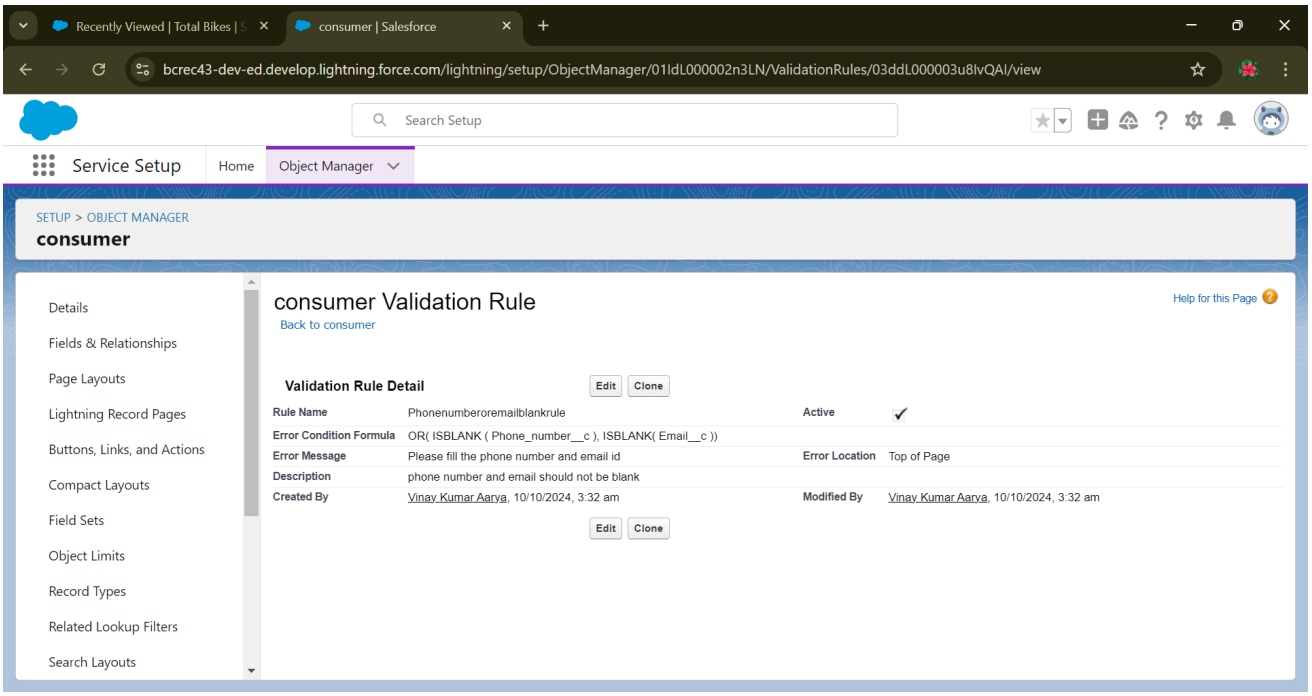
Step 7. Validation Rule

Creating validation rule for phone number field in consumer object:

1. Go to the setup page >> click on object manager >> From drop down click edit for consumer object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “Phonenumberoremailblankrule ”.

4. Enter the description as “phone number and email number should not be blank”.
5. Enter the formula as “OR(ISBLANK(phone_number__c) , ISBLANK(email__c))” and check the syntax.
6. Save the validation rule.

Screenshot of validation rule of Phone no. & email:



Step 8. Profile

Creating two profile: Owner and Agent Profile.

Create Owner Profile:

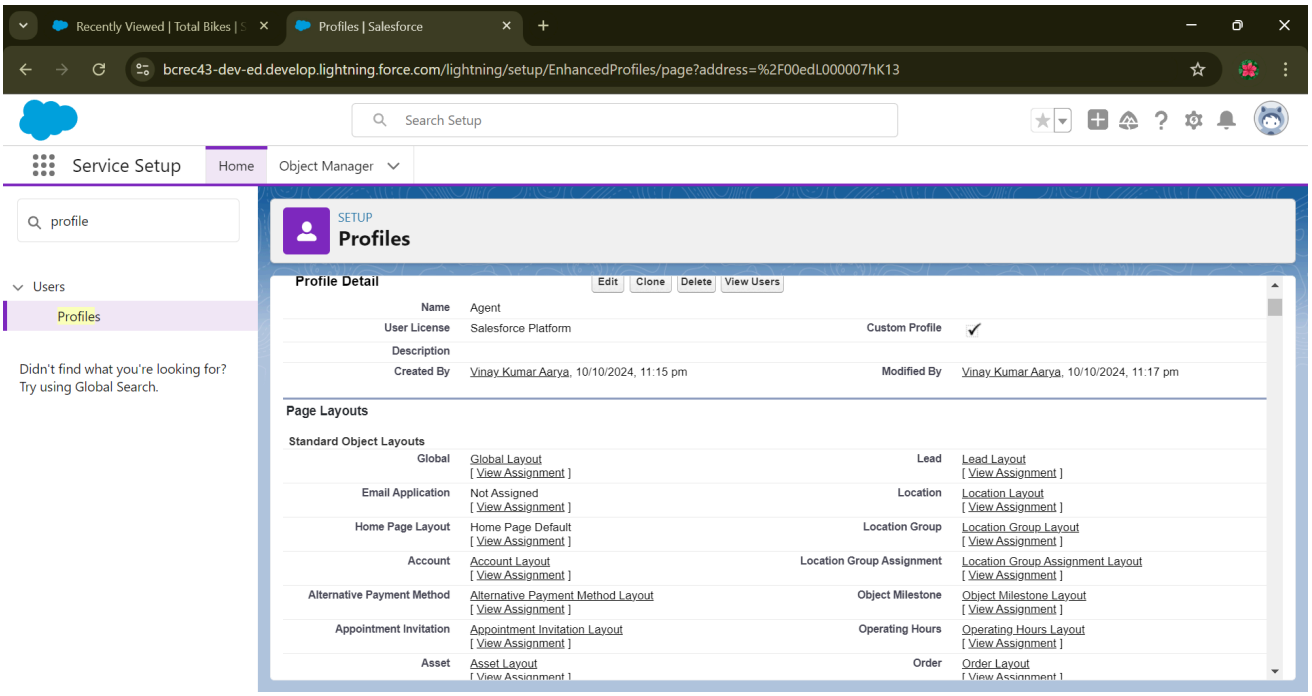
1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (owner) >> Save.
2. Scroll down to Custom Object Permissions and Give access permissions for Total Bikes, consumers , Bike Booking and Billing Process objects.

3. Give Access and Save it.

Create Agent Profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard Platform User) >> enter profile name (Agent) >>Save.
2. While still on the profile page, then click Edit.
3. Scroll down to Custom Object Permissions and Give access permissions for Total Bikes, consumer , Bike Bookings and Billing Process objects.
4. Give access and save it.

Screenshot of Agent profile created:



Step 9. Creating Role:

Creating Owner Role:

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click on Expand All and click on add role under whom this role works.
3. Give Label as “owner” and Role name gets auto populated.
4. Then click on Save.
5. Click and save it.

Creating Agent roles

Creating another two roles under manager

1. Go to quick find - Search for Roles - click on set up roles.
2. Click plus on CEO role, and click add role under owner.
3. Give Label as “Agent” and Role name gets auto populated.
4. Then click on Save.

Step 10. Create Flows

Create a flow on Tvs, Hero, Honda and Super Bike:

1. Go to setup >>type Flow in quick find box >> Click on the Flow and Select the New Flow.

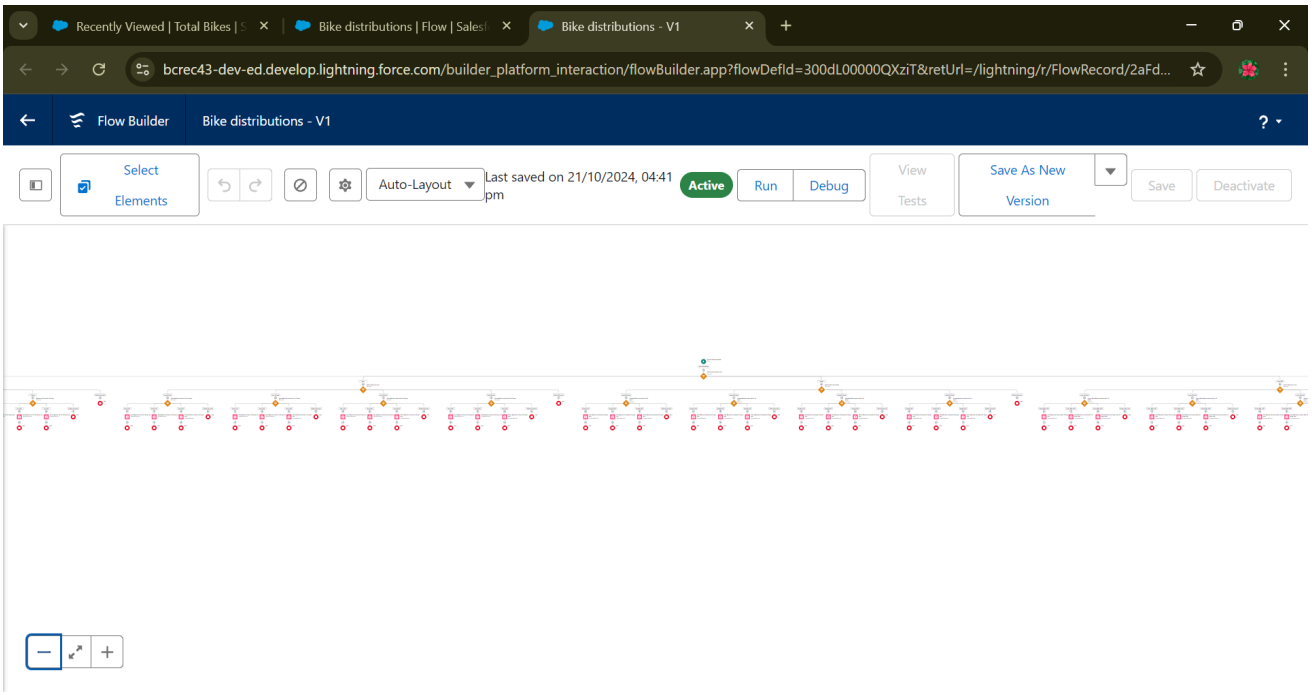
2. Select the Record-triggered flow and Click on Create.
3. Select the Object as a Bike Booking in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimize the flow for: “Actions and Related Records” and Click on Done.
6. Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Decision Element”.
7. Enter the Details Label: Field should be Update, API name: Gets Automatically Generated.
8. Enter the Outcome Details Label: Tvs , Outcome API name: Gets Automatically Generated.
    - Resource: Select Record.Bike booking__c.
    - Operator: Select Equals.
    - Value: Select Tvs
    - Add the same outcome order to Hero, Honda and Super Bike.
    - Click done.
    - Go to flow page
9. Beside dell there is a symbol ‘+’ click on that.
10. Again select decision
11. Enter the Details Label: Field should Update(any one u want), API name: Gets
12. Automatically Generated.

13. select the Outcome Details Label: Tvs 20 kmpl , Outcome API name: Gets Automatically Generated.
    - Resource: Select Record.mileage type.
    - Operator: Select Equals.
    - Value: Select Tvs 20 kmpl.
    - Then again click the symbol ‘+’ outcome details
14. select the Outcome ‘+’ Details Label: Tvs 30 kmpl , Outcome API name: Gets Automatically Generated.
    - Resource: Select Record.mileage type.
    - Operator: Select Equals.
    - Value: Select Tvs 30.
    - Then again click the symbol’+’ outcome details
15. Enter the Outcome Details Label: Tvs 50 kmpl , Outcome API name: Gets Automatically Generated.
    - Resource: Select Record.mileage type.
    - Operator: Select Equals.
    - Value: Select Tvs 50. Same as create Tvs 80 kmpl.
16. Click done.
17. So go to the flow page select ‘+’ after Tvs 20 then again select the decision.
18. Enter the Details Label: months selected , API name: Gets Automatically Generated.

19. Enter the Outcome Details Label: Tvs 1(20) , Outcome API name: Gets Automatically Generated.
    - Resource: Select Record.how many months.
    - Operator: Select Equals.
    - Value: 1.
20. Enter the Outcome Details Label: Tvs 2(20) , Outcome API name: Gets Automatically Generated.
    - Resource: Select Record.how many months.
    - Operator: Select Equals.
    - Value: Select 2.
21. Click '+' outcome details.
22. Enter the Outcome Details Label: Tvs 3(20) , Outcome API name: Gets Automatically Generated.
    - Resource: Select Record.how many months.
    - Operator: Select Equals.
    - Value: Select 3.
23. Follow the above picture you will understand.
24. After Tvs 1(20) there is '+' symbol like Tvs 2(20), TVS 3(20).
25. Click on '+' then select update records
30. Enter the Details Label: one month of Tvs 20 rate , API name: Gets Automatically Generated.
31. Field:- Amount__c , value:- for Tvs 1(20) - 5000, Tvs 2(20) - 8000, Tvs 3(20) - 12000. Follow for all these finally
32. Click done.

33. Follow the above Step and create another: Tvs 30 kmpl, Tvs 50 kmpl and Tvs 80 kmpl.
34. After that same step to done on Hero, Honda, Super Bike.
35. Save the flows: Flow label - Bike distribution.

Screenshot of all flows:



Step 11. Create Apex

Create a class APEX:

1. Navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on the File menu in the toolbar, and click on new-Class.
4. Enter the Class name BikeBookingHandler.

API Name:- Bike_Bookings__c(as per your org go to bike booking object and copy from that).
3.mileage__c (as per your org go to bike booking object and copy from that).
4.bike_company__c.(as per your org go to bike booking object and copy from that).

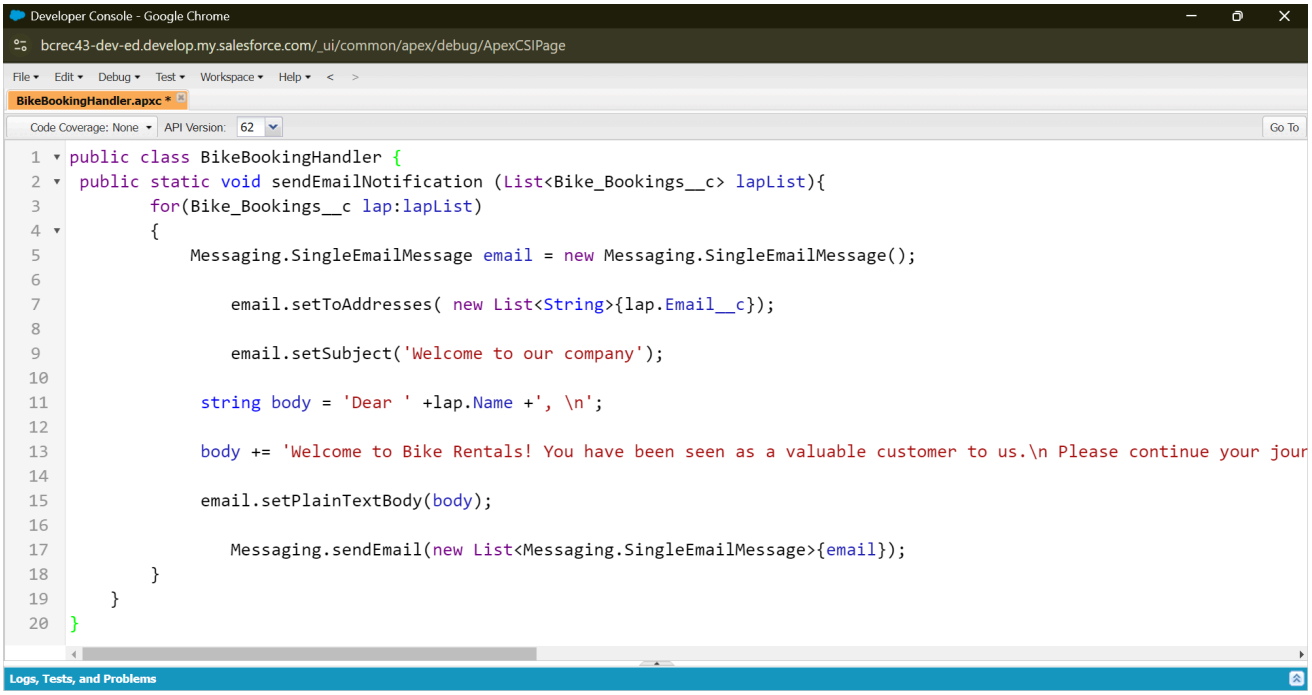
Code:

```
public class BikeBookingHandler {
    public static void sendEmailNotification
(List<Bike_Bookings__c> lapList){
        for(Bike_Bookings__c lap:lapList)
        {
            Messaging.SingleEmailMessage email = new
Messaging.SingleEmailMessage();
                email.setToAddresses( new
List<String>{lap.Email__c});
                email.setSubject('Welcome to our company');
              string body = 'Dear ' +lap.Name +', \n';
              body += 'Welcome to Bike 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 Bike Amount = ' + lap.Amount__c + ' \n
Company type = '+lap.company_name__c +' \n Bike Mileage
type = '+lap.Bike_mileage__c;
              email.setPlainTextBody(body);
```

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

        }
    }
}
```

Screenshot of created class apex:



Create a Trigger APEX:

1. Navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on the File menu in the toolbar, and click on new-Trigger.
4. Enter the trigger name: BikeBooking.

API names

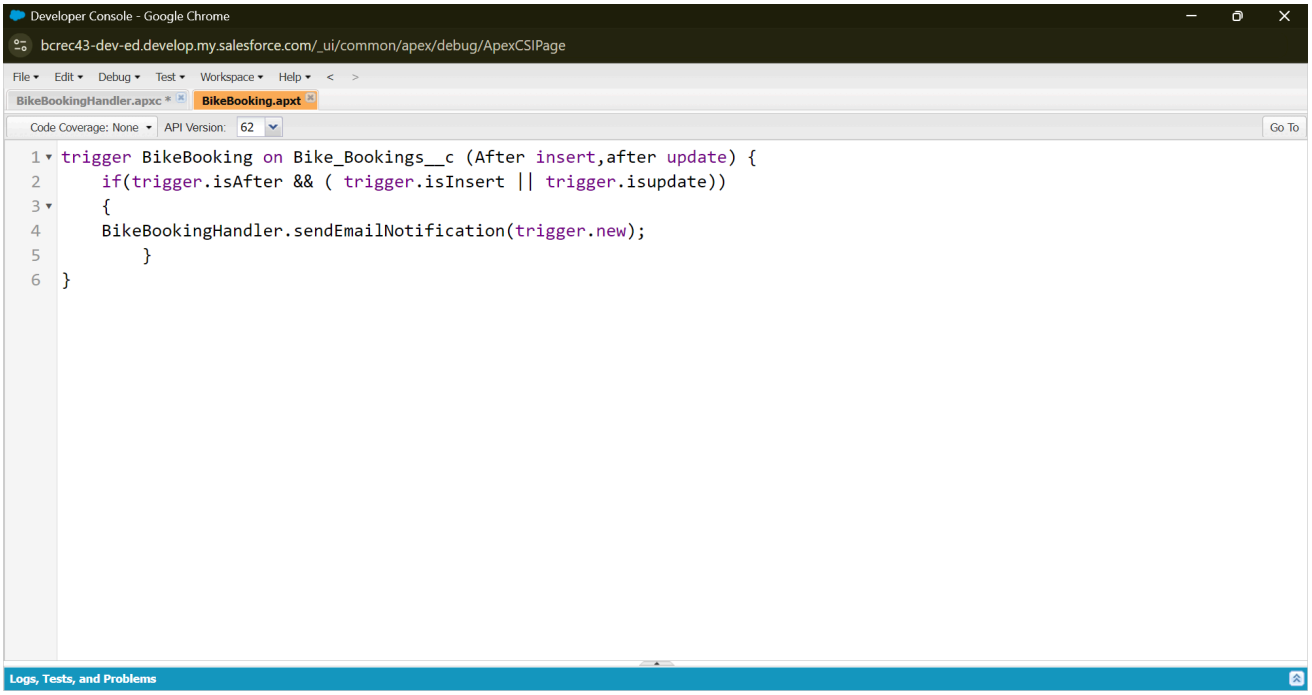
1.BikeBooking - trigger name

2.Bike_Bookings__c -as per your org(go to Bike bookings object and copy from that object api name).

Code:

```
trigger BikeBooking on Bike_Bookings__c (After insert,after update) {
    if(trigger.isAfter && ( trigger.isInsert || trigger.isupdate))
   {
   BikeBookingHandler.sendEmailNotification(trigger.new);
       }
}
```

Screenshot of created trigger apex:

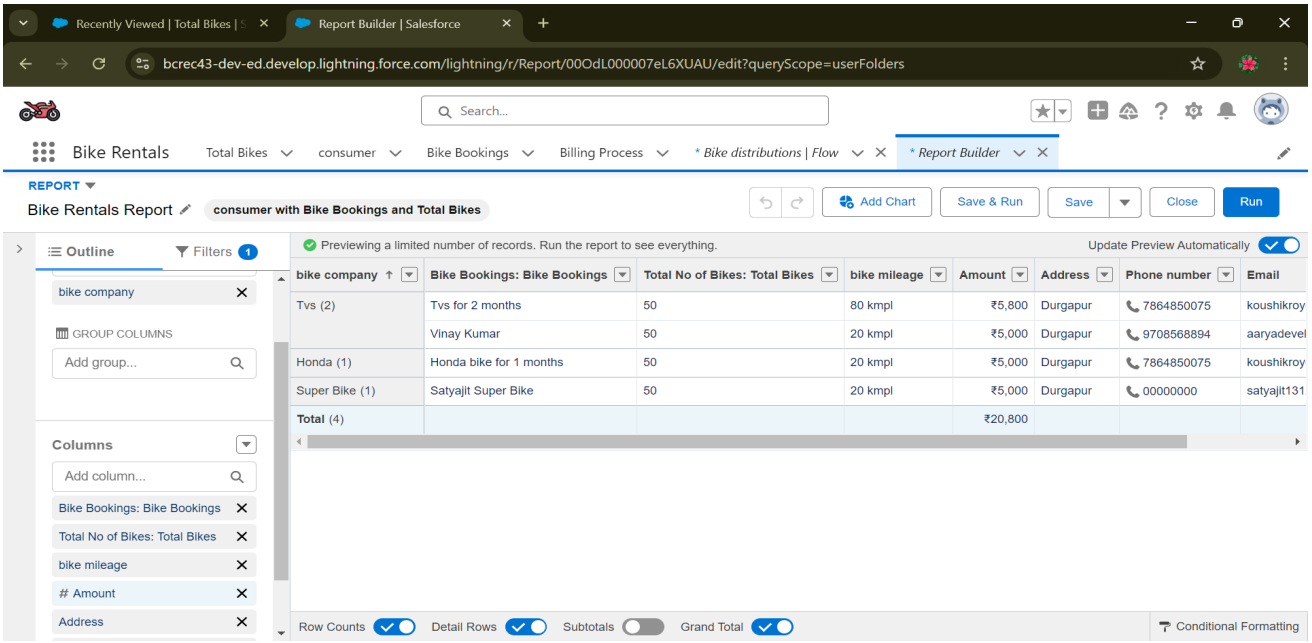


Step 12. Create Report

1. Go to the app -click on the reports tab

2. Click New Report.
3. Select report type from category or from report type panel or from search panel “consumer with Bike Bookings and total bikes” >> click on start report.
4. Customize your report.
5. Add fields from left pane as shown below.
6. Click the column drop down and select bucket list.
7. Click apply it.

Screenshot of created report:



Create sharing report to owner:

1. Click edit drop down and select subscribe option
2. Follow as per below image.
3. After selecting the run report as a “another person” select your personal account or whom you want to send that mail to.
4. Click save.

Step 13. Create Dashboards

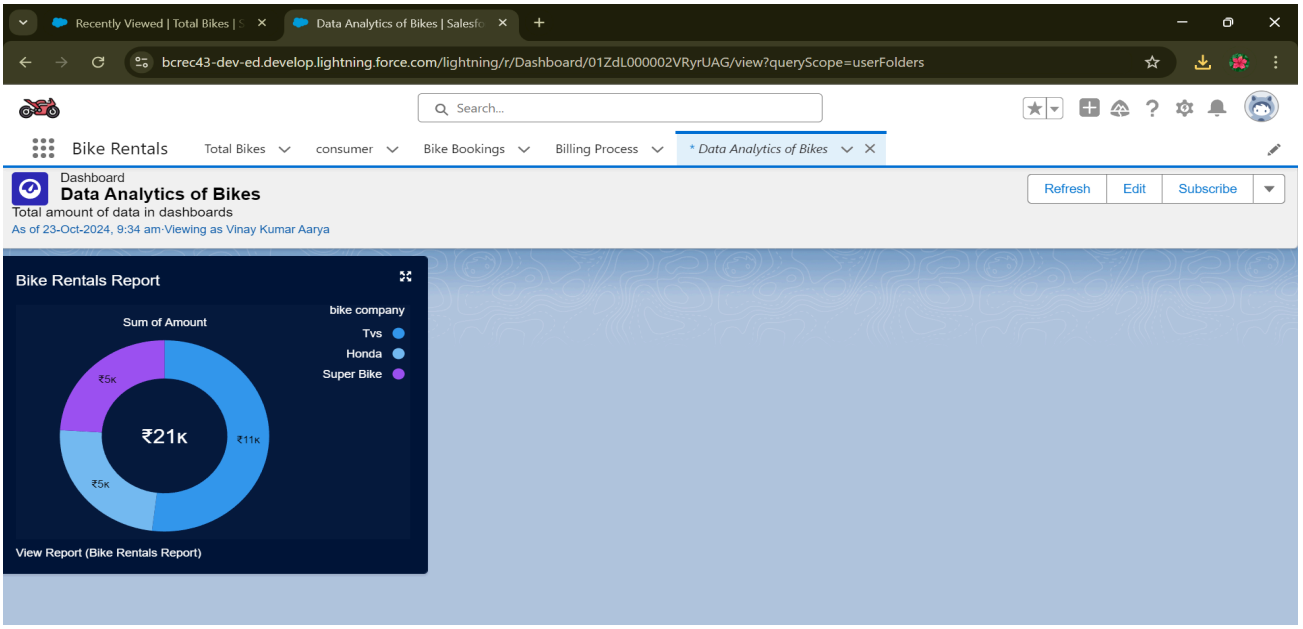
Create Dashboard Folder:

1. Click on the app launcher and search for the dashboard.
2. Click on the dashboard tab.
3. Click the new folder, give the folder label as “total rent amount”.
4. Folder unique names will be auto populated.
5. Click save.

Create Dashboard:

1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that was created, and click on create.
3. Select add component.
4. Select a Report and click on select.
5. Select the dark component and add to the dashboards.
6. Save it.
7. Click done.

Screenshot of created dashboard:



5. Testing and Validation

Approach to Testing:

- **Unit Testing:**
  - Extensive unit testing was performed on all Apex classes and triggers to ensure the core logic of the CRM system functions as expected. This includes validations for correct handling of bike rentals, updates to mileage, and ensuring rental periods (1, 2, or 3 months) are correctly applied.
  - Triggers for automatic email notifications and data updates were tested to ensure real-time functionality without errors.
- **User Interface Testing:**
  - User interface testing was conducted to verify the usability and functionality of the system from both the shop owner's and customer's perspectives.
  - Test cases included scenarios such as initiating a new rental, updating a rental, viewing reports, and receiving email notifications.
  - The responsiveness of the Lightning Pages was validated to ensure compatibility with different devices, including desktops and mobile devices.

6. Key Scenarios Addressed by Salesforce in the Implementation Project

The implementation addresses the following key scenarios:

- **Bike Rental Process:** Customers can rent bikes by selecting a company, choosing the bike by mileage, and specifying the rental duration. Salesforce automates the rental process and manages contracts efficiently.
- **Automated Customer Notifications:** Salesforce triggers send automatic emails to customers confirming the rental details, renewal reminders, and notifications of upcoming end-of-rental periods.
- **Bike Maintenance Alerts:** Shop owners receive alerts when bikes reach certain mileage thresholds, helping them manage maintenance schedules proactively.
- **Rental Analysis and Reporting:** Shop owners receive comprehensive monthly reports, including rental statistics, bike usage trends, and revenue analysis through Salesforce Dashboards.
- **Data Security:** Salesforce's data security model ensures that sensitive customer and rental

information is protected while allowing authorized personnel to access relevant data.

7. Conclusion

The "Bike Rentals CRM Solution" project successfully delivered a comprehensive, easy-to-use platform for managing bike rentals. By integrating custom objects, workflows, and automated processes within Salesforce, the system streamlines the entire rental lifecycle from customer onboarding to reporting and analytics.

Key achievements include:

- Fully automated customer notifications and bike rental management workflows.
- Monthly reports and dashboards that empower shop owners with critical business insights.
- A user-friendly interface with customizable components, ensuring ease of use for both shop owners and customers.
- Robust data security measures that ensure compliance and protect user information.
- Successful unit and interface testing that validated the functionality and reliability of the solution.