

FIELD SERVICE WORK ORDER OPTIMIZATION

By

M.Shiva Sainath Reddy

2111IT010076@mallareddyuniversity.ac.in

ABSTRACT

The Field Service Work Order Optimization System aims to enhance the efficiency of a company's installation and repair services. It uses a centralized database to assign tasks to the most appropriate technicians based on their skills, location, and availability. By using a task prioritization algorithm, the system ensures that technicians are matched with the right jobs. Automated notifications keep technicians updated, while analytical insights help improve service over time. This system increases operational efficiency, reduces costs, and boosts customer satisfaction in the field service industry.

Key Technologies:

1. Salesforce Field Service: A platform that provides tools for scheduling, dispatching, and real-time communication between technicians and managers.
2. Artificial Intelligence (AI) & Machine Learning (ML): These technologies predict service demand, optimize technician schedules, and match technicians to jobs based on key factors like skill and proximity.
3. Predictive Analytics: Analyzes historical data to predict future service needs and prevent potential problems before they arise.
4. Internet of Things (IoT): IoT devices collect real-time data from equipment in the field, helping with timely maintenance and quick issue resolution.

Implementation Phases:

1. Salesforce Field Service Setup: Implementing scheduling, dispatching, and communication tools.
2. AI & ML Integration: Developing algorithms for smarter technician scheduling and task matching.
3. Predictive Analytics: Building models to predict service needs based on past data.
4. IoT Integration: Bringing in data from connected field devices to improve response times and maintenance.

Potential Challenges:

1. Data Integration: Combining data from multiple sources and older systems can be challenging.
2. Adoption by Stakeholders: Ensuring everyone is on board and comfortable with the new system.
3. Scalability: Making sure the system can grow and handle larger workloads in the future.
4. Data Security: Keeping customer and business data safe from breaches.

Measurable Outcomes:

1. Improved Efficiency
2. Higher Customer Satisfaction
3. Optimized Operations

Functional Requirements:

1. Managing Work Orders
2. Scheduling and Dispatching Technicians
3. Resource Management
4. Mobile Access for Technicians
5. Customer Communication Tools
6. Reporting and Analytics
7. Integration with Other Systems
8. User Access and Security Controls
9. Ongoing Maintenance and Support

By fulfilling these requirements, the system will streamline field operations, enhance customer service, and help achieve business objectives.

INDEX PAGE

SI No.	Module or Tasks Labels	Page No.
1	<u>Task 1: Object</u> 1.1 Create Technician Object 1.2 Create WorkOrder Object 1.3 Create Assignment Object	6-8
2	<u>Task 2: Tabs</u> 2 Create a custom tab	9
3	<u>Task 3: The Lightning App</u> 3 Create a Lightning App	10-11
4	<u>Task 4: Fields & Relationship</u> 4.1 Creating Lookup Field in Assignment Object 4.2 Manage your picklist values 4.3 Manage your picklist values 4.4 Creating Formula Field in WorkOrder Object	12-15
5	<u>Task 5: Profiles</u> 5 Technician Profile	16
6	<u>Task 6: Users</u> 6 Create User	17

7	<u>Task 7: Apex Trigger</u> 7.1 Create an Apex Class 7.2 Create an Apex Trigger 7.3 Create an Apex Class 7.4 Create an Apex Trigger 7.5 Create an Apex Class 7.6 Create an Apex Trigger 7.7 Create an Asynchronous Apex Class 7.8 Create an Apex Schedule Class 7.9 Create an Schedule apex	18-28
8	<u>Task 8: Reports & Dashboards</u> 8.1 Report 8.2 Create Reports 8.3 Dashboard 8.4 Create Dashboards	29-32

INTRODUCTION

The Field Service Work Order Optimization System streamlines operations for a company providing installations and repairs. Utilizing a robust database, the system efficiently matches work orders with skilled technicians based on technicians' location, availability, and skills. The system employs a prioritization algorithm, focusing on assigning tasks to technicians. Automated communication keeps technicians informed, while analytics offer insights for continuous improvement. Overall, this solution maximizes efficiency, reduces operational costs, and improves customer satisfaction in the dynamic realm of field service operations.

Task 1:

1.1 Create Technician Object:

An entity representing field technicians, capturing details like skills, name, location, availability, and contact information for optimized service dispatch.

Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

CSV File Details

Encoding Format ⓘ
Unicode (UTF8)

Values Separated By
Comma

Field Label Source
☐ Enter manually
☒ Detect from row

* Field Labels Row
1

Import 5 rows of Data? ⓘ
☐ No, skip import
☒ Yes, import data

Record Name Field ⓘ
Let Salesforce Create a Default R

Fields 7 of 7 to import

☐ Hide mapped fields

IMPORT FILE FIELD NAME		SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS ⓘ	FIELD PREVIEW
✓ Technician ID	×	Technician ID	Text	✓	T-0001
✓ Name	×	Name	Text	✓	Raghu
✓ Phone	×	Phone	Integer	✓	7892341560
✓ Email	×	Email	Email	✓	lasyachakrala04@gmail.com
✓ Location	×	Location	Text	✓	Hyderabad
✓ Availability	×	Availability	Text	✓	Available

Back

Next

After creating technician details, the Quick box looks like the below

<div> <div> <div>SETUP</div> <div>Object Manager</div> <div>1 Items, Sorted by Last Modified</div> </div> </div>		<div> <div> <div>tech</div> <div>Schema Builder</div> <div>Create ▾</div> </div> </div>			
LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Technician	Technician__c	Custom Object		25/09/2024	✓

1.2 Create WorkOrder Object:

An entity tracking service tasks, detailing job requirements, status, assigned technician, and customer information for efficient field operations.

Welcome to Salesforce: Verify

Object Manager | Salesforce

Student - Skill Wallet

Object creator

object-creator.salesforce.com/load-data-wizard.xhtml

Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

CSV File Details

Encoding Format

Unicode (UTF8)

Values Separated By

Comma

Field Label Source

☐ Enter manually
 ☒ Detect from row

* Field Labels Row

1

Import 2 rows of Data?

☒ No, skip import
 ☐ Yes, import data

Record Name Field

WorkOrder ID

Fields 7 of 7 to import

☐ Hide mapped fields

IMPORT FILE FIELD NAME		SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
✓ WorkOrder ID	×	WorkOrder ID	Text	✓	WO-(0001)
✓ Email	×	Email	Email	✓	example1@workorder.com
✓ Service Type	×	Service Type	Picklist	✓	Maintenance
✓ Description	×	Description	Text Area (Long)	✓	
✓ Location	×	Location	Picklist	✓	Pune
✓ Priority	×	Priority	Picklist	✓	Low

Back

☒

Next


After creating the WorkOrder Custom object it looks like the below







<div> <div> <div>SETUP</div> <div>Object Manager</div> <div>1 Items, Sorted by Last Modified</div> </div> </div>		<div> <div> <div>WorkOrder</div> <div>Schema Builder</div> <div>Create ▾</div> </div> </div>			
LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
WorkOrder	WorkOrder__c	Custom Object		26/09/2024	✓

1.3 Create Assignment Object :

An entity linking technicians to work orders, detailing assignment dates, priority, status, and specific tasks for optimized field service.

After creating the Assignment custom object, the object manager bar looks the below






Setup


Home

Object Manager

 **Object Manager**

2 Items. Sorted by Last Modified

Label	API Name	Type	Description	Last Modified	Deployed
Assignment	Assignment__c	Custom Object		26/09/2024	<input checked="" type="checkbox"/>
Location Group Assignment	LocationGroupAssignment	Standard Object			



Task 2:

Creating a Custom Tab

A user interface element in Salesforce that provides access to custom objects, records, or webcontent, enhancing navigation and organization of data within the Salesforce environment.

To create a Tab:(Assignment)

1. Go to the setup page --> type Tabs in the Quick Find bar --> click on tabs --> New (under the custom object tab)
2. Select Object(Assignment) --> Select any tab style --> Next (Add to profiles page) keep it as default --> Next (Add to Custom App) keep it as default --> Save.

Note: Tabs for WorkOrder & Technician objects do get created automatically. We do not need to create tabs for those objects.

After following the above steps, the output looks like this:

The screenshot shows the Salesforce Setup interface. The top navigation bar includes the Salesforce logo, a search bar labeled "Search Setup", and several utility icons. Below the navigation bar, the "Setup" menu is open, showing "Home" and "Object Manager". The left sidebar contains a search bar with "tabs" entered, and a list of categories under "User Interface", including "Rename Tabs and Labels" and "Tabs". The main content area is titled "Custom Tabs" and includes a "Help for this Page" link. Below the title, there is a descriptive paragraph about custom tabs. The interface is divided into four sections: "Custom Object Tabs", "Web Tabs", "Visualforce Tabs", and "Lightning Component Tabs". Each section has a "New" button and a "What is This?" link. The "Custom Object Tabs" section contains a table with three rows of existing tabs.

Action	Label	Tab Style	Description
Edit Del	Assignments	Airplane	
Edit Del	Technician	Box	
Edit Del	WorkOrder	Box	

The "Web Tabs", "Visualforce Tabs", and "Lightning Component Tabs" sections each display the message: "No [Tab Type] Tabs have been defined".

Task 3 :

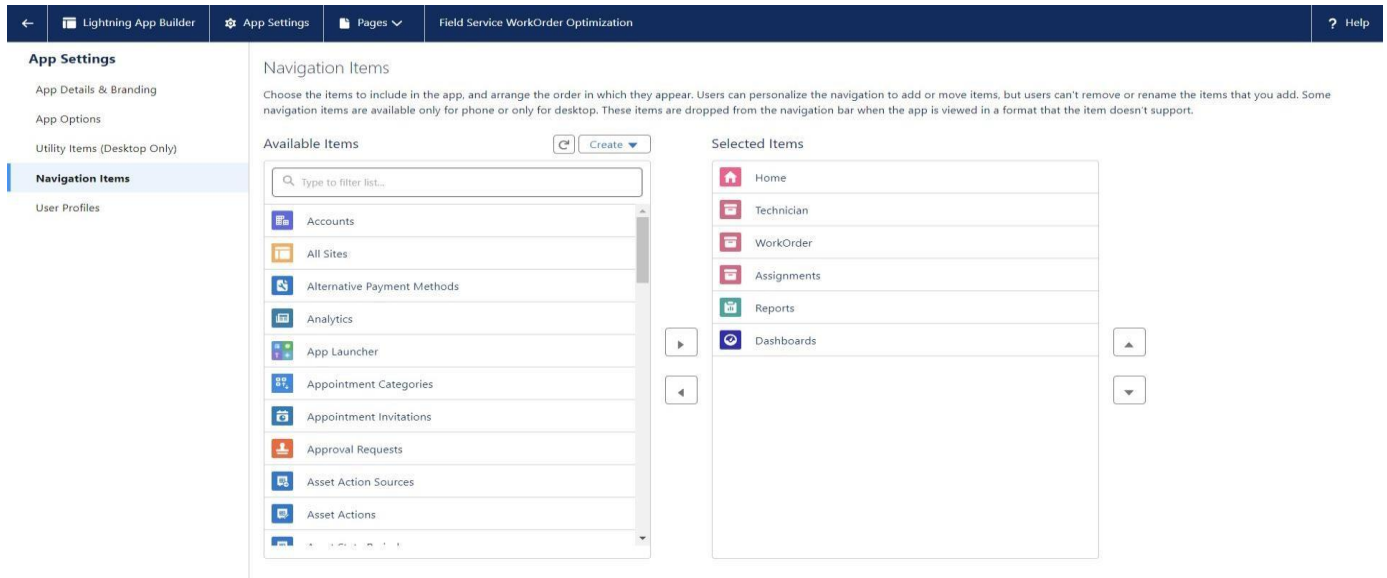
Create a Lightning App

To create a lightning app page:

1. Go to the setup page --> search “app manager” in quick find --> select “app manager” --> click on New lightning App.
2. Fill the app name in app details and branding as follow
App Name : Field Service WorkOrder Optimization Developer
Name : this will be auto populated
Description : Give a meaningful description
Image : optional (if you want to give any image you can, otherwise not mandatory)
Primary color hex value : keep this default

The screenshot shows the 'App Details & Branding' configuration page in the Lightning App Builder. The left sidebar lists 'App Settings' with 'App Details & Branding' selected. The main area is divided into 'App Details' and 'App Branding' sections. Under 'App Details', there are input fields for 'App Name' (containing 'Field Service WorkOrder Optimization'), 'Developer Name' (containing 'Field_Service_WorkOrder_Optimization'), and 'Description' (containing 'Give a meaningful description'). Under 'App Branding', there is an 'Image' section with an 'Upload' button and a 'Primary Color Hex Value' section with a dropdown menu showing a blue color and the hex value '#0070D2'. Below these is an 'Org Theme Options' checkbox labeled 'Use the app's image and color instead of the org's custom theme'. At the bottom, there is an 'App Launcher Preview' showing a blue square with 'FS' and the text 'Field Service WorkOrder O...' and 'Give a meaningful description'. The top navigation bar shows 'Lightning App Builder', 'App Settings', 'Pages', and 'Field Service WorkOrder Optimization'. The bottom status bar shows system icons, language (ENG IN), and time (23:00 26-09-2024).

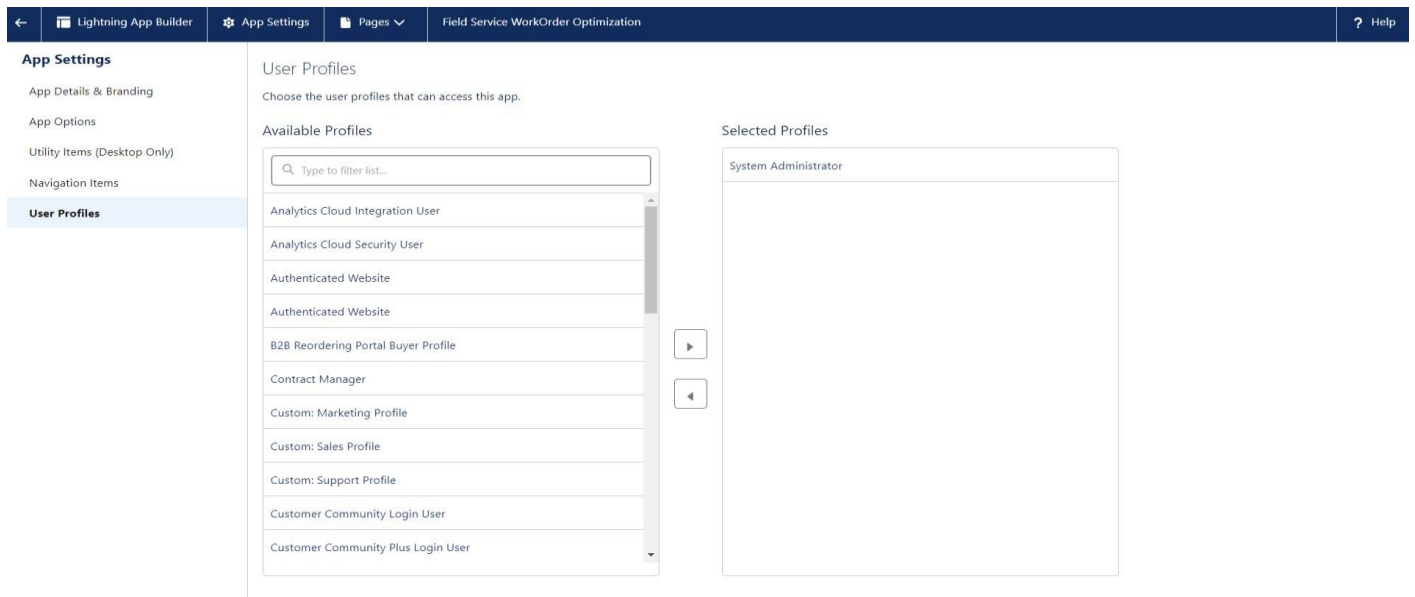
3. Then click Next --> (App option page) keep it as default --> Next --> (Utility Items) keep it as default --> Next
4. To Add Navigation Items:



Search the items in the search bar(Home, WorkOrder, Technician, Assignment, Reports, Dashboard) from the search bar and move it using the arrow button. Next. Note: select asset the custom object which we have created in the previous activity.

5. To Add User Profiles:

Search profiles (System administrator) in the search bar --> click on the arrow button --> save & finish.



This is the output after completion of following the above procedure.

Task 4 :

4.1 Creating Lookup Field in Assignment Object

A lookup field in the Assignment Object establishes a relationship with another object, such as Technicians or Work Orders, enabling users to link and reference related records for improved data organization and relational tracking.

Setup

Home

Object Manager

Assignment

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

Assignment Custom Field

WorkOrder ID

Back to Assignment

Validation Rules (0)

Custom Field Definition Detail

Field Information

Field Label	WorkOrder ID	Object Name	Assignment
Field Name	WorkOrder_ID	Data Type	Lookup
API Name	WorkOrder_ID__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Sree Lasya Chakrala, 25/09/2024, 11:36 pm	Modified By	Sree Lasya Chakrala, 26/09/2024, 8:38 pm

Lookup Options

Related To	WorkOrder	Child Relationship Name	Assignments
Related List Label	Assignments		
Required	<input type="checkbox"/>		

4.2 Manage your picklist values

WorkOrder

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

Picklist Values Used

Active and inactive picklist values 4 (1,000 max)

Field Dependencies

No dependencies defined.

Validation Rules

No validation rules defined.

Values

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit Del Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakrala, 25/09/2024, 10:56 pm
<input type="checkbox"/> Edit Del Deactivate	Nasik	Nasik	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakrala, 25/09/2024, 11:38 pm
<input type="checkbox"/> Edit Del Deactivate	Warangal	Warangal	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakrala, 25/09/2024, 11:38 pm
<input type="checkbox"/> Edit Del Deactivate	Nanded	Nanded	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakrala, 25/09/2024, 11:38 pm

Inactive Values

No inactive Values values defined.

4.3 Manage your picklist values :

Add following values to the respective fields in WorkOrder object:

Field	Values
Priority	High
Service Type	Hardware repair Troubleshoot/Debugging Lane-Management

SETUP > OBJECT MANAGER

WorkOrder

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

Picklist Values Used

Active and inactive picklist values 4 (1,000 max)

Field Dependencies

New

Field Dependencies Help ?

No dependencies defined.

Validation Rules

New

Validation Rules Help ?

No validation rules defined.

Values

New

Reorder

Replace

Printable View

Chart Colors

Delete Selected

Deactivate Selected

Replace Selected

Values Help ?

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit Del Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakraborty, 25/09/2024, 10:56 pm
<input type="checkbox"/> Edit Del Deactivate	High	High	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakraborty, 26/09/2024, 8:40 pm
<input type="checkbox"/> Edit Del Deactivate	Low	Low	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakraborty, 26/09/2024, 8:40 pm
<input type="checkbox"/> Edit Del Deactivate	Medium	Medium	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakraborty, 26/09/2024, 8:40 pm

Inactive Values

Delete Unused Values

Inactive Values Help ?

No Inactive Values values defined.

SETUP > OBJECT MANAGER

WorkOrder

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

Picklist Values Used

Active and inactive picklist values 4 (1,000 max)

Field Dependencies

New

Field Dependencies Help ?

No dependencies defined.

Validation Rules

New

Validation Rules Help ?

No validation rules defined.

Values

New

Reorder

Replace

Printable View

Chart Colors

Delete Selected

Deactivate Selected

Replace Selected

Values Help ?

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit Del Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakraborty, 25/09/2024, 10:56 pm
<input type="checkbox"/> Edit Del Deactivate	Hardware repair	Hardware repair	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakraborty, 25/09/2024, 11:41 pm
<input type="checkbox"/> Edit Del Deactivate	Troubleshoot/Debugging	Troubleshoot/Debugging	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakraborty, 25/09/2024, 11:41 pm
<input type="checkbox"/> Edit Del Deactivate	Lane-Management	Lane-Management	<input type="checkbox"/>	Assigned dynamically	Sree Lasya Chakraborty, 25/09/2024, 11:41 pm

Inactive Values

Delete Unused Values

Inactive Values Help ?

No Inactive Values values defined.

4.4 Creating Formula Field in WorkOrder Object

A formula field in the Work Order Object automatically calculates and displays data based on other fields or custom logic. This feature streamlines data entry, ensures consistency, and provides real-time insights without manual updates.


1. Repeat steps 1 and 2 mentioned in activity 1
2. Select Data type as “Formula” and click Next.
3. Give Field Label and Field Name as “Date” and select formula return type as “Date” and click next.
4. Under Advanced Formula, write the formula and click “Check Syntax”
Formula: CreatedDate
5. Next--> Next--> Save.

The screenshot shows the Salesforce Setup interface. At the top, there's a navigation bar with 'Setup', 'Home', and 'Object Manager'. Below this, the 'WorkOrder' object is selected. The left sidebar shows a list of setup options, with 'Fields & Relationships' highlighted. The main area is titled 'Simple Formula' and 'Advanced Formula'. Under 'Simple Formula', there are dropdowns for 'Select Field Type' (set to 'WorkOrder') and 'Insert Field' (set to '-- Insert Merge Field --'). An 'Insert Operator' dropdown is also present. The formula text area contains 'Date (Date) =' followed by 'CreatedDate' on a new line. At the bottom, a 'Check Syntax' button is visible, along with a message: 'No syntax errors in merge fields or functions. (Compiled size: 20 characters)'.


4.5 Creating Remaining fields for the respective objects

Now create the remaining fields using the data types mentioned in the table.

SI No	Object Name	Field				
1	Assignment	<table><tr><th>Field Name</th><th>Datatype</th></tr><tr><td><ul style="list-style-type: none">● Technician ID● Assignment Date● Completion Date</td><td>Lookup(Technician) Formula: return type : Date (WorkOrder_ID_r.Date_c) Formula: return type : Date IF(ISPICKVAL(WorkOrder_ID_r.Status_c , 'Resolved'), WorkOrder_ID_r.LastModifiedDate , NULL)</td></tr></table>	Field Name	Datatype	<ul style="list-style-type: none">● Technician ID● Assignment Date● Completion Date	Lookup(Technician) Formula: return type : Date (WorkOrder_ID_r.Date_c) Formula: return type : Date IF(ISPICKVAL(WorkOrder_ID_r.Status_c , 'Resolved'), WorkOrder_ID_r.LastModifiedDate , NULL)
Field Name	Datatype					
<ul style="list-style-type: none">● Technician ID● Assignment Date● Completion Date	Lookup(Technician) Formula: return type : Date (WorkOrder_ID_r.Date_c) Formula: return type : Date IF(ISPICKVAL(WorkOrder_ID_r.Status_c , 'Resolved'), WorkOrder_ID_r.LastModifiedDate , NULL)					



[Setup](#)
[Home](#)
[Object Manager](#)


 SETUP > OBJECT MANAGER
Assignment

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

8 Items, Sorted by Field Label

[New](#)
[Deleted Fields](#)
[Field Dependencies](#)
[Set History Tracking](#)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assignment Date	Assignment_Date__c	Formula (Date)		
Assignment ID	Name	Auto Number		✓
Completion Date	Completion_Date__c	Formula (Date)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Technician ID	Technician_ID__c	Lookup(Technician)		✓
WorkOrder ID	WorkOrder_ID__c	Lookup(WorkOrder)		✓

Task 5 :

Technician Profile

1. Go to setup --> type profiles in the quick find box --> click on profiles --> click on new profile.
2. Select 'Standard Platform User' for existing profile and give 'Technician' for Profile Name and click on Save.
3. While still on the profile page, then click Edit.
4. While still on the profile page, then click Edit.
5. Scroll down and Click on Save.
6. Now from the profile detail page scroll down to custom field level security click on view next to WorkOrder object.
7. Click on Edit, enable the check box for the status field.
8. Click on Save.

The screenshot displays the Salesforce Setup interface. The left sidebar shows the navigation menu with 'Setup' selected. The main content area is titled 'Profiles' and shows the 'Technician' profile. The profile details include the name 'Technician', user license 'Salesforce Platform', and a custom profile checkbox checked. The 'Page Layouts' section lists various standard object layouts and their assignments. The 'Custom Field Level Security' section is partially visible at the bottom.

Profile Detail

Name	Technician
User License	Salesforce Platform
Description	
Created By	Steve Lasya Chakraborty, 26/09/2024, 12:17 am
Modified By	Steve Lasya Chakraborty, 26/09/2024, 6:26 pm

Page Layouts

Standard Object Layouts	Global	Lead
Global	Global Layout [View Assignment]	Lead Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Location Location Layout [View Assignment]
Home Page Layout	Home Page Default [View Assignment]	Location Group Location Group Layout [View Assignment]
Account	Account Layout [View Assignment]	Location Group Assignment Location Group Assignment Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Object Milestone Object Milestone Layout [View Assignment]
Appointment Invitation	Appointment Invitation Layout [View Assignment]	Operating Hours Operating Hours Layout [View Assignment]
Customer	Customer Layout [View Assignment]	Skill Requirement Skill Requirement Layout [View Assignment]
D&B Company	D&B Company Layout [View Assignment]	Social Persona Social Persona Layout [View Assignment]
Data Use Legal Basis	Data Use Legal Basis Layout [View Assignment]	Store Store Layout [View Assignment]
Data Use Purpose	Data Use Purpose Layout [View Assignment]	Task Task Layout [View Assignment]
Digital Wallet	Digital Wallet Layout [View Assignment]	Threat Detection Feedback Threat Detection Feedback Layout [View Assignment]
Email Message	Email Message Layout [View Assignment]	Time Slot Time Slot Layout [View Assignment]
Engagement Channel Type	Engagement Channel Type Layout [View Assignment]	User User Layout [View Assignment]
Event	Event Layout [View Assignment]	User Profile User Profile Layout [View Assignment]
Feed Item	Feed Item Layout [View Assignment]	User Provisioning Account User Provisioning Account Layout [View Assignment]
Fulfillment Order	Fulfillment Order Layout [View Assignment]	User Provisioning Log User Provisioning Log Layout [View Assignment]
Fulfillment Order Item Adjustment	Fulfillment Order Item Adjustment Layout [View Assignment]	User Provisioning Request User Provisioning Request Layout [View Assignment]
Fulfillment Order Item Tax	Fulfillment Order Item Tax Layout [View Assignment]	Waitlist MISSING LABEL - Property/File - val Waitlist not found in section StandardLayouts [View Assignment]
Fulfillment Order Product	Fulfillment Order Product Layout [View Assignment]	Web Store Configuration Web Store Configuration Layout [View Assignment]
Idea	Varies by Record Type [View Assignment]	Work Type Work Type Layout [View Assignment]

Task 6 :

Create User

User is engaged in the Field Service Workforce Optimization Project, utilizing Salesforce to optimize field operations, improve resource management, and enhance customer service through efficient scheduling, real-time tracking, and comprehensive analytics.

1. Go to setup --> type users in the quick find box --> select users --> click New user.
2. Fill in the fields
 1. First Name : Elina
 2. Last Name : Gilbert
 3. Alias : Give an Alias Name
 4. Email id : Give your Personal Email id
 5. Username : Username should be in this form: text@text.text
 6. Nick Name : Give a Nickname
 7. Role :
 8. User license : Salesforce Platform
 9. Profiles : Technician

The screenshot shows the Salesforce Setup interface. The left sidebar contains navigation options: Setup, Home, Object Manager, Users, Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, Feature Settings, Data.com, and Prospector. The main content area displays the 'Users' setup page for a new user named 'Elina Gilbert'. The 'User Detail' section includes fields for Name, Alias, Email, Username, Nickname, Title, Company, Department, Division, Address, Time Zone, Locale, Language, Delegated Approver, Manager, Receive Approval Request Emails, and Federation ID. The 'Role' section includes fields for User License, Profile, Active, Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, Mobile Push Registrations, Data.com User Type, Accessibility Mode (Classic Only), Debug Mode, and High-Contrast Palette on Charts. The user is currently assigned the 'Salesforce Platform' license and the 'Technician' profile. The 'Active' checkbox is checked. The 'Marketing User' checkbox is unchecked. The 'Offline User' checkbox is unchecked. The 'Knowledge User' checkbox is unchecked. The 'Flow User' checkbox is unchecked. The 'Service Cloud User' checkbox is unchecked. The 'Site.com Contributor User' checkbox is unchecked. The 'Site.com Publisher User' checkbox is unchecked. The 'WDC User' checkbox is unchecked. The 'Mobile Push Registrations' checkbox is unchecked. The 'Data.com User Type' dropdown is set to 'View'. The 'Accessibility Mode (Classic Only)' checkbox is unchecked. The 'Debug Mode' checkbox is unchecked. The 'High-Contrast Palette on Charts' checkbox is unchecked.

Task 7 :

7.1 Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
4. Give the Apex Class name as "WorkOrderClass".
5. Click ok.
6. Now write the code logic here

7. Source Code:

```
public class WorkOrderClass {
    public static void workOrder(List<WorkOrder_C> newListWorkOrder){ Map<Integer,
        List<String>> maptotech = new map<Integer,List<String>>();integer num = 0;
        List<WorkOrder_c> properWo = new List<WorkOrder_c>(); List<Assignment _
        c> lstAssignment = new List<Assignment_c>(); List<Technician_c>
        technicianToAssignment = new List<Technician_c>();for(WorkOrder_c iter :
        newListWorkOrder){
            List<String> lststring = new List<string>(); If(iter.Service_Type_c
            != null && iter.Location_c != null ){
                num = num+1;
                properWo.add(iter);
                lststring.add(iter.Service_Type_c);lststring.add(iter.Location_c);

                maptotech.put(num,lststring);
            }
        }
        Map<integer,Id> techId = new Map<integer,Id>();
        Map<Id,Technician_c> allTechnician = new Map<Id,Technician_c>([SELECT Id, Name,Phone_
        c, Location_c, Skills_c, Availability_c, Name_c, Email_c FROM Technician_c]);
        integer num2 = 0;
        For(Technician_c T : allTechnician.values()){num2 =
            num2+1;
            if(maptotech.get(num2) != null){
```

```

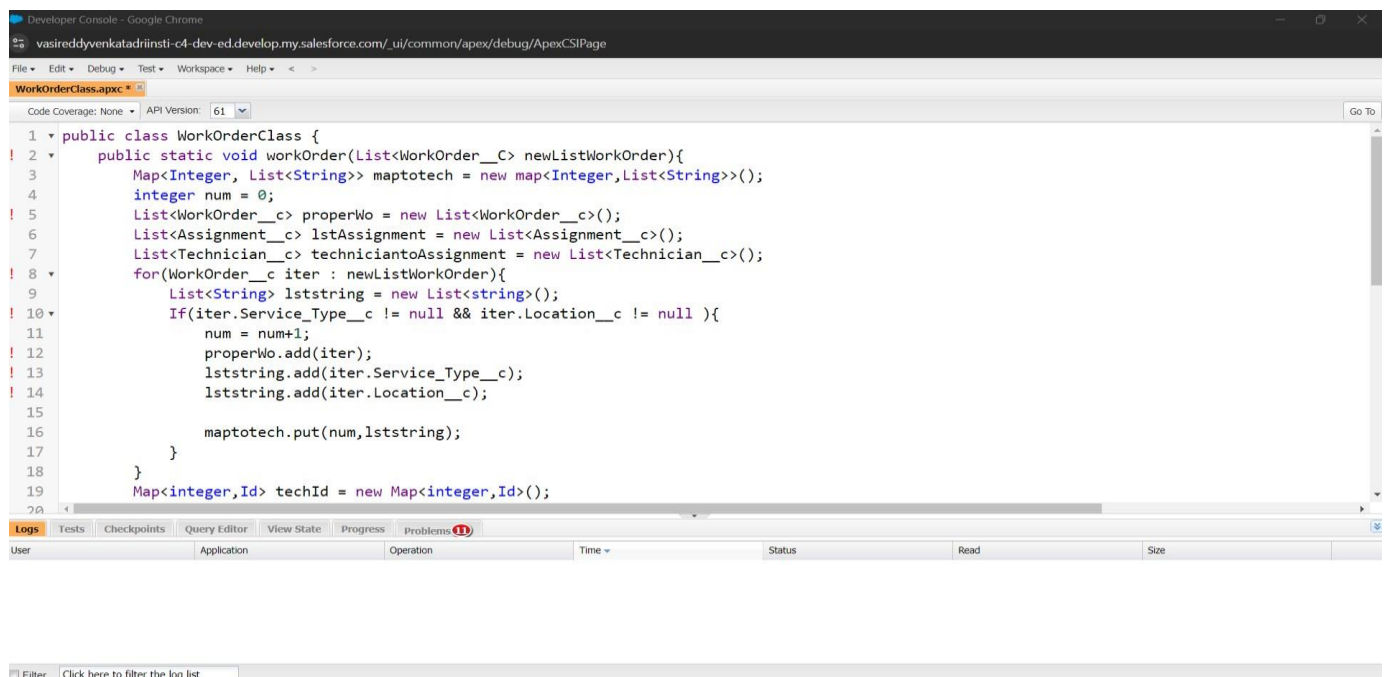
        List<string> valofmap = maptotech.get(num2);
        system.debug('error 1 ----> the maptotech is empty ---> ' + maptotech.get(num2));
        if(valofMap.contains(t.Skills_c) && ValofMap.contains(t.Location_c) &&
t.Availability_c == 'Available'){
            techid.put(num2,t.Id);
        }
    }

}

integer num3 = 0;
For(WorkOrder_c W : properWo){
    num3 = num3 + 1;
    Assignment_c A = new Assignment_c();
    A.WorkOrder_ID_c = W.Id;
    A.Technician_ID_c = techid.get(num3);
    lstAssignment.add(A);
}
If(!lstAssignment.IsEmpty()){ insert
    lstAssignment;
}
}
}

```

8. Save the code.(click on file --> Save)



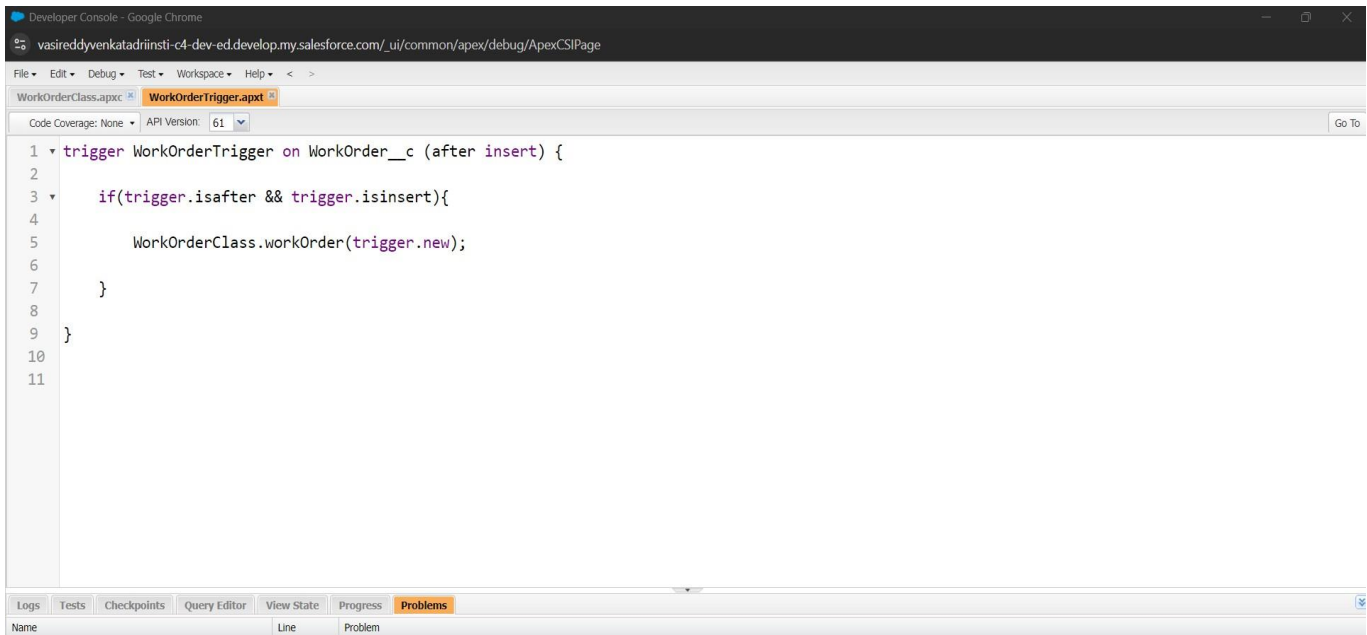
7.2 Create an Apex Trigger

1. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
2. Give the Apex Trigger name as “WorkOrderTrigger”, and select “WorkOrder_c” from the dropdown for sObject.
3. Click Submit.
4. Now write the code logic here

Source Code:

```
trigger WorkOrderTrigger on WorkOrder_c (after insert) { if(trigger.isafter &&
    trigger.isinsert){
        WorkOrderClass.workOrder(trigger.new);
    }
}
```

5. Save the code.(click on file --> Save)



7.3 Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
4. Give the Apex Class name as “AssigningEmail”.
5. Click ok.
6. Now write the code logic here

7. Source Code:

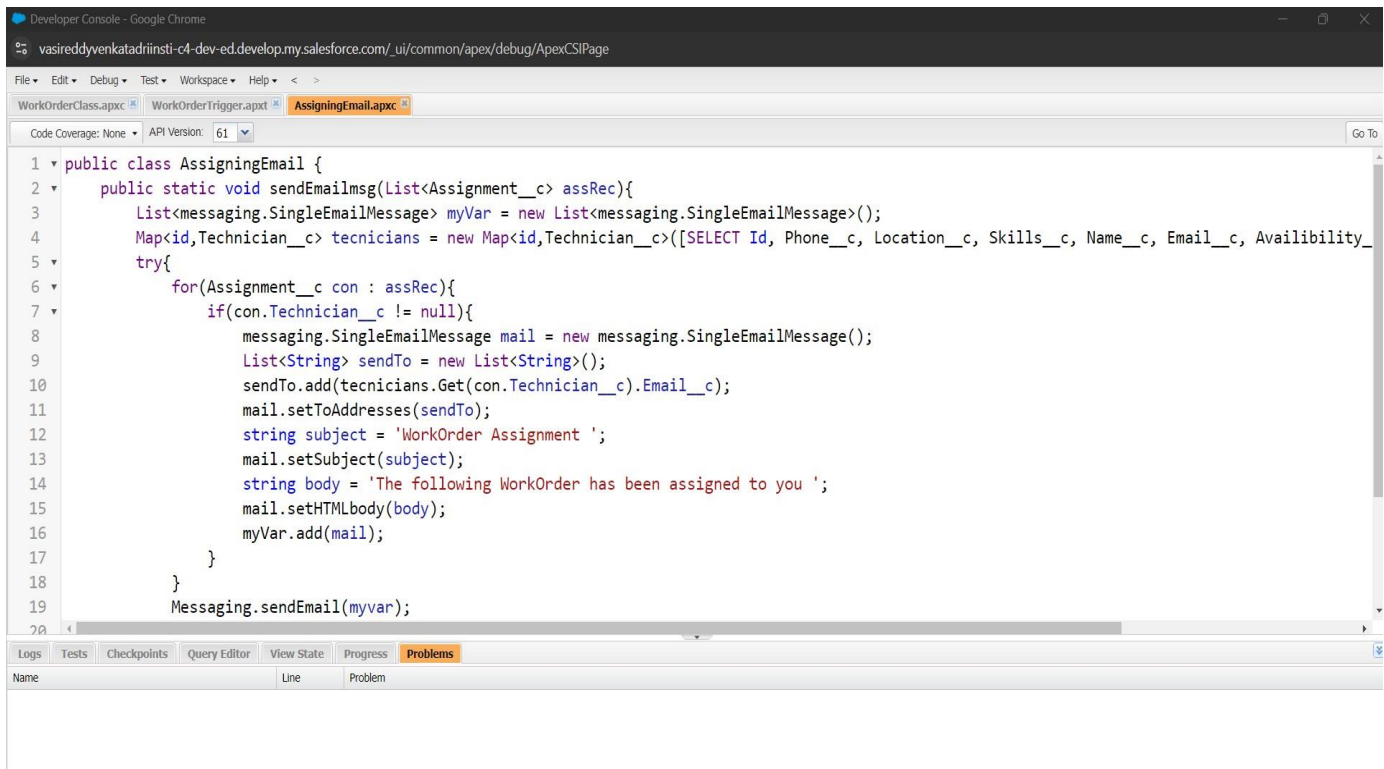
```
public class AssigningEmail {  
    public static void sendEmailmsg(List<Assignment_c> assRec){  
        List<messaging.SingleEmailMessage> myVar = new  
List<messaging.SingleEmailMessage>();  
        Map<id,Technician_c> technicians = new Map<id,Technician_c>([SELECT Id, Phone_c,Location_c,  
Skills_c, Name_c, Email_c, Availibility_c, Name FROM Technician_c]);  
        try{  
            for(Assignment_c con : assRec){  
                if(con.Technician_ID_c != null){  
                    messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();List<String>  
sendTo = new List<String>(); sendTo.add(technicians.Get(con.Technician_ID_c).Email_c);  
                    mail.setToAddresses(sendTo);  
                    string subject = 'WorkOrder Assignment '  
                    mail.setSubject(subject);  
                    string body = 'The following WorkOrder has been assigned to you '  
                    mail.setHTMLbody(body);  
                    myVar.add(mail);  
                }  
            }  
            Messaging.sendEmail(myvar);  
        }  
    }  
}
```

```

        catch(exception e){
            system.debug('Error ----- > ' + e.getMessage());
        }
    }
}

```

8. Save the code.(click on file --> Save)

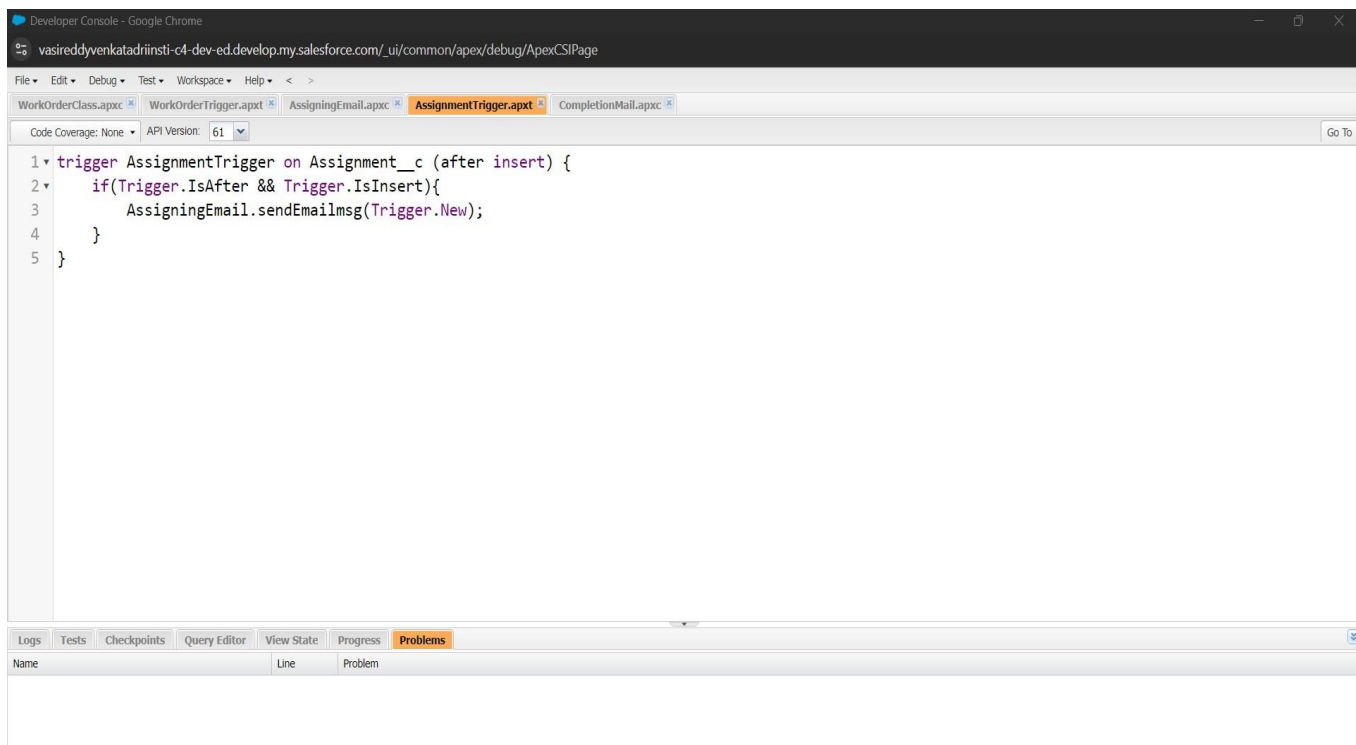


7.4 Create an Apex Trigger

To create a new Apex Class follow the below steps:

1. Click on the file --> New --> Apex Class.
2. Give the Apex Trigger name as “AssignmentTrigger”, and select “Assignment_c” from the dropdown for sObject.
3. Click Submit.
4. Now write the code logic here
5. **Source Code:**

```
trigger AssignmentTrigger on Assignment_c (after insert) {  
    if(!Trigger.IsAfter && Trigger.IsInsert){  
        AssigningEmail.sendEmailmsg(Trigger.New);  
    }  
}
```
6. Save the code.(click on file --> Save)



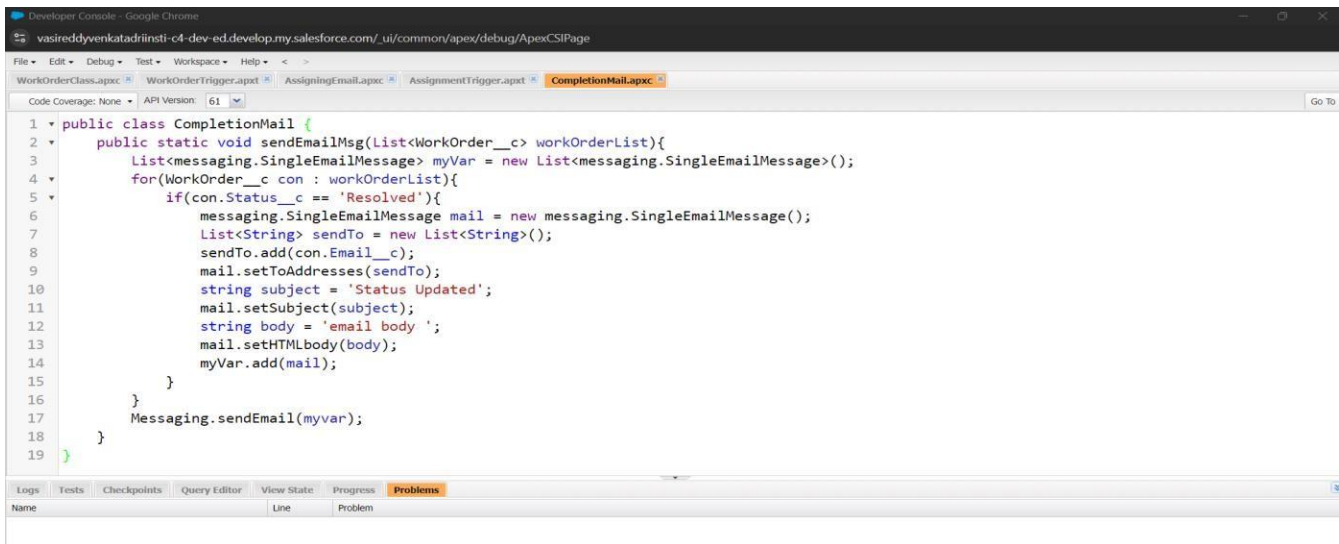
7.5 Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
4. Give the Apex Class name as “CompletionMail”.
5. Click ok.
6. Now write the code logic here

7. Source Code:

```
public class CompletionMail {  
    public static void sendEmailMsg(List<WorkOrder_c> workOrderList){  
        List<messaging.SingleEmailMessage> myVar = new  
List<messaging.SingleEmailMessage>(); for(WorkOrder_  
c con : workOrderList){  
    if(con.Status_c == 'Resolved'){  
        messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();List<String>  
        sendTo = new List<String>();  
        sendTo.add(con.Email_c);  
        mail.setToAddresses(sendTo); string  
        subject = 'Status Updated';  
        mail.setSubject(subject);  
        string body = 'email body ';  
        mail.setHTMLbody(body);  
        myVar.add(mail);  
    }  
}  
    Messaging.sendEmail(myvar);  
}
```

8. Save the code.(click on file --> Save)



7.6 Create an Apex Trigger

1. Click on the file --> Open.
2. A pop up window opens click on Triggers, then select “WorkOrderTrigger” and click on “Open”
3. Now write the code logic here.
4.

```
WorkOrderClass.workOrder(trigger.new);
```



```
    }
```



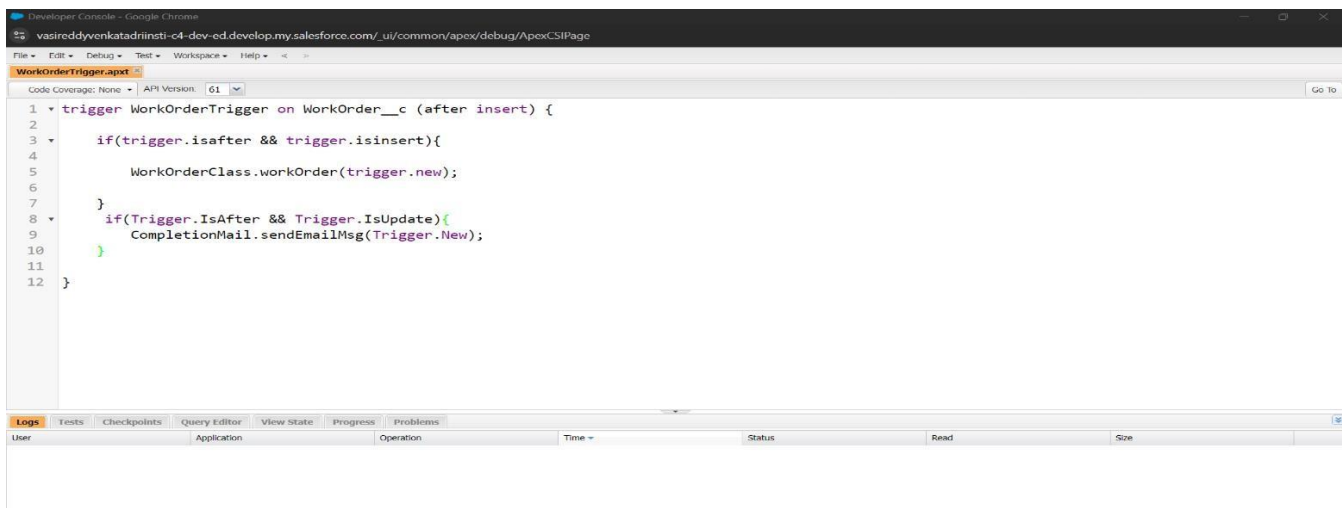
```
    if(trigger.IsAfter && trigger.IsUpdate){ CompletionMail.sendEmailMsg(trigger.New);
```



```
    }
```



```
}
```
5. Save the code.(click on file --> Save)



7.7 Create an Asynchronous Apex Class

Create an Apex Class to Delete all the WorkOrder records which meets the following criteriaL

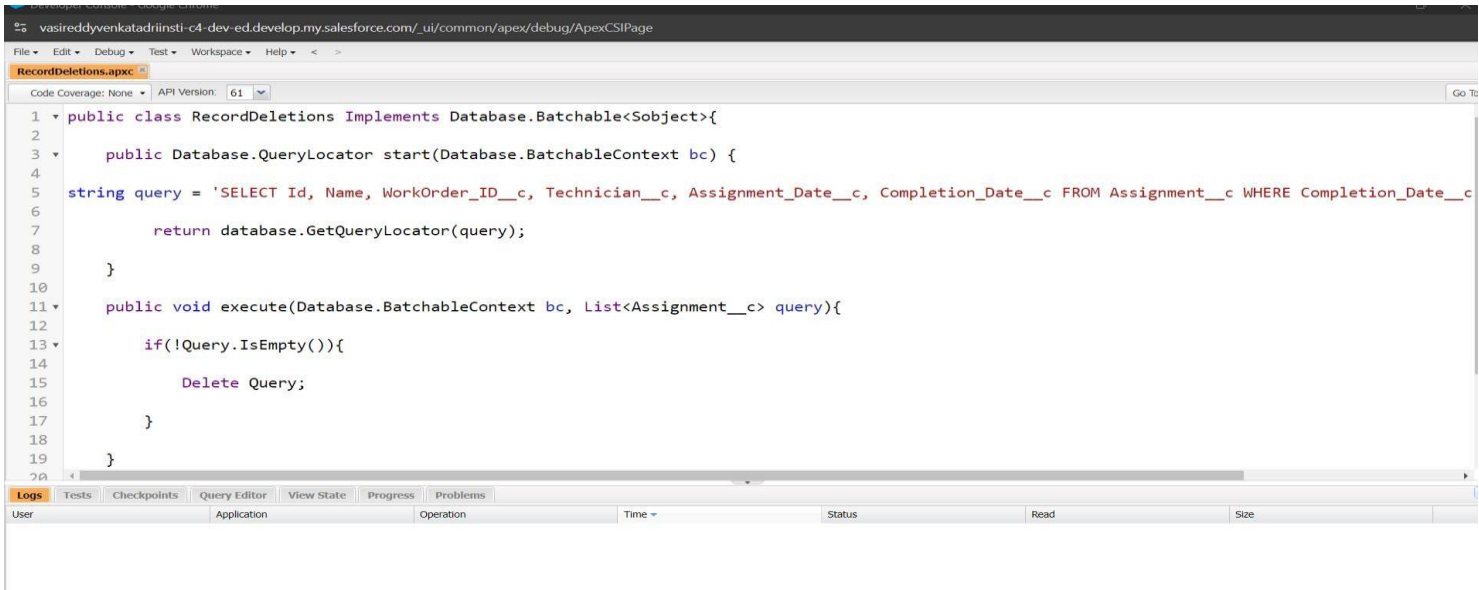
1. Completed date should be more than 30 days.
2. Status should be 'Resolved'.

Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
4. Give the Apex Class name as "RecordDeletion".
5. Click ok.
6. Now write the code logic here

```
public class RecordDeletions Implements Database.Batchable<Subject>{public
    Database.QueryLocator start(Database.BatchableContext bc) {
        string query = 'SELECT Id, Name, WorkOrder_ID_c, Technician_ID_c, Assignment_Date_
        c, Completion_Date_c FROM Assignment_c WHERE Completion_Date_c =
        LAST_N_DAYS:30';
        return database.GetQueryLocator(query);
    }
    public void execute(Database.BatchableContext bc, List<Assignment_c> query){
        if(!Query.IsEmpty()){
            Delete Query;
        }
    }
    public void finish(Database.BatchableContext bc){
    }
}
```

7. Save the code.(click on file --> Save)



7.8 Create an Apex Schedule Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
Click on the file --> New --> Apex Class.
4. Give the Apex Class name as “ScheduleClass”.
5. Click ok.
6. Now write the code logic here

Source Code:

```

global class ScheduleClass implements Schedulable {global
    void execute(SchedulableContext SC) {
        RecordDeletions delrec = new RecordDeletions();
        database.executeBatch(delrec, 200);
    }
}

```

7. Save the code.(click on file ? Save)

```

1 global class ScheduleClass implements Schedulable {
2     global void execute(SchedulableContext SC) {
3         RecordDeletions delrec = new RecordDeletions();
4         database.executeBatch(delrec, 200);
5     }
6 }

```

7.9 Create a Schedule Apex

Schedule the Apex class:

1. From the Setup page search for “Apex Classes” in quick search.
2. Click on “Schedule Apex” as shown below.
3. Click on Schedule Apex and enter the Job name.
4. Job Name : DeleteAssignmentSchedule
5. Apex Class : ScheduleClass (from clicking on lookup icon)
6. Frequency : Monthly
7. Preferred Start Time : Select any time
8. Click Save.

Apex Classes

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.09%
 You are currently using 5,217 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount of Apex Code used includes Apex Classes and Triggers defined in your organization.

[Estimate your organization's code coverage](#) ⓘ

[Compile all classes](#) ⓘ

View: All Create New View

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

							Developer Console	New	Generate from WSDL	Run All Tests	Schedule Apex
Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By					
Edit Del Security	AssigningEmail		62.0	Active	1,224	M.shiva sainath Reddy , 26/10/2024, 10:36 pm					
Edit Del Security	CompletionMail		62.0	Active	801	M.shiva sainath Reddy , 26/10/2024, 10:38 pm					
Edit Del Security	RecordDeletions		62.0	Active	593	M.shiva sainath Reddy , 26/10/2024, 10:47 pm					
Edit Del Security	ScheduleClass		62.0	Active	207	M.shiva sainath Reddy , 26/10/2024, 10:48 pm					
Edit Del Security	WorkOrderClass		62.0	Active	1,952	M.shiva sainath Reddy , 26/10/2024, 10:32 pm					

Dynamic Apex Classes

Dynamic Apex extends your programming reach by interacting with Lightning Platform components.

Task 8 :

8.1 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 --> click onstart report.
- 4. Customize your report
- 5. Add fields from left pane as shown below
- 6. Grouped by workorder ID
- 7. Save or run it.

Search...

★

+

🔍

Sales

Home

Opportunities

Leads

Tasks

Files

Accounts

Contacts

Campaigns

Reports

Reports

Recent

4 items

Search recent reports...

New Report

REPORTS	Report Name	Description	Folder	Created By	Created On	Subscribe
Recent	WorkOrders Status Reports		Private Reports	M.shiva sainath Reddy	27/10/2024, 7:04 am	
Created by Me	New Technician Report		Private Reports	M.shiva sainath Reddy	27/10/2024, 7:31 am	
Private Reports	Technician and Assignment Details Report		Private Reports	M.shiva sainath Reddy	27/10/2024, 7:08 am	
Public Reports	New Assignments with WorkOrder ID Report		Private Reports	M.shiva sainath Reddy	27/10/2024, 6:43 am	
All Reports						
FOLDERS						
All Folders						

8. **Note:** Reports may get varied from the above pictures as the data might be different.

Search...

★

+

🔍

Sales

Home

Opportunities

Leads

Tasks

Files

Accounts

Contacts

Campaigns

Dashboards

Reports

Chatter

Groups

Calendar

More

Report: Assignments with WorkOrder ID

New Assignments with WorkOrder ID Report

Enable Field Editing

Search

Add Chart

Filter

Refresh

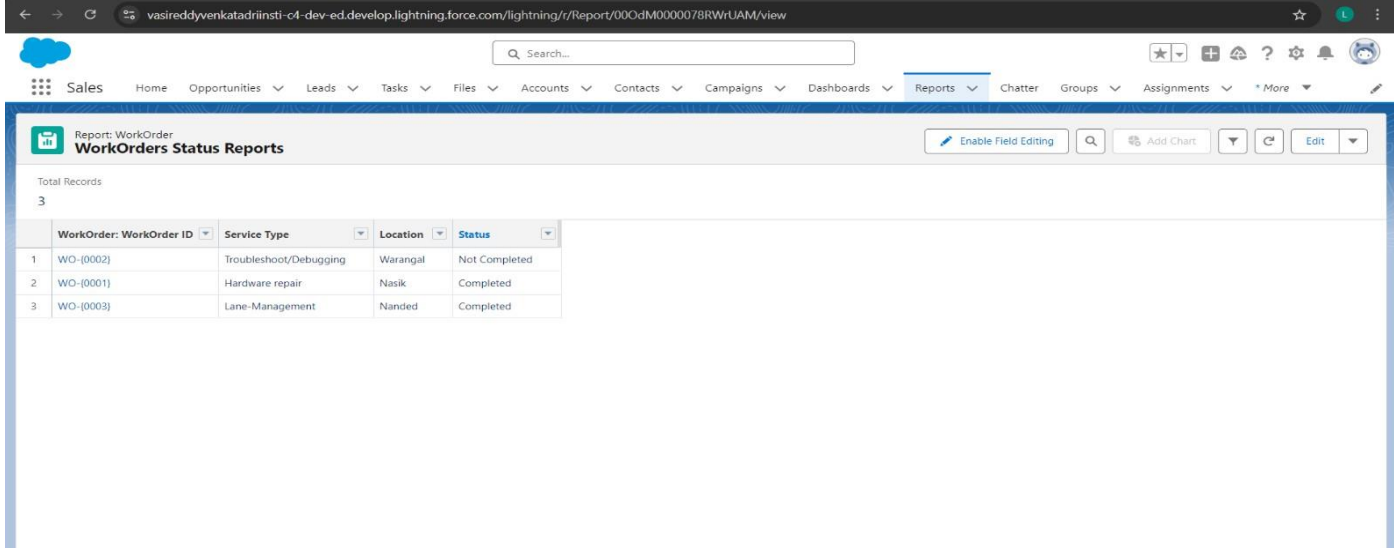
Edit

Total Records
5

	Assignment: Assignment ID	WorkOrder ID: WorkOrder ID
1	A-0002	WO-[0002]
2	A-0001	WO-[0001]
3	A-0003	WO-[0003]
4	A-0004	WO-[0004]
5	A-0005	WO-[0005]

8.2 Create Reports

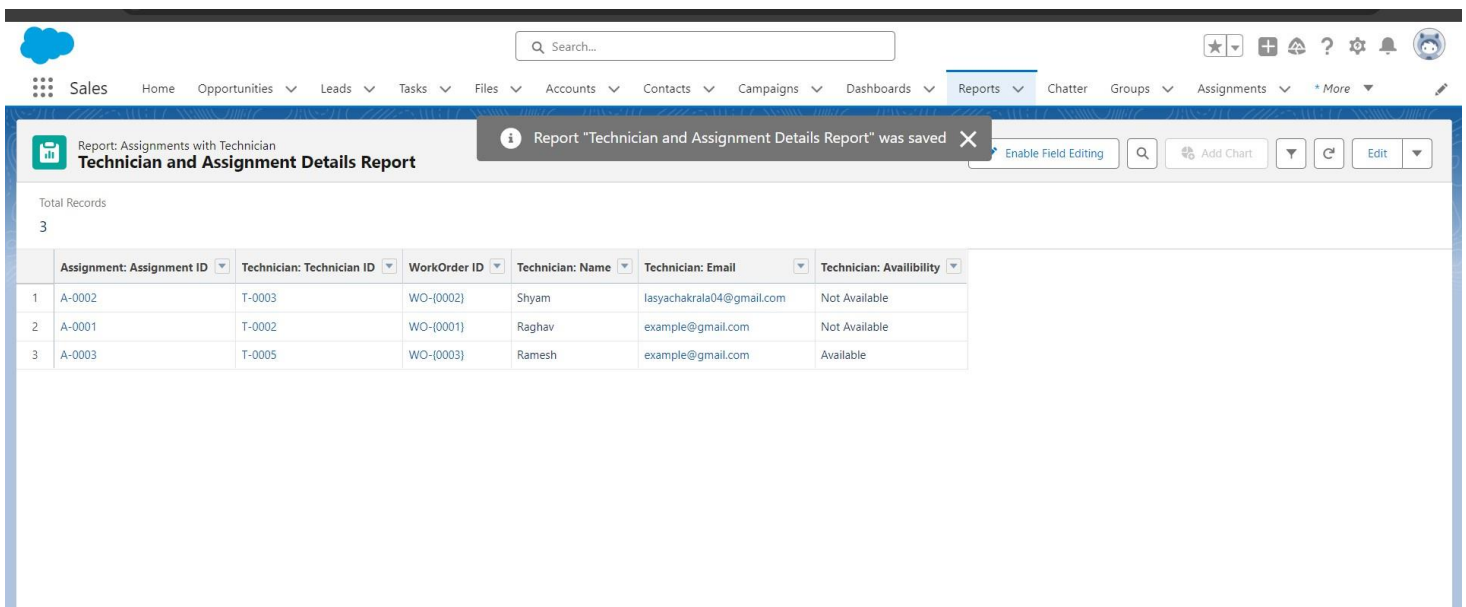
8.2.1 Create a report with report type: “WorkOrders Status Reports”.



The screenshot shows the Salesforce interface for a report titled "Report: WorkOrder WorkOrders Status Reports". The report displays a table with 3 total records. The table columns are: WorkOrder ID, Service Type, Location, and Status. The data rows are:

	WorkOrder: WorkOrder ID	Service Type	Location	Status
1	WO-[0002]	Troubleshoot/Debugging	Warangal	Not Completed
2	WO-[0001]	Hardware repair	Nasik	Completed
3	WO-[0003]	Lane-Management	Nanded	Completed

8.2.2 Create a report with report type: “Technician and Assignment Details Reports”.



The screenshot shows the Salesforce interface for a report titled "Report: Assignments with Technician Technician and Assignment Details Report". A notification banner at the top states "Report 'Technician and Assignment Details Report' was saved". The report displays a table with 3 total records. The table columns are: Assignment ID, Technician ID, WorkOrder ID, Technician Name, Technician Email, and Technician Availability. The data rows are:

	Assignment: Assignment ID	Technician: Technician ID	WorkOrder ID	Technician: Name	Technician: Email	Technician: Availability
1	A-0002	T-0003	WO-[0002]	Shyam	lasyachakrala04@gmail.com	Not Available
2	A-0001	T-0002	WO-[0001]	Raghav	example@gmail.com	Not Available
3	A-0003	T-0005	WO-[0003]	Ramesh	example@gmail.com	Available

8.3 Dashboard

8.3.1 Go to the app --> click on the Dashboards tabs.

8.3.2 Give a Name and click on Create.

8.3.3 Select add component.

8.3.4 Select a Report which we have created in the previous activities and click on select.

8.3.5 Click Add then click on Save and then click on Done.

The screenshot shows the Salesforce Lightning interface. The top navigation bar includes a search bar and various icons. The main menu on the left lists Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, and a user profile. The 'Dashboards' tab is selected, showing 'Dashboard 1'. A widget titled 'New Assignments with WorkOrder ID Report' is displayed, containing a table with 5 rows of assignment data.

Assignment: Assignment ID ↑	WorkOrder ID: WorkOrder ID
A-0001	WO-(0001)
A-0002	WO-(0002)
A-0003	WO-(0003)
A-0004	WO-(0004)
A-0005	WO-(0005)

Below the table is a link: 'View Report (New Assignments with WorkOrder ID Report)'.

8.4 Create Dashboards

Create another Dashboard as we discussed in activity 3 which shows the details of completed work order status in a vertical bar graph.

The screenshot shows the Salesforce Lightning interface with the 'Reports' tab selected. A report titled 'New WorkOrder Report' is displayed, showing a bar chart of record counts by service type and a table of the underlying data.

Bar Chart Data:

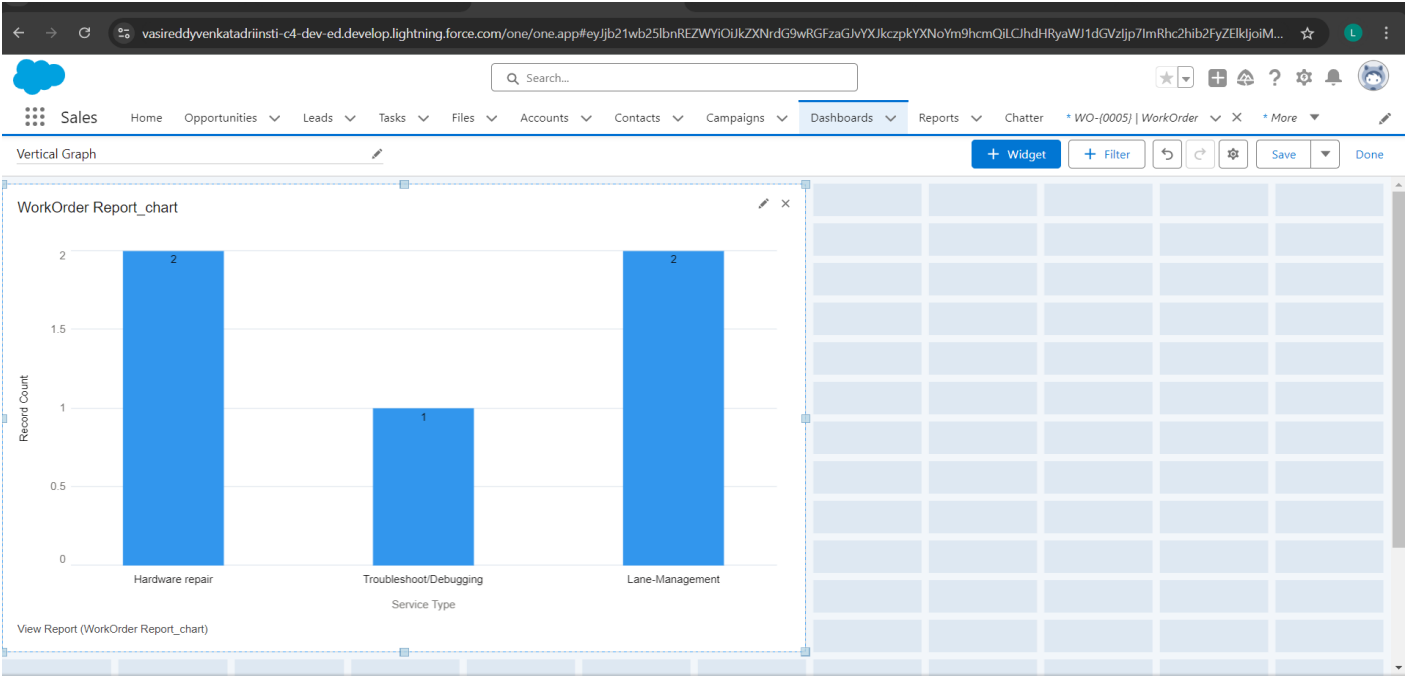
Service Type	Record Count
Hardware repair	2
Troubleshoot/Debugging	1
Lane-Management	2

Table Data:

Service Type ↑	WorkOrder: WorkOrder ID ↓	Email ↓	Location ↓
Hardware repair (2)	WO-(0001)	example@workorder.com	Nasik
	WO-(0004)	example3@workorder.com	Nasik
Subtotal			
Troubleshoot/Debugging (1)	WO-(0002)	example2@workorder.com	Warangal
Subtotal			
Lane-Management (2)	WO-(0003)	example1@workorder.com	Nanded
	WO-(0005)	example@workorder.com	Nanded

At the bottom, there are checkboxes for 'Row Counts', 'Detail Rows', 'Subtotals', and 'Grand Total', all of which are checked.

The above figure tells us about the report we used to create the following chart and dashboard.



This is the last task we created a dashboard based upon the reports of workorder and details.