Field service workorder optimization

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, availibility, and skills. The system employs a prioritization algorithm, focusing on assigning tasks to technician. 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.

Salesforce:

This is a overall platform. It's a cloud-based CRM and customer service platform that helps businesses manage their sales, marketing, customer service, and other processes.

Creating Developer Account:

This is likely a step in the process of getting started with Salesforce development. It involves setting up a developer account to access the Salesforce platform and tools for building custom applications.

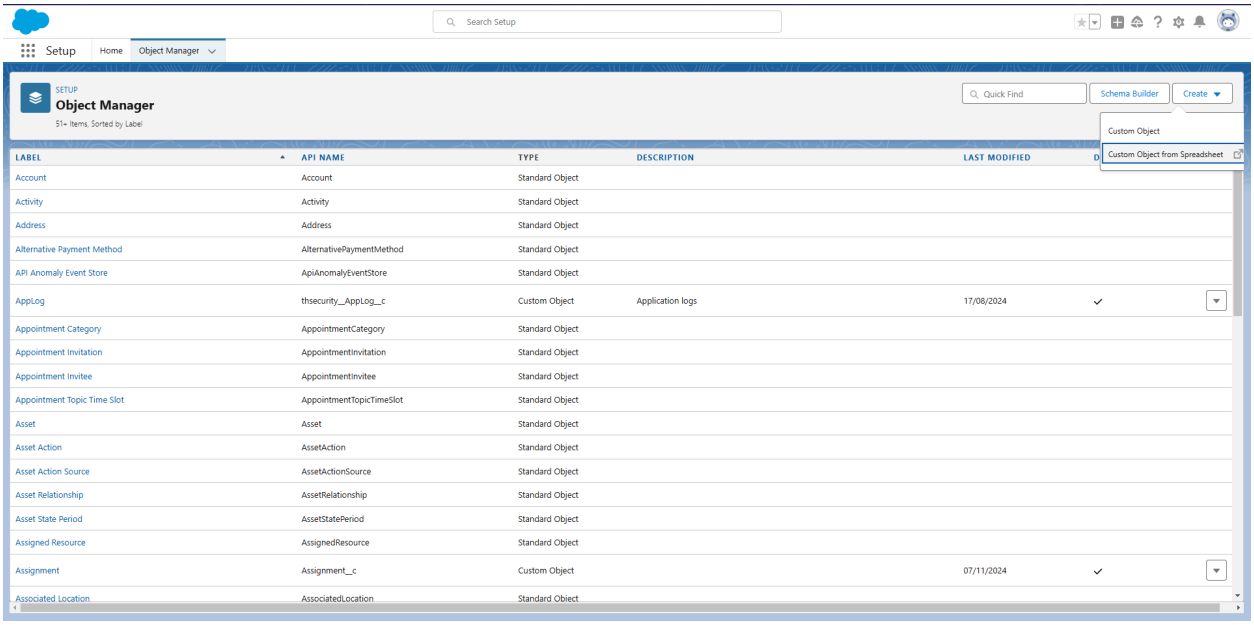
Account Activation:

This is likely a step in the process of activating a Salesforce account. It involves verifying your email address and completing any necessary setup steps to fully access the platform.

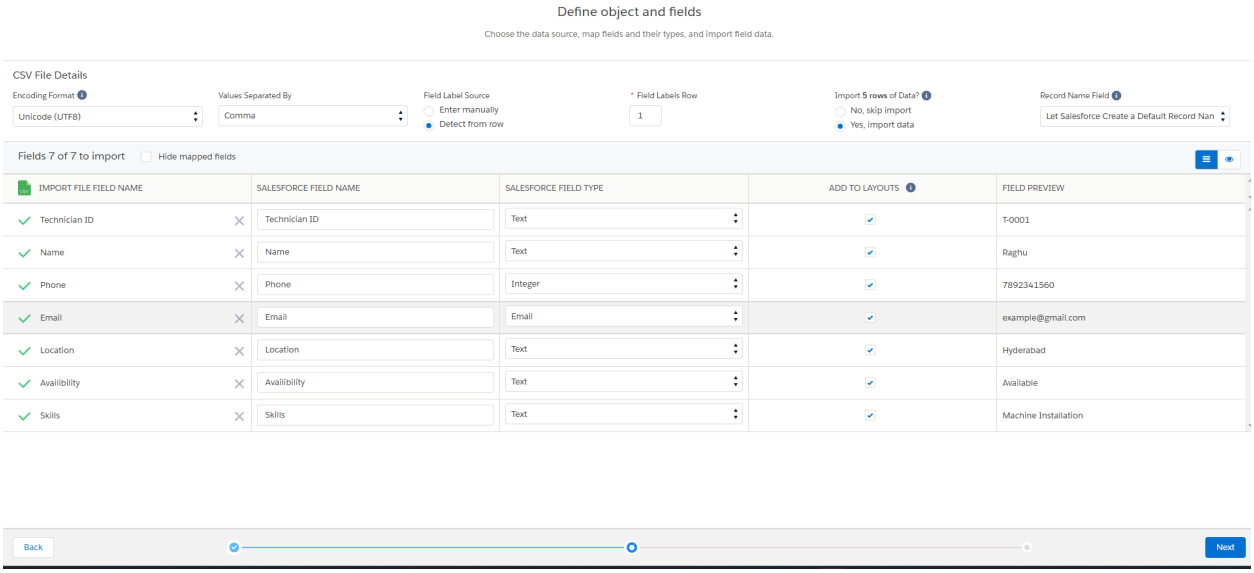
Object:

In the context of Salesforce, an "Object" refers to a database table that stores information about a specific entity, such as a customer, product, or opportunity. It's a fundamental

building block in Salesforce development.



Create Technician Object : This option allows you to create a new Object in Salesforce to store information about your technicians. This can include fields for their name, contact information, certifications, skills, availability, and other relevant details. Having a Technician Object helps you manage and track your field service technicians effectively.



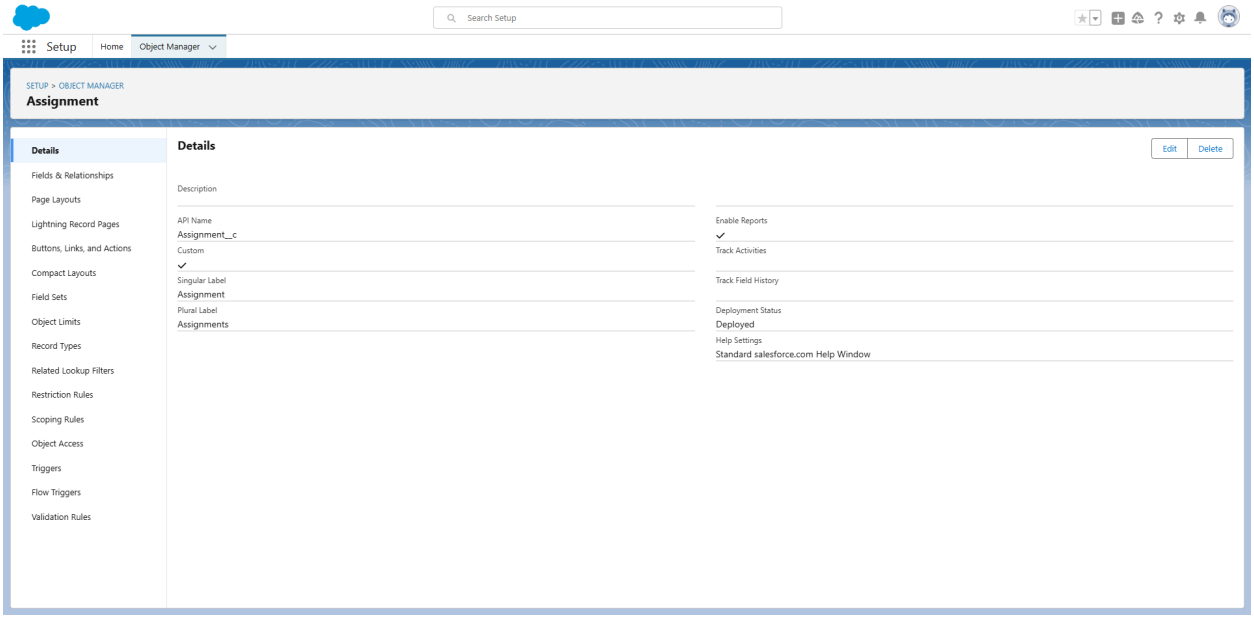
Create Work Order Object:

This option allows you to create a new Object in Salesforce to store information about work orders. This can include fields for the customer, service type, priority, description, status, assigned technician, and other relevant details. The Work Order Object is crucial for managing and tracking the progress of service requests.

Create Assignment Object:

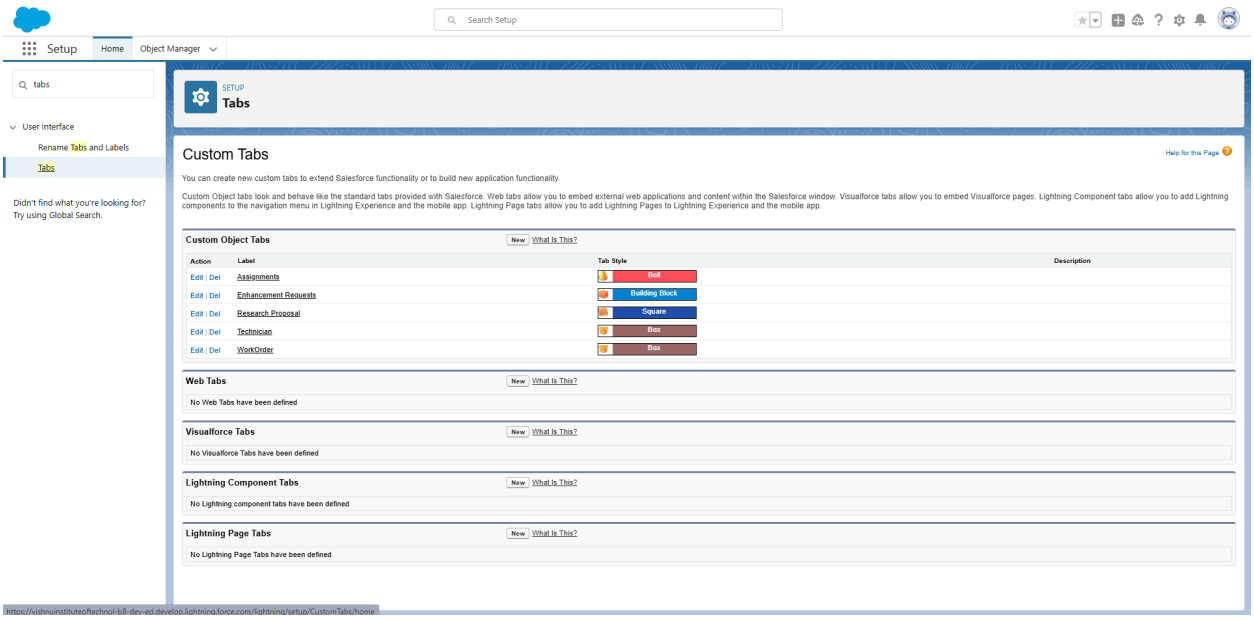
This option allows you to create a new Object in Salesforce to store information about

assignments. This can include fields for the work order, assigned technician, start time, end time, status, and other relevant details. The Assignment Object helps you track the work assigned to technicians and monitor their progress.



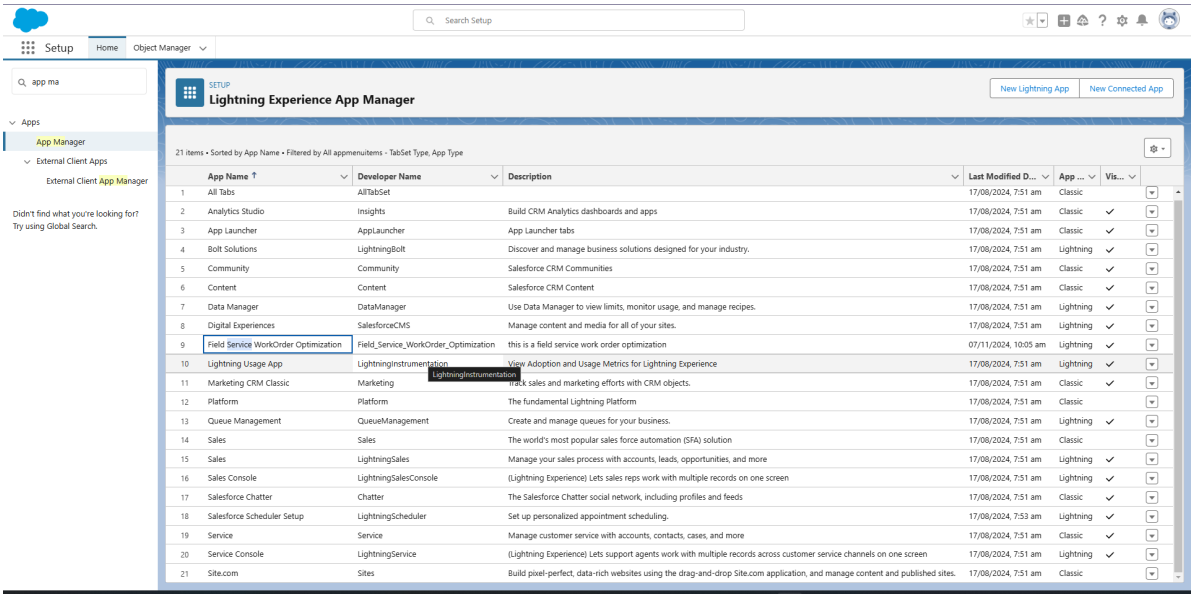
Tabs:

Tabs in Salesforce are used to organize and access different parts of your application. They can be used to display different Objects, reports, dashboards, or other functionalities. By creating tabs for your Technician, Work Order, and Assignment Objects, you can easily access and manage the information related to them.
Here , we will create object named assignment.



The Lightning App:

Salesforce Lightning is a modern, mobile-ready user interface that provides a faster, more intuitive user experience. It's designed to help users work more efficiently and effectively. Create A Lightning App:This option allows you to create a new Lightning App. A Lightning App is a collection of components that work together to provide a specific functionality. You can create Lightning Apps for various purposes, such as sales, marketing, customer service, and more.



Fields & Relationships:

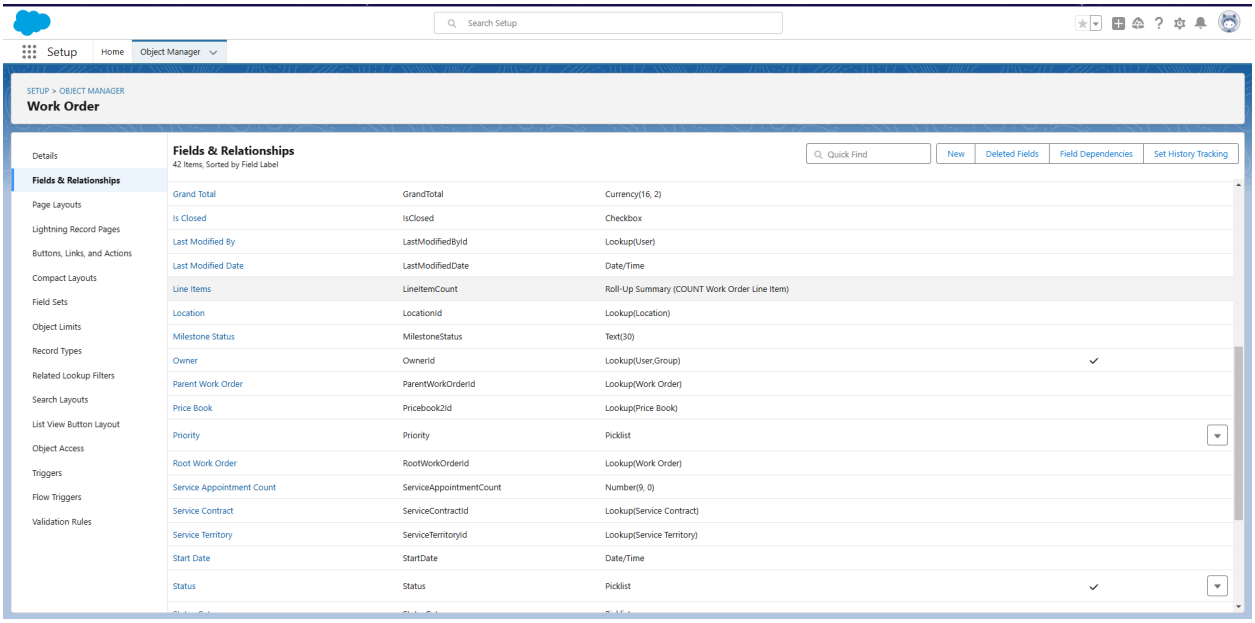
Fields and relationships are fundamental building blocks in Salesforce, allowing you to define the structure and relationships between different data elements.

Creating Lookup Field in Assignment Object:
A lookup field is a type of field that allows you to reference records from another Object. In this case, you'll be creating a lookup field in the Assignment Object to reference records from the Technician Object. This will allow you to associate each assignment with a specific technician.

Manage Your Picklist Values:
A picklist field is a type of field that allows users to select values from a predefined list. In this context, you'll be managing the picklist values for fields like "Status" and "Priority" in your Objects. This ensures consistency and standardization in data entry.

Creating Formula Field in WorkOrder Object:
A formula field is a field that automatically calculates a value based on other fields in the same Object or related Objects. In this case, you'll be creating a formula field in the WorkOrder Object to calculate the estimated duration of the work order based on the complexity and other relevant factors.

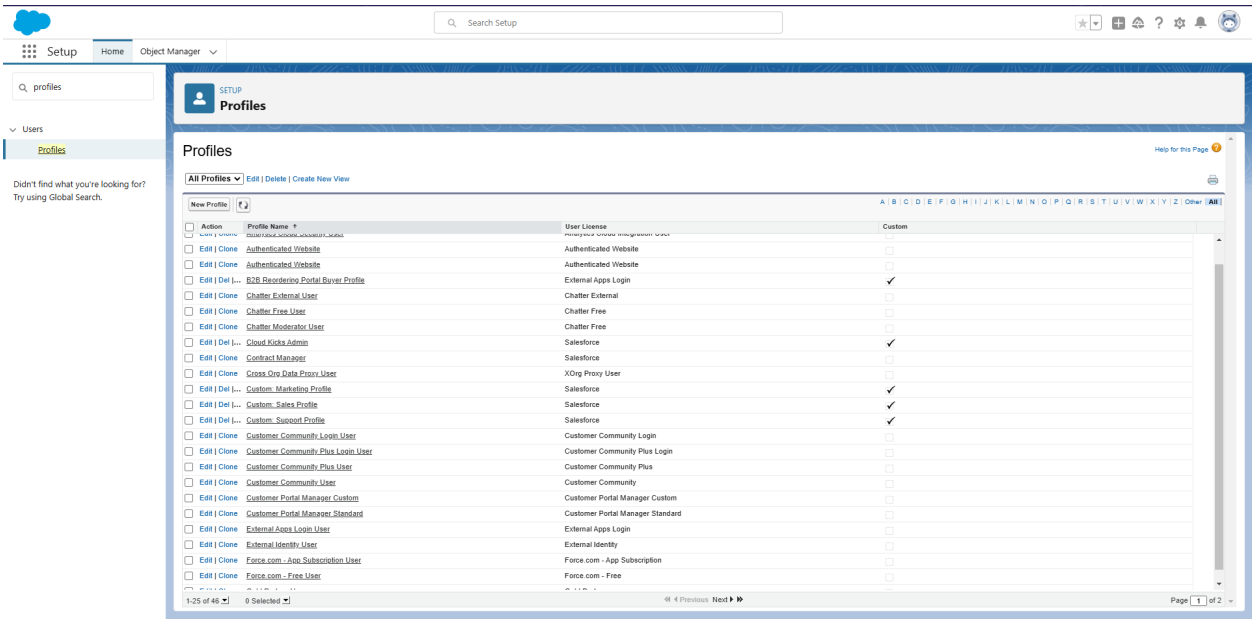
Creating Remaining Fields for the Respective Objects:
This refers to creating any additional fields that are needed for your specific use case. This could include fields like "Description," "Start Date," "End Date," "Notes," and more.



Profiles:

Profiles in Salesforce control the access and permissions of users to different parts of the system. You can create different Profiles for different roles within your organization, such as sales reps, marketing managers, customer service agents, and technicians.

Technician Profile:
This is a specific Profile that is designed for field service technicians. It grants them access to the necessary Objects, fields, and reports to perform their job duties. This Profile can be configured to restrict access to sensitive data and limit the actions that technicians can perform.



Users:

Users are the individuals who have access to your Salesforce org. Each user is associated with a Profile, which determines their level of access to the system.
We are going to create a user with the following details

1. First Name : Elina
2. Last Name : Gilbert
3. Alias : Give a 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 : <No role>
8. User license : Salesforce Platform
9. Profiles : Technician

Apex Class:

A Building Block of Salesforce Development*

In Salesforce, an Apex class is a blueprint or template for creating objects that encapsulate data and behavior. It's a powerful tool that allows developers to extend the functionality of the platform beyond standard configuration.

Key Components of an Apex Class:

Methods: Functions that perform specific tasks within the class.

Variables: Data storage elements that can hold values.

Constructors: Special methods used to initialize objects.

Static Initialization Blocks: Code blocks that execute when the class is first loaded.

Why Use Apex Classes?

1. Custom Business Logic: Create custom business logic to automate tasks, validate data, and enforce business rules.

2. Trigger Events: Write code that executes automatically when specific events occur, such as record creation or update**.**

3. Web Services: Build web services to integrate Salesforce with external systems.

4. Batch Apex: Process large datasets efficiently in batches.

5. Scheduled Apex: Execute code at specific intervals or on a schedule.

Basic Structure of an Apex Class:

```
public class MyClassName {
  // Class variables
   public String myVariable;
// Constructor
   public MyClassName() {
      // Constructor logic
   }
// Methods
   public void myMethod() {
      // Method logic
   }
}
```

Apex Trigger:

Apex Triggers are pieces of Apex code that execute automatically in response to certain events, such as record creation, updates, or deletions. They can be used to automate tasks, enforce business rules, and customize the behavior of your Salesforce org.

When to Use Apex Triggers?

Apex Triggers are ideal for scenarios where you want to:

- Automate Business Processes: Perform actions automatically when records are created, updated, or deleted.
- Enforce Data Integrity: Validate data before it's saved, ensuring that it meets specific criteria.
- Enhance Functionality:Extend Salesforce's capabilities by adding custom functionality or integrations.

- Trigger Workflow Actions:Initiate workflows based on record changes.

Creating an Apex Trigger:

1. Create a New Apex Class:
   - Navigate to Setup > Develop > Apex Classes.
   - Click "New" to create a new Apex class.
   - Define the trigger logic within the class, including the trigger context and the code to be executed.
2. Define the Trigger Context:
   - The trigger context provides information about the event that triggered the code, including the type of event (before insert, after insert, etc.), the list of affected records, and other relevant details.

3. Write the Trigger Logic:
   - The trigger logic is the code that will be executed when the trigger is fired. You can perform various actions within the trigger, such as:
      - Updating record fields
      - Creating or deleting records
      - Calling other Apex classes or web services
      - Sending email notifications
      - Updating related records
4. Save and Deploy the Trigger:
   - Save the Apex class and ensure that it compiles successfully.
   - Deploy the trigger to your Salesforce organization.

Example Apex Trigger

```
trigger MyTrigger on Account (after insert) {
    for (Account acc : Trigger.new) {
        acc.Owner = UserInfo.getUserId();
        update acc;
    }
}
```

Report:This probably refers to a pre-existing report that has already been created and can be viewed or analyzed.

Create Reports: This option likely allows users to create new reports from scratch, customizing the data, filters, and visualizations as needed.

Dashboard:This might refer to an existing dashboard that has been created and is ready for viewing.

Create Dashboards: This option probably enables users to build new dashboards, combining

multiple reports and visualizations to provide a comprehensive overview of data.