Field Service WorkOrder Optimization

By

Mani Sankar Chintagunti

manisankarchintagunti@gmail.com

PROJECT ABSTRACT

The **Field Service Program Optimization Project** ambitions to streamline the workflow of a enterprise using Salesforce's powerful platform for installation and maintenance. The software makes a speciality of a complicated gadget that effectively manages work orders and strategically recruits area engineers primarily based on key elements.

To achieve this, the procedure begins by creating a Salesforce Developer Edition environment, which defines custom objects together with Technician, WorkOrder, Assignment and so on. Thus these objects are essential for the business enterprise and garage of the records the corporation ought to carry out its activities. The **Technician** item carries records approximately each technician, together with their competencies and contemporary status, while the **WorkOrder** object takes the information of every service request and then the **Assignment** item with that those are connected, and coupled to each an arrangement with the most certified engineer.

The coronary heart of the system is a precedence set of rules that automates paintings orders. This set of rules analyzes various parameters, including proximity to places of work, their availability, and the particular abilities required for every undertaking By optimizing this system, the system reduces travel time, provides the sources used are extensive, and make sure that each process is treated by a technician with an appropriate information.To keep all stakeholders informed, the project integrates **interactive automation** functions using Apex getting to know and trigger capabilities. These tools send notifications to technicians about new sports, updates, and finished paintings to decorate efficiency and timeliness.

In addition to centralized skills, the gadget has sturdy analytics and reporting talents leveraging Salesforce's effective tools. Customized analytics and dashboards had been developed to offer bank insight into common key commercial enterprise metrics which include reaction time, technician productiveness, and very last job cost This data permits managers to be proactive demonstrates field office control and knowledgeable selections to sell continuous development. Also the product emphasizes safety and privileges to manage client data and authorizations thru its identifiers, ensuring authenticity and confidentiality of debts Besides deleting vintage information, respecting responsibilities accomplished is constantly computed by way of an asynchronous apex magnificence to hold a inexperienced and easy database.

The final solution is introduced through a custom designed Lightning software, which presents an unbreakable and flexible user revel in to address area shipping services. This carrier simplifies operations for customers by means of consolidating all relevant statistics and tools right into a single interface and complements their capability to efficaciously execute their obligations. Overall, the Territory Management System Optimization project uses Salesforce's comprehensive talent set to offer and develop certain statistics that improves usual performance, reduces charges, and will increase client pride dynamic and adaptive management of territorial applications.

INDEX PAGE

INTRODUCTION

The Field Service Program Optimization Program is a strategic initiative aimed at changing how leading organizations approach field service management, including installation maintenance and operations at its core Service uses the Salesforce platform Woman implement and simplify Work to solve many ongoing challenges and solve the field work Optimize the business process , wants to improve field efficiency and effectiveness with a focus on driving technician performance higher, and on increasing customer satisfaction

Achieving this goal requires the creation and configuration of custom Salesforce objects such as Technician, WorkOrder, and Assignment for the role. These features are optimized for capturing and monitoring critical field operational data. Consideration of location, space and specialist skills, as well as a sophisticated prioritization process to ensure services are provided to only the most qualified technicians, make this operate more like a manufacturing plant than it does business is not only convenient but increases responsiveness and accuracy in all field applications

The program further enhances its capabilities by adding advanced virtual communication tools and real-time analytics. These tools make it easy for field teams to communicate seamlessly and address new issues immediately. By delivering actionable insights derived from real-time data, the system empowers organizations to make rational decisions faster, improving service delivery and operational efficiency.

The Field Operations Excellence System transforms traditional farm operations management by combining advanced technology and process improvements to create a more dynamic and efficient farm operations system. It goes beyond traditional ways of managing and optimizing materials, using automation to align existing technology with business needs This automation reduces operating costs by eliminating waste on the Enhanced analytics Enables real-time performance monitoring and data-driven improvements, resulting in faster, greater turnaround times for customer satisfaction etc. etc. Overall, the system for Field operations management improves from an operational aesthetic to a functional system, improving efficiency and customer experience.

Task 1: Creating a Salesforce Developer Edition Org

Building a Salesforce Developer Edition org Provides a controlled surroundings for builders to check, innovate, and create custom answers the usage of Salesforce's improvement gear and functions This gadget is vital for prototyping, trying out, and refining packages to satisfy specific commercial enterprise desires.

1.1 Sign Up for a Salesforce Developer Edition Org

1. **Visit the Salesforce Developer Signup Page:**
   Go to Salesforce Developer Signup.
2. **Fill Out the Signup Form:**
   First Name & Last Name - Enter your full name.
   Email - Provide a valid email address (note: this is for account activation purposes).
   Role - Select "Developer" from the dropdown menu.
   Company - Enter your college name or your organization's name if you are currently employed.
   Country - Select "India" from the dropdown menu.
   Postal Code - Enter your pin code.
   Username - Create a username that combines your name and company or organization, in the format: **username@organization.com**. This need not be an actual email address.

1.2 Activate Account

1. **Check Email Inbox:**
   Go to the email account you used to join up and look for an electronic mail from Salesforce with the concern line related to account activation. The e-mail may take 5-10 mins to reach.
2. **Verify Account:**
   Open the email and click on the **Verify Account** link. This will direct you to the Salesforce account activation page.
3. **Set Password and Security Question:**
   On the activation page, create a password for your Salesforce account and answer a security question. Click **Change Password** to complete the

activation process.

4. **Access Your Salesforce Setup Page:**
   After successfully changing your password, you will be redirected to your Salesforce setup page, where you can start configuring your Developer Edition org.

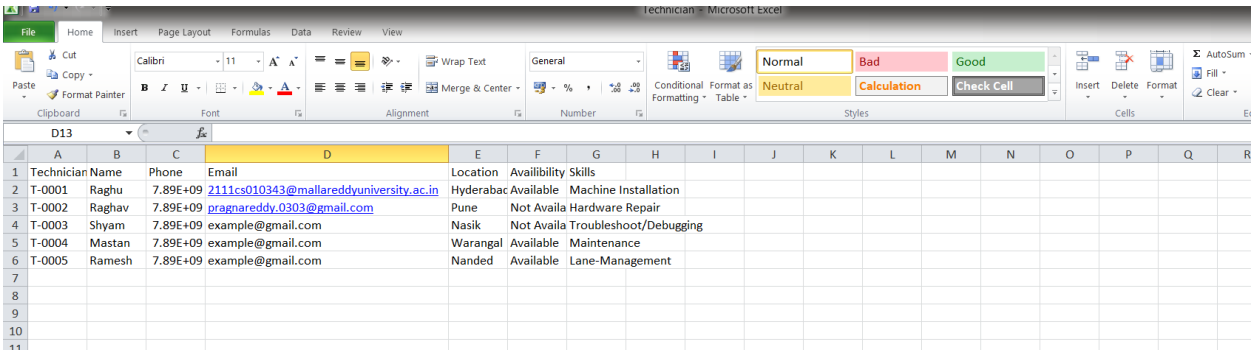
By following these steps, we will have set up your Salesforce Developer Edition org, allowing you to begin experimenting with and developing custom Salesforce solutions tailored to specific business needs.

Task 2: Object Creation

In Salesforce, objects are used to store and manage data according to business needs. In this project, custom properties such as Technician, WorkOrder, and Assignment were created for efficient field resource management. These objects act like database tables, with fields defined to capture specific types of information, ensuring that all necessary data is organized and easily accessible within the Salesforce platform.

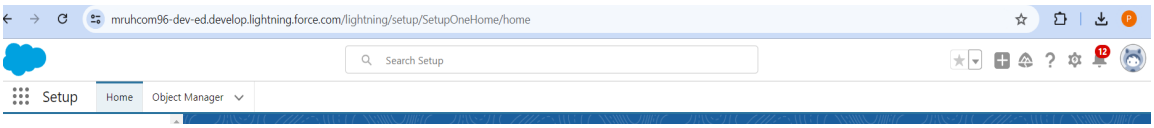
2.1 Creating the Technician Object

To create a Technician object, first, download and open the provided spreadsheet, change the email column by adding your email for at least one or two records, and save the file as "Technician.csv." ".

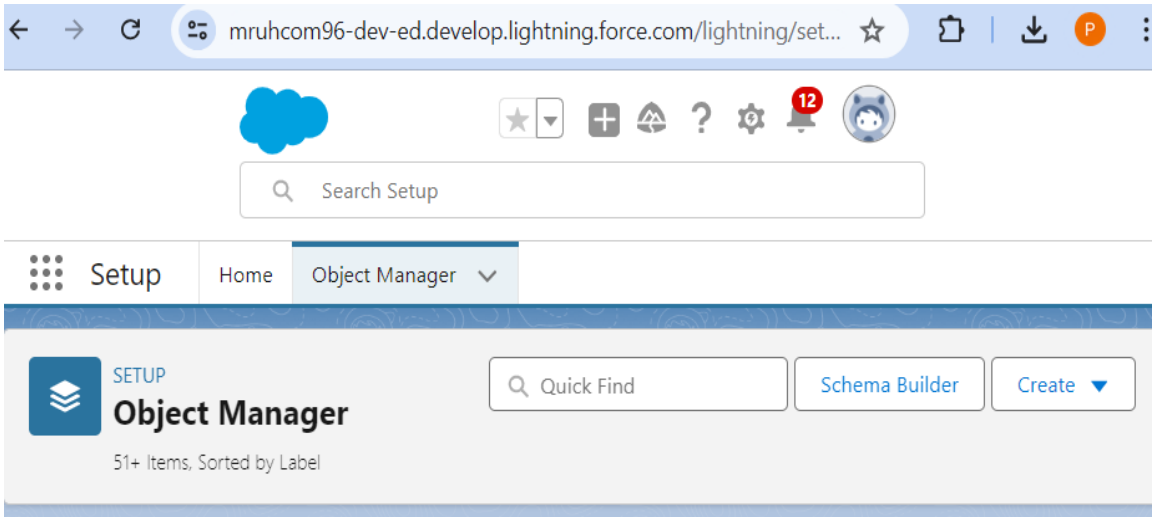


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Technician	Name	Phone	Email	Location	Availibility	Skills											
2	T-0001	Raghu	7.89E+09	2111cs010343@mallareddyuniversity.ac.in	Hyderabac	Available	Machine Installation											
3	T-0002	Raghav	7.89E+09	pragnareddy.0303@gmail.com	Pune	Not Availa	Hardware Repair											
4	T-0003	Shyam	7.89E+09	example@gmail.com	Nasik	Not Availa	Troubleshoot/Debugging											
5	T-0004	Mastan	7.89E+09	example@gmail.com	Warangal	Available	Maintenance											
6	T-0005	Ramesh	7.89E+09	example@gmail.com	Nanded	Available	Lane-Management											
7																		
8																		
9																		
10																		

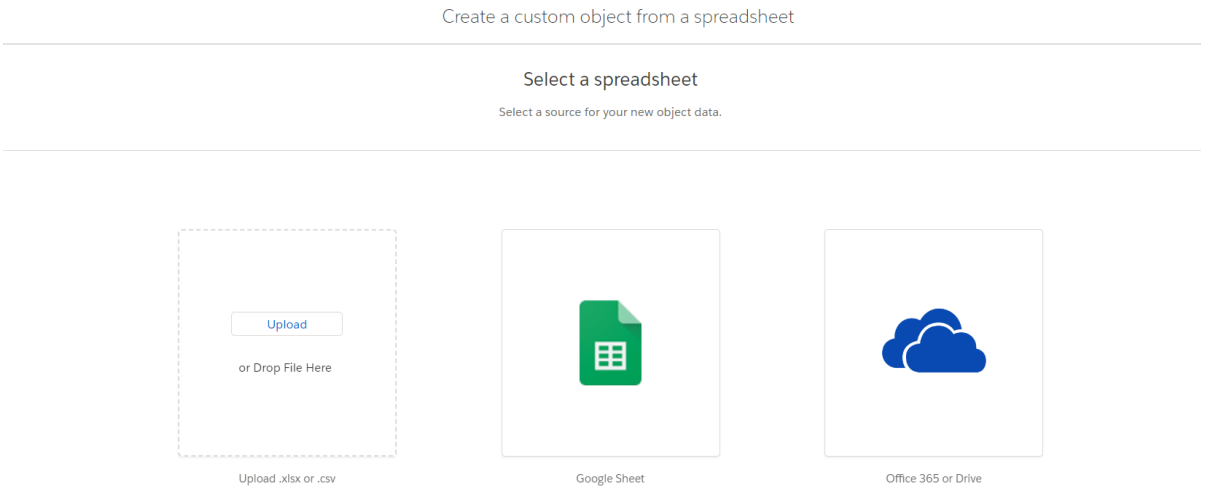
1. Log in to your Salesforce account and click on the Gear icon, then select "**Setup**".



2. **Access Object Manager**: In the Setup menu, navigate to the "Object Manager" tab.

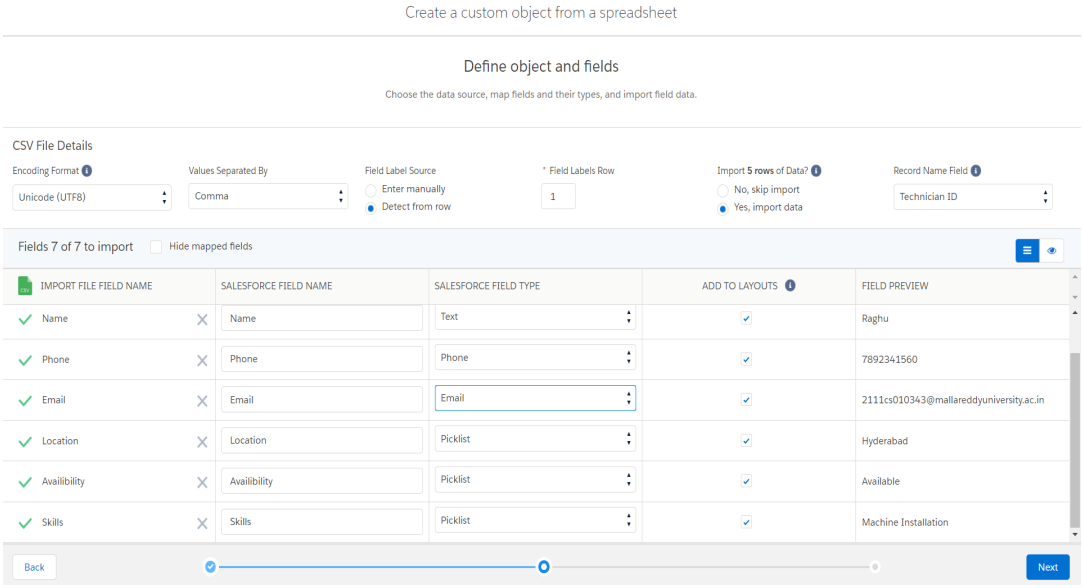


3. **Create Object from Spreadsheet**: Click "Create" and choose "Custom Object from Spreadsheet."
4. **Log In and Upload**: Click "Login With Salesforce," enter your credentials, and authorize access. Upload the Technician.csv file, ensuring Salesforce detects and populates the fields correctly.



5. **Configure and Finish**: Verify "Technician ID" as the Record Name field, check field

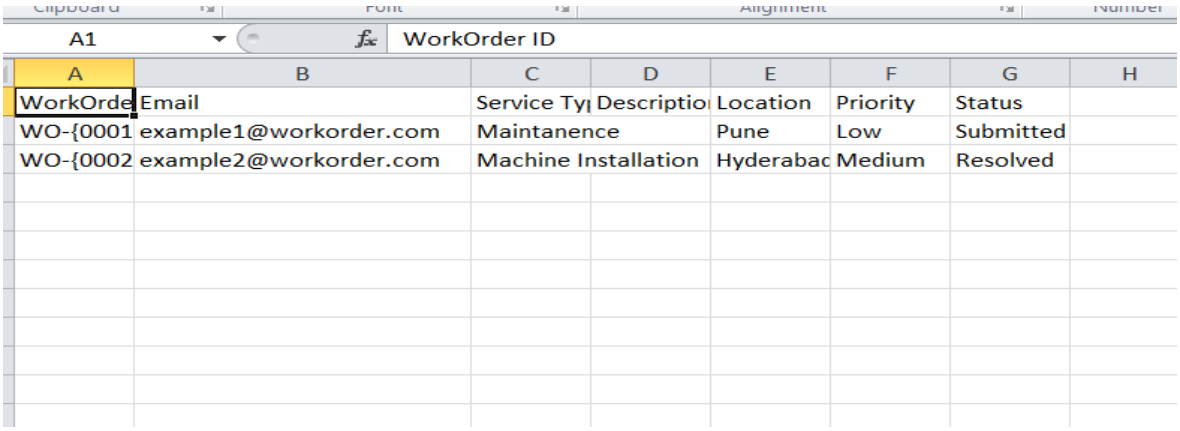
data types, and click "Next" and "Finish" to complete the creation and import of the Technician object.



2.2 Create WorkOrder Object

The WorkOrder object will manage information related to field service tasks and assignments.

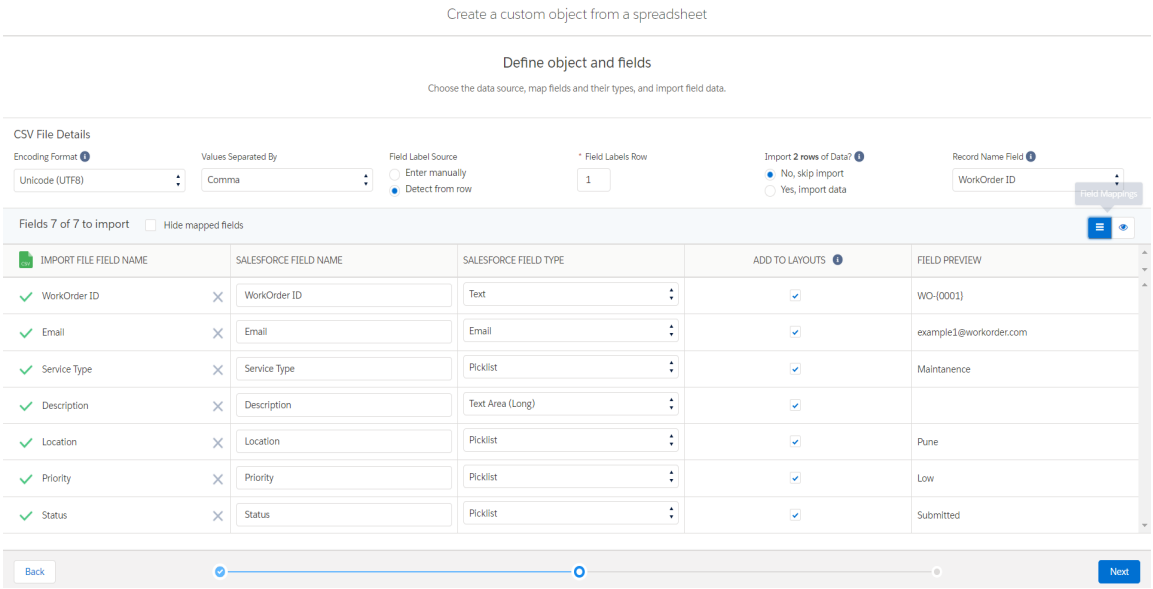
1. **Prepare the CSV File:** Obtain and review the WorkOrder CSV file template to ensure it meets the requirements.



2. **Repeat Steps to Create Custom Object:**

- Follow the same process as used for creating the Technician object.
- Upload the WorkOrder CSV file and verify that all field mappings are correct.

3. **Create Assignment Object:** The Assignment object will track assignments made to technicians and their related details.

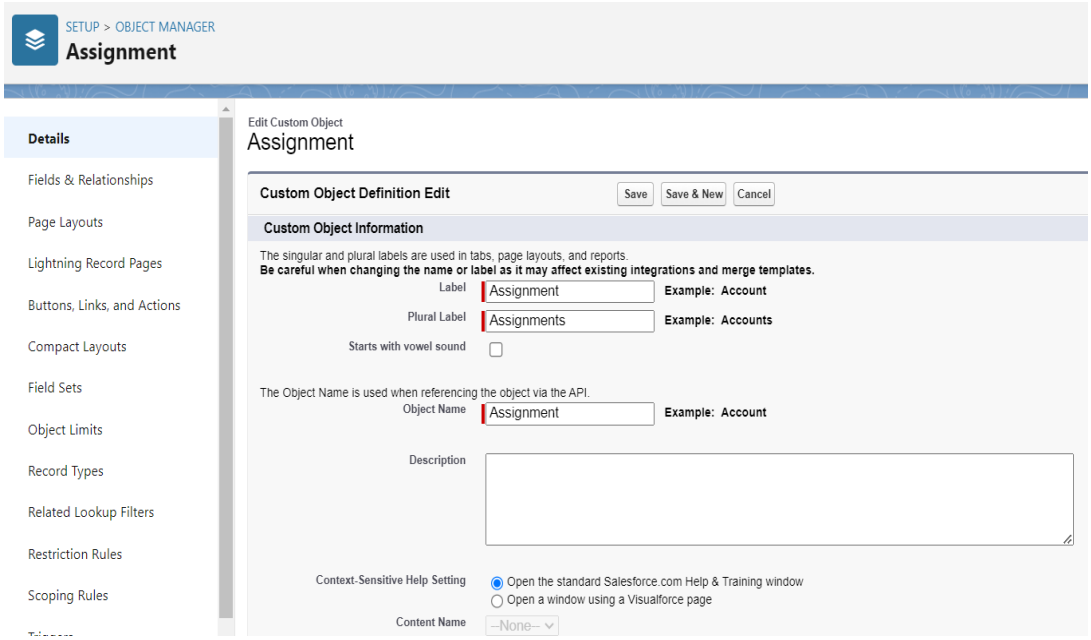


2.3 Create Assignment Object

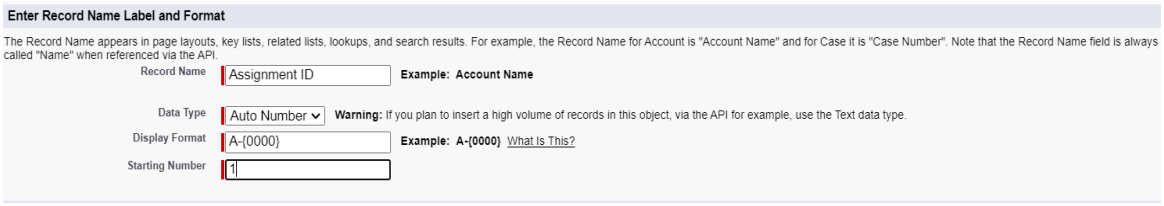
1. In Salesforce Setup, type **Object Manager** into the Quick Find box and select it.

2. Click on **Create** and choose **Custom Object** from the available options.

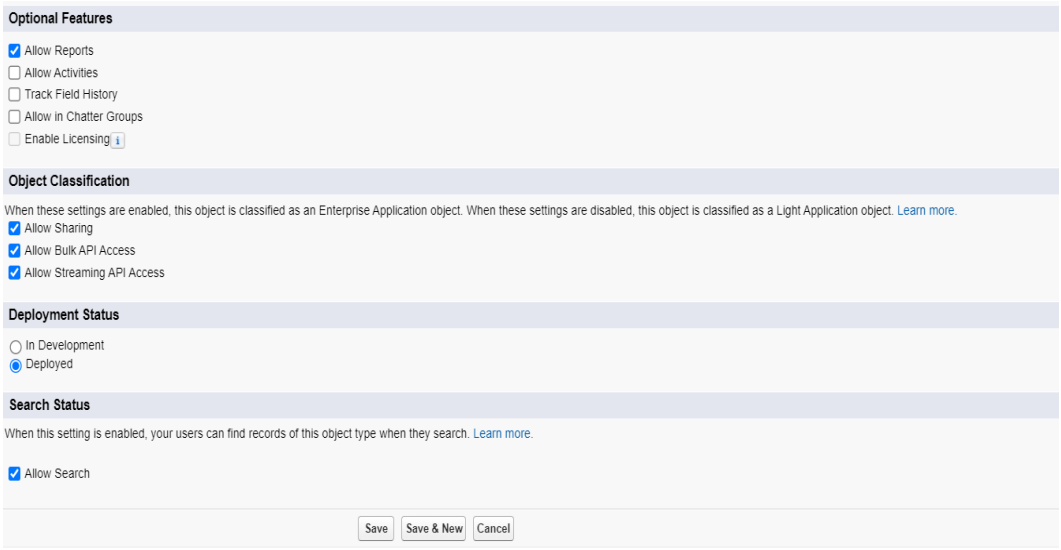
3. **Enter Object Details:**
   Label Name: Enter `Assignment` for the object label.
   Plural Label Name: Enter `Assignments` for the plural form.



4. **Configure Record Name:**
   Record Name: Set the Record Name to **Assignment ID**.
   Data Type: Choose `Auto Number` to automatically generate unique IDs for each record.
   Display Format: Use the format `A-{0000}` to ensure IDs are generated in the format "A-0001", "A-0002", etc.
   Starting Number: Set the starting number to `1` to begin numbering from 1.



5. Check the options to **Allow reports** and **Allow search** to enable reporting and searching functionalities for the Assignment object.



6. Click **Save** to complete the creation of the Assignment object and make it available for use in Salesforce.

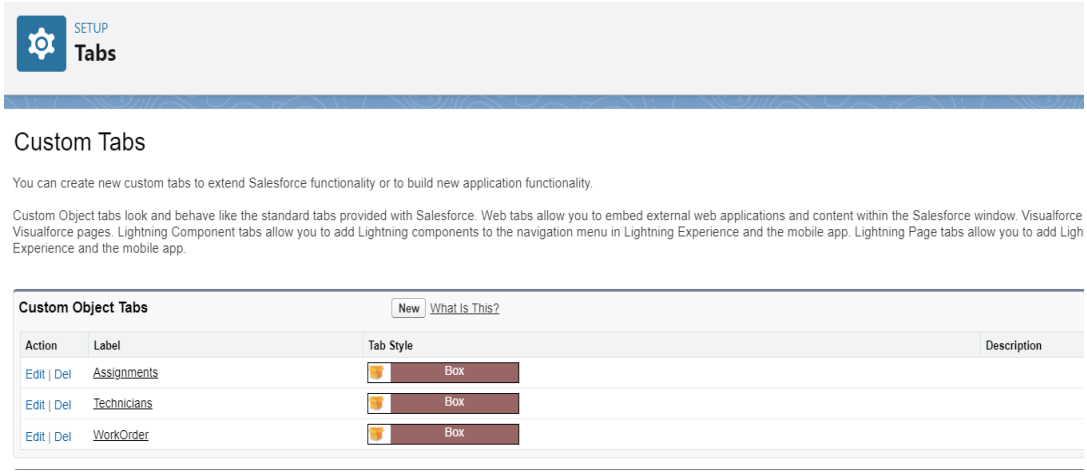
Task 3: Tabs

In Salesforce, a tab functions as a user interface element that allows users to create and view records for specific objects.

3.1 Creating a Custom Tab for Assignment Object

In Salesforce, tabs serve as the primary interface for interacting with object records. To facilitate user access to Assignment records, follow these steps to create a custom tab.

1. Log in to your Salesforce Developer Edition account. Click the Gear icon (⚙) in the top-right corner and select **Setup**.
2. In the Setup menu, search for "Tabs" using the Quick Find bar and click on it to access the tab configuration page.
3. Under **Custom Object Tabs**, click **New** to start creating a tab for the Assignment object.



4. Select the **Assignment** object from the dropdown menu and choose an appropriate icon to represent it. Click **Next** to proceed.

5. Keep the default settings to add the tab to profiles with access to the Assignment object and include it in the default app. Adjust if necessary, then click **Save**.

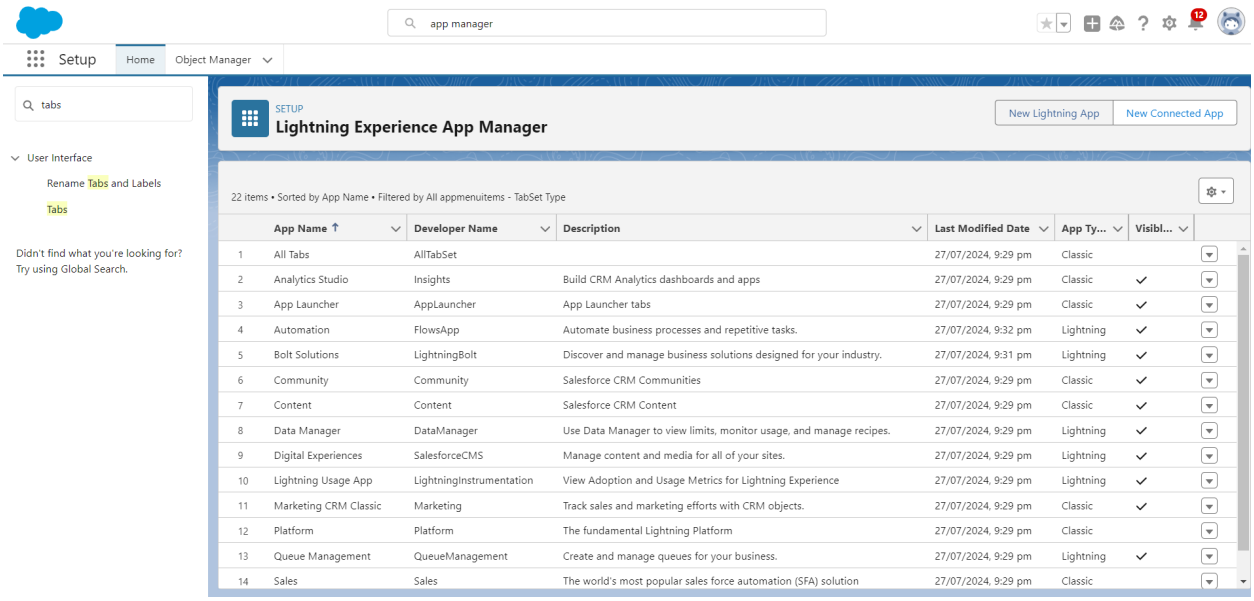
Task 4: Lightning App

Creating a database is just the beginning; users need an efficient way to access it. A Lightning App provides this by offering a user-friendly interface where users can easily navigate and manage the objects and access key features like reports and dashboards.

4.1 Creating a Lightning App

To create a Lightning App, go to the Setup page, find “App Manager”, and select “New Lightning App”. Enter information such as the name of the application such as “Field Service WorkOrder Optimization” to proceed to the configuration screens. Add navigation items such as Home, WorkOrder, Technician, and Reports, then assign the app to the “System Administrator” profile. Save and complete the settings and complete the creation of the Lightning App.

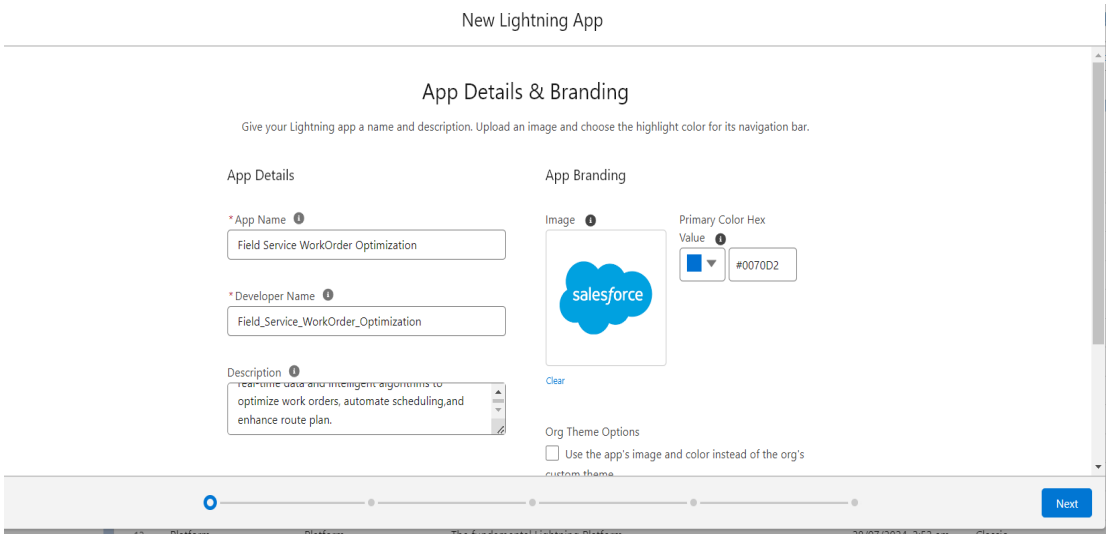
1. Begin by accessing the **Setup** page in Salesforce. This can be done by clicking the gear icon (⚙) in the top-right corner of the Salesforce interface.
2. In the **Quick Find** search bar on the left-hand side, type **“App Manager”** and select it from the search results to open the App Manager page.
3. **Create New Lightning App**: Click on **“New Lightning App”** to initiate the process of creating a new Lightning App.



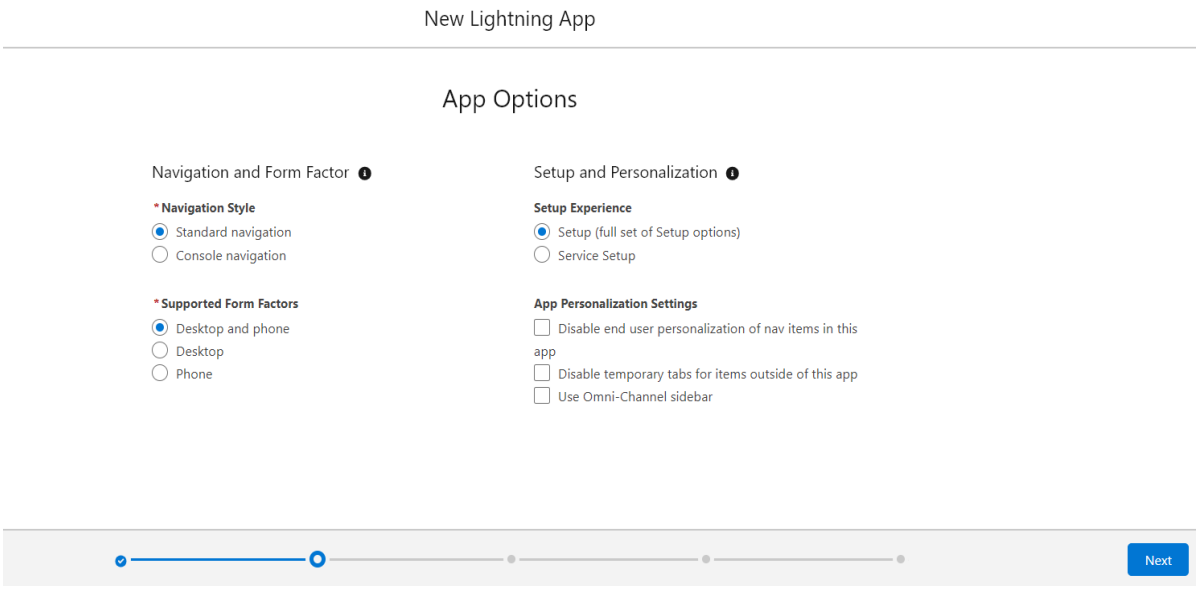
4. **Configure App Details and Branding**:

- **App Name**: Enter **Field Service WorkOrder Optimization** in the **App Name** field.
- **Developer Name**: This field will automatically populate based on the App Name.

- **Description**: Provide a meaningful description of the app's purpose and functionality.
- **Image**: Optionally, upload an image to represent the app visually. This step is not mandatory.
- **Primary Color**: Set the primary color for the app by entering a hex value or keeping the default setting.

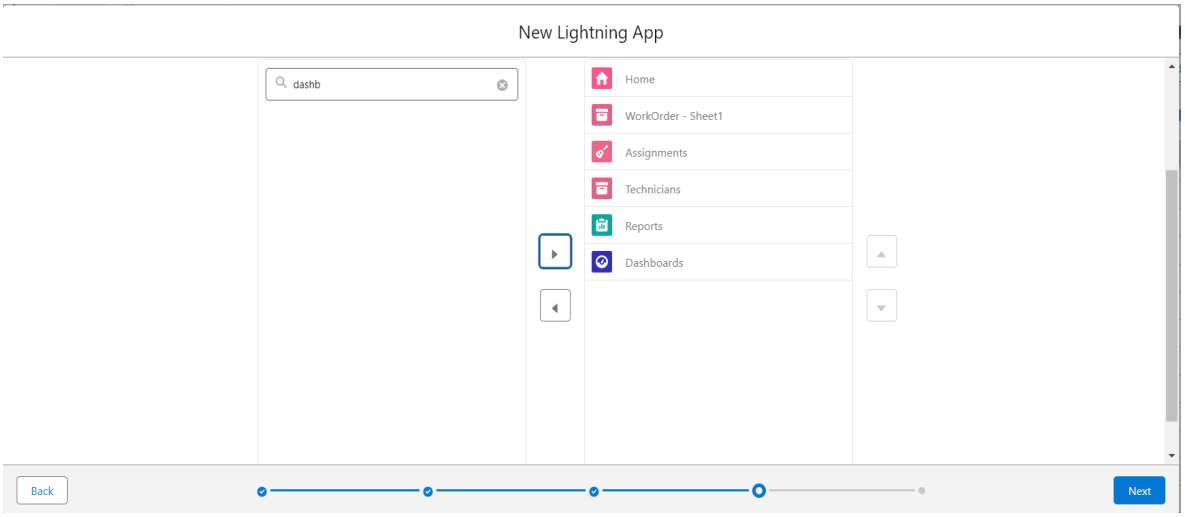


5. Click **Next** to move to the **App Options** page. Here, you can retain the default settings unless specific customizations are required.

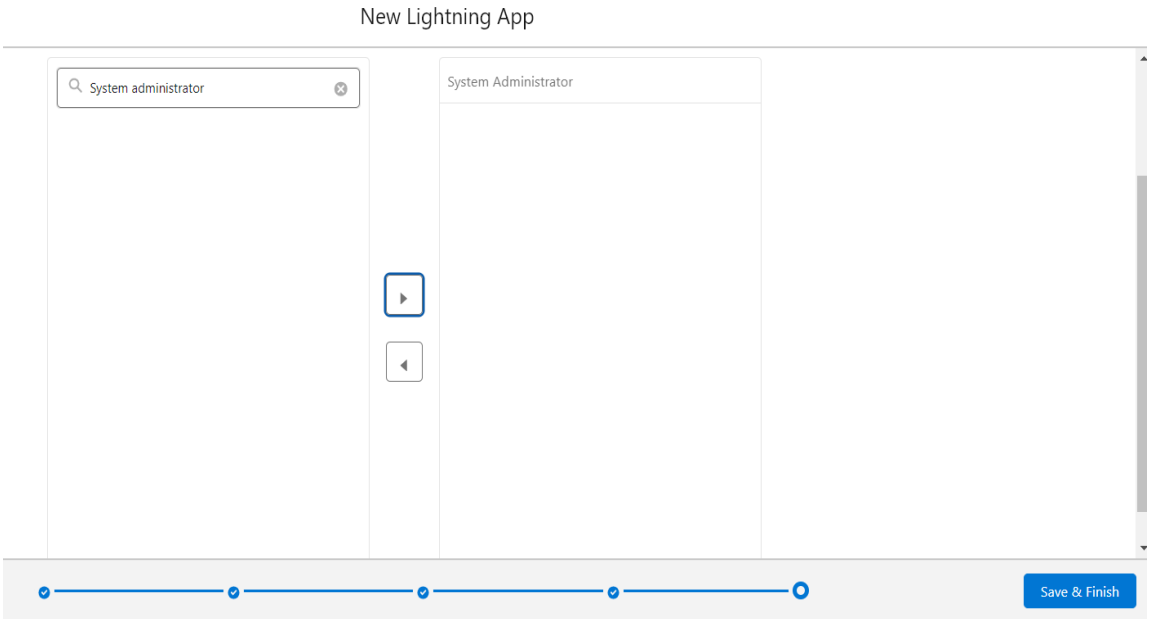


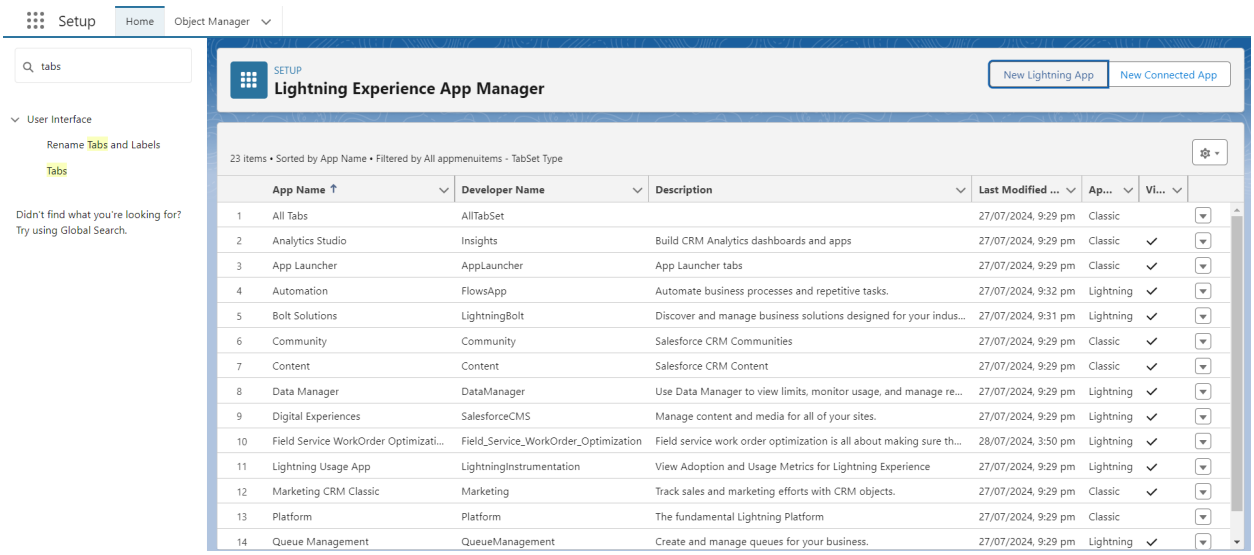
6. On the **Utility Items** page, you can leave the default settings or configure utility items if necessary. Click **Next** to proceed.

7. Use the search bar to find and select items for the app's navigation menu, including Home, WorkOrder, Technician, Assignment, Reports, and Dashboard. Move these selected items to the navigation section using the arrow button, then click **Next** to confirm the additions.



8. Search for "System Administrator" in the profile search bar to assign the app to users with this profile, move it to the selected profiles section using the arrow button, and then click **Save** and **Finish** to complete the app creation process.



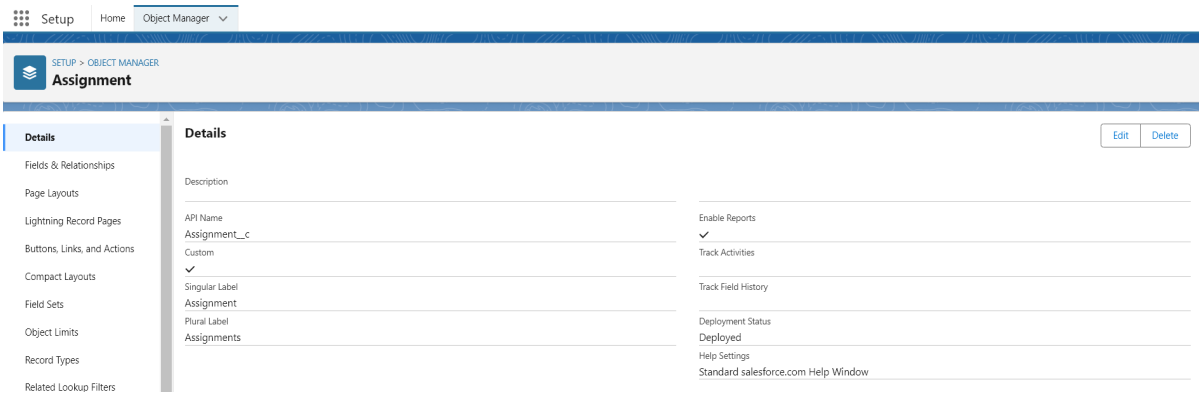


Task 5: Fields & Relationships

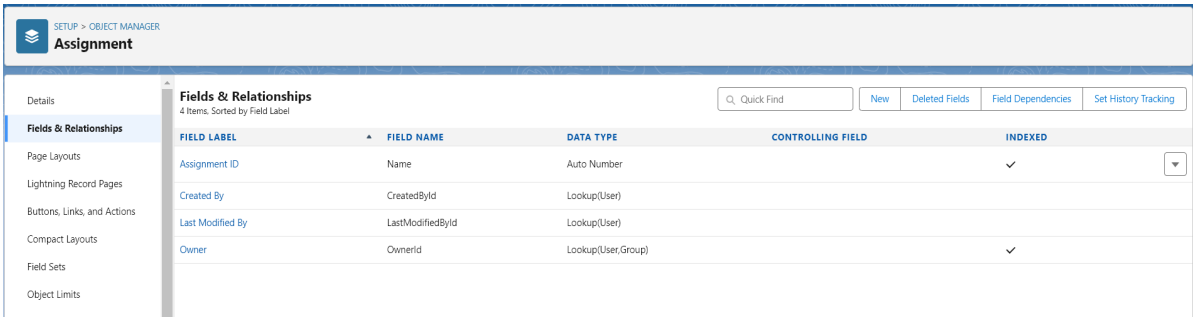
Defining fields within the objects specifies the types of data they store. Setting up relationships between objects integrates this data, enhancing the system's functionality and relevance to organizational needs.

5.1 Creating a Lookup Field in the Assignment Object

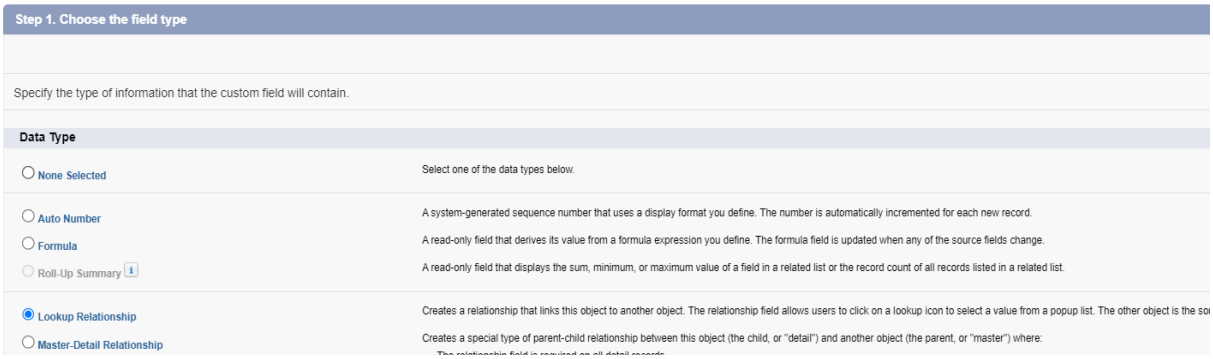
1. Log in to Salesforce and click on the Gear icon (⚙) in the top-right corner to open the Setup menu. Type "Object Manager" in the Quick Find bar and select "Object Manager" from the search results.
2. In the Object Manager, search for "Assignment" and click on the "Assignment" object to open its settings.



3. Within the Assignment object settings, click on "Fields & Relationships" from the left-hand menu.

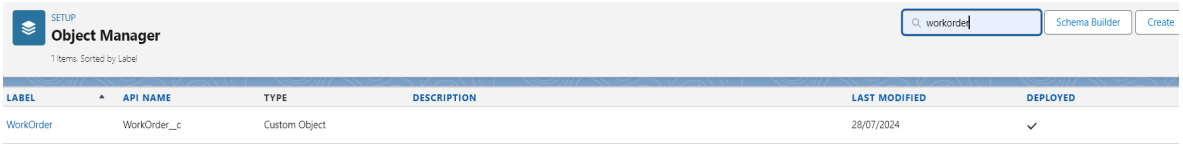


4. Click on the "New" button to start creating a new field.
5. Choose "Lookup Relationship" as the data type and click "Next."



6. **Configure Lookup Field**:
    - For "Related To," select the "WorkOrder" object from the dropdown menu. Avoid selecting other standard objects with similar names to prevent confusion.
    - Enter "WorkOrder ID" as the Field Label and click "Next."

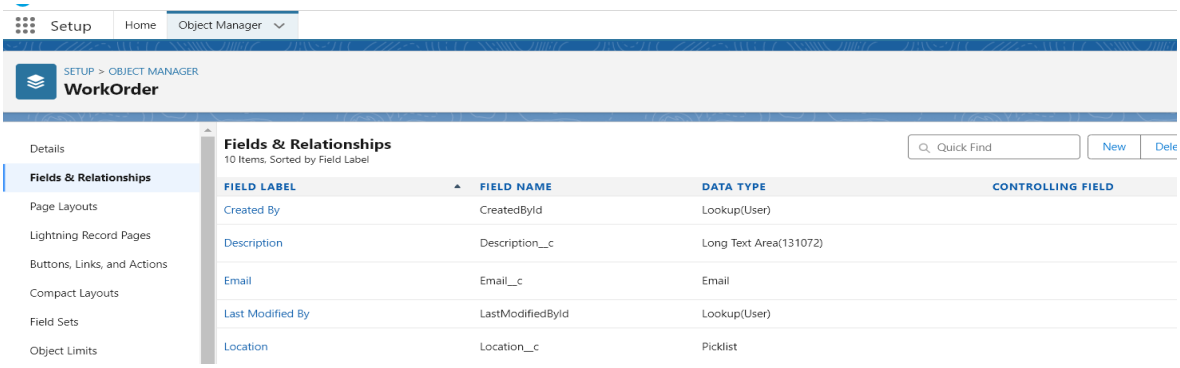
7. **Set Field-Level Security**:
    - Click "Next" to proceed through the field-level security settings.
    - Click "Save & New" to create the field and return to the creation screen for additional fields, if needed.



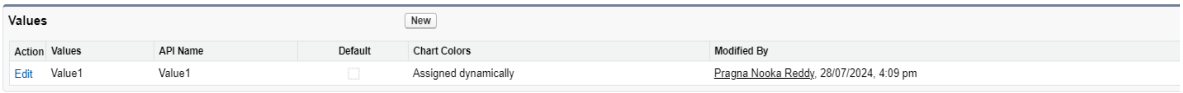
5.2 Manage your picklist values

To manage picklist values in Salesforce, go to the Object Manager, select the WorkOrder object, and edit the Location field. Add new values like Nasik, Warangal, and Nanded, then save your changes.

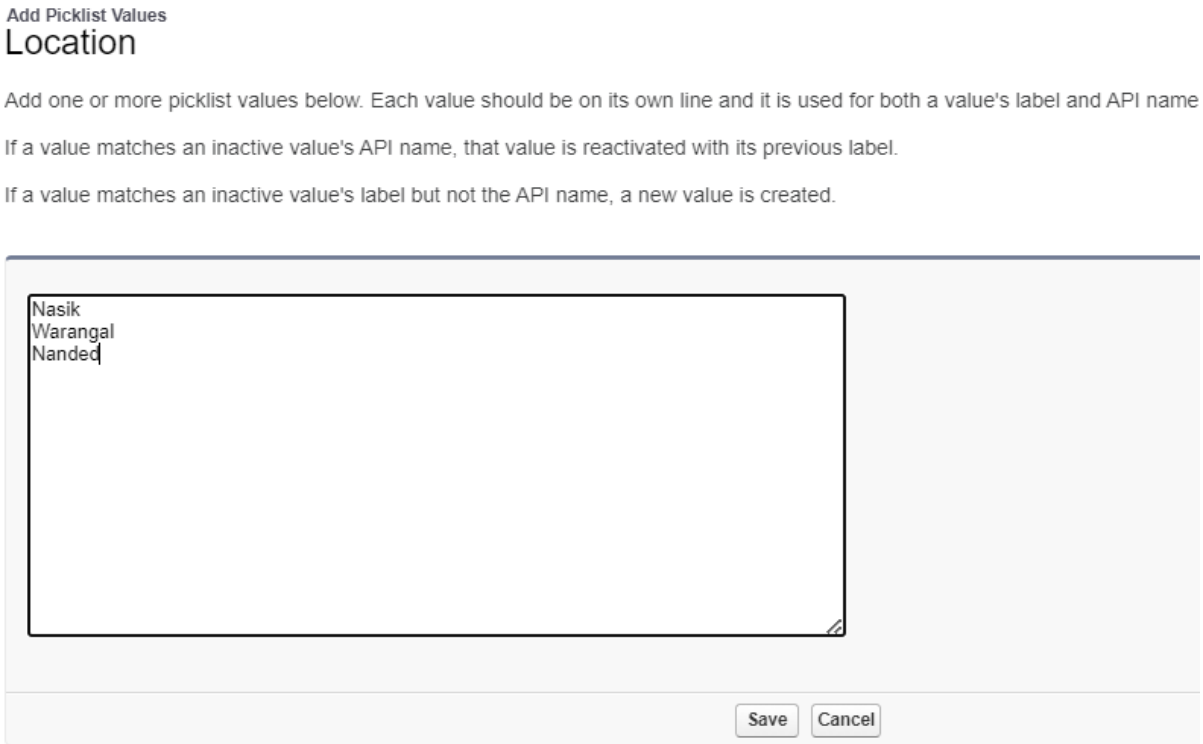
1. Begin by accessing the Setup page within Salesforce and selecting **Object Manager** from the menu
2. Use the search function to find and choose the **WorkOrder** object from the list of available objects.
3. Within the WorkOrder object settings, go to the **Fields & Relationships** section to manage the field details.



4. Locate the **Location** field and click on it to view its properties. Scroll down to the **Values** section.



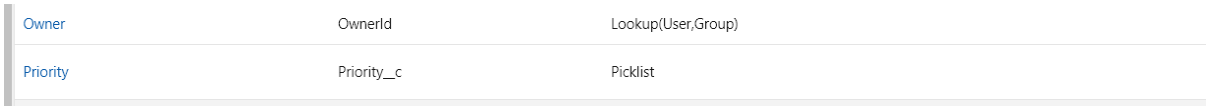
5. Click the **New** button to introduce additional picklist values. Enter the new values: **Nasik**, **Warangal**, and **Nanded**.

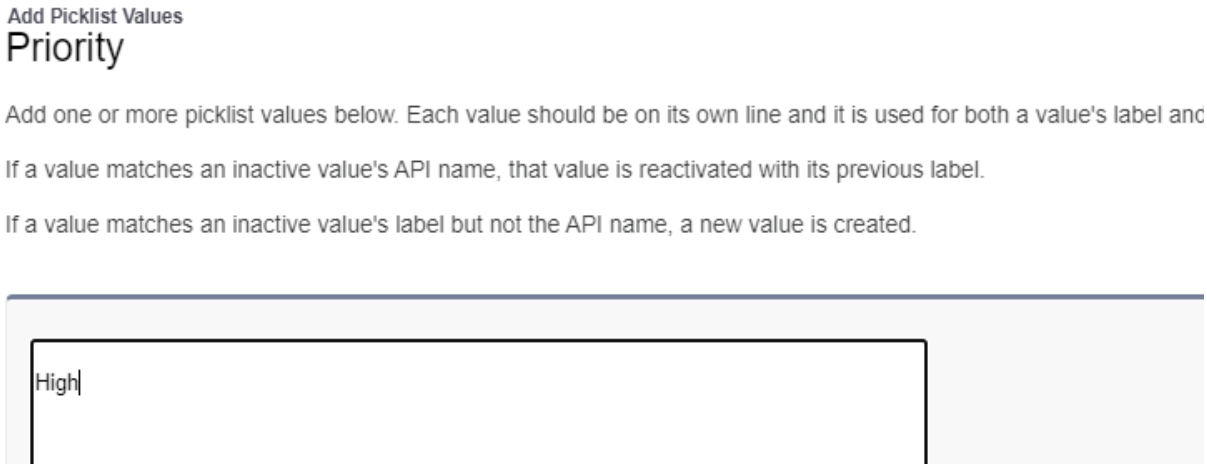


6. After entering the new values, click **Save** to update the field with the new picklist options.

5.3 Manage your picklist values

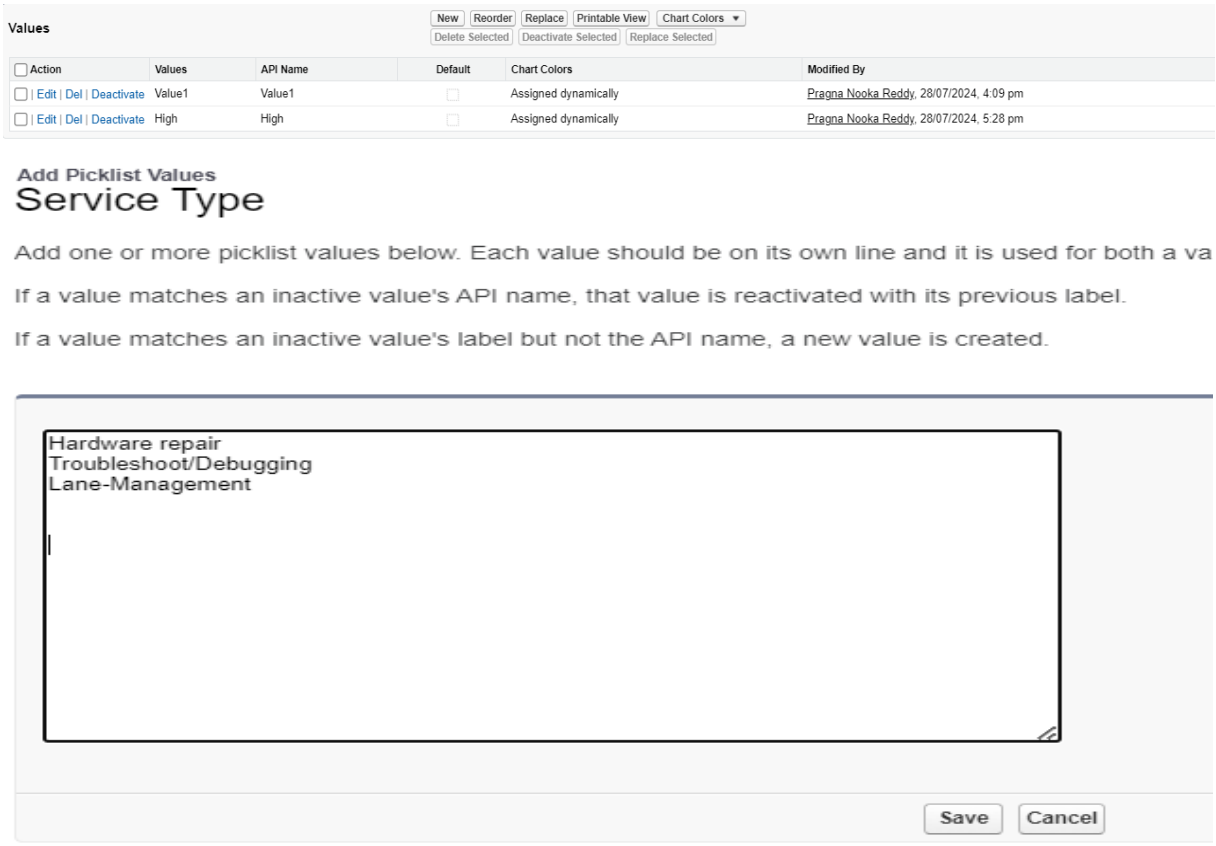
1. Navigate to the Object Manager from the Salesforce Setup page, select the WorkOrder object, and then access Fields & Relationships within the WorkOrder object settings.
2. **Modify the Priority Field**:Find and click on the Priority field, scroll to the Values section, click New, add the value "**High**," and then click Save to apply the changes.





3. **Modify the Service Type Field**:

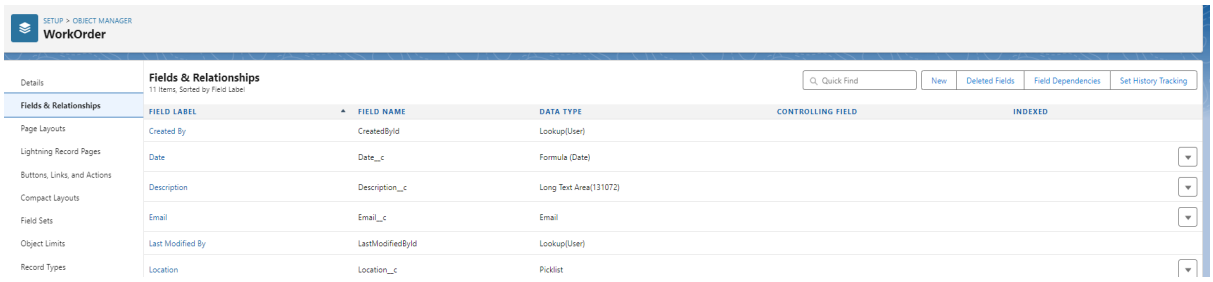
- Find and click on the **Service Type** field.
- Scroll to the **Values** section and click **New**.
- Add the values: **Hardware repair**, **Troubleshoot/Debugging**, **Lane-Management**.
- Click **Save** to apply the changes.



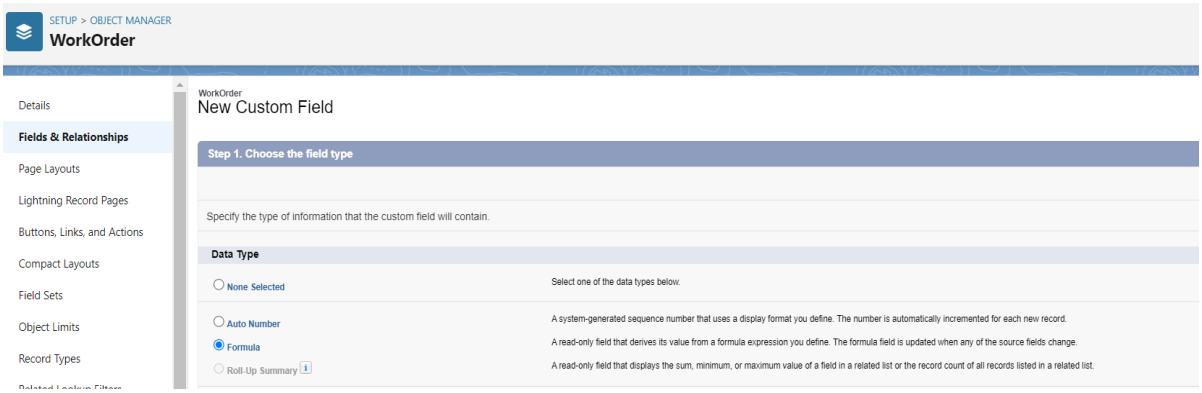
5.4 Creating Formula Field in WorkOrder Object

To create a formula field in the WorkOrder object, go to **Object Manager** > **WorkOrder** > **Fields & Relationships**. Choose **New**, select **Formula**, set the field label and name to **Date**, use `**CreatedDate**` as the formula, and save.

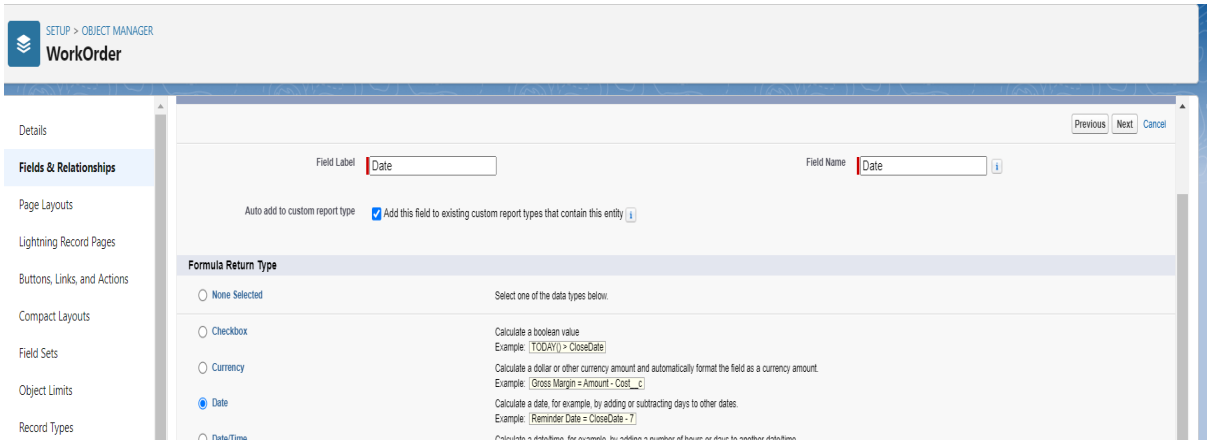
1. Navigate to the Salesforce Setup page and select **Object Manager** from the menu.
2. **Select the WorkOrder Object:** In the Object Manager, search for and select the **WorkOrder** object to access its settings.
3. Within the WorkOrder object settings, locate and click on **Fields & Relationships** to manage the fields associated with this object.



4. **Create a New Field:** Click on **New** to start creating a new field for the WorkOrder object.
5. **Choose Data Type:** Select **Formula** as the data type for the new field and click **Next** to proceed with the formula field setup.

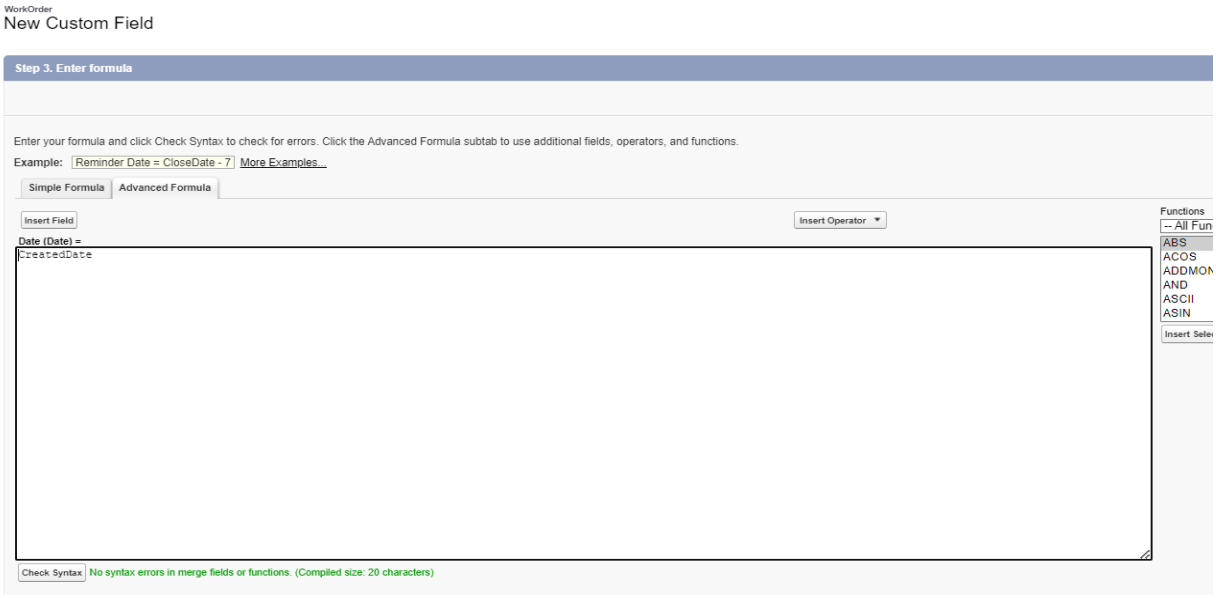


6. **Define Field Details:** Enter **Date** for the **Field Label** and **Field Name**. Choose **Date** as the formula return type from the available options. Click **Next** to continue.



7. **Enter Formula:**
    - In the **Advanced Formula** section, input the formula CreatedDate.

- **Formula: CreatedDate**
- This formula will capture the date when the record was created. Click **Check Syntax** to ensure there are no errors in the formula.



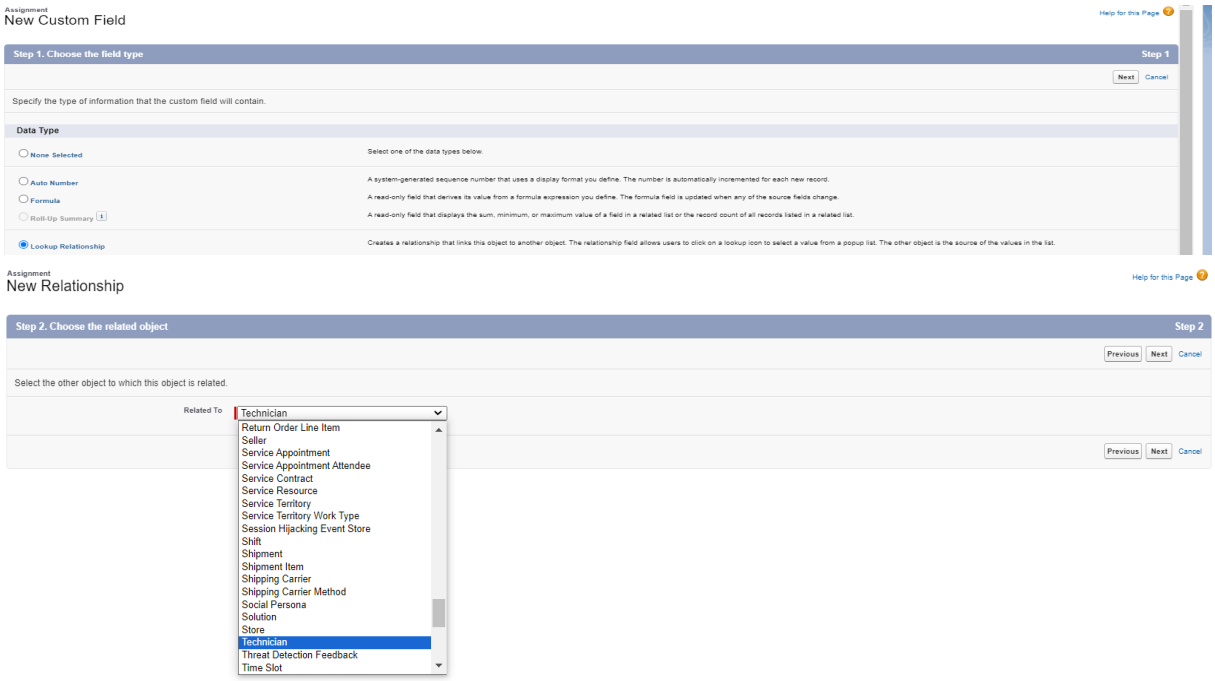
8. **Save the Field:**
    - After verifying the syntax, click **Next**, then review your settings. Click **Save** to finalize the creation of the formula field.

5.5 Creating Remaining fields for the respective objects

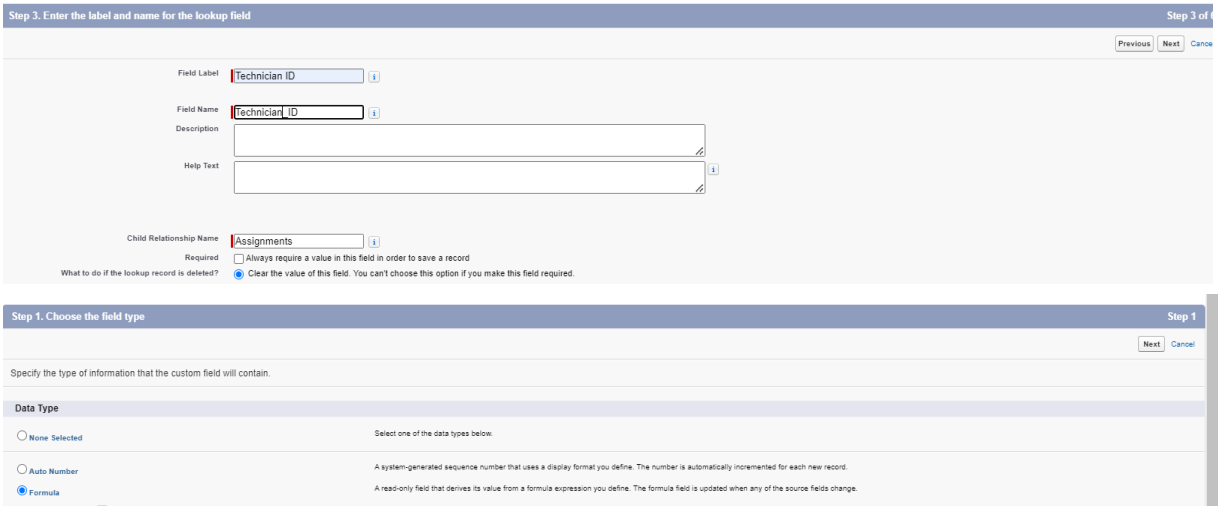
The project configured Salesforce to manage field service operations by creating custom objects and fields, developing a Lightning App for user access, and setting up navigation and profiles. This solution enhances efficiency and data management for the organization.

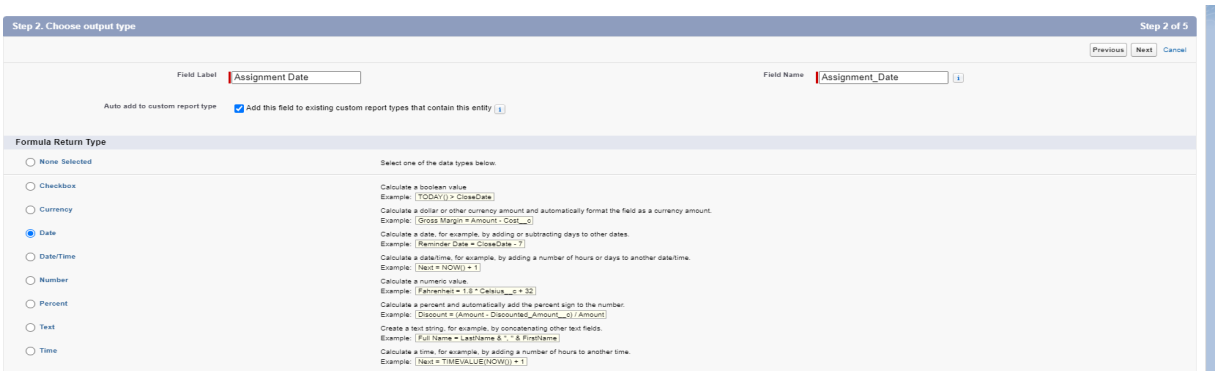
To create the remaining fields for the respective objects:

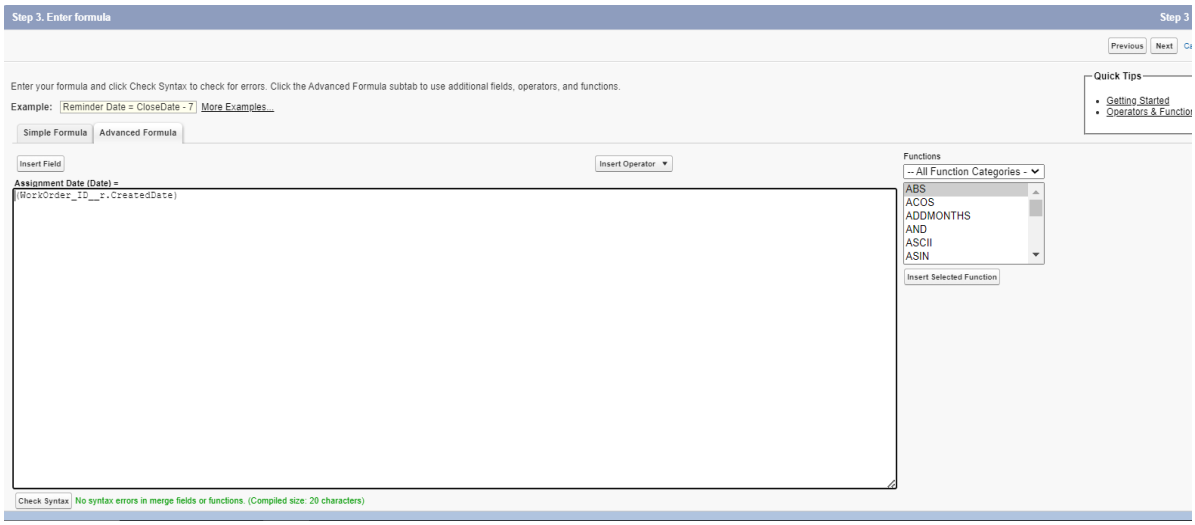
1. **Assignment Object**:
    - **Technician ID**: Create a field with the datatype **Lookup** referencing the **Technician** object.



- **Assignment Date**: Create a field with the datatype **Formula**, returning a **Date**. Use the formula **`WorkOrder_ID__r.Date__c`.**

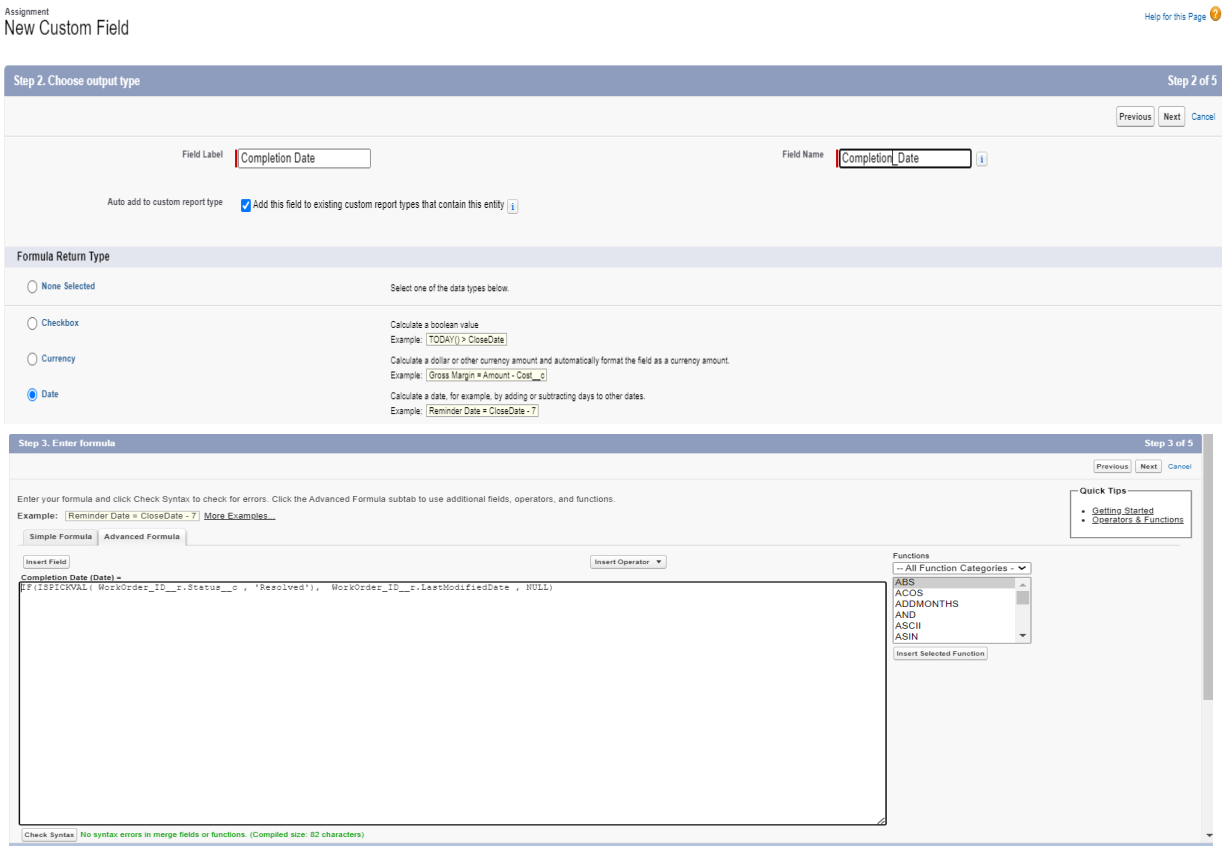






- **Completion Date**: Create a field with the datatype **Formula**, returning a **Date**. Use the formula

  `IF(ISPICKVAL(WorkOrder_ID__r.Status__c, 'Resolved'), WorkOrder_ID__r.LastModifiedDate, NULL).`



In Salesforce, navigate to the Object Manager, select the respective object, and add these fields with the specified data types and formulas.

Task 6: Profiles

Profiles in Salesforce control user permissions and access levels, defining what users can view and modify within the system. They manage access to objects, fields, and records, ensuring users have the necessary permissions for their roles while maintaining data security and operational efficiency. Each profile tailors access based on the user's responsibilities, balancing functionality and protection.

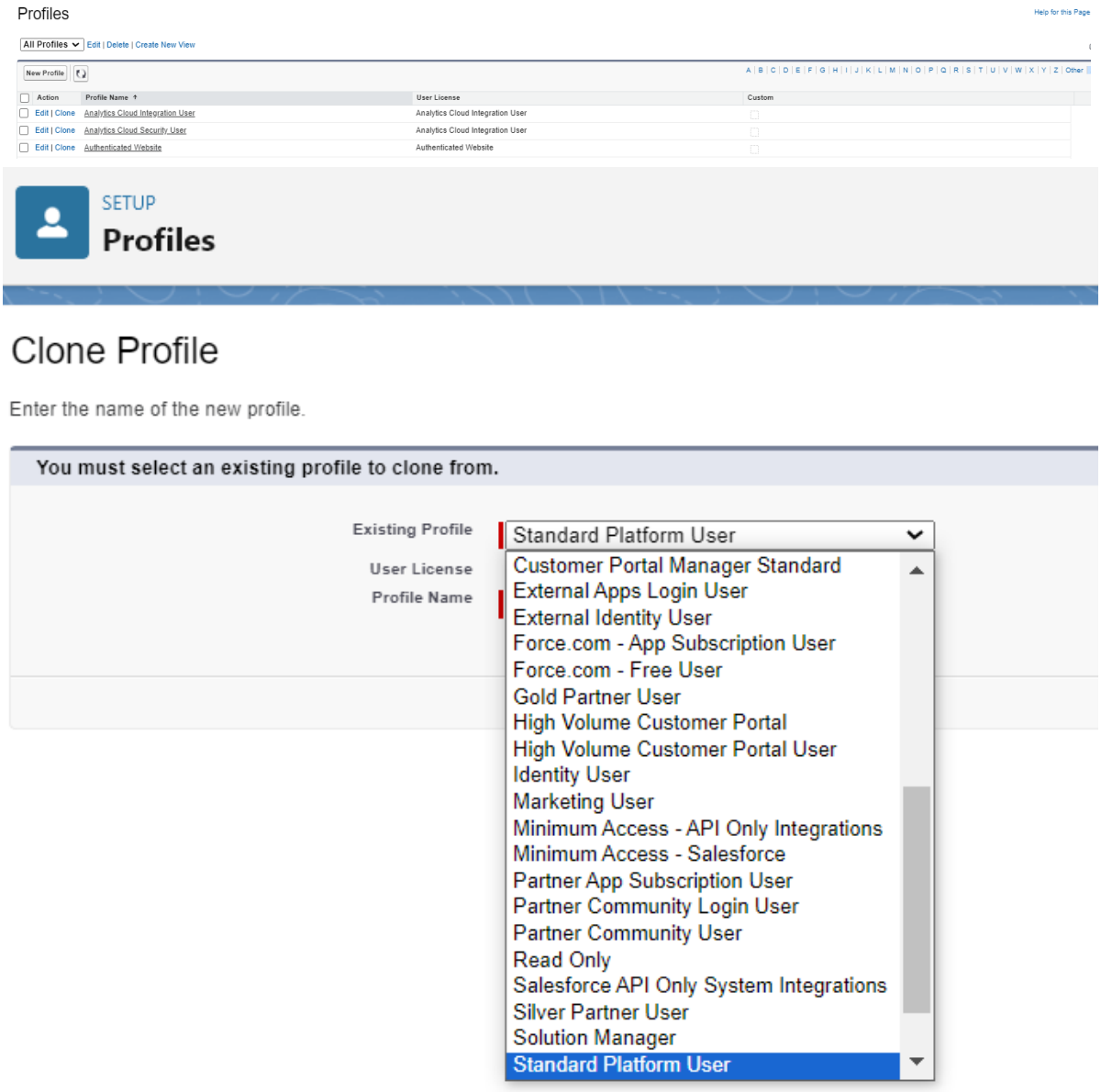
6.1 Technician Profile

Profiles in Salesforce define user permissions and access levels, controlling what users can see and do within the system. They ensure appropriate access based on user roles, maintaining data security and operational efficiency.

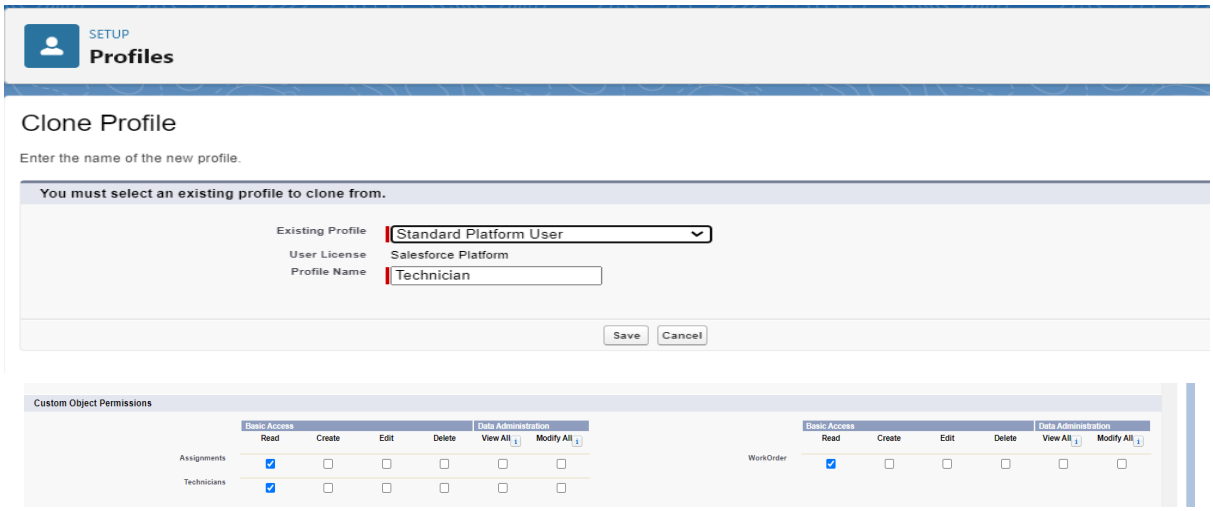
1. Navigate to Setup by clicking the gear icon, then type "Profiles" in the Quick Find

box and select "Profiles" from the search results.

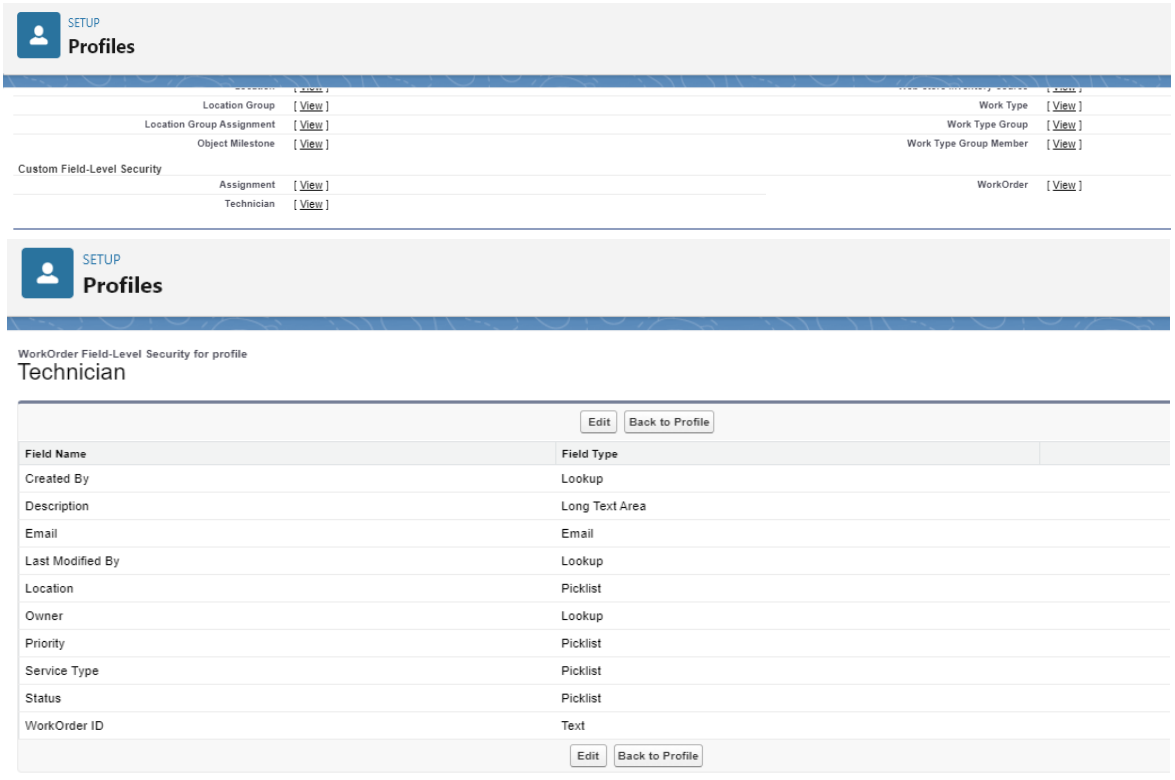
2. Click on "New Profile." Choose "Standard Platform User" as the base profile and name the new profile "Technician." Click "Save" to create the profile.

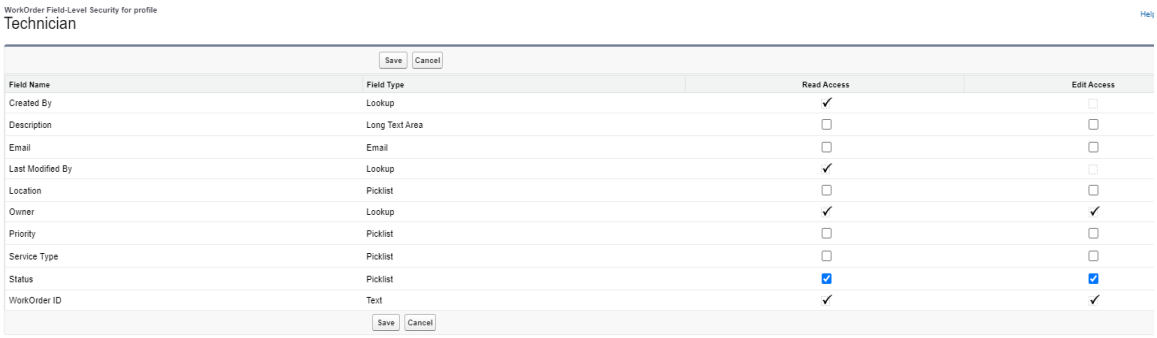


3. **Edit Profile Permissions:** After saving, click "Edit" on the Technician profile page. Scroll down to the "Custom Object Permissions" section. Grant "Read Only" access to the Technician, WorkOrder, and Assignment objects.



4. **Adjust Field-Level Security:** Continue on the profile page and scroll down to "Custom Field Level Security." Click "View" next to the WorkOrder object. Then click "Edit," check the box to enable access for the Status field, and click "Save."




WorkOrder Field-Level Security for profile
Technician

Save Cancel

Field Name	Field Type	Read Access	Edit Access
Created By	Lookup	✓	
Description	Long Text Area	☐	☐
Email	Email	☐	☐
Last Modified By	Lookup	✓	
Location	Picklist	☐	☐
Owner	Lookup	✓	✓
Priority	Picklist	☐	☐
Service Type	Picklist	☐	☐
Status	Picklist	☑	☑
WorkOrder ID	Text	✓	✓

Save Cancel

These steps ensure that the Technician profile has the appropriate read-only access to the necessary objects and the required visibility to the Status field in the WorkOrder object.

Task 7: Users

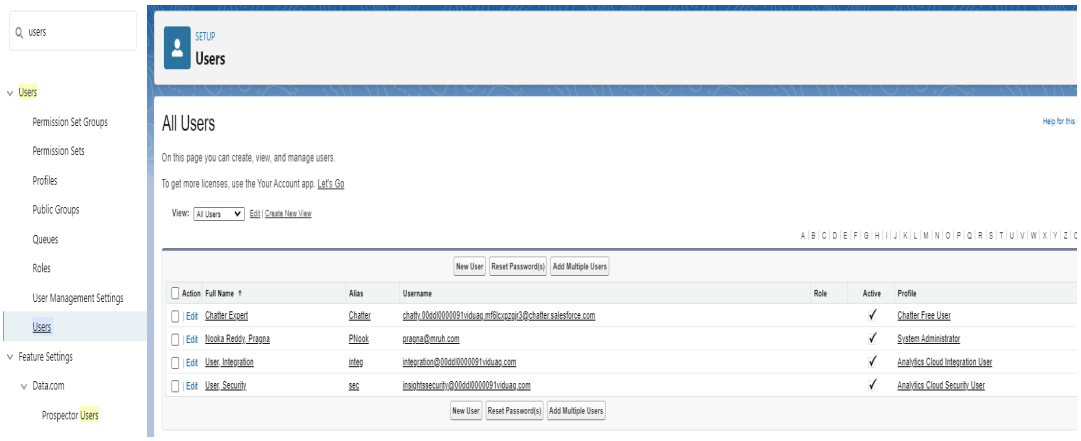
Users in Salesforce are the employees who interact with the system based on their roles and assigned profiles. Their access and permissions are tailored to fit their job functions, allowing them to effectively perform their tasks within the Salesforce environment. This ensures that each user has the appropriate level of interaction with the system's features and data.

7.1 Creating a User in Salesforce

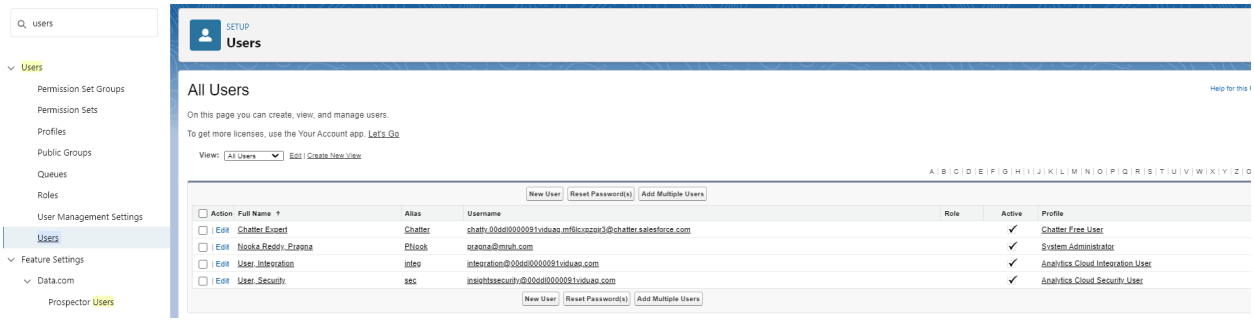
To create a user in Salesforce, go to the "Users" section in Setup, click "New User," enter the required details, and then click "Save" to complete the process.

To create a new user in Salesforce, follow these detailed steps:

1. **Navigate to Users Section**:
    - Log in to your Salesforce account.
    - Click on the Gear icon (⚙) in the top-right corner to access the Setup menu.
    - In the Quick Find box, type "Users" and select "Users" from the search results.

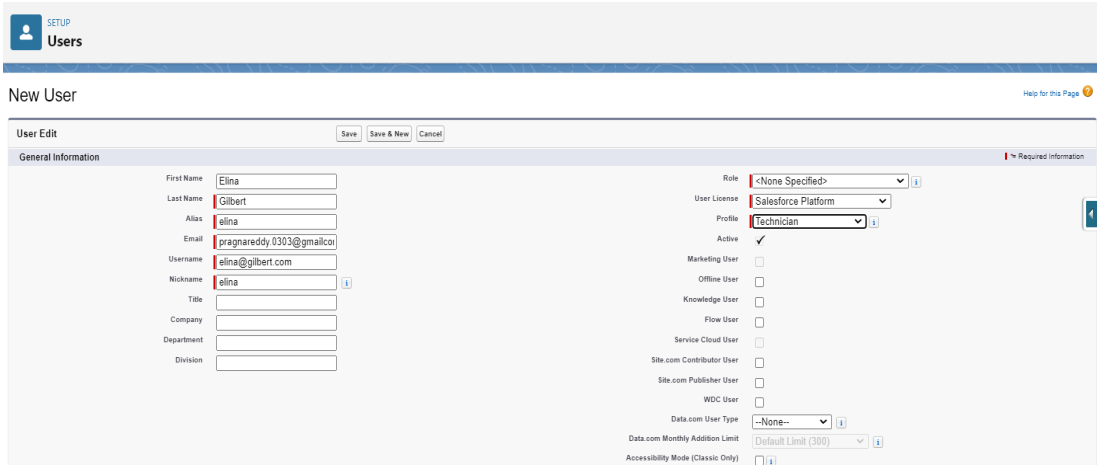


2. **Start New User Creation**:
    - On the Users page, click the "New User" button to begin the process of creating a new user.

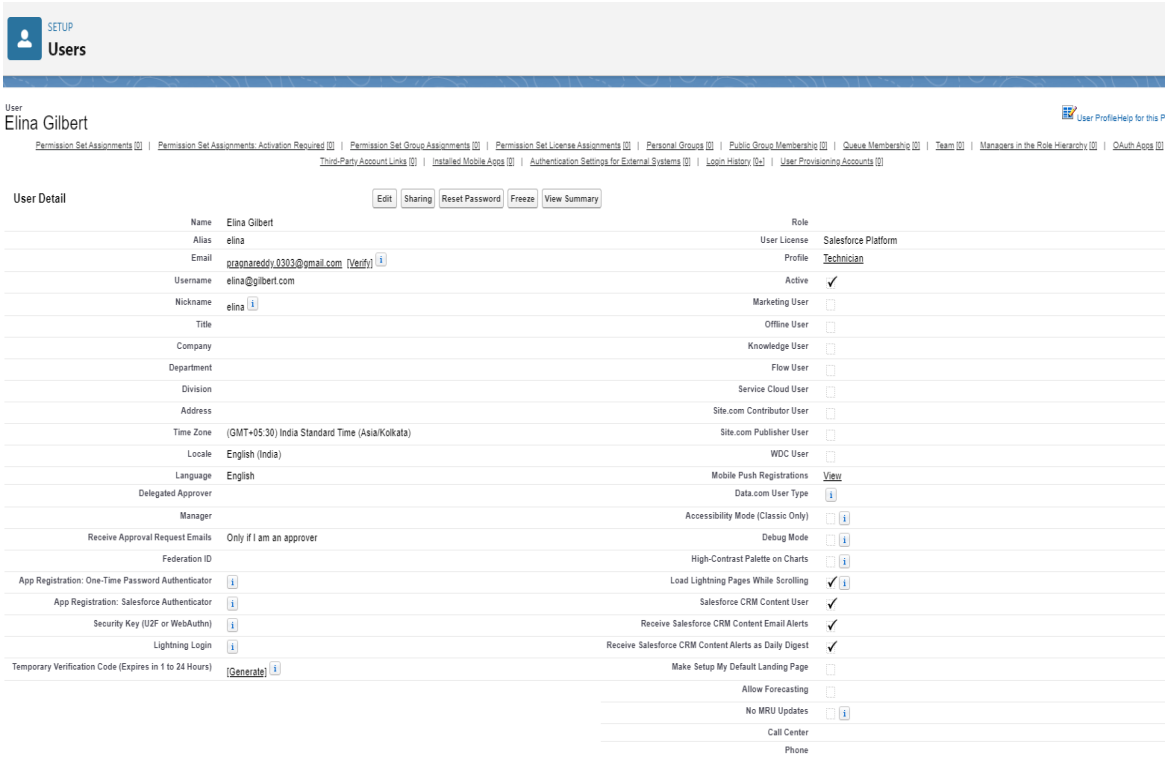


3. **Enter User Information**:
    - **First Name**: Enter "Elina" in the First Name field.
    - **Last Name**: Enter "Gilbert" in the Last Name field.
    - **Alias**: Provide a short alias for the user (e.g., "EGilbert").
    - **Email**: Enter your personal email address in the Email field.
    - **Username**: Enter a unique username in the format "text@text.text". Ensure it is in the proper email format but does not have to be a real email address.
    - **Nickname**: Enter a suitable nickname for the user.
    - **Role**: Optionally, select a role if required; this can be left blank if not needed.

- **User License**: Select "Salesforce Platform" from the User License dropdown menu.
- **Profile**: Choose "Technician" from the Profile dropdown menu to assign the relevant profile permissions to this user.



4. **Save the New User**:
   - After filling in all the required fields, click the "Save" button at the bottom of the page to create the user.



By following these steps, will successfully create a new user in Salesforce with the appropriate permissions and access rights as defined by the selected profile.

Task 8: Apex Triggerss

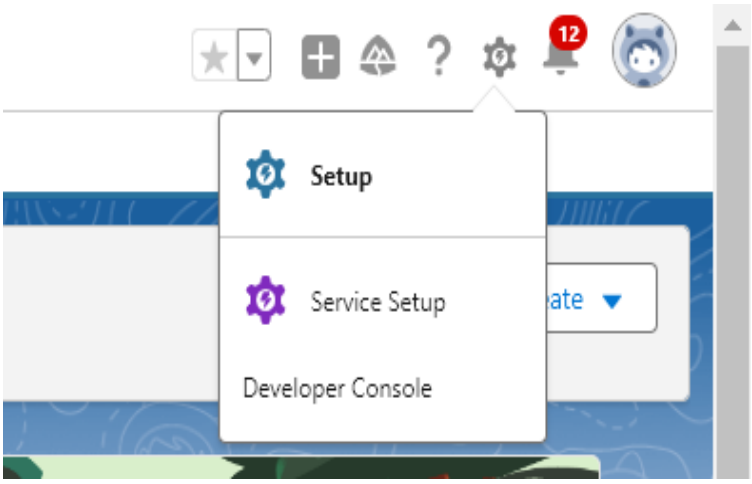
Apex Triggers in Salesforce are custom code snippets that automatically execute in response to specific events on Salesforce objects, such as inserting, updating, or deleting records. They allow developers to perform custom operations, enforce business rules, and automate processes. For instance, an `after insert` trigger can be used to send notifications or create related records, while an `after update` trigger might be used to log changes or update other records based on the new data.

*8.*1 Detailed Steps for Creating an Apex Class in Salesforce

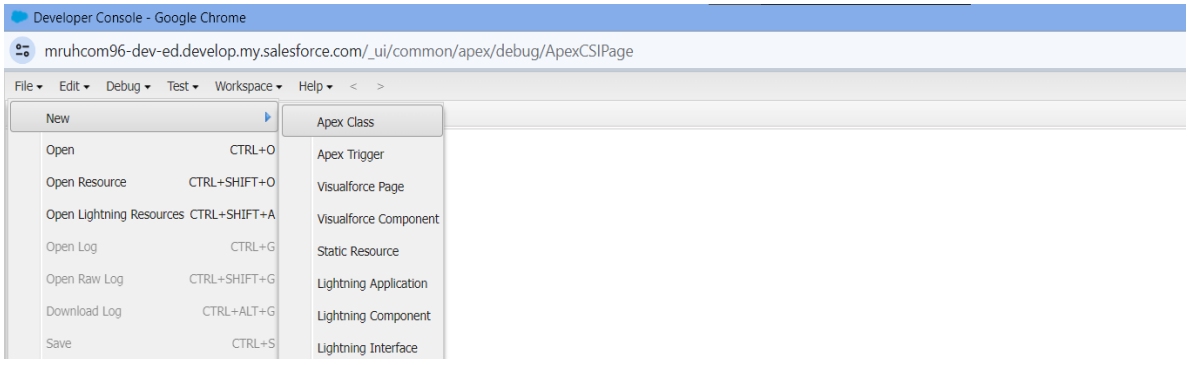
The `WorkOrderClass` Apex class processes a list of `WorkOrder__c` records to assign technicians based on service type and location. It maps work orders to technicians who match the criteria, creating `Assignment__c` records to link each work order with a

suitable technician. The class ensures that only available technicians with the necessary skills and location are assigned, and inserts the assignments into the database.

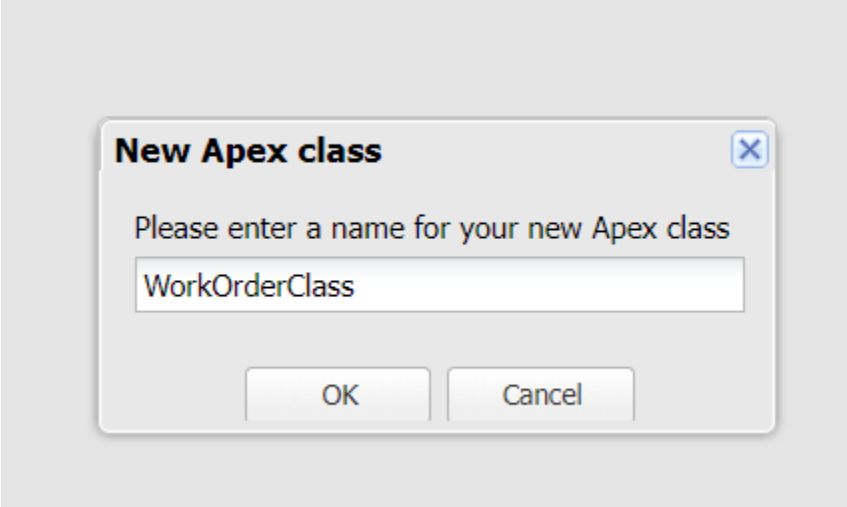
1. To access the Developer Console, log in to your Salesforce account, click the Gear icon (⚙) in the top-right corner to open the Setup menu, and select "Developer Console" from the dropdown, which will open a new window with the Developer Console interface.



2. To create a new Apex Class in the Developer Console, click "File" in the top-left menu, select "New," choose "Apex Class," enter "WorkOrderClass" as the class name in the prompt that appears, and click "OK" to create the class.



3. After creating the class, a new tab for "WorkOrderClass" will open in the Developer Console, where you can write provided source code into the editor.



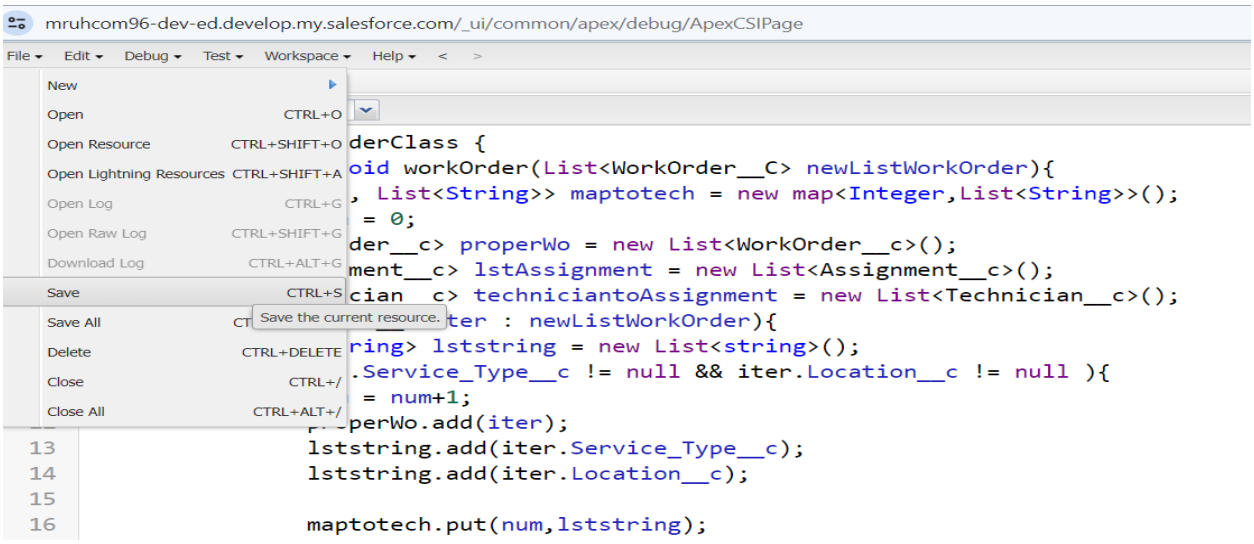
```
public class WorkOrderClass {
    public static void workOrder(List<WorkOrder_Sheet1__c> newListWorkOrder){
        Map<Integer, List<String>> maptotech = new map<Integer,List<String>>();
        integer num = 0;
        List<WorkOrder_Sheet1__c> properWo = new List<WorkOrder_Sheet1__c>();
        List<Assignment__c> lstAssignment = new List<Assignment__c>();
        List<Technician__c> techniciantoAssignment = new List<Technician__c>();
        for(WorkOrder_Sheet1__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, Availibility__c, Name__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.Availibility__c == 'Available'){
                techid.put(num2,t.Id);
            }
            }

        }
        integer num3 = 0;
        For(WorkOrder_Sheet1__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;
        }
    }
}
```

This code defines a class `WorkOrderClass` with a method `workOrder` that processes a list of `WorkOrder` records and assigns them to available technicians based on their skills and location.

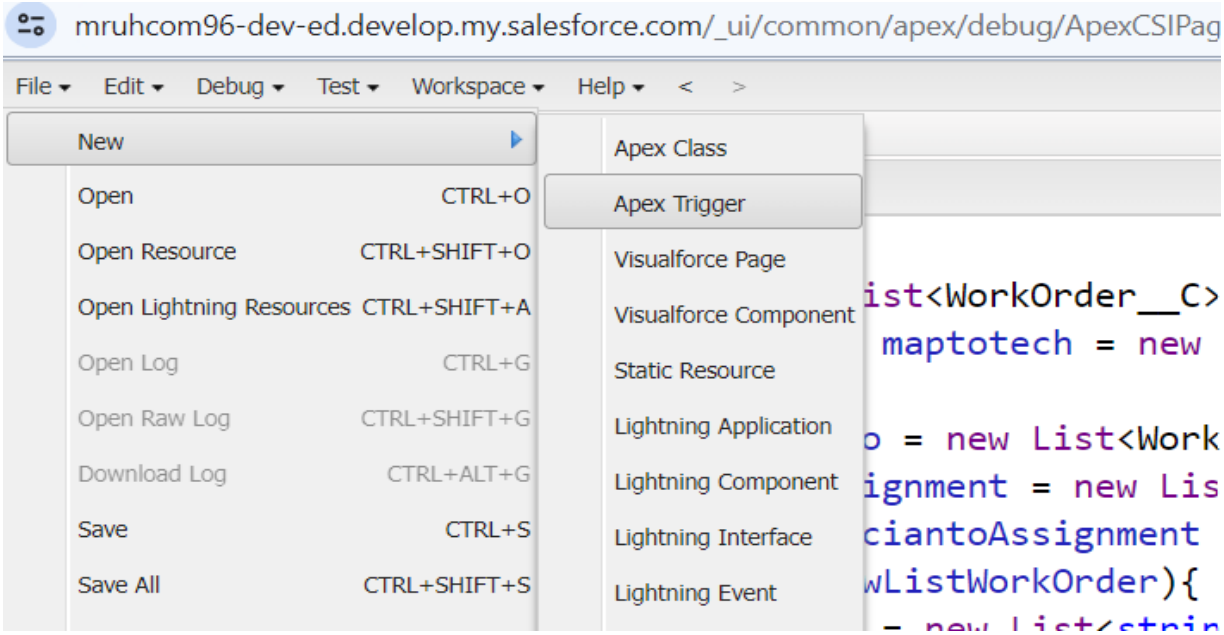
4. After entering the code, click on "File" in the Developer Console menu and Select "Save" to save your Apex class.



*8.*2 Create an Apex Trigger

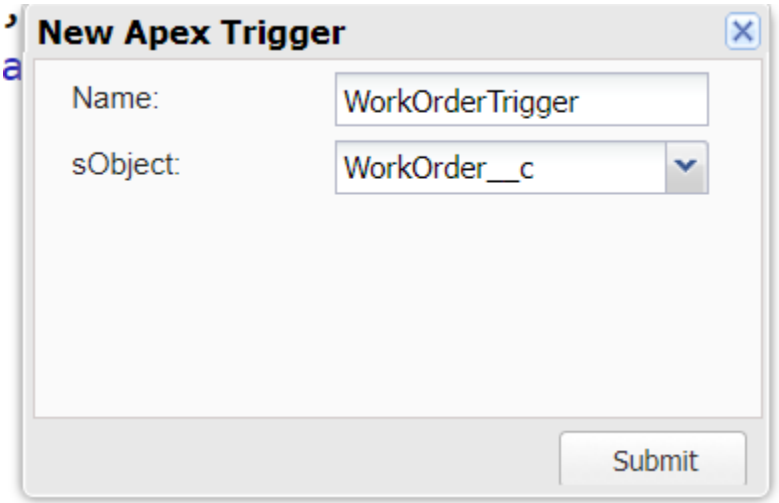
The `WorkOrderTrigger` Apex trigger executes the `WorkOrderClass.workOrder` method after a new `WorkOrder__c` record is inserted, automating the assignment of technicians based on the work order details.

1. Log in to Salesforce, click the Gear icon (⚙), select "Developer Console" from the Setup menu, and a new Developer Console window will open.
2. In the Developer Console, go to "File" in the top-left menu, select "New," and then choose "Apex Trigger" from the dropdown options.

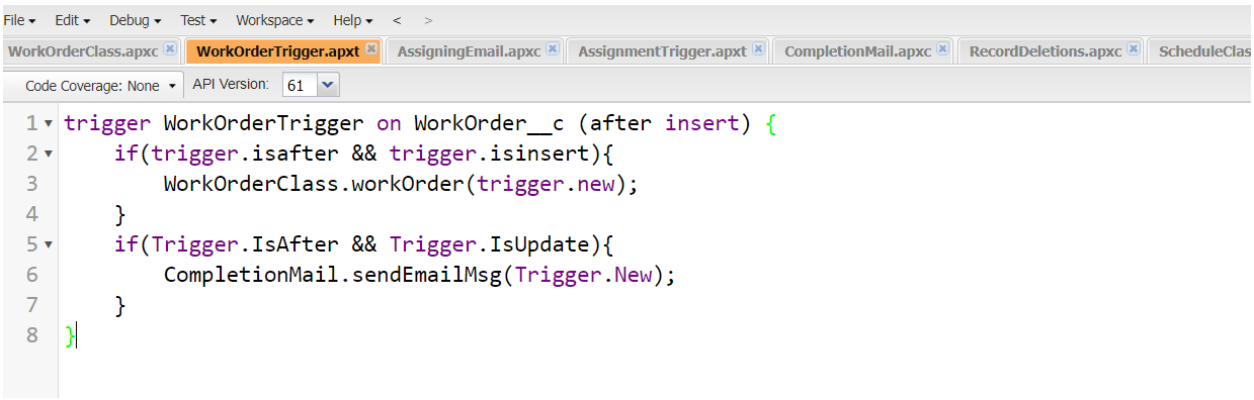


3. **Define the Trigger**:

- In the prompt that appears, enter "WorkOrderTrigger" as the name of your new Apex Trigger.
- From the dropdown menu labeled "sObject," select "WorkOrder__c" to associate the trigger with the WorkOrder object.
- Click "Submit" to create the trigger.



4. In the new "WorkOrderTrigger" tab that opens in the Developer Console, enter the provided source code into the editor.



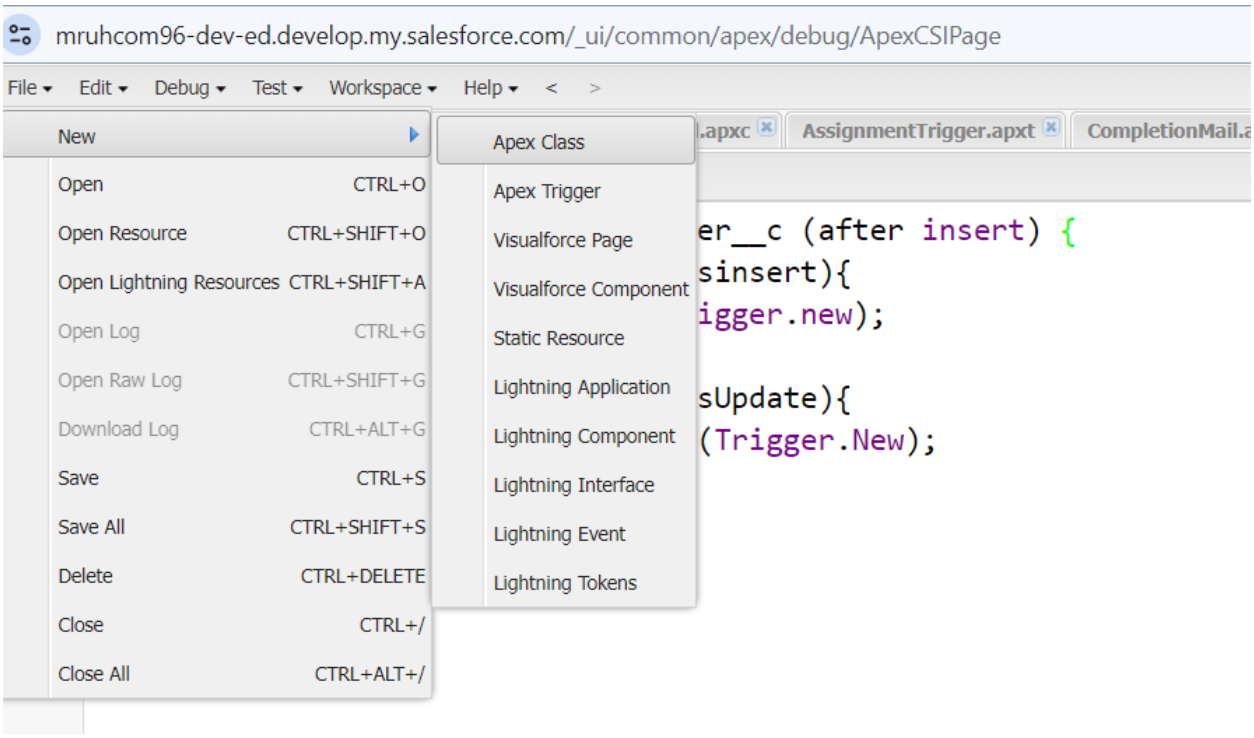
5. Once you've entered the code, save your work by clicking on "File" and then "Save" from the menu.

8.3 Create an Apex Class

The `AssigningEmail` class is used to send email notifications for assignment records. Its `sendEmailmsg` method takes a list of `Assignment__c` records, retrieves technician details from the `Technician__c` object, and creates email messages for each assignment. It sets the recipient to the technician's email address and includes assignment details in the email body. The class then sends the emails and handles any exceptions that may occur during the process, ensuring that technicians are informed of new assignments efficiently.

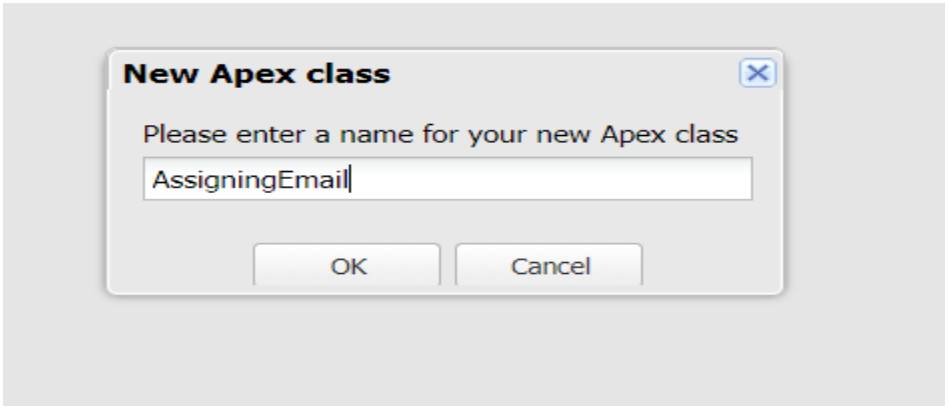
1. Log in to your Salesforce account, click on the Gear icon (⚙) in the top-right corner to open the Setup menu, and select "Developer Console" to open a new window with the Developer Console interface.
2. **Create a New Apex Class:**

- In the Developer Console, go to the top-left corner and click on "File."
- From the dropdown menu, select "New" and then choose "Apex Class."

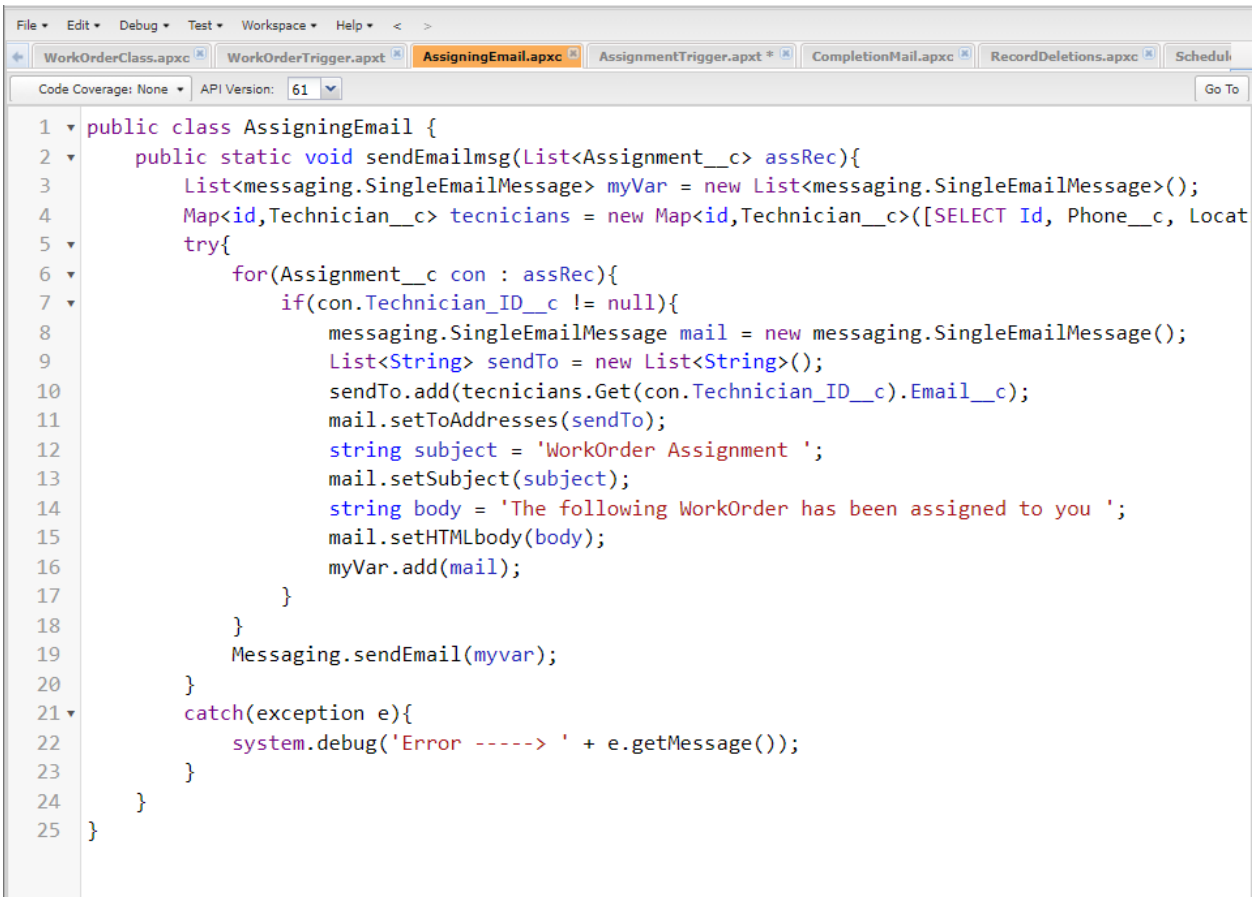


- A prompt will appear asking for the class name. Enter "AssigningEmail" as the name for the new Apex class.

- Click "OK" to create the class. A new tab will open in the Developer Console with a blank editor ready for the new class.



3. In the newly opened tab for "AssigningEmail," enter the following source code into the editor:



4. After entering the code into the editor, go to "File" in the top-left menu of the Developer Console.Click "Save" to save the new Apex class. Ensure there are no syntax errors or issues before saving.

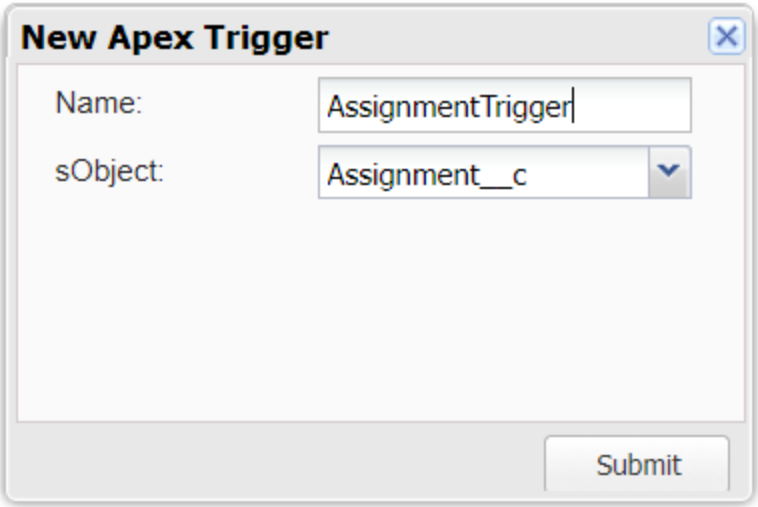
8.4 Create an Apex Trigger

The AssignmentTrigger is an Apex Trigger designed to automate email notifications upon the creation of new Assignment__c records. Configured to execute after the insert operation, it invokes the sendEmailmsg method of the AssigningEmail class. This method processes the newly inserted assignment records and sends out email notifications to the relevant technicians. By automating the notification process, the trigger ensures timely communication and improves operational efficiency.

1. Log in to your Salesforce account, click the Gear icon (⚙) in the top-right corner

to open the Setup menu, and select "Developer Console" from the dropdown to open it in a new window.

2. In the Developer Console, click on "File" in the top-left menu and Select "New" from the dropdown options, then choose "Apex Trigger."
3. To configure the Apex Trigger, enter "AssignmentTrigger" as the Trigger Name in the prompt, select "Assignment__c" from the sObject dropdown, and click "Submit" to create the trigger.



4. A new tab will open in the Developer Console for the "AssignmentTrigger"enter the following source code into the editor.

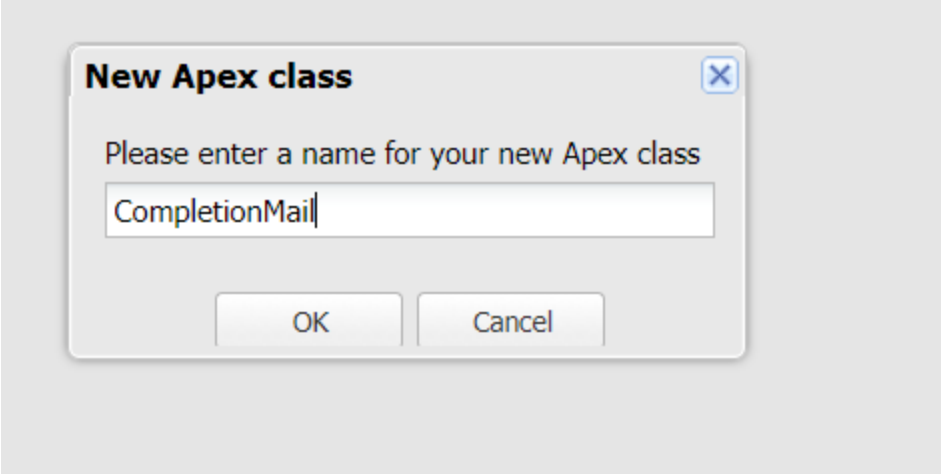
5. Click on "File" and then "Save" to save your changes.

By following these steps, you will successfully create an Apex Trigger that executes after a new record is inserted into the "Assignment__c" object, triggering an email notification through the AssigningEmail class.

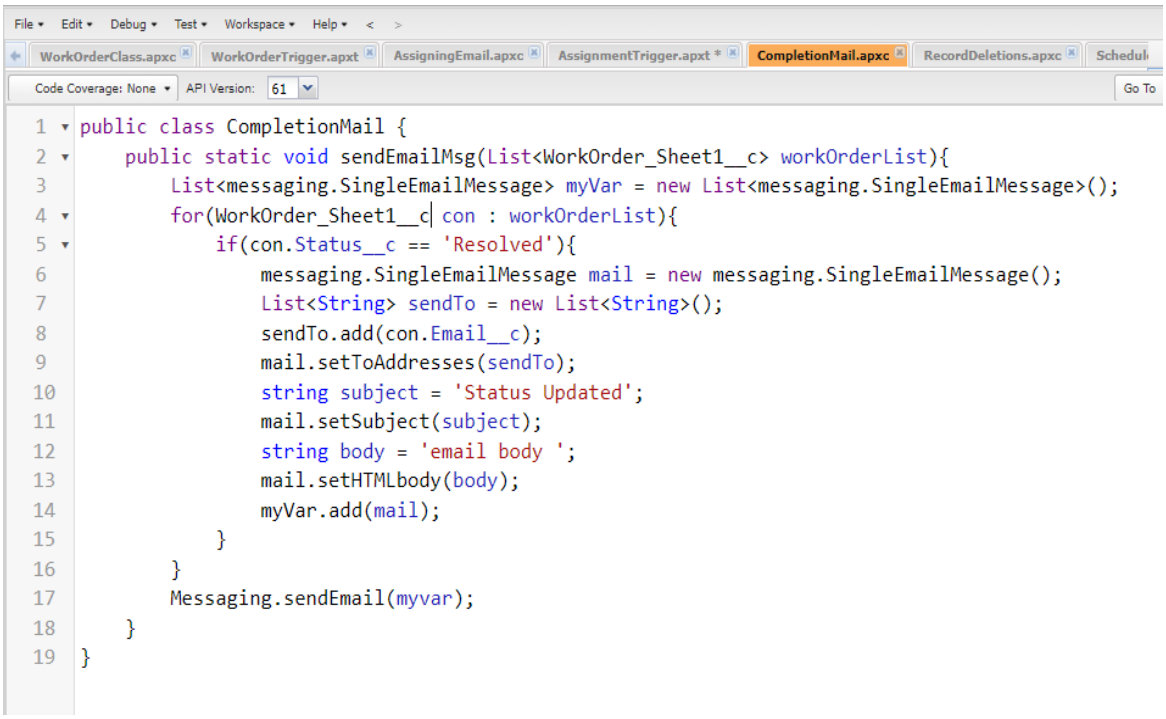
8.5 Create an Apex Class

The `CompletionMail` Apex Class is designed to automate email notifications for `WorkOrder__c` records with a status of 'Resolved'. This class includes the `sendEmailMsg` method, which processes a list of `WorkOrder__c` records. It checks each record's status, and if it is 'Resolved', the method prepares an email notification. The email is configured with a subject and body, and is sent to the address specified in the `Email__c` field of each work order. This approach ensures that stakeholders are promptly informed of resolved work orders, facilitating effective communication and updates.

1. To access the Developer Console in Salesforce, log in to your Salesforce account, click on the Gear icon (⚙) in the top-right corner to open the Setup menu, and select "Developer Console" from the dropdown menu to open a new console window.
2. In the Developer Console, click on "File" in the top-left menu, choose "New," select "Apex Class," enter "CompletionMail" in the prompt, and click "OK" to create the class.



3. A new tab will open for the "CompletionMail" class and enter the following source code into the editor:



4. After pasting the code, save the class by clicking on "File" and then "Save" in the Developer Console.

This process creates an Apex Class named "CompletionMail" that sends an email notification when a work order's status is updated to 'Resolved'.

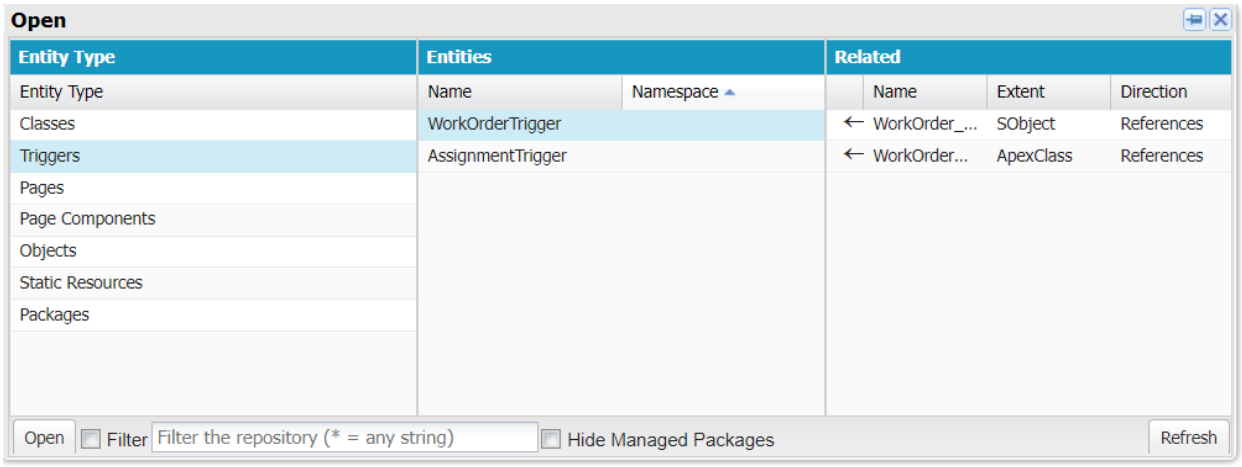
8.6 Create an Apex WorkOrderTrigger

The `WorkOrderTrigger` Apex Trigger is designed to handle operations on `WorkOrder__c` records. It activates after a record is inserted or updated. When a new `WorkOrder__c` record is inserted, the trigger invokes the `workOrder` method from the `WorkOrderClass`, which processes the new work orders and assigns technicians based on their skills and availability. If an existing work order is updated, the trigger calls the `sendEmailMsg` method from the `CompletionMail` class to send an email notification if the work order's status is 'Resolved'. This trigger integrates data processing and notification functionality, automating updates and communication related to work orders.

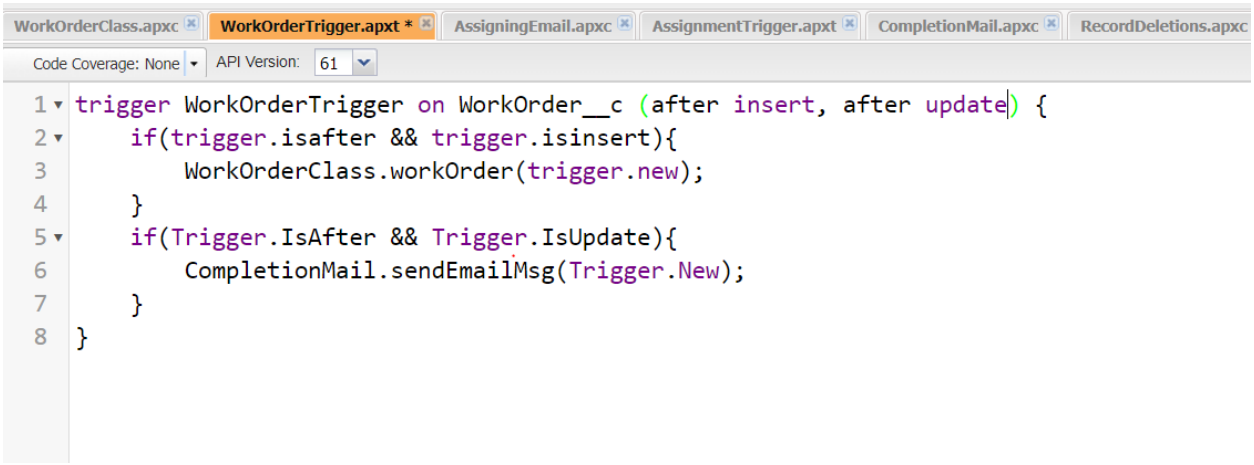
1. Log in to your Salesforce account, click on the Gear icon (⚙) in the top-right corner to open the Setup menu, and select "Developer Console" from the

dropdown menu to open a new window.

2. In the Developer Console, click on "File" in the top-left menu, select "Open" from the dropdown, choose "Triggers" from the popup window, then select “WorkOrderTrigger” and click on “Open” to access the trigger file.



3. In the newly opened "WorkOrderTrigger" tab, copy and paste the following source code into the editor:




```
trigger WorkOrderTrigger on WorkOrder__c (after insert, after update) {
    if(trigger.isafter && trigger.isinsert){
        WorkOrderClass.workOrder(trigger.new);
    }
    if(Trigger.IsAfter && Trigger.IsUpdate){
        CompletionMail.sendEmailMsg(Trigger.New);
    }
}
```

4. After entering the code, click on "File" and select "Save" to save the changes to your trigger.

By following these steps, you have successfully created and configured the "WorkOrderTrigger" to handle after insert and after update events in Salesforce.

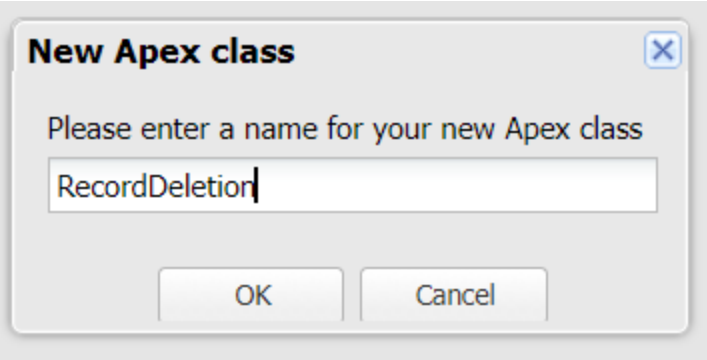
8.7 Create an Asynchronous Apex Class

The `RecordDeletion` Apex Class is an asynchronous class designed to handle the deletion of `Assignment__c` records that meet specific criteria. It implements the `Database.Batchable<Sobject>` interface, allowing it to process large data volumes in batches. The class starts by defining a query to select records where the `Completion_Date__c` is within the last 30 days. This query is used in the `start` method to create a `Database.QueryLocator`. In the `execute` method, the class processes each batch of records and deletes those that match the criteria, ensuring that outdated records are efficiently removed. Finally, the `finish` method is called after all batches have been processed; while it does not perform any actions in this implementation, it can be customized for additional tasks. This class automates the cleanup process, helping maintain data quality and manageability.

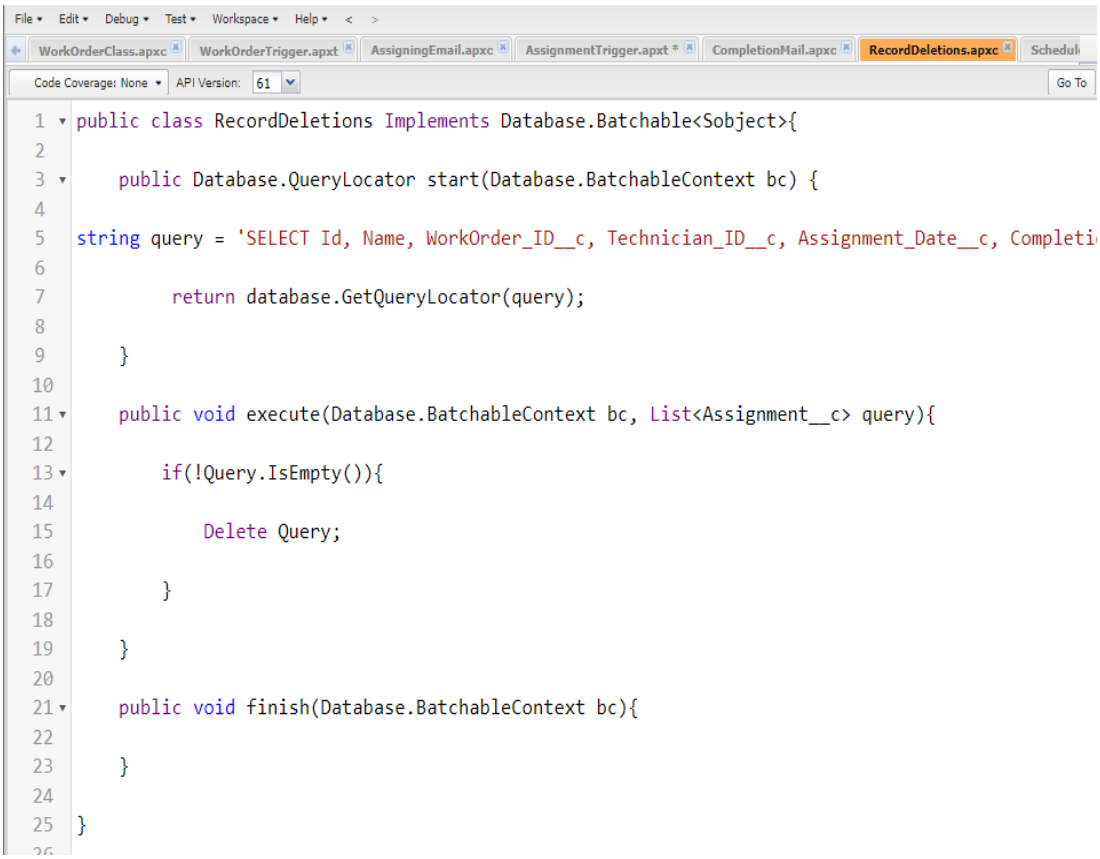
To create an asynchronous Apex class for deleting WorkOrder records based on specific criteria, follow these detailed steps:

1. Click on the Gear icon (⚙) in the top-right corner of the page to open the Setup menu and Select "Developer Console" from the dropdown menu to open a new console window.

2. In the Developer Console, click on "File" in the top-left menu, select "New" and choose "Apex Class," then enter "RecordDeletion" as the class name and click "OK" to create it.



3. In the new tab for the "RecordDeletion" class, enter the following source code:



4. Click on "File" and then "Save" to save the Apex class.

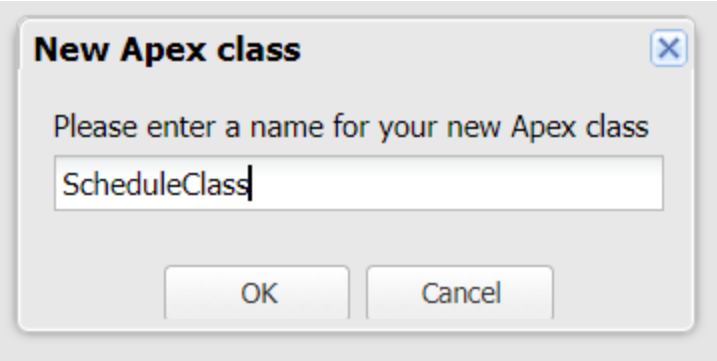
This class implements the Database.Batchable interface, allowing it to process large data sets asynchronously. It queries for Assignment__c records where the Completion_Date__c is within the last 30 days and the Status__c is 'Resolved', and deletes those records.

8.8 Create an Apex Schedule Class

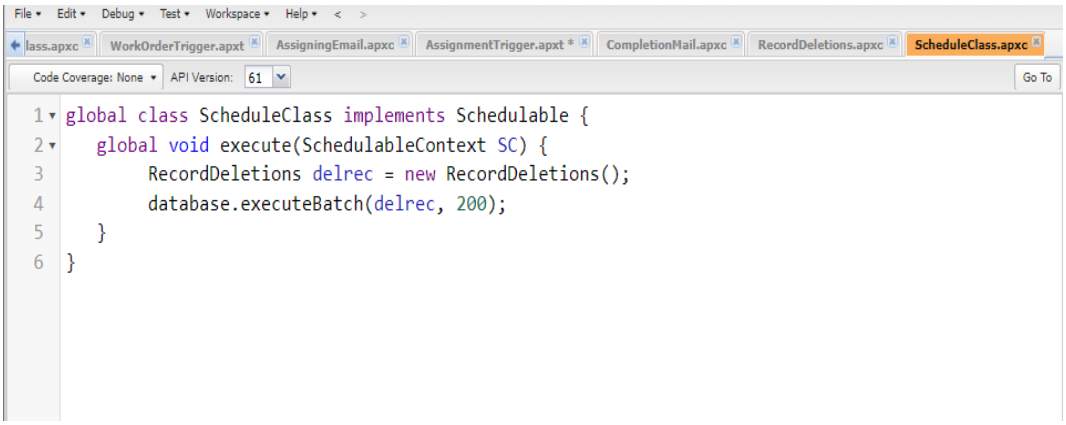
The `ScheduleClass` Apex Class is designed to automate batch processing in Salesforce by implementing the `Schedulable` interface. This allows the class to be scheduled to run at specific intervals. In its `execute` method, the class creates an instance of the `RecordDeletions` batch class, which handles the deletion of `Assignment__c` records that are over 30 days old and marked as 'Resolved'. By using the `database.executeBatch` method with a batch size of 200, the class processes records in manageable chunks, ensuring efficiency and compliance with Salesforce governor limits. This automation helps maintain data accuracy and system performance

by regularly cleaning up outdated records.

1. Log in to Salesforce, click on the Gear icon (⚙) in the top-right corner to open the Setup menu, and select "Developer Console" from the dropdown to open a new console window.
2. In the Developer Console, click "File" in the top-left menu, select "New" and choose "Apex Class," enter "ScheduleClass" as the class name, and click "OK" to create the class.



3. In the new Apex Class editor, enter the following code:

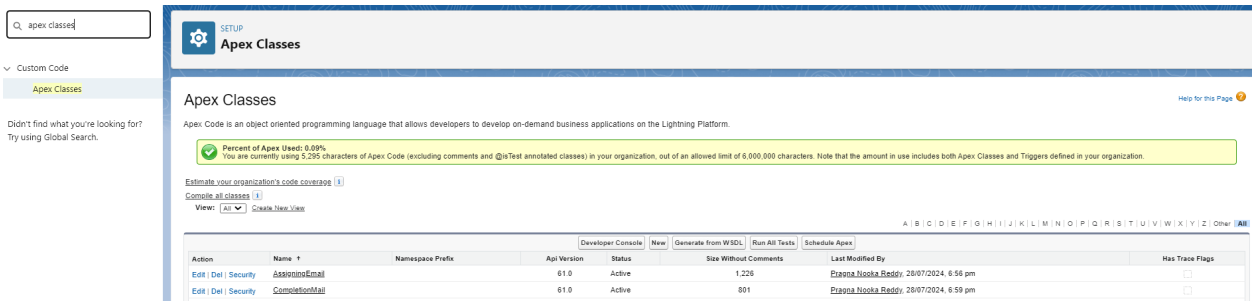


4. Click on "File" and then "Save" to save your changes.

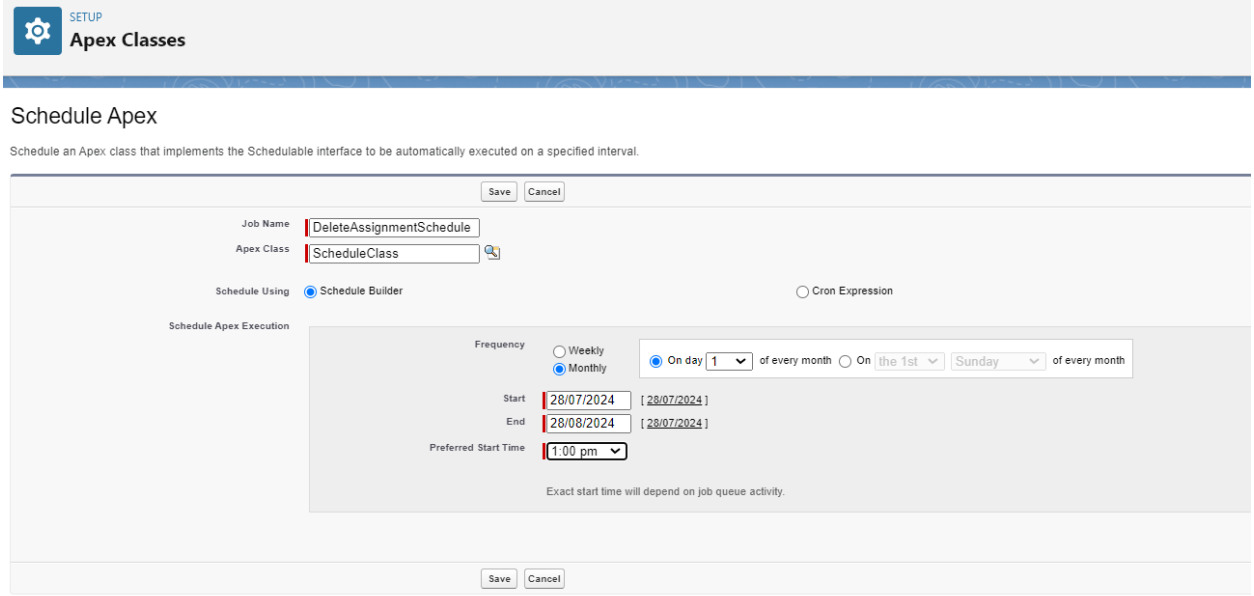
8.9 Create a Schedule Apex

To schedule the `ScheduleClass` Apex class, start by navigating to the Setup page in Salesforce and searching for "Apex Classes" using the Quick Find box. Select "Apex Classes" from the search results. Next, click on "Schedule Apex" to open the scheduling interface. Enter the job name as `DeleteAssignmentSchedule` and select `ScheduleClass` from the lookup icon for the Apex Class field. Set the frequency to "Monthly" and choose a preferred start time that suits your needs. Finally, click "Save" to schedule the Apex class, ensuring it runs automatically according to the specified schedule.

1. From the Setup page, search for "Apex Classes" in the Quick Find box and click on it from the search results.
2. Click on "Schedule Apex."



3. **Enter Job Details**:
    - **Job Name**: Enter "DeleteAssignmentSchedule."
    - **Apex Class**: Select "ScheduleClass" using the lookup icon.
    - **Frequency**: Choose "Monthly."
    - **Preferred Start Time**: Select any preferred time.



4. Click "Save" to finalize the scheduling.

Task 9: Reports & Dashboards

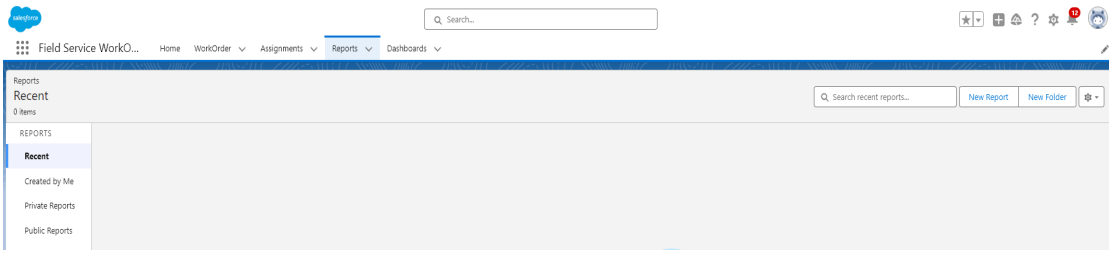
Salesforce Reports and Dashboards are essential tools for data analysis and visualization within the Salesforce platform. Reports enable users to extract and organize data from Salesforce objects into structured formats such as tables, charts, and graphs. They allow for deep analysis through customization options like filters and groupings, making it easier to focus on specific metrics or trends. Dashboards build on this by providing a consolidated, graphical view of key performance indicators from multiple reports. They present data through various components like charts, gauges, and tables, offering a snapshot of performance and trends. Together, Reports and Dashboards help users monitor performance, gain insights, and make informed decisions, transforming raw data into actionable information that drives business strategies and operational efficiency.

9.1 Report Creation in Salesforce

To create a report in Salesforce, go to the **Reports** tab and click **New Report**. Choose a report type from the available categories or search for one, then click **Start Report**. Customize the report by adding fields from the left pane and grouping data by **WorkOrder ID**. Adjust filters and sorting as needed. Finally, save your report or run it to view the results. Note that the appearance and options might vary based on your Salesforce setup and data.

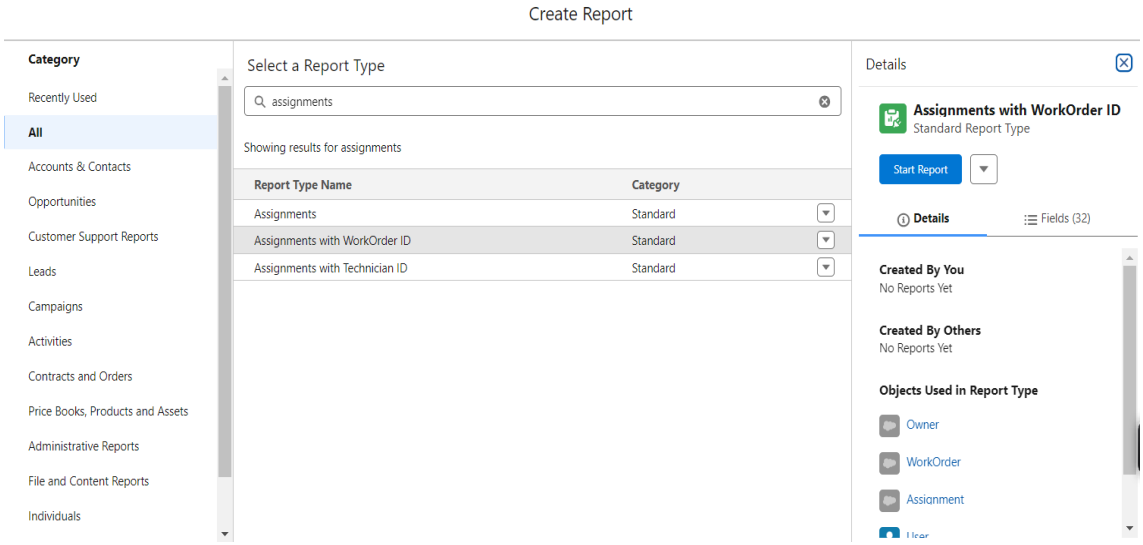
Salesforce reports provide a structured way to visualize and analyze data within the Salesforce platform, allowing users to monitor performance, track metrics, and make informed decisions. The process of creating a report involves several steps:

1. Begin by navigating to your Salesforce app. Locate and click on the **Reports** tab, which is typically positioned on the main navigation bar.



2. Initiate the report creation process by clicking the **New Report** button. This option is usually available at the top right corner of the Reports tab page.

3. Choose the appropriate report type from the categories or the report type panel. You can also use the search panel to find a specific type of report. Click **Start Report** once you have selected the desired type to proceed with customization.

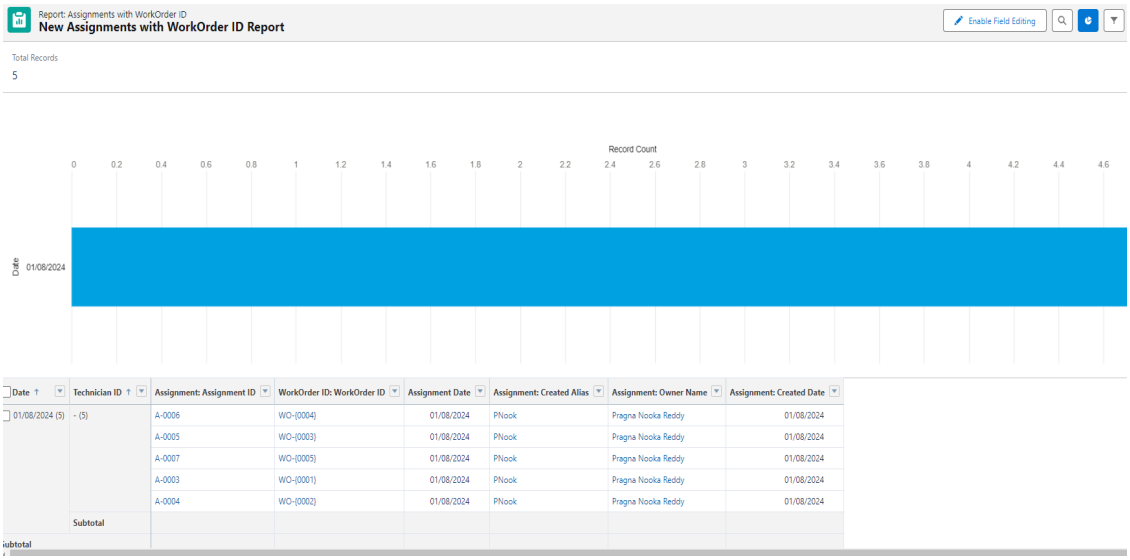


4. **Customize Your Report:**
    - **Add Fields:**
        - In the report builder, use the left pane to drag and drop fields into the report layout. These fields will define what data is displayed in

the report.

- **Group Data:**
    - Organize your data by grouping it based on the **WorkOrder ID**. This grouping helps in sorting and analyzing the data more effectively by categorizing it under specific work orders.
- **Apply Filters and Sorts:**
    - Customize the report further by applying filters to narrow down the data to relevant criteria. Adjust the sorting options to arrange the data in the desired order.

5. **Save or Run the Report:**
    - After configuring the report to meet your needs, you can either save it for future use or run it immediately to view the results. Click **Save** to store the report with a specific name and description or click **Run** to generate and view the report based on the current settings.



The "New Assignments with WorkOrder ID Report" is an essential tool in our project management process. It provides a comprehensive overview of work order assignments, including details such as the technician assigned, a brief description of the assignment, and the corresponding WorkOrder ID. The report also tracks the assignment date, the timestamp when it was created, and identifies the assignment owner. This allows for effective tracking of tasks, ensuring they are assigned promptly and to the correct personnel. Additionally, the report features a bar graph that visually represents key metrics, such as the distribution of assignments over time or by technician. This visual tool aids in quickly assessing workload and identifying potential

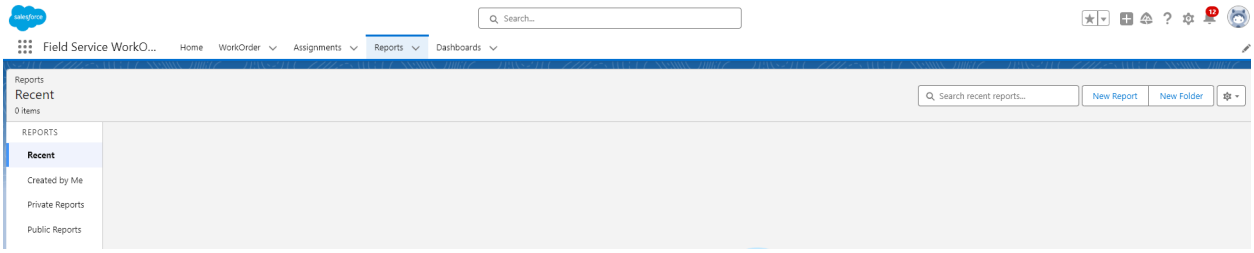
bottlenecks. Overall, this report is crucial for maintaining organization, ensuring accountability, and facilitating efficient project execution.

9.2 Creating Reports

9.2.1 Create a WorkOrders Status Report:

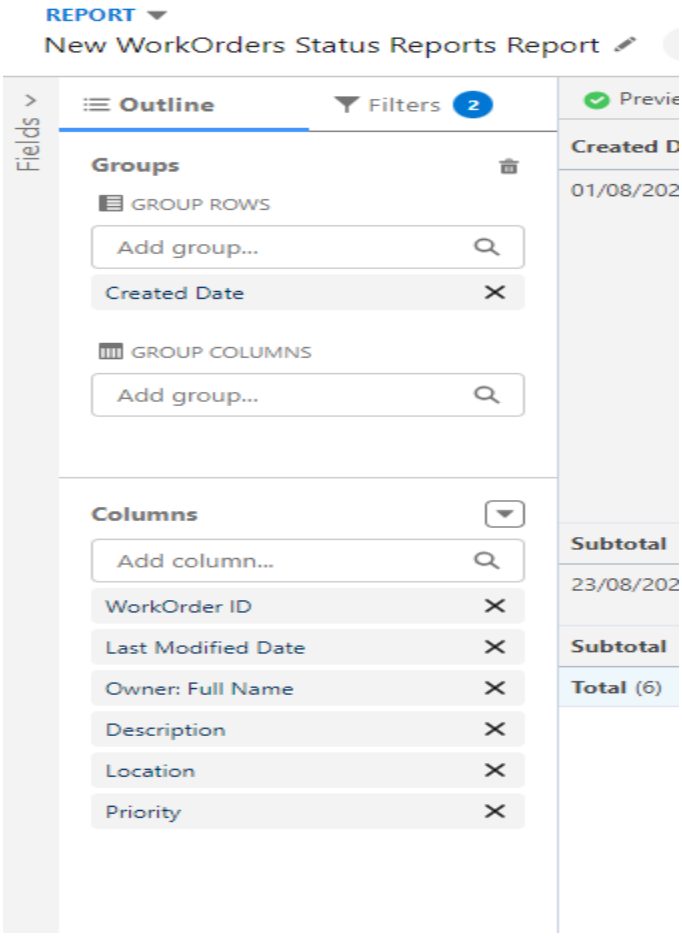
Start by navigating to the **Reports** tab in your Salesforce app and click on **New Report**. Choose **WorkOrders Status Reports** from the available report types to focus on work order statuses. Customize your report by dragging and dropping relevant fields, grouping the data by **WorkOrder ID**, and applying necessary filters. Save or run the report to analyze work order statuses in detail.

1. Go to the Salesforce app where you want to create the report and Click on the **Reports** tab from the main navigation menu.
2. Click the **New Report** button to start the report creation process, choose the desired report type from the available options or search for it, and then click **Start Report** to proceed.

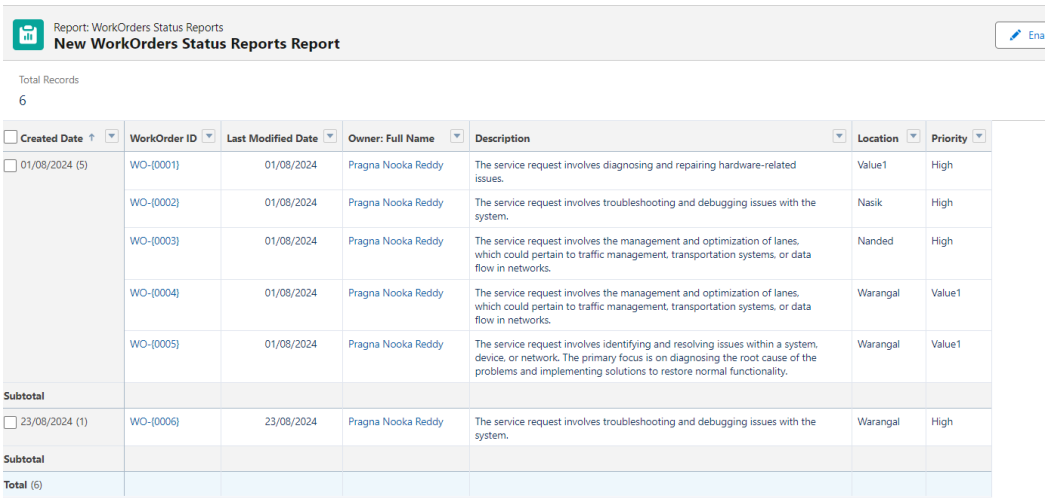


3. **Customize the Report**:

- Use the left pane to add fields to the report. Drag and drop the fields you want to include.
- Apply any necessary filters or groupings. For example, group the data by **WorkOrder ID** if needed.
- Adjust the report layout and format according to your requirements.



4. **Save or Run the Report**:
    - Click **Save** to save the report with a specific name and description for future access.
    - Alternatively, click **Run** to generate and view the report immediately.



Report: WorkOrders Status Reports
New WorkOrders Status Reports Report

Total Records
6

Created Date	WorkOrder ID	Last Modified Date	Owner: Full Name	Description	Location	Priority
01/08/2024 (5)	WO-{0001}	01/08/2024	Pragna Nooka Reddy	The service request involves diagnosing and repairing hardware-related issues.	Value1	High
	WO-{0002}	01/08/2024	Pragna Nooka Reddy	The service request involves troubleshooting and debugging issues with the system.	Nasik	High
	WO-{0003}	01/08/2024	Pragna Nooka Reddy	The service request involves the management and optimization of lanes, which could pertain to traffic management, transportation systems, or data flow in networks.	Nanded	High
	WO-{0004}	01/08/2024	Pragna Nooka Reddy	The service request involves the management and optimization of lanes, which could pertain to traffic management, transportation systems, or data flow in networks.	Warangal	Value1
	WO-{0005}	01/08/2024	Pragna Nooka Reddy	The service request involves identifying and resolving issues within a system, device, or network. The primary focus is on diagnosing the root cause of the problems and implementing solutions to restore normal functionality.	Warangal	Value1
Subtotal						
23/08/2024 (1)	WO-{0006}	23/08/2024	Pragna Nooka Reddy	The service request involves troubleshooting and debugging issues with the system.	Warangal	High
Subtotal						
Total (6)						

5. **Review and Adjust**:
    - Review the generated report to ensure it meets your needs.
    - Make any adjustments or refinements as necessary before finalizing

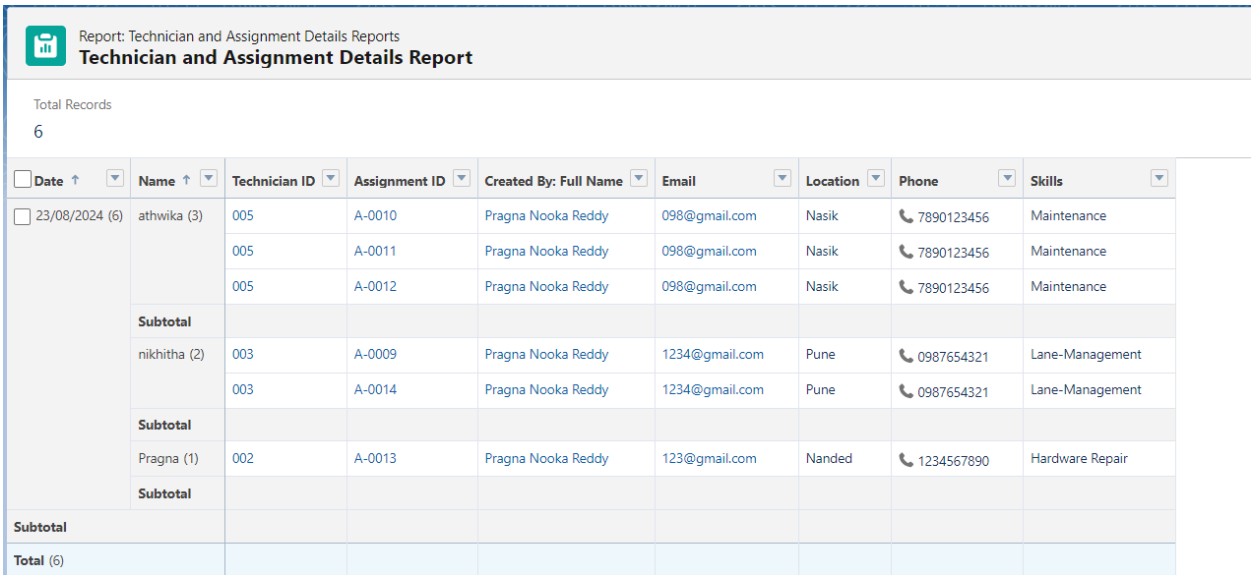
9.2.2 Create a Technician and Assignment Details Report:

Access the **Reports** tab and initiate a new report by selecting **Technician and Assignment Details Reports** as the report type. Customize this report by including fields related to technicians and their assignments, adjusting groupings and filters as needed. Save the report with a descriptive name, or run it immediately to gain insights into technician assignments and their details.

1. Go to the Salesforce app where you want to create the report and Click on the **Reports** tab from the main navigation menu.
2. Click on the **New Report** button.

3. In the report type selection panel, either browse through the available categories or use the search bar and Locate and select the **Technician and Assignment Details Reports** report type.
4. Click on **Start Report** to open the report builder interface.
5. **Customize the Report:**
    - Use the **Fields** pane on the left to drag and drop relevant fields into the report layout.
    - Organize the data by adding appropriate groupings or filters related to technicians and assignments.
    - Apply any necessary filters to focus on specific details or data ranges.

6. **Save or Run the Report:**
    - Click on **Save** to store the report. Provide a name, description, and folder location for the report.
    - Alternatively, click on **Run** to view the report immediately.



Report: Technician and Assignment Details Reports
Technician and Assignment Details Report

Total Records
6

Date	Name	Technician ID	Assignment ID	Created By: Full Name	Email	Location	Phone	Skills
23/08/2024 (6)	athwika (3)	005	A-0010	Pragna Nooka Reddy	098@gmail.com	Nasik	7890123456	Maintenance
		005	A-0011	Pragna Nooka Reddy	098@gmail.com	Nasik	7890123456	Maintenance
		005	A-0012	Pragna Nooka Reddy	098@gmail.com	Nasik	7890123456	Maintenance
	Subtotal							
	nikhitha (2)	003	A-0009	Pragna Nooka Reddy	1234@gmail.com	Pune	0987654321	Lane-Management
		003	A-0014	Pragna Nooka Reddy	1234@gmail.com	Pune	0987654321	Lane-Management
	Subtotal							
	Pragna (1)	002	A-0013	Pragna Nooka Reddy	123@gmail.com	Nanded	1234567890	Hardware Repair
	Subtotal							
Subtotal								
Total (6)								

The "Technician and Assignment Details Report" provides a comprehensive overview of technicians assigned to tasks, detailing their names, unique IDs, assignment IDs, and contact information. It includes their locations (e.g., Nasik, Pune, Nanded) and specific skills (e.g., Maintenance, Lane Management, Hardware Repair), ensuring efficient task allocation and communication. This report is essential for managing resources effectively and ensuring tasks are matched with the appropriate technician skills.

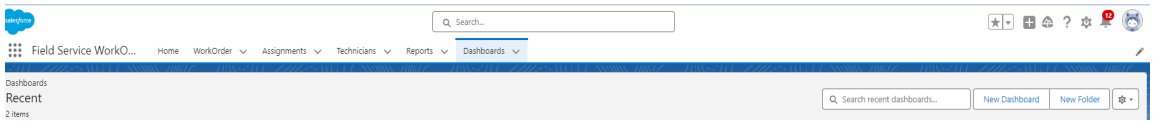
9.3 Dashboard Creation in Salesforce

A Salesforce Dashboard is a powerful tool that provides a visual representation of key business data and metrics. By using dashboards, users can consolidate and display data from multiple reports into a single, interactive view, enabling them to quickly analyze performance, monitor trends, and make informed decisions. Dashboards can be customized with various components, such as charts, tables, and gauges, which help to visualize data in different formats. Users can create dashboards to track sales performance, customer service metrics, project progress, and more, making them an essential feature for efficient data management and strategic planning within the Salesforce platform.

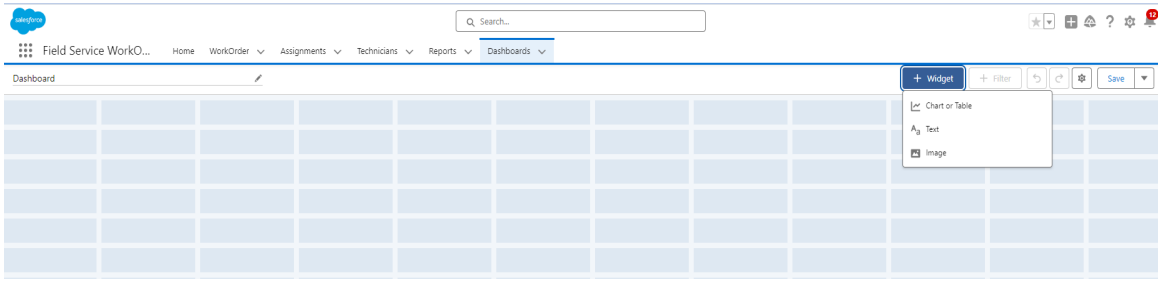
To create a dashboard in Salesforce, follow these detailed steps:

1. Open your Salesforce application and click on the "Dashboards" tab from the top navigation bar. This will take you to the main dashboard page where you can create and manage your dashboards.

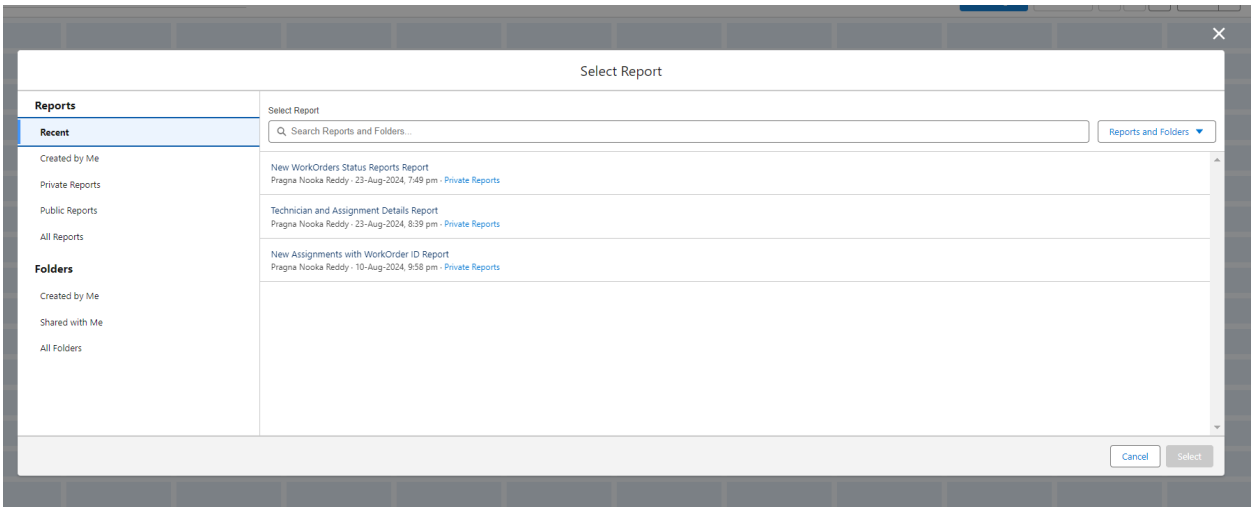
2. On the Dashboard page, click on the "New Dashboard" button. A pop-up window will appear where you need to provide a name for your dashboard. Enter a relevant name that reflects the purpose of the dashboard, and then click on "Create" to start building your new dashboard.



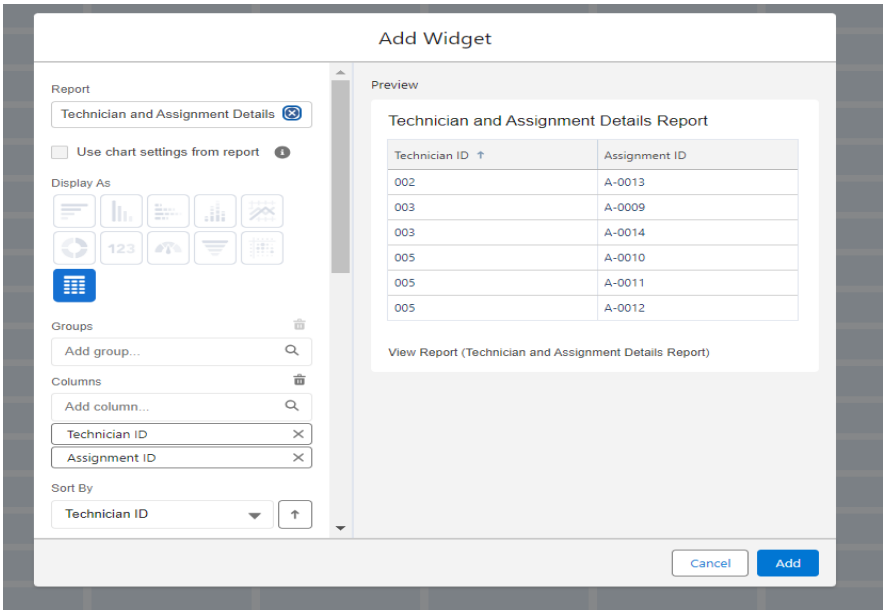
3. Once the dashboard is created, you will be taken to the dashboard editor. Here, click on the "widget" button to start adding visual elements to your dashboard. Components can include charts, graphs, tables, or gauges that display key data metrics.



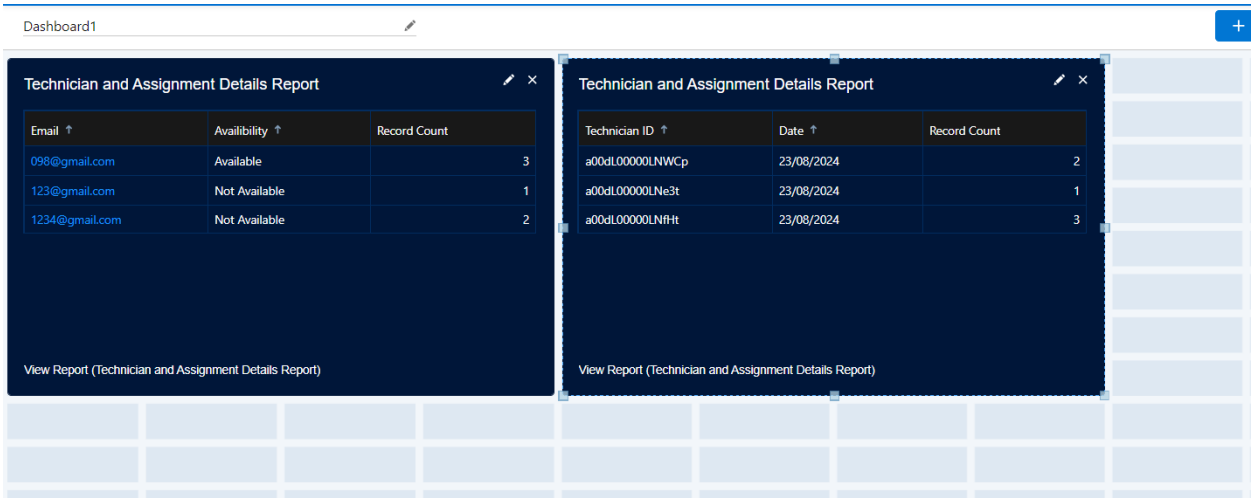
4. After clicking "widget," you will be prompted to select a report to use as the data source for your component. Choose one of the reports you created in the previous steps, such as "WorkOrders Status Reports" or "Technician and Assignment Details Reports." Select the desired report and click "Select" to confirm.



5. Configure the selected component by choosing the type of visualization (e.g., bar chart, pie chart, line graph) that best represents the data. Adjust the settings and layout options to customize the appearance of the component according to your preferences.



6. Once you have configured the component, click on the "Add" button to place it onto your dashboard. You can reposition and resize the component as needed to fit the layout of your dashboard.
7. After adding all the necessary components and arranging them as desired, click on the "Save" button to save your dashboard. Finally, click on "Done" to exit the dashboard editor and view your completed dashboard.

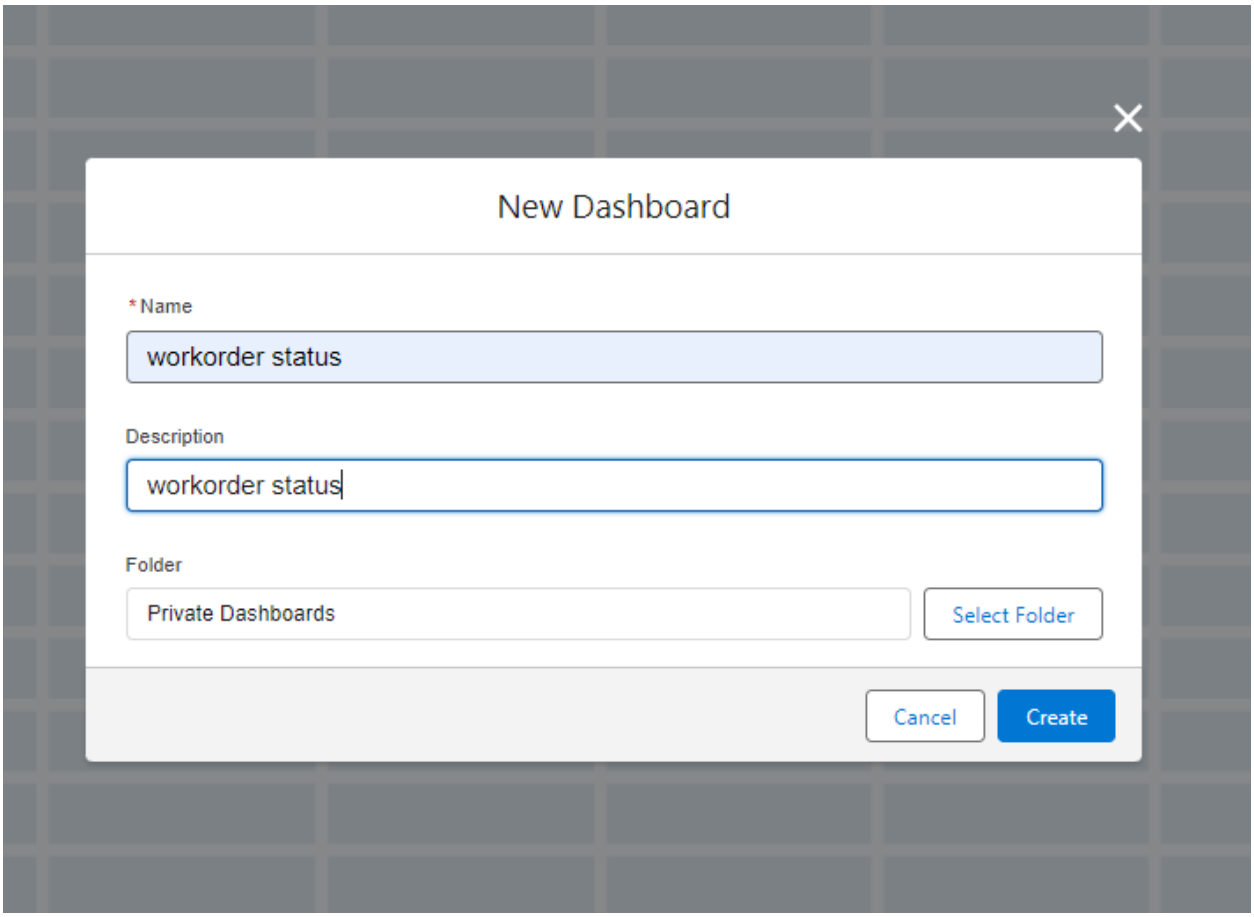


9.4 Creating work order status Dashboard

To create a Salesforce Dashboard that displays the details of completed work order statuses in a vertical bar graph, follow these detailed steps:

1. Open your Salesforce application and click on the "Dashboards" tab from the main menu. If the tab isn't visible, you can find it by searching in the App Launcher.

2. Click on the "New Dashboard" button. A prompt will appear asking you to provide a name for your dashboard. Enter a descriptive name, such as " Work Order Status " and click "Create."



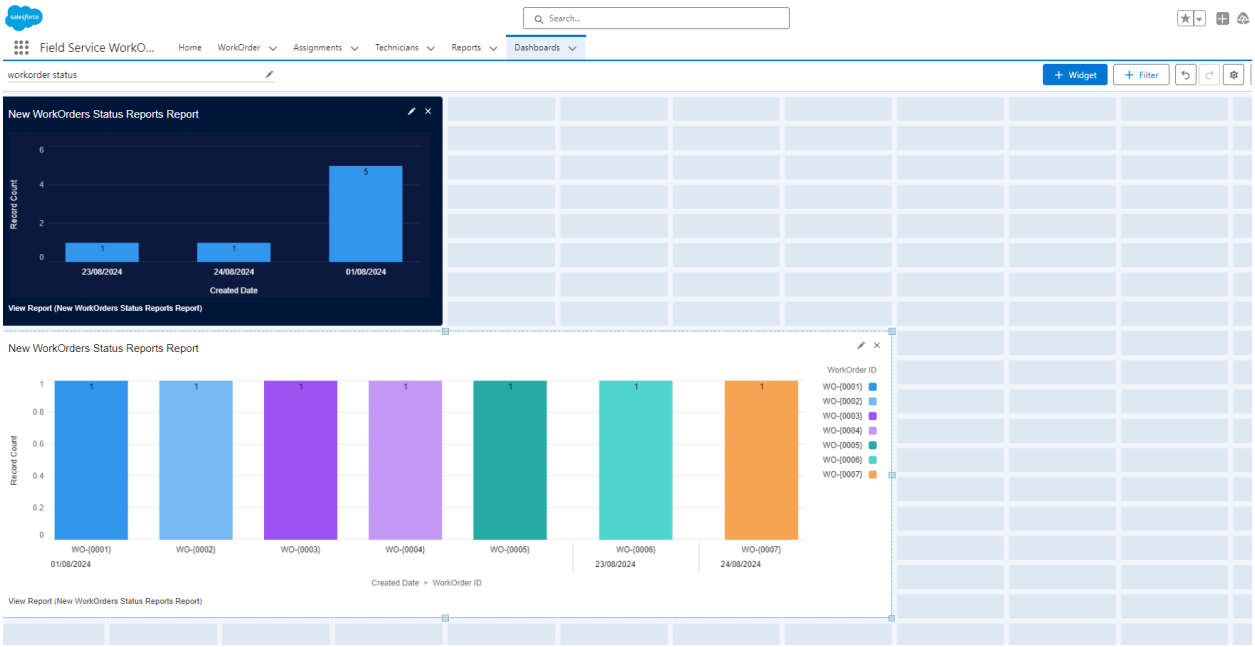
3. In the dashboard editor, click on "Widget." This action will open a panel where you can select the type of component you want to add.

4. Choose the report that contains the work order data you want to visualize. This should be the report created in previous activities that shows the status of work orders. Select the appropriate report from the list and click "Select."

5. In the component type selection, choose "Vertical Bar Chart" as the display format. This will allow you to visually represent the completed work order statuses in a bar graph format.

6. Configure the vertical bar chart by setting the X-axis to display the different statuses and the Y-axis to display the number of work orders for each status. You can also customize the chart colors, labels, and title to make the dashboard more informative and visually appealing.

7. After configuring the bar graph, click "Add" to include it in your dashboard. Then, click on "Save" to save the changes to your dashboard.



Conclusion

This project highlights the effective use of Salesforce tools such as Apex classes, triggers, reports, and dashboards to enhance operational efficiency and decision-making. By implementing Apex classes and triggers, we automated the assignment and notification processes, ensuring timely communication and streamlined workflows. The creation of detailed reports, such as the "WorkOrders Status Report" and the "Technician and Assignment Details Report," provided valuable insights into work order statuses and technician assignments, enabling better resource allocation and performance monitoring.

Furthermore, the dashboards created, including the vertical bar graph displaying completed work order statuses, offered a powerful visual representation of key metrics, allowing stakeholders to quickly assess project progress and make informed decisions. Overall, this project has demonstrated the importance of leveraging Salesforce's capabilities to improve data management, automate processes, and enhance overall project performance. The tools and strategies implemented will continue to provide significant value in managing work orders and assignments efficiently, ultimately contributing to improved productivity and customer satisfaction.