

FIELD SERVICE WORK ORDER OPTIMIZATION

By Kovuri Sai Tejaswini

21bq1a05b6@vvit.net

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 s

INDEX PAGE

SI No.	Module or Tasks Labels	Page No.
1	<u>Task 1: Object</u> 1. Create Technician Object 2. Create Work Order Object 3. 1.3 Create Assignment Object	5-7
2	<u>Task 2: Tabs</u> 1. Create a custom tab	8
3	<u>Task 3: The Lightning App</u> 1. Create a Lightning App	9-10
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 Work Order Object	11-14
5	<u>Task 5: Profiles</u> 1. Technician Profile	16

6	<u>Task 6: Users</u> <ol style="list-style-type: none"> 1. Create User 	17
7	<u>Task 7: Apex Trigger</u> <ol style="list-style-type: none"> 1. Create an Apex Class 2. Create an Apex Trigger 3. Create an Apex Class 4. Create an Apex Trigger 5. Create an Apex Class 6. Create an Apex Trigger 7. Create an Asynchronous Apex Class 8. Create an Apex Schedule Class 9. Create an Schedule apex 	18-28
8	<u>Task 8: Reports & Dashboards</u> <ol style="list-style-type: none"> 1. Report 2. Create Reports 3. Dashboard 4. Create Dashboards 	29-31

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 2 rows of Data? [?]
☒ No, skip import
☐ Yes, import data

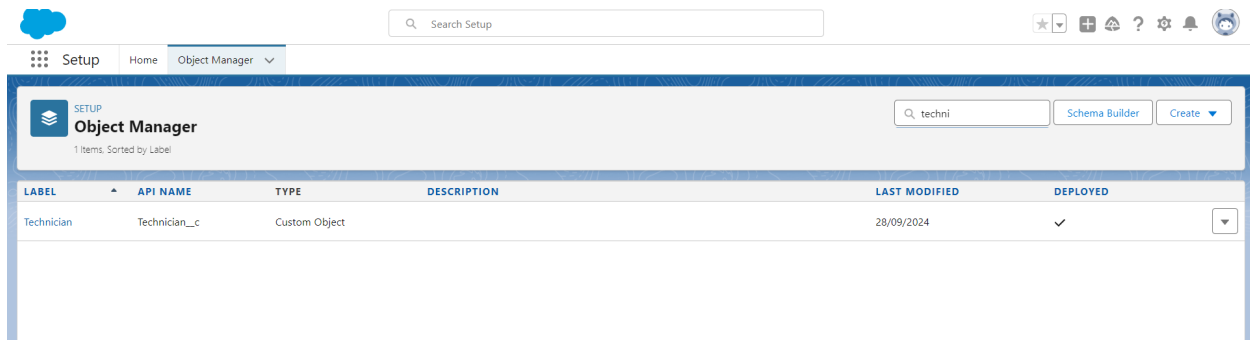
Record Name Field [?]
WorkOrder ID

Fields 7 of 7 to import ☐ Hide mapped fields

IMPORT FILE FIELD NAME		SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS [?]	FIELD PREVIEW
✓ Service Type	×	Service Type	Text	✓	Maintenance
✓ Description	×	Description	Text Area	✓	
✓ Location	×	Location	Text	✓	Pune
✓ Priority	×	Priority	Text	✓	
✓ Status	×	Status	Text	✓	

Review the following errors
• One or more fields aren't mapped properly.
Fix those rows or remove them.

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.

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 FIELD FILE NAME	SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
Service Type	Service Type	Text	✓	Maintenance
Description	Description	Text Area	✓	
Location	Location	Text	✓	Pune
Priority	Priority	Text	✓	
Status	Status	Text	✓	

Review the following errors

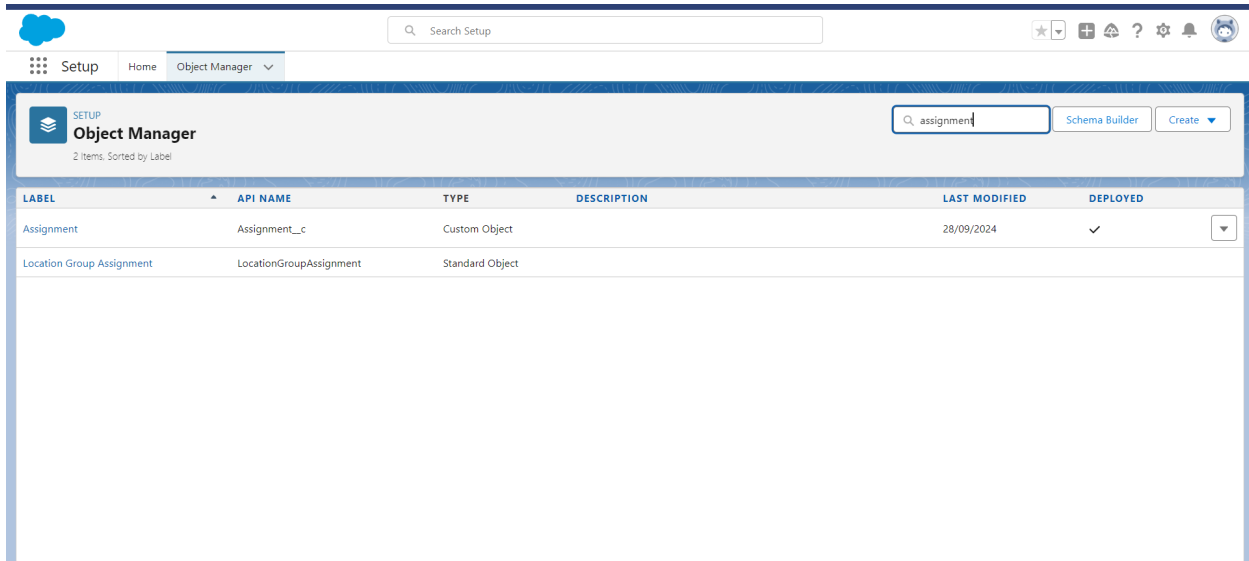
- One or more fields aren't mapped properly. Fix those rows or remove them.

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' (selected). A search bar labeled 'Search Setup' is also present. Below the navigation bar, the 'Object Manager' section is active, showing a search bar with 'assignment' entered, a 'Schema Builder' button, and a 'Create' button. The main content area displays a table of objects:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Assignment	Assignment__c	Custom Object		28/09/2024	✓
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 left sidebar has a search bar with 'tabs' entered. Under 'User Interface', 'Rename Tabs and Labels' is selected, and 'Tabs' is highlighted. The main content area is titled 'Custom Tabs' and includes a 'New' button and a 'What Is This?' link. Below this is a table of Custom Object Tabs. The table has columns for Action, Label, Tab Style, and Description. The rows are: Assignments (Castle), Providers (Bank), Resources (Castle), Technicians (Box), and WorkOrder (Box). Below the table, there are sections for 'Web Tabs' (No Web Tabs have been defined) and 'Visualforce Tabs'.

Action	Label	Tab Style	Description
Edit Del	Assignments	Castle	
Edit Del	Providers	Bank	
Edit Del	Resources	Castle	
Edit Del	Technicians	Box	
Edit Del	WorkOrder	Box	

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 sub-items: 'App Details & Branding' (selected), 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles'. The main content area is titled 'App Details & Branding' and includes instructions: 'Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.' It is divided into two columns: 'App Details' and 'App Branding'. 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 'Field Service WorkOrder Optimization'). Under 'App Branding', there is an 'Image' upload area with an 'Upload' button, a 'Primary Color Hex Value' dropdown set to '#0070D2', and an 'Org Theme Options' checkbox (unchecked) with the label 'Use the app's image and color instead of the org's custom theme'. At the bottom, an 'App Launcher Preview' shows a blue square icon with 'FS' and the app name 'Field Service WorkOrder O...'.

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

App Settings

App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Account Brands
Accounts
All Sites
Alternative Payment Methods
Analytics
App Launcher
Appointment Categories
Appointment Invitations
Approval Requests
Asset Action Sources

Selected Items

Home
Technicians
Assignments
Reports
Dashboards

Search the items in the search bar(Home, WorkOrder, Technician, Assignment, Reports, Dashboard) from the search 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.

←

Lightning App Builder

App Settings

Pages

Field Service WorkOrder Optimization

Help

App Settings

App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles

User Profiles

Choose the user profiles that can access this app.

Available Profiles

Analytics Cloud Integration User
Analytics Cloud Security User
Authenticated Website
Authenticated Website
B2B Reordering Portal Buyer Profile
Contract Manager
Custom: Marketing Profile
Custom: Sales Profile
Custom: Support Profile
Customer Community Login User

Selected Profiles

System Administrator

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 the 'Assignment' custom field. 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, List View Button Layout, and Restriction Rules. The main content area is titled 'Assignment Custom Field WorkOrder ID' and includes a 'Back to Assignment' link. Below the title are tabs for 'Custom Field Definition Detail' (selected), 'Validation Rules', and 'Where is this used?'. The 'Custom Field Definition Detail' tab shows the following information:

Field Information	
Field Label	WorkOrder ID
Field Name	WorkOrder_ID
API Name	WorkOrder_ID__c
Description	
Help Text	
Data Owner	
Field Usage	
Data Sensitivity Level	
Compliance Categorization	
Created By	SAITEJASWINI KOVURI
Modified By	SAITEJASWINI KOVURI

Below the field information, there are 'Lookup Options':

Lookup Options	
Related To	WorkOrder
Related List Label	Assignments
Required	<input type="checkbox"/>
What to do if the lookup record is deleted?	Clear the value of this field.

4.2 Manage your picklist values

4.3 Manage your picklist values : Add following values to the respective fields in WorkOrder object:

The screenshot shows the Salesforce Setup interface for the 'WorkOrder' object. 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, List View Button Layout, and Restriction Rules. The main content area is titled 'WorkOrder' and includes a 'Back to WorkOrder' link. Below the title are tabs for 'Picklist Options' (selected), 'Field Dependencies', 'Validation Rules', and 'Values'. The 'Picklist Options' tab shows the following information:

Picklist Options	
Restrict picklist to the values defined in the value set	<input type="checkbox"/>
Controlling Field	[None]

Below the picklist options, there are 'Picklist Values Used':

Picklist Values Used	
Active picklist values	5 (1,000 max)
Inactive picklist values	1 (4,000 max)

Below the picklist values, there are 'Field Dependencies' and 'Validation Rules' sections, both showing 'No dependencies defined' and 'No validation rules defined' respectively.

Below the validation rules, there are 'Values' section with a table of picklist values:

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit Del Deactivate	Pune	Pune	<input type="checkbox"/>	Assigned dynamically	SAITEJASWINI KOVURI
<input type="checkbox"/> Edit Del Deactivate	Hyderabad	Hyderabad	<input type="checkbox"/>	Assigned dynamically	SAITEJASWINI KOVURI
<input type="checkbox"/> Edit Del Deactivate	Nasik	Nasik	<input type="checkbox"/>	Assigned dynamically	SAITEJASWINI KOVURI

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

Home

Object Manager

Search Setup

Star

Share

Help

Settings

Notifications

Profile

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 picklist values 3 (1,000 max)

Inactive picklist values 1 (4,000 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

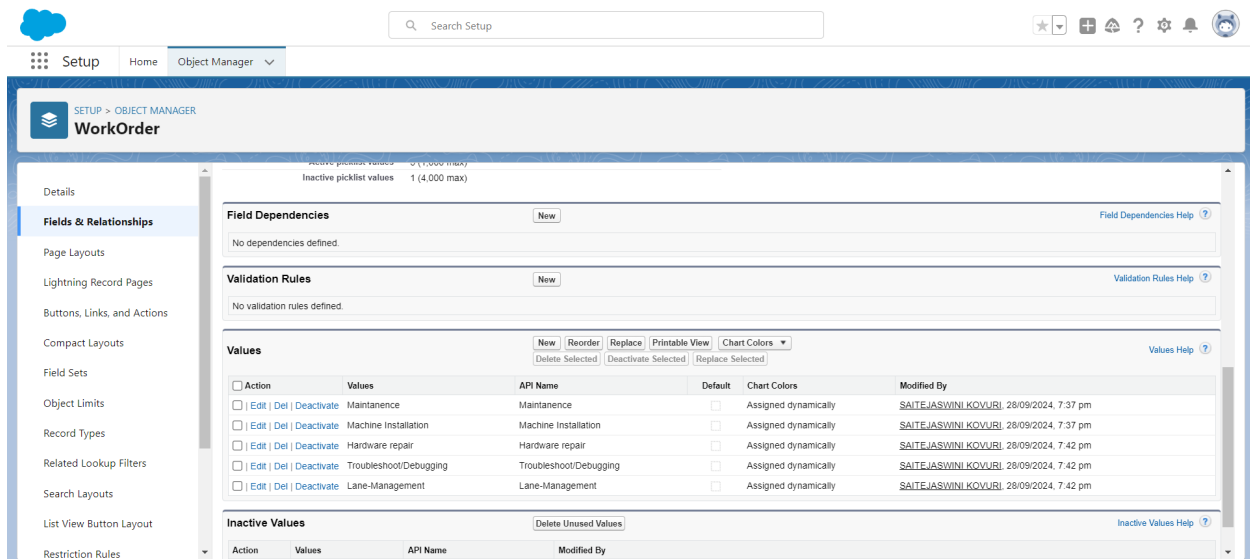
Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit Del Deactivate	Low	Low	<input type="checkbox"/>	Assigned dynamically	SAITEJASWINI KOVURI 28/09/2024, 7:37 pm
<input type="checkbox"/> Edit Del Deactivate	Medium	Medium	<input type="checkbox"/>	Assigned dynamically	SAITEJASWINI KOVURI 28/09/2024, 7:37 pm
<input type="checkbox"/> Edit Del Deactivate	High	High	<input type="checkbox"/>	Assigned dynamically	SAITEJASWINI KOVURI 28/09/2024, 7:40 pm

Inactive Values

Delete Unused Values

Inactive Values Help

Action	Values	API Name	Modified By
--------	--------	----------	-------------



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	Object	Field	
		Field Name	Datatype
1	Assignment	<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.LastModifie dDate , NULL)
--	--	--	--



Search Setup



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

Fields & Relationships

8 Items, Sorted by Field Label

Q Quick Find

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.

 **SETUP**
Profiles

Profiles [Help for this Page](#)

All Profiles [Edit](#) | [Delete](#) | [Create New View](#)

New Profile 

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#) | [Other](#) | [A](#)

<input type="checkbox"/> Action	Profile Name ↑	User License	Custom
<input type="checkbox"/> Edit Clone	Analytics Cloud Integration User	Analytics Cloud Integration User	<input type="checkbox"/>
<input type="checkbox"/> Edit Clone	Analytics Cloud Security User	Analytics Cloud Integration User	<input type="checkbox"/>
<input type="checkbox"/> Edit Clone	Authenticated Website	Authenticated Website	<input type="checkbox"/>
<input type="checkbox"/> Edit Clone	Authenticated Website	Authenticated Website	<input type="checkbox"/>
<input type="checkbox"/> Edit Delete ...	R2R Reordering Portal Ruxer Profile	External Apps oain	<input checked="" type="checkbox"/>

Clone Profile

Enter the name of the new profile.

You must select an existing profile to clone from.

Existing Profile

Standard Platform User

User License

Salesforce Platform

Profile Name

Save

Cancel

Custom Object Permissions

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Assets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Services	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assignments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Billings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bookings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candidates	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Child object	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Crews	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customer Orders	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employees	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employment Websites	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Flights	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Jewel Customers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Job Applications	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Lead Scoring Rules	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Leaves	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Parent object 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Parent object 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Passengers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Positions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prices	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Projects	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ProjectTasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reviews	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sessions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Students	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
StudentSessions	<input 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"/>
Trainers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
WorkOrder	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Custom Field-Level Security

Asset	[View]	Lead Scoring Rule	[View]
Asset Service	[View]	Leave	[View]
Assignment	[View]	Parent object 1	[View]
Billing	[View]	Parent object 2	[View]
Booking	[View]	Passenger	[View]
Candidate	[View]	Position	[View]
Child object	[View]	Price	[View]
Crew	[View]	Project	[View]
Customer Order	[View]	ProjectTask	[View]
Employee	[View]	Review	[View]
Employment Website	[View]	Sessions	[View]
Flight	[View]	Student	[View]
Item	[View]	StudentSession	[View]
Jewel Customer	[View]	Technician	[View]
Job Application	[View]	Trainer	[View]
Knowledge	[View]	WorkOrder	[View]

WorkOrder Field-Level Security for profile Technician

[Help for](#)

Save Cancel			
Field Name	Field Type	Read Access	Edit Access
Created By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Date	Date	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Description	Long Text Area	<input type="checkbox"/>	<input type="checkbox"/>
Email	Email	<input type="checkbox"/>	<input type="checkbox"/>
Last Modified By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Location	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Owner	Lookup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Priority	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Service Type	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Status	Picklist	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
WorkOrder ID	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Save Cancel			

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 a search bar with 'users' entered and a list of navigation items: Users, Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, Users (highlighted), Feature Settings, Data.com, and Prospector Users. The main content area is titled 'All Users' and includes a description: 'On this page you can create, view, and manage users. To get more licenses, use the Your Account app. Let's Go'. Below this is a table of users with columns for Action, Full Name, Alias, Username, Role, Active, and Profile. The table lists six users: Chatter Expert, Gilbert Elina, Jane Jane, KOUUBI SAITEJASWINI, User Integration, and User Security. Each user has an 'Edit' link and a checkbox. The table also includes buttons for 'New User', 'Reset Password(s)', and 'Add Multiple Users'.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatter_00ddm0000050zruau.ehimwfb3link@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit	Gilbert Elina	eglb	sali@co.in		✓	Technician
<input type="checkbox"/> Edit	Jane Jane	jgrev	jane_grav.fygmimmoaim.0f8huu2fucdf.055gm2rauc07@co.in		✓	Customer Community User
<input type="checkbox"/> Edit	KOUUBI SAITEJASWINI	SKOUU	vinav@co.in	SF Admin	✓	System Administrator
<input type="checkbox"/> Edit	User Integration	integ	integration@00ddm0000050zruau.com		✓	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User Security	sec	insightssecurity@00ddm0000050zruau.com		✓	Analytics Cloud Security User

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)



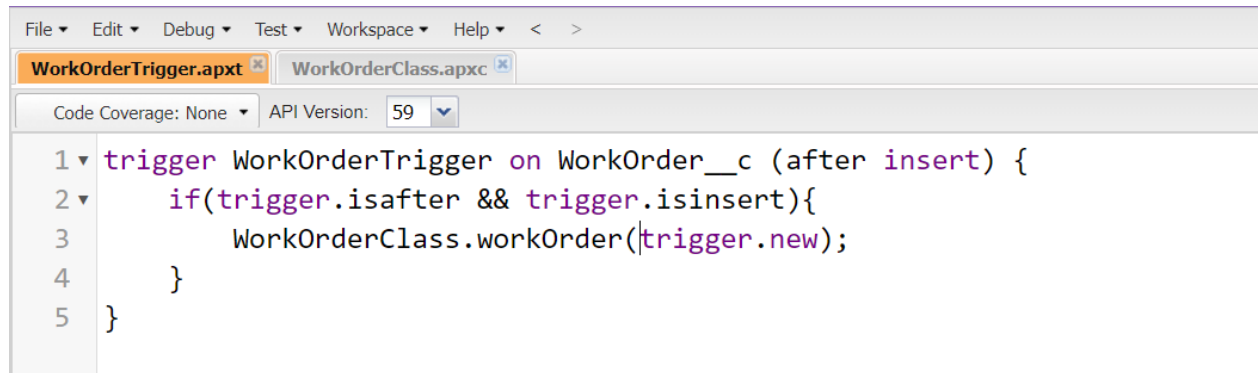
7.2 Create an Apex Trigger

1. To create new Apex Class follow the below a 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. steps
4. Now write the code logic here Source

```

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

```



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

```

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, 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<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());
}
}
}

```



7.4 Create an Apex Trigger

To create a new Apex Class follow the below steps:

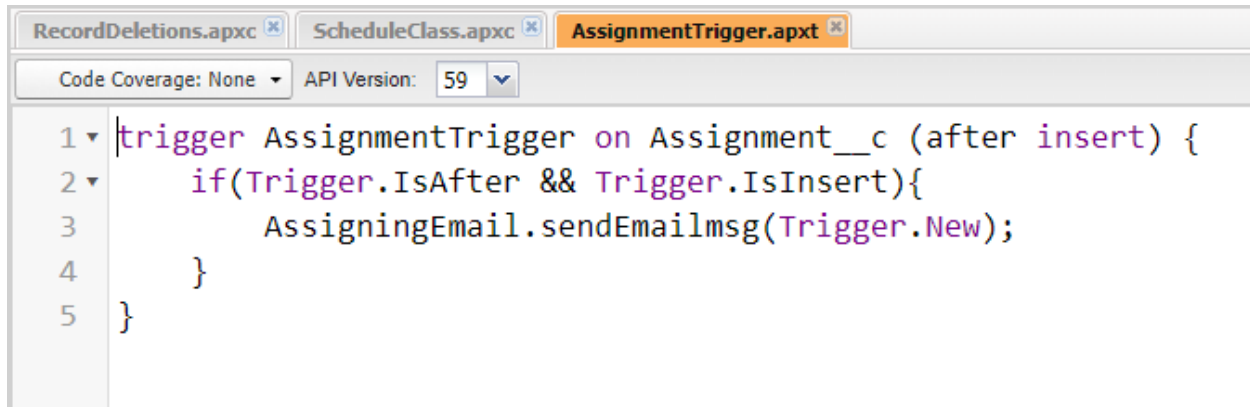
1. Click on the file --> New --> Apex Class.
2. Give the Apex Trigger name as "AssignmentTrigger", and select "Assignment c" from the dropdown for sObject.
3. Click Submit.
4. Now write the code logic here

Source Code:

```

trigger AssignmentTrigger on Assignment__c (after insert) {
    if(Trigger.IsAfter && Trigger.IsInsert){
        AssigningEmail.sendEmailmsg(Trigger.New);
    }
}

```



```
1 trigger AssignmentTrigger on Assignment__c (after insert) {
2     if(Trigger.IsAfter && Trigger.IsInsert){
3         AssigningEmail.sendEmailmsg(Trigger.New);
4     }
5 }
```

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

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);
    }
}

```

```

1 public class CompletionMail {
2     public static void sendEmailMsg(List<WorkOrder__c> workOrderList){
3         List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
4         for(WorkOrder__c con : workOrderList){
5             if(con.Status__c == 'Resolved'){
6                 messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
7                 List<String> sendTo = new List<String>();
8                 sendTo.add(con.Email__c);
9                 mail.setToAddresses(sendTo);
10                string subject = 'Status Updated';
11                mail.setSubject(subject);
12                string body = 'email body ';
13                mail.setHTMLbody(body);
14                myVar.add(mail);
15            }
16        }
17        Messaging.sendEmail(myvar);
18    }
19 }

```

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.

```

1 trigger WorkOrderTrigger on WorkOrder__c (after insert, after update) {
2     if(Trigger.IsAfter && Trigger.IsInsert){
3         WorkOrderClass.workOrder(trigger.new);
4     }
5     if(Trigger.IsAfter && Trigger.IsUpdate){
6         CompletionMail.sendEmailMsg(trigger.New);
7     }
8 }

```

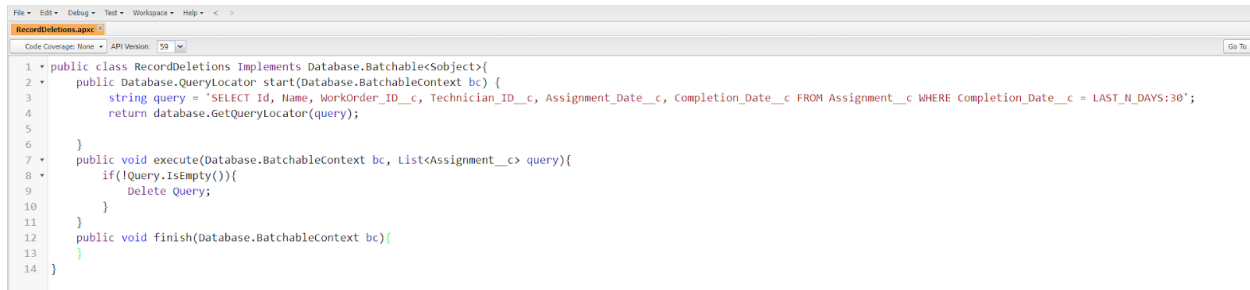
7.7 Create an Asynchronous Apex Class

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



```
1 public class RecordDeletions implements Database.Batchable<Object>{
2     public Database.QueryLocator start(Database.BatchableContext bc) {
3         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';
4         return database.GetQueryLocator(query);
5     }
6 }
7 public void execute(Database.BatchableContext bc, List<Assignment__c> query){
8     if(!query.isEmpty()){
9         Delete query;
10    }
11 }
12 public void finish(Database.BatchableContext bc){
13 }
14 }
```

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);
    }
}
```

```
RecordDeletions.apxc x ScheduleClass.apxt x
Code Coverage: None API Version: 59
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.

Search Setup

Setup Home Object Manager

apex

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: 0.36%
You are currently using 21,850 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 in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage
Compile all classes
View: All Create New View

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit	ApexClass_Test	thsecurity	56.0	Active	4,059	SAITEJASWINI KOVURI 16/06/2024, 1:55 pm	<input type="checkbox"/>
Edit Security	ApexClassModel	thsecurity	56.0	Active	1,422	SAITEJASWINI KOVURI 16/06/2024, 1:55 pm	<input type="checkbox"/>
Edit Security	ApexClassRestResource	thsecurity	56.0	Active	842	SAITEJASWINI KOVURI 16/06/2024, 1:55 pm	<input type="checkbox"/>
Edit Security	ApexClassService	thsecurity	56.0	Active	1,737	SAITEJASWINI KOVURI 16/06/2024, 1:55 pm	<input type="checkbox"/>
Edit Security	APIRequest	thsecurity	56.0	Active	2,568	SAITEJASWINI KOVURI 16/06/2024, 1:55 pm	<input type="checkbox"/>
Edit	APIRequest_Test	thsecurity	56.0	Active	2,274	SAITEJASWINI KOVURI 16/06/2024, 1:55 pm	<input type="checkbox"/>
		thsecurity	56.0	Active	68	SAITEJASWINI KOVURI 16/06/2024, 1:55 pm	<input type="checkbox"/>

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 Save or run it.

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

The screenshot shows the 'Reports' section of the Hive App. The top navigation bar includes the Hive App logo and a search bar. The main menu shows 'Reports' as the active tab. The 'Recent' section displays a single report titled 'Technician and Assignment Details' created by SAITEJASWINI KOVURI on 29/9/2024 at 12:04 am. The left sidebar lists various report categories and folders.

The screenshot shows the 'Technician and Assignment Details' report in the Hive App. The report displays a table with 2 records. The columns are: Assignment ID, Technician ID, Assignment Date, Technician ID: Email, Technician ID: Name, and Technician ID: Skills. The data rows show two assignments: one for Raghu (Machine Installation) and one for Raghav (Hardware repair).

	Assignment: Assignment ID	Technician ID: Technician ID	Assignment Date	Technician ID: Email	Technician ID: Name	Technician ID: Skills
1	A-0001	T-0001	28/09/2024	example1@gmail.com	Raghu	Machine Installation
2	A-0002	T-0002	28/09/2024	example2@gmail.com	Raghav	Hardware repair

8.2 Create Reports

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

The screenshot shows the Hive App interface with the 'WorkOrder Status report' selected. The report displays 4 total records in a table with columns: WorkOrder, Status, Email, and Location. The data is as follows:

	WorkOrder: WorkOrder	Status	Email	Location
1	a06dM000004VXTf	Submitted	example1@workorder.com	Pune
2	a06dM000004VXTg	Resolved	example2@workorder.com	Hyderabad
3	WorkOrder1	Submitted	example11@gmail.com	Pune
4	WorkOrder2	Resolved	example12@gmail.com	Hyderabad

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

8.3 Dashboard

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

Give a Name and click on Create

Select add component.

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

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

The screenshot shows the Hive App interface with the 'Technician and Assignment Details' report selected. The report displays 2 total records in a table with columns: Assignment, Technician ID, Assignment Date, Technician ID: Email, Technician ID: Name, and Technician ID: Skills. The data is as follows:

	Assignment: Assignment ID	Technician ID: Technician ID	Assignment Date	Technician ID: Email	Technician ID: Name	Technician ID: Skills
1	A-0001	T-0001	28/09/2024	example1@gmail.com	Raghu	Machine Installation
2	A-0002	T-0002	28/09/2024	example2@gmail.com	Raghav	Hardware repair

8.3 Dashboard

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

Give a Name and click on Create. Select add component.

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

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

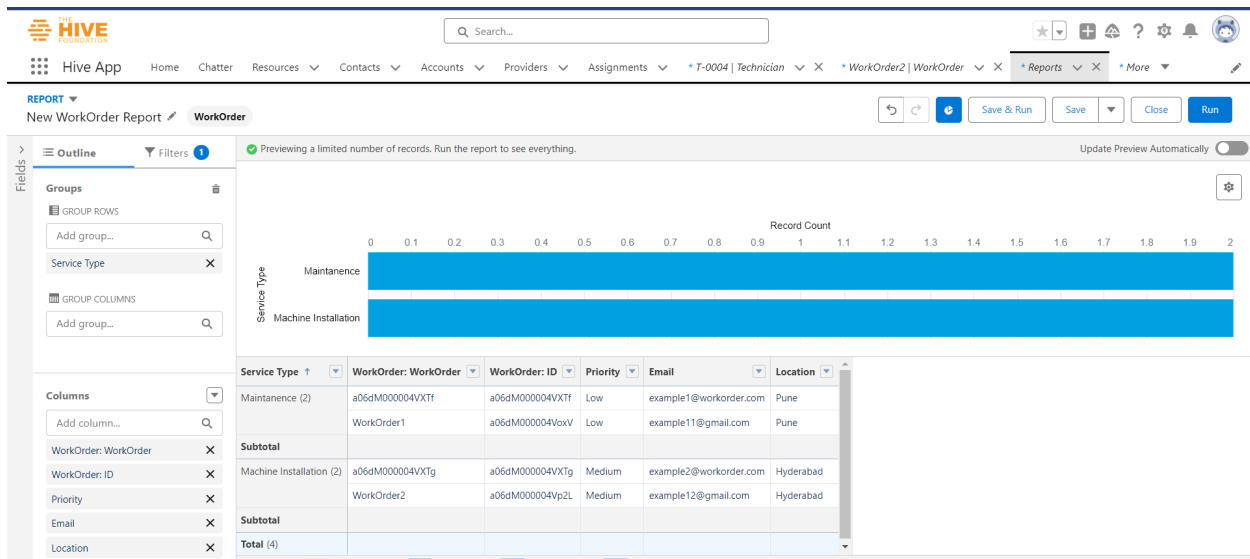
The screenshot displays the Hive App interface. At the top, there's a navigation bar with the Hive logo, a search bar, and various utility icons. Below this is a sidebar menu with options like Home, Chatter, Resources, Contacts, Accounts, Providers, Assignments, and a selected item 'Technician and Assignment...'. The main content area is titled 'Dashboard1' and shows a report titled 'Technician and Assignment Details'. This report is presented as a table with two data rows. Below the table, there's a link to 'View Report (Technician and Assignment Details)'. The background of the dashboard has a subtle blue pattern.

Assignment: Assign--	Technician ID: Tec--	Assign--	Technician I--	Technician --
A-0001	T-0001	28/09/20	example1@gr	Raghu
A-0002	T-0002	28/09/20	example2@gr	Raghav

[View Report \(Technician and Assignment Details\)](#)

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.



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

