

Retail Management Application

(An optional Guided Project for partial fulfillment of Salesforce Supported Virtual Internship Program 2023)



Organised By: Smart Bridge
Powered By: Smart Internz



Submitted By : Ayush Bhagat (B.Tech CSE 4th Year)
Birla Institute Of Applied Sciences, Bhimtal

Uttarakhand

Introduction

Salesforce is a cloud-based customer relationship management (CRM) platform that helps businesses of all sizes manage their customer interactions. Salesforce Retail Management Application is a suite of cloud-based applications that help retailers manage their sales, inventory, and customer relationships.

The Salesforce Retail Management Application includes the following modules:

- **Sales Cloud:** This module helps retailers manage their sales pipeline, track leads and opportunities, and close deals.
- **Service Cloud:** This module helps retailers provide excellent customer service by providing a single view of the customer across all channels.
- **Commerce Cloud:** This module helps retailers sell their products online and in-store.
- **Marketing Cloud:** This module helps retailers create and execute marketing campaigns that reach their target audience.
- **Analytics Cloud:** This module helps retailers track their performance and make data-driven decisions.

The Salesforce Retail Management Application can be used to:

- **Automate sales processes:** The Salesforce Retail Management Application can automate sales processes, such as lead generation, qualification, and tracking. This can free up sales representatives to focus on building relationships with customers.
- **Improve customer relationships:** The Salesforce Retail Management Application can help retailers improve customer relationships by providing a single view of the customer across all channels. This can help retailers provide more Personalized customer service and marketing messages.

- **Increase sales:** The Salesforce Retail Management Application can help retailers increase sales by providing insights into customer behavior and trends. This information can be used to target marketing campaigns and optimize product offerings.
- **Reduce costs:** The Salesforce Retail Management Application can help retailers reduce costs by automating tasks and providing insights into operations. This can help retailers save time and money.

The Salesforce Retail Management Application is a powerful tool that can help retailers improve their sales, customer relationships, and bottom line.

Here are some of the advantages of using Salesforce for retail management:

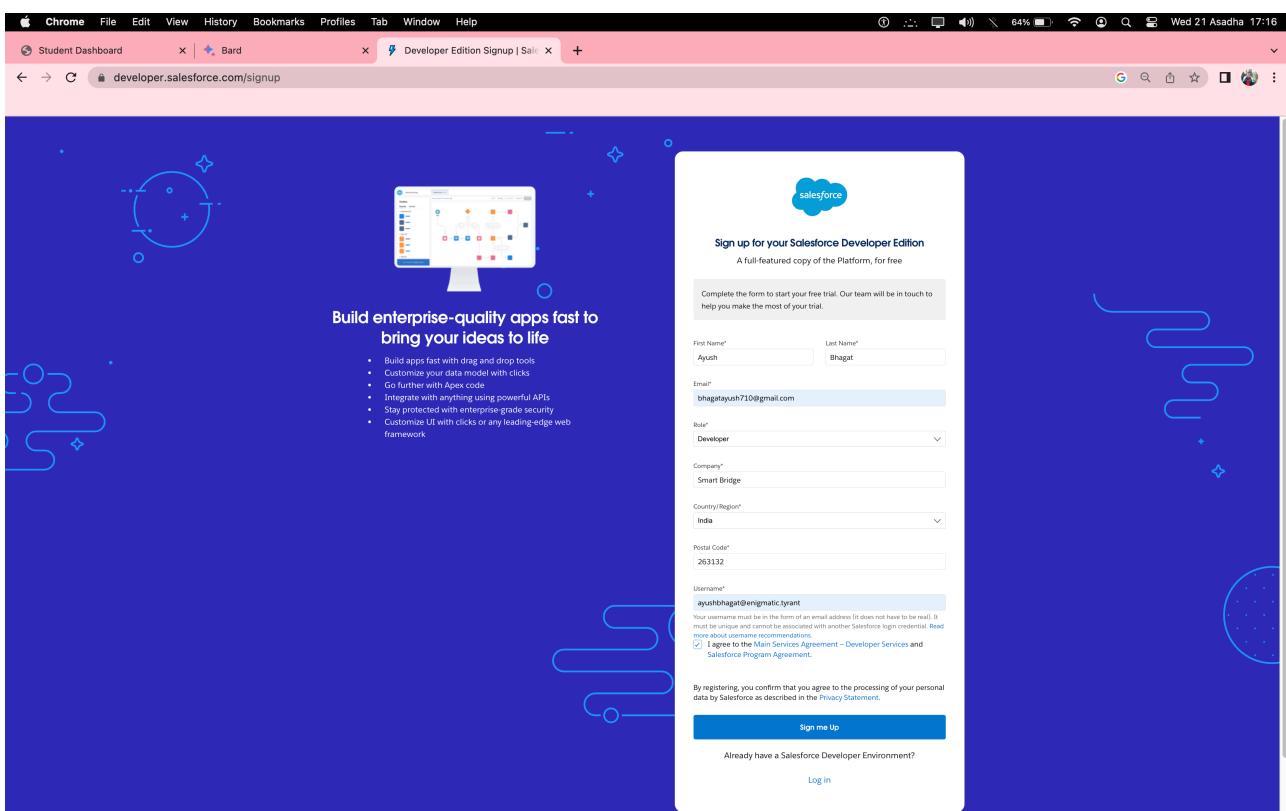
- **Increased visibility and control:** Salesforce provides a single view of all customer data, which allows retailers to make better decisions about their business.
- **Improved efficiency:** Salesforce can automate many of the manual tasks involved in retail management, such as order processing and inventory management. This frees up employees to focus on more strategic tasks.
- **Enhanced customer service:** Salesforce can help retailers provide better customer service by providing a more personalized experience. For example, retailers can use Salesforce to track customer preferences and send them relevant product recommendations.
- **Increased agility:** Salesforce is a cloud-based platform, which means that retailers can access it from anywhere. This gives retailers the flexibility to adapt to changing market conditions.

If you are a retailer looking for a way to improve your business, Salesforce is a great option. Salesforce can help you increase sales, improve customer relationships, and reduce costs.

Milestone 1: Creating Developer Org

Creating a developer org in salesforce.

1. Go to developers.salesforce.com/Signup
2. Click on sign up.
3. On the sign up form, enter the following details :
 - (i) First name & Last name
 - (ii) Email
 - (iii) Role : Developer
 - (iv) Company : College Name
 - (v) County : India
 - (vi) Postal Code : pin code
 - (vii) Username : should be a combination of your name and company This need not be an actual email id, you can give anything in the format: username@organization.com



Milestone 2: Creation Of Objects

- Click on the gear icon and then select Setup.
- Click on the object manager tab just beside the home tab.
- After the above steps, have a look on the extreme right you will find a Create Dropdown click on that and select Custom Object.
- On the Custom Object Definition page, create the object as follows:
 - Label: Warehouse
 - Plural Label: Warehouses
 - Record Name: Warehouse Name
 - Check the Allow Reports checkbox
 - Check the Allow Search checkbox
 - Click Save.
- In the same way create **2** more objects as Sales order, Dispatch/ Tracking.

The screenshot shows a browser window with the URL enigmatictyrant2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01l5j000000K0dy/Details/view. The page is titled "Object Manager" under "SETUP > OBJECT MANAGER". A sidebar on the left lists various setup categories. The main content area shows the "Warehouse" object details. The API Name is set to "Warehouse__c". The "Description" field is empty. Under "Fields & Relationships", the "Plural Label" is listed as "Warehouses". On the right, checkboxes for "Enable Reports" (checked), "Track Activities", and "Track Field History" are visible. Deployment status is shown as "Deployed" with a link to "Standard salesforce.com Help Window". At the bottom right are "Edit" and "Delete" buttons.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Student Dashboard Dispatch/Tracking | Salesforce

enigmatictyrant2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01l5000000KOe8/Details/view

Setup Home Object Manager

SETUP > OBJECT MANAGER Dispatch/Tracking

Details

Description

API Name: Dispatch_Tracking__c

Custom: ✓

Singular Label: Dispatch/Tracking

Plural Label: Dispatch/Tracking

Enable Reports: ✓

Track Activities: ✓

Track Field History

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Details

Edit Delete

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Triggers

Flow Triggers

Validation Rules

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Student Dashboard Sales order | Salesforce

enigmatictyrant2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01l5000000KOe8/Details/view Press [fn] F to exit full screen

Setup Home Object Manager

SETUP > OBJECT MANAGER Sales order

Details

Description

API Name: Sales_order__c

Custom: ✓

Singular Label: Sales order

Plural Label: Sales orders

Enable Reports: ✓

Track Activities: ✓

Track Field History

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Details

Edit Delete

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Triggers

Flow Triggers

Validation Rules

Milestone 3: Creating a tab

Tabs in Salesforce help users view the information at a glance. It displays the data of objects and other web content in the application.

- **Standard Object Tabs:** Standard object tabs display data related to standard objects
- **Custom Object Tabs:** Custom object tabs displays data related to custom objects.
- **Web Tabs:** Web Tabs display any external Web-based application or Web page in a Salesforce tabs.
- **Visualforce Tabs:** Visualforce Tabs display data from a Visualforce Page.

1. Enter Tabs in Quick Find and select Tabs.
2. Under Custom Object Tabs, click New.
3. For Object, select Warehouse.
4. For Tab Style, select any icon.
5. Leave all defaults as is. Click Next, Next, and Save
6. In the same way create Tabs for all Custom Objects – Sales order, Dispatch /Tracking.

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Chrome, File, Edit, View, History, Bookmarks, Profiles, Tab, Window, Help. The URL is enigmatictyrant2-dev-ed.lightning.force.com/lightning/setup/CustomTabs/home.
- Left Navigation:** Setup, Home, Object Manager. Under "User Interface", "Tabs" is selected.
- Page Title:** SETUP Tabs
- Section:** Custom Tabs
- Description:** You can create new custom tabs to extend Salesforce functionality or to build new application functionality.
- Table:** Custom Object Tabs

Action	Label	Tab Style	Description
Edit Del	Dispatch/Tracking	Chalkboard	
Edit Del	Sales orders	Sailboat	
Edit Del	Warehouses	Globe	

- Section:** Web Tabs

No Web Tabs have been defined.

- Section:** Visualforce Tabs

No Visualforce Tabs have been defined.

- Section:** Lightning Component Tabs

No Lightning component tabs have been defined.

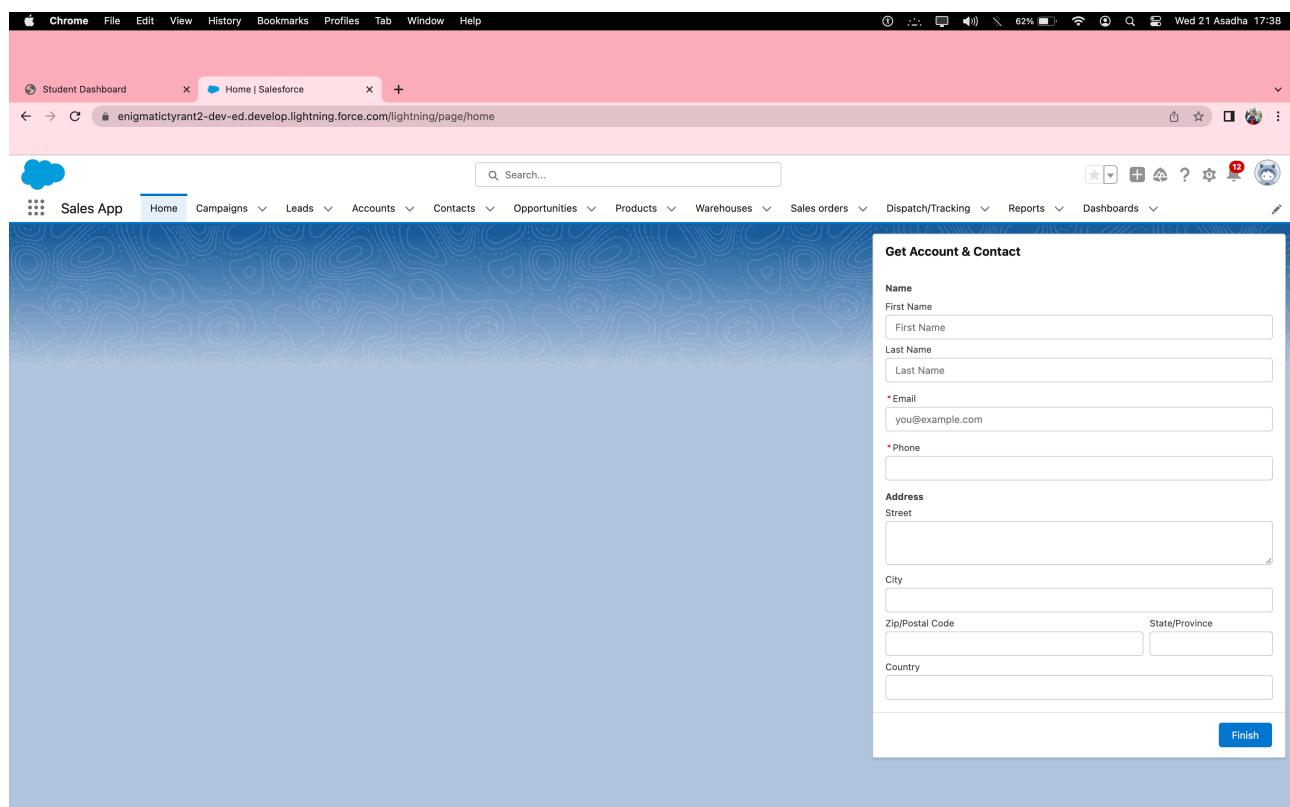
- Section:** Lightning Page Tabs

No Lightning Page Tabs have been defined.

Milestone 4: Create The Sales App

Custom Apps: Custom apps are created according to need of user. Custom Apps are made by using standard and custom tabs together.

1. From Setup, enter App Manager in the Quick Find and select App Manager.
2. Click New Lightning App.
3. Enter Sales App as the App Name, then click Next
4. Under App Options, leave the default selections and click Next.
5. Under Utility Items, leave as is and click Next.
6. From Available Items, select Campaign, Leads, Accounts, Contacts, Opportunities, Products, Warehouse, Sales order, Dispatch/Tracking, Reports, and Dashboards and move them to Selected Items. Click Next.
7. From Available Profiles, select System Administrator and move it to Selected Profiles. Click Save & Finish.



Milestone 5: Create Fields And Relationship

Fields - Fields store data values that are required for a particular object in a record.

An object relationship in Salesforce is **a two-way association between two objects**. Relationships are created by creating custom relationship fields on an object. This is done so that when users view records, they can also see and access related data.

These are fields and their data types we need to create make them one by one –

Object	Fields	Datatype
Contact	Account website	Lo
Warehouse	Address	Text Area
	Location	Text (50)
Sales Order	Status	Picklist
	Customer	Lookup (Account)
	Contact	Lookup (contact)
	Order date	Date
Dispatch/Tracking	Dispatched	Checkbox
	Tracking ID	Text
	Sales Order	Master Detail (Sales Order)
	Ex	Date

- **Creation Of Fields For The Dispatch/Tracking Object:**

1. Click the gear icon and select Setup. This launches Setup in a new tab.
2. Click the Object Manager tab next to Home.
3. Select Dispatch/Tracking
4. Select Fields & Relationships from the left navigation

5. Click New
6. Select the Text as the Data Type, click Next.
7. For Field Label, enter Tracking ID & length = 40.
8. Click Next, Next, then Save & New.

Similarly we create fields for **Warehouse** object- Address, Location select datatype according table and create fields for **Dispatch/Delivery** object- Dispatched, Expected date of delivery select datatype according table.

- **Create A Master-Detail Relationship On Dispatch/Delivery Object:**

- Select Master-Detail Relationship as the Data Type and click Next.
- For Related to, enter Sales order.
- Click Next.
- For Field Label, enter Sales order.
- Click Next, Next, Next and Save.

- **Create A Pick-List Field On Sales Order:**

- From Setup, click Object Manager and select Sales order.
- Click Fields & Relationships, then New.
- Select Picklist as the Data Type and click Next.
- For Field Label enter Status
- Select Enter values, with each value separated by a new line and enter these values:
 - Open
 - Hold
 - Shipped
 - Returned
- Click Next, Next, then Save & New.

- **Create A Lookup Relationship On Sales Order Object With Account Object:**

- Select look up Relationship as the Data Type and click Next.
- For Related to, enter Account.
- Click Next.
- For Field Label, enter Customer.
- Click Next, Next, Next and Save.
- Similarly create one more lookup field- Field Name - Contact Related to - Contact

- **Create A Lookup Relationship On Sales Order Object With Contact Object With Use Of Lookup Filter:**

Lookup Filter- Lookup filters limit the records available in the lookup. A lookup filter can reference other fields on the same record (source); fields on the records of the lookup object (target); fields on the user's record, profile, and role; and fields on records directly related to the target object.

Follow steps 1 to 5 of field creation from activity 1 then follow below steps.

1. Select look up Relationship as the Data Type and click Next.
2. For Related to, enter **Contact**.
3. Click Next.
4. For Field Label, enter **Contact**.
5. Click **lookup filter**.
6. Provide filter as given
Contact: Account ID equals Sales Order: Customer
7. Click Next, Next, Next and Save.

o Cross-Object Formula Field:

A cross-object formula field is basically a formula field. A cross-object formula can reference merge fields from a master object if an object is on the detail side of a master-detail relationship.

A cross-object formula works with Lookup relationships as well as in Master detail relationship. You can reference fields from objects that are up to 10 relationships away.

1. Select your object from object selection has Contact.
2. And select the option fields and relationships.
3. At the top right side you can find a new select that option.
4. Now you have to select data type as **formula**.
5. And you will navigate to enter the details page where you give the field label.
6. And give the label name has Account Website
7. Select formula return type **Text**
8. In the formula field enter this formula Account. Website.
9. Click next you will navigate to field level security click on visible checkbox so that it is visible to all profiles.

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, etc. The main content area is titled 'Fields & Relationships' under the 'Dispatch/Tracking' object. It displays a table with 7 items, sorted by Field Label. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table rows are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Dispatch/Tracking Name	Name	Text(80)		
Dispatched	Dispatched__c	Checkbox		
Expected date of delivery	Expected_date_of_delivery__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Sales order	Sales_order__c	Master-Detail(Sales order)		
Tracking ID	Tracking_ID__c	Text(40)		

Student Dashboard Sales order | Salesforce

enigmaticallyrant2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01l5j000000K0e3/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER

Sales order

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Triggers

Flow Triggers

Validation Rules

Fields & Relationships
10 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contact	Contact__c	Lookup(Contact)		✓
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(Account)		✓
Email id	Email__id__c	Email		✓
Last Modified By	LastModifiedById	Lookup(User)		
Order confirmed	Order_confirmed__c	Picklist		
Order date	Order_date__c	Date		✓
Owner	OwnerId	Lookup(User,Group)		✓
Sales order Number	Name	Auto Number		✓
Status	Status__c	Picklist		

10. Select the next option, select the page layout and save it.

Student Dashboard Warehouse | Salesforce

enigmaticallyrant2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01l5j000000K0dy/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER

Warehouse

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Triggers

Flow Triggers

Validation Rules

Fields & Relationships
6 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text Area(255)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Location	Location__c	Text(50)		
Owner	OwnerId	Lookup(User,Group)		✓
Warehouse Name	Name	Text(80)		✓

○ Milestone 6 Creation Of Validation Rule:

1. Navigate to object manager and select Account object.
2. In details section scroll down and find validation rule in it.
3. Click new, give the label name and in edit error conditional formula give the formula- LEFT(Phone, 1) <> "+" .
4. And in error message give the description has Phone number must begin with + (country code).
5. In error location select field.
6. Save

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc., with 'Validation Rules' selected. The main content area is titled 'Account Validation Rule' and shows a single record with the following details:

Validation Rule Detail	
Rule Name	phone_number_has_international_format
Error Condition Formula	LEFT(Phone, 1) <> "+"
Description	Phone number must begin with + (country code).
Created By	Ayush Bhagat, 12/07/2023, 6:08 am
Modified By	Ayush Bhagat, 12/07/2023, 6:08 am

The 'Active' checkbox is checked. The 'Error Location' is set to 'Phone'. The top navigation bar shows tabs for Student Dashboard, Account | Salesforce, Post Attendee - Zoom, and a new tab. The address bar indicates the URL is enigmatictyrant2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/Account/ValidationRules/03d5j000000Hk0VAAS/view.

Create Record (Sales Order):

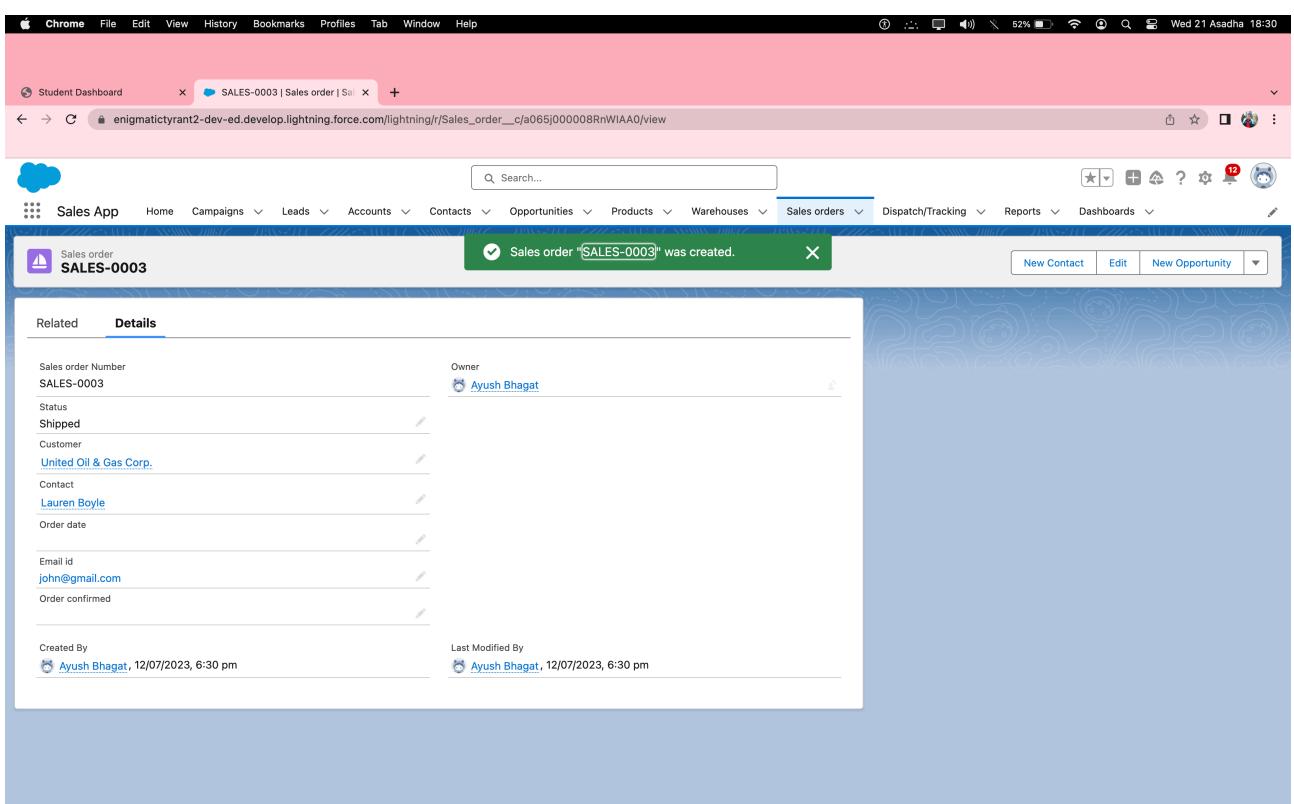
1. Click on App Launcher on left side of screen.
2. Search Sales App & click on it.
3. Click on Sales Order tab.
4. Click new button
5. Fill all Sales Order record details.
6. Click on Save Button

View Record (Sales Order):

1. Click on App Launcher on left side of screen.
2. Search Sales App & click on it.
3. Click on Sales Order Tab.
4. Click on any record name. you can see the details of the Sales Order

Delete Record (Sales Order):

1. Click on App Launcher on left side of screen.
2. Search Sales App & click on it.
3. Click on Sales Order Tab.
4. Click on Arrow at right hand side on that Particular record.
5. Click delete and delete again.



○ **Milestone 8 Reports:**

Reports in Salesforce is a list of records that meet a particular criterion which gives an answer to a particular question. These records are displayed as a table that can be filtered or grouped based on any field.

There are 4 types of report formats in Salesforce:

1. **Tabular Reports:**

This is the most basic report format. It just displays the row of records in a table with a grand total. While easy to set up they can't be used to create groups of data or charts and also cannot be used in Dashboards. They are mainly used to generate a simple list or a list with a grand total.

2. **Summary Reports:**

It is the most commonly used type of report. It allows grouping of rows of data, view subtotal, and create charts.

3. **Matrix Report:**

It is the most complex report format. Matrix report summarizes information in a grid format. It allows records to be grouped by both columns and rows. It can also be used to generate dashboards. Charts can be added to this type of report.

4. **Joined Reports:**

These types of reports let us create different views of data from multiple report types. The data is joined reports are organized in blocks. Each block acts as a subreport with its own fields, columns, sorting, and filtering. They are used to group and show data from multiple report types in different views.

Report types:

Report type determines which set of records will be available in a report. Every report is based on a particular report type. The report type is selected first when we create a report. Every report type has a primary object and one or more related objects. All these objects must be linked together either directly or indirectly.

A report type cannot include more than 4 objects.

Once a report is created its report type cannot be changed.

There are 2 types of report types:

1. Standard Report Types:

Standard Report Types are automatically included with standard objects and also with custom objects where “Allow Reports” is checked.

Standard report types cannot be customized and automatically include standard and custom fields for each object within the report type. Standard report types get created when an object is created, also when a relationship is created.

2. Custom Report Types:

Custom report types are reporting templates created to streamline the reporting process. Custom Reports are created by an administrator or User with “Manage Custom Report Types” permission. Custom report types are created when standard report types cannot specify which records will be available on reports.

In custom report types we can specify objects which will be available in a particular report. The primary object must have a relationship with other objects present in a report type either directly or indirectly.

1. Viewer:

With this access level, users can see the data in a report but

cannot make any changes except cloning it into a new report.

2. Editor:

With this access level, users can view and modify the reports it contains and can also move them to/from any other folders they have access level as Editor or Manager.

3. Manager:

With this access level, users can do everything Viewers & Editors can do, plus they can also control other user's access levels to this folder. Also, users with Manager Access levels can delete the report.

View Report

1. Click on App Launcher on left side of screen.
2. Search “Sales App” & click on it.
3. Click on Reports Tab.
4. Click on New Sales orders with Customer Report and see records

The screenshot shows the Salesforce Lightning interface on a Mac OS X desktop. The browser window title is "Reports | Salesforce". The page URL is "enigmatictyrant2-dev-ed.lightning.force.com/lightning/o/Report/home?queryScope=mru". The top navigation bar includes "Home", "Campaigns", "Leads", "Accounts", "Contacts", "Opportunities", "Products", "Warehouses", "Sales orders", "Dispatch/Tracking", "Reports", and "Dashboards". On the left, the App Launcher sidebar lists "Reports", "Recent", "Created by Me", "Private Reports", "Public Reports", "All Reports", "Folders", "All Folders", "Created by Me", "Shared with Me", and "Favorites". The main content area displays a table of recent reports. The table columns are: Report Name, Description, Folder, Created By, Created On, and Subscribed. Two reports are listed:

Report Name	Description	Folder	Created By	Created On	Subscribed
Sample Flow Report: Screen Flows	Which flows run, what's the status of each interview, and how long do users take to complete the screens?	Public Reports	Automated Process	11/7/2023, 6:28 pm	
New Sales orders with Customer Report		Private Reports	Ayush Bhagat	12/7/2023, 6:16 am	

Milestone 9 Dashboards:

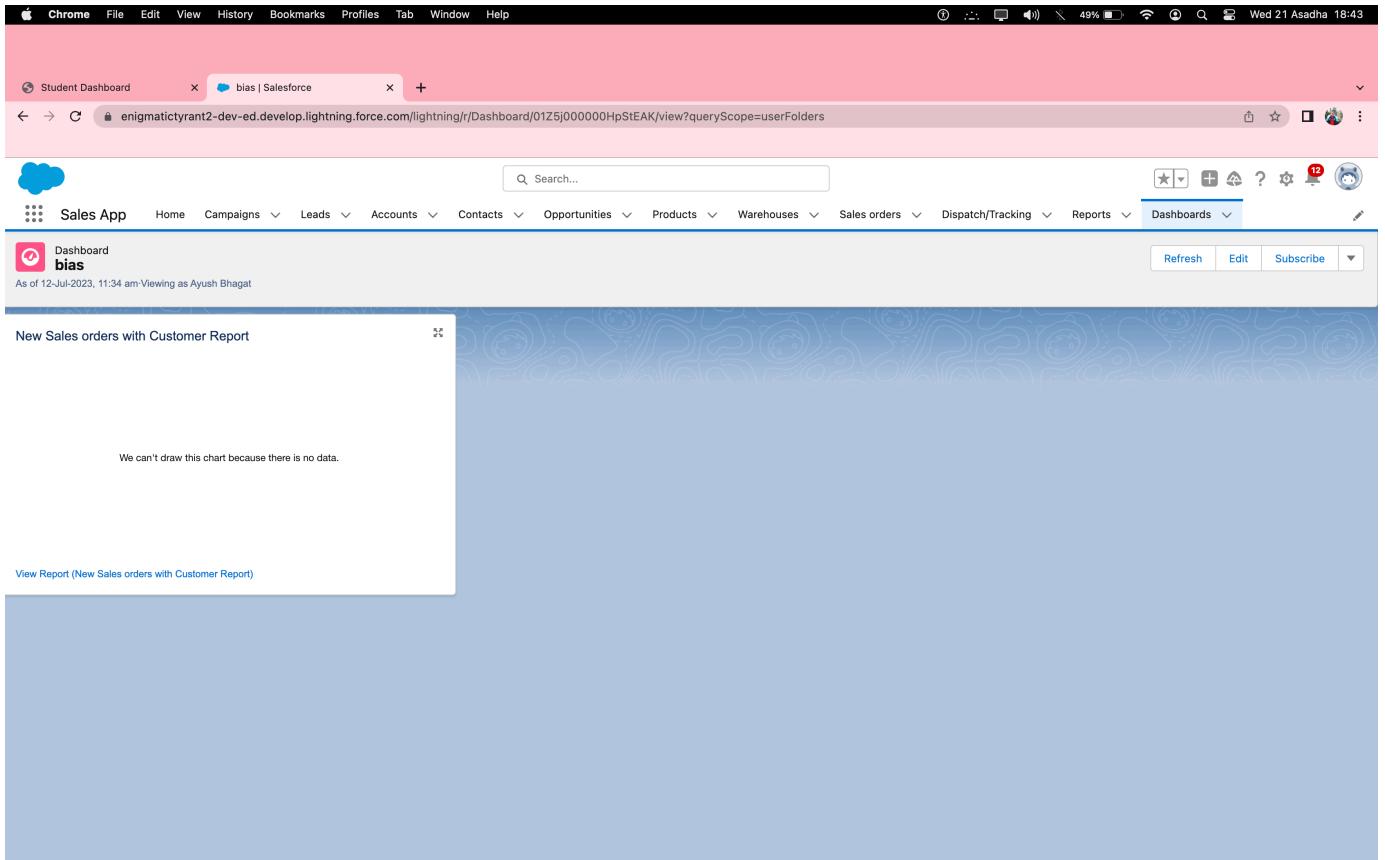
Dashboards let us curate data from reports using charts, tables, and metrics. If your colleagues need more information, then they're able to view your dashboard's data-supplying reports. Dashboard filters make it easy for users to apply different data perspectives to a single dashboard.

Create Dashboard:

1. Click on Dashboards tab from the “Sales App” application,
2. Click on new dashboard
3. Give name- Sales App Dashboard
4. Click create
5. Give your dashboard a name and click on +component.
6. Select the New Sales orders with Customer Report which you created.
7. For the data visualization select any of the chart, table etc as your wish.
8. Click add and save

View Dashboard

1. Click on App Launcher on left side of screen.
2. Search Sales App & click on it.
3. Click on Dashboard Tab.
4. Click on Sales App Dashboard and see graph view of records

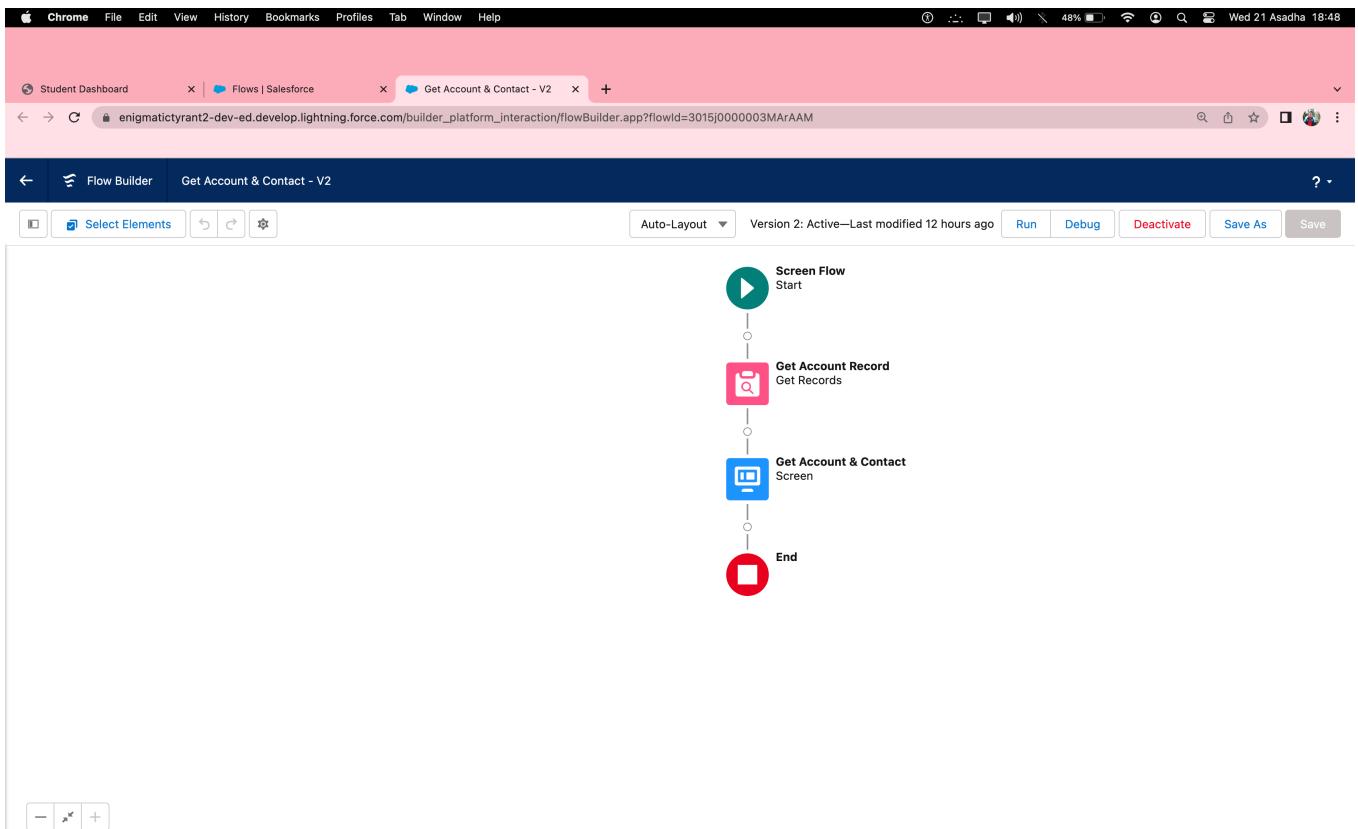


○ Milestone 10 Flows:

Flows are an automation tool provided by Salesforce which can be used to perform various tasks like, Sending an Email, Posting a chatter, Sending custom Notifications &, etc. Flow is the most powerful automation tool provided by Salesforce. It can be trigger for record insert, update and record delete and it can be run for both after and before events.

1. Click on Dashboards tab from the “Sales App” application,
2. Click on new dashboard
3. Give name- Sales App Dashboard
4. Click create
5. Give your dashboard a name and click on +component.
6. Select the New Sales orders with Customer Report which you created.
7. For the data visualization select any of the chart, table etc as your wish.

8. Click add and save.

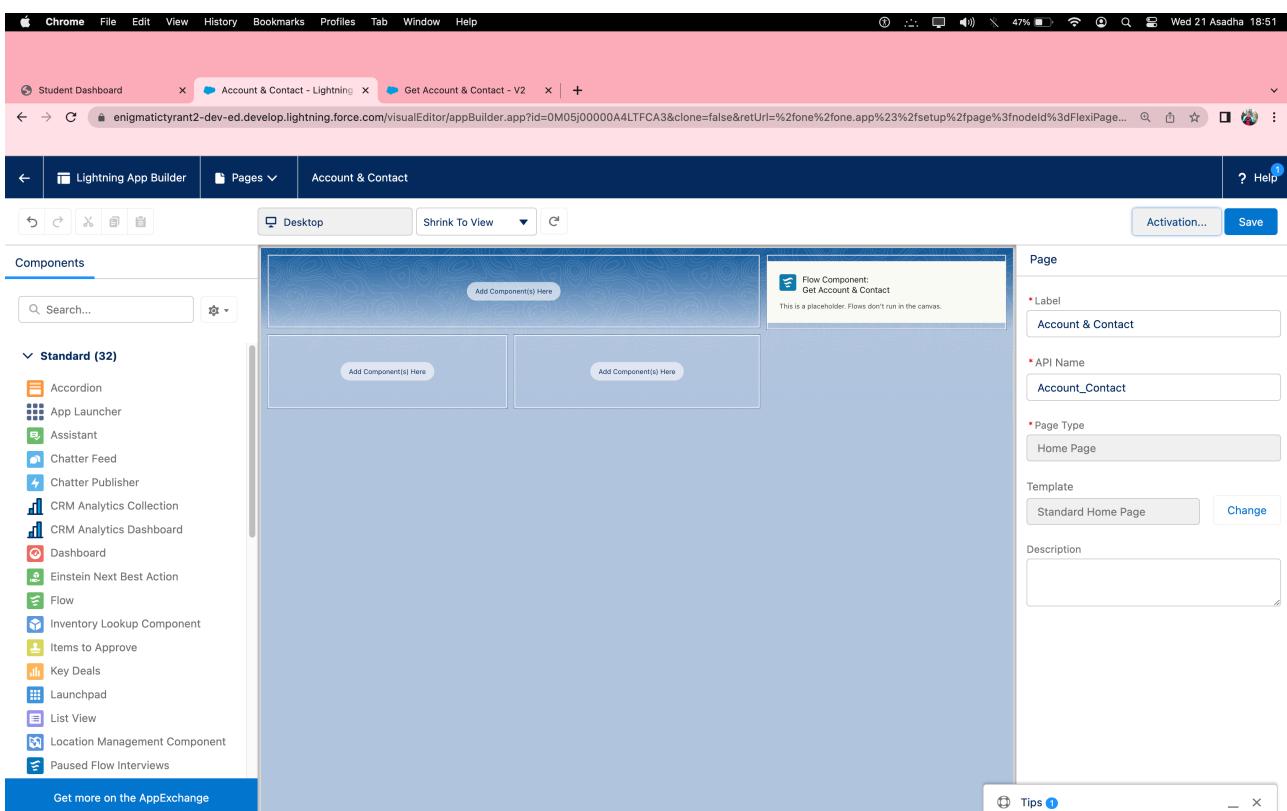


- **To Create Lightning Home Page:**

1. Click on setup gear.
2. Now search for lightning App builder.
3. And select New option
4. In create a new lightning page select Home page.
5. Select Next
6. Give the label name - Account & Contact
7. Choose a standard home page.
8. Now in the component section select flow and drag down it to Corner of the page.
9. At the right side select the flow Get Account & Contact.
10. at the right side top of the page click on Save.
11. You will get the populate notification and click on activate.
12. you will get an activation pop up select App and profile.
13. Select Sales app in lightning app selection.

14. In profiles select System administrator, Standard user, Standard platform user.
15. Save it.
16. Now click on app launcher and search for Sales App
17. At the right side corner you can find a Pencil icon to personalize navigation click on that.
18. Click on add more items and in available items click on all and search for home.
19. Move the home page to top and click on save.

- **To Send An Email Alert To The Customer Once Order Is Confirmed:**



To send an email alert to the customer once order is confirmed.

1. Navigate to setup click on object manager
2. Select sales order as object
3. click on fields and relationships
4. Select Email id field and click on edit
5. Than in general options select Required Field and save it.

Now Create a new field Order confirmed:

1. On the same object sales order create a Picklist field.
2. Give the label name as Order confirmed.
3. And in Values give 1) Yes
2) No
4. Make it as Required field.

The screenshot shows the Salesforce Setup interface. The left sidebar is titled 'Object Manager' and lists various options like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Sales order Custom Field Order confirmed'. It shows the 'Custom Field Definition Detail' section with the following details:

- Field Information:** Field Label: Order confirmed, Field Name: Order_confirmed, API Name: Order_confirmed_c, Description: (empty), Help Text: (empty), Data Owner: (empty), Field Usage: (empty), Data Sensitivity Level: (empty), Compliance Categorization: (empty).
- General Options:** Required: , Default Value: (empty).
- Picklist Options:** Restrict picklist to the values defined in the value set: (unchecked), Controlling Field: [New].

At the bottom, there is a 'Picklist Values Used' section which is currently empty.

The screenshot shows the Salesforce Setup interface. The left sidebar is titled 'Object Manager' and lists various options like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Sales order'. It shows the configuration for the 'Order confirmed' picklist field:

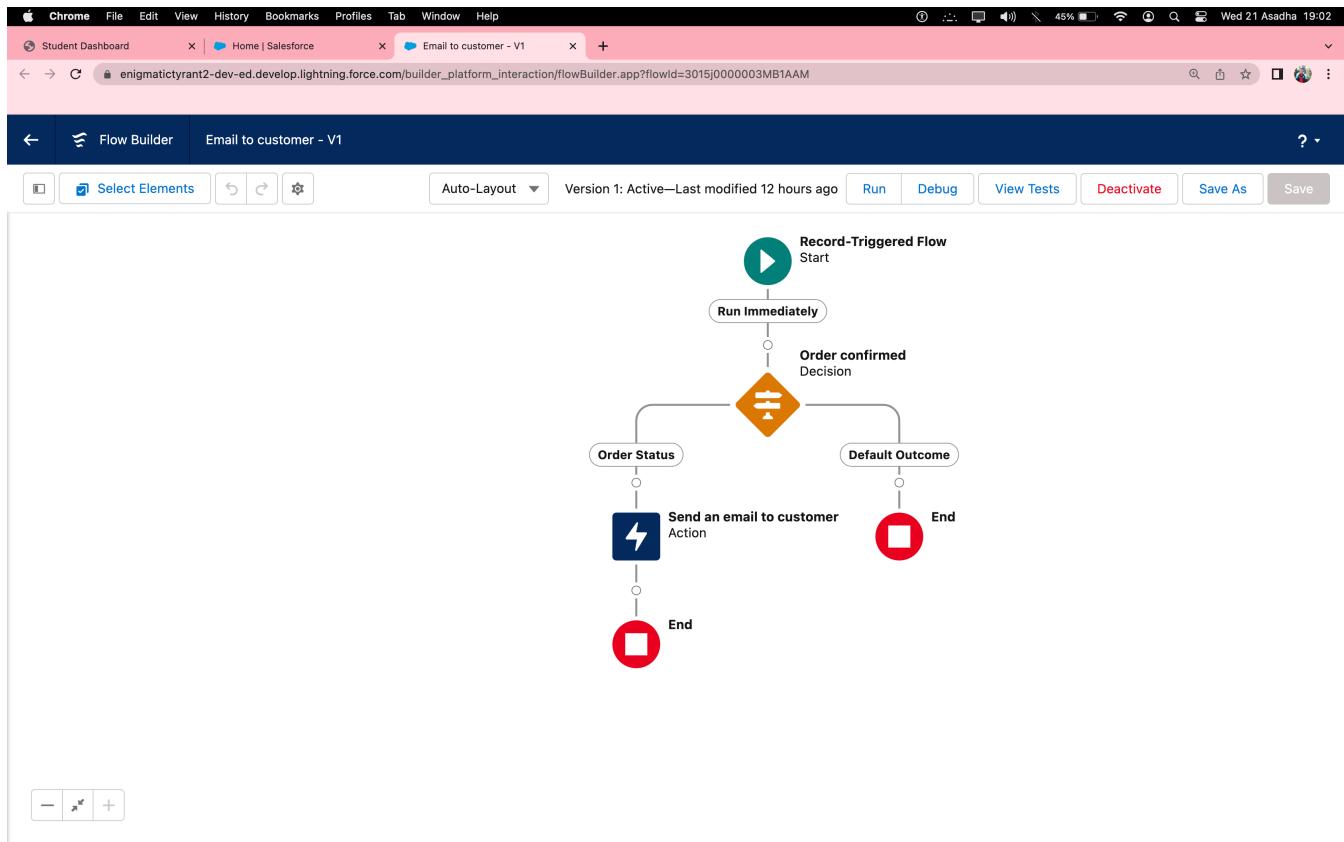
- Picklist Options:** Restrict picklist to the values defined in the value set: (unchecked), Controlling Field: [New].
- Picklist Values Used:** Active picklist values: 2 (1,000 max), Inactive picklist values: 0 (4,000 max).
- Field Dependencies:** No dependencies defined.
- Validation Rules:** No validation rules defined.
- Values:** A table showing two entries:

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit Del Deactivate	Yes	Yes	<input type="checkbox"/>	Assigned dynamically	Ayush Bhagat, 12/07/2023, 7:00 am
<input type="checkbox"/> Edit Del Deactivate	No	No	<input type="checkbox"/>	Assigned dynamically	Ayush Bhagat, 12/07/2023, 6:59 am
- Inactive Values:** No Inactive Values values defined.

At the bottom, there is a 'Back To Top' link and a note: 'Always show me ▾ more records per related list'.

- **To Create Record Trigger Flow:**

1. Click on setup and search for flows.
2. Than click on new flow
3. Select Record Trigger flow as your flow
4. In Object search for Sales order
5. In Configure Trigger select A record is created
6. In set entry conditions All conditions are Met
7. In fields search for Order confirmed
8. Operator - Equals
9. In values select - Yes
10. In optimize the flow for Select Action and related Record.
11. Now click on add element and in Logic section
select Decision element
12. Give the element name as Is Order confirmed
13. In Outcome Details Label as Order Status
14. In condition requirement All Conditions are Met
15. In resource select \$Record than field as Order confirmed
16. Operator - Equals
17. Value as Yes
18. Click on done
19. Now Below the Order status Click on add element and
select Action as your element.
20. Now in action select Send email
21. Give the label as Send an email to customer
22. Now in body select \$Record and than select Id
23. Move your cursor to front before \$record and give the text
as Hi Your order with than comes \$record is confirmed
24. In recipients email address list - select Record and than
Email id as your field
25. In subject give it as - Order confirmed.
26. Click on done.
27. Save the flow as Email to customer
28. And activate the flow.



○ Milestone 11 Triggers:

A Trigger is a segment of Apex code which executes before or after inserting or modifying a Salesforce record based on the condition provided. There are different types of triggers based on the action going to be performed.

They are Before Triggers and After Triggers. Triggers allow modification of another record of the same type or different type.

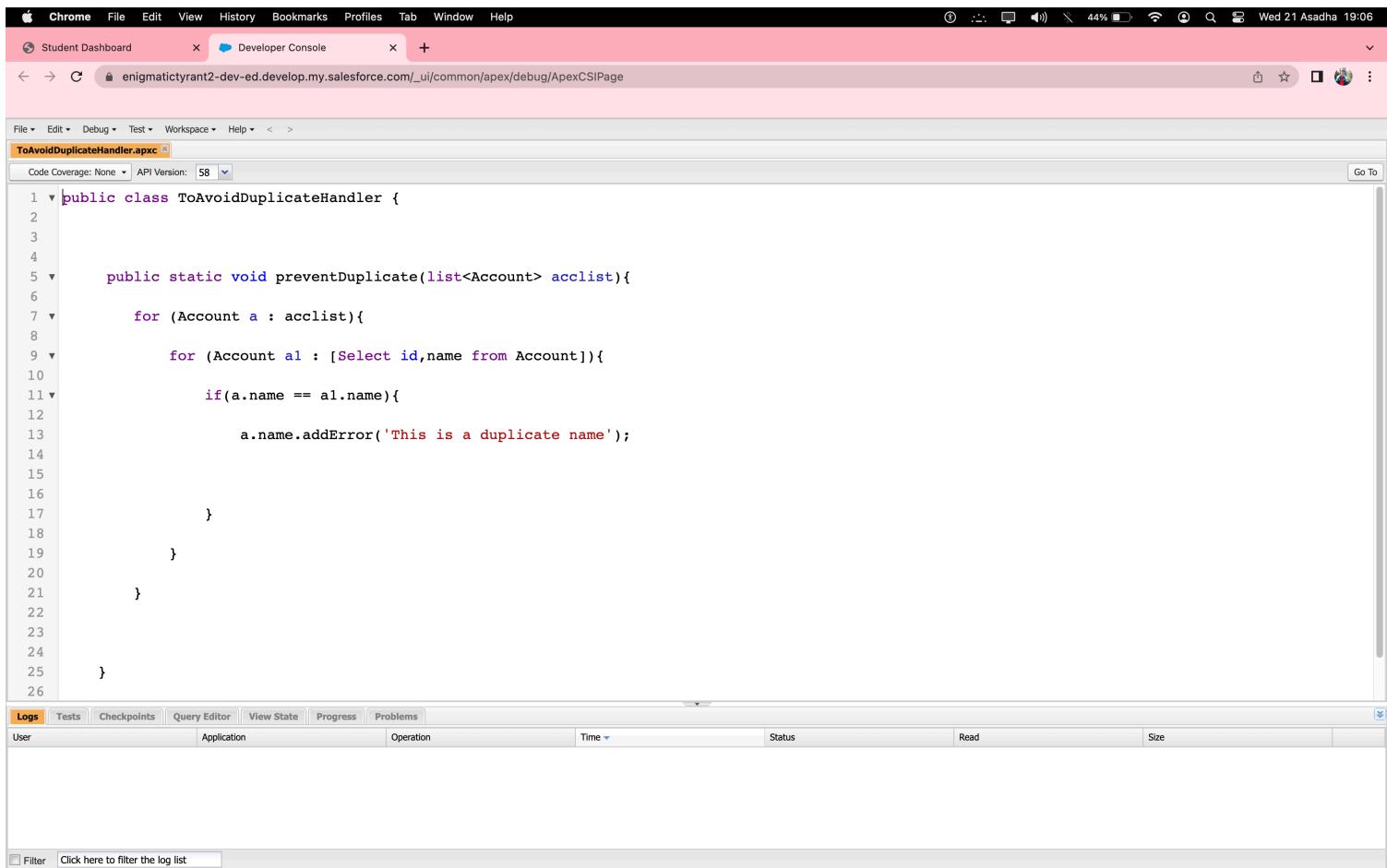
“Before” Apex Triggers. These are used to update or validate the value in a record before you save it to your Salesforce database.

“After” Apex Triggers. These are used to access the values contained within a record and use that value to make changes to other records in your Salesforce database. Unlike “Before” triggers, “After” triggers are read-only.

Trigger On Account To Prevent Duplicate Name

Trigger on Account to prevent Duplicate Name.

1. Click on Setup and select developer console
2. Click on file and than New
3. Select Apex Class give the name as Toavoidduplicateshandler



The screenshot shows the Salesforce Developer Console interface in a Chrome browser. The title bar indicates the URL is enigmactictyrant2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The main area displays the code for the apex class ToAvoidDuplicateHandler. The code checks for duplicate account names by comparing the 'name' field of each account against all other accounts in the list. If a duplicate is found, an error message is added to the account's name field.

```
1 public class ToAvoidDuplicateHandler {  
2  
3  
4  
5     public static void preventDuplicate(list<Account> acclist){  
6         for (Account a : acclist){  
7             for (Account a1 : [Select id,name from Account]){  
8                 if(a.name == a1.name){  
9                     a.name.addError('This is a duplicate name');  
10                }  
11            }  
12        }  
13    }  
14  
15}  
16  
17}  
18  
19}  
20  
21}  
22  
23  
24  
25}  
26}
```

Below the code editor, there is a logs section with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is selected. The log table has columns for User, Application, Operation, Time, Status, Read, and Size. A filter input field at the bottom left says "Click here to filter the log list".

• Trigger:

1. Click on Setup and select developer console
2. Click on file and than New
3. Select Apex trigger give the name as Toavoidduplicates
4. Sobject as Account.

```
1 trigger toavoidduplicates on Account (before insert) {
2
3
4
5 if(trigger.isBefore){
6
7 ToAvoidDuplicateHandler.preventDuplicate(trigger.new);
8
9 }
10
11
12
13 }
```