

**FIELD SERVICE**  
**WORKORDER OPTIMIZATION**

**BY**

**JAJJARA SUBHASHINI**

**jajjarasubhashini123@gmail.com**

## **ABSTRACT**

**Objective:** 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.

### **Key Technologies:**

**1. Salesforce Field Service:** Salesforce's comprehensive field service management solution will be the core platform, providing tools for scheduling, dispatching, and real-time communication.

**2. Artificial Intelligence & Machine Learning:** Integrated with Salesforce, AI and ML algorithms will predict service demands, optimize scheduling, and match the right technician to the right job based on skill set, location, and availability.

**3. Predictive Analytics:** Leveraging Salesforce's analytics capabilities, predictive models will forecast service needs and preemptively address potential issues by analyzing historical data and current conditions.

**4. Internet of Things (IoT):** IoT devices will provide real-time data from field equipment, which will be integrated into Salesforce for proactive maintenance and swift response to issues.

### **Implementation Phases:**

**1. Salesforce Field Service:** Salesforce's comprehensive field service management solution will be the core platform, providing tools for scheduling, dispatching, and real-time communication.

**2. Artificial Intelligence & Machine Learning:** Integrated with Salesforce, AI and ML algorithms will predict service demands, optimize scheduling, and match the right technician to the right job based on skill set, location, and availability.

**3. Predictive Analytics:** Leveraging Salesforce's analytics capabilities, predictive models will forecast service needs and preemptively address potential issues by analyzing historical data and current conditions.

**4. Internet of Things (IoT):** IoT devices will provide real-time data from field equipment, which will be integrated into Salesforce for proactive maintenance and swift response to issues

**Potential Challenges:**

- 1. Data Integration:** Ensuring seamless integration of various data sources and legacy systems.
- 2. Change Management:** Managing the transition and ensuring buy-in from all stakeholders.
- 3. Scalability:** Ensuring the solution can scale to accommodate growth and increased demand.
- 4. Security and Privacy:** Protecting sensitive customer and operational data from breaches.

**Measurable Outcomes:**

1. Efficiency Metrics
2. Customer Satisfaction
3. Operational Excellence

**Functional Requirements:**

1. Work Order Management
2. Scheduling and Dispatching
3. Resource Management
4. Mobile Access
5. Customer Communication
6. Analytics and Reporting
7. Integration
8. User Management and Security
9. Maintenance and Support

By fulfilling these functional requirements, the Salesforce Field Service Work Order Optimization project will enhance the efficiency of field operations, improve customer satisfaction, and achieve overall business objectives.

## INDEX PAGE

SI No.	Module or Tasks Labels	Page No
1.	<b><u>Task 1: Object</u></b>  1.1 Create Technician Object  1.2 Create WorkOrder Object  1.3 Create Assignment Object	6-8
2.	<b><u>Task 2: Tabs</u></b>  2 Create a custom tab	8-9
3.	<b><u>Task 3: The Lightning App</u></b>  3 Create a Lightning App	10-11
4.	<b><u>Task 4: Fields &amp; Relationship</u></b>  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.	<b><u>Task 5: Profiles</u></b>  5 Technician Profile	16
6.	<b><u>Task 6: Users</u></b>  6 Create User	17

7.	<b><u>Task 7: Apex Trigger</u></b>  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.	<b><u>Task 8: Reports &amp; Dashboards</u></b>  8.1 Report  8.2 Create Reports  8.3 Dashboard  8.4 Create Dashboards	28-33

# 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 <sup>i</sup> Values Separated By Field Label Source \* Field Labels Row Import 5 rows of Data? <sup>i</sup> Record Name Field <sup>i</sup>

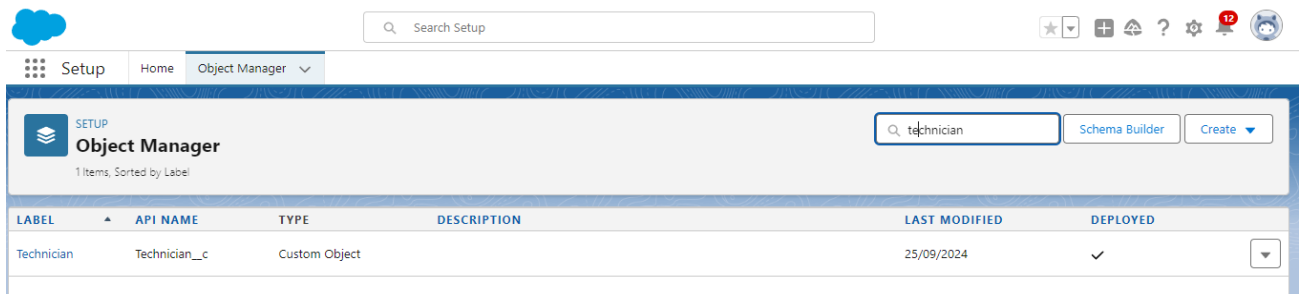
Unicode (UTF8) Comma ☐ Enter manually ☒ Detect from row 1 ☐ No, skip import ☒ Yes, import data Let Salesforce Create a Def: <sup>i</sup>

Fields 7 of 7 to import ☐ Hide mapped fields  

 IMPORT FILE FIELD NAME		SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS <sup>i</sup>	FIELD PREVIEW
✓ Technician ID	✕	Technician ID	Text	<input checked="" type="checkbox"/>	T-0001
✓ Name	✕	Name	Text	<input checked="" type="checkbox"/>	Raghu
✓ Phone	✕	Phone	Integer	<input checked="" type="checkbox"/>	7892341560



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



## 1.2 Create WorkOrder Object:

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

CSV File Details

Encoding Format:  Values Separated By:  Field Label Source: ☐ Enter manually ☒ Detect from row \* Field Labels Row:  Import 2 rows of Data?: ☒ No, skip import ☐ Yes, import data Record Name Field:

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	<input checked="" type="checkbox"/>	WO-(0001)
✓ Email	✗ Email	Email	<input checked="" type="checkbox"/>	example1@workorder.com
✓ Service Type	✗ Service Type	Picklist	<input checked="" type="checkbox"/>	Maintenance

Back Next

After creating the WorkOrder Custom object it looks like the below



### 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

The screenshot displays the Salesforce Object Manager interface. At the top, there is a navigation bar with 'Setup', 'Home', and 'Object Manager' tabs. Below this, the 'Object Manager' section is active, showing a search bar with 'assignment' entered, a 'Schema Builder' button, and a 'Create' dropdown. The main area contains a table with two rows of object information.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Assignment	Assignment__c	Custom Object		27/09/2024	✓
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 web content, 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", with "Tabs" selected. The main content area is titled "Custom Tabs" and includes a "Help for this Page" link. It contains a description of custom tabs and two sections: "Custom Object Tabs" and "Web Tabs".

**Custom Object Tabs**

Action	Label	Tab Style	Description
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Assignments</a>	Airplane	
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Technician</a>	Safe	
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">WorkOrder</a>	Safe	

**Web Tabs**

No Web 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

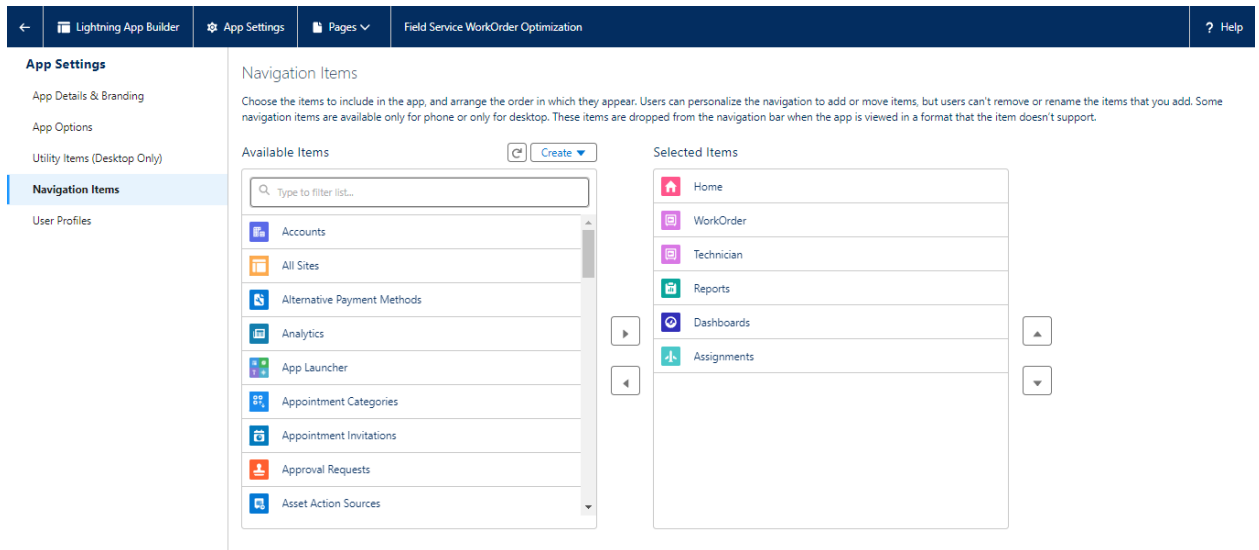
Image: optional (if you want to give any image you can, otherwise not mandatory)

Primary color hex value: keep this default

The screenshot shows the 'App Details & Branding' configuration page in the Lightning App Builder. The page is divided into two main sections: 'App Details' and 'App Branding'. The 'App Details' section includes fields for 'App Name' (Field Service WorkOrder Optimization), 'Developer Name' (Field\_Service\_WorkOrder\_Optimization), and 'Description' (Field Service WorkOrder Optimization maximizes efficiency by automating the...). The 'App Branding' section includes an 'Image' upload button, a 'Primary Color Hex Value' dropdown (set to #0070D2), and 'Org Theme Options' (with a checkbox to 'Use the app's image and color instead of the org's custom theme'). At the bottom, there is an 'App Launcher Preview' showing the app icon (FS) and the app name and description.

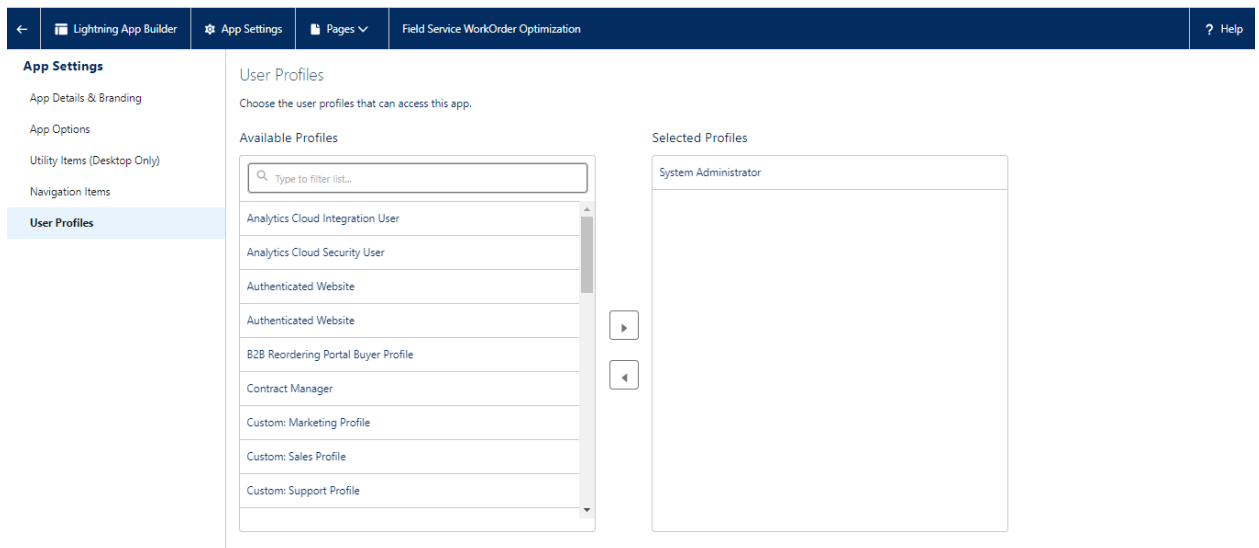
3. Then click Next --> (App option page) keep it as default --> Next --> (Utility Items) keep it as default --> Next

4. To Add Navigation Items



Search the items in the search bar (Home, WorkOrder, Technician, Assignment, Reports, and 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.

The screenshot shows the Salesforce Setup interface for creating a custom field. The breadcrumb trail is **Setup > OBJECT MANAGER > WorkOrder**. The left sidebar contains navigation links: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, and List View Button Layout. The main content area is titled **WorkOrder Custom Field Location** with a **Back to WorkOrder** link. It includes tabs for **Custom Field Definition Detail** (selected), **Validation Rules**, **Set Field-Level Security**, **View Field Accessibility**, and **Where is this used?**. The **Field Information** section contains the following details:

Field Label	Location	Object Name	WorkOrder		
Field Name	Location	Data Type	Picklist		
API Name	Location__c				
Description					
Help Text					
Data Owner					
Field Usage					
Data Sensitivity Level					
Compliance Categorization					
Created By	SUBHASHINI JAJJARA	25/09/2024, 7:39 pm	Modified By	SUBHASHINI JAJJARA	25/09/2024, 7:39 pm

The **General Options** section shows **Required** as ☐ and **Default Value** as .

### 4.2 Manage your picklist values

The screenshot shows the Salesforce Setup interface for managing picklist values. The breadcrumb trail is **Setup > OBJECT MANAGER > WorkOrder**. The left sidebar is identical to the previous screenshot. The main content area is titled **WorkOrder** and includes tabs for **Validation Rules** (selected) and **Values**. The **Validation Rules** section shows **No dependencies defined.** and **No validation rules defined.** with a **New** button. The **Values** section includes buttons for **New**, **Reorder**, **Replace**, **Printable View**, and **Chart Colors**. Below these is a table of picklist values:

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/>   Edit   Del   Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA 25/09/2024, 7:39 pm
<input type="checkbox"/>   Edit   Del   Deactivate	Nasik	Nasik	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA 25/09/2024, 8:35 pm
<input type="checkbox"/>   Edit   Del   Deactivate	Warangal	Warangal	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA 25/09/2024, 8:35 pm
<input type="checkbox"/>   Edit   Del   Deactivate	Nanded	Nanded	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA 25/09/2024, 8:35 pm
<input type="checkbox"/>   Edit   Del   Deactivate	Pune	Pune	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA 26/09/2024, 4:11 pm
<input type="checkbox"/>   Edit   Del   Deactivate	Hyderabad	Hyderabad	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA 26/09/2024, 4:11 pm

Below the table is the **Inactive Values** section with a **Delete Unused Values** button and the message **No Inactive Values values defined.** At the bottom, there is a **Back To Top** link and a note: **Always show me more records per related list**.

### 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

Field Dependencies: No dependencies defined.

Validation Rules: No validation rules defined.

Values: WorkOrder Custom Field: Priority ~ Salesforce - Developer Edition

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 7:39 pm
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	Low	Low	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 12:53 pm
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	Medium	Medium	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 12:53 pm
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	High	High	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 12:53 pm

Inactive Values: No inactive values defined.

Field Dependencies: No dependencies defined.

Validation Rules: No validation rules defined.

Values: WorkOrder Custom Field: Priority ~ Salesforce - Developer Edition

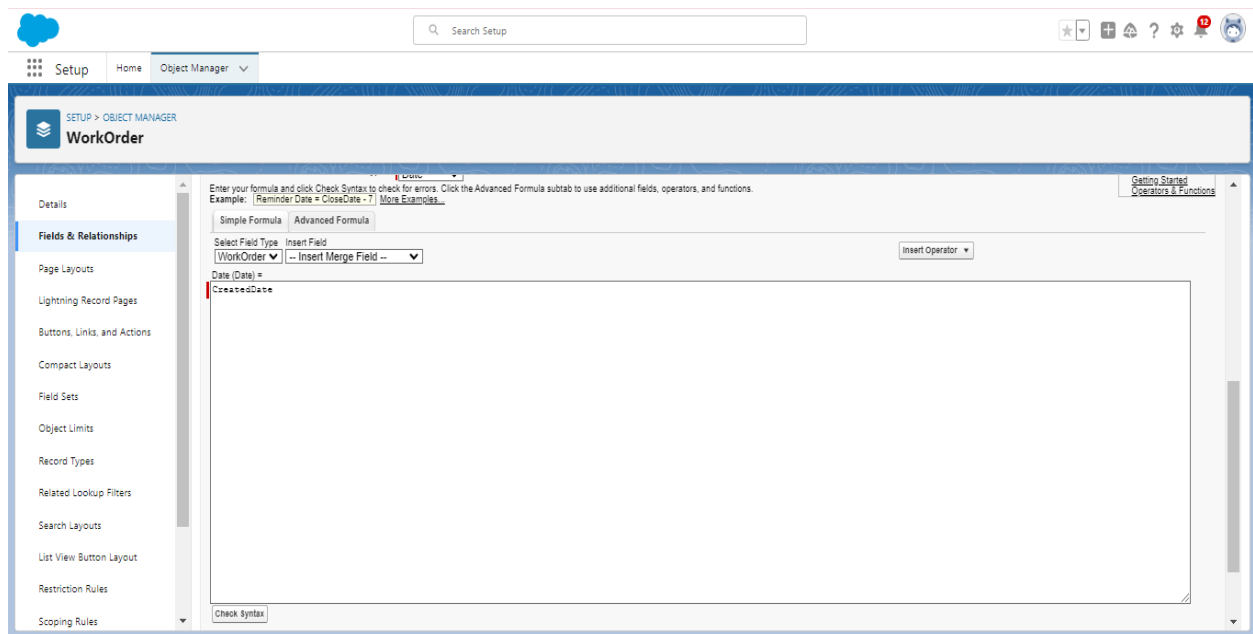
Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	Value1	Value1	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 7:39 pm
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	Machine Installation	Machine Installation	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 12:55 pm
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	Hardware Repair	Hardware Repair	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 12:55 pm
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	Troubleshoot/Debugging	Troubleshoot/Debugging	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 12:55 pm
<input type="checkbox"/> Edit   <input type="checkbox"/> Del   <input type="checkbox"/> Deactivate	Maintenance	Maintenance	<input type="checkbox"/>	Assigned dynamically	SUBHASHINI JAJJARA, 26/09/2024, 12:55 pm

Inactive Values: No inactive 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.










## 4.5 Creating Remaining fields for the respective objects


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

SL NO	Object Name	Field	
1	Assignment	Field Name	Data type
		<ul style="list-style-type: none"> <li>• Technician ID</li> <li>• Assignment Date</li> <li>• Completion Date</li> </ul>	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

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)		<input type="checkbox"/>
Assignment ID	Name	Auto Number		<input checked="" type="checkbox"/>
Completion Date	Completion_Date__c	Formula (Date)		<input type="checkbox"/>
Created By	CreatedById	Lookup(User)		<input type="checkbox"/>
Last Modified By	LastModifiedById	Lookup(User)		<input type="checkbox"/>
Owner	OwnerId	Lookup(User,Group)		<input checked="" type="checkbox"/>
Technician ID	Technician_ID__c	Lookup(Technician)		<input checked="" type="checkbox"/>
WorkOrder ID	WorkOrder_ID__c	Lookup(WorkOrder)		<input checked="" type="checkbox"/>

## 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.

**SETUP Profiles**

Object	Read	Create	Edit	Delete	View All	Modify All
Contact Point Addresses	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Consents	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Emails	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Custom Object Permissions**

Object	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Assignments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technician	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Session Settings**

Session Times Out After:  [i](#)

Session Security Level Required at Login:  [i](#)

**SETUP Profiles**

Invoice	<a href="#">View</a>
Invoice Line	<a href="#">View</a>
Lead	<a href="#">View</a>
Location	<a href="#">View</a>
Location Group	<a href="#">View</a>
Location Group Assignment	<a href="#">View</a>
Object Milestone	<a href="#">View</a>
Waitlist Participant	<a href="#">View</a>
Waitlist Service Resource	<a href="#">View</a>
Waitlist Work Type	<a href="#">View</a>
Web Store Inventory Source	<a href="#">View</a>
Work Type	<a href="#">View</a>
Work Type Group	<a href="#">View</a>
Work Type Group Member	<a href="#">View</a>
WorkOrder	<a href="#">View</a>
WorkOrder - Sheet1	<a href="#">View</a>

**Custom Field-Level Security**

Assignment	<a href="#">View</a>
Technician	<a href="#">View</a>



## 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](mailto: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 for creating a new user. The left sidebar contains a search bar with 'users' entered and a list of navigation options including Users, Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, Feature Settings, Data.com, and Prospectors. The main content area is titled 'User Elina Gilbert' and includes a 'User Detail' section with various fields and checkboxes.

User Detail	
Name	Elina Gilbert
Alias	elina
Email	jaiarasubheshini23@gmail.com <a href="#">Verify</a>
Username	jaiarasubheshini23@gmail.com
Nickname	elina <a href="#">i</a>
Title	
Company	
Department	
Division	
Address	4-25,Kotha Rajapet,Chilakaluripet,Guntur, Andhra Pradesh 522020 ANDAMAN AND NICOBAR ISLAND India
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Locale	English (India)
Language	English
Delegated Approver	
Manager	
Receive Approval Request Emails	Only if I am an approver
Role	Salesforce Platform
User License	Salesforce Platform
Profile	Technician
Active	<input checked="" type="checkbox"/>
Marketing User	<input type="checkbox"/>
Offline User	<input type="checkbox"/>
Knowledge User	<input type="checkbox"/>
Flow User	<input type="checkbox"/>
Service Cloud User	<input type="checkbox"/>
Sales.com Contributor User	<input type="checkbox"/>
Site.com Publisher User	<input type="checkbox"/>
WDC User	<input type="checkbox"/>
Mobile Push Registrations	<a href="#">View</a>
Data.com User Type	<a href="#">i</a>
Accessibility Mode (Classic Only)	<input type="checkbox"/> <a href="#">i</a>
Debug Mode	<input type="checkbox"/> <a href="#">i</a>

## 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 newListWorkOrder){  
  
        Map<Integer, List> maptotech = new map<Integer,List>();  
  
        integer num = 0;  
  
        List properWo = new List();  
  
        List lstAssignment = new List();  
  
        List technicianToAssignment = new List();  
  
        for(WorkOrder__c iter : newListWorkOrder){  
  
            List lststring = new List();  
  
            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 techId = new Map();

Map allTechnician = new Map([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 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);

    IstAssignment.add(A);

}

If(!IstAssignment.isEmpty()){

    insert IstAssignment;

}

}

}

```

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



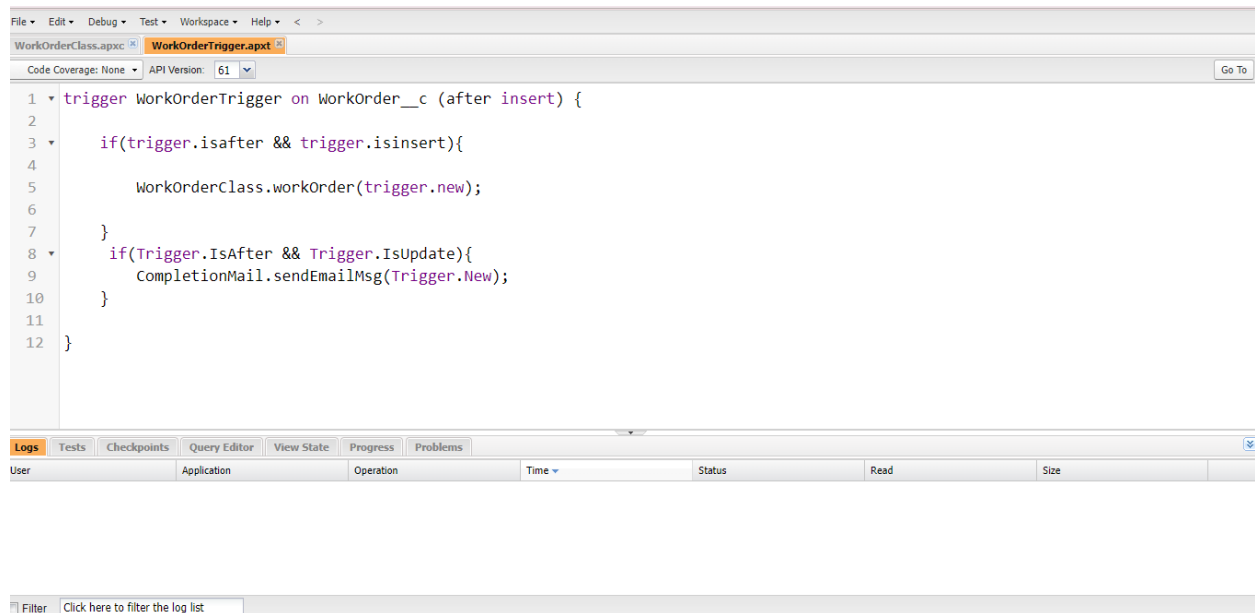
## 7.2 Create an Apex Trigger

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 assRec){

        List<messaging. SingleEmailMessage> myVar = new List<messaging. SingleEmailMessage>();

        Map technicians = new Map([SELECT Id, Phone__c,
Location__c, Skills__c, Name__c, Email__c, Availability__c, Name FROM Technician__c]);

        try{

            for(Assignment__c con : assRec){

```

```

        if(con.Technician_ID__c != null){
            messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
            List sendTo = new List();
            sendTo.add(tecnicos.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){

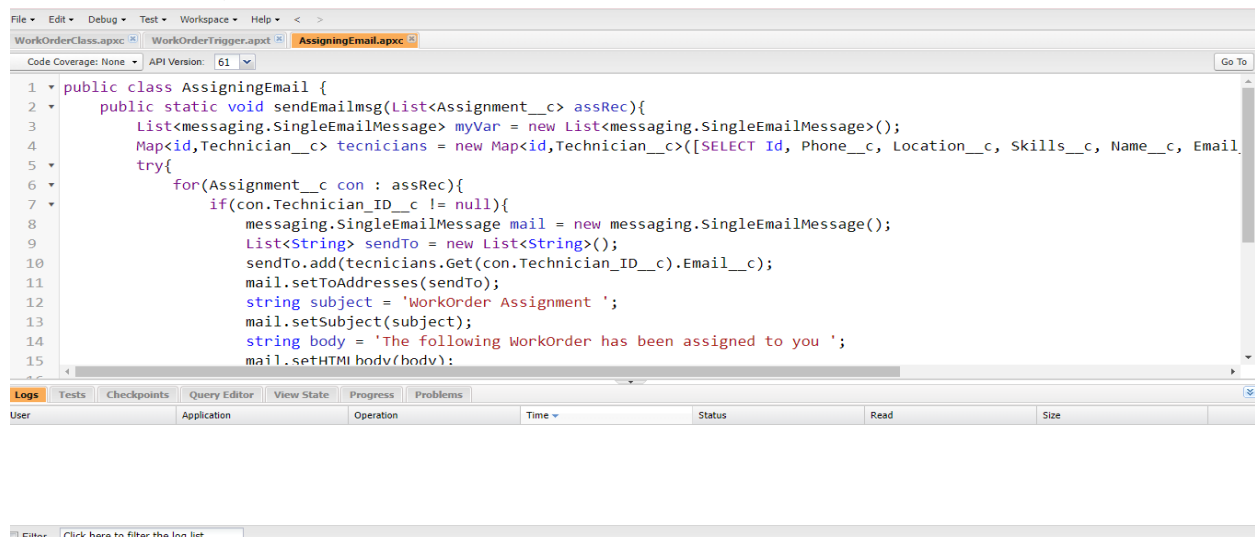
    system.debug('Error -----> ' + e.getMessage());

}

}

```

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



## 7.4 Create an Apex Trigger

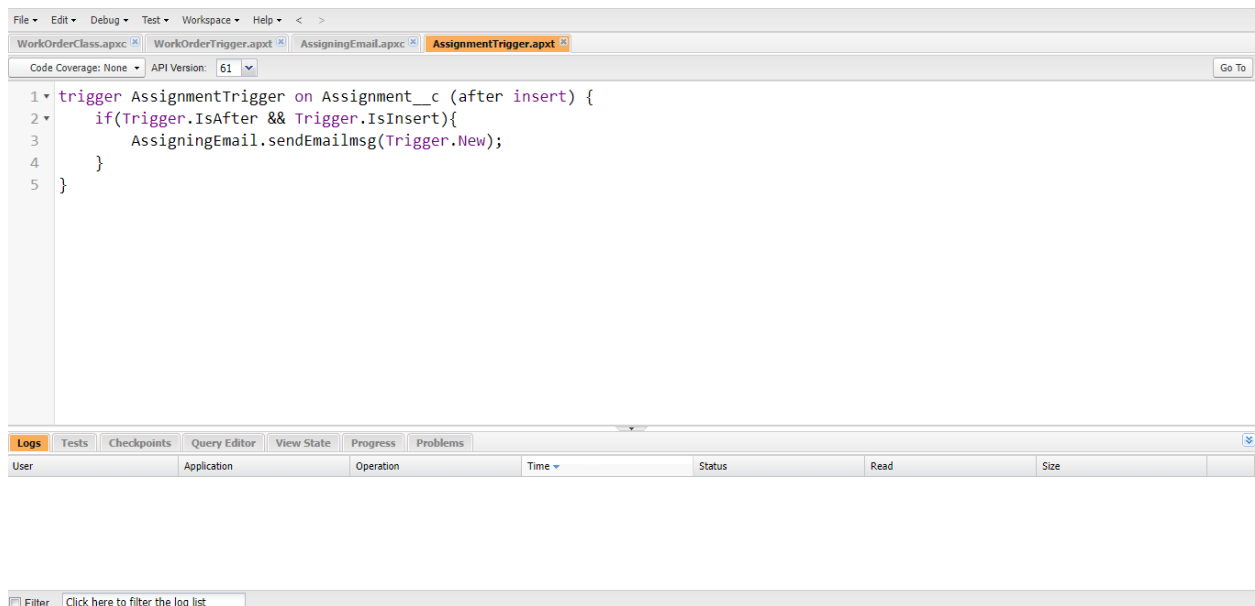
To create a new Apex Class follow the below steps:

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 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 sendTo = new List();  
  
                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);  
    }  
}
```



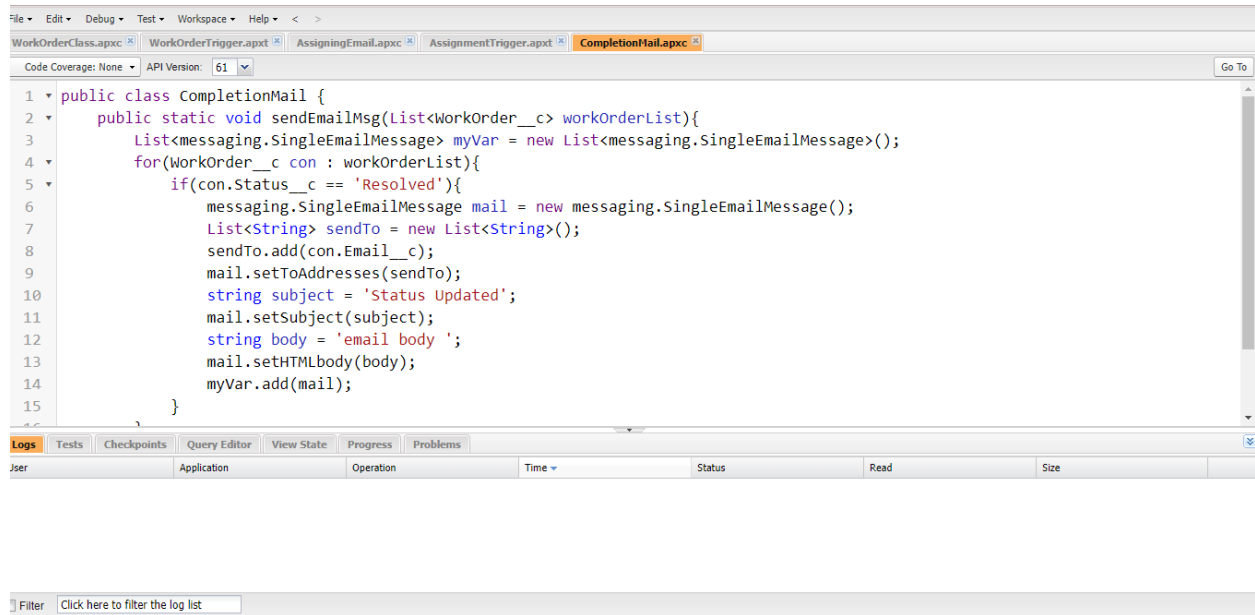
```

    }

}

```

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); }

```

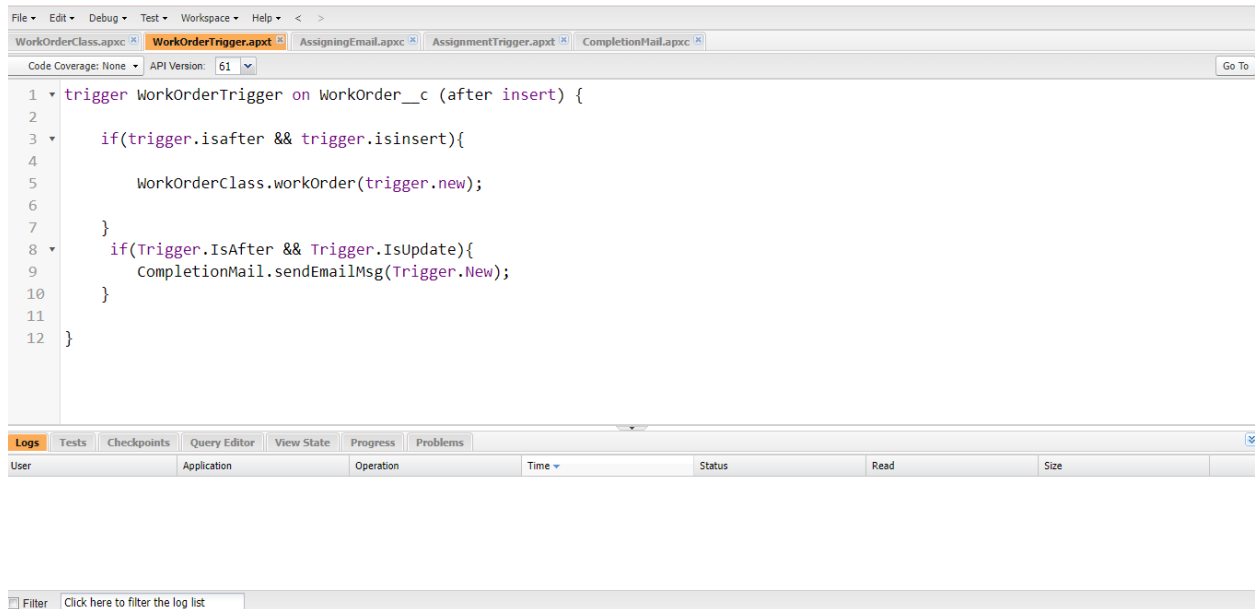
if(Trigger.IsAfter && Trigger.IsUpdate){

    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 criteria

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{
    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 query){

        if(!Query.IsEmpty()){

            Delete Query;

        }

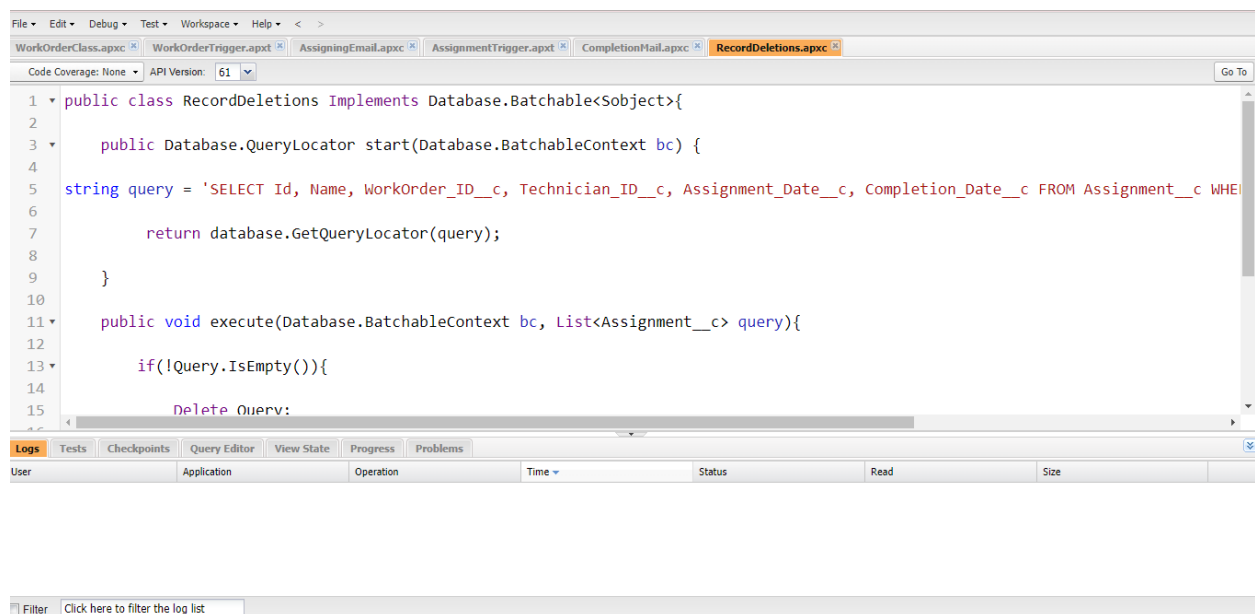
    }

    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.

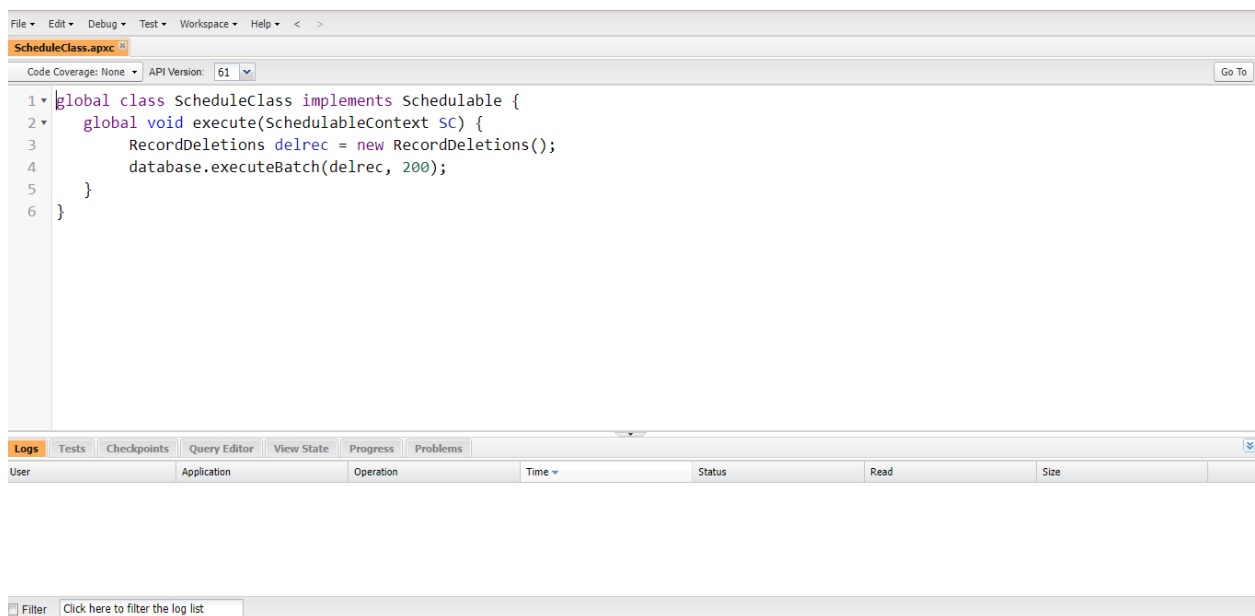
4. Give the Apex Class name as “ScheduleClass”.

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)



**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 5,004 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 8,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage

Compile all classes

View: All

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flag
Edit   Del   Security	AssigningEmail		61.0	Active	1,228	SUBHASHINI JAJUARA 20/09/2024, 2:10 pm	<input type="checkbox"/>
Edit   Del   Security	CompletionMail		61.0	Active	801	SUBHASHINI JAJUARA 20/09/2024, 2:13 pm	<input type="checkbox"/>
Edit   Del   Security	RecordDeletions		61.0	Active	593	SUBHASHINI JAJUARA 20/09/2024, 2:15 pm	<input type="checkbox"/>
Edit   Del   Security	ScheduleClass		61.0	Active	207	SUBHASHINI JAJUARA 20/09/2024, 2:28 pm	<input type="checkbox"/>
Edit   Del   Security	WorkOrderClass		61.0	Active	1,954	SUBHASHINI JAJUARA 20/09/2024, 2:04 pm	<input type="checkbox"/>

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 on start report.

4. Customize your report

5. Add fields from left pane as shown below

6. Grouped by workorder ID

7. Save or run it.

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

Sales

Home

Opportunities

Leads

Tasks

Files

Accounts

Contacts

Campaigns

Dashboards

Reports

Chatter

More

Q Search...

★

+

🔍

?

⚙️

12

👤

Reports

Created by Me

Q Search reports created by me...

New Report

New Folder

⚙️

REPORTS

Recent

Created by Me

Private Reports

Public Reports

All Reports

FOLDERS

All Folders

Created by Me

Shared with Me

Report Name	Description	Folder	Created By	Created On	Subscribed
New Assignments Report		Private Reports	SUBHASHINI JAJJARA	28/9/2024, 2:43 pm	<div>▼</div>
New Assignments Report		Private Reports	SUBHASHINI JAJJARA	28/9/2024, 2:44 pm	<div>▼</div>

To Do List

Sales

Home

Opportunities

Leads

Tasks

Files

Accounts

Contacts

Campaigns

Dashboards

Reports

Chatter

More

Q Search...

★

+

🔍

?

⚙️

12

👤

Report: Assignments

New Assignments Report

Enable Field Editing

Q

Add Chart

▼

↺

Edit

▼

Total Records  
2

	Assignment: Assignment ID	Technician ID
1	A-0001	-
2	A-0002	T-0001

To Do List

Sales

Home

Opportunities

Leads

Tasks

Files

Accounts

Contacts

Campaigns

Dashboards

Reports

Chatter

More

Q Search...

★

+

🔍

?

⚙️

12

👤

Report: Assignments

New Assignments Report

Enable Field Editing

Q

Add Chart

▼

↺

Edit

▼

Total Records  
2

	Assignment: Assignment ID	WorkOrder ID
1	A-0001	WO-[0001]
2	A-0002	WO-[0002]

To Do List

## 8.2 Create Reports

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

The screenshot shows the Salesforce Reports interface. At the top, there is a navigation bar with the Salesforce logo, a search bar, and various utility icons. Below the navigation bar, the 'Reports' tab is selected. The main content area displays a report titled 'Report: WorkOrder' and 'New WorkOrder Report'. The report shows 'Total Records' as 2. Below this, there is a table with two columns: 'WorkOrder: WorkOrder ID' and 'Status'. The table contains two rows of data.

	WorkOrder: WorkOrder ID	Status
1	WO-[0001]	Submitted
2	WO-[0002]	Resolved

At the bottom left of the report area, there is a 'To Do List' icon.

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

The screenshot shows the Salesforce Reports interface. At the top, there is a navigation bar with the Salesforce logo, a search bar, and various utility icons. Below the navigation bar, the 'Reports' tab is selected. The main content area displays a report titled 'Report: Assignments' and 'New Assignments Report'. The report shows 'Total Records' as 2. Below this, there is a table with two columns: 'Assignment: Assignment ID' and 'Technician ID'. The table contains two rows of data.

	Assignment: Assignment ID	Technician ID
1	A-0002	T-0001
2	A-0001	-

At the bottom left of the report area, there is a 'To Do List' icon.

### 8.3 Dashboard

1. Go to the app --> click on the Dashboards tabs.
2. Give a Name and click on Create.
3. Select add component.
4. Select a Report which we have created in the previous activities and click on select.
5. Click Add then click on Save and then click on Done.

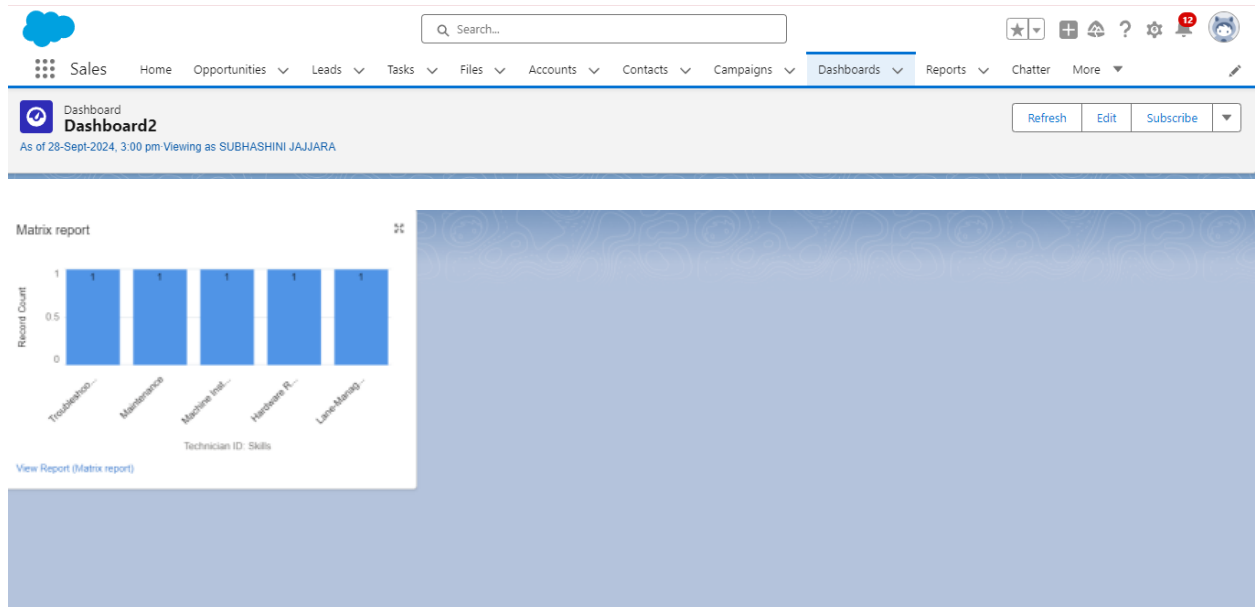
The screenshot displays the Salesforce user interface. At the top, there is a navigation bar with a search bar and various utility icons. Below this is a tabbed menu with options like Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, and More. The 'Dashboards' tab is currently selected. The main content area shows a dashboard titled 'Dashboard1' with a subtitle 'As of 28-Sept-2024, 2:57 pm Viewing as SUBHASHINI JAJJARA'. On the right side of the dashboard, there are buttons for 'Refresh', 'Edit', 'Subscribe', and a dropdown arrow. On the left side, a component titled 'New Assignments Report' is visible, containing a table with two columns: 'Assignment: Assignment ID ↑' and 'Technician ID'. The table has two rows of data. Below the table, there is a link that says 'View Report (New Assignments Report)'. At the bottom left of the dashboard, there is a 'To Do List' icon.

Assignment: Assignment ID ↑	Technician ID
A-0001	-
A-0002	T-0001



## 8.4 Create Dashboards

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



*Thank you*