SARASWATHY COLLEGE OF ENGINEERING AND TECHNOLOGY

OLLAKUR,TINDIVANAM - 604305

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BACHELOR OF ENGINEERING

2024-2025

FIFTH   SEMESTER

SALESFORCE   DEVELOPER

TEAM   MEMBERS :

K.LOGESHWARI   421822104028

G.KIRAN   421822104027

K.KEERTHANA   421822104026

P.KARANRAJ   421822104025

SARASWATHY COLLEGE OF ENGINEERING AND TECHNOLOGY

OLLAKUR,TINDIVANAM - 604305

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BACHELOR OF ENGINEERING

2024-2025

certified that this is a bonafide record of work done by

Name   : Kiran.G

NM ID   : 543026783AC0FA2A75E4FBB8BEE4898C

Universite Reg.no   : 421822104027

Semester   : 5

Branch   : CSE

Year   : 2024-2025

Staff-in-Charge   **Head of the Department**

Submitted for the_____

Practical Examination held on_____

Internal Examiner   **External Examiner**

SALES AUTOMOBILE USING SALESFORCE CRM

PROJECT VIEW:

This project is focused on creating a comprehensive Salesforce Automobile Information CRM, designed to address the challenges of managing and tracking automotive sales, inventory, and customer relationships in the automobile industry. The goal is to deliver a solution that leverages Salesforce CRM, cloud technologies, and automation to improve the management of customer interactions, streamline inventory tracking, and optimize sales processes. Through this project, we aim to enhance operational efficiency, customer experience, and data accuracy, while supporting the long-term goals of the automobile dealership or sales organization.

OBJECTIVES:

Business Goals:
- Streamline the management of customer data, vehicle inventory, and sales processes.
- Enhance customer engagement through personalized service and targeted marketing.
- Improve operational efficiency by automating routine tasks and workflows.

Specific Outcomes:

- Develop an integrated system for tracking vehicle inventory, customer inquiries, sales leads, and follow-ups.
- Implement automated workflows for sales, service, and customer support processes.
- Enable detailed reporting and analytics for performance monitoring and decision-making.
- Ensure seamless integration with other systems like financial and service management tools.

Salesforce Key Features and Concepts Utilized:
This section highlights the main Salesforce functionalities applied in this CRM system:
- **Salesforce Sales Cloud:** Used for managing leads, opportunities, and sales pipelines.
- **Salesforce Service Cloud:** Provides customer service functionality, including case management and customer support.
- **Custom Objects and Fields:** Used to track vehicle information (make, model, year, VIN, etc.) and manage inventory.
- **Apex Triggers and Classes:** Used to automate processes such as updating inventory status, sending notifications, or creating follow-up tasks for sales representatives.
- **Reports and Dashboards:** Provide real-time insights into sales performance, customer activity, and inventory levels.
- **Lightning Web Components (LWC):** Custom user interfaces for improved user experience and interaction with the CRM.

Detailed Steps to Solution Design:
Data Models:

Define custom objects like Vehicle Inventory, Customer, and Sales Opportunity to store all relevant data.

Establish relationships between objects such as Customer to Sales Opportunity and Vehicle Inventory to Sales Opportunity.

Design custom fields in each object (e.g., Vehicle Color, Model Year, Price, Customer Preferences).

User Interface Design:

- Create custom Lightning pages for sales reps, service agents, and management to easily access and update information.
- Utilize Lightning App Builder to create a user-friendly dashboard that displays key metrics such as active sales leads, available inventory, and customer interactions.

Business Logic:

- Define automation rules for sales and service processes (e.g., automating lead assignment, setting up reminders for follow-up, and creating tasks based on specific triggers).
- Use Flow and Process Builder to streamline sales workflows, such as sending automated emails to customers after a purchase or sending alerts when inventory is running low.

AUTOMOBILE INFORMATION OBJECT:

What Is an Object?

Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects
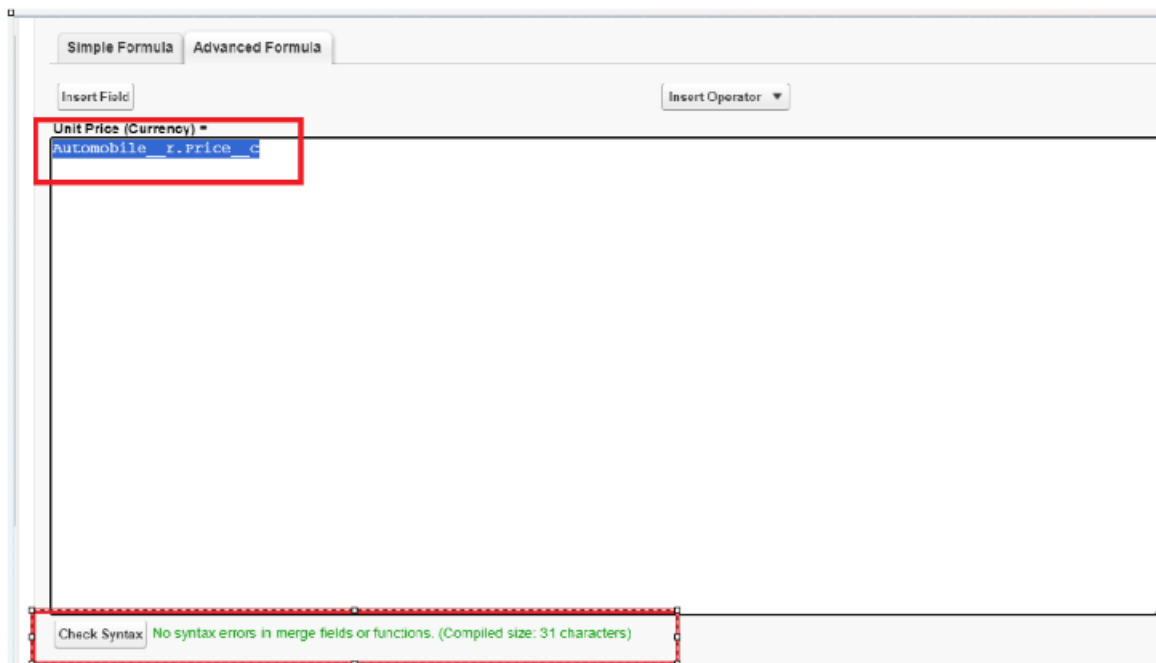
Salesforce objects are of two types:

1. **Standard Objects**: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. **Custom Objects**: Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

Create Automobile Information Object

1. Download and open this spreadsheet, save it as AutomobileInformation.csv.

Creating Formula Field in Opportunity Automobile Object

Updating field in Invoice Object

Creating Remaining Fields in Objects

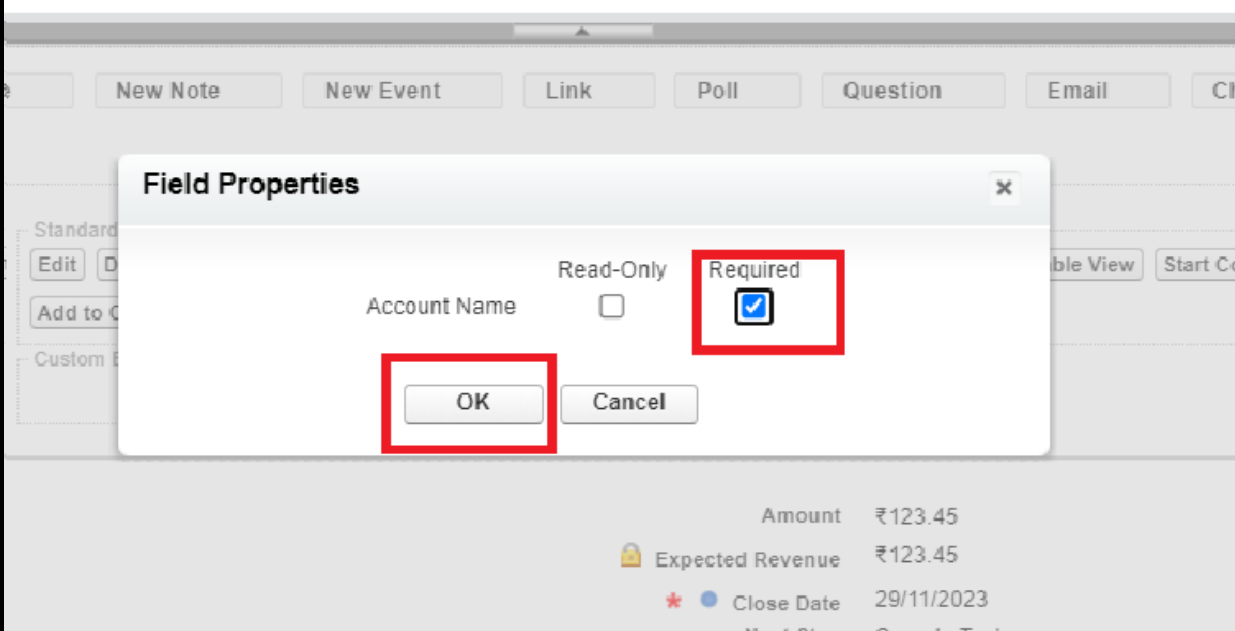| s.no	Object name	Fields	
1	Invoice	**Field Name**	Opportunity
		Data type	Master Detail relationship Object : Opportunity

Page Layouts:

Page Layout in Salesforce allows us to customize the design and organize detail and edit pages of records in Salesforce. Page layouts can be used to control the appearance of fields, related lists, and custom links on standard and custom objects' detail and edit pages.

Edit the Page layout for Opportunity Object

Step 1: Go to Setup >> Click on Object Manager >> On the search bar, select Opportunity Layout. You can notice Page Layouts on the left panel

Step 2: Click on Page Layouts, Click on 'Opportunity Layouts'.

Step 3: In the Opportunity Detail Section, you can see various fields. Go on Account And Click on that Properties icon of Account name Field.

Step 4: check the Required box for Account name and click on Ok.

Step 5: Click on Save.

Step 4 : Adjust the Fields as given below for A good looking view.

Step 5 : Click on Save.

Apex Trigger

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.
A trigger is Apex code that executes before or after the following types of operations:

insert
update
delete
merge
upsert
undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

Before Trigger: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

After Trigger: This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

Opportunity Automobile quantity

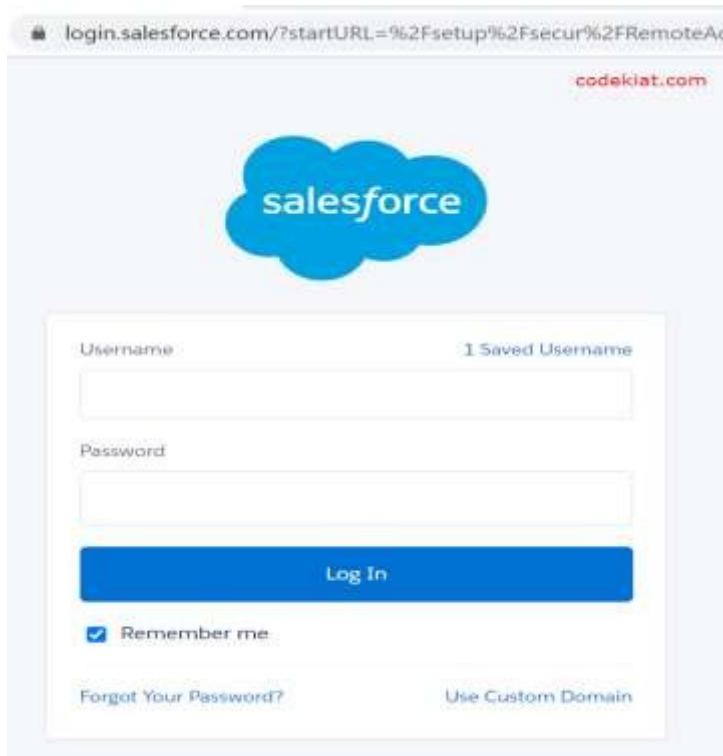### Code:

```
public class OpportunityHandlerClass {

public static void opportunityAutomobileQuantity(List<Opportunity> LstOpportunity, Map<Id,Opportunity> OldMapOpportunity){
    set<Id> opportunityIds = new set<Id>();
    for(Opportunity opp : LstOpportunity){
        if(opp.StageName =='Closed Won' ){
            opportunityIds.add(opp.Id);
        }
    }
    Map<Id,Opportunity_Automobile__c> lstOpportunityAutomobile =new
Map<Id,Opportunity_Automobile__c>([SELECT Id, Opportunity__c, Automobile__c, Quantity__c,
Unit_Price__c, Total_Price__c FROM Opportunity_Automobile__c Where Opportunity__c IN:
opportunityIds]);

    set<Id> AutoInformationIds = new set<Id>();
    for(Opportunity_Automobile__c OppAuto: lstOpportunityAutomobile.values()){
        if(OppAuto.Automobile__c != null){
            AutoInformationIds.add(OppAuto.Automobile__c);
        }
    }
    List<Automobile_Information__c> lstAutomobileInfomation = new List<Automobile_Information__c>();
    Map<Id,Automobile_Information__c> MapAutomobileInformation = New
Map<Id,Automobile_Information__c>([SELECT Quantity__c, Price__c, Name, Id FROM
Automobile_Information__c WHERE Id IN: AutoInformationIds]);
    For(Opportunity_Automobile__c AutoOpp : lstOpportunityAutomobile.Values()){
        decimal num = 0;
        if(AutoOpp.Automobile__c == MapAutomobileInformation.get(AutoOpp.Automobile__c).Id &&
OldMapOpportunity.get(AutoOpp.Opportunity__c).stagename != 'Closed Won'){

            num = MapAutomobileInformation.get(AutoOpp.Automobile__c).Quantity__c- AutoOpp.Quantity__c;
```

Create Lightning Web Component:

XML File Code:

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
<apiVersion>58.0</apiVersion>
<isExposed>true</isExposed>
 <targets>
    <target>lightning__RecordAction</target>
    <target>lightning__RecordPage</target>
 </targets>
  </LightningComponentBundle>
```

JS File Code :

```javascript
import { LightningElement, api, track, wire } from 'lwc';
import getInvoices from '@salesforce/apex/OpportunityInvoiceswithLWC.getInvoices';
export default class InvoiceOpportunity extends LightningElement {
@api recordId;
@track invoiceCollection
cols = [
   {label:"ID" , fieldName:'Name'},
   {label:"Opportunity Id" , fieldName:'Opportunity__c'},
   {label:"Quantity" , fieldName:'Quantity__c'},

    {label:"Total Price" , fieldName:'Total_Price__c'},
     {label:"Purchase Date" , fieldName:'Purchase_Date__c'}
]
```

Create Lightning Web Component:

JS File Code :

```
import { LightningElement, api, track, wire } from 'lwc';
import getInvoices from '@salesforce/apex/OpportunityInvoiceswithLWC.getInvoices';
export default class InvoiceOpportunity extends LightningElement {
@api recordId;
@track invoiceCollection
cols = [
    {label:"ID" , fieldName:'Name'},
    {label:"Opportunity Id" , fieldName:'Opportunity__c'},
    {label:"Quantity" , fieldName:'Quantity__c'},
    {label:"Unit Price" , fieldName:'Unit_Price__c'},
    {label:"Total Price" , fieldName:'Total_Price__c'},
    {label:"Purchase Date" , fieldName:'Purchase_Date__c'}
]
@wire(getInvoices,{OpportunityId:'$recordId'})
invoicefunction({data,error}){
  console.log(this.recordId +'this is record Id');
    if(data){
      console.log(data);
      this.invoiceCollection = data
    }if(error){
     console.log('this is error')
     console.log('error');
    }
 }
}
```
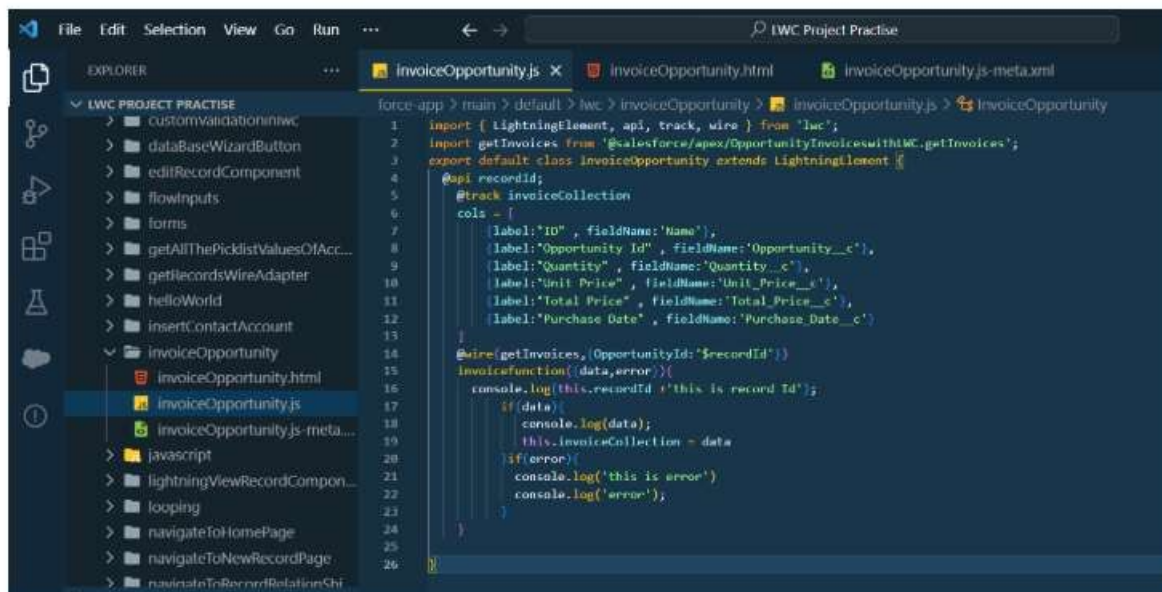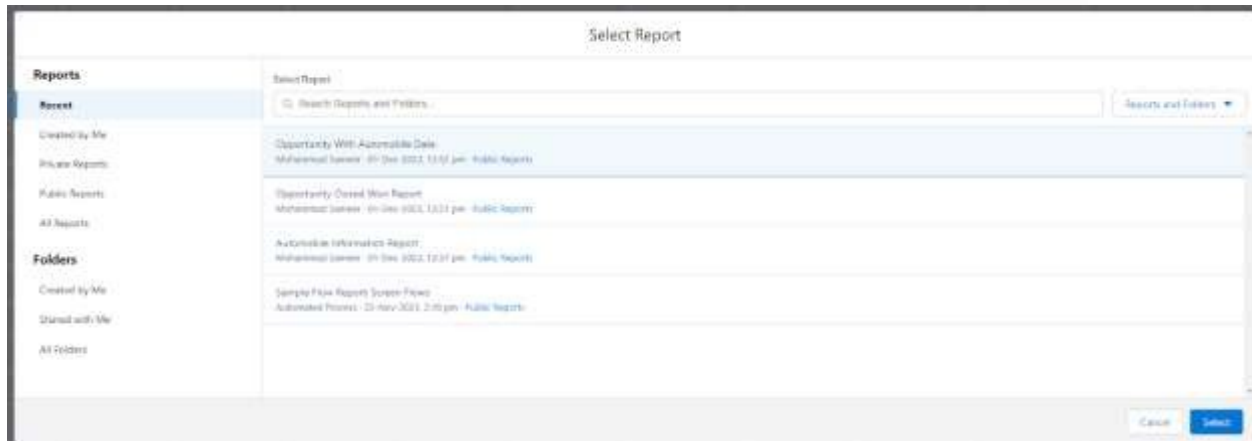
5.Click Add then click on Save and then click on Done.

The Created Dashboard will look like this.

Conclusion:
Summary of Achievements:

The Salesforce Automobile Information CRM project has successfully integrated sales and service functionalities into a single platform, improving both internal operations and customer satisfaction. The CRM system now allows seamless tracking of inventory, sales opportunities, and customer interactions while automating key processes to boost efficiency. With enhanced reporting and analytics, the organization can make data-driven decisions to grow their business and improve the customer experience. The solution is scalable, customizable, and alignswith the long-term goals of the automobile dealership, setting a foundation for future growth and digital transformation.