**Name:** Ch.K.Abhiram

**Institution:** Gayatri Vidya Parishad College of Engineering

**EMail Id:** 21131A4407@gvpce.ac.in

**Project Title:** **“Field Service Work Order Optimization”**

**1. Project Overview**

The Field Service Work Order Optimization System is designed to streamline the scheduling and assignment of work orders for a company specializing in installations and repairs. By utilizing Salesforce as a comprehensive platform, this project addresses the primary challenges of matching work orders with skilled technicians based on real-time factors such as technician location, availability, and skill set. The goal is to enhance operational efficiency, reduce operational costs, and increase customer satisfaction by providing a responsive and data-driven approach to field service management.

**2. Objectives**

**Business Goals:**

* Enhance the efficiency and reliability of field service operations.
* Improve resource allocation, minimizing travel time and maximizing productivity.
* Increase customer satisfaction by providing timely and skilled responses to service requests.

**Specific Outcomes:**

* Automated work order assignment based on technician availability and skill level.
* Real-time updates and notifications to keep technicians informed.
* Insightful analytics on technician performance and work order management for continuous improvement.

**3. Salesforce Key Features and Concepts Utilized**

**Salesforce Developer Edition:**

A Developer Edition org was established to develop, prototype, and refine customizations in a controlled environment, providing a platform to create tailored solutions suited to the company’s unique needs.

**Custom Objects:**

**Technician:** Object representing the field technicians, with fields for availability, skills, and location.

**WorkOrder:** Object for each service request, including fields for priority, customer location, and assigned technician.

**Assignment:** Object that links work orders with technicians, allowing efficient tracking and management of assignments.

**Custom Tabs and Lightning App:**

* Custom tabs for easy access to Technician, WorkOrder, and Assignment records.
* Lightning App designed for streamlined navigation and improved user experience.

**Fields and Relationships:**

* Lookup fields in the Assignment object for connecting work orders with technicians.
* Picklist values for fields like Technician Skills and Work Order Priority, allowing easy categorization and filtering.
* Formula fields in WorkOrder to automatically calculate relevant data, such as estimated time to completion.

**Profiles and Permissions:**

Technician profile with permissions specifically tailored to the technician role, ensuring appropriate access to relevant data.

**Apex Classes and Triggers:**

* Custom Apex Class for automating work order assignment based on defined criteria (e.g., technician availability and proximity).
* Apex Trigger for updating technician availability and sending notifications upon assignment.
* Asynchronous Apex Class and Scheduled Apex for regular updates to assignment availability and handling complex data updates.

**Reports and Dashboards:**

* Custom reports to monitor key metrics like technician performance and assignment efficiency.
* Dashboards for quick insights into field service operations, providing visual representations of data for decision-makers.

1. **Detailed Steps to Solution Design**

* **Created\_models:**

**Technician Data Preparation**

* 1. Edited email records in the Technician spreadsheet and saved it as Technician.csv for import.

**Salesforce Setup Access**

* 1. Logged into Salesforce, accessed Setup via the Gear icon, and navigated to **Object Manager**.

**Technician Object Creation from Spreadsheet**

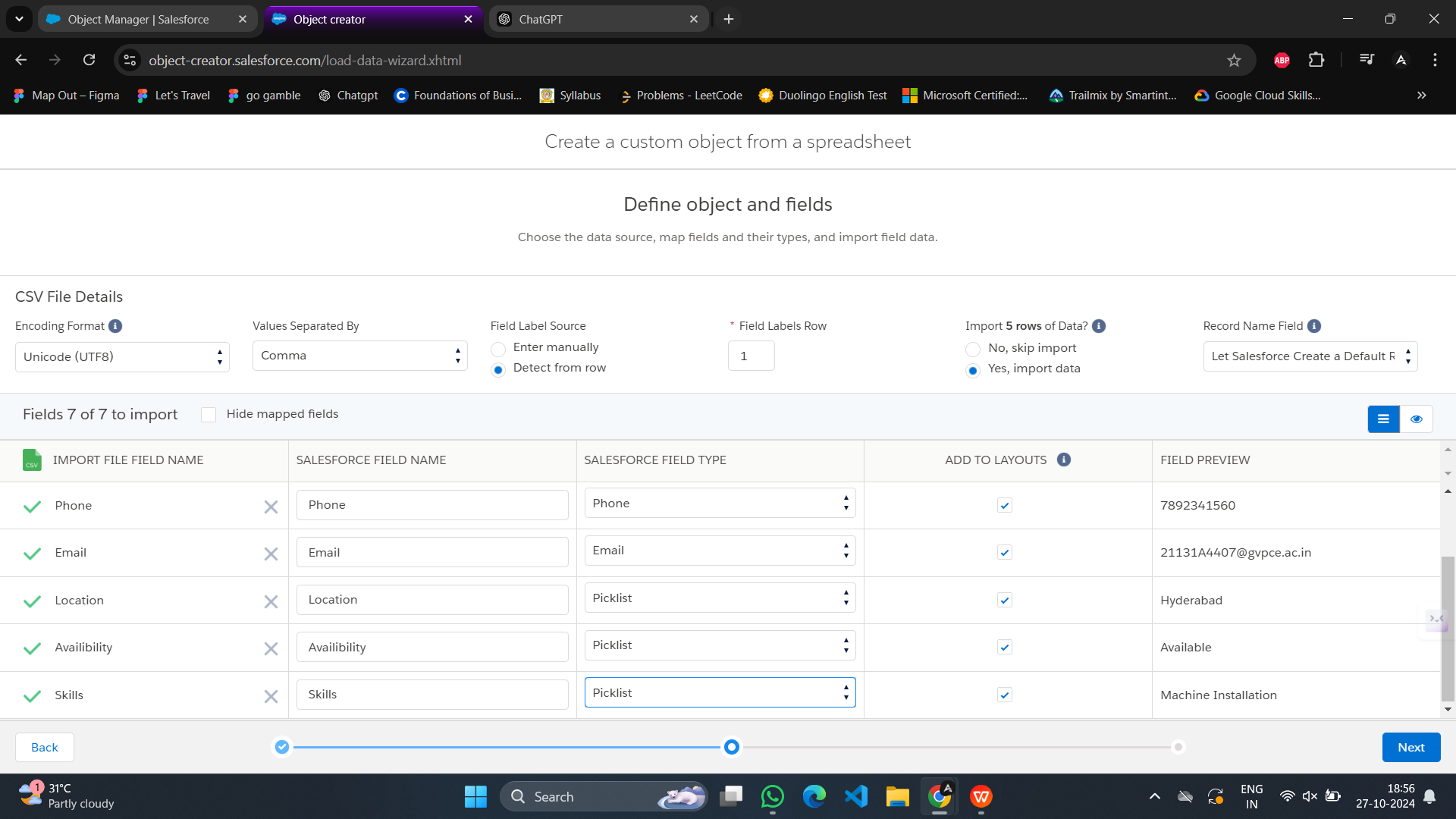
* 1. Uploaded Technician.csv, verified field mappings, set **Technician ID** as the Record Name, and finalized the Technician object setup with auto-imported data.

**WorkOrder Object Creation**

* 1. Created the WorkOrder object using similar steps to the Technician object for consistency and data accuracy.

**Assignment Object Creation**

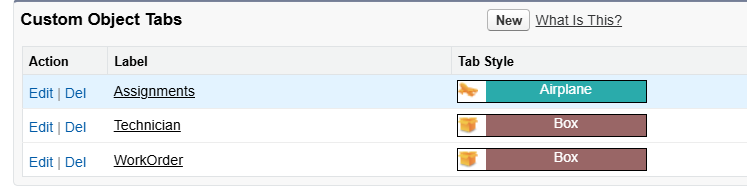
* 1. Manually created the Assignment object, configured fields, enabled auto-numbering for Assignment ID, and allowed reporting and search functionalities.



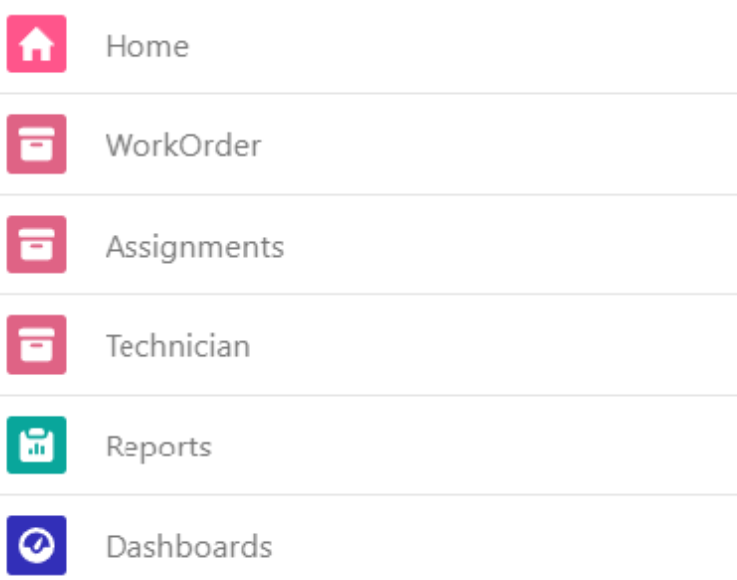
* **User Interface Design:**

**Tab Creation for Assignment Object**

* 1. **Access Tabs Setup:**  
     Navigated to **Setup > Tabs** using the Quick Find bar, clicked **New** under the Custom Object Tabs section.
  2. **Configure Assignment Tab:**  
     Selected **Assignment** as the Object, chose a tab style, then proceeded with default settings for profile and app visibility.
  3. **Save Settings:**  
     Saved the Assignment tab setup, confirming successful creation. (Note: Tabs for WorkOrder and Technician objects were created automatically, no additional setup needed.)
* **User Interface Setup**
  1. **Assignment Tab Interface:**  
     Configured the Assignment object tab interface to allow efficient navigation and access within Salesforce.



* **Creating a ligthning app**· **Creating a Lightning App for Field Service WorkOrder Optimization**
* **App Manager Setup:**  
  Accessed **App Manager** in Setup via the Quick Find bar, then selected **New** to create the app.
* **App Details and Branding:**  
  Entered the following:
  + **App Name:** Field Service WorkOrder Optimization
  + **Developer Name:** Auto-populated
  + **Description:** Provided a meaningful description of the app’s purpose
  + **Primary Color:** Kept as default; image optional
* **Adding Navigation Items:**  
  Added navigation items (Home, WorkOrder, Technician, Assignment, Reports, Dashboard) to the app by searching and moving them into the selected items list. Ensured **Assignment** was selected as the custom object from prior activities.
* · **Adding User Profiles**
* **Profile Selection:**  
  Searched for **System Administrator** in the profiles section, selected it, and proceeded with default settings for App Options and Utility Items.
* **Finalization:**  
  Saved and completed the setup for the Field Service WorkOrder Optimization app.



* **Setting Fields and relationships**

· **Lookup Field in Assignment Object**

* · **Setup Process:** Accessed **Object Manager** for the Assignment object, created a lookup field labeled **WorkOrder ID**, linked to the WorkOrder object, and saved.

· **Managing Picklist Values in WorkOrder Object**

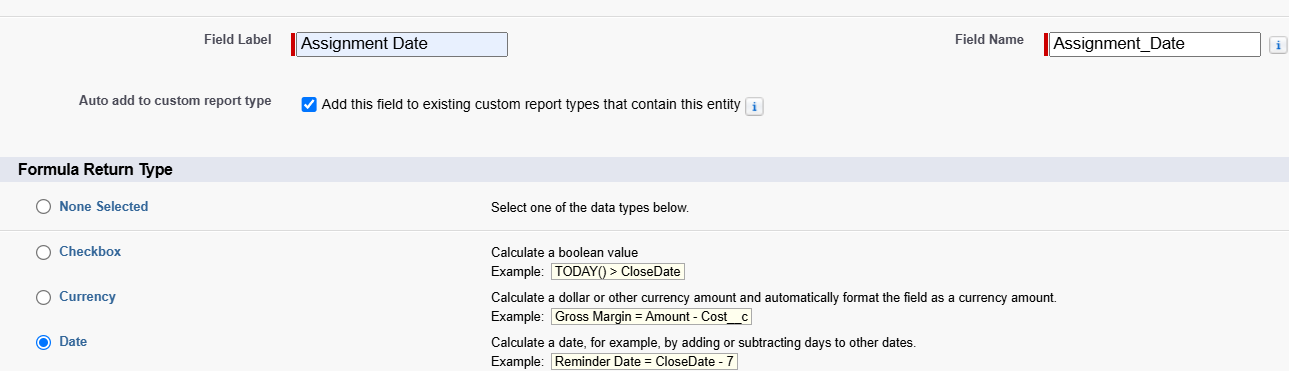
* · **Location Field:** Added new location values: Nasik, Warangal, and Nanded.
* **Priority Field:** Added **High** as a value.
* **Service Type Field:** Added values **Hardware Repair, Troubleshoot/Debugging,** and **Lane-Management**.

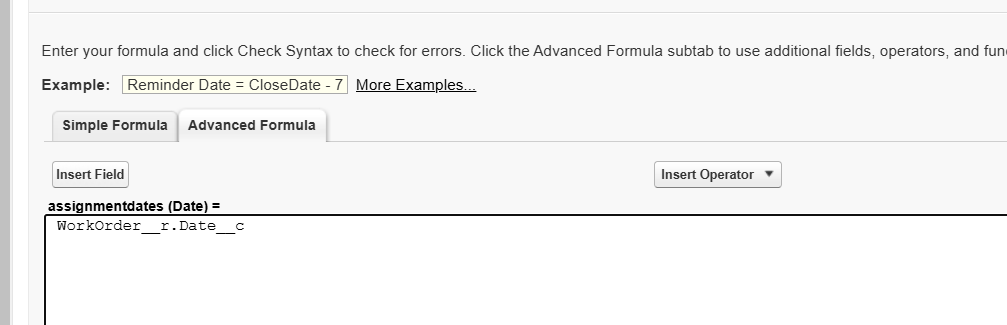
· **Formula Field in WorkOrder Object**

* · Created a **Date** formula field with return type **Date**. The formula retrieves the date field value directly.

· **Additional Fields in Assignment Object**

* · **Technician ID:** Created as a lookup to the Technician object.
* **Assignment Date:** Formula field with return type **Date**, pulling WorkOrder\_ID\_\_r.Date\_\_c.
* **Completion Date:** Formula field with return type **Date**, conditional on WorkOrder\_ID\_\_r.Status\_\_c being **Resolved**, in which case it retrieves LastModifiedDate; otherwise, returns **NULL**.





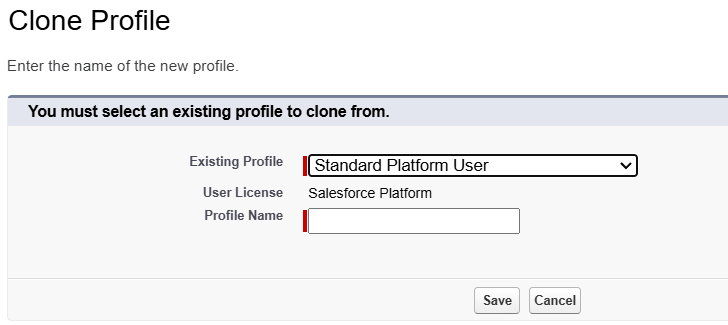
* **Creating profile**

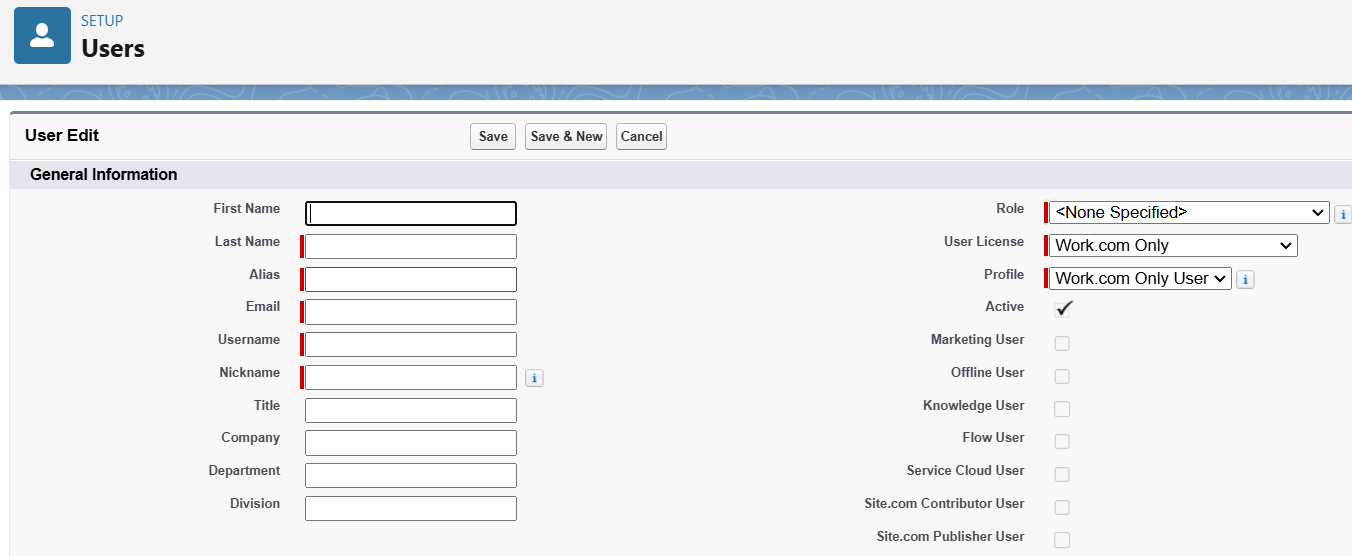
· **Create Technician Profile**

* · Navigated to **Setup > Profiles**, selected **New Profile** based on **Standard Platform User**, and named it **Technician**.

· **Configure Permissions**

* · Edited **Custom Object Permissions** to give **Read-only access** for Technician, WorkOrder, and Assignment objects.
* Updated **Field-Level Security** for the WorkOrder object, enabling access to the **Status** field.





* **New User Setup**
* In **Setup > Users**, selected **New User** and filled in:
  + **First Name:** Elina
  + **Last Name:** Gilbert
  + **Alias:** Custom alias
  + **Email:** Personal email
  + **Username:** In the format text@text.text
  + **Nickname:** Custom nickname
  + **User License:** Salesforce Platform
  + **Profile:** Technician
* **Creating classes and triggers**

**WorkOrderTrigger.apxt**

trigger WorkOrderTrigger on WorkOrder\_\_c (after insert) {

if(trigger.isafter && trigger.isinsert){

WorkOrderClass.workOrder(trigger.new);

}

}

**ScheduleClass.apxt**

global class ScheduleClass implements Schedulable {

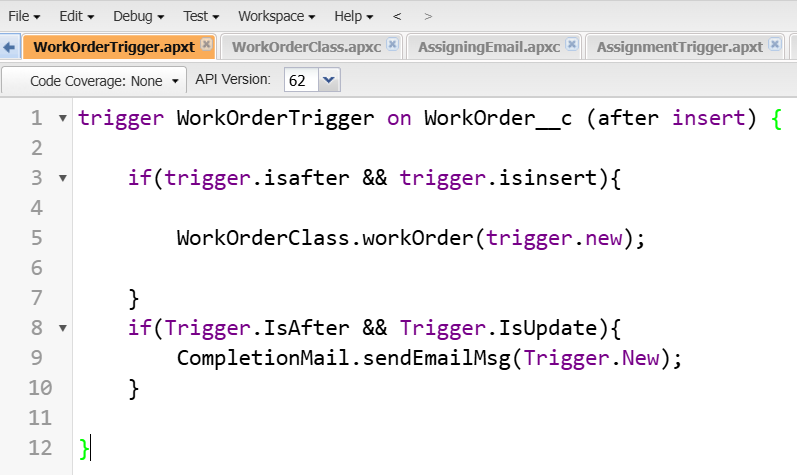
global void execute(SchedulableContext SC) {

RecordDeletions delrec = new RecordDeletions();

database.executeBatch(delrec, 200);

}

}



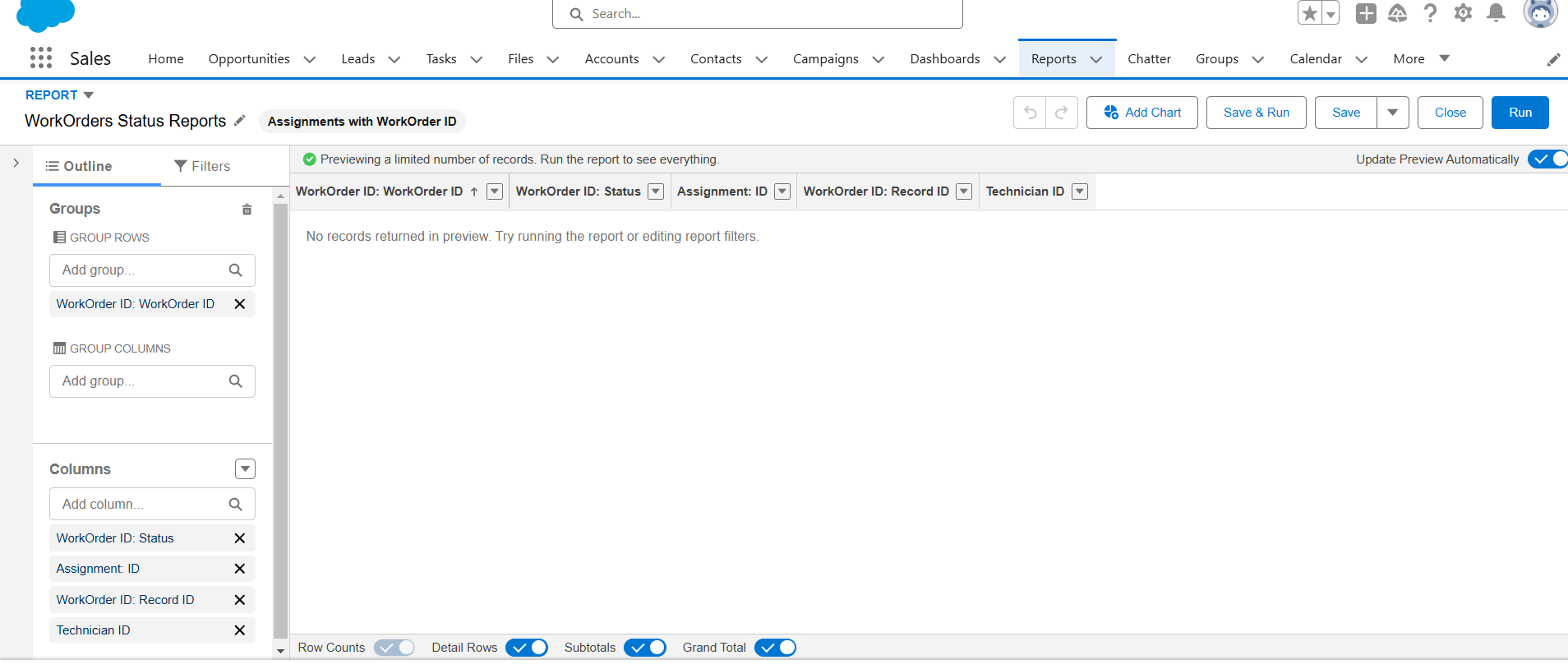
* **Creating Reports and Dashboards**

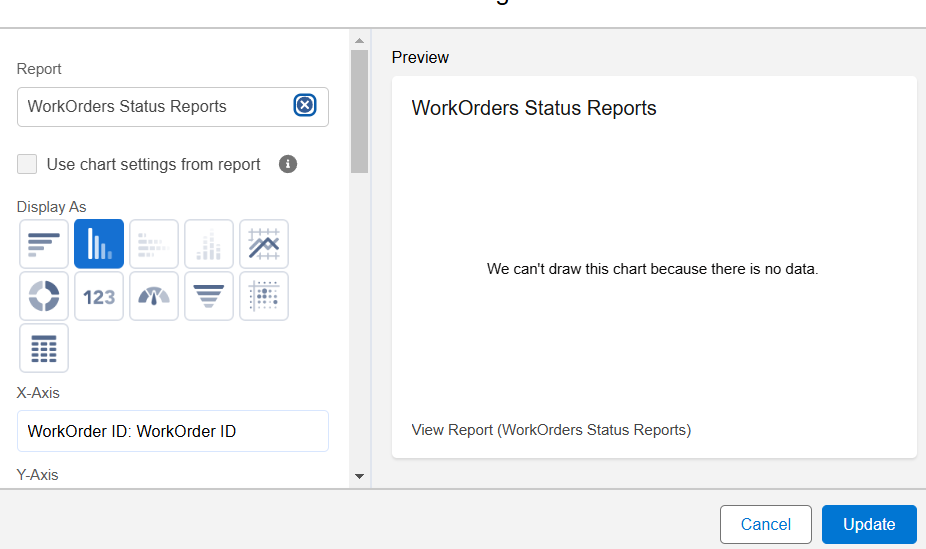
· **Creating a Report**

* · Accessed the **Reports** tab, selected **New Report**, and chose a report type.
* Customized by adding fields from the left pane, grouped by **WorkOrder ID**.
* Saved and ran the report for data analysis.

· **Creating a Dashboard**

* · Navigated to the **Dashboards** tab, created a new dashboard.
* Configured a **vertical bar graph** displaying **completed WorkOrder statuses**.





**5. Testing and Validation**

**Unit Testing:**

Developed unit tests for all Apex classes and triggers, covering core functionalities like work order assignment, technician availability updates, and notifications to ensure reliability.

**User Interface Testing:**

Conducted UI tests to confirm that custom tabs, Lightning App, and field configurations met user requirements and provided an intuitive experience.

**6. Key Scenarios Addressed by Salesforce in the Implementation Project**

**Automated Assignment Matching:**

Using Salesforce’s database and Apex triggers, work orders are automatically assigned to the most suitable technician based on availability, skills, and proximity.

**Real-Time Communication:**

The system sends automated notifications to technicians, keeping them updated on assignments and helping them manage their schedules effectively.

**Performance Analytics:**

Reports and dashboards provide insights into technician performance and overall efficiency of the work order assignments, facilitating data-driven decisions for future improvements.

**7. Conclusion**

**Summary of Achievements:**

The Field Service Work Order Optimization System has successfully automated the process of assigning and managing work orders, resulting in improved efficiency, reduced operational costs, and enhanced customer satisfaction. Salesforce's robust platform, along with custom objects, automation through Apex, and data insights via reports and dashboards, has been pivotal in delivering a solution that is both scalable and efficient for field service operations.