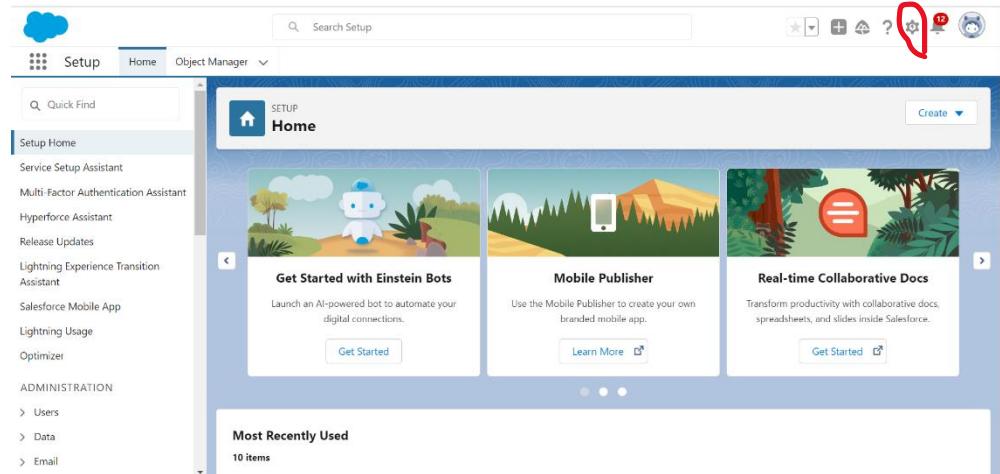


Salesforce

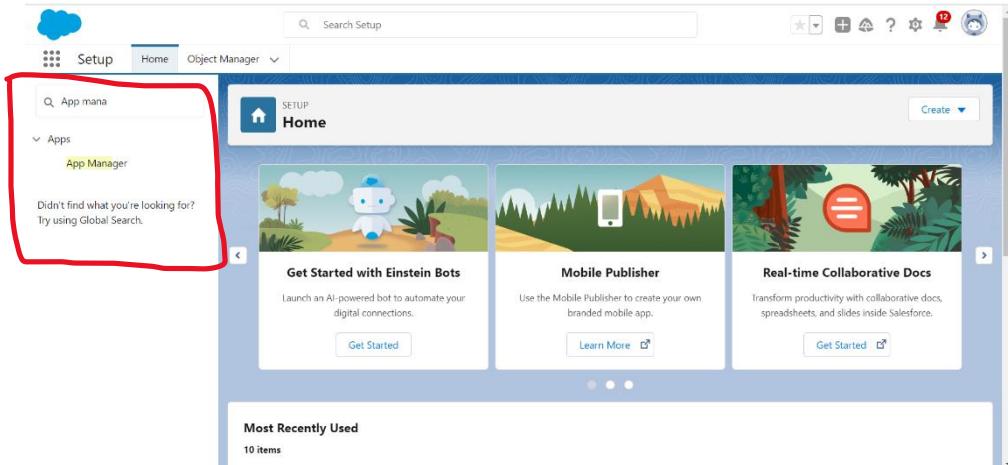
Day-01

1. To create an app in salesforce

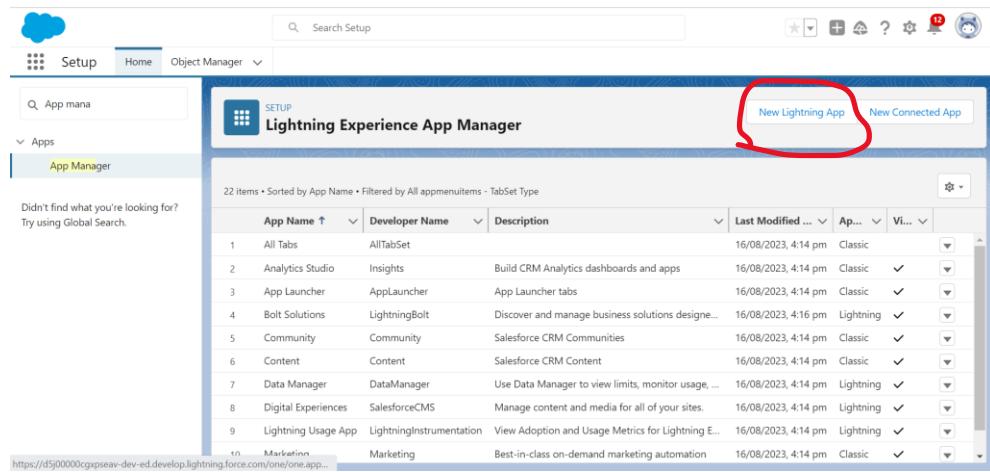
- Go to setup  from by clicking on this icon.



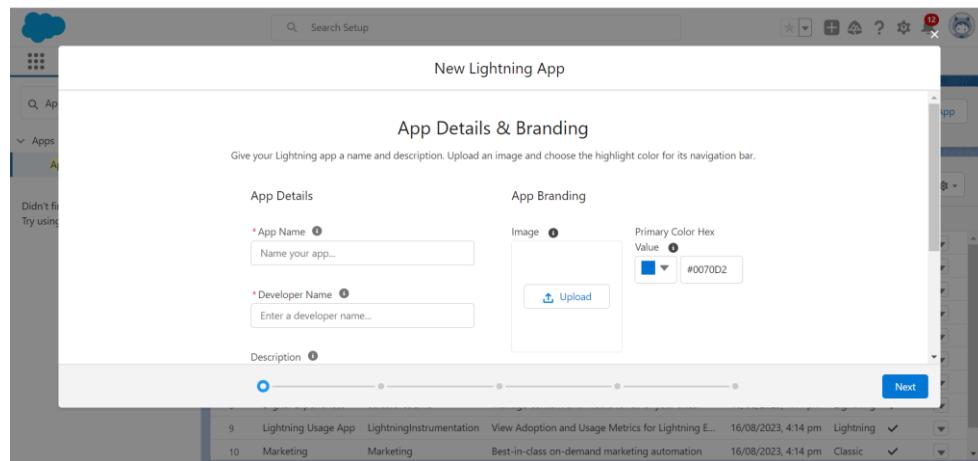
- Click on Quick search box and type **App manager**.



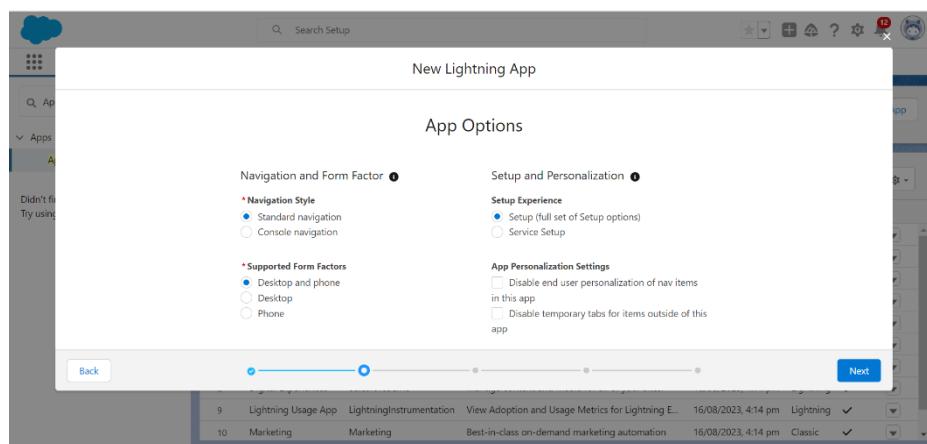
- Open App manager by clicking on it.
- Click on **New Lightning App** on right hand side of the screen.



- Enter app details like name, developer name, and photos.

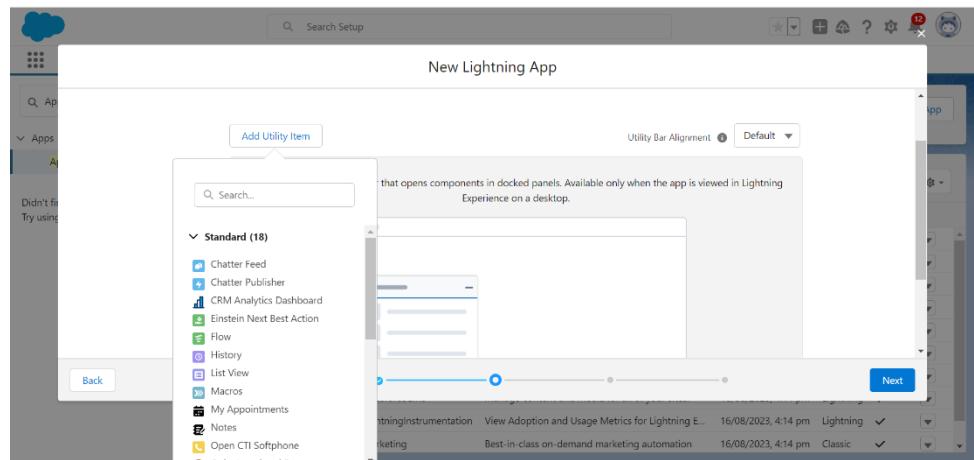


- Click next.
- Choose navigation style **Standard navigation** or **console navigation** as per requirements.
- Select setup experience **Setup (full set of setup options)**.
- Select supported form factors **desktop and phone**.



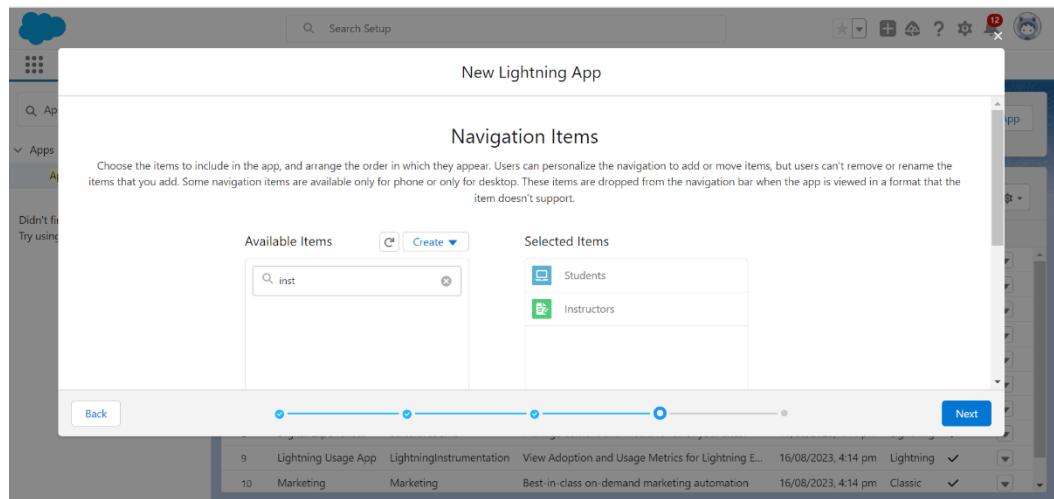
- Click on next.

- Add utility item like chatter feed (this is shown bottom of the frontend)



app).

- Click next.
- In next window add available items like created custom object from other app can be added example: - students, instructor. We can select or we can keep blank, and we can create later.

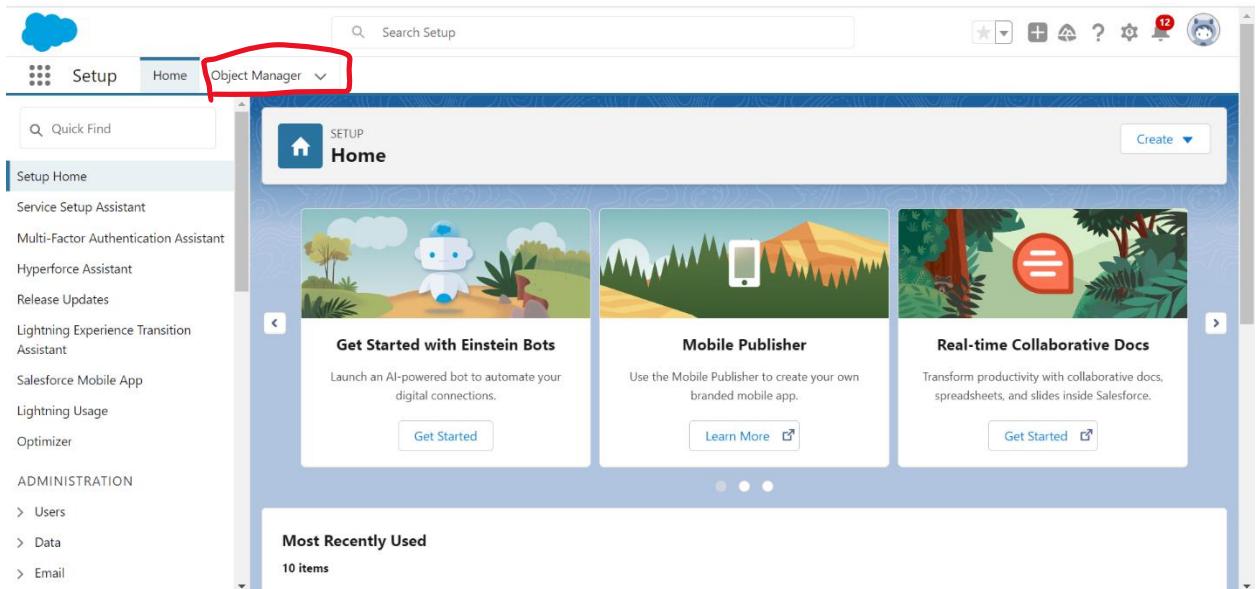


- Click on next.
- Next, we have to select user profile it depends on project but always choose system administrators.

- Click on save & finish.
- New app will be created.

2. To create custom Objects

- Click on this  icon and open step up.
- Click on **object manager**.



- Click on create and from drop down select **custom object**.

- Enter label, plural label, description.

New Custom Object

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label	example	Example: Account
Plural Label		Example: Accounts
Starts with vowel sound	<input type="checkbox"/>	

The Object Name is used when referencing the object via the API.

Object Name	example	Example: Account
-------------	---------	------------------

Description

- Enter record name label and format like IDs and use below format.

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name	example Id	Example: Account Name
Data Type	Auto Number	
Display Format	e-(0000)	Example: A-(0000) What Is This?
Starting Number	1	

Optional Features

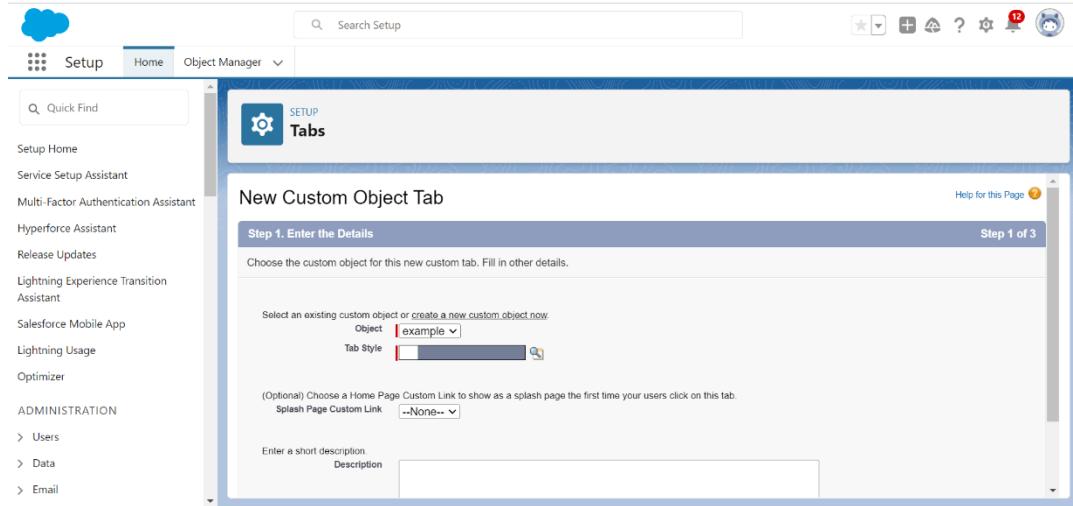
Allow Reports
 Allow Activities
 Track Field History
 Allow in Chatter Groups
 Enable Licensing [?](#)

Object Classification

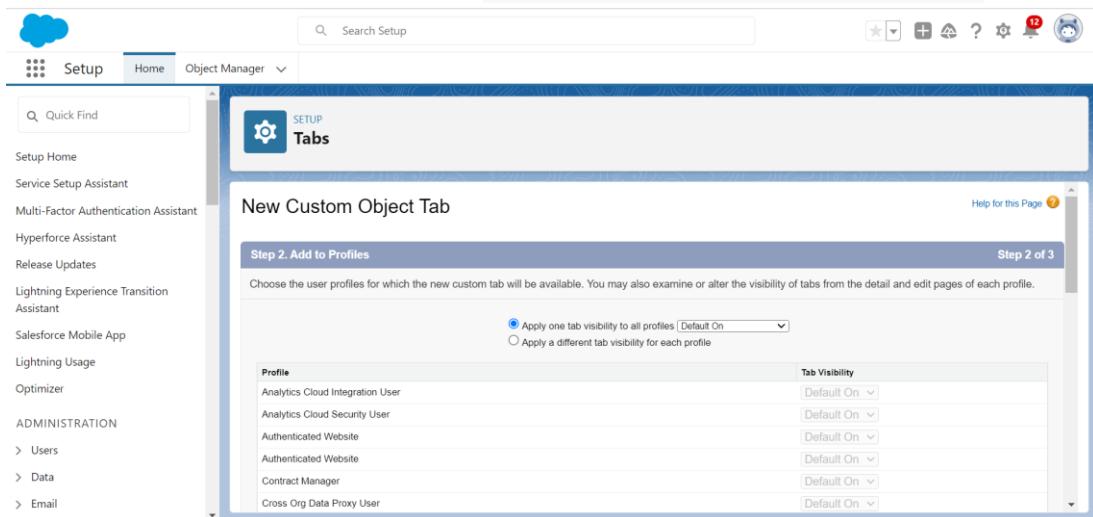
When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more](#).

- Check this all boxes.
- Always check box **Launch New Custom Tab Wizard after saving this custom object.**

- Click on save new custom object will be created.
- Now next Tab page will be open. Tab should be created to show the custom object on frontend.



- Select table style by clicking on it.
- Click on next.
- Now select profile. For now, select Apply one tab visibility to all profiles.



- Click on next.

- Now check the box in which we want to add this custom app. For now, I want to add the custom app to “Zahid khan tech school”. So select this app or any custom app in which you want to add.

Custom App	<input type="checkbox"/> Include Tab
Platform (standard__Platform)	<input type="checkbox"/>
Sales (standard__Sales)	<input type="checkbox"/>
Service (standard__Service)	<input type="checkbox"/>
Marketing (standard__Marketing)	<input type="checkbox"/>
Sample Console (standard__ServiceConsole)	<input type="checkbox"/>
High Volume Customer Portal User	<input type="checkbox"/>
Authenticated Website User	<input type="checkbox"/>
App Launcher (standard__AppLauncher)	<input type="checkbox"/>
Community (standard__Community)	<input type="checkbox"/>

Append tab to users' existing personal customizations

Previous Save Cancel

- Now click on save.
- Go to the app by clicking on app launcher and refresh the page you will see the **examples** custom object in your app.

The screenshot shows the Salesforce app launcher interface. At the top, there's a search bar with placeholder text 'Search...'. Below it, a navigation bar includes the cognizant logo, the organization name 'Zahid khan Tech sc...', and tabs for 'Students', 'Instructors', and 'examples'. On the far right of the navigation bar are various icons for account management and notifications. The main content area is titled 'examples' and 'Recently Viewed'. It displays a message stating '0 items • Updated a few seconds ago' and 'example Id'. Below this, a note says 'You haven't viewed any examples recently. Try switching list views.' At the bottom of the content area, there's a link to 'Chatter Feed'.

Day-2

3. To create fields for custom object

- Click on this icon and open step up.
- Now click on object manager.
- Search for custom object from quick find.

The screenshot shows the Salesforce Object Manager interface. At the top, there's a search bar with placeholder text 'Search Setup' and a setup icon. Below it, a navigation bar includes 'Setup', 'Home', and 'Object Manager'. On the far right of the navigation bar are various icons for account management and notifications. The main content area is titled 'Object Manager' and shows a table with one item. The table has columns: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The single row shows 'example' as the label, 'example__c' as the API name, 'Custom Object' as the type, and '24/08/2023' as the last modified date. The 'DEPLOYED' column contains a checkmark icon.

- Click on **example object**.
- Now click on Filed & relationships.
- Click on new.

SETUP > OBJECT MANAGER
example

Fields & Relationships
4 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
example Id	Name	Auto Number		▼
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		▼

- Next choose the field type.
- There are numbers of field available. Which is given below.

SETUP > OBJECT MANAGER
example

Specify the type of information that the custom field will contain.

Data Type

None Selected Select one of the data types below.

Auto Number A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.

Formula A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

Roll-Up Summary A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.

Lookup Relationship Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.

Master-Detail Relationship Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:

- The relationship field is required on all detail records.
- The ownership and sharing of a detail record are determined by the master record.
- When a user deletes the master record, all detail records are deleted.

The screenshot shows a list of field types in the Salesforce setup interface:

- Checkbox: Allows users to select a True (checked) or False (unchecked) value.
- Currency: Allows users to enter a dollar or other currency amount and automatically formats the field as a currency amount. This can be useful if you export data to Excel or another spreadsheet.
- Date: Allows users to enter a date or pick a date from a popup calendar.
- Date/Time: Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.
- Email: Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.
- Geolocation: Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.
- Number: Allows users to enter any number. Leading zeros are removed.
- Percent: Allows users to enter a percentage number, for example, '10' and automatically adds the percent sign to the number.
- Phone: Allows users to enter any phone number. Automatically formats it as a phone number.
- Picklist: Allows users to select a value from a list you define.
- Picklist (Multi-Select): Allows users to select multiple values from a list you define.
- Text: Allows users to enter any combination of letters and numbers.
- Text Area: Allows users to enter up to 255 characters on separate lines.
- Text Area (Long): Allows users to enter up to 131,072 characters on separate lines.
- Text Area (Rich): Allows users to enter formatted text, add images and links. Up to 131,072 characters on separate lines.
- Text (Encrypted): Allows users to enter any combination of letters and numbers and store them in encrypted form.
- Time: Allows users to enter a local time. For example, "2:40 PM", "14:40", "14:40:00", and "14:40:50.600" are all valid times for this field.
- URL: Allows users to enter any valid website address. When users click on the field, the URL will open in a separate browser window.

At the bottom right of the screen are 'Next' and 'Cancel' buttons.

- Choose any one data type and click on next.
- Now enter details of the field like Field label, length, Helping text, Check box **Required box** if value is required for this filed. Check **Unique box** if value of this filed should be unique.

The screenshot shows the 'New Custom Field' creation screen in the Salesforce Object Manager:

Object Manager > **example**

Step 2. Enter the details (Step 2 of 4)

Fields & Relationships

Field Label: Weekdays

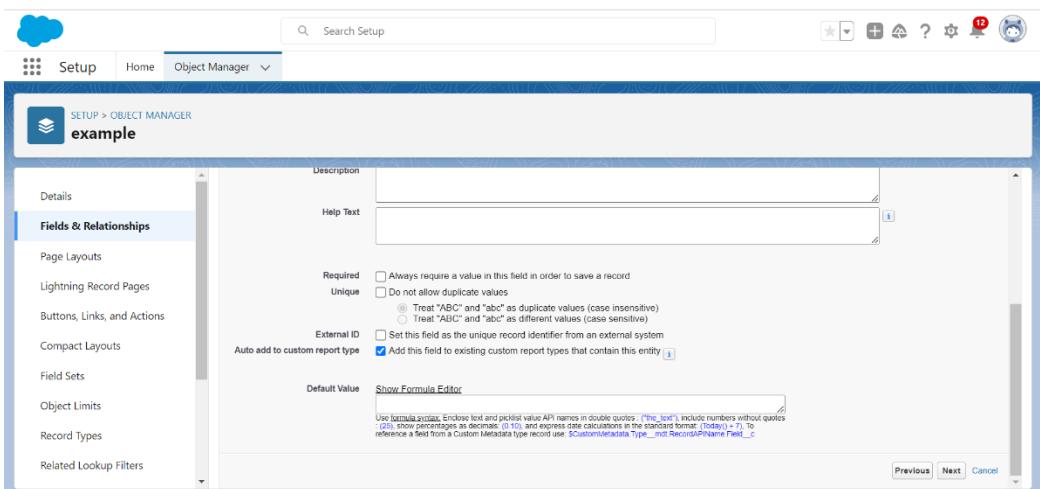
Length: 100

Field Name: Weekdays

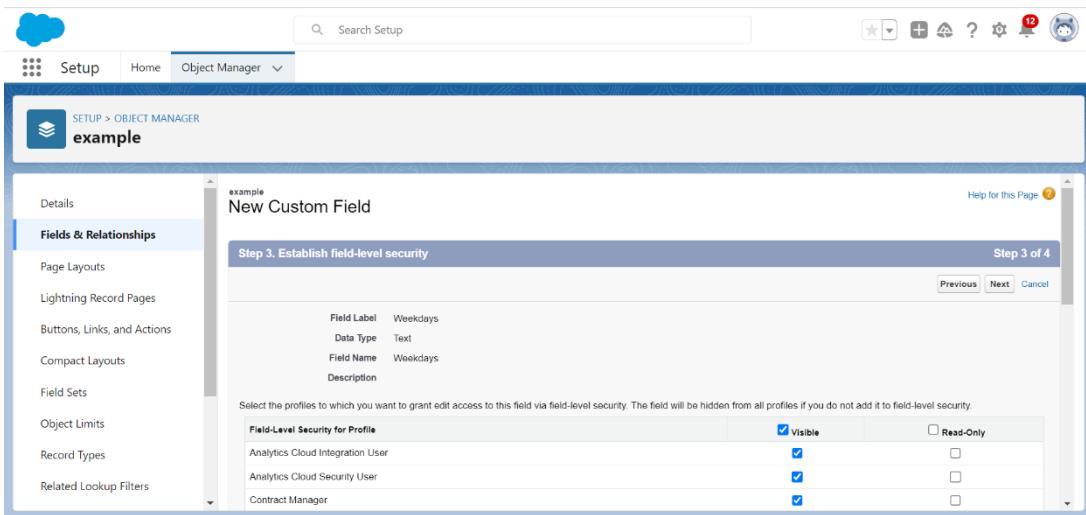
Description: (empty)

Help Text: (empty)

Buttons at the top: Search Setup, Home, Object Manager, etc. Buttons at the bottom: Previous, Next, Cancel.

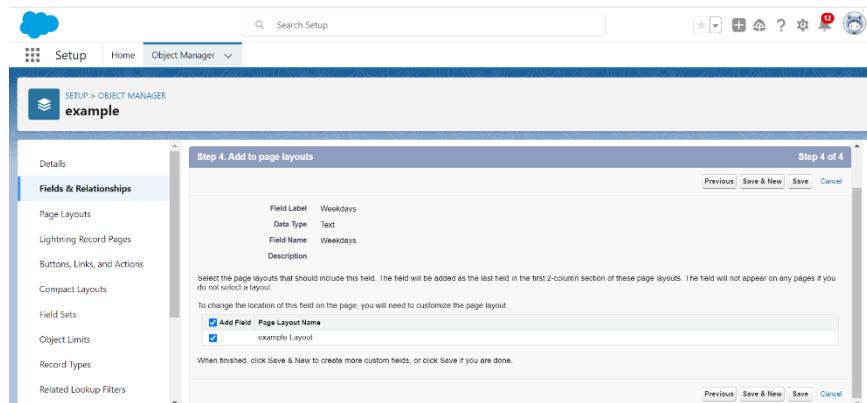


- Click on next.
- Now select filed level security for now we are selecting visible all by checking



box.

- Click on next.
- After that click on save. If we want to add more field just click on save& new.



- Now we can check by launching our app through App launcher.

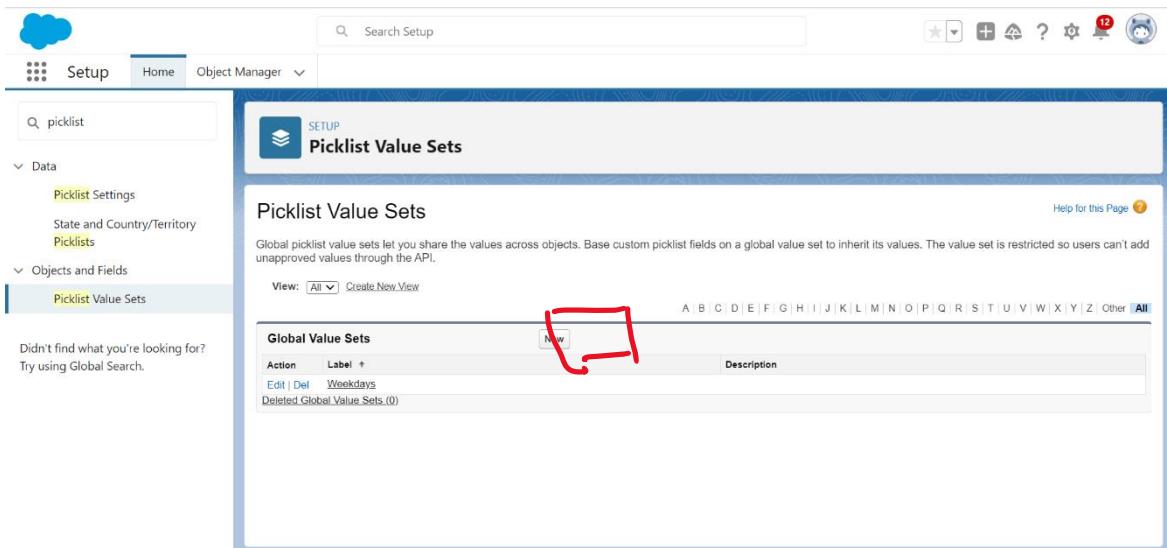
Day-03

4. Global Picklist.

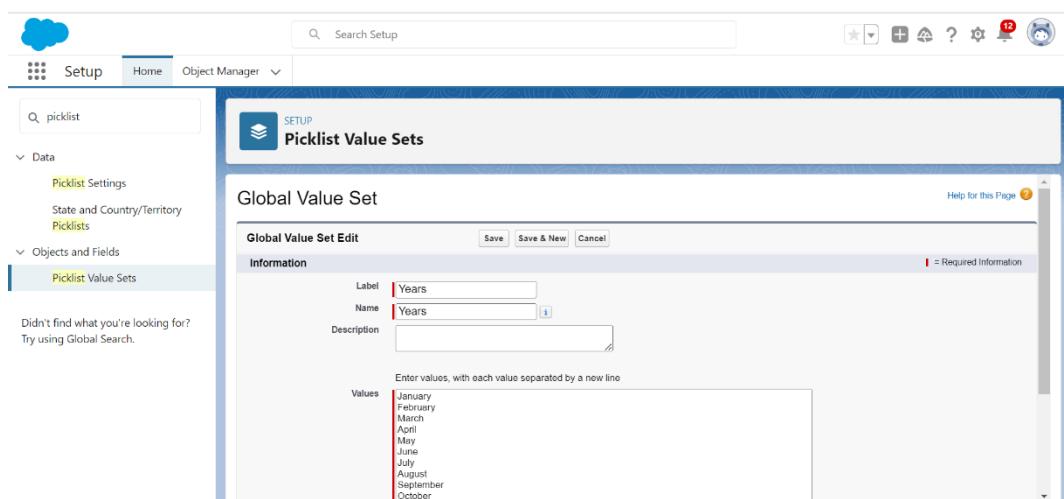
Definition: - Global Picklist is a universal picklist means we can create once and can be used 'n' number of field of any objects.

Following steps used for creating global picklist.

- Click on this  icon and open step up.
- Click on Quick find and search for **Picklist value set**.
- Click on new.



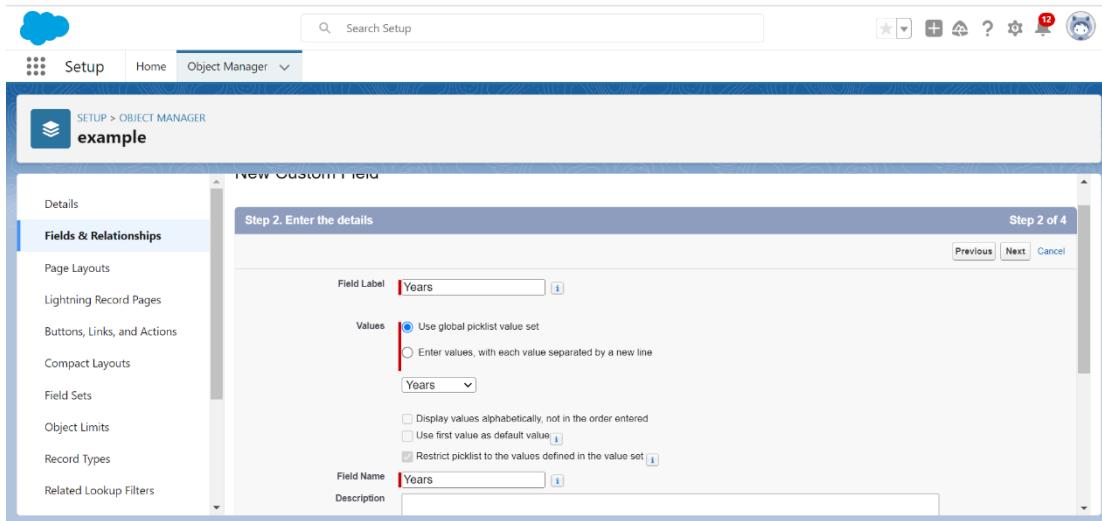
- Now enter the **Label** and **Name** and enter value for examples I am using years name.



- Now click on save. I want to make more global picklist just click on save & new.
- Global picklist created now we can use this in different objects.

How to use global picklist in example custom object.

- Go to **object manager**.
- Search for by custom object name like example.
- Click on field & relationships.
- click on new.
- Now select picklist data type.
- Click on next.
- Enter label name and always select **Use global picklist value set**.
- From drop down select global picklist which is created.



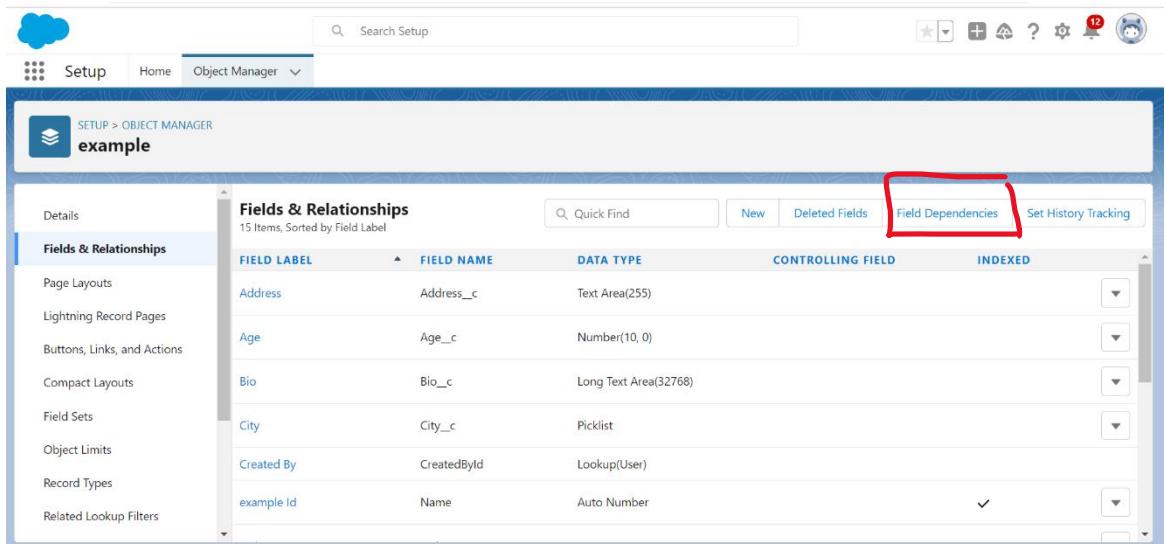
- Click on next.
- Now select field-level security. For now, check visible checkbox.
- Click on next.
- Now click on save.
- Now go to app from App launcher we will find picklist on frontend.

5. Create field dependencies.

Definition: - Field dependencies can be made using picklist to picklist or checkbox to picklist. In which one field is dependent on another field.

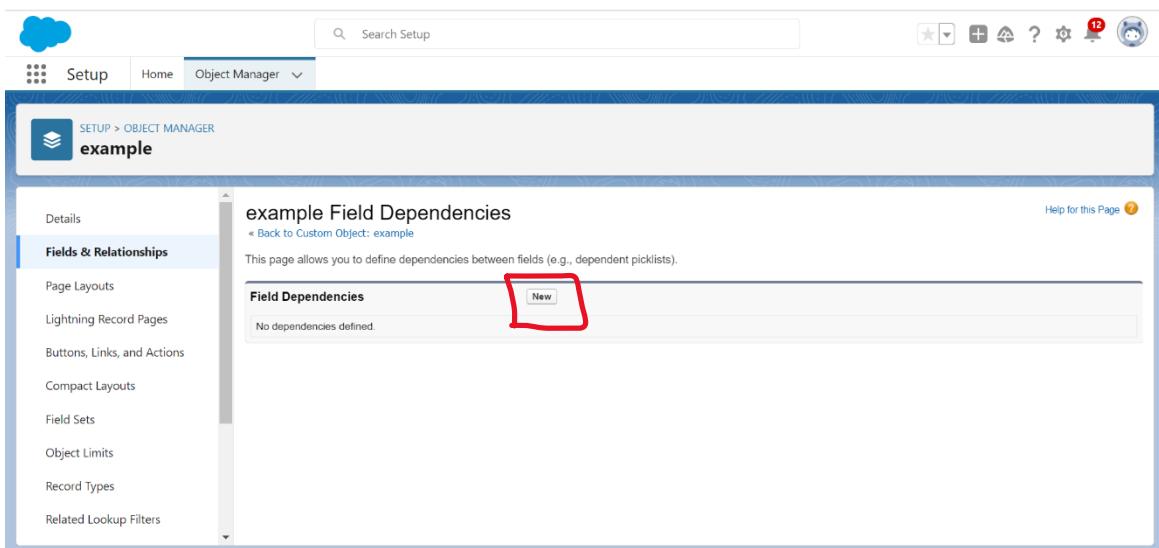
Following steps used for creating Field dependencies. (Picklist to picklist)

- Click on this  icon and open step up.
- Click on **Object manager** and search for custom object in which you want to make Field dependencies.
- Now click on Field & relationship.
- Now click on field dependencies.



The screenshot shows the Salesforce setup interface for an object named 'example'. On the left, there's a sidebar with options like Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, etc. The main area is titled 'Fields & Relationships' and lists several fields: Address, Age, Bio, City, Created By, and example Id. The 'Field Dependencies' button in the top right corner of the table header is highlighted with a red box.

- Click on new.



The screenshot shows the 'example Field Dependencies' page. It has a 'Field Dependencies' section with a 'New' button, which is highlighted with a red box. The page also includes a note: 'This page allows you to define dependencies between fields (e.g., dependent picklists).'

- Now choose **Controlling field** “The field after selecting this next filed option will be open.” (For Ex. I am choosing **state** field as controlling field)

Now choose **Depended field** “The filed which is open after selecting the option which is set in controlling field. (For Ex. I am choosing **City** field as controlling field).

New Field Dependency

Create a dependent relationship that causes the values in a picklist or multi-select picklist to be dynamically filtered based on the value selected by the user in another field.

- The field that drives filtering is called the "controlling field." Standard and custom checkboxes and picklists with at least one and less than 300 values can be controlling fields.
- The field that has its values filtered is called the "dependent field." Custom picklists and multi-select picklists can be dependent fields.

Step 1. Select a controlling field and a dependent field. Click Continue when finished.

Step 2. On the following page, edit the filter rules that control the values that appear in the dependent field for each value in the controlling field.

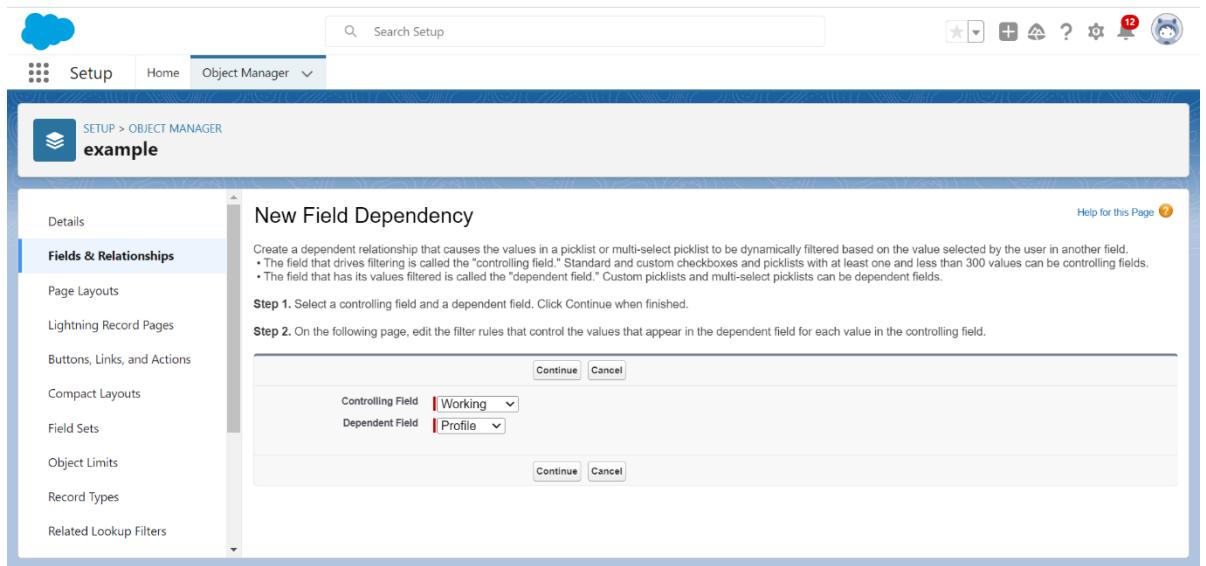
- Click on continue.
- Now select the value means after selecting a particular option which depended on field option should show. (For ex. If I select Gujarat the only city of Ahmedabad will open)
- After selection all option click on **Includes value**.

States:	Gujarat	Haryana	Karnataka	Kerala	Madhya Pradesh
City:	Ahmedabad	Ahmedabad	Ahmedabad	Ahmedabad	Ahmedabad
	Asansol	Asansol	Asansol	Asansol	Asansol
	Bengaluru	Bengaluru	Bengaluru	Bengaluru	Bengaluru
	Bhopal	Bhopal	Bhopal	Bhopal	Bhopal
	Chandigarh	Chandigarh	Chandigarh	Chandigarh	Chandigarh
	Chennai	Chennai	Chennai	Chennai	Chennai
	Durgapur	Durgapur	Durgapur	Durgapur	Durgapur
	Gurgaon	Gurgaon	Gurgaon	Gurgaon	Gurgaon
	Hyderabad	Hyderabad	Hyderabad	Hyderabad	Hyderabad
	Indore	Indore	Indore	Indore	Indore
	Jaipur	Jaipur	Jaipur	Jaipur	Jaipur
	Jodhpur	Jodhpur	Jodhpur	Jodhpur	Jodhpur
	Kochi	Kochi	Kochi	Kochi	Kochi
	Kolkata	Kolkata	Kolkata	Kolkata	Kolkata
	Lucknow	Lucknow	Lucknow	Lucknow	Lucknow

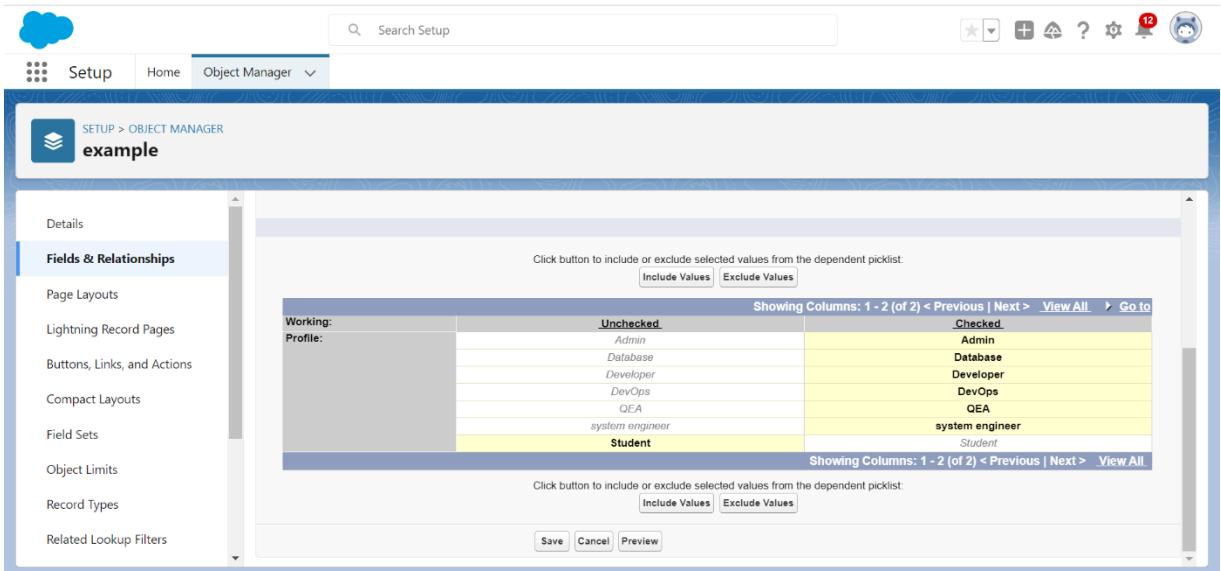
- After selecting all related options click on save.

Following steps used for creating Field dependencies. (checkbox to picklist)

- Click on field dependencies.
- Click on new.
- Now select controlling field checkbox (ex. working).
- Select dependent field picklist (ex. Profile)



- Now select what should be open after checking the box as shown in figure.
- Click on includes values.



- Now click on save.

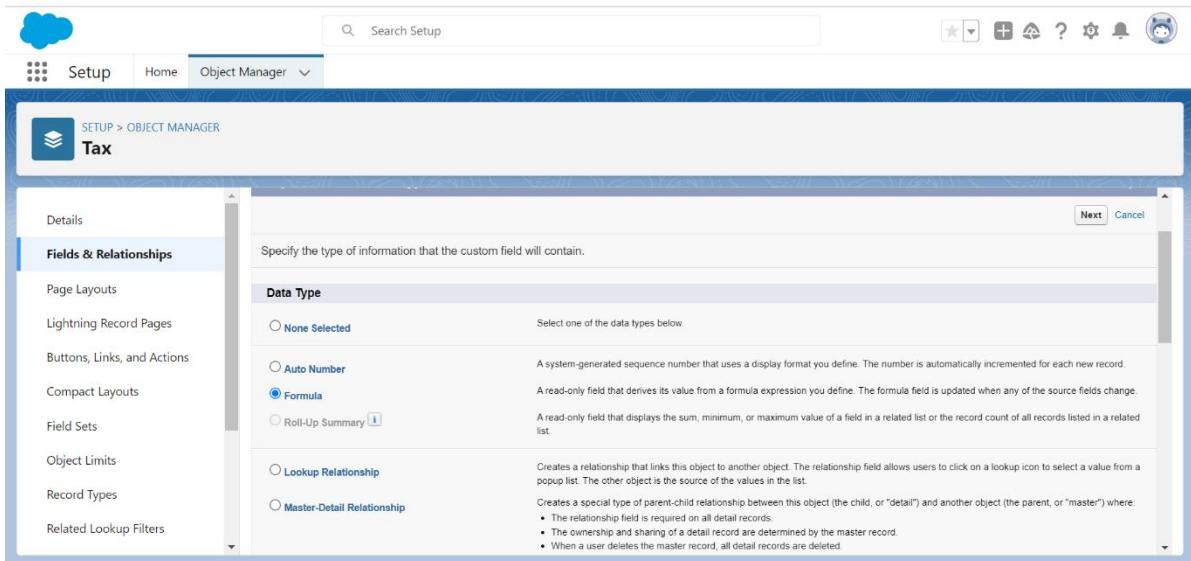
Day-04

6. Formula field

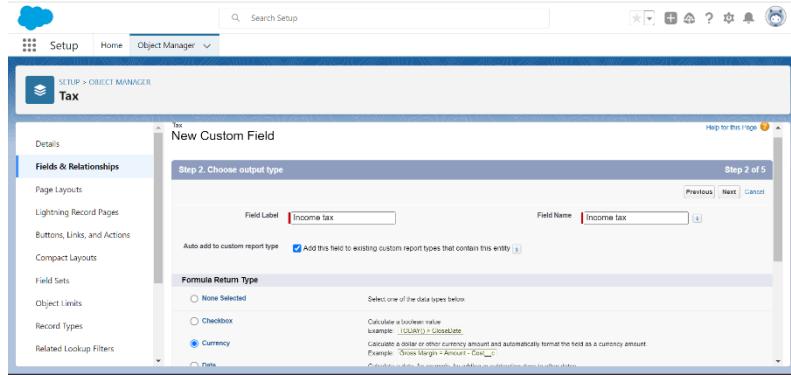
- Formula field is used to give some mathematical conditions on a particular field. (For ex. We have a salary field, and we want to calculate bonus for this we can use formula field.).
- It is read only field and populate value based on expression.

Following steps used for creating Formula field.

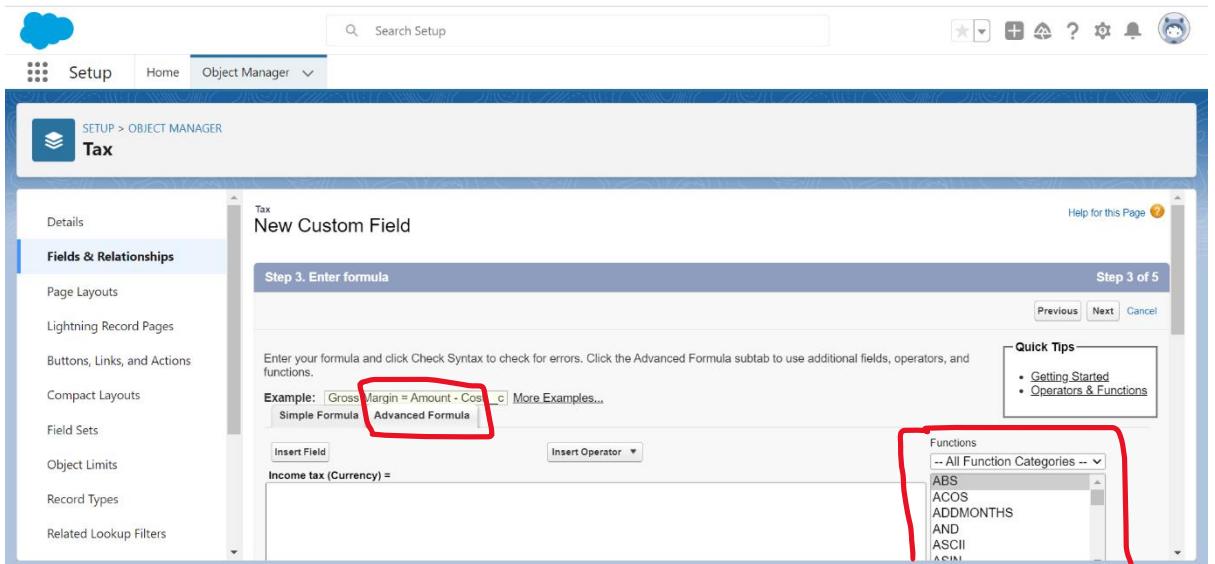
- Click on this  icon and open step up.
- Click on **Object manager** and search for custom object (ex. Tax object) .
- Click on Field & relationship.
- Now click on new.
- Select field type formula.



- Click on next.
- Now enter field label and choose formula return type. (For ex. We enter income tax as field name and currency for return type.) we are selecting currency because the salary field used currency data type.



- Click on next.
- Now click on advance formula tab.
- From function we can choose any logical or mathematical function according to our needs.



- Now choose the function and click on insert selected function. click on insert field which you want to use for formula calculations.
(For example: - here we are choosing salary field and IF function to calculate tax.)
Syntax: - IF(logical_test, value_if_true, value_if_false).

Insert salary with condition field in place of logical_test and value for true and false.

The screenshot shows the Salesforce Setup interface for creating a formula field. The left sidebar shows 'Fields & Relationships' selected. The main area has the formula `Income tax (Currency) = IF(salary < 500000 , (salary * 100)/10, (Salary * 100)/20)`. A red box highlights the 'Insert Field' button. To the right, a 'Quick Tips' panel and a function list are shown, with the 'IF' function highlighted by a red box.

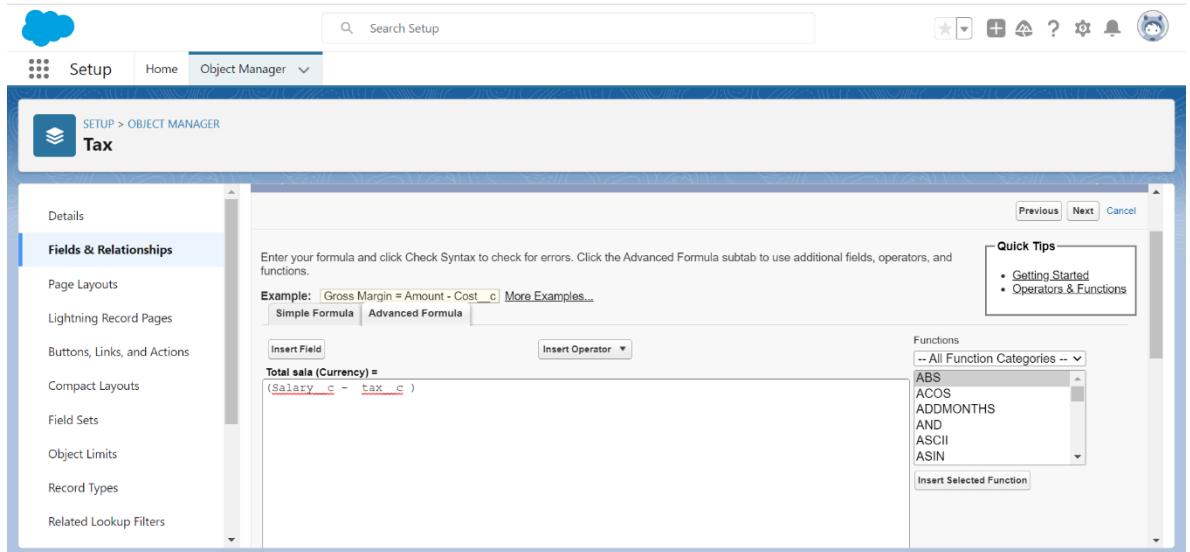
false.

- Now click on check syntax and check the box **Treat blank fields as zeroes**.
- Click on next.
- Click on visible check box and next.
- Click on save.
- Now go to front-end app and check.

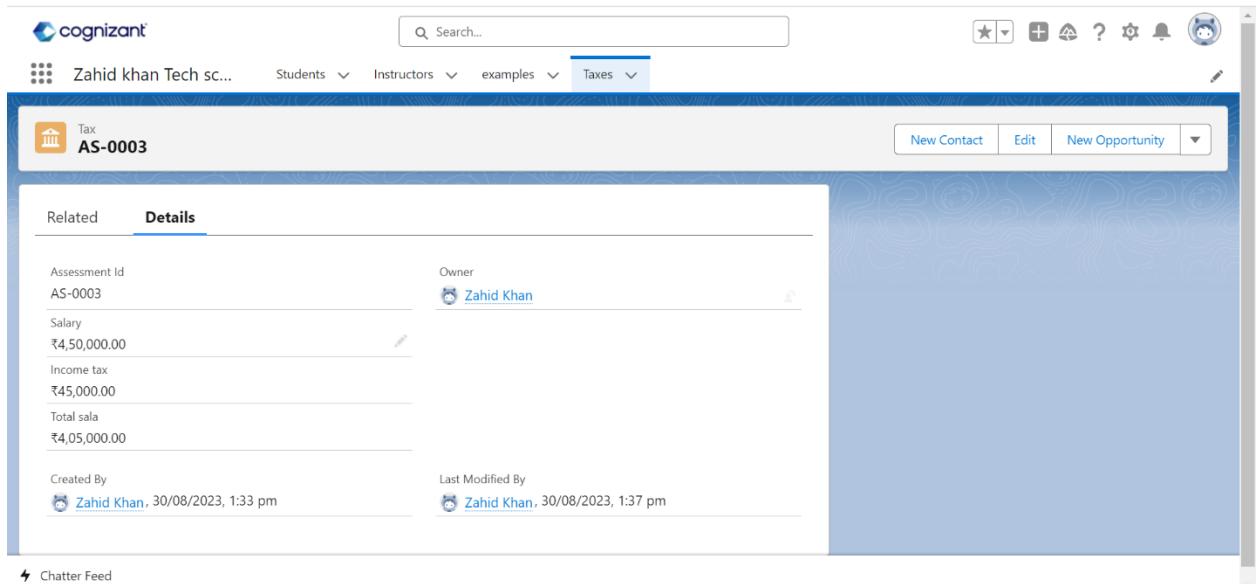
The screenshot shows a Tax record in the Cognizant application. The record ID is AS-0002. The 'Income tax' field is highlighted with a red box and contains the value ₹50,000.00. Other fields visible include Assessment Id (AS-0002), Salary (₹5,00,000.00), Owner (Zahid Khan), and Created By (Zahid Khan).

- Now create another formula field name as Total salary.
- Go to Field & relationship of tax abject and click on new.
- Now choose formula as field type.
- Click on next.
- Enter field label and field name as Total salary.
- Check currency as formula filed type.

- Click on next.
- Click on insert field and choose both salary field and income tax field and minus both for calculate total salary.



- Click on visible check box and next.
- Click on save.
- Now go to front-end app and check.

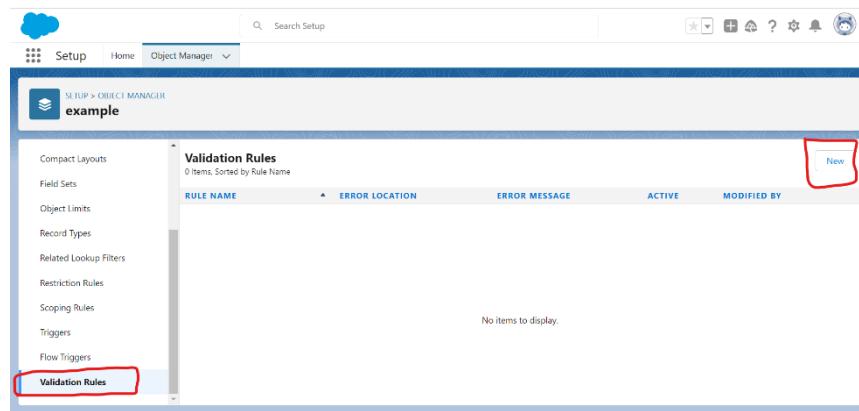


Validation Rule

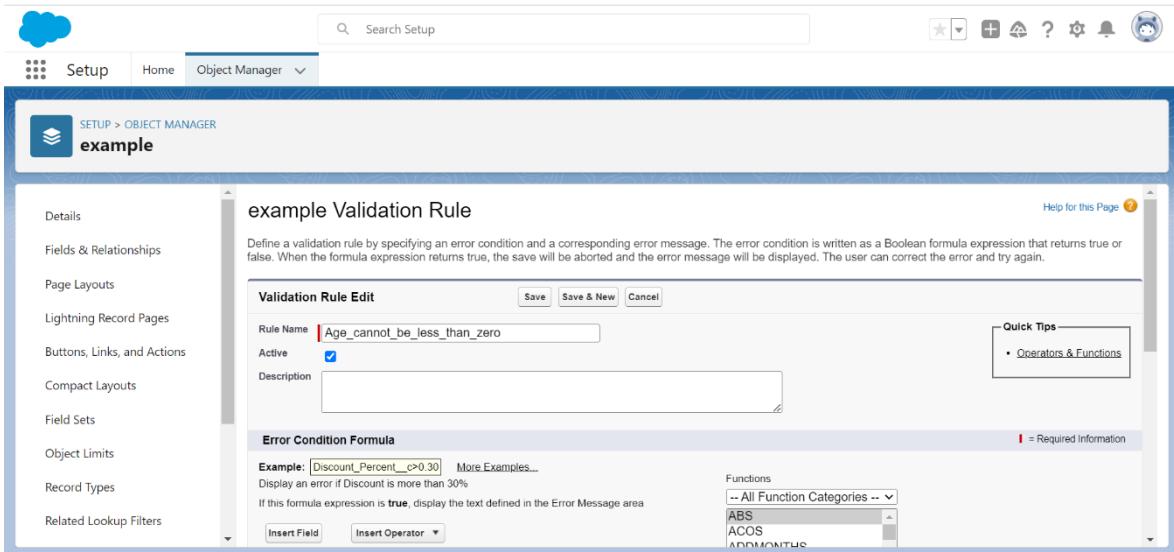
Validation rules restrict record creation & updating based on conditions. Validation work with number only.

Following steps used for creating Validation Rule.

- Click on this  icon and open step up.
- Click on **Object manager** and search for custom object (ex. example object).
- Click on validation rules option from side menu.
- Then click on new to create new validation rules.



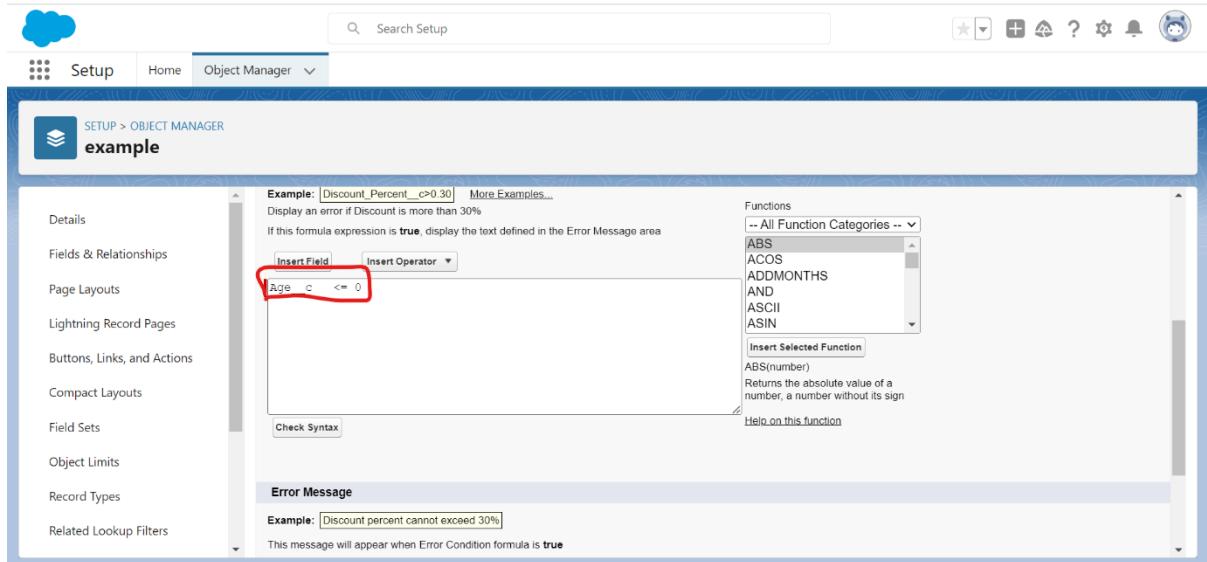
- Make validation rules for **Age field**.
- Enter the Rule name and check the active check box.



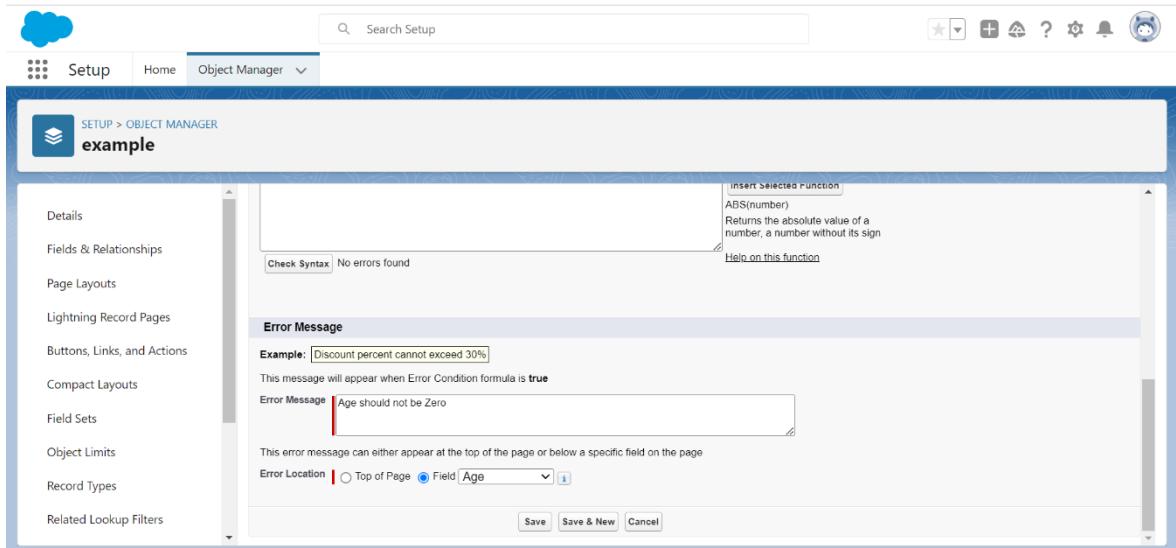
The screenshot shows the 'example Validation Rule' edit screen. The 'Rule Name' field is set to 'Age_cannot_be_less_than_zero'. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains the formula 'Discount_Percent__c>0.30'. The 'Functions' dropdown shows 'ABS' selected.

- Click on insert field to insert **age field**.

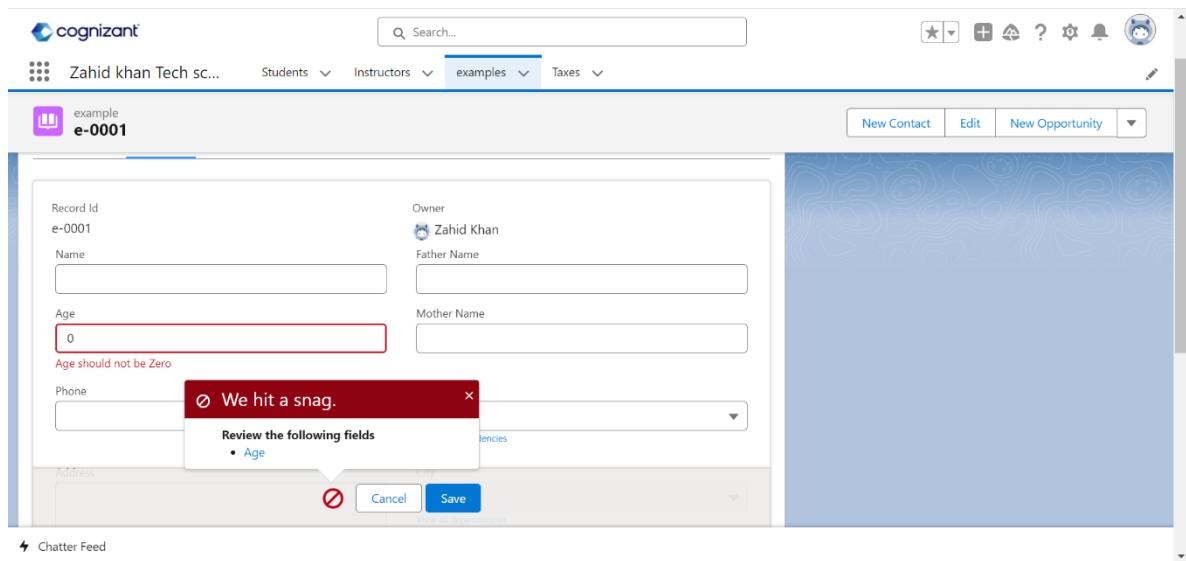
- Now make conditions for error (Age cannot less than or equal to 0). As show in screen shot.



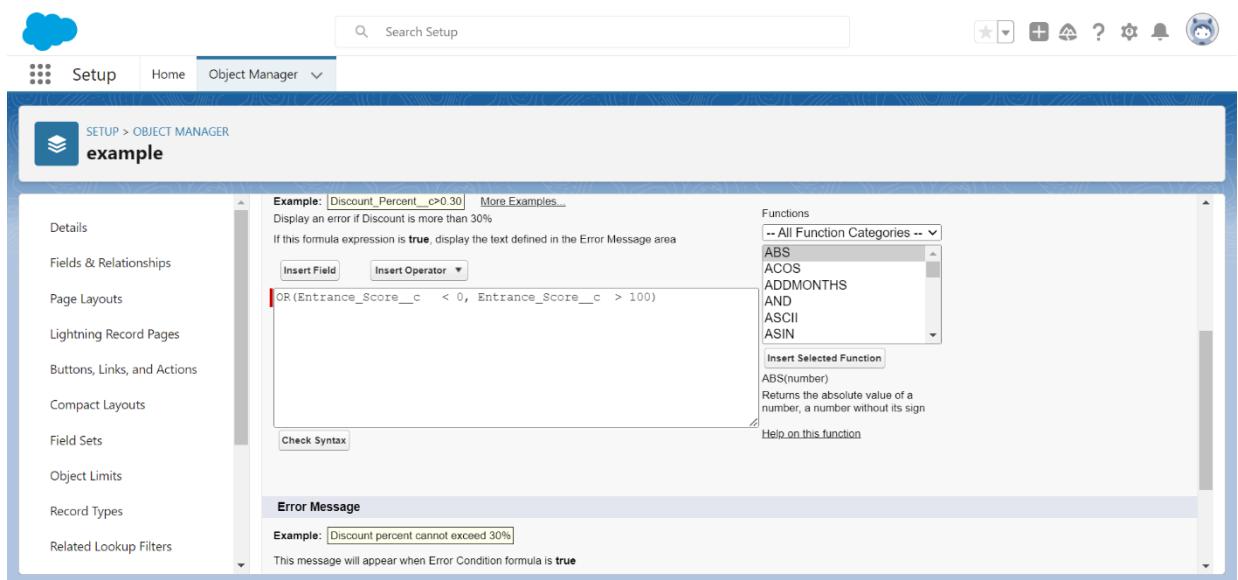
- Click on check syntax for checking the error.
- Now enter the error message which is to be displayed when condition failed.
- We must choose error location where error message should be displayed on top of the page or in field. Here we choose field and field will be age.



- click on save.
- Go to front-end and check the result.



- Now creating 2nd validation rule for same object but different field like for Entrance score field. The condition is Entrance score should be between 1 to 100.
- Select validation rule from side menu and click on new.
- Enter rule name and check the box of active.
- Now enter the validation rule as per screen shot.



- Enter the error message and error location.
- Click on save and see the result on front-end.

The screenshot shows a Salesforce page for a custom object named 'examples'. A modal window is open, displaying a form with fields: Address, City, Working, Profile, Bio, and Entrance Score. The 'Entrance Score' field contains the value '0', which is highlighted with a red border and a yellow background. An error message below the field states: 'Entrance score must between 1 to 100.' At the bottom of the modal are buttons for 'Cancel' and 'Save'.

Page Layout

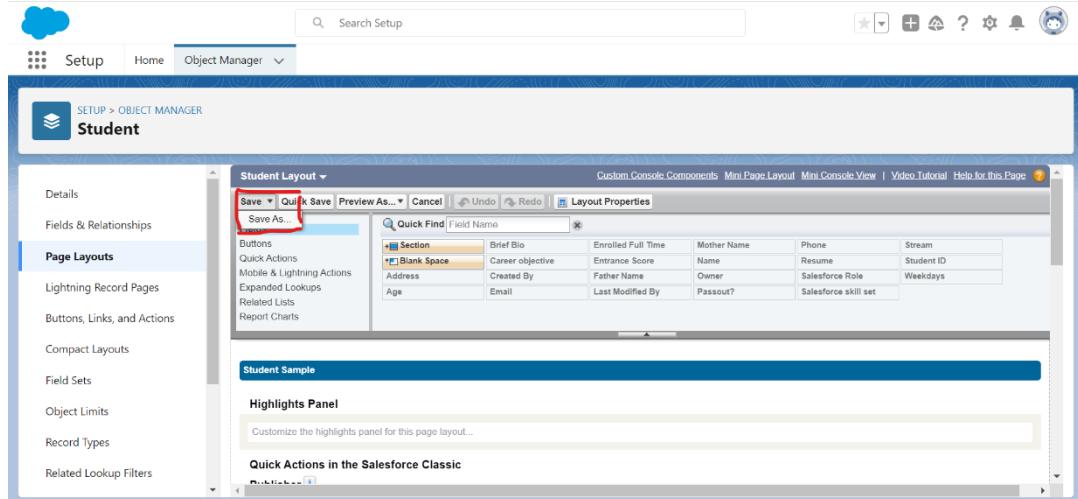
Page layouts manage field and sections.

Steps to create page layout.

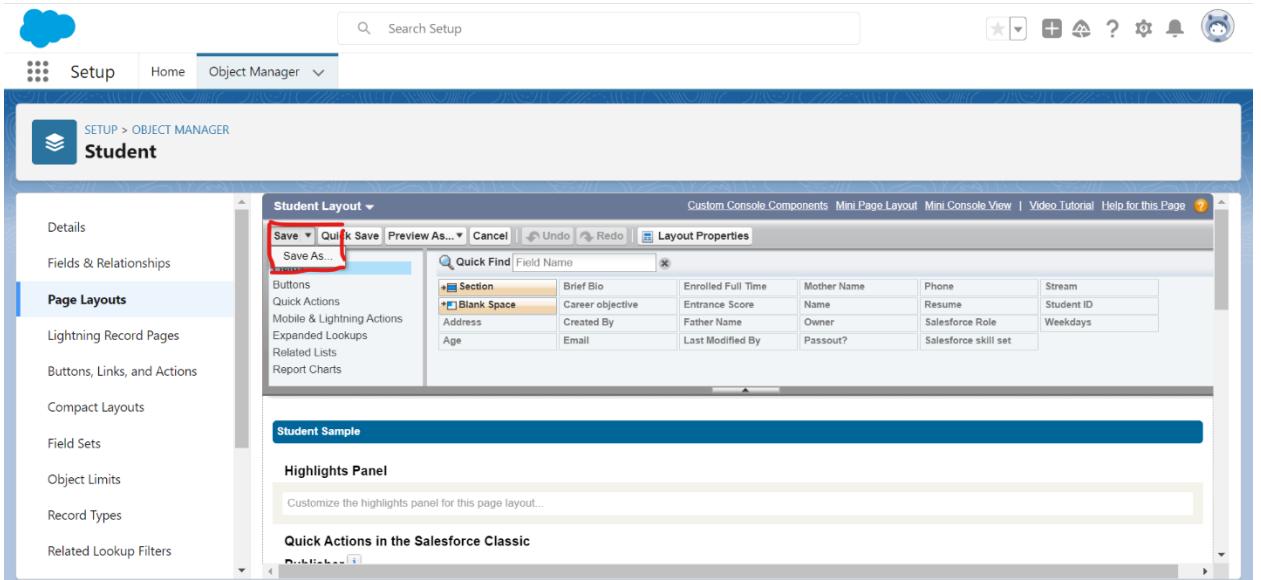
- Click on this icon and open step up.
- Click on **Object manager** and search for custom object (ex. student object).
- Click on page layout option from the right-hand side menu.
- Click on student layout.

The screenshot shows the Salesforce Setup interface under the 'Object Manager' for the 'Student' object. On the left, there is a sidebar with options like 'Details', 'Fields & Relationships', and 'Page Layouts'. The 'Page Layouts' option is selected. The main area displays a table titled 'Page Layouts' with one item: 'Student Layout'. The table has columns for 'PAGE LAYOUT NAME', 'CREATED BY', and 'MODIFIED BY'. The 'PAGE LAYOUT NAME' column shows 'Student Layout', the 'CREATED BY' column shows 'Zahid Khan, 17/08/2023, 1:26 pm', and the 'MODIFIED BY' column shows 'Zahid Khan, 30/08/2023, 3:59 pm'. There are buttons for 'New' and 'Page Layout Assignment' at the top right of the table.

- Now creating two-page layouts by cloning the student layout. The 1st clone layout is fresher and 2nd for experienced.
- To clone the layout, click on student layout and click on save arrow and select save as.



- Now enter the page layout name (ex. Fresher layout) and click on save.



- After clicking on save clone fresher layout page will be created.
- Now repeat the step number same for creating clone Experienced layout page.

- Now both fresher and experienced layout is created,

The screenshot shows the Salesforce setup interface for the 'Student' object. The left sidebar has sections like Details, Fields & Relationships, Page Layouts (which is selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main area is titled 'Page Layouts' and shows a table with three items:

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Experienced Layout	Zahid Khan, 30/08/2023, 10:53 pm	Zahid Khan, 30/08/2023, 10:53 pm
Fresher layout	Zahid Khan, 30/08/2023, 10:50 pm	Zahid Khan, 30/08/2023, 10:50 pm
Student Layout	Zahid Khan, 17/08/2023, 1:26 pm	Zahid Khan, 30/08/2023, 3:59 pm

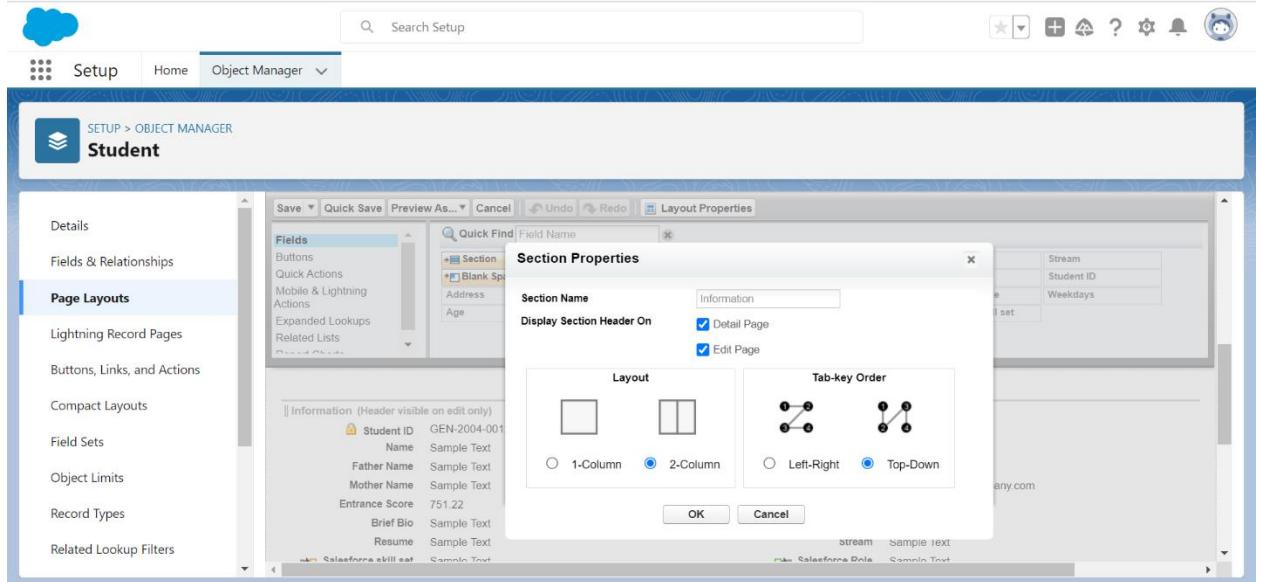
Steps to create sections in page layout.

- To create sections of the fresher layout, click on fresher layout.
- To create sections for different field we must select **sections** from quick find.

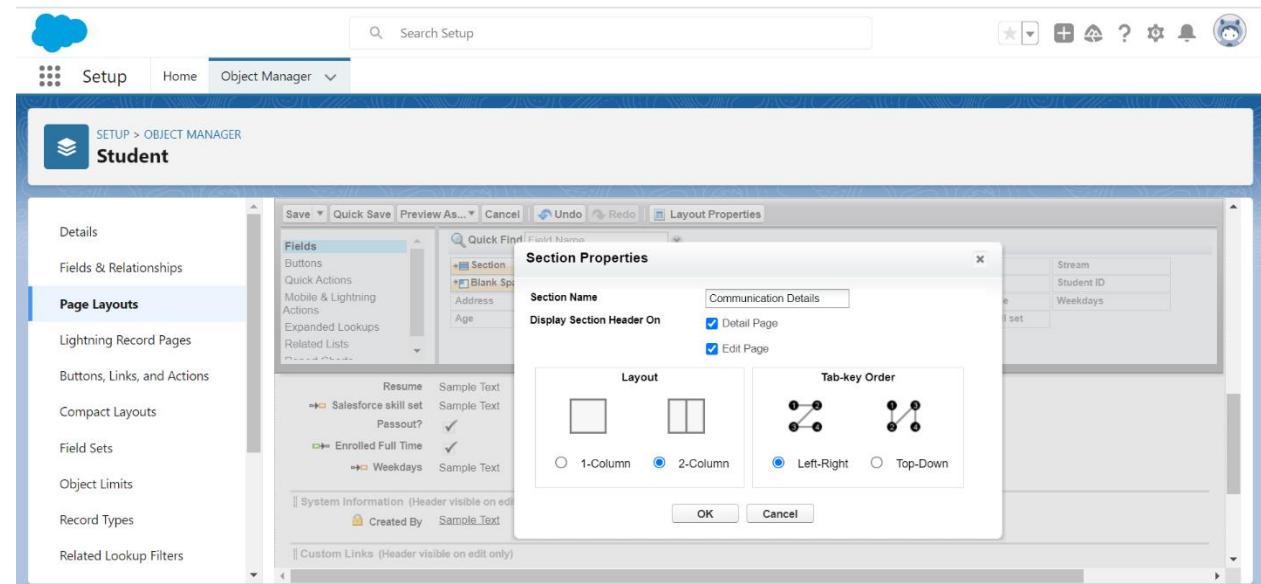
The screenshot shows the Salesforce setup interface for the 'Student' object, specifically editing the 'Fresher layout'. The left sidebar has the same sections as before. The main area shows the layout editor with a 'Fields' section on the left and a table of fields on the right. A red box highlights the 'Section' item in the 'Quick Find' dropdown menu above the table.

- To create sections for different field like information, communication, qualification, salesforce skills and professional information.

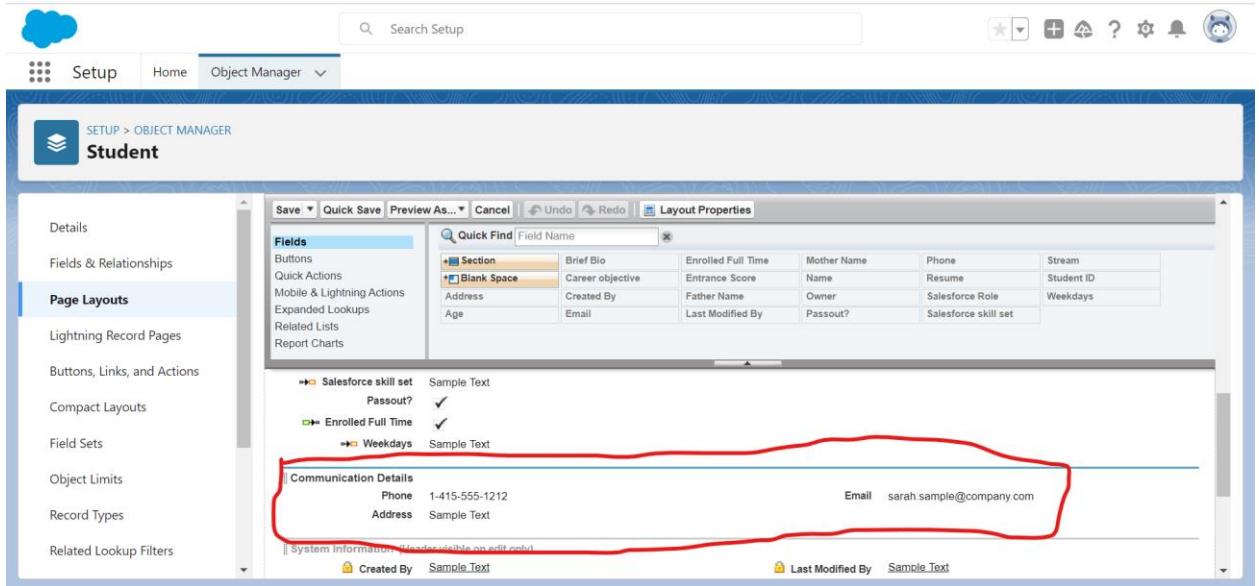
- First, we create information field which is already in layout, but we can edit by clicking on setting on right hand side. Information name can't be change and check details page and edit page of display section headers on. Layout should be 2 column and tab-key order top-down.



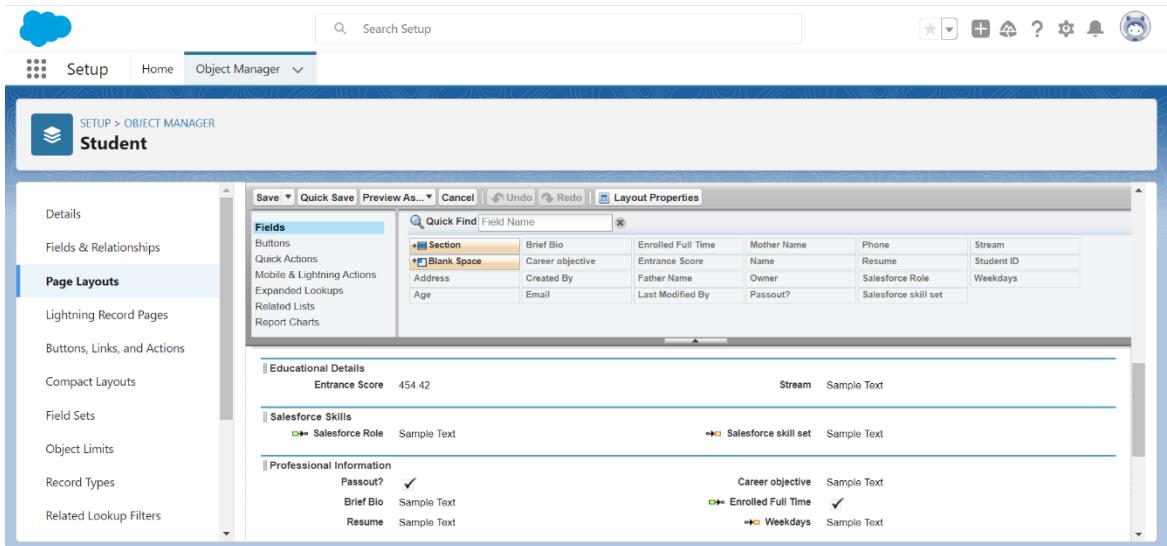
- Under information sections only name, age, father name, mother name consist.
- Secondly, create communication section, for this select sections and drag below information section.



- Communication section consist of Phone, email, address field. Select these filed from information filed and drag them into communication section.



- Similarly create education information, salesforce skills and professional information.
- Click on save.



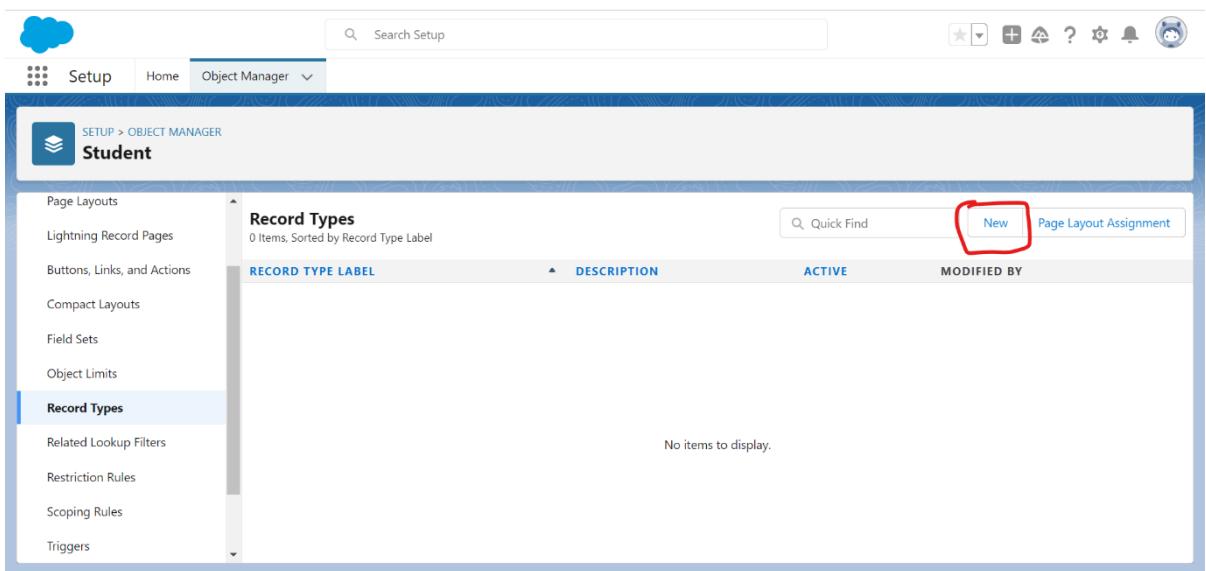
- now create same for experienced layout.

Record Type

After creating page layout so it needs to be control for front-end so we can do this using Record type. Record type is used to show the clone layout (ex. fresher & experienced layout) in front-end. we can also control picklist using record type.

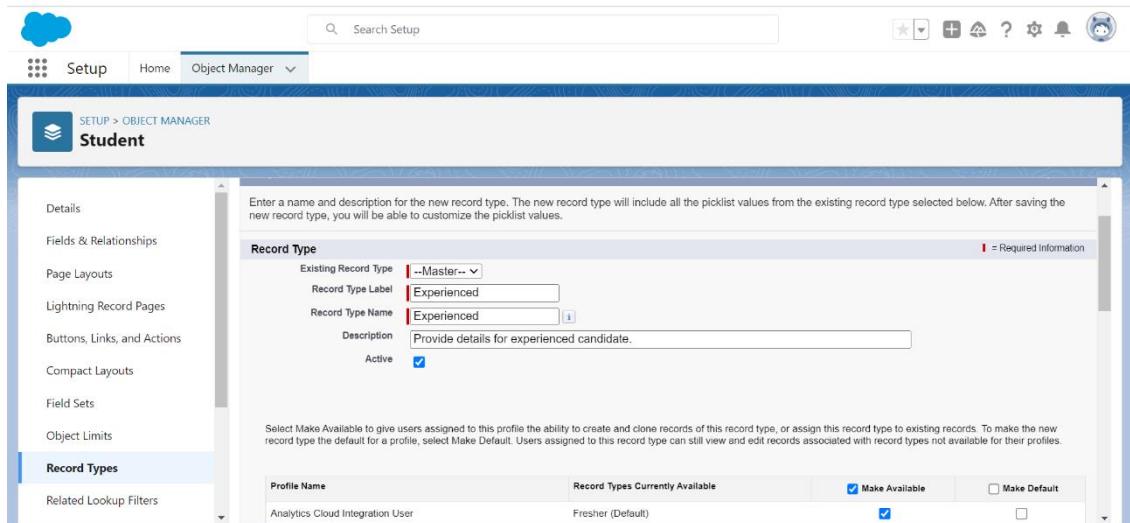
Steps to create record type.

- Go to object manager and search for student custom object.
- Now open the student object and click on **record type** from the left side menu.
- Now click on new.



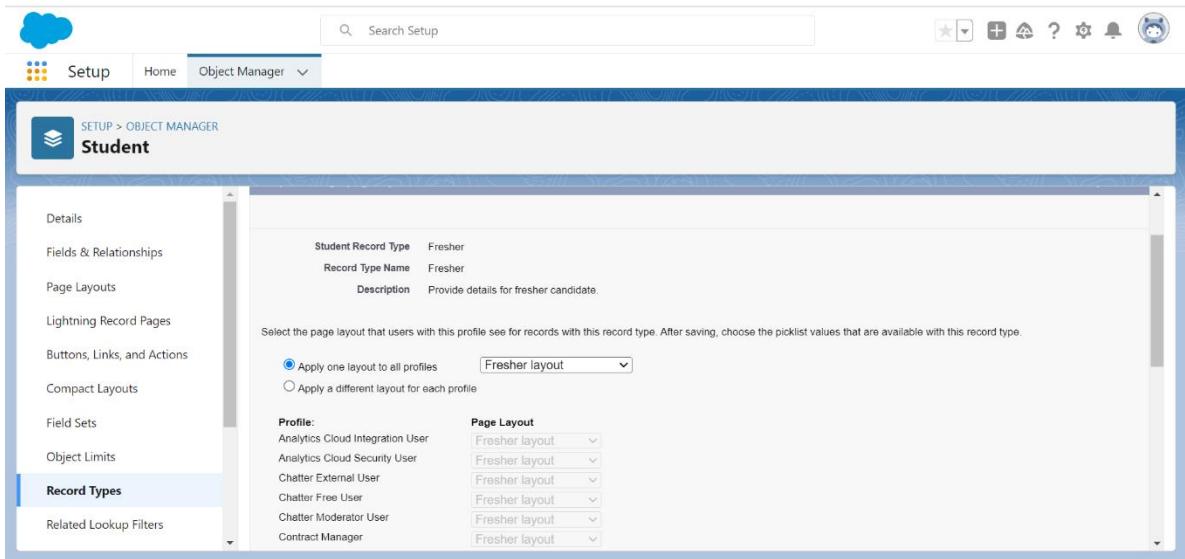
The screenshot shows the Salesforce Object Manager for the 'Student' object. On the left sidebar, 'Record Types' is selected. In the main area, the 'Record Types' list is displayed with the heading 'Record Types' and a note '0 Items, Sorted by Record Type Label'. A 'New' button is located at the top right of the list, and it is circled in red. Other buttons include 'Quick Find' and 'Page Layout Assignment'.

- Now enter details like record type level, record type name and description and check the box of make available for profile name.



The screenshot shows the 'Record Type' creation form for the 'Student' object. The 'Record Type' section includes fields for 'Existing Record Type' (set to '--Master--'), 'Record Type Label' (set to 'Experienced'), 'Record Type Name' (set to 'Experienced'), 'Description' (containing 'Provide details for experienced candidate.'), and 'Active' (checkbox checked). Below this, the 'Profile Name' section lists 'Analytics Cloud Integration User' with 'Record Types Currently Available' set to 'Fresher (Default)' and 'Make Available' checked. A note at the bottom states: 'Select Make Available to give users assigned to this profile the ability to create and clone records of this record type, or assign this record type to existing records. To make the new record type the default for a profile, select Make Default. Users assigned to this record type can still view and edit records associated with record types not available for their profiles.'

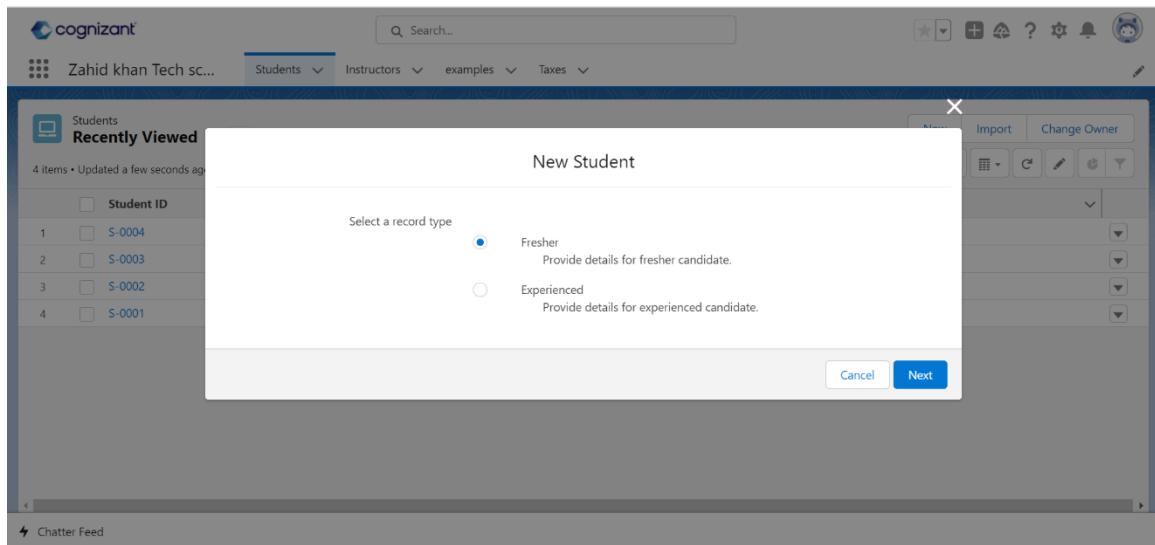
- Click on next.
- Now select the layout.



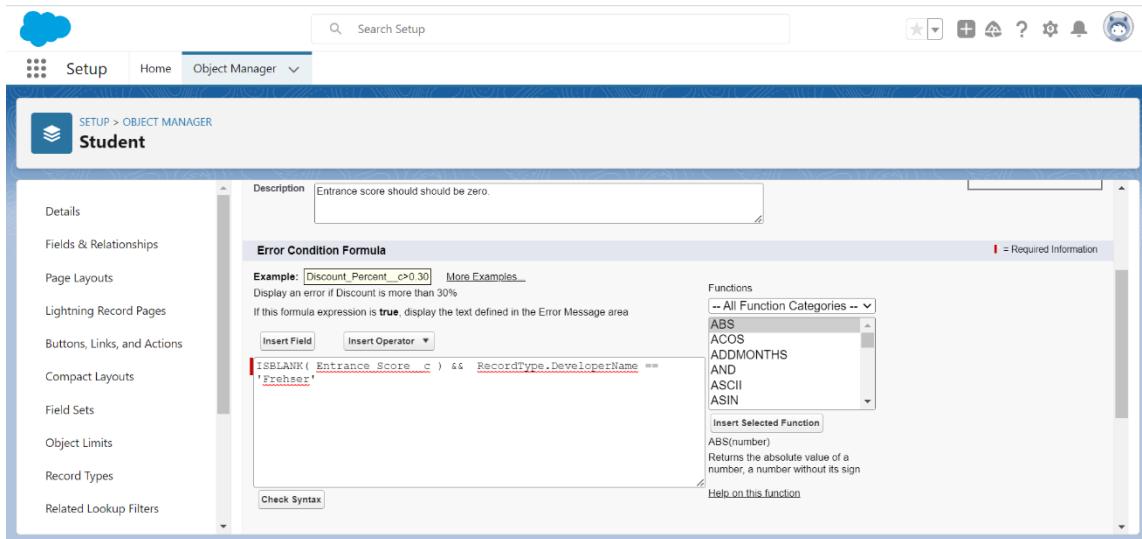
- Click on save.
- Now similar create record type for experience layout.
- Later, if we want to change the layout of record type just click on page layout assignment.

Record Types			
2 items, Sorted by Record Type Label			
RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Experienced	Provide details for experienced candidate.	✓	Zahid Khan, 31/08/2023, 4:52 pm
Fresher	Provide details for fresher candidate.	✓	Zahid Khan, 31/08/2023, 4:49 pm

- now go to front-end and check.



- When we enter the details for experienced it's through error of entrance score and that field is not present there, so we must do record type check in validation rule.
- Go to validation rule of student object and click on **Entrance_score_Not_Equal_to_zero**.
From validation rule and edit this as below.



- now using this formula only validation rule will apply on fresher layout not in experienced layout.

Steps to control picklist using record type.

- Go to **Record type** of student layout and click on fresher layout.
- click on edit against the salesforce role inside picklist available for edit.

The screenshot shows the Salesforce Setup interface with the Object Manager selected. Under the Student object, the Record Types section is active. A red box highlights the 'Picklists Available for Editing' table, which lists four fields: Salesforce Role, Salesforce skill set, Stream, and Weekdays. Each field has an 'Edit' link and a modified date of 31/08/2023, 4:49 pm.

Action	Field	Modified Date
Edit	Salesforce Role	31/08/2023, 4:49 pm
Edit	Salesforce skill set	31/08/2023, 4:49 pm
Edit	Stream	31/08/2023, 4:49 pm
Edit	Weekdays	31/08/2023, 4:49 pm

- Now select the value which is visible under student layout in front-end like admin, developer and quality analyst.

The screenshot shows the Salesforce Setup interface with the Object Manager selected. Under the Student object, the Record Types section is active. A red box highlights the 'Available Values' and 'Selected Values' dropdowns. The 'Available Values' dropdown contains Business Analyst, Architect, and Project manager. The 'Selected Values' dropdown contains Admin, Developer, and Quality Analyst. There are 'Add' and 'Remove' buttons between the two dropdowns.

- now click on save.
- Similar create picklist for experienced layout in same way.

Compact Layout

- Its show the selected field in record header.
- Red mark is the place for compact layout.
- Compact layout means selected field like information, communication details will show in red mark place which is shown in below figure.

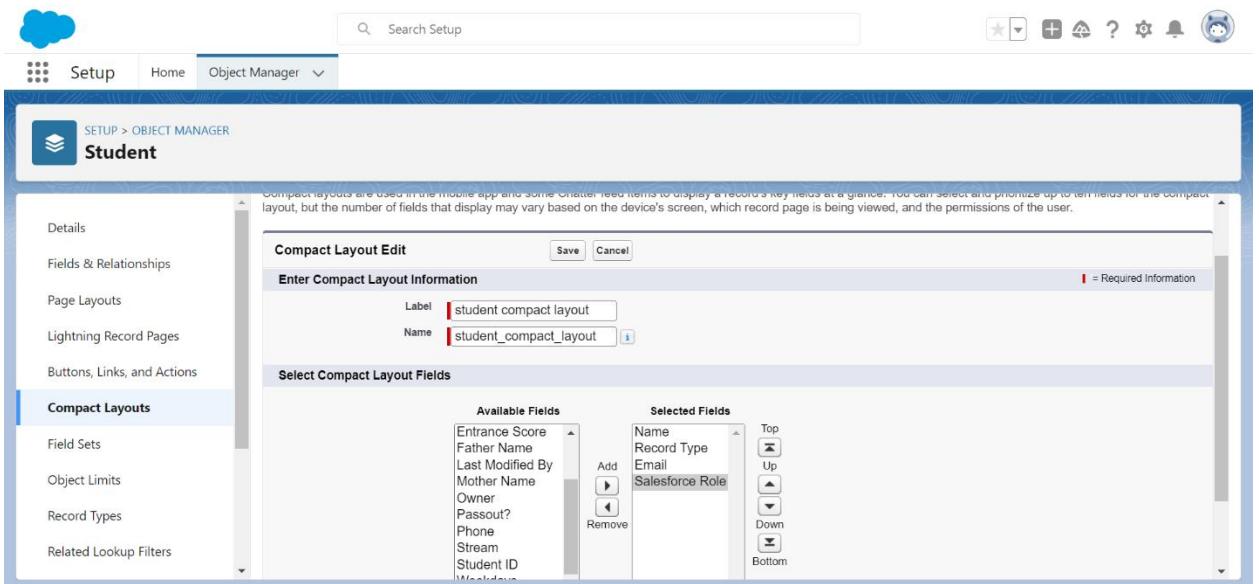
The screenshot shows a Salesforce page for a student record. At the top, there's a navigation bar with tabs for Students, Instructors, examples, and Taxes. Below the navigation is a search bar and a toolbar with various icons. The main area displays a student record for 'Student S-0006'. A red box highlights the header section where the student's name and ID are displayed. Below this, there are tabs for Related and Details, with the Details tab selected. Under the Details tab, there are sections for Information, Communication Details, Educational Details, Salesforce Skills, and Professional Information. The Professional Information section is expanded, showing fields for Passout? (with a checked checkbox), Career objective, Brief Bio (with a text input), and Enrolled Full Time (with a date input). A red box highlights the 'Professional Information' section. At the bottom right of the page, there are buttons for New Contact, Edit, and New Opportunity.

Steps to create compact layout.

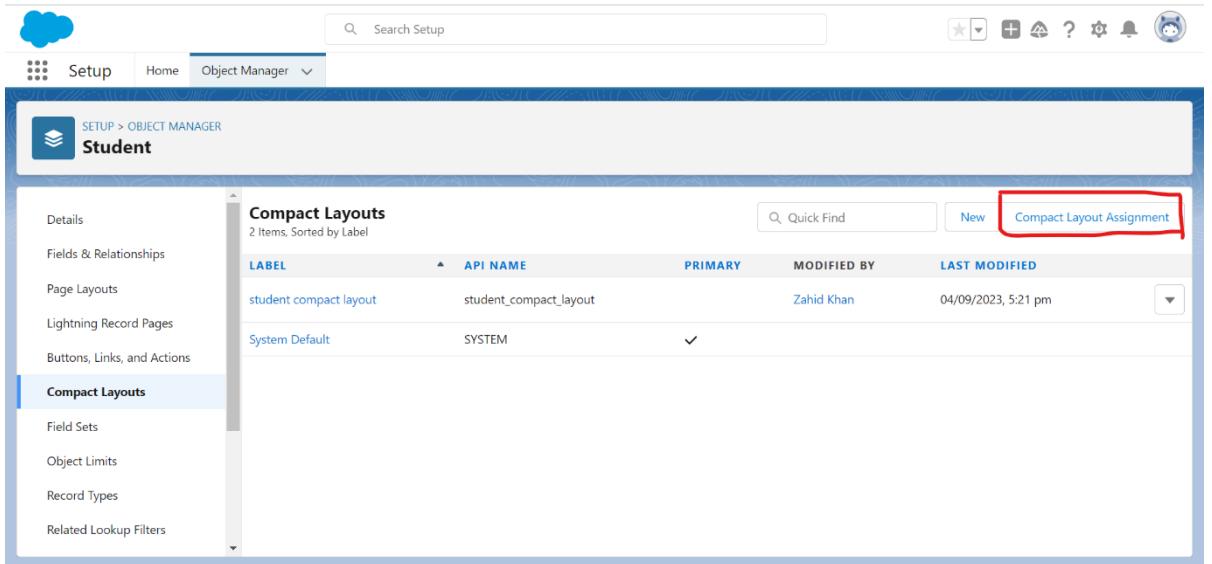
- Click on this icon and open step up.
- Click on **Object manager** and search for student object.
- Select compact layout option from left side menu.
- Then click on new.

The screenshot shows the Salesforce Setup page with the 'Object Manager' selected. On the left, there's a sidebar with links like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, and Compact Layouts. The 'Compact Layouts' link is highlighted with a red box. The main content area shows a table titled 'Compact Layouts' with one item: 'System Default' (API Name: SYSTEM). There are columns for LABEL, API NAME, PRIMARY, MODIFIED BY, and LAST MODIFIED. At the top right of the table, there are 'New' and 'Compact Layout Assignment' buttons, with the 'New' button highlighted by a red box. The status bar at the bottom indicates '1 Items, Sorted by Label'.

- Enter label name under **compact layout information**.
- Now select the field from **compact field layout**.

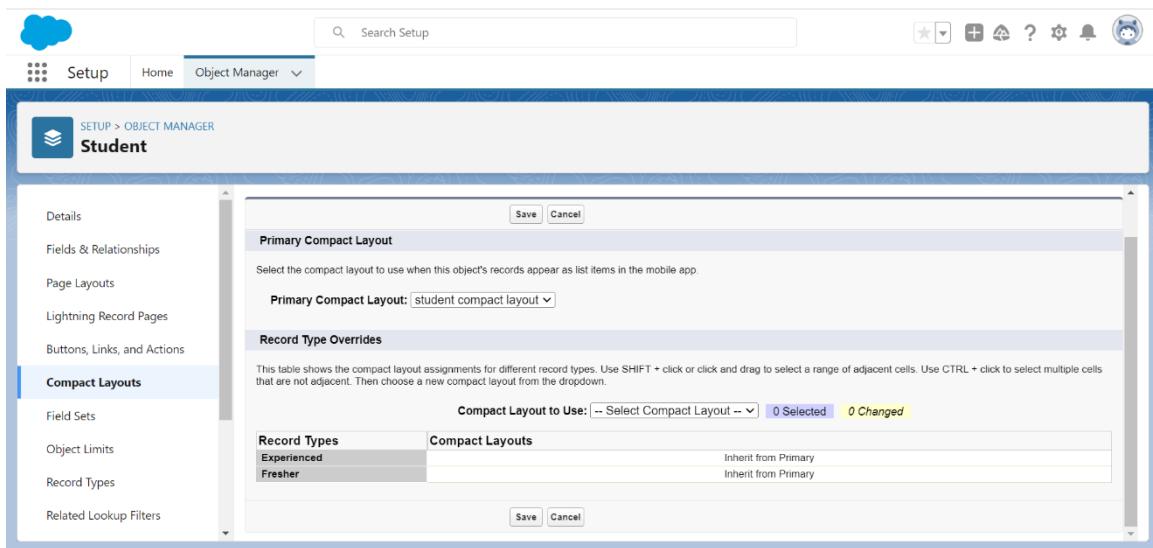


- Now click on save.
- Now again click on compact layout from left side menu.
- click on **compact layout assignment**.

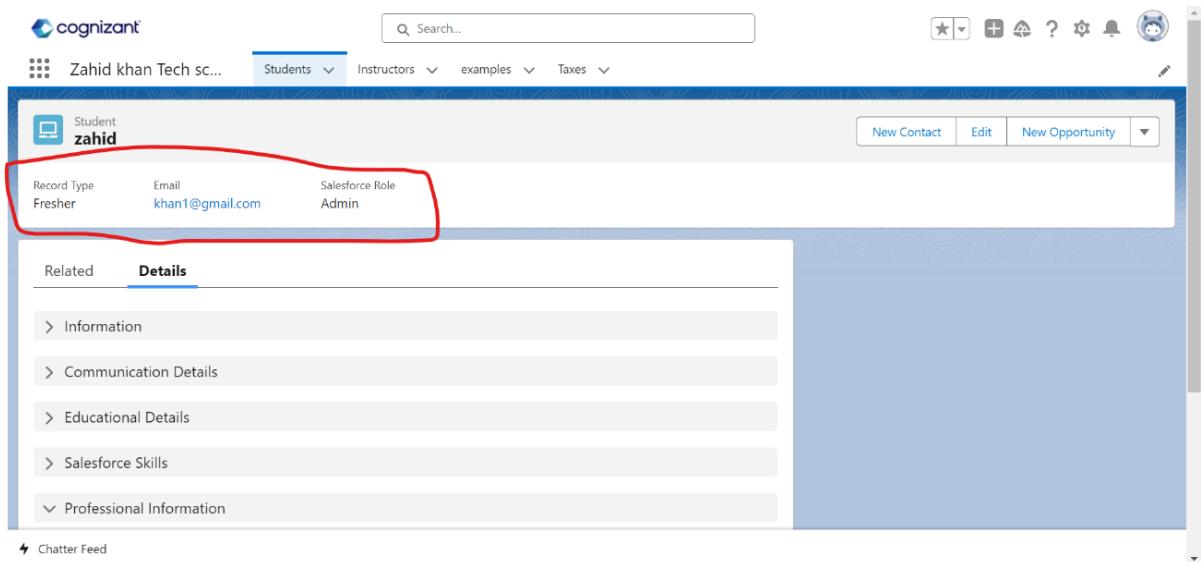


- Select **primary compact layout** is **student compact layout** which is created.
- Now select **compact layout to use** is **student compact layout**.

- Now click on save.



- Go to front-end and check.



Day-05

Relationship

Relationships are two types.

- One to many.
- Many to many.
- Whenever create look-up or master-details we need to create on child object, and they will look up to parent. (For ex: - students(child) and class (parent)).

One to many relationships.

- Lookup relationships.
- Master-details relationships.

Lookup relationships

- creates relationships that links one object to another object.
- The relationship field allows users to click on a lookup icon to select a value from the popup list.
- The other is the source of the values in the list.
- Loosely coupled relationships. Its mean if we delete parent record so the dependent or child records can't be deleted.
- Look-up relationship always created from child and need to select parent object.

Steps to create lookup relationship.

- Click on  this icon and open step up.
- Create a custom object named as Class(parent).
- Type record name as class and data type will be text **Enter Record Name Label and Format.**
- Click on save to create new object.
- Now create 2 field [course duration(text), fees(currency)].
- Now go to student object (child).
- Click on field and relationship.
- Click on new.

- Select look-up from the data type.

The screenshot shows the Salesforce setup interface for creating a new field. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Student'. On the left, a sidebar lists options like 'Details', 'Fields & Relationships', 'Page Layouts', etc. The 'Fields & Relationships' section is selected. In the main content area, a 'Data Type' section is displayed with several options: 'None Selected' (selected), 'Auto Number', 'Formula', 'Roll-Up Summary', 'Lookup Relationship' (highlighted with a red box), 'Master-Detail Relationship', and 'External Lookup Relationship'. Each option has a detailed description below it.

- Click on next.
- Now select **class object (parent)** in related to field.

The screenshot shows the 'New Relationship' wizard step 2. The title is 'Student New Relationship'. The sidebar on the left is identical to the previous screenshot. The main area is titled 'Step 2. Choose the related object'. It contains a sub-instruction 'Select the other object to which this object is related.' Below this is a 'Related To' input field containing the word 'Class'. At the bottom right, there are 'Previous', 'Next', and 'Cancel' buttons.

- Click on next.

Student
New Relationship

Step 3. Enter the label and name for the lookup field Step 3 of 6

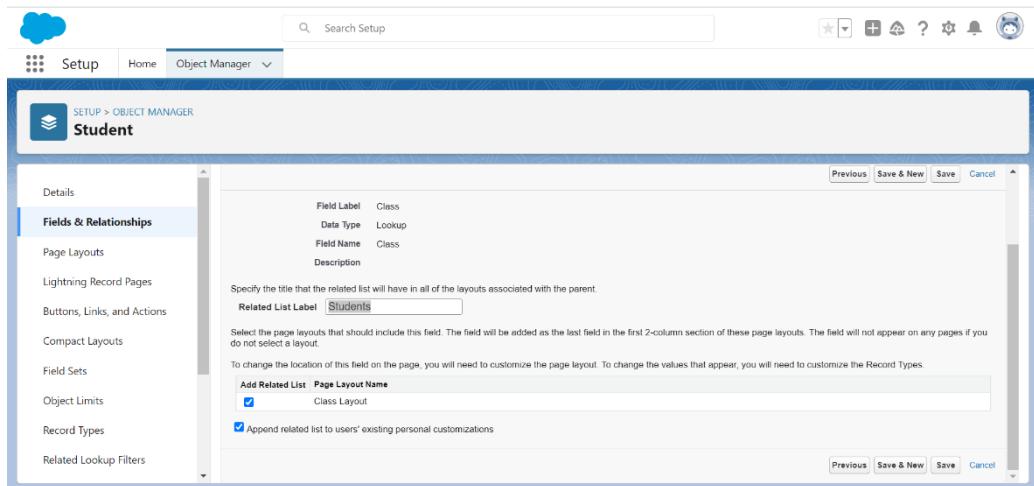
Field Label	<input type="text" value="Class"/>	Previous	Next	Cancel
Field Name	<input type="text" value="Class"/>			
Description	<input type="text"/>			
Help Text	<input type="text"/>			
Child Relationship Name	<input type="text" value="Students"/>			
Required	<input type="checkbox"/> Always require a value in this field in order to save a record			
What to do if the lookup record is deleted?	<input checked="" type="radio"/> Clear the value of this field. You can't choose this option if you make this field required.			
	<input type="radio"/> Don't allow deletion of the lookup record that's part of a lookup relationship.			
Auto add to custom report type	<input checked="" type="checkbox"/> Add this field to existing custom report types that contain this entity			
Lookup Filter				
Optionally, create a filter to limit the records available to users in the lookup field. Tell me more!				
Show Filter Settings				
Previous	Next	Cancel		

- Click on next.
- Checkbox to visible all and click on next.
- Add page-layout name for all three layouts.

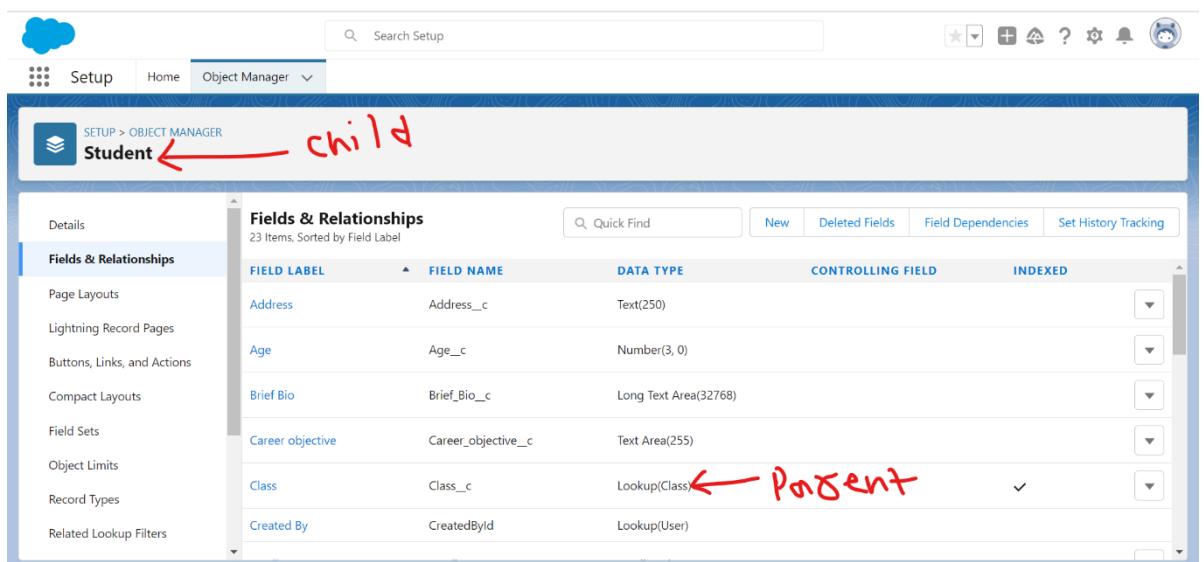
The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Search bar with "Search Setup", a gear icon, and other navigation icons.
- Breadcrumbs:** SETUP > OBJECT MANAGER > Student
- Left Sidebar:** A sidebar with various options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters.
- Current Screen:** Step 5. Add reference field to Page Layouts (Step 5 of 6).
- Form Data:**
 - Field Label: Class
 - Data Type: Lookup
 - Field Name: Class
 - Description: (empty)
- Instructions:** Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.
- Selectable Layouts:**
 - Add Field: Page Layout Name
 - Experienced Layout
 - Fresher layout
 - Student Layout
- Buttons:** Previous, Next, Cancel.

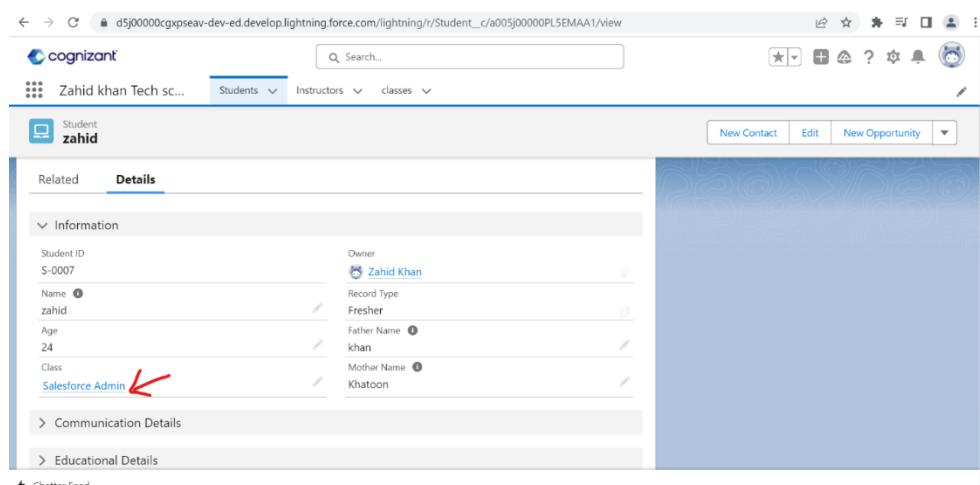
- Click on next.
- On next page you will get option to change related list label.



- click on save.



- Now go to front-end and check the result.



Master-Detail Relationship

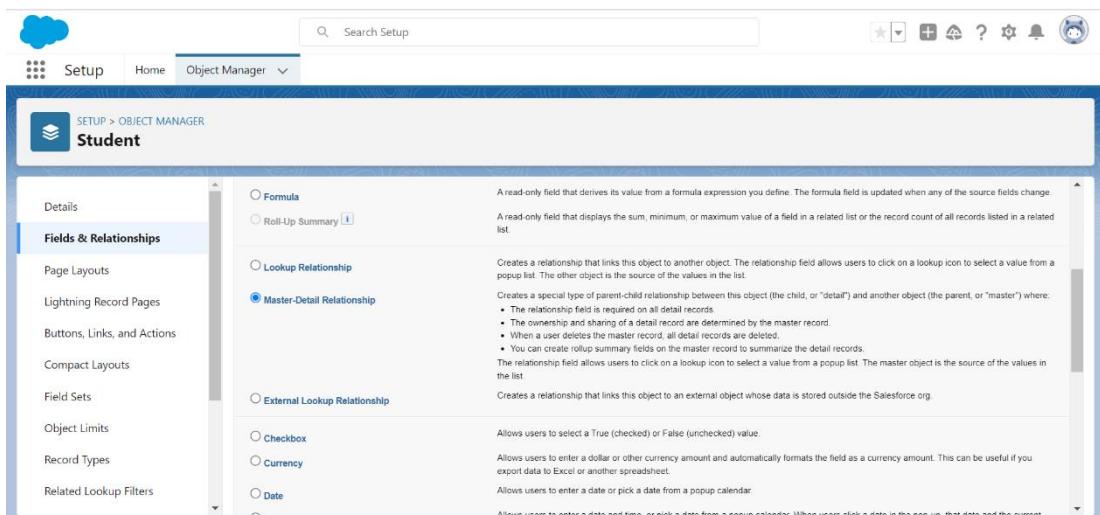
Creates a special type of parent-child relationship between two objects.

One is known as child/detail where we create master-detail relationships field, and another is known as parent/master.

- Required on all detail records.
- Ownership and sharing of a detail record is determined by the master record.
- If user deletes the master record then all detail records are deleted.
- One can create rollup summary fields on the master records so that detail records can be summarized.
- Tightly coupled relationship.

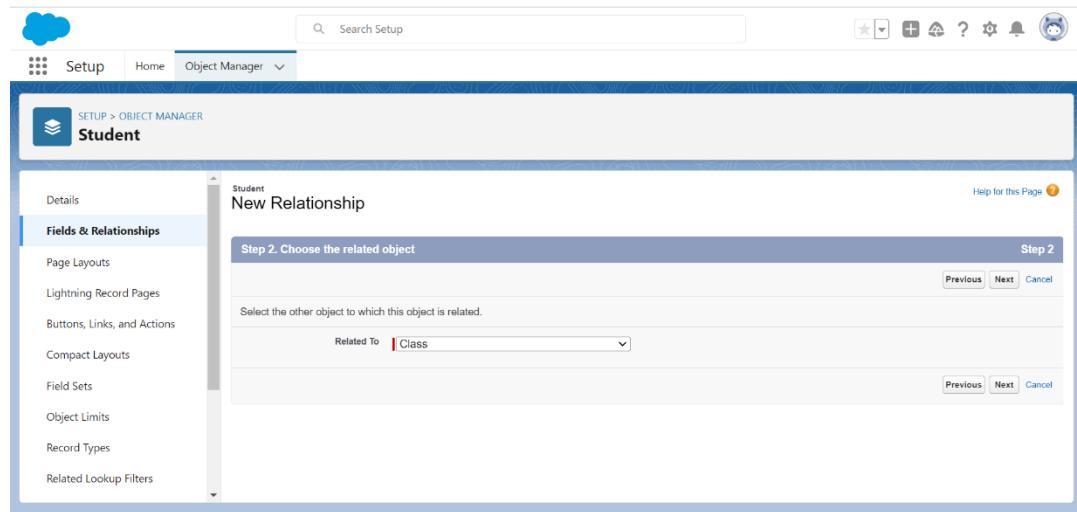
Steps to convert the look-up relationship to master-detail relationship.

- Go to student object and click on new to create new field.
- Now choose master-detail from data type.

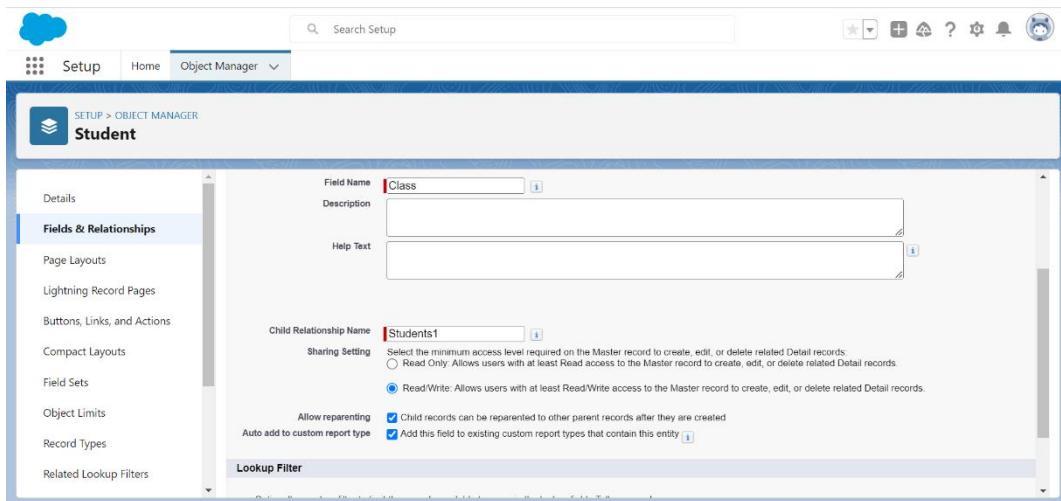


- Click on next.

- Now choose parent object (i.e., class object) in **Related field** to.



- click on next.
- If we want to change anything from child record so we must select allow repeating option.



- Click on next.
- Visible checkbox already checked because it's a required field.
- Click on next.
- Now when we click save button, we got error that we cannot create master-detail relationship because some records already created in students object(child).

To solve this error, we have 2 options.

- Do not create any records in child object (students object) before creating master-detail relationship.

You must first create a Lookup relationship, populate the lookup field with

The screenshot shows the Salesforce Setup interface under Object Manager for a custom object named "Student". The "Fields & Relationships" tab is selected. A prominent error message reads: "Cannot Create Master-Detail Relationship" followed by the sub-instruction: "You cannot create a new Master-Detail relationship on an existing custom object if records already exist. You must first create a Lookup relationship, populate the lookup field with data in all records, and then change the relationship type to Master-Detail." Below the message, there is a link to "Click here to return to the previous page."

- To convert the lookup relationship into master-detail relationship. All records have parent object.

The screenshot shows a custom Salesforce instance with a "Students" object. The list view displays columns for Student ID, Name, Age, Email, and Class. The "Class" column is highlighted with a red box. A callout box with the text "Parent field." points to the "Class" column. The list view shows 5 items, all belonging to the "Salesforce developer" class.

	Student ID ↑	Name	Age	Email	Class
1	S-0001	zahid khan		zahid.khan@gmail.com	Salesforce developer
2	S-0002	shahid		shahid@gmail.com	Salesforce developer
3	S-0003	wahid		wahid@gmail.com	Salesforce developer
4	S-0004	zahid khan	10	khan@gmail.com	Salesforce developer
5	S-0007	zahid	24		Salesforce Admin

- now go to student object and edit the class field which is created previously.

SETUP > OBJECT MANAGER
Student

Fields & Relationships
23 Items, Sorted by Field Label

Age	Age_c	Number(3, 0)
Brief Bio	Brief_Bio_c	Long Text Area(32768)
Career objective	Career_objective_c	Text Area(255)
Class	Class_c	Lookup(Class)
Created By	CreatedBy	Lookup(User)
Email	Email_c	Email (Unique)
Enrolled Full Time	Enrolled_Full_Time_c	Checkbox

- Now click on change field type.

SETUP > OBJECT MANAGER
Student

Edit Student Custom Field
Class

Custom Field Definition Edit Change Field Type Save Cancel

Field Information

Field Label	Class	Data Type	Lookup
Field Name	Class		
Description			
Help Text			
Data Owner	User		
Field Usage	--None--		
Data Sensitivity Level	--None--		
Compliance Categorization	Available	Chosen	

- Now change the field type from look-up relationship to master-detail relationship.

SETUP
Object Manager

Step 1. Choose the field type

Step 1

Next | Cancel

Data Type

None Selected Select one of the data types below.

Lookup Relationship Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.

Master-Detail Relationship Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:

- The relationship field is required on all detail records.
- The ownership and sharing of a detail record are determined by the master record.
- When a user deletes the master record, all detail records are deleted.
- You can create rollup summary fields on the master record to summarize the detail records.

 The relationship field allows users to click on a lookup icon to select a value from a popup list. The master object is the source of the values in the list.

- Now click on next and click on save.
- Now look-up relationship is now changed with master-detail relationship.

SETUP > OBJECT MANAGER

Student

Fields & Relationships		
22 Items, Sorted by Field Label		
	FIELD NAME	DATA TYPE
Address	Address_c	Text(250)
Age	Age_c	Number(3, 0)
Brief Bio	Brief_Bio_c	Long Text Area(32768)
Career objective	Career_objective_c	Text Area(255)
Class	Class_c	Master-Detail(Class)
Created By	CreatedById	Lookup(User)
Email	Email_c	Email (Unique)

- Now go to students records in front-end and check.

Steps to convert the master-detail relationship to look-up relationship.

- To convert the master-detail relationship to look-up relationship firstly, delete the all Roll-up summary field from Class object(parent) and also erased the roll-up summary field from delete filed session.

SETUP > OBJECT MANAGER

Class

Fields & Relationships				
6 Items, Sorted by Field Label				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Class Name	Name	Text(80)		✓
Course Duration	Course_Duration_c	Text(100)		✗
Created By	CreatedById	Lookup(User)		✗
Fees	Fees_c	Currency(10, 2)		✗
Last Modified By	LastModifiedById	Lookup(User)		✗
Owner	OwnerId	Lookup(User,Group)		✓

- To delete permanently click on erased.

Class Deleted Fields

Salesforce starts to hard delete these fields after 15 days. The hard-delete process permanently removes all metadata and system data for the field. Erase skips the 15-day grace period, and immediately sets the field to ready for the hard-delete process. Undelete (if available) restores the field.

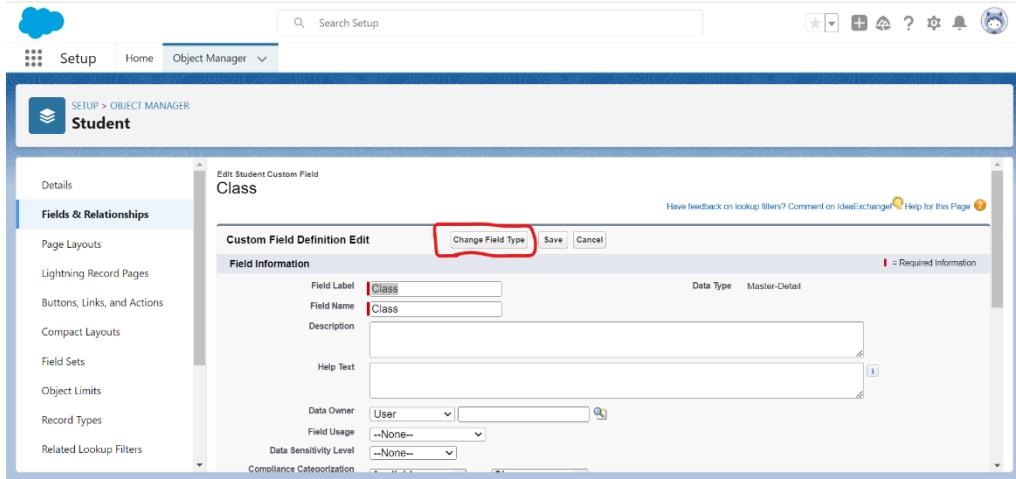
Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Erase Undelete	Total students	Total_students_del__c	Roll-Up Summary (COUNT Student)			Zahid Khan, 06/09/2023, 10:42 pm

- Now go to child object (student object).
- Click on field & relationship. Click on drop down in parent object (class object) and click on edit.

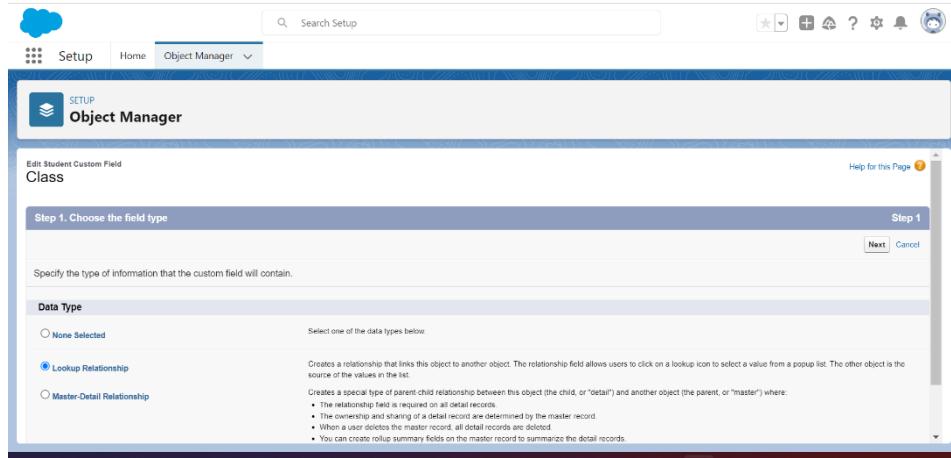
Fields & Relationships
22 Items, Sorted by Field Label

	Field Label	API Name	Type	
Career objective	Career_objective__c	Text Area(255)		
Class	Class__c	Master-Detail(Class)	▼	▼
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email (Unique)	▼	
Enrolled Full Time	Enrolled_Full_Time__c	Checkbox		
Entrance Score	Entrance_Score__c	Number(5, 2)		
Father Name	Father_Name__c	Text(100)		

- now click on change field type.



- Now select look-up relationship and click next and save.



- Now master-detail is converted into look-up relationship.

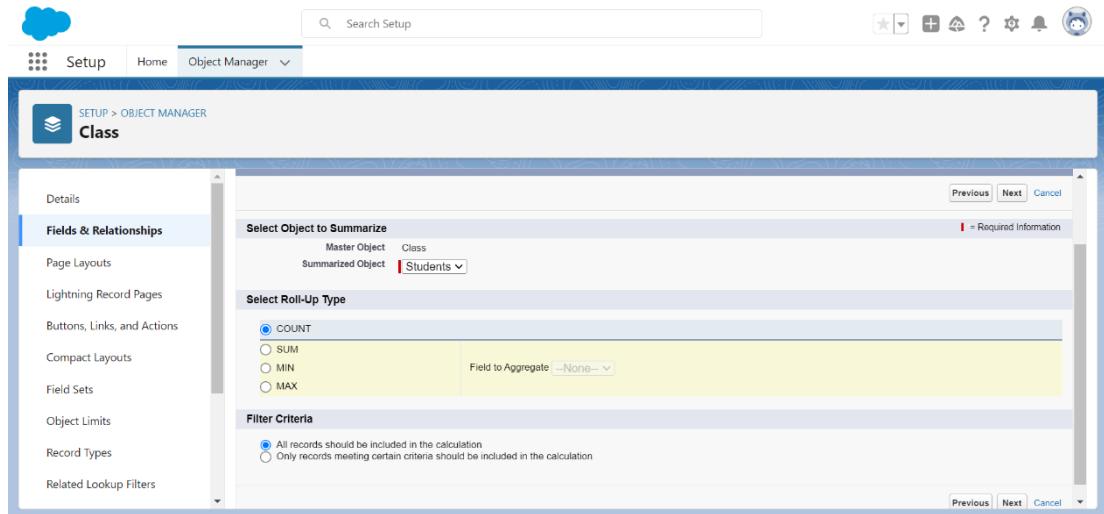
Rollup Summary Field

- A read only field that displays the sum, minimum or maximum value of a field in a related list.
- This field also can count all available records available in related list.
- Rollup summary fields always created on parent object.

Steps to create Rollup summary.

- Go to object manager from set-up and search for class object (parent object).
- Now click on fields & relationships.
- Click on new.
- Select Rollup summary field from the data type.
- Click on next.
- Enter Field label and field name.
- Click on next.
- Now select student from the select object to summarize to object.

- Now select Roll-up type. Here we are selecting total of the student object records, so we choose **count**. if we need to calculate highest of any field so we choose **max**, for lowest of any field choose **min** and for any field add all field select **sum**.



- we also apply field criteria. But for now, we are choosing **All records should be included**.
- Click on next.
- Checkbox visible box.
- Click on next. Click on save.
- Now roll-up summary field is created.

Fields & Relationships					
FIELD LABEL		FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Class Name	Name		Text(80)		✓
Course Duration	Course_Duration_c		Text(100)		▼
Created By	CreatedBy		Lookup(User)		▼
Fees	Fees_c		Currency(10, 2)		▼
Last Modified By	LastModifiedBy		Lookup(User)		▼
Owner	OwnerId		Lookup(User,Group)		✓
Total students	Total_students_c		Roll-Up Summary (COUNT Student)		▼

- Now go to front-end and check class object(parent).

The screenshot shows a Salesforce Class object detail page. The 'Fees' field is set to ₹10,000.00 and the 'Total students' field is set to 4. A red box highlights the 'Total students' value, and an arrow points from it to a callout box labeled 'Roll-up summary'.

Many to many relationships.

- It can be accomplished by junction object.

Steps to create many to many relationships.

- To create many to many relationships firstly, we need to create a junction object (i.e., stdClass object).
- Now go to field & relationship of stdClass object.
- Now we have to create two master-details relationship one for student object and 2nd for class object.

The screenshot shows the Salesforce Setup > Object Manager interface for creating a new custom object named StdClass. The 'Fields & Relationships' tab is selected. Under 'Data Type', 'Master-Detail Relationship' is selected, and a detailed description is provided:

Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.

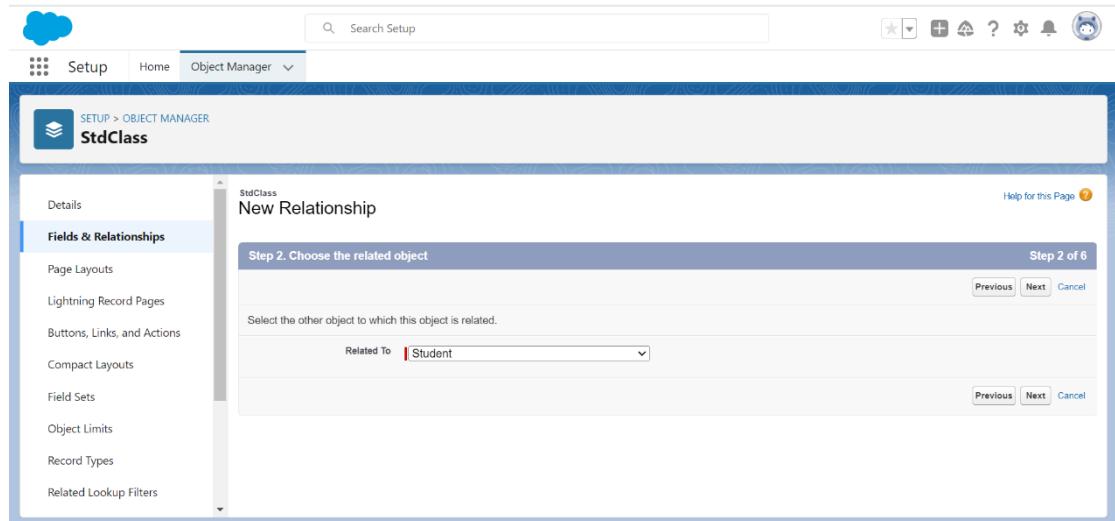
Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:

- The relationship field is required on all detail records.
- The ownership and sharing of a detail record are determined by the master record.
- When a user deletes the master record, all detail records are deleted.
- You can create rollup summary fields on the master record to summarize the detail records.

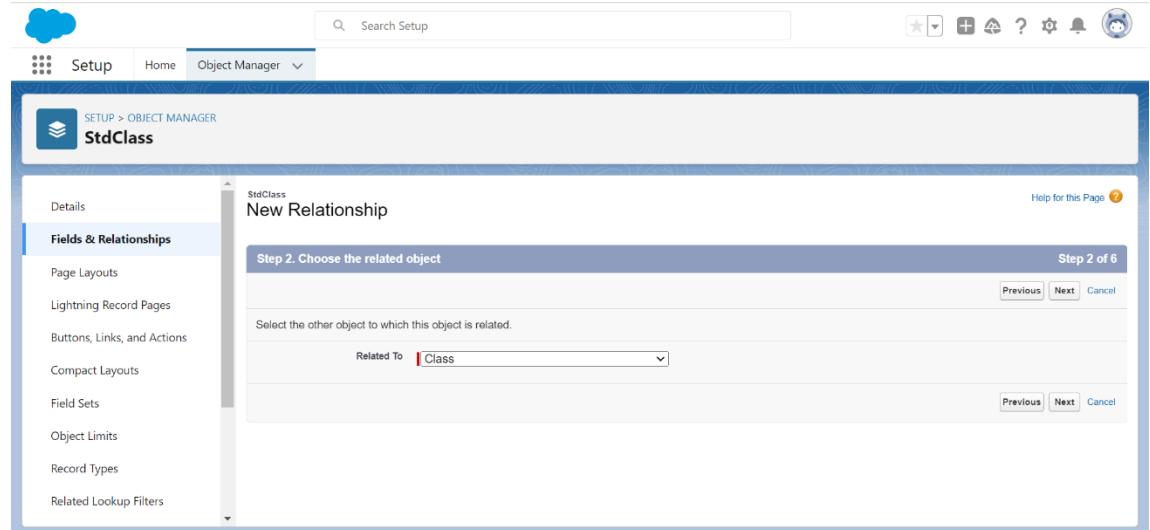
The relationship field allows users to click on a lookup icon to select a value from a popup list. The master object is the source of the values in the list.

- Click on next.

- Select student in **Related to** drop down.



- Now click next, next and save.
- Secondly, create one more master-detail relationship for class. for this select class in **related to** drop down.



- Click on next and enter field label and field name next and save.
- Now go to front-end and check result.

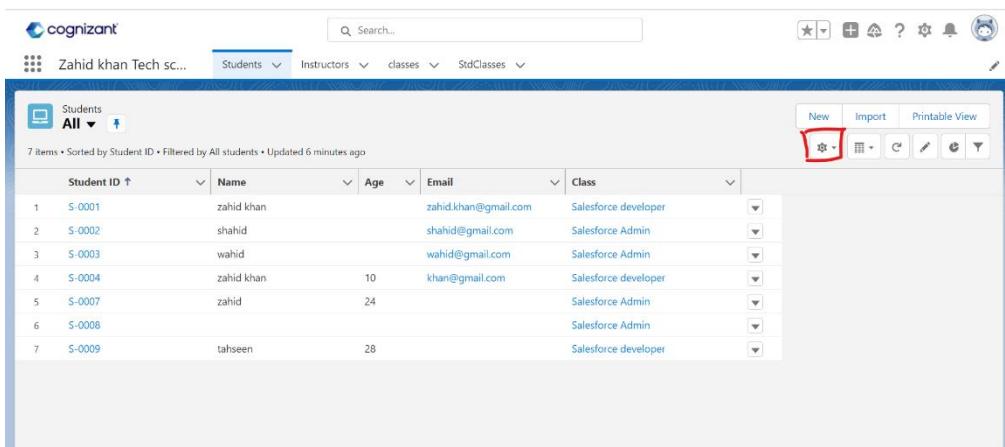
Day-06

List view

- List view shows list of records available under object.
- We can choose which field to display in the list view.
- We can also apply sharing and filters on list view.

Steps to create list view.

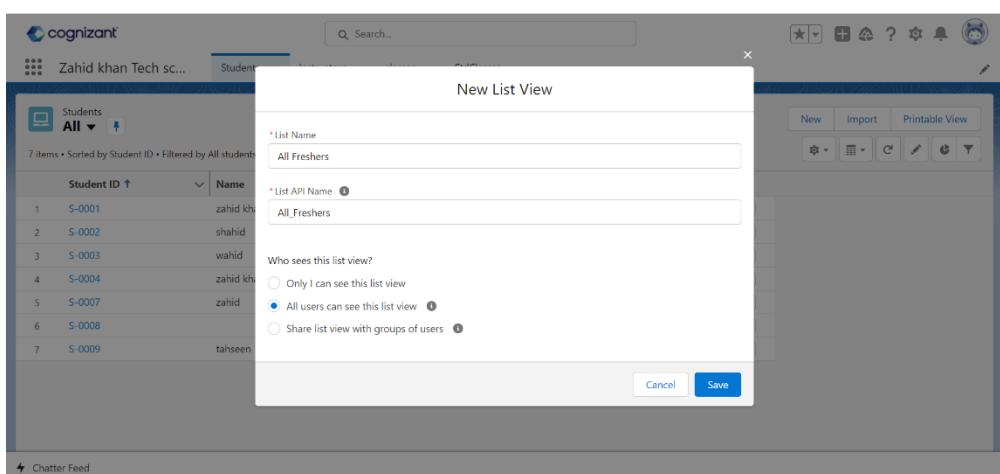
- Firstly, go the front-end of your app.
- Now select the object in which you want to make list view. (We are taking student object for now.)
- Now click on student object tab in front-end.
- Now click  on this icon which is available on right hand side.



The screenshot shows the Salesforce interface for the 'Students' object. The top navigation bar includes the cognizant logo, a search bar, and various navigation links like 'Students', 'Instructors', 'classes', and 'StdClasses'. Below the navigation is a toolbar with icons for 'New', 'Import', and 'Printable View'. A prominent settings icon (a gear with a dropdown arrow) is highlighted with a red box. The main area displays a table of student records with columns: Student ID, Name, Age, Email, and Class. The table has 7 items, sorted by Student ID, and was updated 6 minutes ago. The student data is as follows:

Student ID	Name	Age	Email	Class
1 S-0001	zahid khan		zahid.khan@gmail.com	Salesforce developer
2 S-0002	shahid		shahid@gmail.com	Salesforce Admin
3 S-0003	wahid		wahid@gmail.com	Salesforce Admin
4 S-0004	zahid khan	10	khan@gmail.com	Salesforce developer
5 S-0007	zahid	24		Salesforce Admin
6 S-0008				Salesforce Admin
7 S-0009	tahseen	28		Salesforce developer

- Now select **New** from drop down menu from settings icon.
- Now enter List name (i.e., Freshers). List API name automatic populated.
- Select **all users can see this list view**.



The screenshot shows the 'New List View' dialog box. It has fields for 'List Name' (set to 'All Freshers') and 'List API Name' (set to 'All_Freshers'). Below these, there's a section titled 'Who sees this list view?' with three options: 'Only I can see this list view' (unchecked), 'All users can see this list view' (checked), and 'Share list view with groups of users' (unchecked). At the bottom are 'Cancel' and 'Save' buttons.

- click on save.
- Now new list is created (i.e., All Freshers).

The screenshot shows a list view titled 'All Freshers'. The list contains 7 items, each with a student ID: S-0003, S-0004, S-0007, S-0008, S-0009, S-0001, and S-0002. The item 'S-0003' is highlighted with a red box. On the right side, there is a 'Filters' sidebar with a single filter applied: 'Filter by Owner My students'.

- Now we have to add filter so that when we click on All freshers list its shows only fresher application.
- For this click on Add filter which is available on right side.

The screenshot shows the same 'All Freshers' list view as before, but now it only displays the 7 student IDs: S-0003, S-0004, S-0007, S-0008, S-0009, S-0001, and S-0002. This is because a filter has been applied. The 'Add Filter' button in the filters sidebar is highlighted with a red box.

- Now enter field in which filter will apply for this example we select record type because we want to show all application of freshers. Operator is **Equal** and value is fresher.

The screenshot shows a Salesforce page for the 'Students' object. The title bar includes the cognizant logo, a search bar, and navigation links for Students, Instructors, classes, and StdClasses. The main content area displays a list titled 'All Freshers' with 7 items, sorted by Student ID. To the right of the list is a 'Filters' sidebar. A red box highlights the 'New Filter*' section, which contains fields for 'Field' (set to 'Record Type'), 'Operator' (set to 'equals'), and 'Value' (set to '1 option selected'). Below this is a 'Done' button.

- Now click on done.
- Click on save.

This screenshot shows the same Salesforce page after the filter has been saved. The 'Save' button in the top right corner of the sidebar is highlighted with a red circle. The filter configuration in the sidebar now shows the saved condition: 'Record Type* equals Fresher'. The main list view remains the same, displaying 7 items for 'All Freshers'.

- Now when we click on fresher its show only fresher application.
- Similarly create list view for experienced application.

The screenshot shows a CRM application interface for 'Zahid khan Tech sc...'. The top navigation bar includes 'Students', 'Instructors', 'classes', and 'StdClasses'. A search bar is at the top right. On the left, there's a sidebar with a 'Students' icon and a dropdown menu showing 'All Experineced' (with a red box around it) and 'All Freshers'. The main area displays a table with two rows:

	Record Type
developer	Experienced
developer	Experienced

At the bottom right of the main area, there are several icons: New, Import, Printable View, and others.

- if you want to open that filter option then click on cup icon at right corner.

This screenshot shows the same CRM interface after clicking the filter icon (cup icon) from the previous screenshot. The 'Filters' panel on the right is now open, displaying the applied filter: 'Record Type equals Experienced'. It also shows options to 'Add Filter' and 'Remove All'.

The main table area now shows two student records:

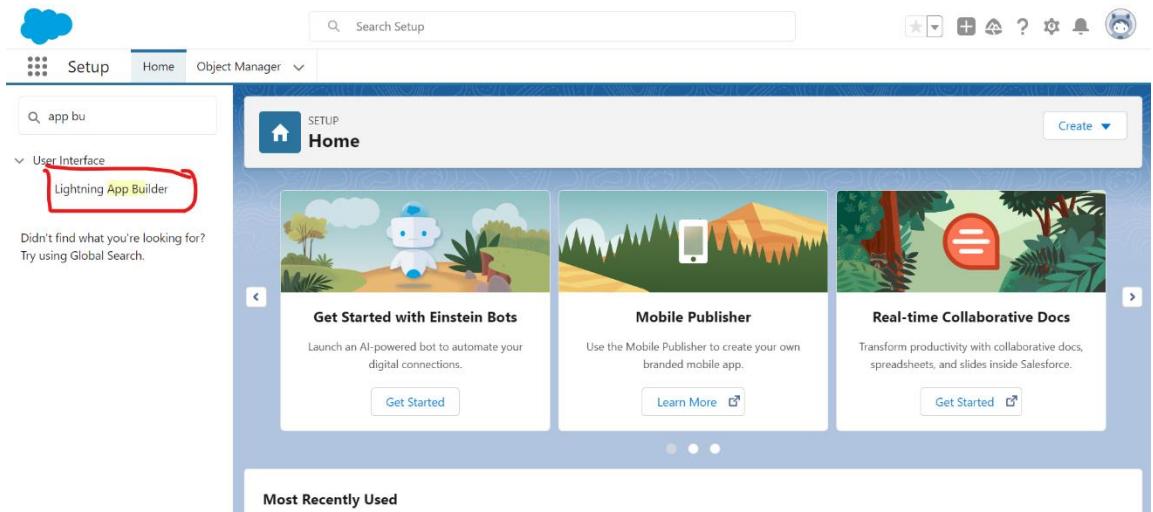
	Student ID	Name	Class	Record Type
1	S-0012	Shahid	Salesforce developer	Experienced
2	S-0013	Rahul	Salesforce developer	Experienced

Lighting pages

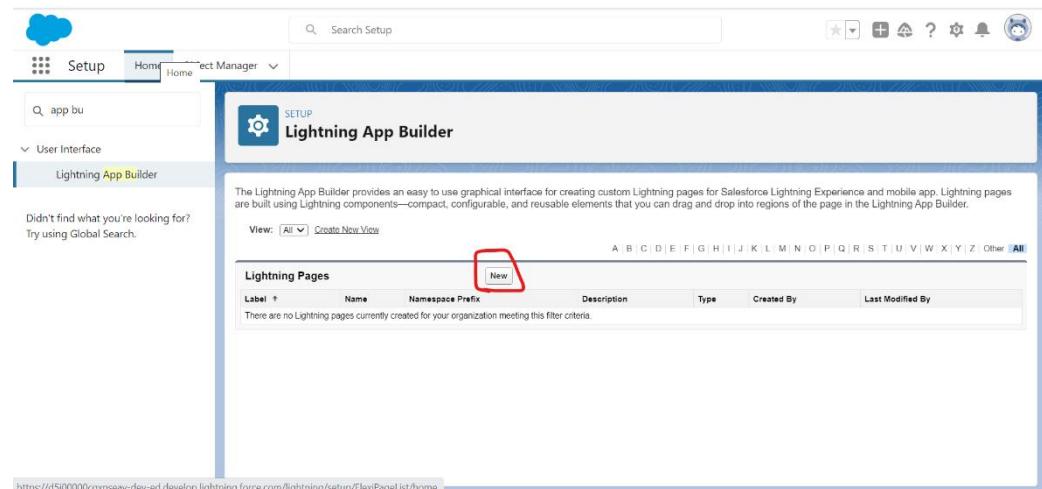
- Lighting pages is used to create home page, record page and app pages for any app.

Step to create home page.

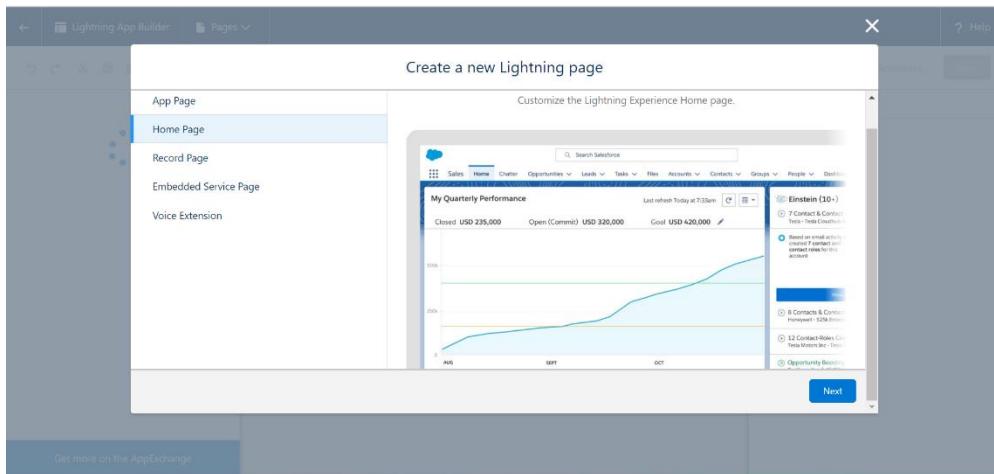
- Firstly, go to setup.
- At home, enter **Lightning App builder** in quick search.



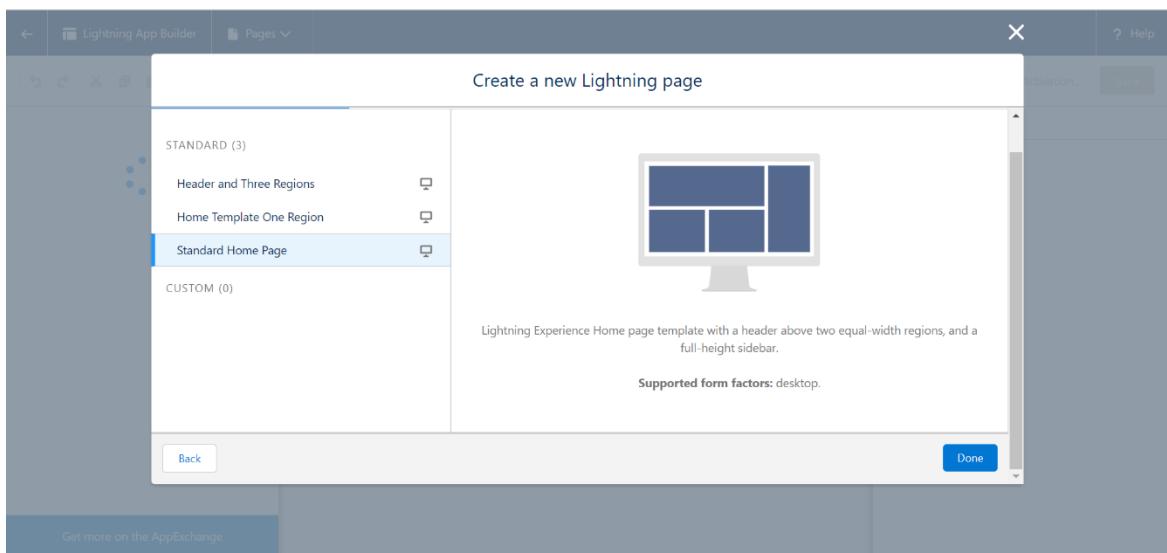
- click on that.
- After that click on new.



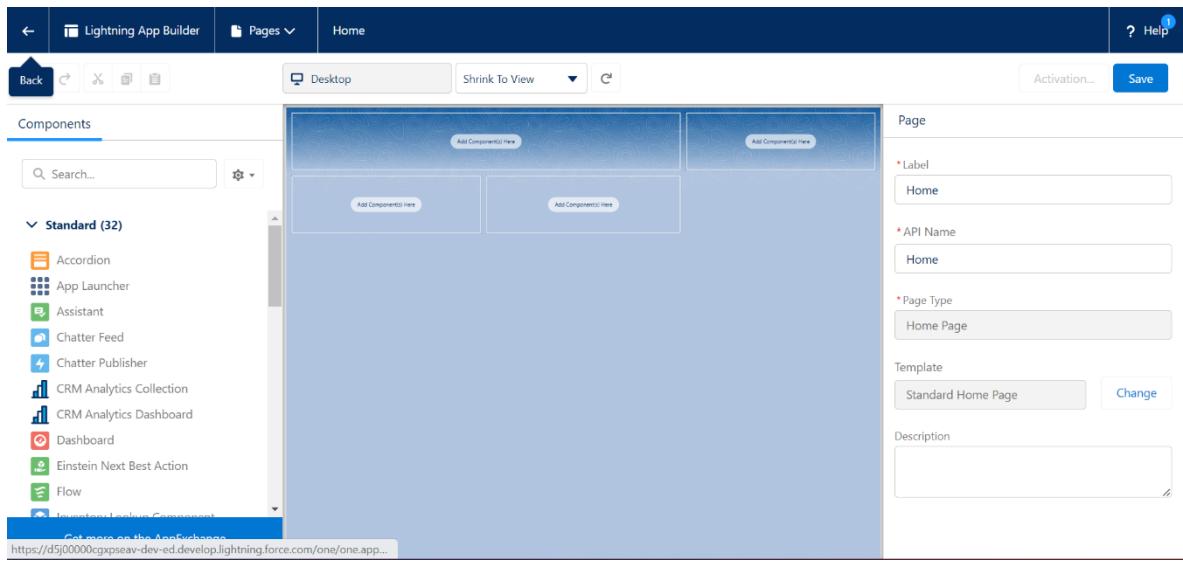
- After that click on home page and click on next.



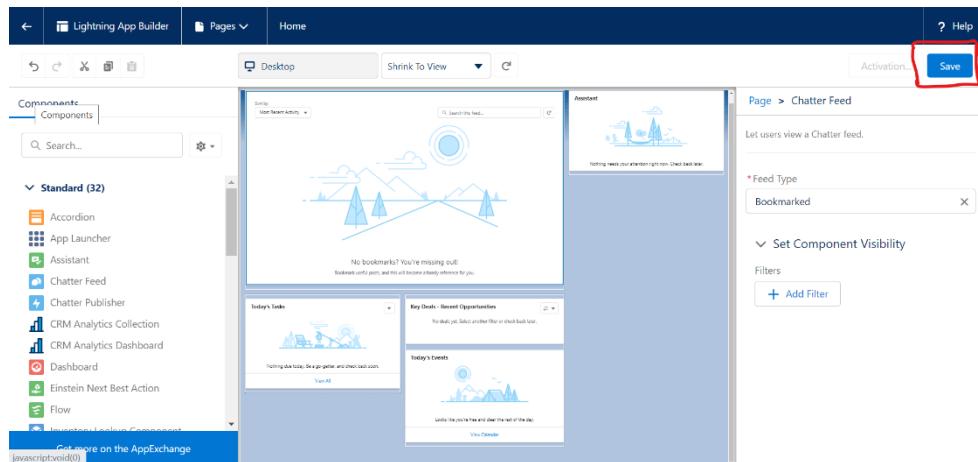
- Now enter label (i.e., Home).
- After that select Templates and click on done.



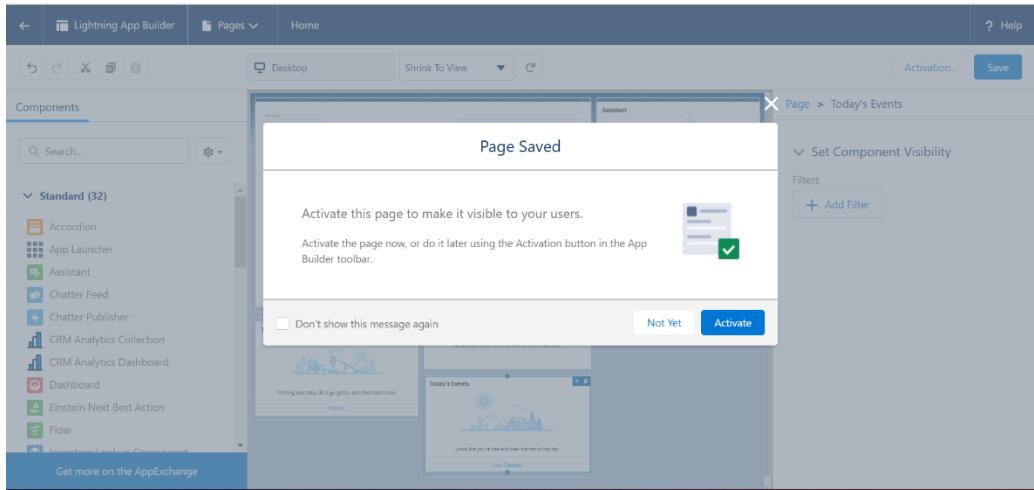
- Below image show lightning page in which we can add components.



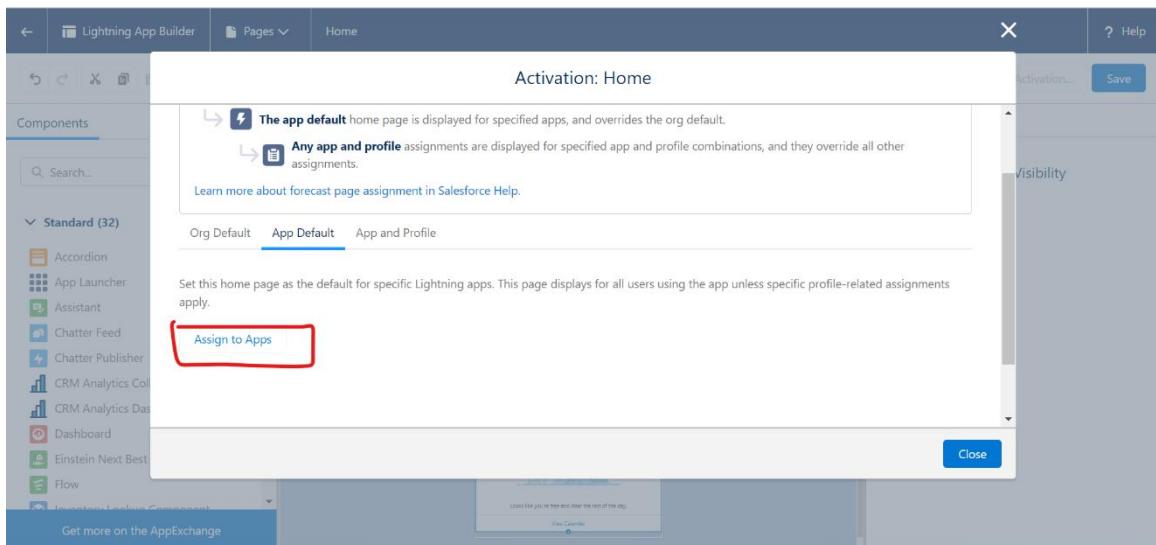
- Now let's add component from left hand side menu. just select the component and drag to the add component box in blank lightning page (i.e., we select Assistance component from left hand side components menu).
- This way we can add components.
- Now click on save.



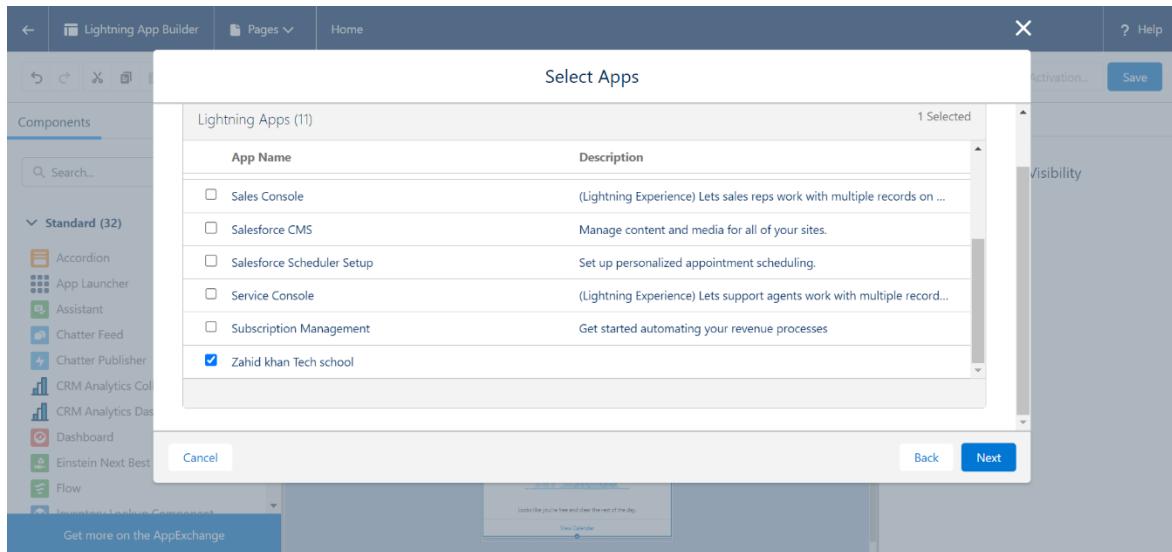
- After that click on activate.



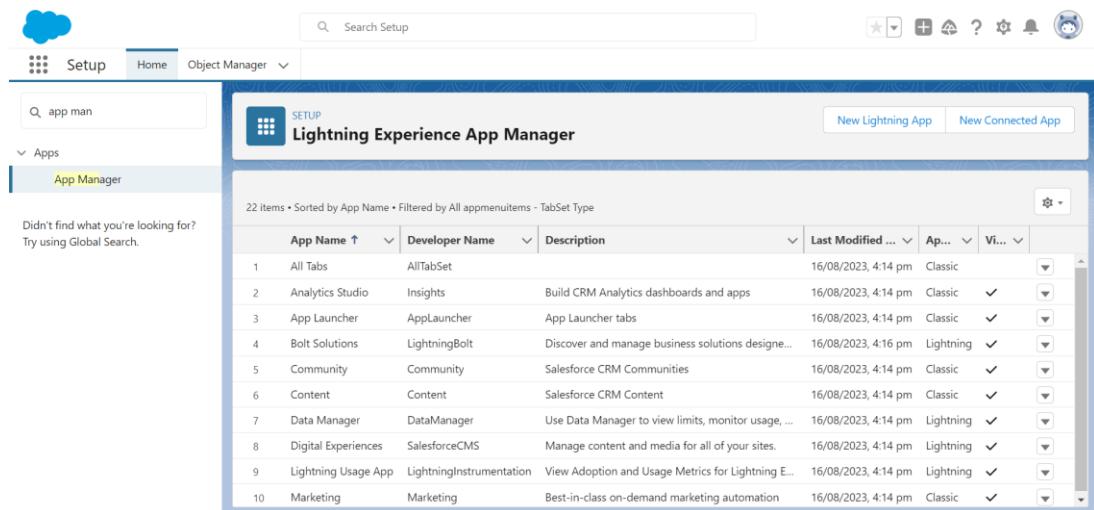
- After that we have 3 options **org defaults**, **App default**, and **App and profile**.
 - I. If we want to add these lightning page to default org, then we will choose **org defaults**.
 - II. If we want to add this page in particular org, then we need to choose **App default**.
 - III. If we want to add this page to any particular page with different profile, then we need to choose **App and profile**.
- Here, we are selecting App default because we have to assign home page to our app.
- Click on App default after that click on **assign to apps**.



- Now choose app name.



- Click on next.
- Now click on save.
- Now enter this newly created page to the app manager. For this again go to home and search for **app manager** in quick box.
- Now click on app manager.



- Search for the app (i.e., Zahid kha tech school). And click on arrow and select edit to edit the page.
- now click on navigation items.

Click to go back; hold to see history

Lightning App Builder

App Settings

Pages Zahid khan Tech school

Help

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

Navigation Items

Navigation Items

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

Available Items

Type to filter list...

- Accounts
- Alert Settings
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Invitations

Selected Items

- Students
- Instructors
- classes
- StdClasses

- After that search for home page which is created and insert that page using the arrow and make the home page on 1st tab and save.

Back

Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

Navigation Items

Navigation Items

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

Available Items

home

- Digital Experiences Home
- Home
- Mobile Home

Selected Items

- Home
- Students
- Instructors
- classes
- StdClasses

Cancel Save

javascript:void(0)

- Now go to front-end of the page.
- After that click on pencil icon on right hand side. Now click on **Add more items**.

The screenshot shows the cognizant application interface. At the top, there is a navigation bar with links for Students, Instructors, classes, and StdClasses. A search bar is located at the top right. On the far right of the header, there is a red rectangular box highlighting a pencil icon, which is typically used for editing or creating new items.

Below the header, there is a table titled "Students" with the following data:

	Student ID ↑	Name	Age	Email	Class
1	S-0010	Zahid	26		Salesforce Admin
2	S-0011	Ahmad	24		Salesforce Admin
3	S-0012	Shahid	28		Salesforce developer
4	S-0013	Rahul	24		Salesforce developer

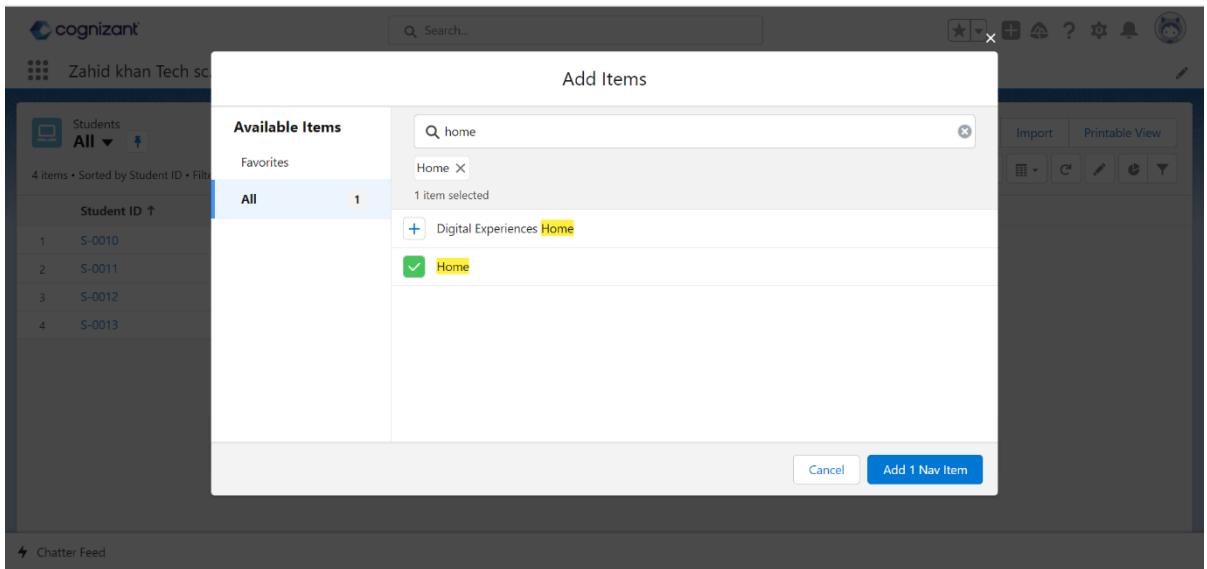
At the bottom left of the main content area, there is a link labeled "Chatter Feed".

- click on Add more items.

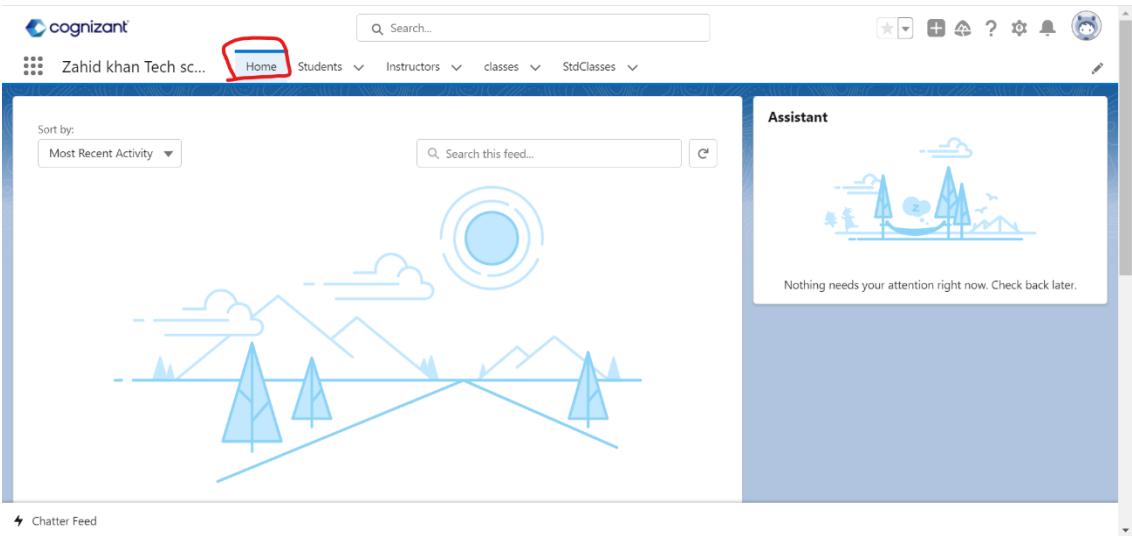
The screenshot shows the cognizant application interface with a modal dialog box titled "Edit Zahid khan Tech school App Navigation Items". This dialog allows users to personalize their navigation bar by adding more items. It includes a section for "NAVIGATION ITEMS (4)" which lists the current items: Students, Instructors, classes, and StdClasses. There is also a button labeled "Add More Items" and a "Cancel" button.

- Now click on All and search for home which is created.

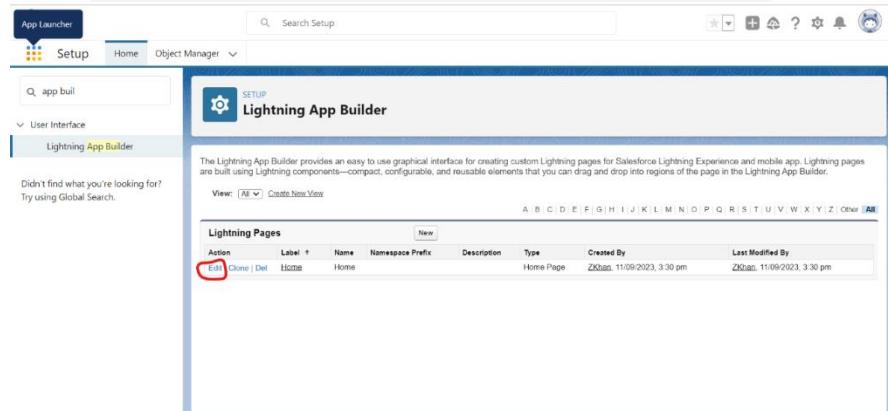
- After that select Home and click on **Add 1 nav Item**.



- Now click on save.
- Now home page added in front-end of app.

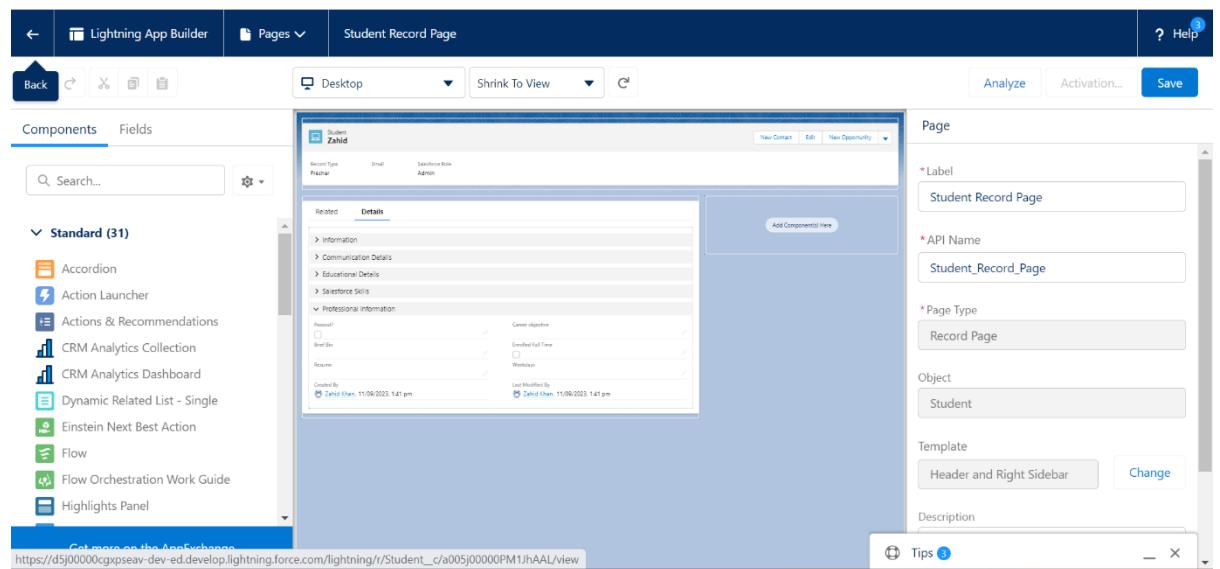
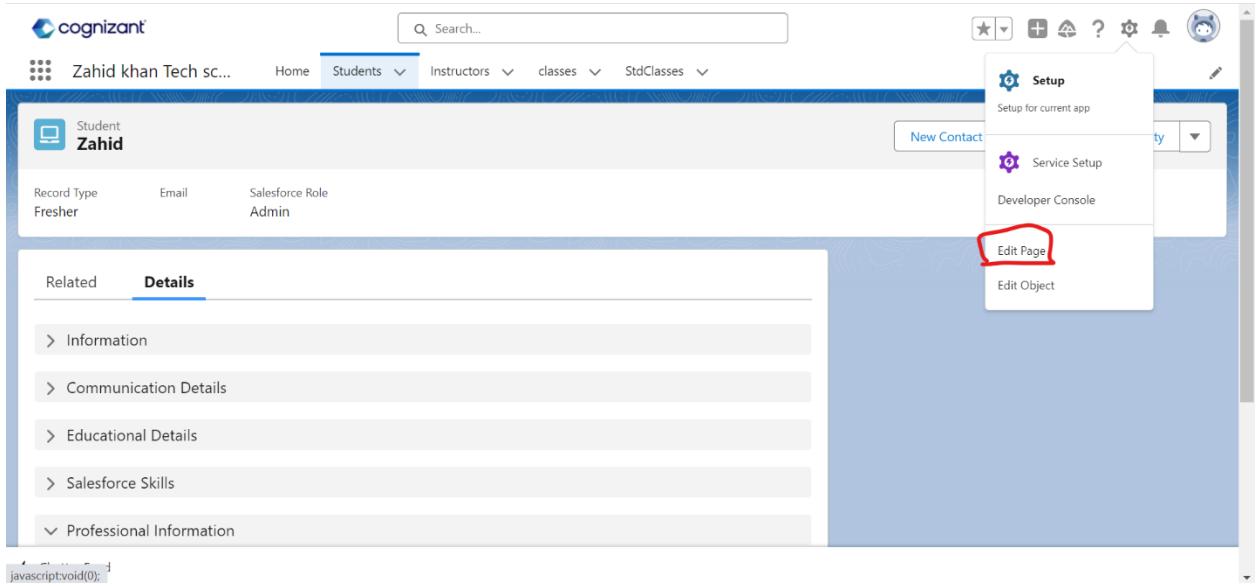


- If you want to edit the created lightning app then go to setup and search for lightning app builder, then click on it after that click on edit button.



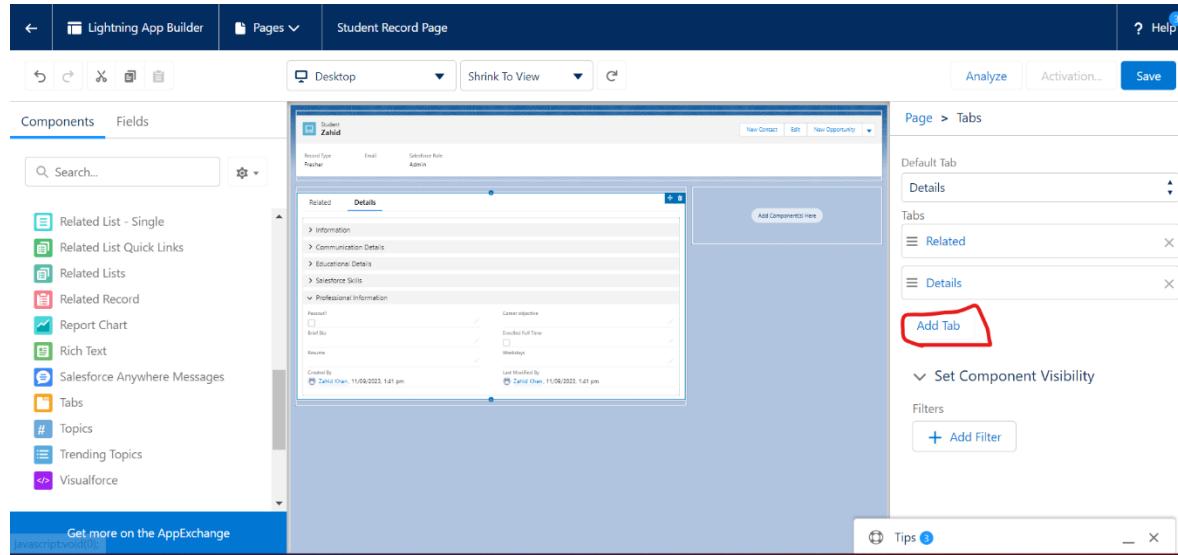
Steps to edit record page:

- To edit the record page of any object. Firstly, go to the front-end, then click on the object which records want to edit and open any records.
- Now click  on icon which is on right side the page.
- After that click on **Edit page** from the drop-down menu.
- After that lightning page of that object will open. For example, we are taking student object here.

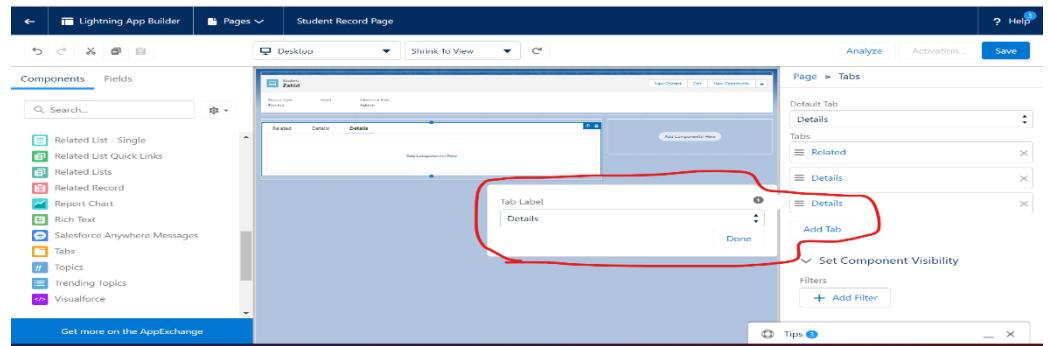


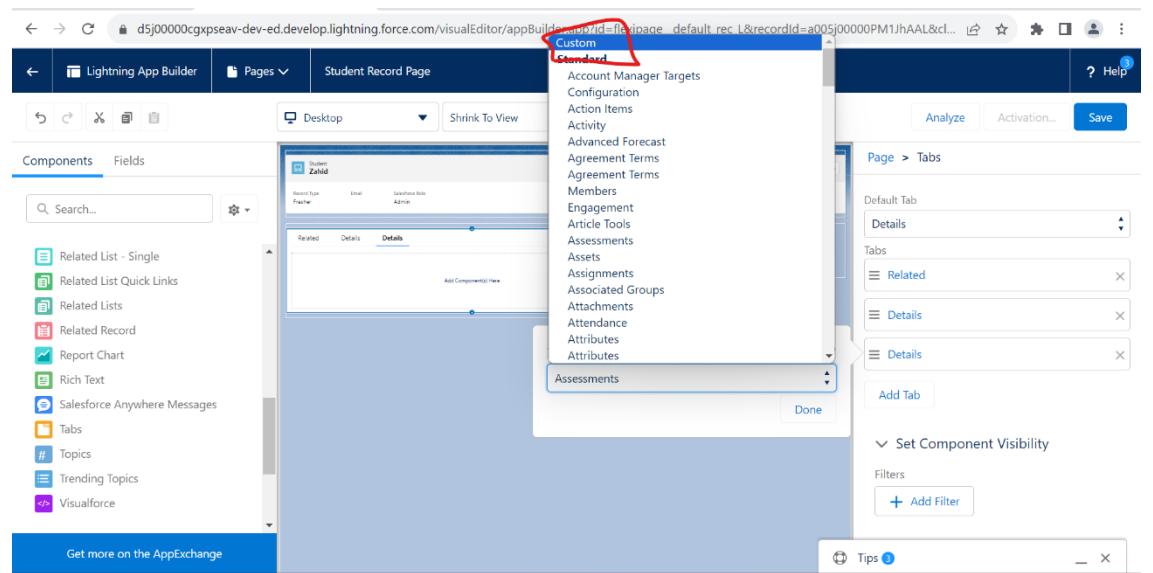
- now we can add any components and field from right side menu. We can also create new tab besides **related**, **details** tab.

- I. For example. We are creating Eligibility tab beside details tab. For this click on **add tab** from right side menu.

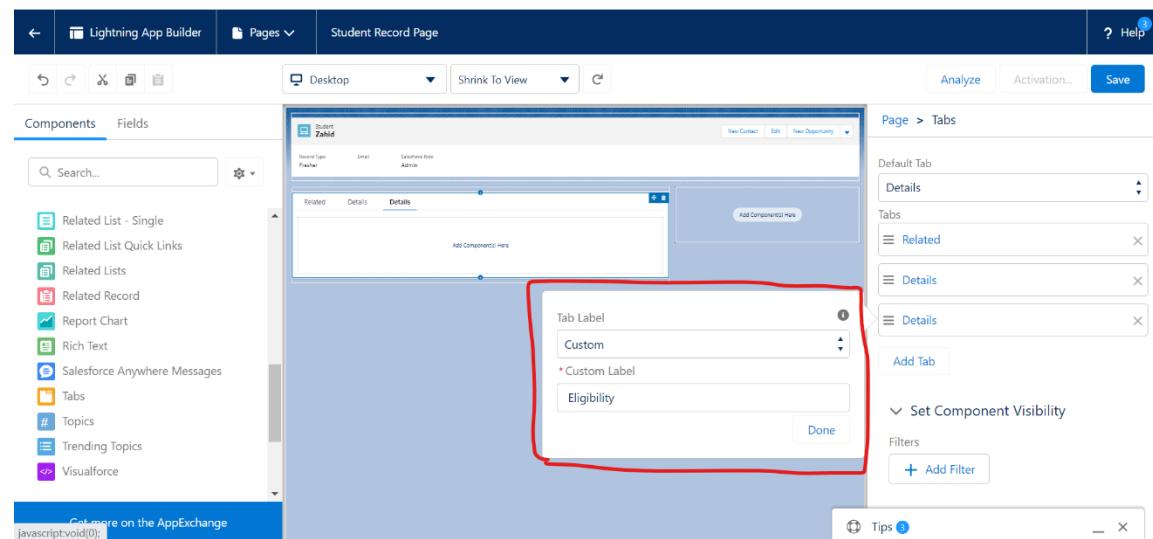


- II. Now click on the name of tab and from menu select custom.

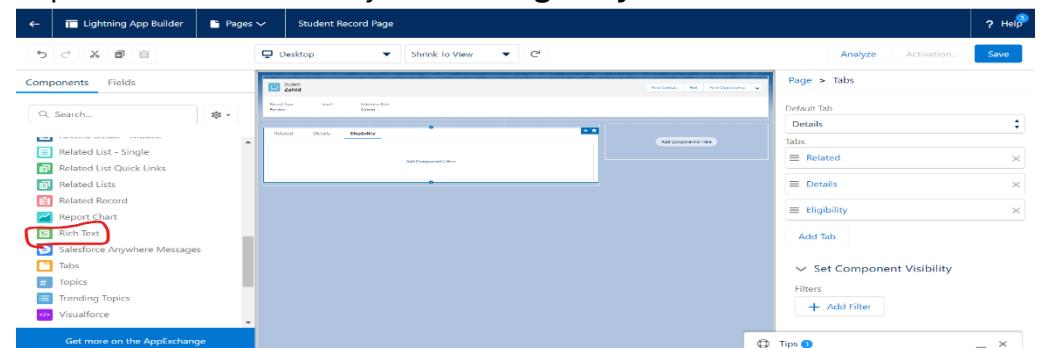




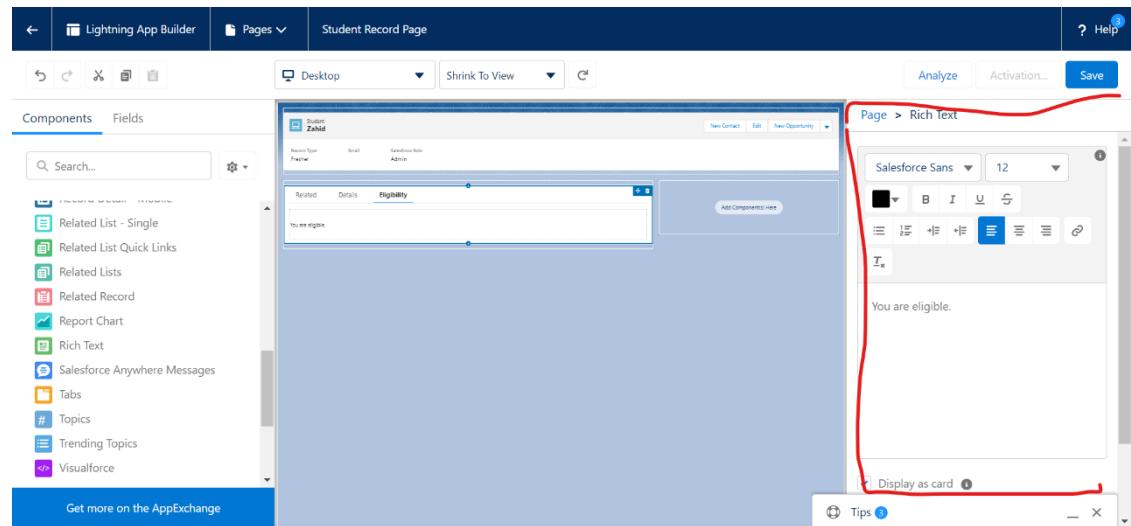
- I. after clicking on custom. Enter label name and click on done.



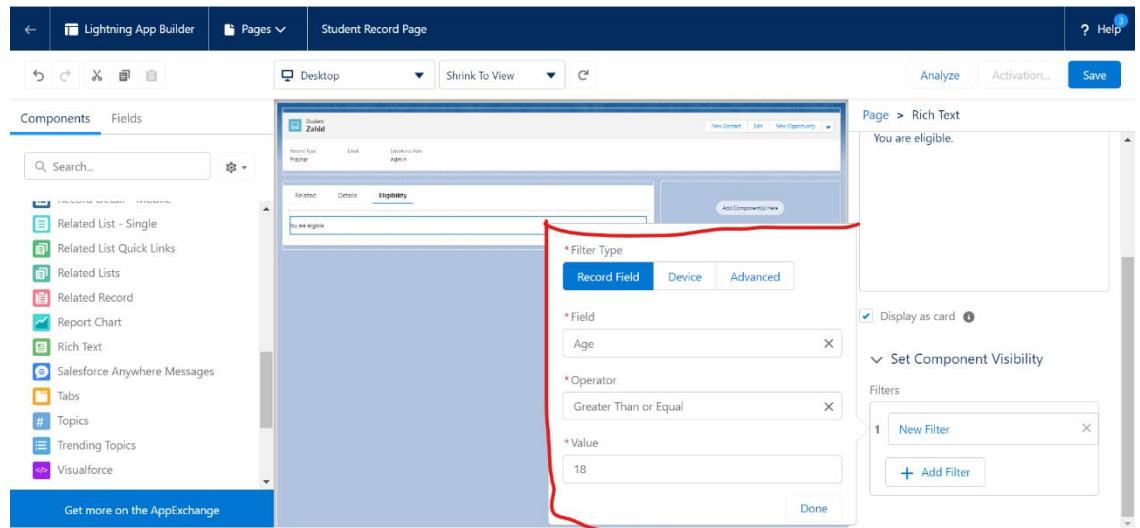
- II. Now new tab is created and we can enter any components in this tab. For example, we are adding rich text from right side menu and place it inside the newly created **eligibility** tab.



III. After that enter any text from right side.

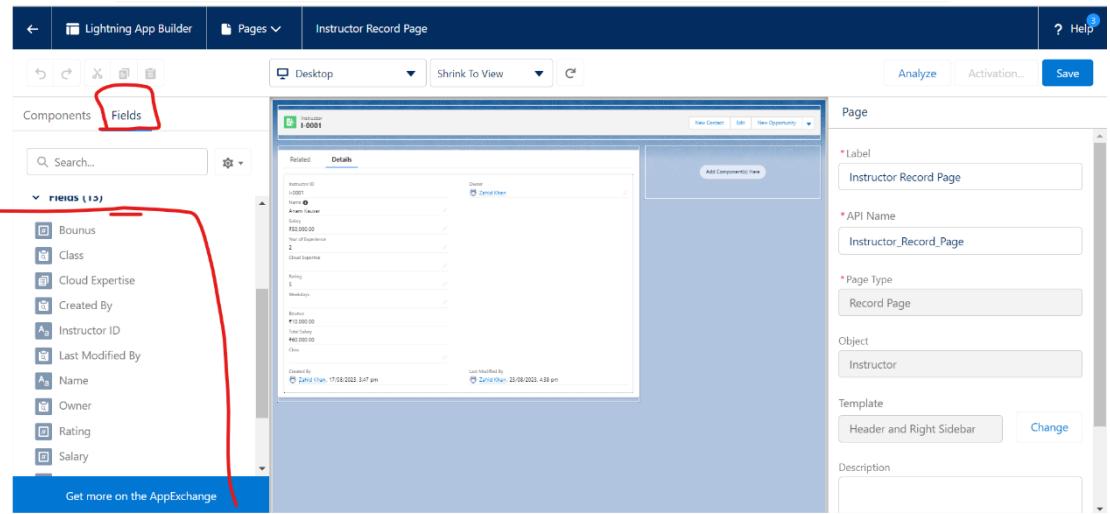


- IV. After saving this text will be shown on student object inside every records.
- V. We can also add some filter based on that this tab will be visible. For example, I make a filter that if age is equal to or greater than 18 then this text inside eligibility tab will be visible. Filter should me apply by clicking on Rich text field inside Eligibility tab.

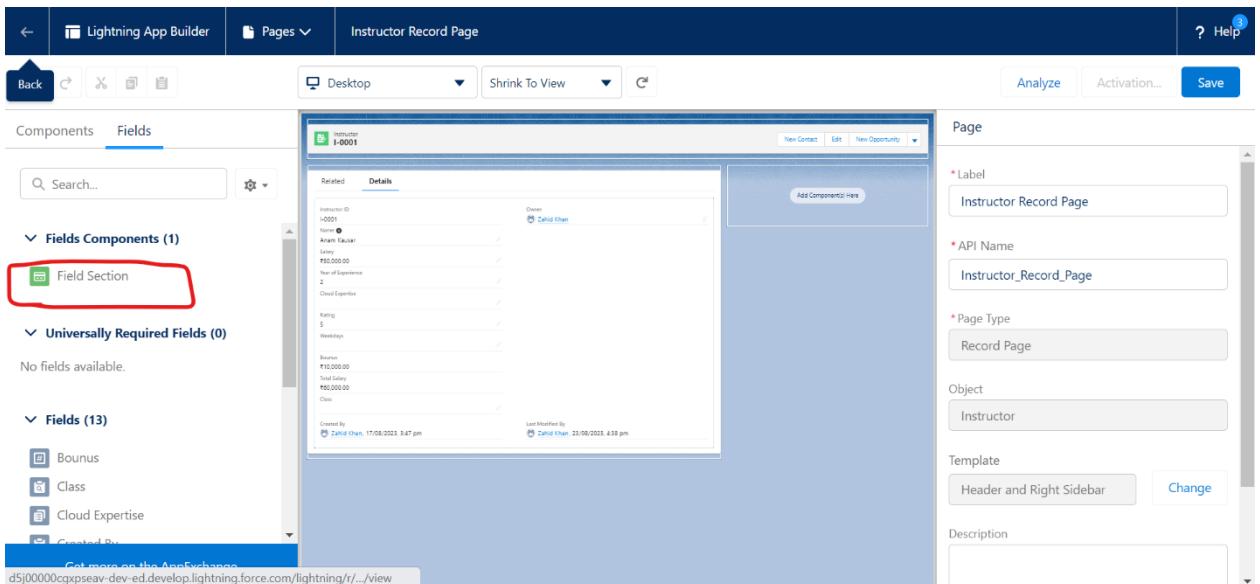


- Now click on save.
- After that click on activate.
- Now click on **App default**.
- After that click on assign as App default and select app i.e., Zahid khan tech school app. And click on next.
- Now select Form factor for both **Desktop and phone**. And click on next.
- Now click on save.
- Now go to front end and check.

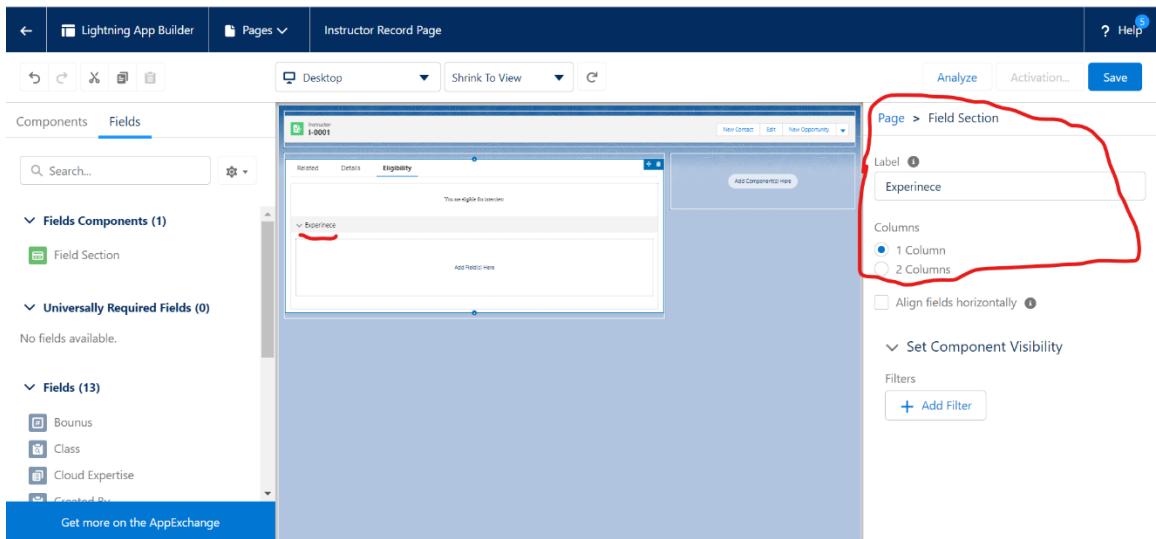
- We can also add field in any object by editing its lighting page by clicking field option on right side.



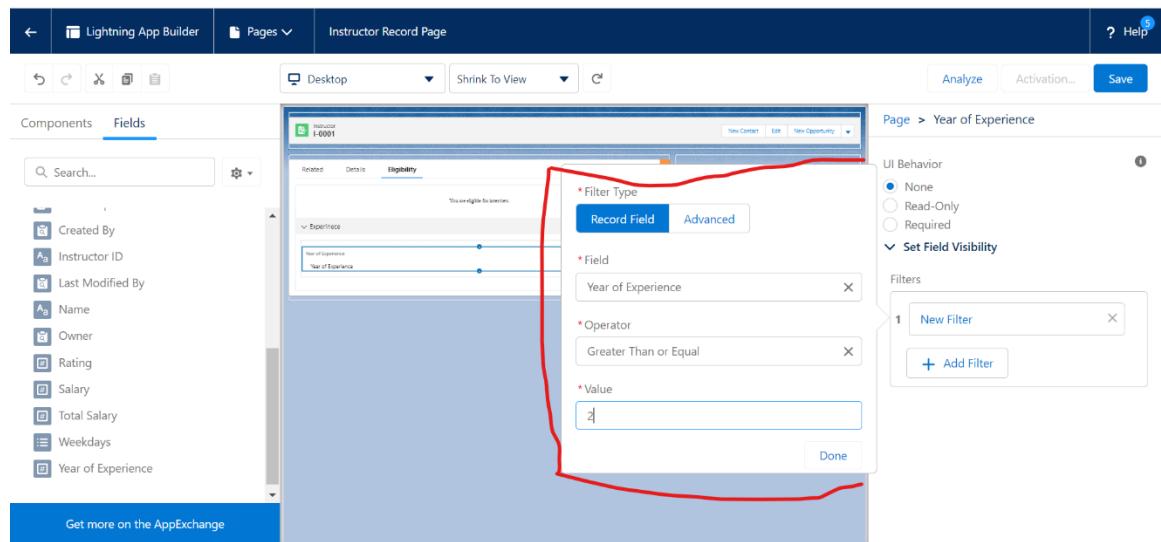
- We can also add different session (like we did from page layout) from field component and add the filed which is to display inside that section.



- For example, we are adding few sections inside **eligibility** tab in instructor object.
- Click on field on right hand side and click on section field and drag it inside eligibility tab and rename the section as experience.



- now click on field from right hand side and drag year of experience field inside the experience section.
- We can also add filter for this field based on that this section will visible under the set filed visibility.



- Now click on save and check result on front-end.

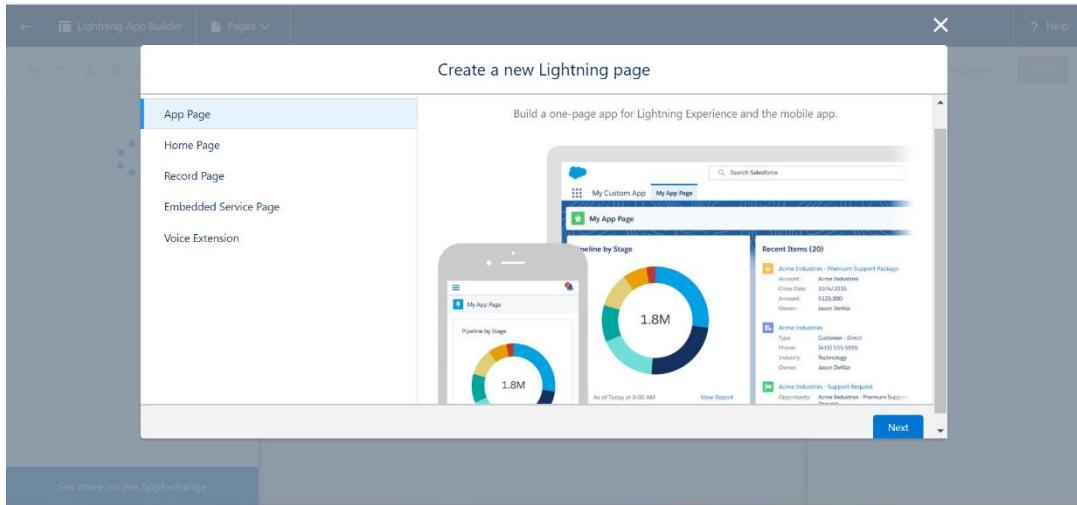
↳ Chatter Feed

Steps to make App page:

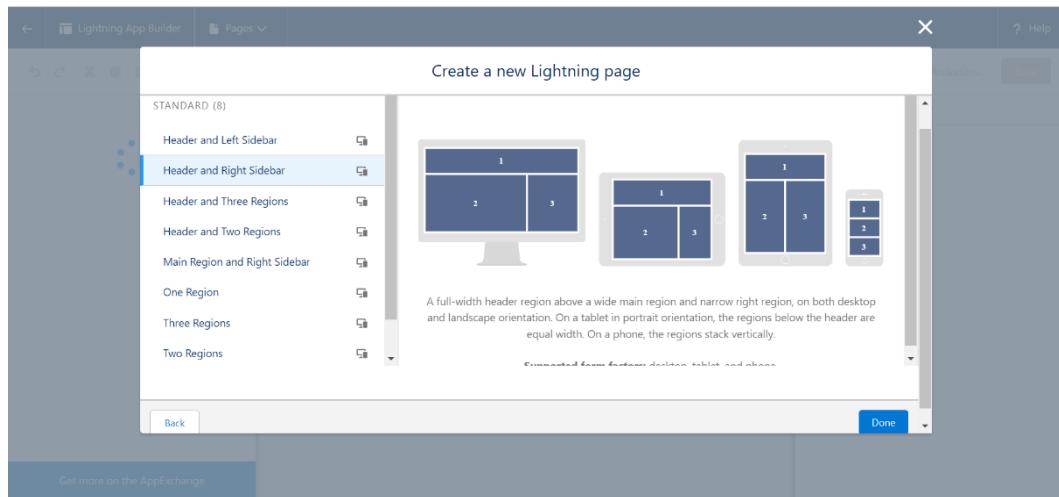
- Firstly, go to setup.
- At home, enter **Lightning App builder** in quick search.

- click on that.
- After that click on new.

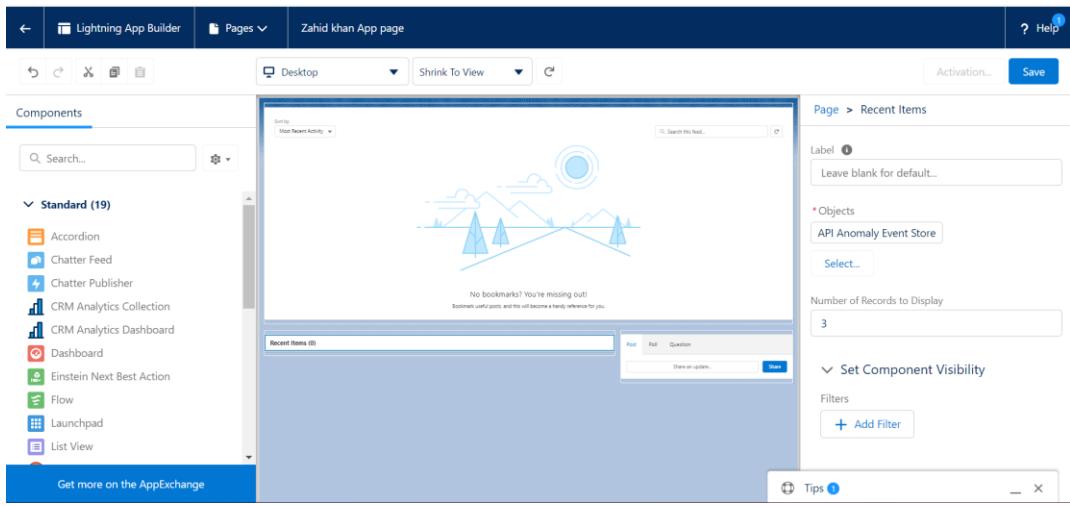
- now click on App page and then next.



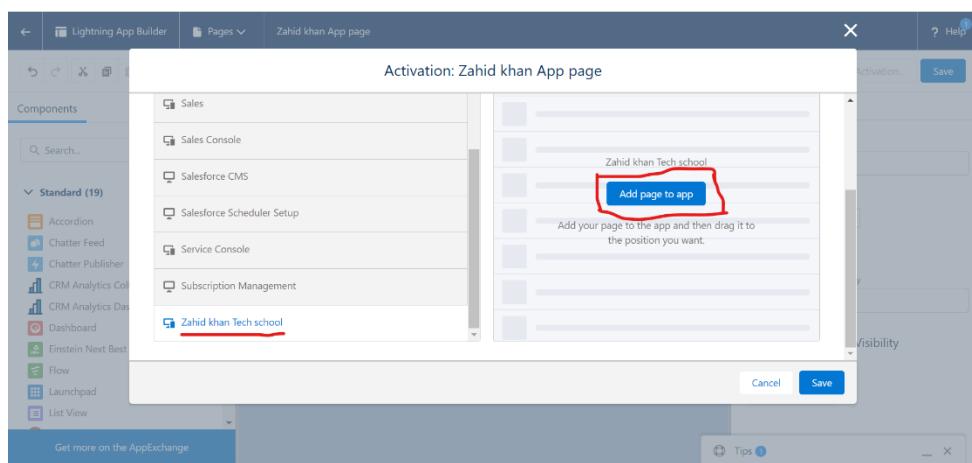
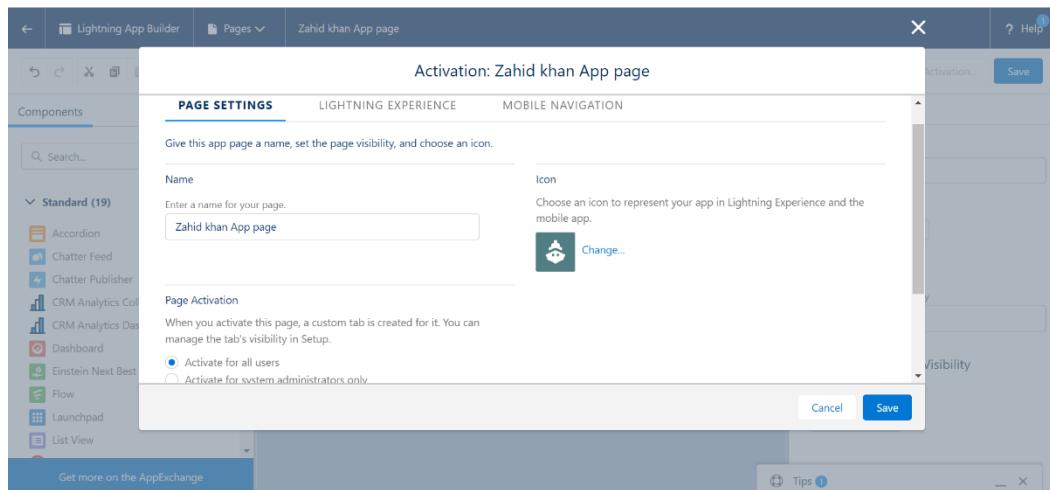
- enter label name. and select templates. And click on done.



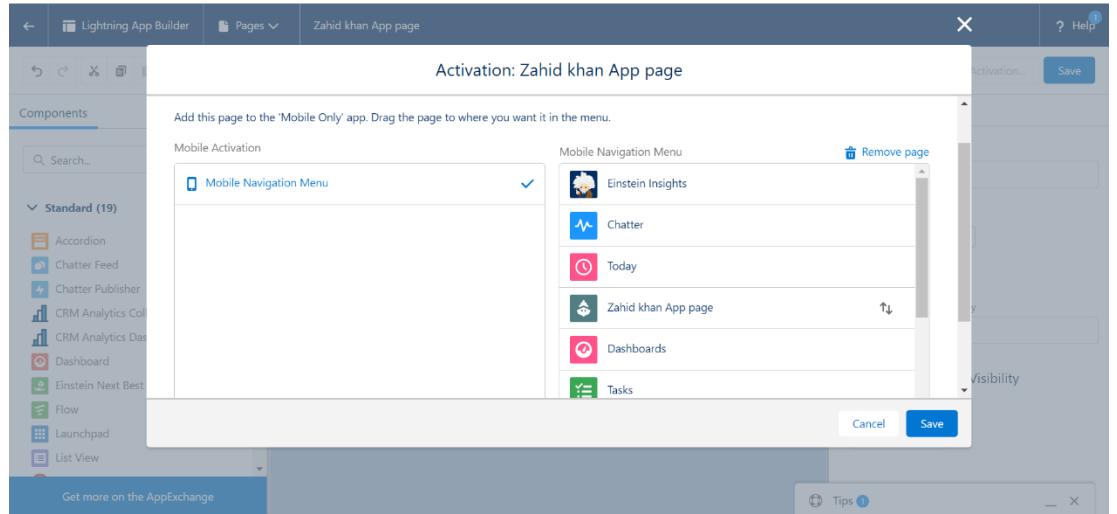
- App page is look like home page. But the different is that in home page we have one in Application but in App page we can have many in one application.
- In app page similarly we add components from left hand side.
 - I. In this example we add Chatter feeder, recent items and chatter publisher from components.



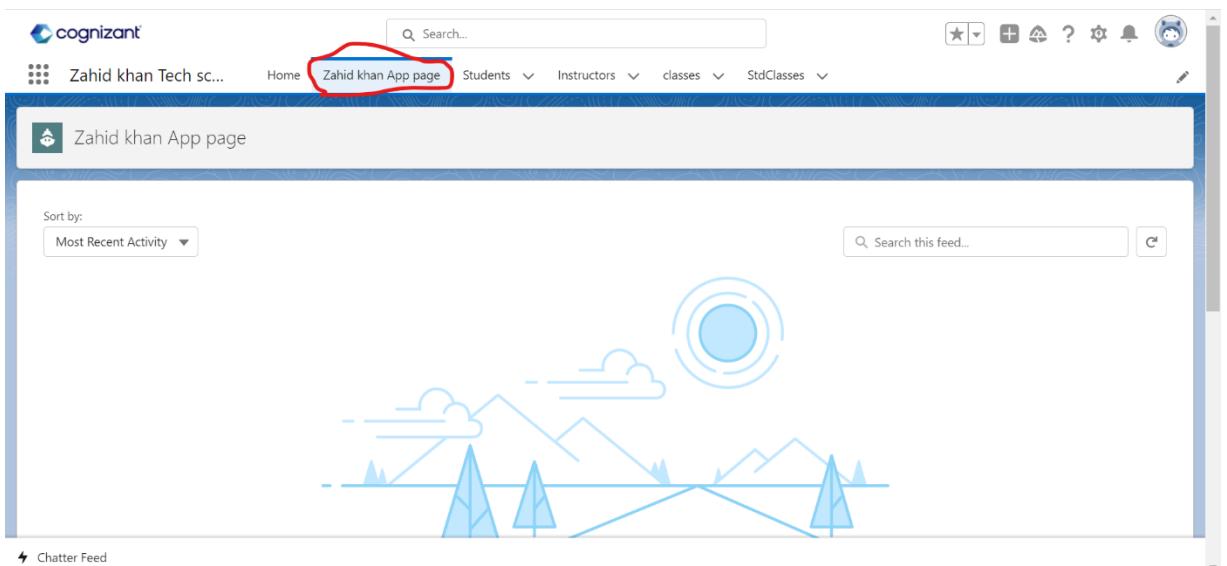
- Now click on save.
- After that click on activate.
- Now **Page Setting** under which we get option that **activate for all user or active for system administrators only**.
- next option we got **Lighting Experience** under which we have to choose App (we are choosing Zahid khan tech school). And click on **add page to app**.



- Third option we get Mobile navigation.in which we have to click on add page to app and we can drag the App to any position.



- Now click on save. This way we assign an App page to an application.
- Now go back to the Zahid khan Tech school app and check the result.



Day-07

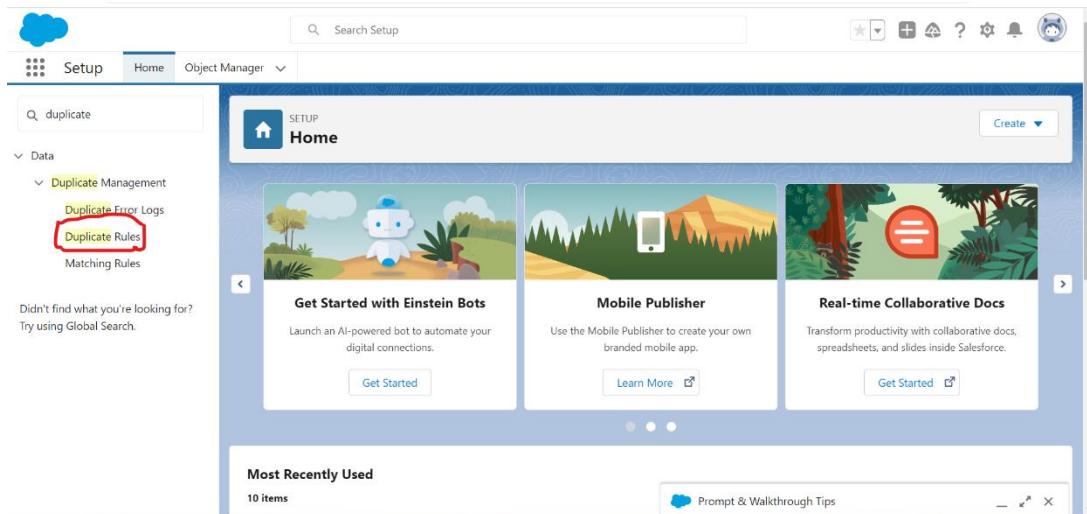
Duplicate Rule & Matching Rule

Duplicate Rule: - duplicate rule prevent creation of duplicate records. It shows either warning or block record creation and updating in case of duplicates. We can create reports on duplicate records as well.

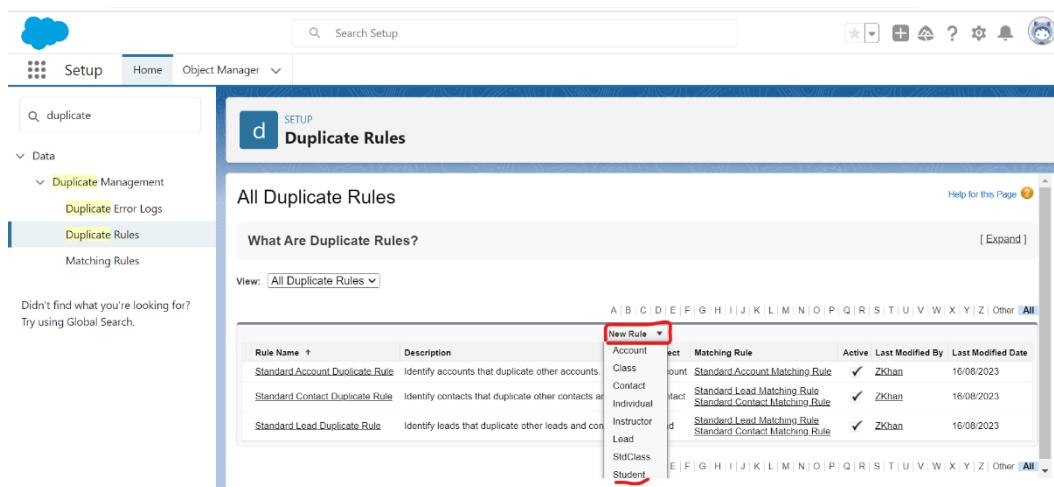
Matching Rule: - Here we need to set the criteria for duplicate check.

Steps to make Duplicate Rule for Records:

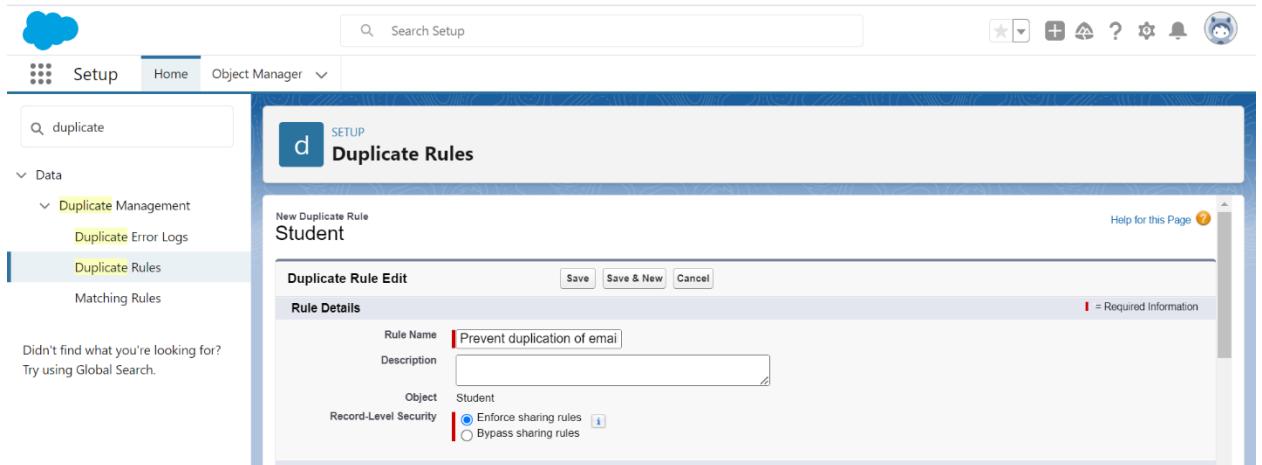
- Click on this  icon and open step up.
- Now click on home and Type **duplicate Rules**. And click on it.



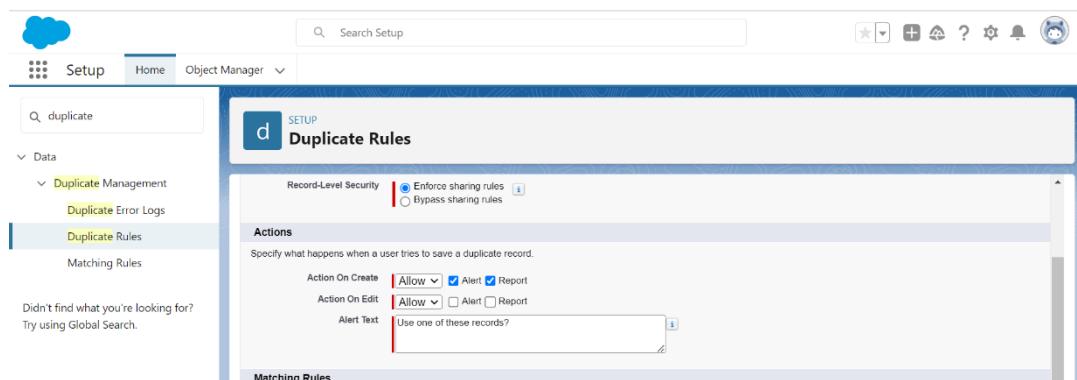
- On next window we got 3 duplication rules already it is standard duplicate rule. We can also create custom duplicate rule.
- To create duplicate rule, click on new Rule drop down and select the object in which new duplicate rule need to create. For example, we are choosing student object.



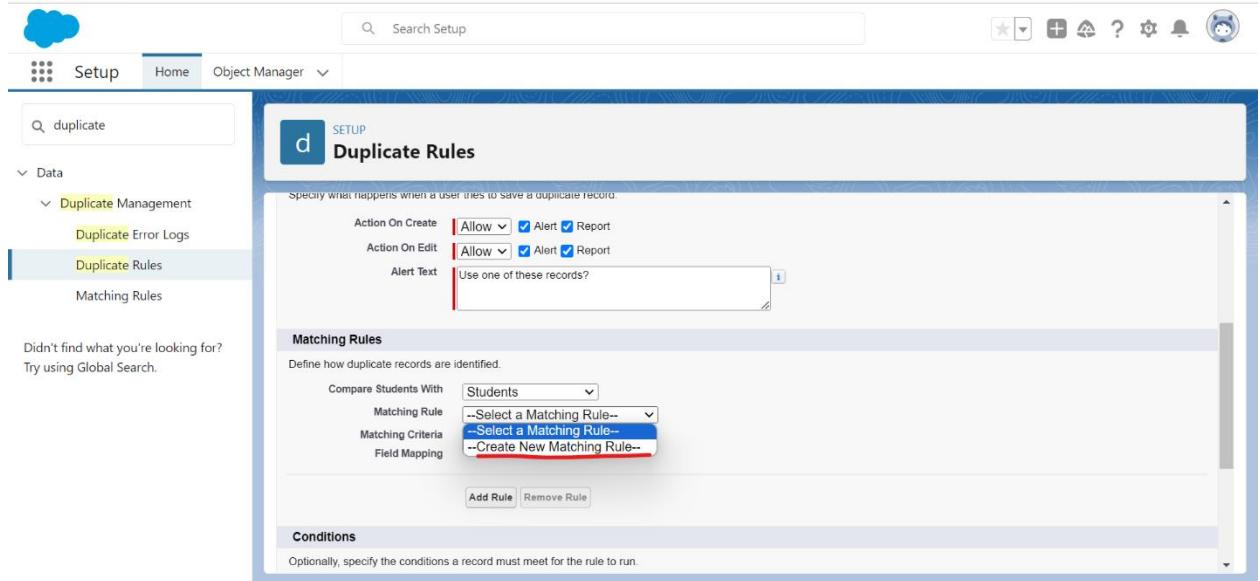
- On next page we have section for Rule details to enter Rule name, description, and Record-level security.
- In record level security we have 2 options Enforce sharing rule, Bypass sharing rule.
- Enforce sharing rule:** - This will compare the records on some specific assign role. Who have access of records.
 - Bypass sharing rule:** - This will compare all the records regardless of user access.



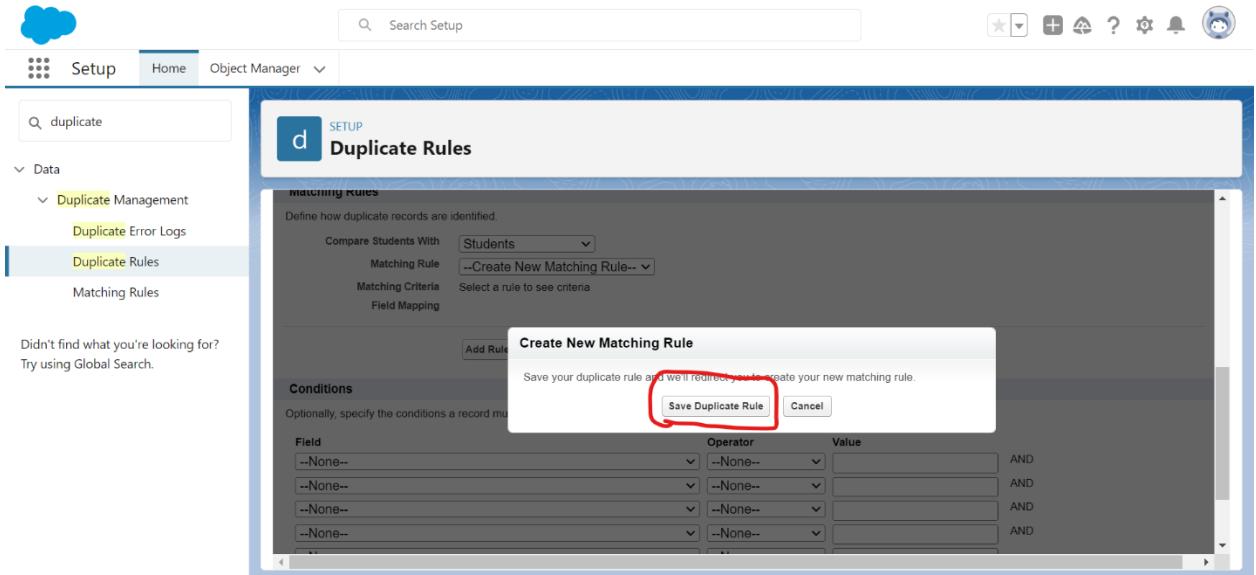
- after that we have a next section called Actions. In which we have three options Action on create, action on edit and alter text.
- In action on create, we have 2 options in drop down **Allow** (means if duplicate value is created then its just give warning and submit the information.) and **Block** (if duplicate value created it gives error and block to submit the form). And with Allow we have 2 check box alert and report always check this box when selecting Allow.
 - In action on edit, we have same option as action on create Allow and Block. Allow give warning and submit form and block give error and block the submission of form until correct the duplicate value. And with Allow we have 2 check box alert and report always check this box when selecting Allow.
 - Now enter Alter text which you want to display when duplicate value will occur.



- Now we have next option **Matching rule** in which we have to create a rule to avoid the duplication of record. In this section we have three options compare student with Students (which is already selected), next is matching rule drop down in which we have to select create new **matching rule**.



- we have one more option called conditions but its optional.
- When we click on **create new matching rule** its show pop up **saves duplicate rule** and redirect to matching rule page.



- on next page we got 2 option Rule details and matching criteria. Rule details already pre-filed we have to set matching criteria. For example, we add email filed duplication rule.

- in matching criteria, we have 3 options fields, matching method, and match blank field. Match black field means like if one blank field is created and try to make another blank field so it's given error.

Matching Criteria

Tell the rule which fields to compare and how.

Field	Matching Method	Match Blank Fields	Logic
Email	Exact	<input checked="" type="checkbox"/>	AND
--None--	Exact	<input type="checkbox"/>	AND
--None--	Exact	<input type="checkbox"/>	AND
--None--	Exact	<input type="checkbox"/>	AND
--None--	Exact	<input type="checkbox"/>	

Add Filter Logic...

Previous Save Cancel

- now click on save.
- After that click on activate so that this matching rule will active.

Matching Rule Detail

Object: Student

Rule Name: Prevent duplication of email matching rule

Unique Name: Prevent_duplication_of_email_matching_rule

Description: Student: Email EXACT MatchBlank = TRUE

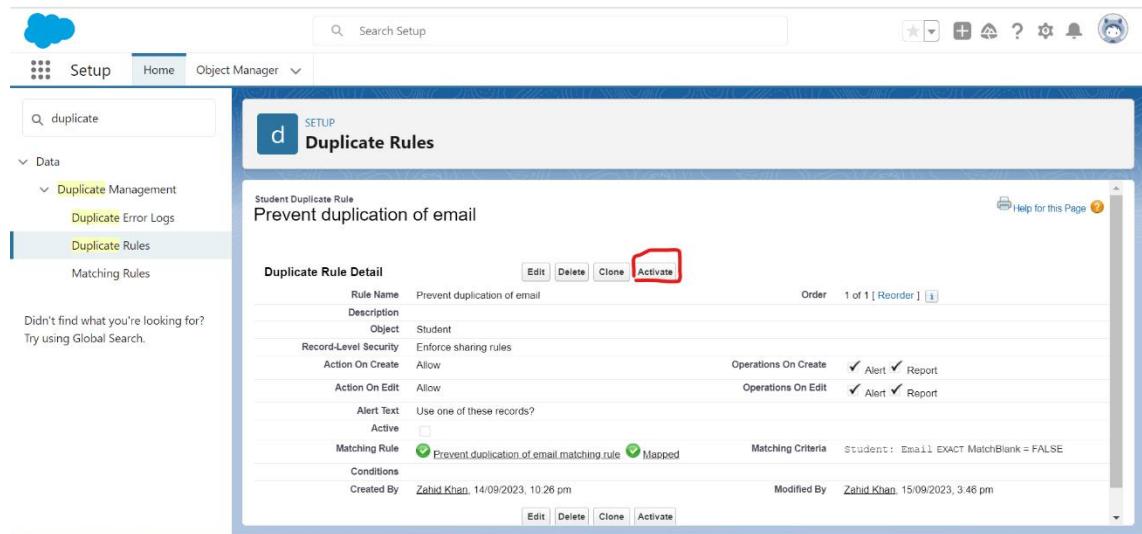
Status: Inactive

Created By: Zahid Khan, 14/09/2023, 10:34 pm

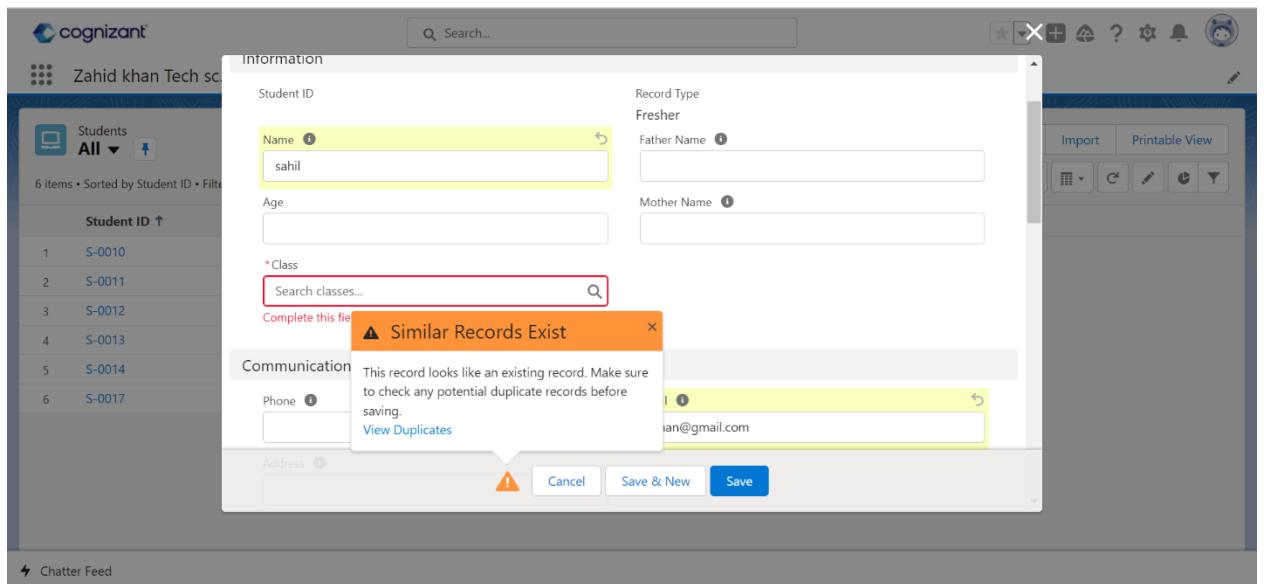
Modified By: Zahid Khan, 14/09/2023, 10:34 pm

Help for this Page ?

- Now it activates.
- After that go to duplicate rule again and click on activate to active the duplicate rule also.



- now go to front-end and check the result.



- it gives warning message for duplicate email and allow us to submit the form because we are selecting allow. If we select block, then duplicate rule will prevent us from submitting the form.

Activities

Activities has 4 options.

- I. Task
- II. Call
- III. Event
- IV. Email

Steps to add activities tab in custom object:

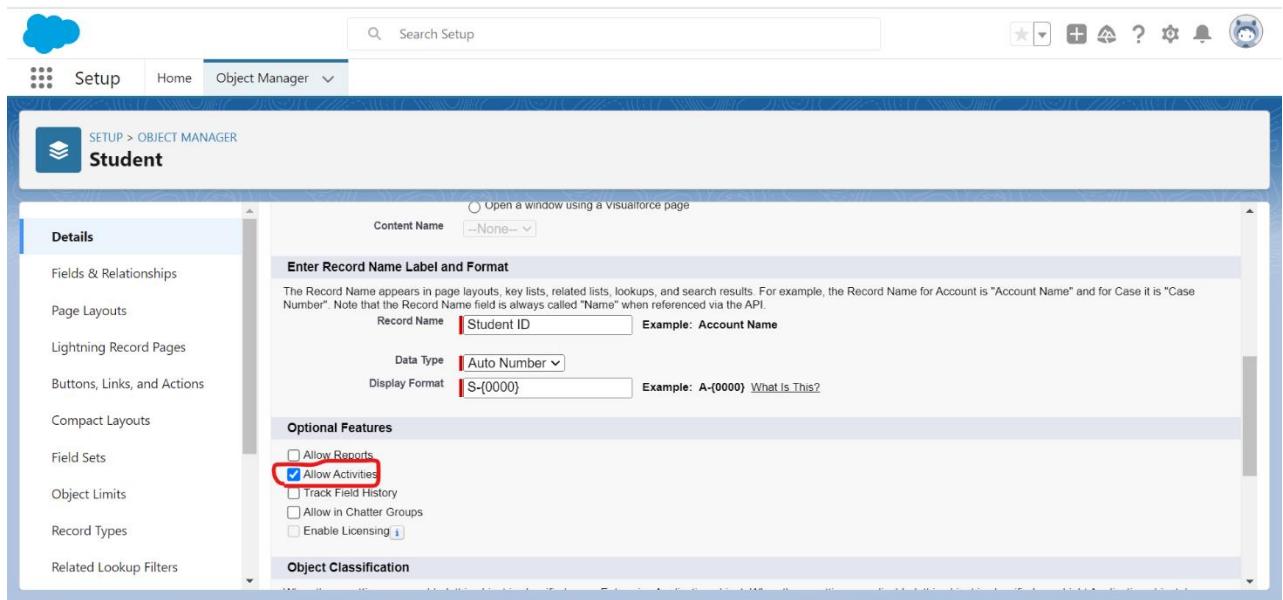
- 1st go the object manager and search for custom object in which we want to add activities tab (i.e., student object).

The screenshot shows the Salesforce Object Manager interface. At the top, there's a search bar with 'student' typed in. Below it, a table lists one item: 'Student'. The table columns are labeled: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The 'Student' entry has an API name of 'Student__c', is a 'Custom Object', and has a description 'This is a custom object.' The last modified date is '15/09/2023' and the deployment status is checked.

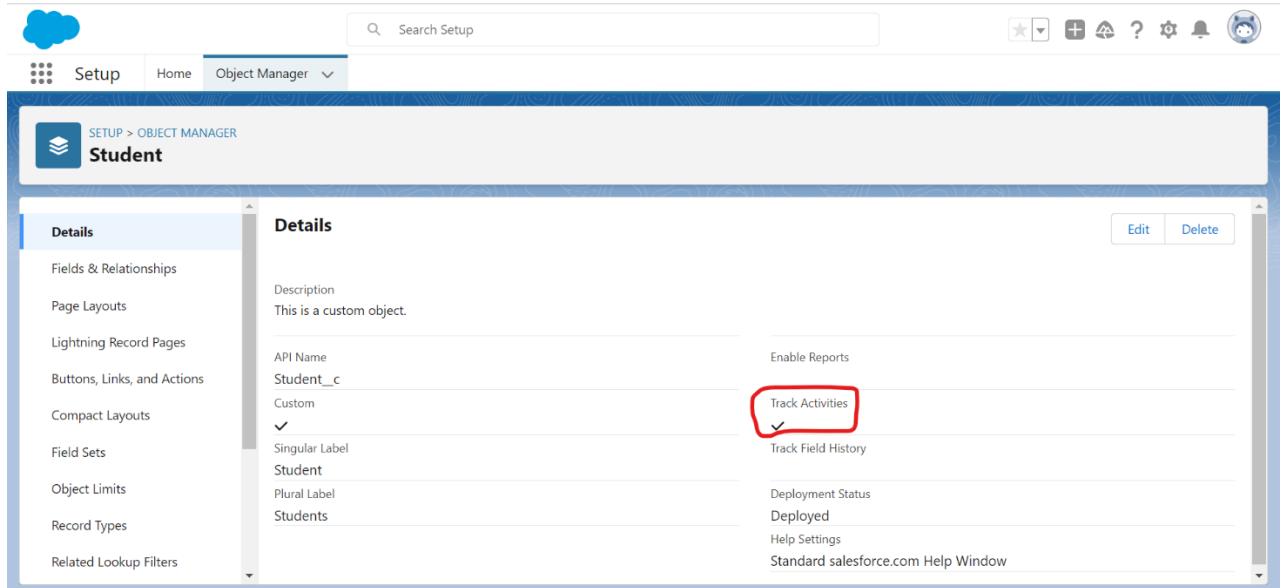
- Click on student object. Task activities will not check to enable activities in students records this box should be checked. For this click

The screenshot shows the 'Student' object details page in the Object Manager. On the right, there's a 'Details' section with various settings. At the top right of this section, there's an 'Edit' button highlighted with a red box. In the 'Details' section, there's a 'Track Activities' checkbox, also highlighted with a red box. Other settings shown include 'Enable Reports', 'Track Field History', 'Deployment Status' (Deployed), and 'Help Settings'.

- On next page go to optional features and check the box of allow activities.



- Now click on save. Now task activities will check.



- Sometimes it will add automatically in record and if is not add automatically then we need to add activities by editing record page.
- Now go to front-end click on student object and open any record. After that click on **setup icon** on right side of the screen and click on **edit page** to open lightning page of student object so that we can add **activities**.

The screenshot shows a student record page for 'Zahid'. The page includes fields for Record Type (Fresher), Email (zahid@gmail.com), and Salesforce Role (Admin). A context menu is open in the top right corner, with 'Edit Page' highlighted and circled in red.

- Now select activities from left side of the side menu and drag it to the student record page. And click on save.

The screenshot shows the Lightning App Builder interface with the 'Activities' component selected and highlighted with a red box. The component is being dragged onto a 'Student Record Page' template.

The screenshot shows the Lightning App Builder interface with the 'Activities' component successfully placed on the student record page. The component displays a timeline of upcoming events and tasks.

- We can also activity tab in lightning page. for this 1st drag the tab in lightning page than drag activities tag under tab. After that rename the tab as **activity**.
- Now go to front-end and check that activities tab will add in student records.

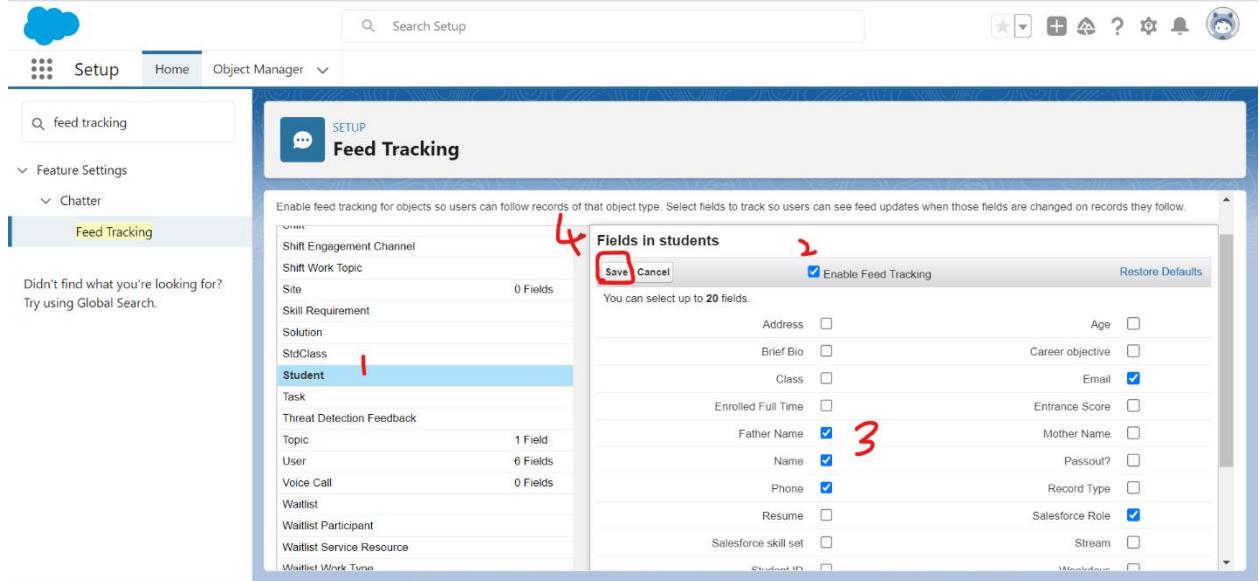
Feed tracking

- Feed tracking enables to track changes of selected fields and related fields as well.
- Changes are shown in the chatter component.
- It shows old value, new value and who changed.
- We select up-to 20 field for tracking field of any object.

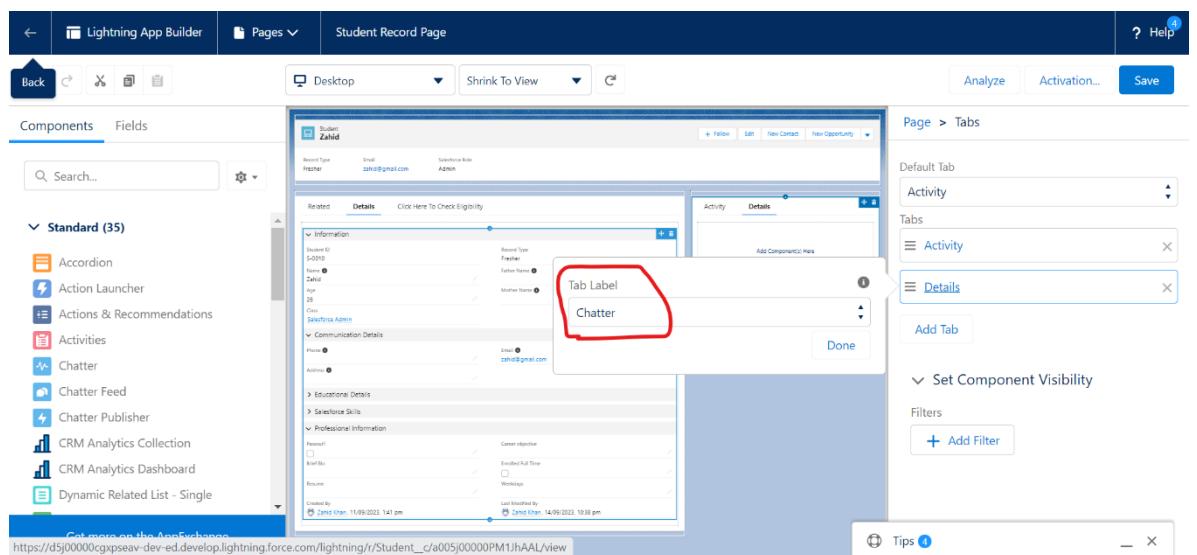
Steps to enable feed tracking:

- 1st click on setup and go to home and search for **feed tracking** in quick search.

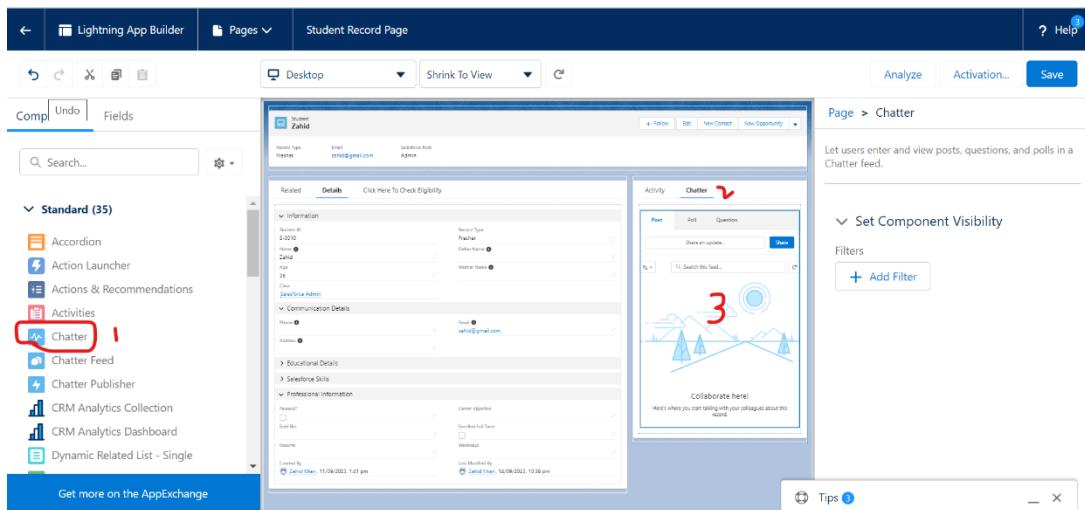
- Now click on feed tracking. Feed tracking page will open which consist of all the object. We can enable feed tracking for any object (for example, we want to enable feed tracking of few fields of student object).
- For enable student field check box of **enable field tracking**. After that select field in which field want to track. And click on save.



- Now go to front-end and check chatter is available or not in any student object. If not the open any student record click on setup icon on top right side and click on edit page to launch the lightning record page to add the chatter for feed tracking.
- 1st click on **activity** tab and add new tab **chatter**. And click on done.



- Now select chatter from left side menu and drag it to the chatter tab.



- Now click on save.
- Now go to front-end and check that chatter is added beside activity.
- Now if we change any field for which feed tracking is enable then it shows the details. Who change this? and what changes?

The screenshot shows a Salesforce page for a student named Zahid khan. The page includes sections for Information, Communication Details, and a Chatter feed. The Chatter feed displays an update from 'Zahid Khan' stating: "updated this record. Just now". The entire Chatter section is highlighted with a red box, and the specific update message is also highlighted with a red box.

Field history tracking

- field history tracking enables to track changes of selected fields.
- Changes are shown in the history related list.
- It shows old value, new value and who changed.

Steps to enable field history tracking:

- We can enable field history tracking for any custom object. Here we are enabling this for student object.
- For this 1st click on set up and click on object manager. Search for student object and click on this to open student object after that click on fields & relationships.
- After that click on **set history tracking**.

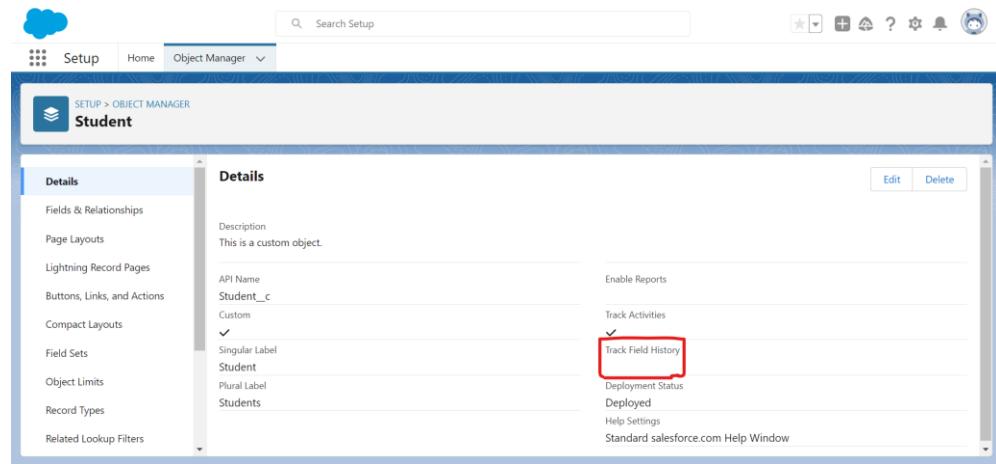
The screenshot shows the Salesforce Setup interface under the Object Manager for the 'Student' object. In the 'Fields & Relationships' section, there is a table listing various fields. The last column of the table has a header 'Set History Tracking'. This header is highlighted with a red box.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	Set History Tracking
Address	Address_c	Text(250)			
Age	Age_c	Number(3, 0)			
Brief Bio	Brief_Bio_c	Long Text Area(32768)			
Career objective	Career_objective_c	Text Area(255)			
Class	Class_c	Master-Detail(Class)			
Created By	CreatedById	Lookup(User)			

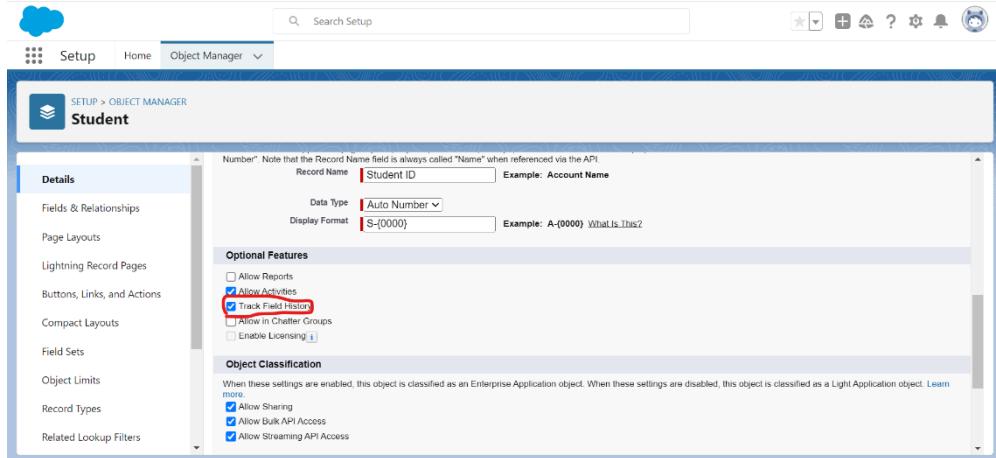
- On next page we will get list of fields of **track old and new value** and **another track change** (these field are text area long or text area rich field) only.
- Now select the field and click on save.

The screenshot shows the 'Track old and new values' configuration page. It contains two main sections: 'Track old and new values' and 'Track changes only'. Under 'Track old and new values', several fields are listed with checkboxes: Address, Career objective, Email, Entrance Score, Mother Name, Passout?, Record Type, Stream, Weekdays, Age, Class, Enrolled Full Time, Father Name, Name, Phone, Salesforce Role, and Student ID. Under 'Track changes only', the fields Brief Bio and Salesforce skill set are listed with checkboxes. Some checkboxes are checked (Email, Name, Brief Bio, Phone), while others are unchecked.

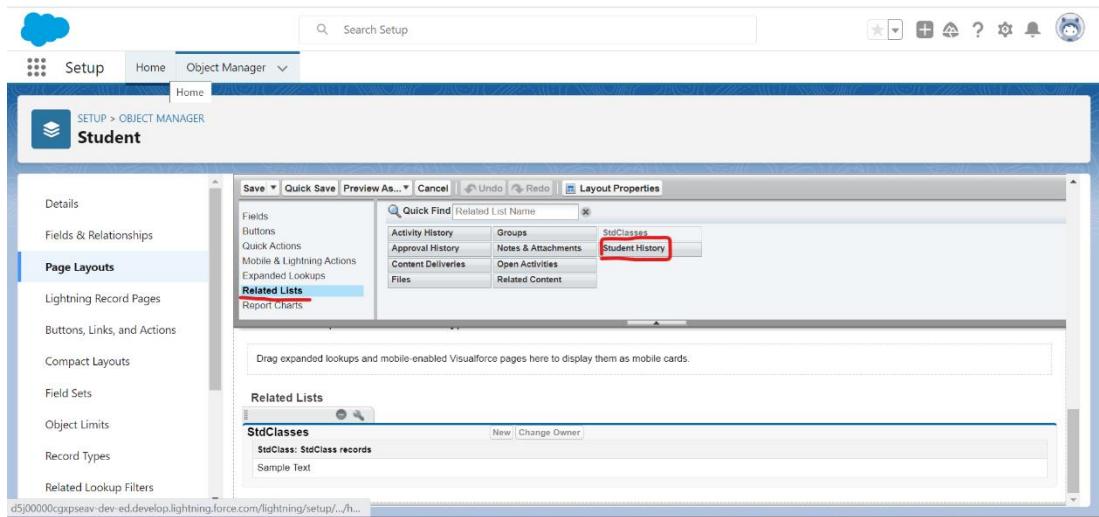
- After that go to the details page of student object and enable the **track field history**.



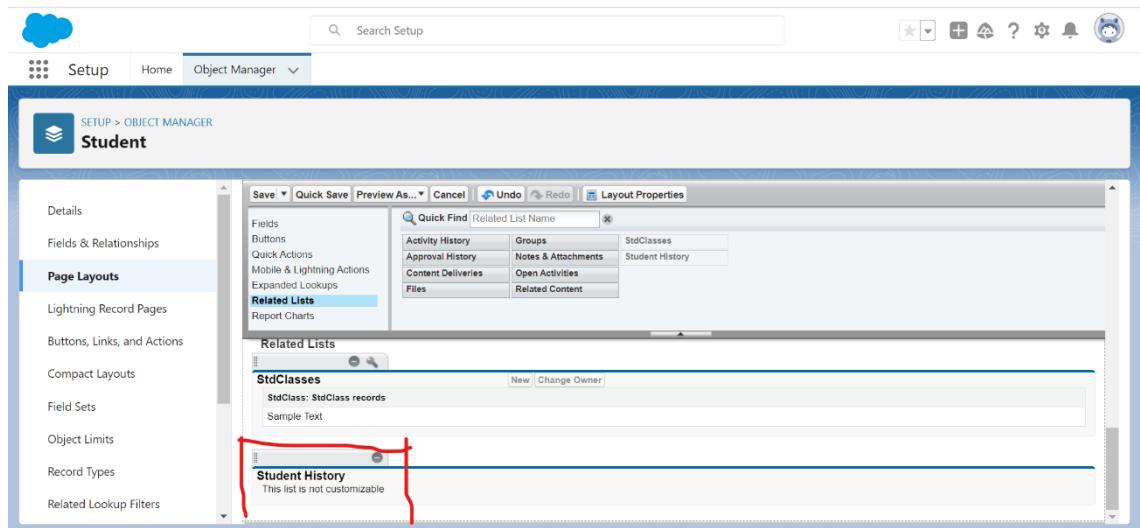
- Click on edit on right side. Under optional field check the box of Track field History.



- Now click on save.
- After that go to page layout of student object. We have to enable History page for both fresher and experience layout. 1st clicks on fresher layout. Now click on **related list** we will get an option **student History**.



- After that drag the student history below the stdClasses under related list.



- now click on save.
- Now similarly also place the student history in experienced layout and student layout.
- Now go to front-end open click on student object after that open any records and click on related tab that there is a student history option.
- Now if I change the field which I selected in set history tracking then I will show in **student history** tab.

