# PWR Components User Guide

By Orektic Solutions LLP

# Contents

# I Introduction

## I.I Orektic Solutions LLP

Orektic Solutions LLP is known for state-of-the-art cloud-based, SAAS solutions that help our clients to transform the way they do their business.

We exist to solve the critical issues facing our clients, both large and small. Our unique approach is not only what differentiates us, but also what makes us successful. We provide a broad range of services and solutions to help organizations facilitate change, achieve their vision, and optimize performance and productivity.

From implementing new business strategies to ultra-efficient work processes, Orektic Solutions is ready to tackle any challenge and put you on the path to success. With state-of-the-art cloud-based, SAAS solutions transform the way you do business.

Orektic is a sister concern of Enzigma Software, a pioneer company in the Salesforce ecosystem for almost 20+ years.

## I.II Enzigma LLC

Enzigma LLC is the exclusive reseller of Orektic products

## II Product Introduction

With our innovative thoughts, we have our new innovative product launched on AppExchange, which is an added advantage for all the Salesforce admins and developers, which enables them with drag and drop-able Lightning Web Components that can be used on the following pages:

- Home page
- App page
- Record details page
- Lightning flows
- Communities

To enhance usability and reduce the time to market for custom requirements. We all know Salesforce Lightning is the new world and classic deprecating soon. Due to many limitations and restrictions, which increase time to market and budgets. Lack of configuration and custom development results in slow adoption for Salesforce. To keep the time to market and budget at a minimum, with the ease of configuration, PWR Components can be the perfect fit.

PWR Components is not just a bundle to extend Salesforce limitations but also helps users to configure and use our new and innovative components which are totally new in the ecosystem.

- PWR Button Group
- PWR Icons
- PWR Progress Bar
- PWR Ratings
- PWR Related List
- PWR Timeline

The above components provide more power to the admins, and developers to format data in a specific format as per the requirements.

All components in the bundle are supported on the Record Page, Home Page, App Page, Lightning Flows, and Communities

It helps Reduce Cost and Time to Production.

## III Product Features

1. **PWR Button Group:**

    - PWR Button is used for calling any actions on the Lightning UI. PWR Buttons can be placed anywhere on the Lightning Pages
    - It looks like the button group in Salesforce. The advantage of using PWR Buttons is that they can be placed anywhere on the Pages so it is easily accessible

2. **PWR Icon:**

    - With the help of Icons users can show their data in a pictorial format or in the icon format

3. **PWR Progress Bar:**

    - PWR Progress Bar component helps salesforce users to get a graphical representation to show the progress of the work
    - PWR Progress Bar can be configured to show/hide the Percentage of the progress, on hovering over the component it shows the current percentage.

4. **PWR Rating:**

    - The PWR Rating component is used for rating purposes. With the help of the PWR Rating component, users can give ratings in different categories
    - We provide the Rating component to give feedback and provide the feature to change the appearance of the stars or the way the ratings are displayed

5. **PWR Related List:**

    - With the help of a related list, we can show the related data of objects and we can show the data up to 5 the level hierarchy
    - We can add more than 10 fields to the related list and so on

6. **PWR Timeline:**

    - We can show the timeline view for all the objects
    - The Timeline helps users view the activity Timeline for all configured objects. We can observe the activity timeline accordingly
    - The filtered format and many more functionalities are available in the timeline

# IV Product Versions

| Version | Release Date | Description |
| --- | --- | --- |
| PWR Components 3.8/3.8.3 | 04/24/2024 | We have added new features in the button group |
| PWR Components 3.5/3.5.21 | 02/20/2024 | We have added new features of the Summarize field and period component in filter |
| PWR Components 3.4/3.4.5 | 10/31/2023 | We have introduced the components: Button Group, Icon, Progress Bar, Rating, Related List, Timeline |
| PWR Components 1.9/1.9.3 | 11/26/2021 | First release |

# V Configuration Guide

## V.I Installation Steps

**To install PWR Components managed package do follow the mentioned steps here:**

1. To install **PWR Components** click here
2. The link will direct you to the PWR Commons application page, it is a prerequisite package for PWR Components
3. Click on Get it now from the apps information page
4. Enter the credentials for the org in which you want to install and log in
5. Choose to install the package in either the production org or Sandbox by clicking ”Install in Production” or ”Install in Sandbox”
6. Tick the checkbox indicating your agreement with the terms and conditions, then proceed by clicking on the ”Confirm and Install” button
7. Select the desired installation option (Install for All Users, Install for Admins Only, or Install for Specific Profiles), and click ”Install”
8. Click ”Done” once the installation is complete
9. It may take some time to complete the installation package. You will be notified through Email once the installation is completed
10. Now open the PWR Commons application from the application manager
11. Click ”Install Now” in front of the **PWR Components** application in the list of products
12. Enter the credentials of the logged-in organization and log in
13. Select the desired installation option, and click ”Install”
14. Click on Done once the installation is completed
15. It may take some time to complete the installation package. You will receive an email notification when the installation is finished

**** For reference and more details about package installation, please click here**

Any potential customer with a package link for PWR Components from the support or sales team, then to install PWR Components managed package do follow the mentioned steps here:

1. Open the installation URL received from our sales or support team in the browser.
2. Enter your credentials for the Salesforce organization in which you want to install the package and then click on **Log In**.
3. Repeat the steps as mentioned above from step 5.

## V.II Pre-requisites

As a prerequisite, the Salesforce admin needs to make sure that the Salesforce org is already set up with their domain name.

In case your org is not set up with the Domain Name then the contents of the Package will not be available for use. To enable the domain name open Setup from the Quick Find box, find "My Domain" and Set up a Domain Name for your org. For more details and information, please click here

# V.III Step by Step Walk through

## V.III.1 PWR Button Group

- **Introduction:**

  PWR Button is used for calling any actions on the Lightning UI. PWR Buttons can be placed anywhere on the Lightning Pages. It looks like the button group in Salesforce. The advantage of using PWR Buttons is that they can be placed anywhere on the Pages, making it easily accessible.

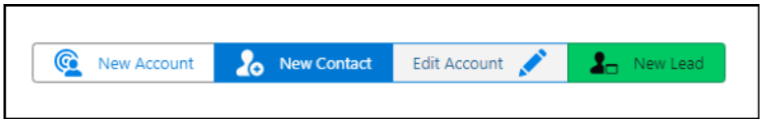


Figure 1: PWR Button Group

PWR Button supports no of Actions like Quick Actions, Global Actions, Redirect, and many more. All the actions are supported on the Home Page, App Page, Record Page, flows, and Community pages as well.

- **Steps to create Button Group config:**
  - Go to the designer tab

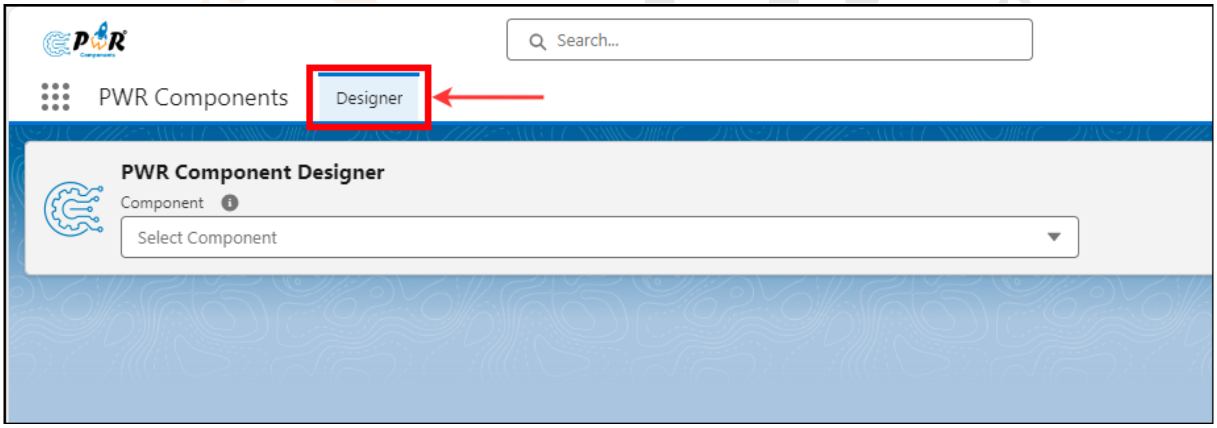


Figure 2: Designer

– Select PWR Button Group component from the list of components

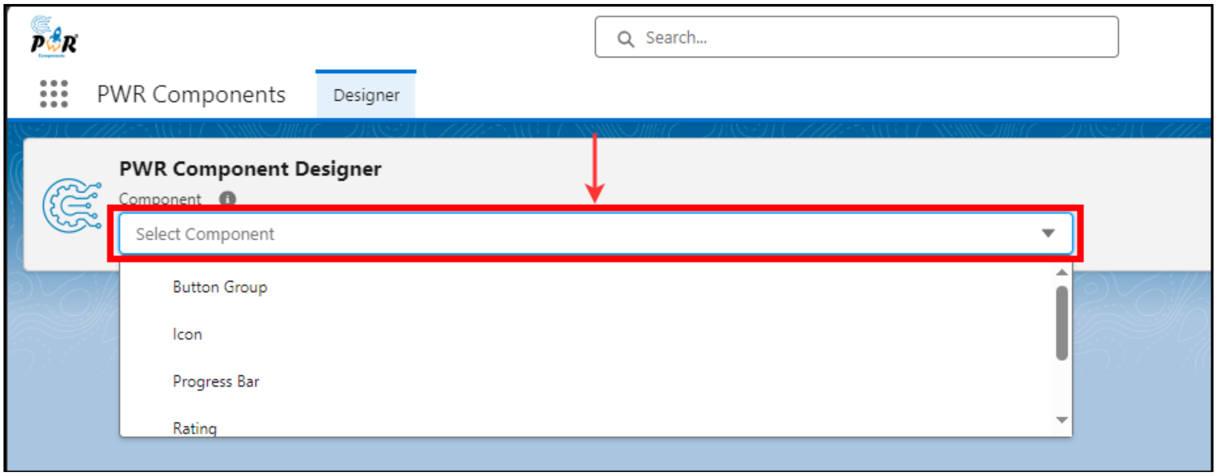


Figure 3: Select Component

– Users can use existing Config, or users can create a new Config

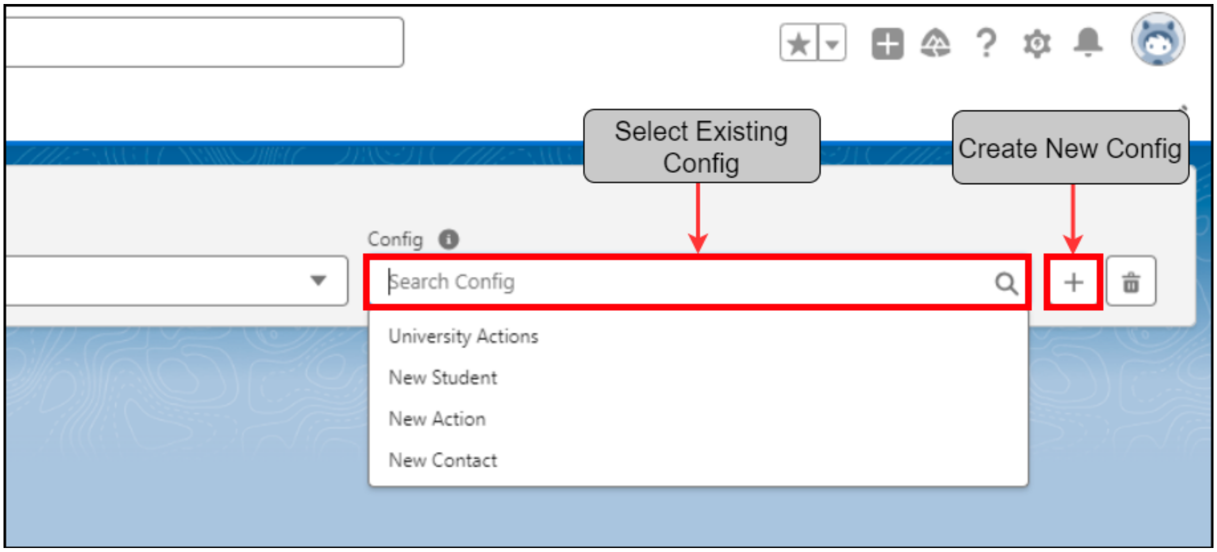


Figure 4: Select Config

– If the user creates a new config

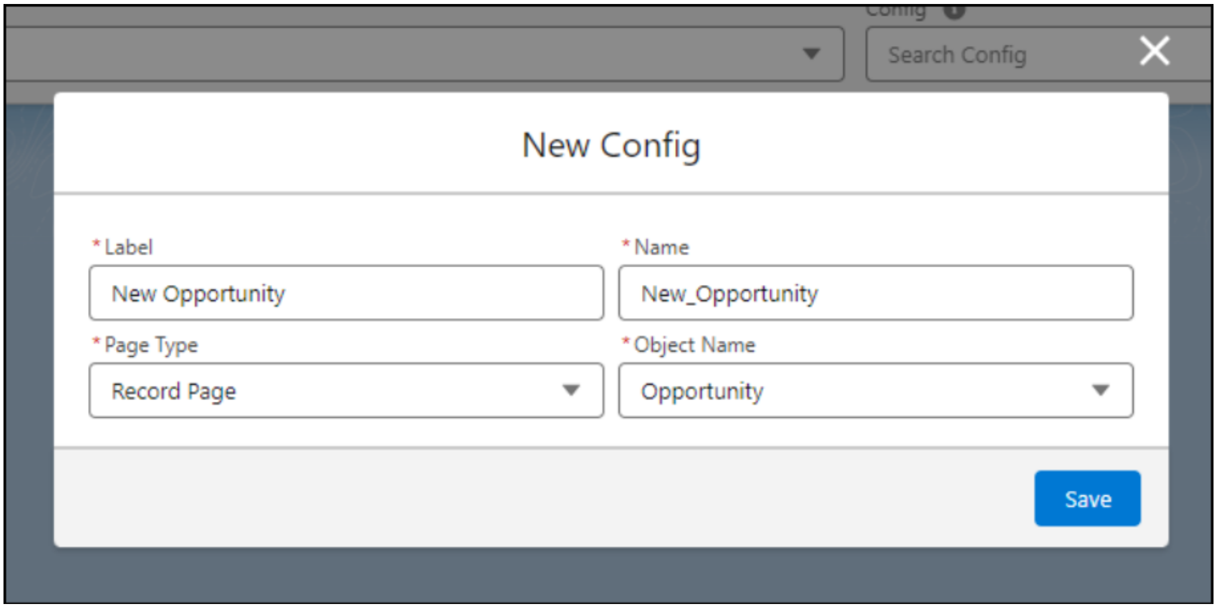


Figure 5: Create Config

* **Label:** Here user can set the title for the config of the component
* **Name:** The API name of the label is shown here
* **Page Type:** Here user can select page type as record page or non-record page; depending on which page the user wants to show the component
  - Non-Record Page: By choosing the non-record page user can use this config on the home page and app page
  - Record Page: By choosing the record page user can use this config on a related record object
* **Object Name:** Here user can choose any object on which that created config will be shown

**Here are the details around different attributes, and which valid information can be input for the same.**

- **Attributes**

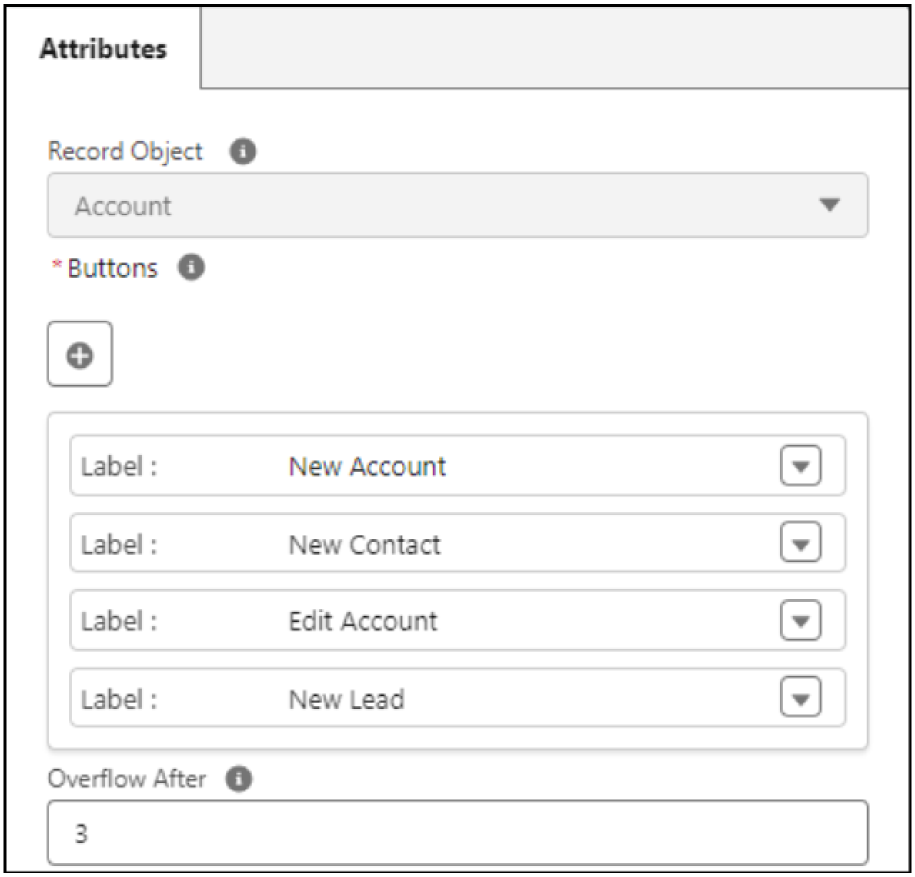


Figure 6: Configuration Section

– **Record Object:** The object name is auto-populated here as you select the object at the time of config creation

– **Buttons:** Here user can add multiple buttons

– **Button Type:** Here user can select the type of button to configure; by default button type is selected

  ⋆ **Button** Here user can add buttons to configure actions

  ⋆ **Header** Providing a distinct title or label for the button group facilitates the categorization of buttons

  ⋆ **Separator** Visual element creating a visible division between buttons, enhancing clarity and categorization within the group

  ⋆ **Submenu** Secondary buttons nested within the main button group, offering additional options or actions related to the primary buttons

  ⋆ **Label:** The label is displayed on the button

    · **Name:** The name attribute specifies the name for the Button

    · **Icon:** You can add the icon to the button

- Left Icon: Displays the icon on the left side of the action label
- Right Icon: Displays the icon on the right side of the action label

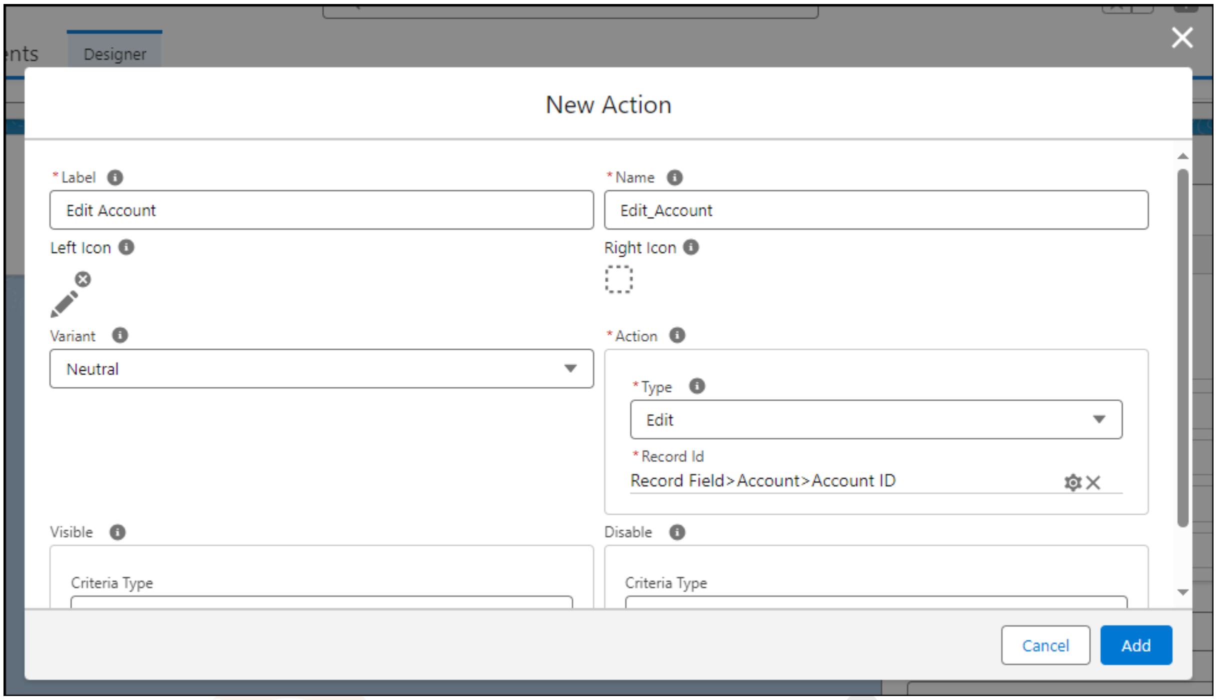


Figure 7: New Button Creation

* **Variant:** Allows to show Buttons in different color formats
  - Neutral: Neutral is the default variant of the Button which is displayed in white color
  - Brand: Brand is a blue Button, used to draw attention to the primary action on a page
  - Outline-brand: Outline-brand is like the brand variant, but the color is used for the label and border only, not the Button color
  - Destructive: Destructive is a red Button used to warn that the action has a negative effect
  - Success: Success is a green Button used to indicate a successful action
  - Text Destructive: This button is used to indicate a destructive action to the user

* **Action:** User can create the actions here. For example, creating new records, navigating to any URL, editing any record, etc

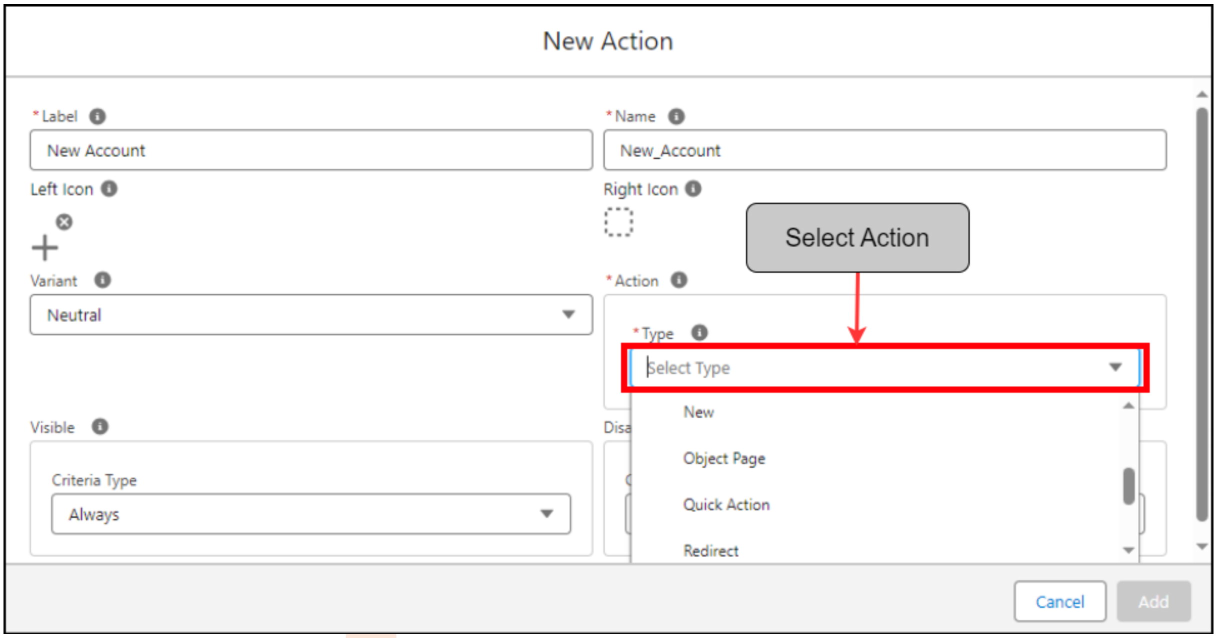


Figure 8: Select Action

- **Type:** Here user can select any specific type for that particular action

- **Edit:**This Action is a feature that allows a user to create a Button for Editing or Modifying an existing Record in the system. The process involves the following steps:

  1. The user fills out the necessary fields such as Label and Name
  2. The user selects 'Edit' from the Action Type picklist
  3. The user has to provide the Record ID of the Record that needs to be edited or modified. This ID can be passed through various methods such as Custom Metadata, Custom Setting, Record Field, or Static which can be selected by the User from the Picklist named as 'Value Source' which appears after the Selection of Action Type
  4. Once these steps are completed, the button is created. When clicked, this button fetches the corresponding record using the provided ID and allows the user to make changes to it

- **New:** This Action is a feature that allows a user to create a Button for generating a New Record for a specific Object in the system. The process involves the following steps:

  1. The user fills out the necessary fields such as Label and Name
  2. The user selects 'New' from the Action Type picklist

3. The user Selects the name of the Object for which a new Record needs to be created from the Object Picklist which has appeared after choosing the Action type as 'New'
4. Once these steps are completed, the button is created. When clicked, this button initiates the process of creating a new record for the specified object.

· **View:** This Action is a feature that allows a user to create a button for viewing an existing record in the system. The process involves the following steps:

1. The user fills out the necessary fields such as Label and Name
2. The user selects 'View' from the Action Type picklist
3. The user has to provide the Record ID of the Record that needs to be Viewed. This ID can be passed through various methods such as Custom Metadata, Custom Setting, Record Field, or Static which can be selected by the User from the Picklist named as 'Value Source' which appears after the Selection of Action Type
4. Once these steps are completed, the button is created. When clicked, this button fetches the corresponding record using the provided ID and allows the user to view it

· **Delete:** This Action is a feature that allows a user to create a button for Deleting an existing record in the system. The process involves the following steps:

1. The user fills out the necessary fields such as Label and Name
2. The user selects 'Delete' from the Action Type picklist
3. The user must provide the Record ID of the Record that needs to be Deleted. This ID can be passed through various methods such as Custom Metadata, Custom Setting, Record Field, or Static which can be selected by the User from the Picklist named as 'Value Source' which appears after the Selection of Action Type
4. Once these steps are completed, the button is created. When clicked, this button fetches the corresponding record using the provided ID and Deletes it from the system

· **Redirect:** This Action is a feature that allows a user to create a button for Redirecting to a different website or page. The process involves the following steps:

1. The user fills out the necessary fields such as Label and Name
2. The user selects 'Redirect' from the Action Type picklist
3. The user provides the URL of the website or page to which they want to Redirect. This can be done by entering or pasting the URL in the designated field, which appears after the Selection Action Type
4. If needed, the user can add custom parameters by clicking on 'Add parameters', which helps to Redirect to a specific Page on the Website,

is Redirected

5. Once these steps are completed, the button is created. When clicked, this button Redirects the user to the specified URL

- **Global Action:** This Action is a feature that allows a user to create a button for executing an existing Global Action in the system. The process involves the following steps:

1. The user fills out the necessary fields such as Label and Name
2. The user selects 'Global Action' from the Action Type picklist
3. The user Selects the name of the Global Actions which is to be used or executed, from Picklist which has appeared after choosing the action type as 'Global Action'
4. Once these steps are completed, the button is created. When clicked, this button executes the specified Global Action. This functionality ensures a streamlined process for utilizing existing Global Actions within the system

- **Quick Action:** This Action is a feature that allows a user to create a button for executing an existing Quick Action related to a specific Object in the system. The process involves the following steps:

1. The user fills out the necessary fields such as Label and Name
2. The user selects 'Quick Actions' from the Action Type picklist
3. The user selects the name of the Object from the picklist whose Quick Action is to be executed
4. After selecting the Object, the user then selects the specific Quick Action of that object which is to be executed from the picklist that appears
5. Once these steps are completed, the button is created. When clicked, this button executes the specified Quick Action related to the chosen Object. This functionality ensures a streamlined process for utilizing existing Quick Actions within the system

- **List View:** This Action is a feature that allows a user to create a button for redirecting/ Opening a List View of an Object in the system, on the new Tab/Page. The process involves the following steps:

1. The user fills out the necessary fields such as Label and Name
2. The user selects 'List View' from the Action Type picklist
3. The user provides the name of the Object related to the List View that needs to be displayed. This can be done by selecting the desired Object from the picklist
4. After selecting the object, the user has to provide the Record ID of the List View in the Filter Name field for that Object's List View. The default filter is 'Recent', but users can select other filters from options such as Custom Metadata, Custom Setting, Record Field, or Static

5. Once these steps are completed, the button is created. When clicked, this button redirects the user to the specified List View related to the chosen Object with the selected filter applied

- **Tab:** This action is a feature that allows a user to create a button for redirecting to a specific Tab related to an Object in the system. The process involves the following steps:

  1. The user fills out the necessary fields such as Label and Name
  2. The user selects 'Tab' from the Action Type picklist
  3. The user provides the name of the Object whose Tab needs to be displayed. This can be done by selecting the desired Object from the picklist
  4. Once these steps are completed, the button is created. When clicked, this button redirects the user to the specified Tab related to the chosen Object

- **Object Page:** This action is a feature that allows a user to create a button for redirecting to a specific Object Page related to an object in the system, on the same Page/Tab where the button is located. The process involves the following steps:

  1. The user fills out the necessary fields such as Label and Name
  2. The user selects 'Object Page' from the Action Type picklist
  3. The user provides the name of the object whose Object Page needs to be displayed. This can be done by selecting the desired object from the picklist
  4. Once these steps are completed, the button is created. When clicked, this button redirects the user to the specified Object Page related to the chosen object. This functionality ensures a streamlined process for navigating and viewing specific sections of the system

- **Custom Component:** This action is a feature that allows a user to create a button for opening or redirecting to a Custom Component present in the system. The custom component could be a Visualforce Component, Lightning Component, etc. The process involves the following steps:

  1. The user fills out the necessary fields such as Label and Name
  2. The user selects 'Custom Components' from the Action Type picklist
  3. The user provides the name of the Custom Component that needs to be opened or redirected to. This can be done by entering the name of the Custom Component in the designated field
  4. Once these steps are completed, the button is created. When clicked, this button opens or redirects the user to the specified Custom Component. This functionality ensures a streamlined process for navigating and accessing specific custom components within the system.

· **Visible:**

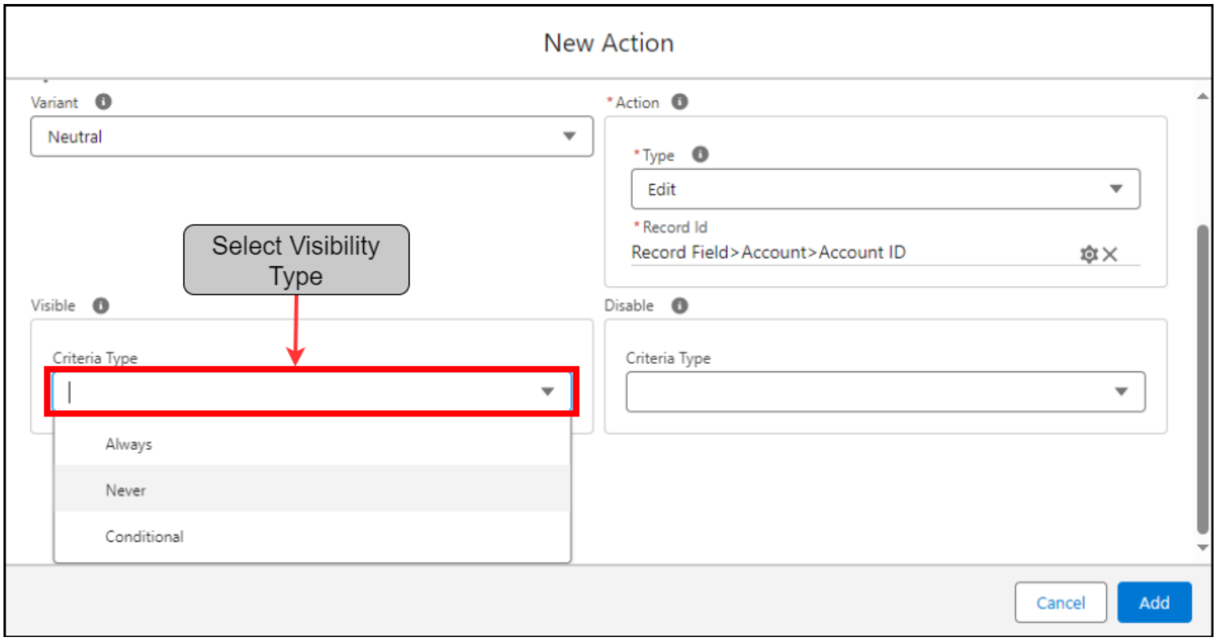


Figure 9: Select Visibility Type

· Never: The button is hidden
· Always: The button is always visible
· Conditional: Conditional visibility specifies when a Button is visible
· **Disable:**

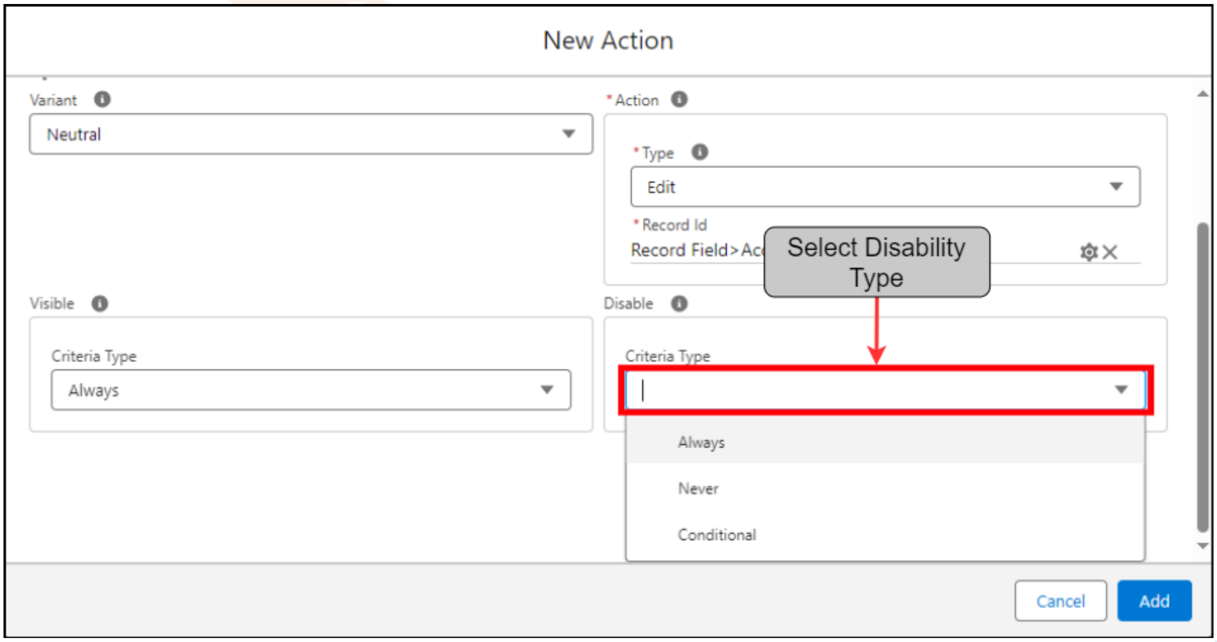


Figure 10: Select Disability Type

· Never: The button is visible

- Always: The button is always disabled
- Conditional: Conditional disability specifies when a Button is disabled

* **Overflow After:**

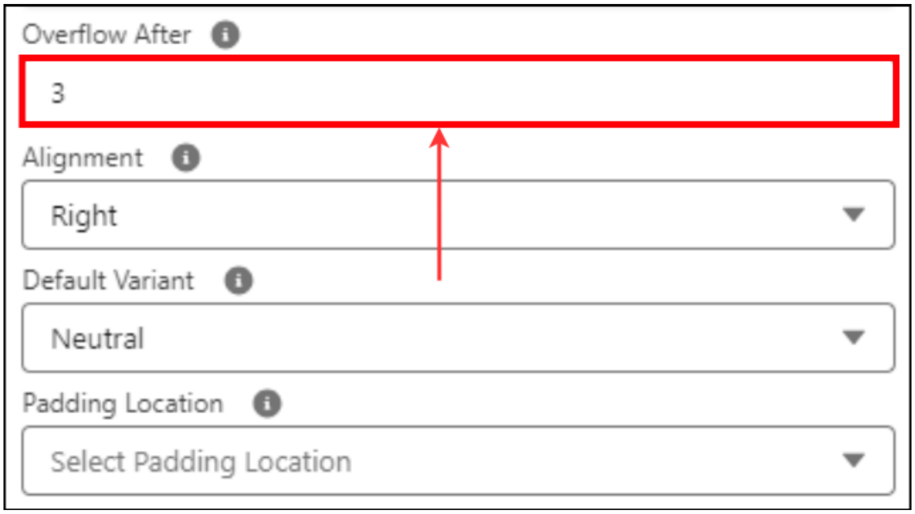


Figure 11: Overflow After

It is a numerical value that determines the number of buttons to be displayed on the page and the number to be displayed in the dropdown. For example, if 'Overflow After' is set to 3, then 3 buttons are displayed on the page and the remaining buttons are displayed in the dropdown

* **Alignment:**

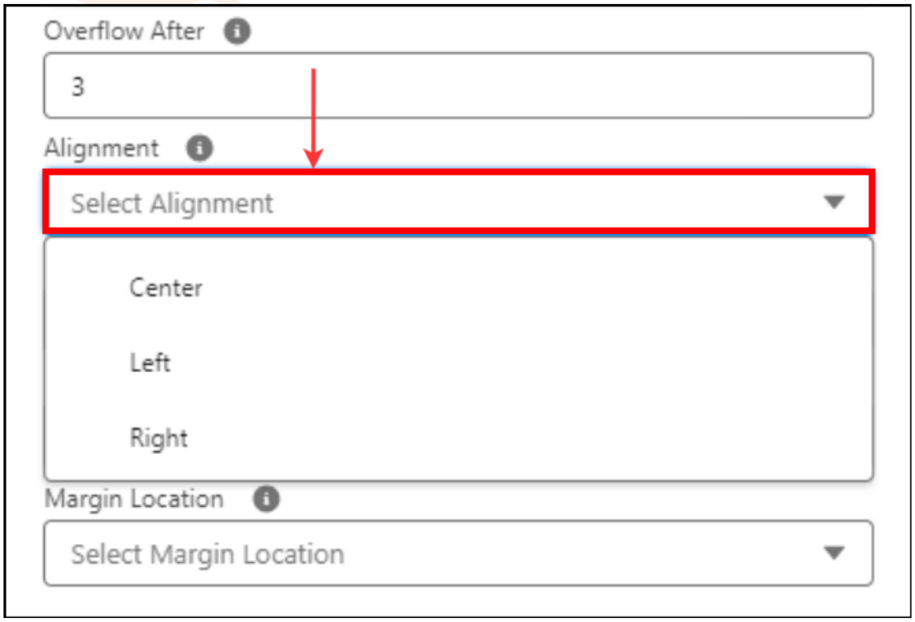


Figure 12: Select Alignment

Here user can select alignment if the buttons

* **Default Variant:**

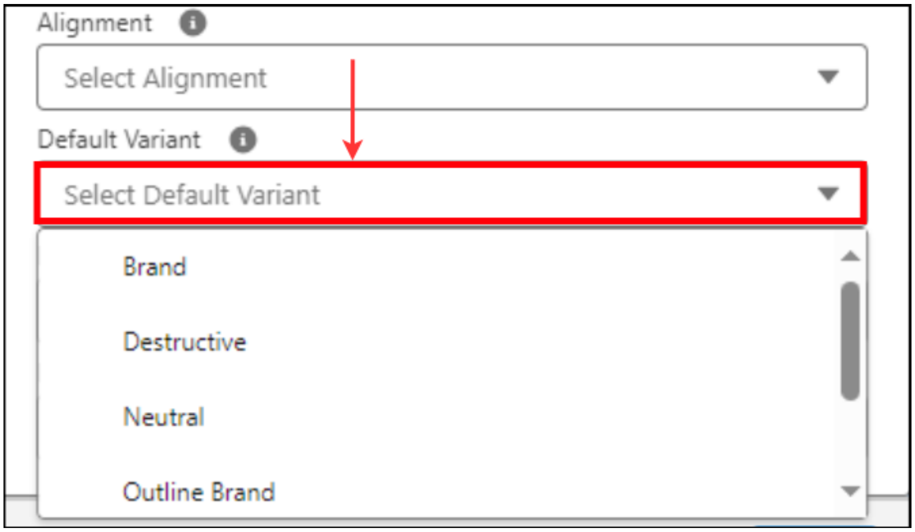


Figure 13: Select Varient

Here user can select the default variant

* **Padding Location**:
Defines the position of the padding for a component. The padding creates extra space within a component.

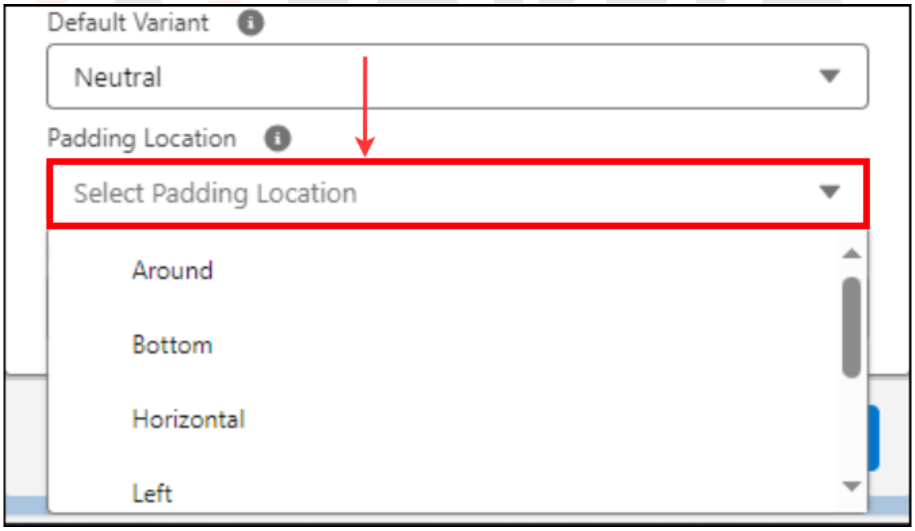


Figure 14: Select Padding Location

- Around: Creates padding around the component
- Top: Creates padding at the top of the component
- Left: Creates padding at the left side of the component
- Bottom: Creates padding at the bottom of the component
- Right: Creates padding at the right side of the component
- Horizontal: Creates padding horizontally

- Vertical: Creates padding vertically

* **Padding Size:** The padding size of the component can be set to:

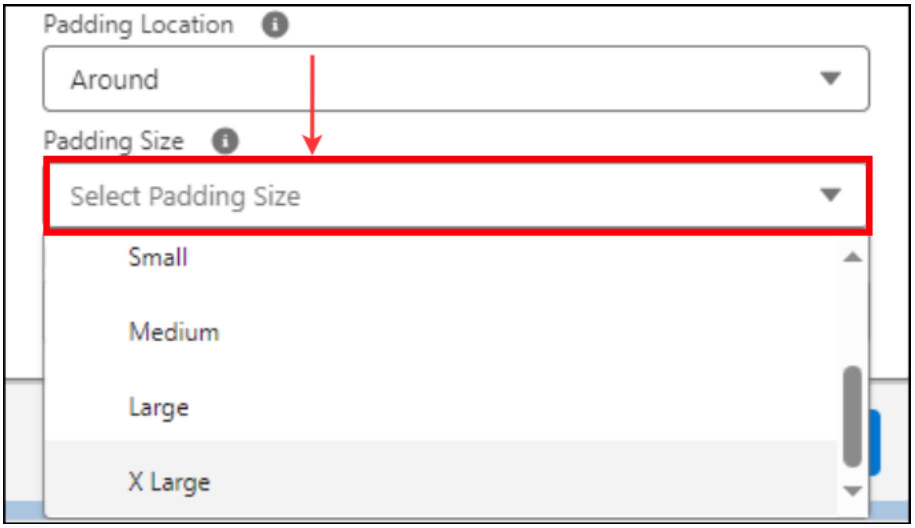


Figure 15: Select Padding Size

- None
- xxx-small
- xx-small
- x-small
- Small
- Medium
- Large
- x-large

* **Margin Location:** Defines the position of the margin for a component. margin creates extra space around a component.

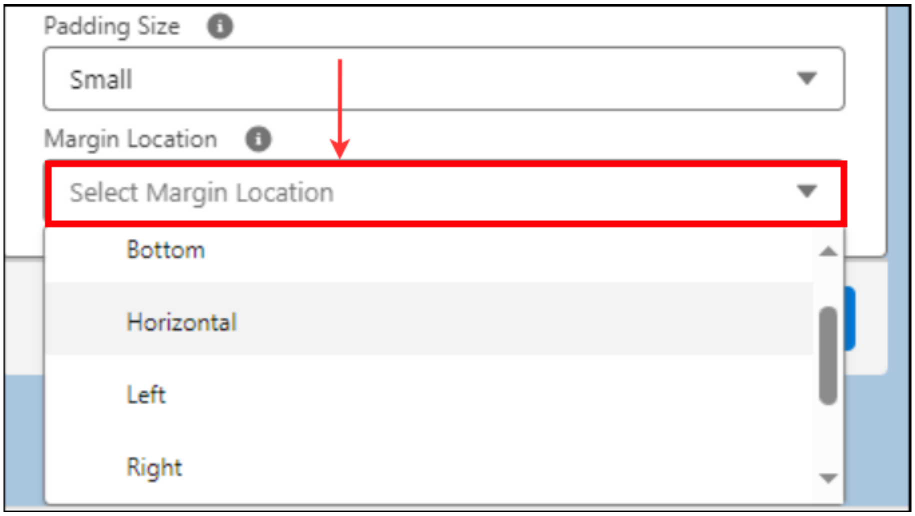


Figure 16: Select Margin Location

- Around: Creates padding around the component
- Top: Creates padding at the top of the component
- Left: Creates padding at the left side of the component
- Bottom: Creates padding at the bottom of the component
- Right: Creates padding at the right side of the component
- Horizontal: Creates padding horizontally
- Vertical: Creates padding vertically

* **Margin Size:** The margin size of the view can be set to:


Figure 17: Select Margin Size

- None
- xxx-small
- xx-small
- x-small
- Small
- Medium
- Large
- x-large

After clicking on the Save button the config will get saved on the component config object and the user can view this button on the Record page, App page, Home Page, Flows, and Community.

– **Preview Section:**

After the user has selected and filled the fields according to their requirements, they can preview the components in the Preview section before saving them on the Record Page, App Page, Flows, and Community Page.

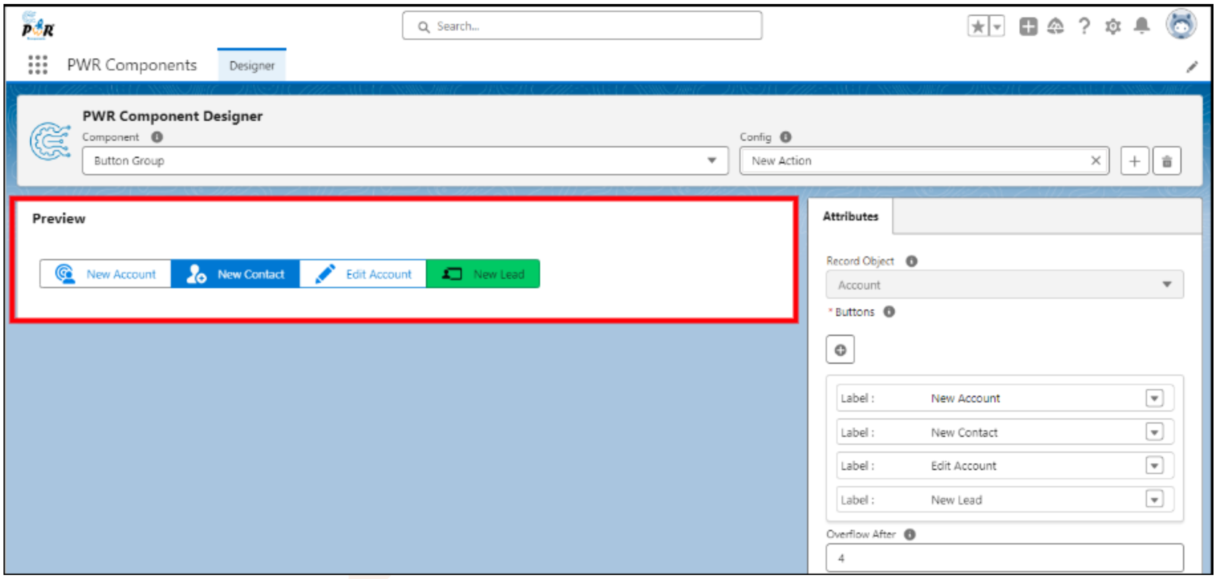


Figure 18: Preview Section

– **PWR Button Group screenshots:**

* PWR Button Group on Home Page:

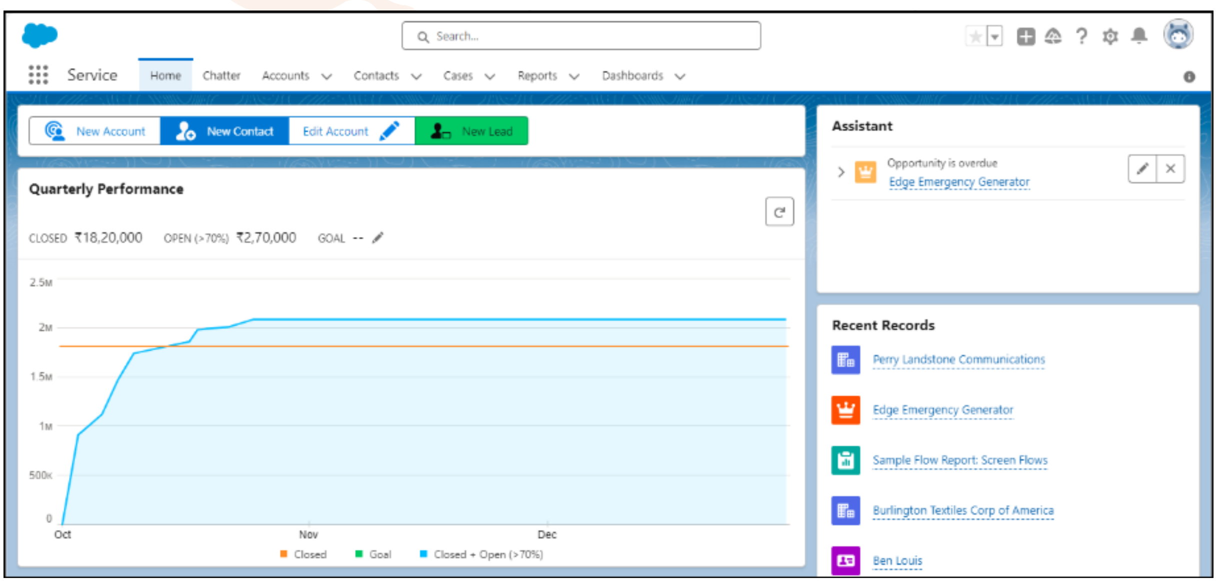


Figure 19: PWR Button Group on Home Page

* PWR Button Group on Record Page

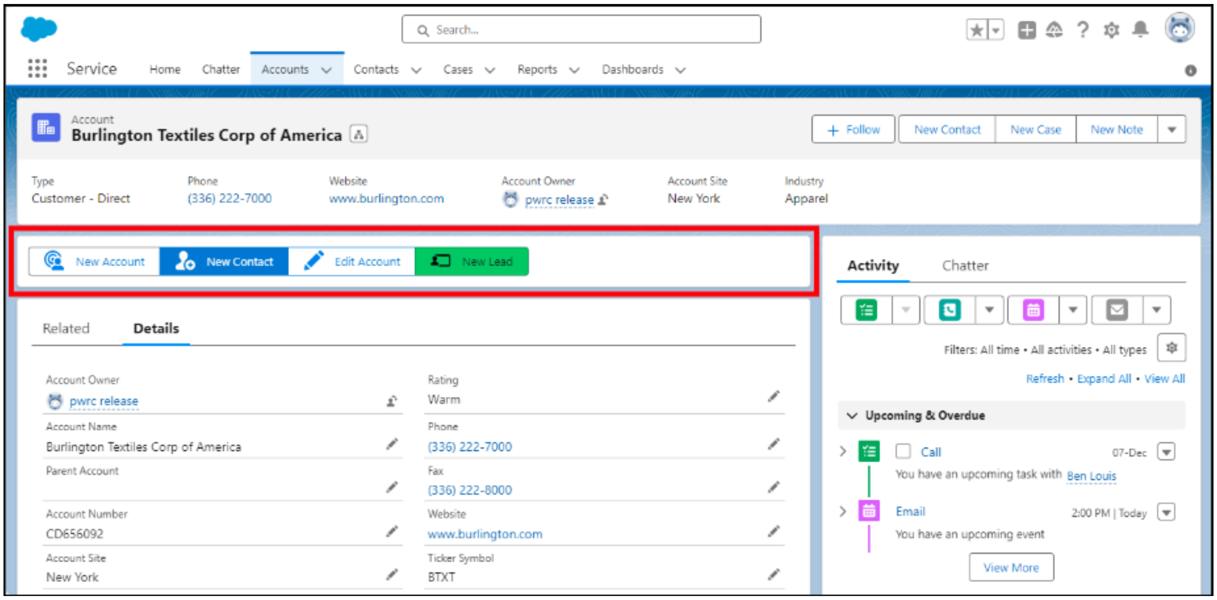


Figure 20: PWR Button Group on Record Page

## V.III.2 PWR Icon:

- **Introduction:**

  The Icon is a symbolic image that helps users quickly and easily identify the content. There are various types of categories in Icons. In that category, the user can select one icon.

Figure 21: PWR Icon

- **Steps to create icon config:**

  * Go to the designer tab

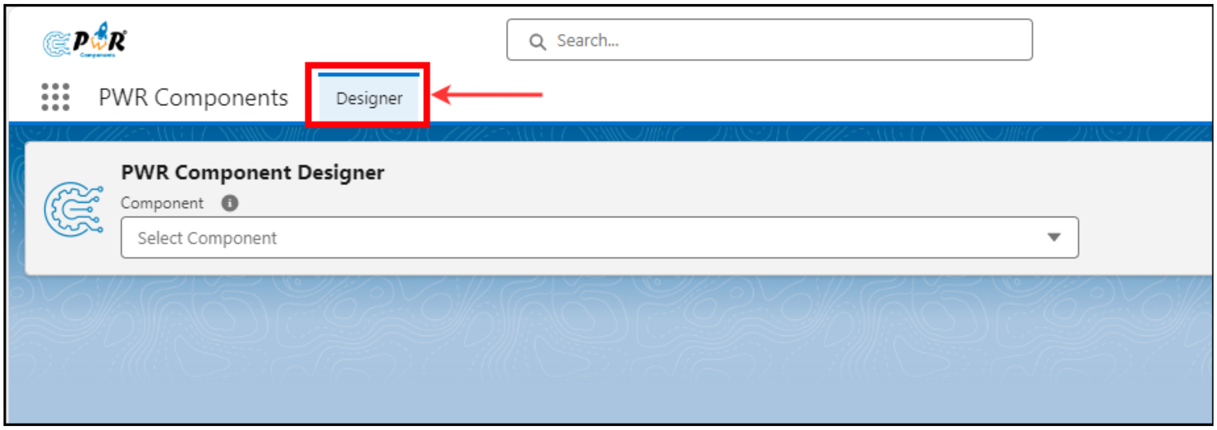


Figure 22: Designer

  * Select the Icon component from the list of components

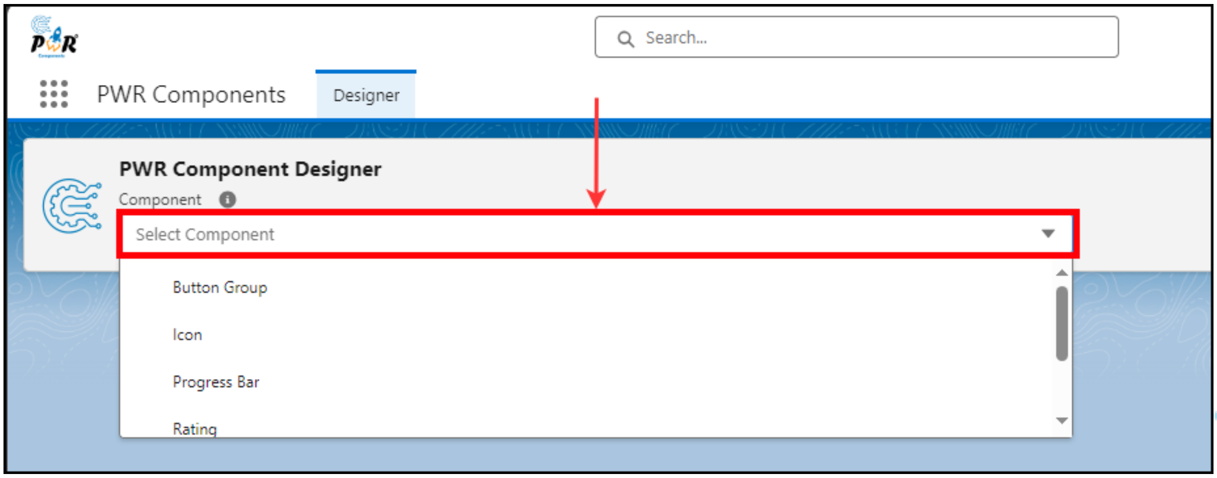


Figure 23: Select Component

* Users can use existing Config, or users can create a new Config

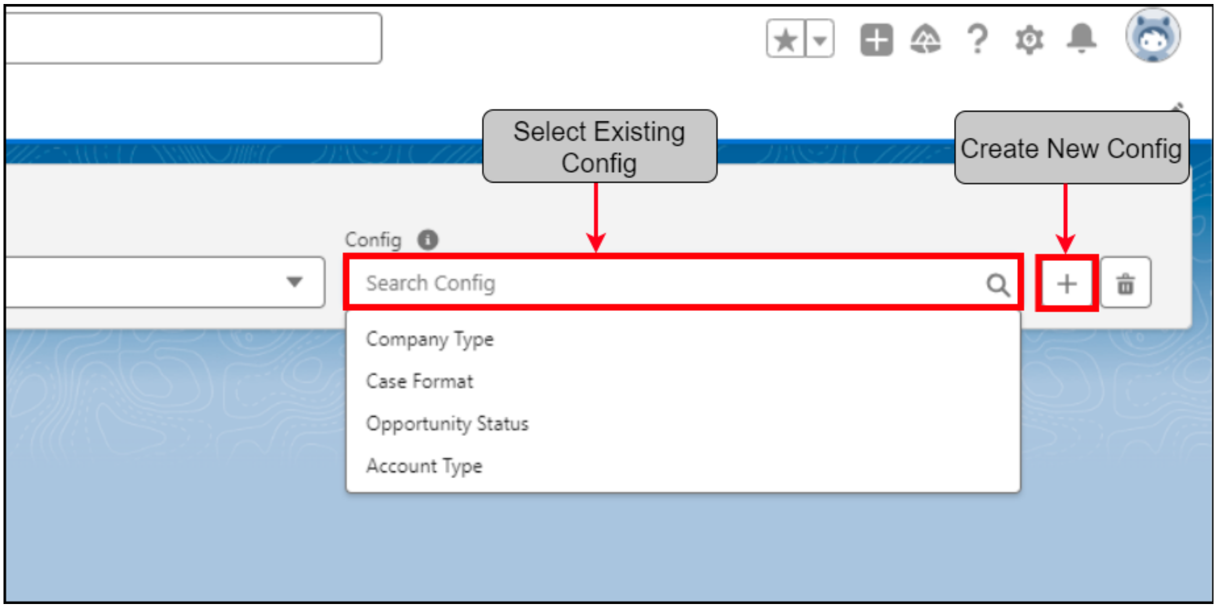


Figure 24: Select Config

* If the user creates a new config

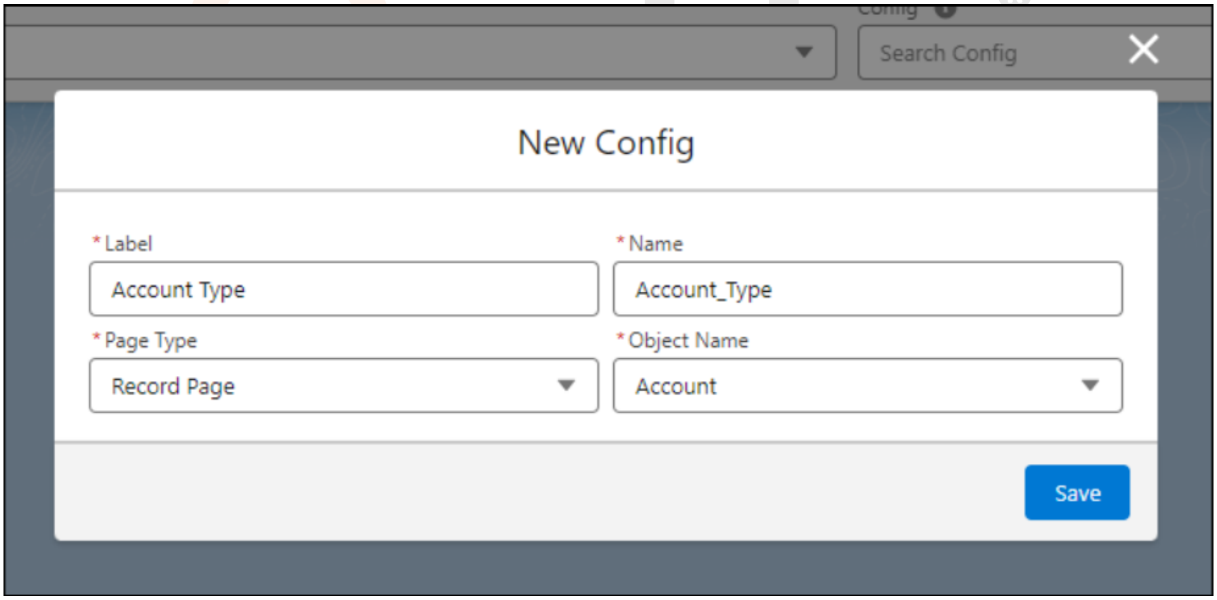


Figure 25: Create Config

- **Label:** Here user can set the title for the config of the component
- **Name:** The API name of the label is shown here
- **Page Type:** Here user can select page type as record page or non-record page; depending on which page the user wants to show the component

- Non-Record Page: By choosing the non-record page user can use this config on the home page and app page
- Record Page: By choosing the record page user can use this config on a related record object
- **Object Name:** Here user can choose any object on which that created config will be shown

**Here are the details around different attributes, and which valid information can be input for the same.**

– **Attributes:**

* **Record Object:** The object name is auto-populated here as you select the object at the time of config creation
* **Record To Preview:** This feature helps you preview and evaluate the icon's appearance according to the selected record

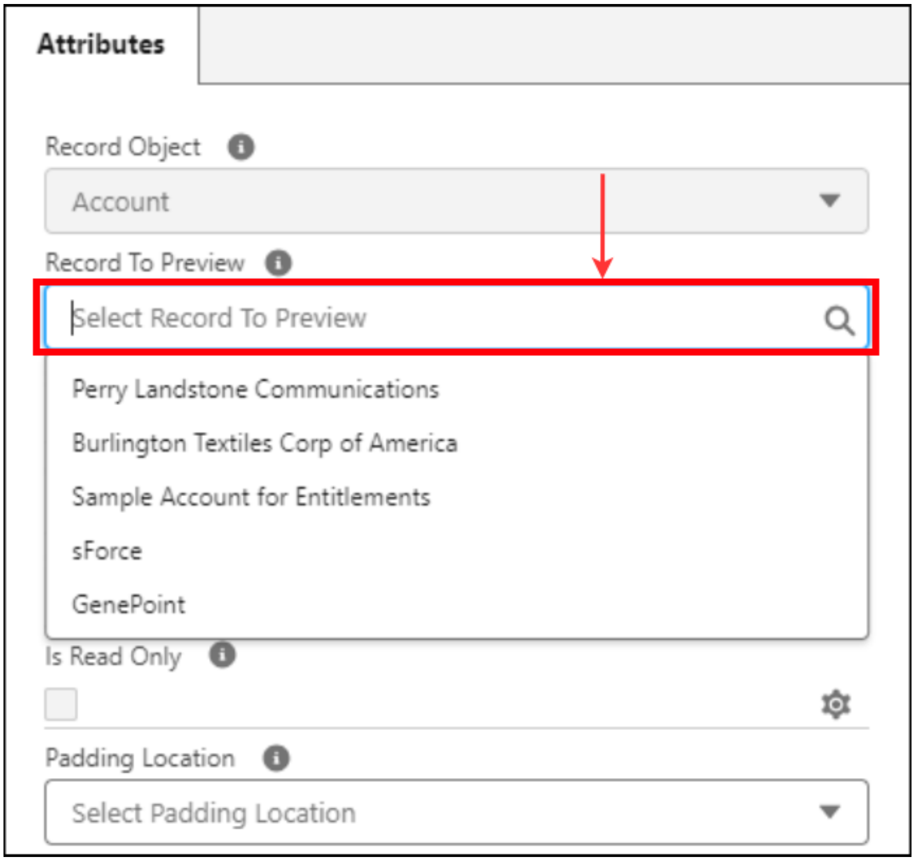


Figure 26: Record To Preview

* **Select Icon:** Here user can select the icon, click on the gearbox (icon configuration)

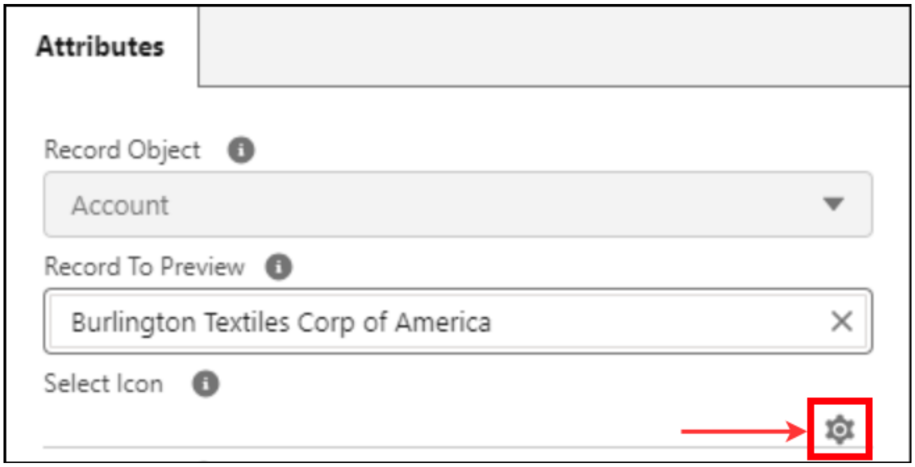


Figure 27: Icon Configuration

- **Value Source:** Here user can select a value source to select an icon and by default value source selected is static

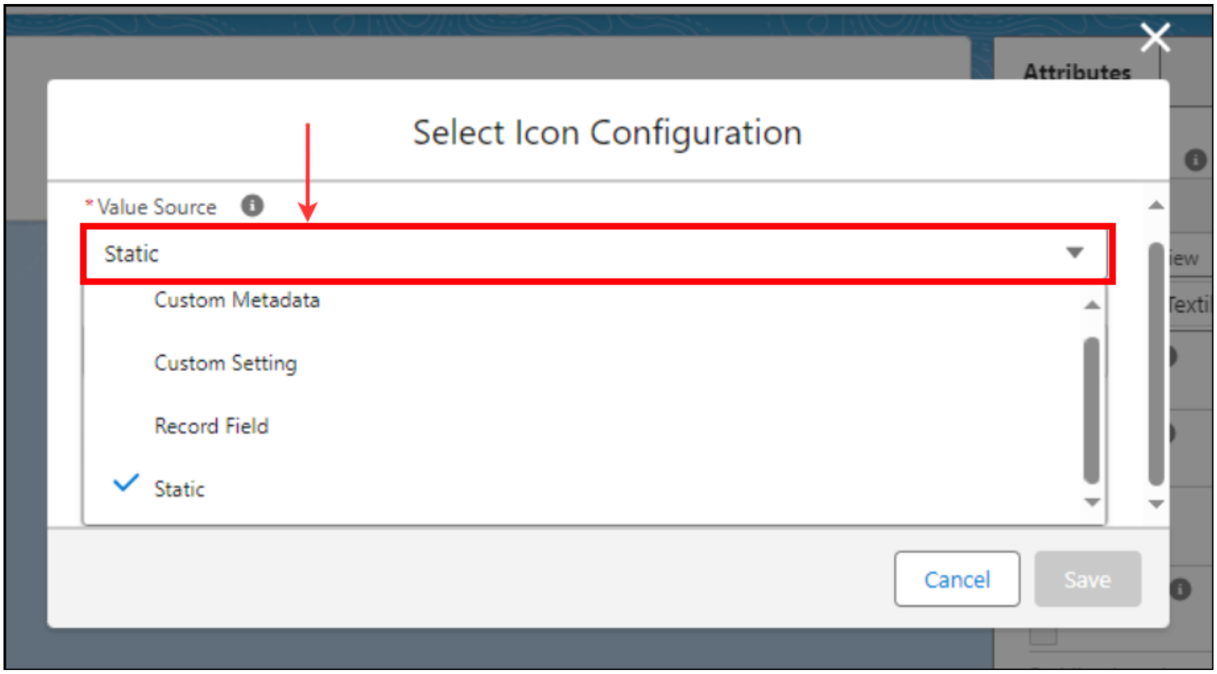


Figure 28: Value Source Configuration

- **Value Type:** This is a read-only field that shows the data type of the field for the icon
- **Value:** Here user can select the icon and the selected icon will display here

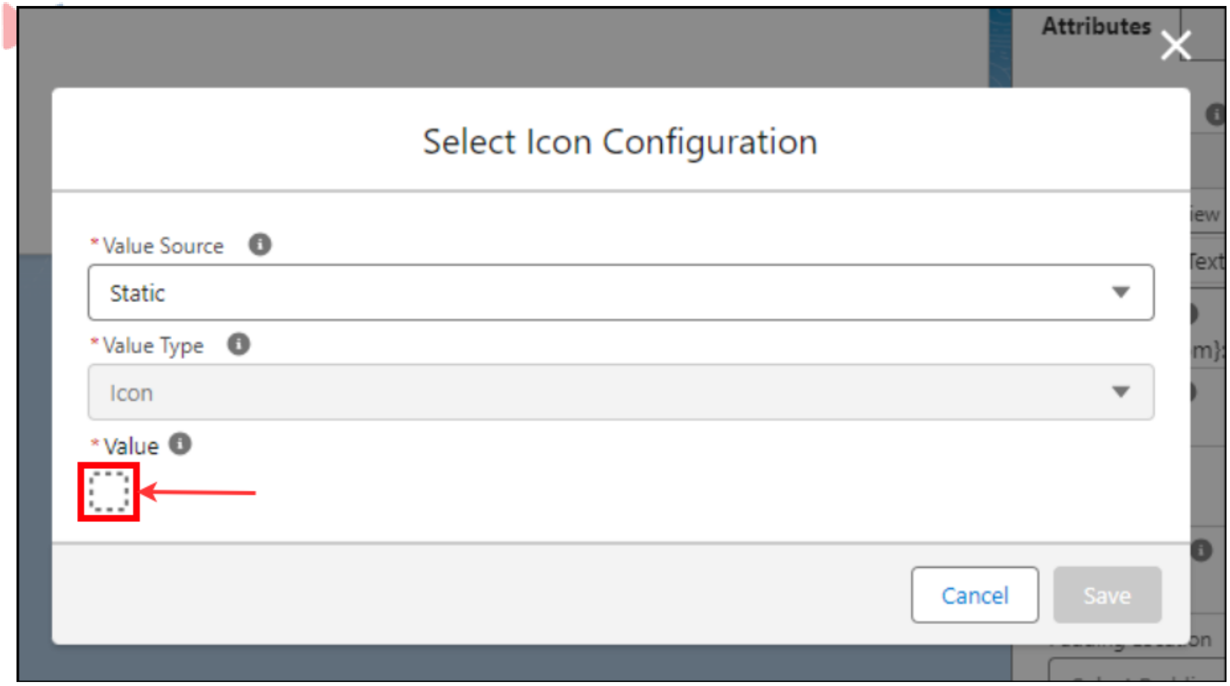


Figure 29: Value Configuration

* **Categories:** Here user can select a category for the icon, click on the gearbox (icon configuration)

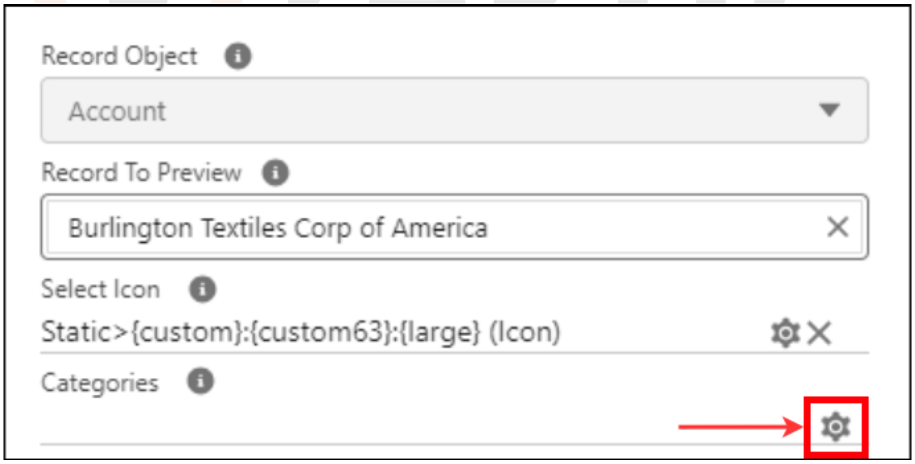


Figure 30: Categories Configuration

- **Value Source:** Here user can select a value source for Category and by default value source selected is static

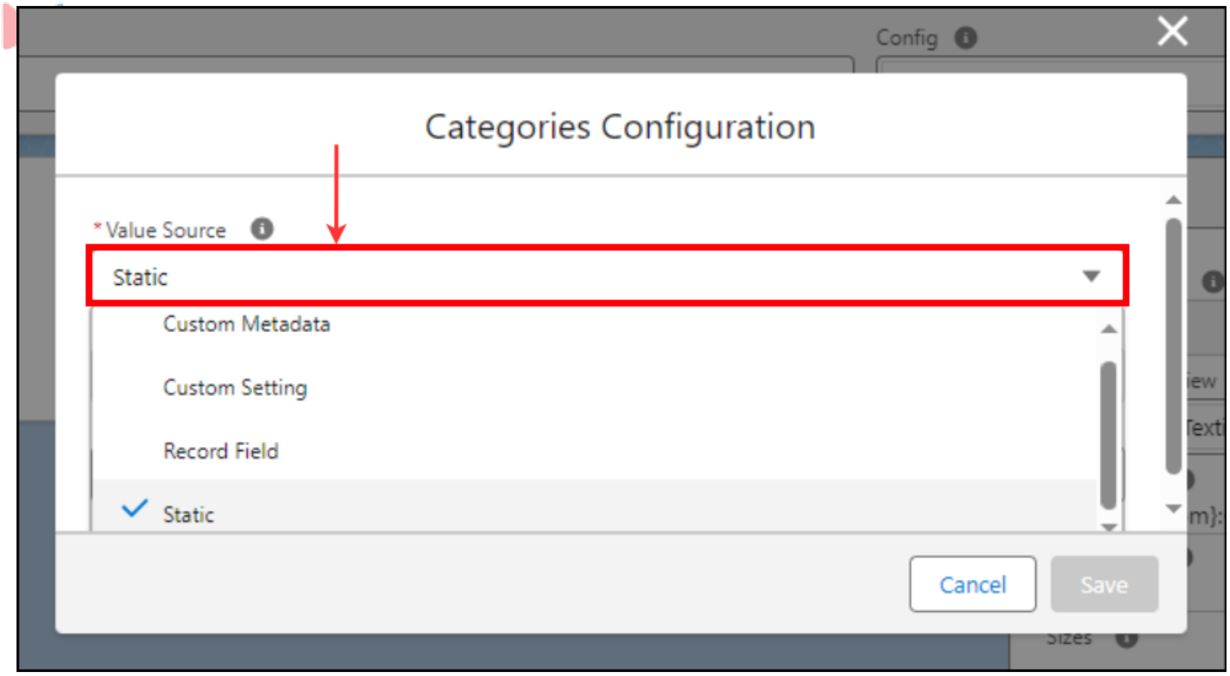


Figure 31: Value Source Configuration

- **Value Type:** This is a read-only field that shows the data type of field for the category
- **Value:** Here user can select the category and the selected category will display here

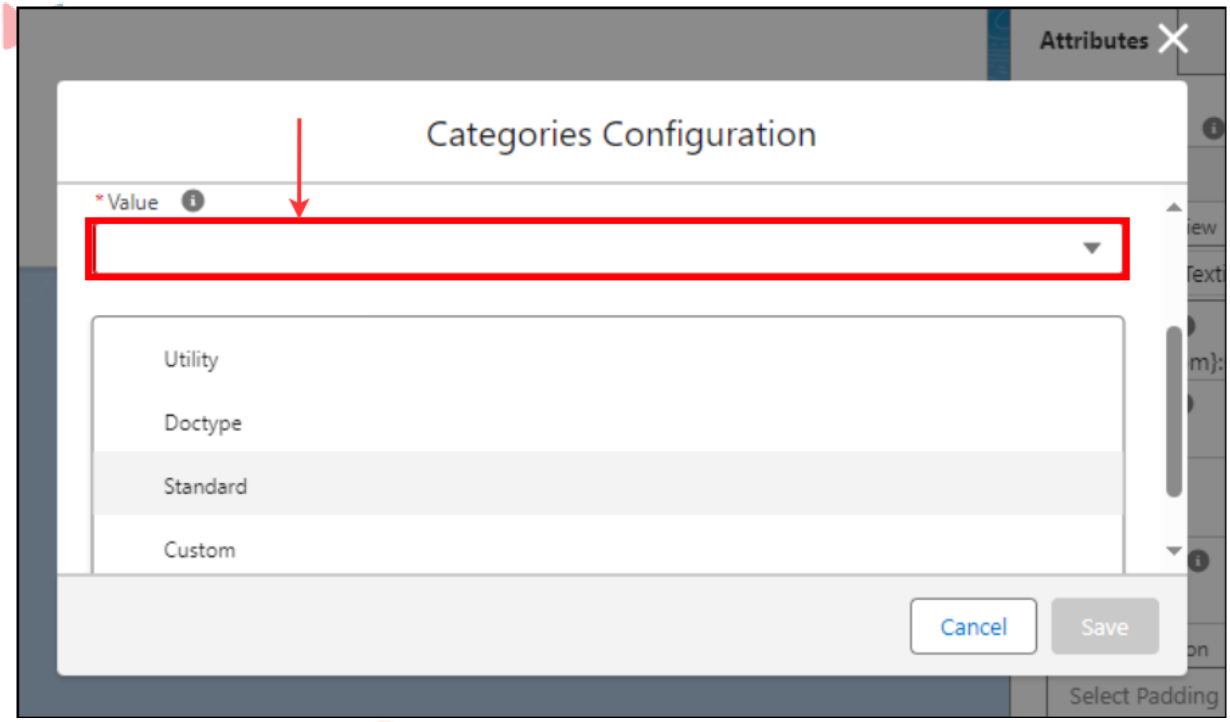


Figure 32: Categories Value Configuration

- Utility
- Doctype
- Standard
- Custom
- Action

* **Sizes:** Here user can select the size for the icon selected icon will display, click on the gearbox (icon configuration) here

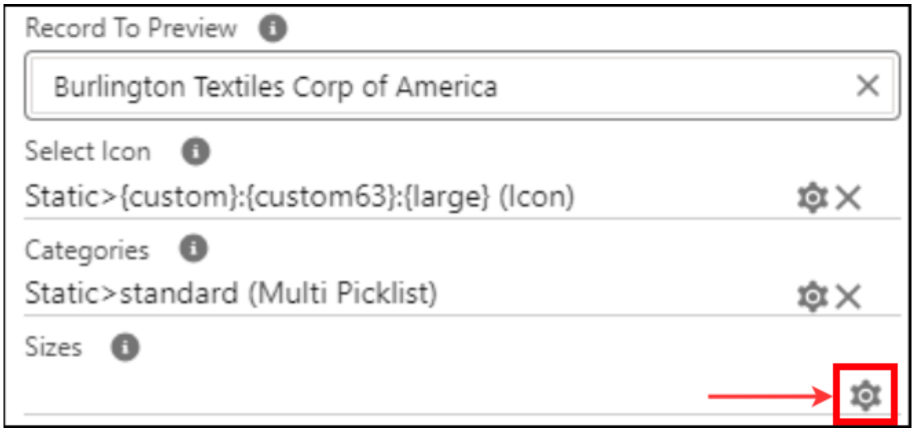


Figure 33: Size Configuration

- **Value Source**: Here user can select a value source for size and by default value source selected is static

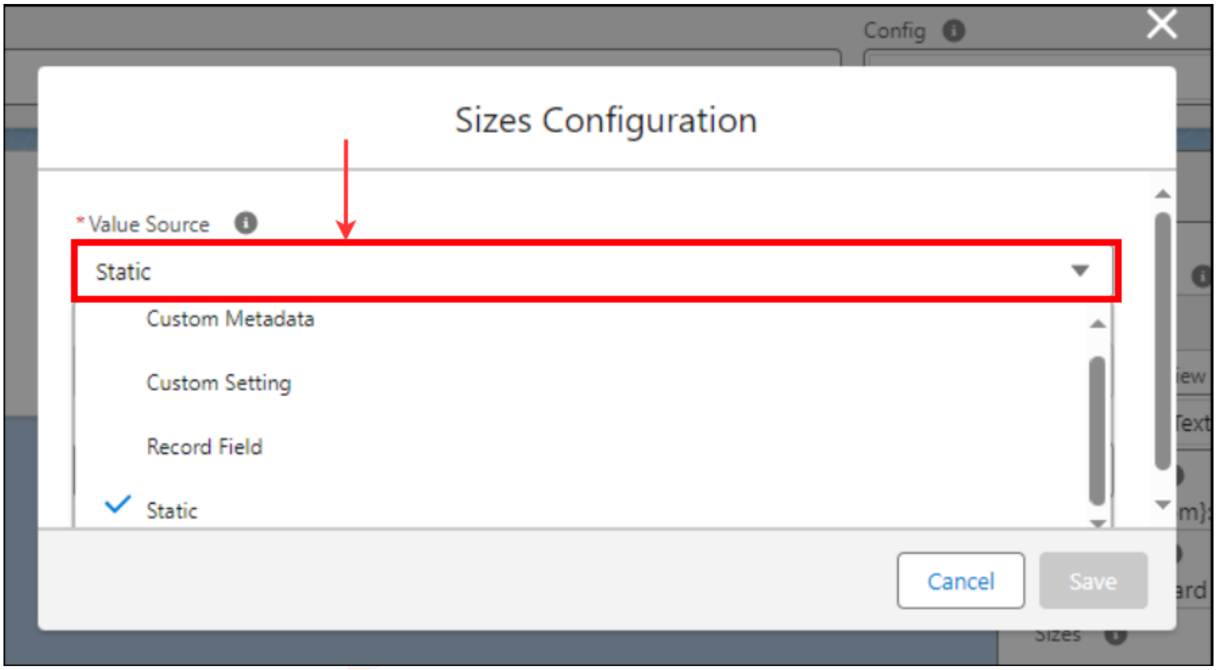


Figure 34: Value Source Configuration

- **Value Type:** This is a read-only field that shows the data type of the field for size
- **Value:** Here user can select the size and the selected size will display here

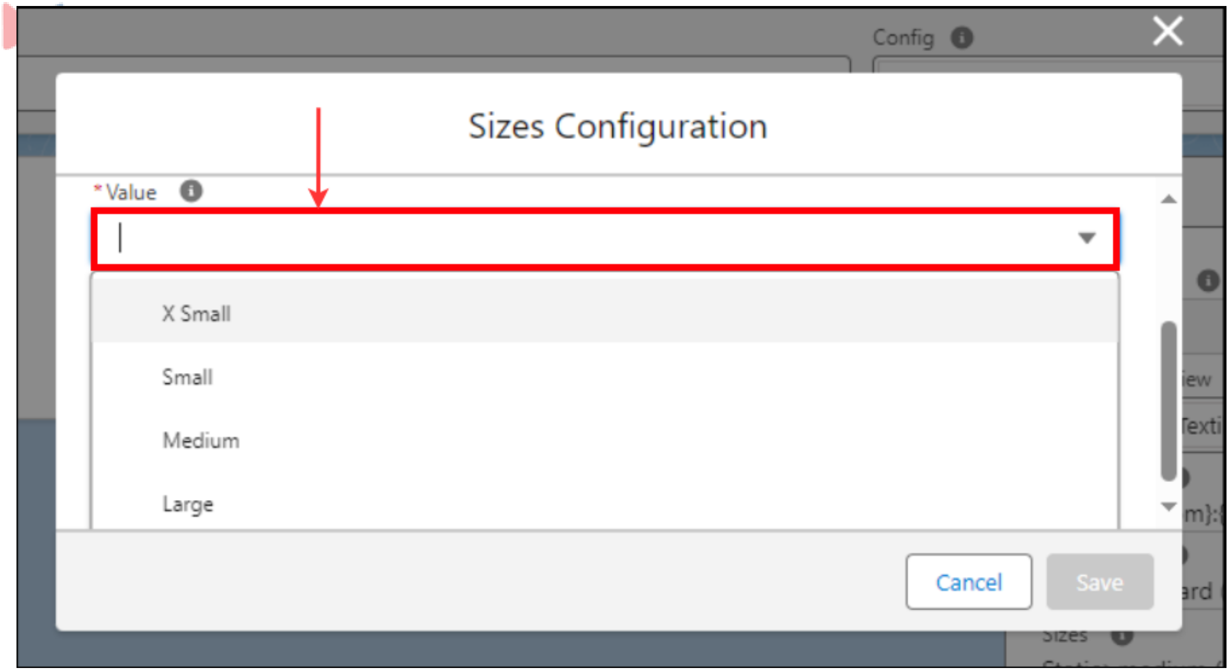


Figure 35: Size Value Configuration

- X Small
- Small
- Medium
- Large

* **Is Read Only:** This field is a checkbox type of field and if the value is checked as true then the user cannot change the value from the page/preview

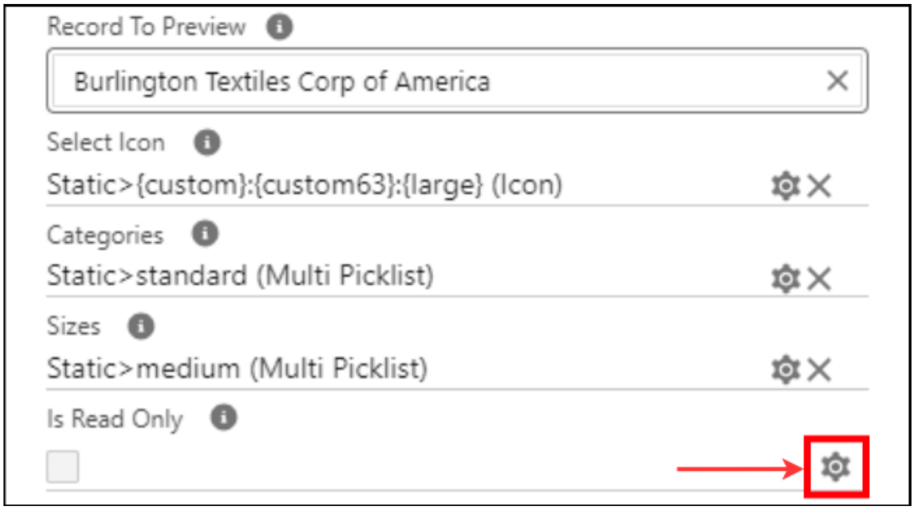


Figure 36: Read Only Configuration

- **Value Source:** Here user can select a value source for is read-only and by default value source selected is static
- **Value Type:** This is a read-only field that shows the data type of field for read-only
- **Value:** Here user can check the checkbox

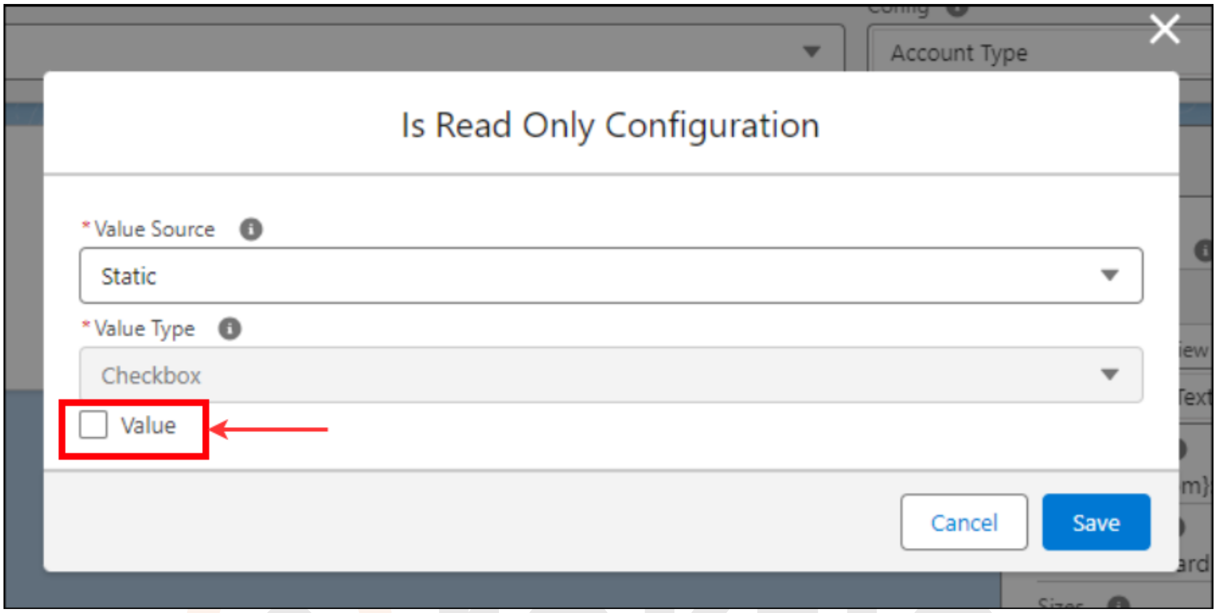


Figure 37: Read Only Value Configuration

* **Padding Location:** Defines the position of the padding for a component. The padding creates extra space within a component

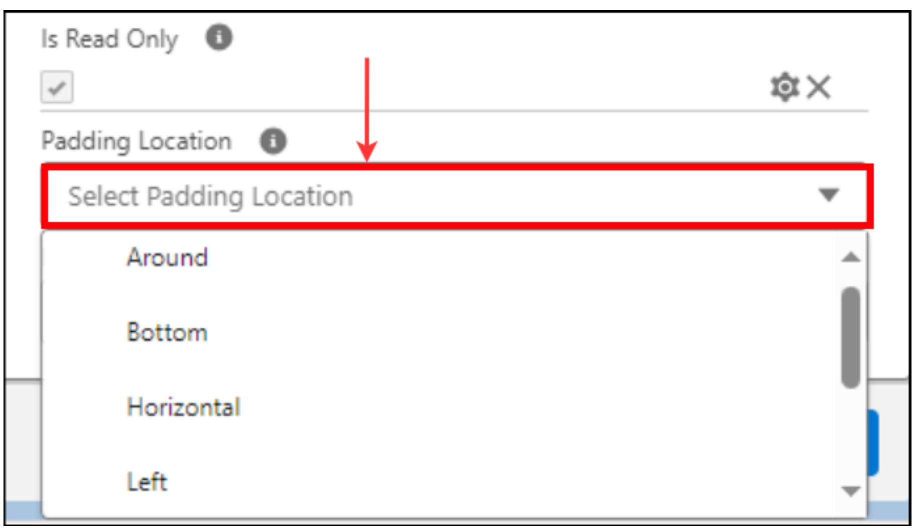


Figure 38: Padding Location Configuration

- **Around:** Creates padding around the component

- **Top:** Creates padding at the top of the component
- **Left:** Creates padding at the left side of the component
- **Bottom:** Creates padding at the bottom of the component
- **Right:** Creates padding at the right side of the component
- **Horizontal:** Creates padding horizontally
- **Vertical:** Creates padding vertically

* **Padding Size:** The padding size of the component can be set to:

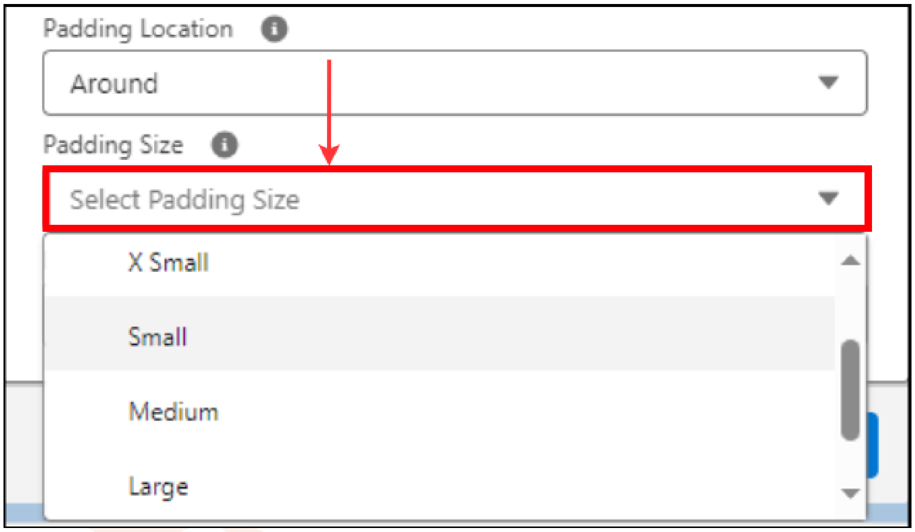


Figure 39: Padding Size Configuration

- None
- xxx-small
- xx-small
- x-small
- Small
- Medium
- Large
- x-large

* **Margin Location:** Defines the position of the margin for a component. margin creates extra space around a component.

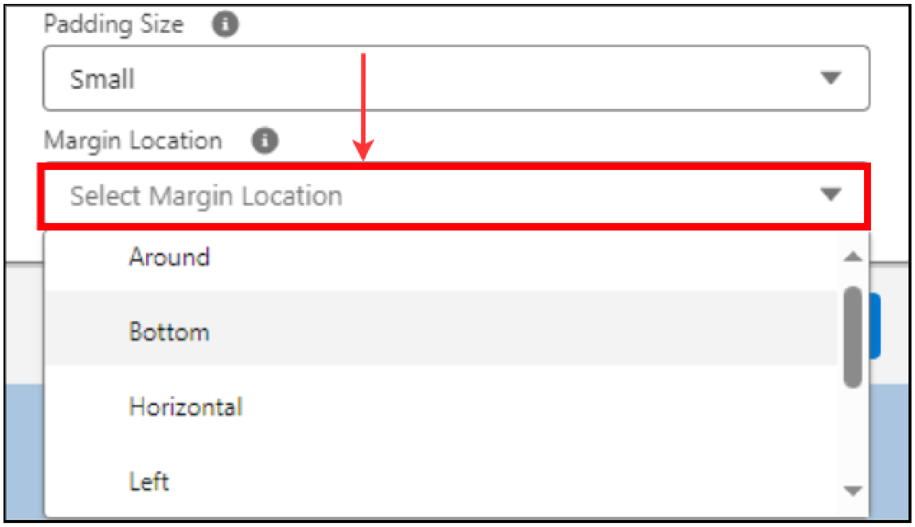


Figure 40: Margin Location Configuration

- **Around:** Margin gets added around the component
- **Top:** Margin gets added at the top of the component
- **Left:** Margin gets added at the left side of the component
- **Bottom:** Margin gets added at the bottom of the component
- **Right:** Margin gets added at the right side of the component
- **Horizontal:** Margin gets added horizontally
- **Vertical:** Margin gets added vertically

* **Margin Size:** The margin size of the view can be set to:

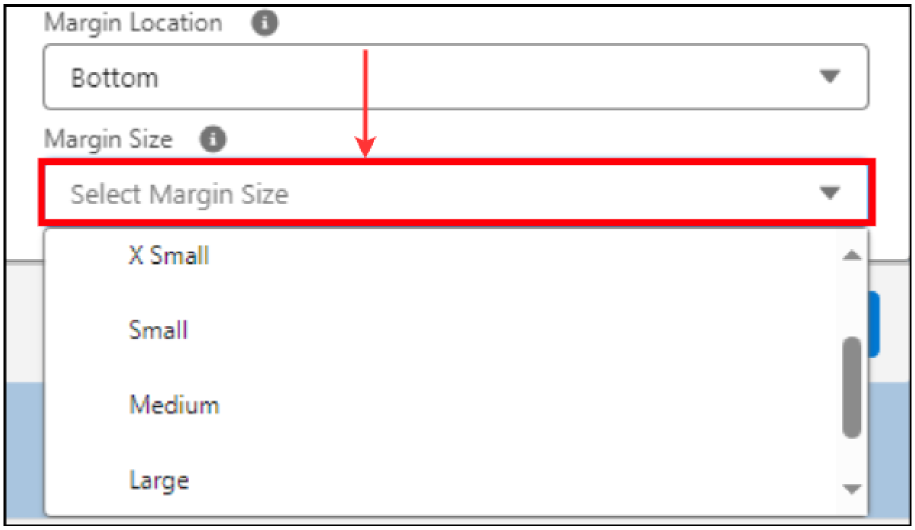


Figure 41: Margin Size Configuration

- None
- xxx-small
- xx-small
- x-small
- Small
- Medium
- Large
- x-large

After clicking on the Save button the config will get saved on the component config object and the user can view this button on the Record page, App page, Home Page, Flows, and Community.

– **Preview Section:**

After the user has selected and filled the fields according to their requirements, they can preview the components in the Preview section before saving them on the Record Page, App Page, Flows, and Community Page.

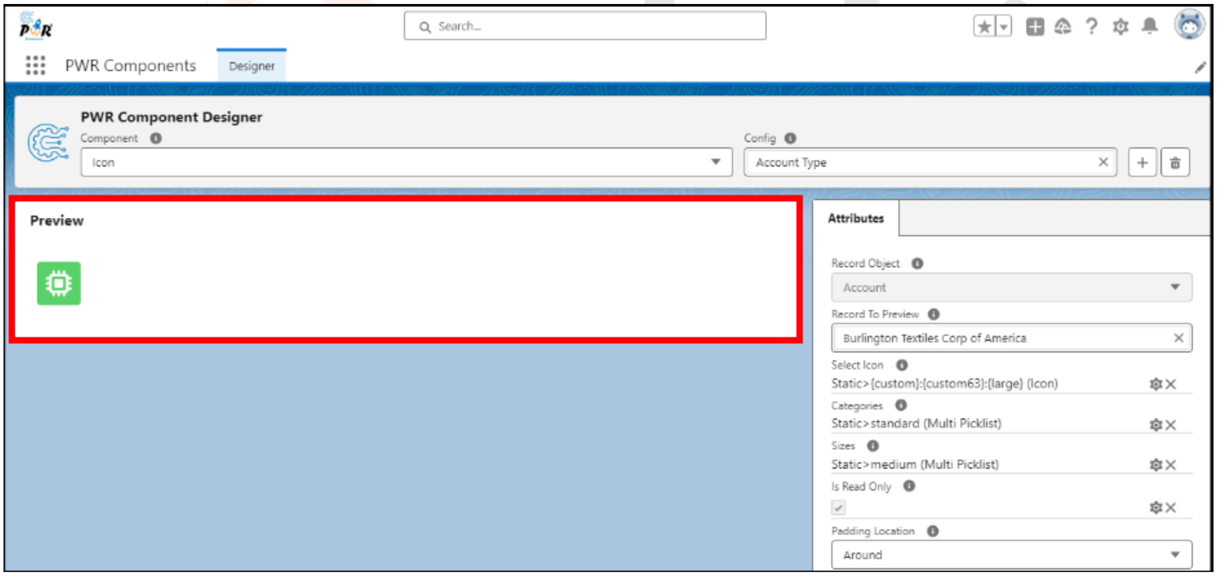


Figure 42: Preview Section

## – PWR Icon screenshots:

* PWR Icon on Record Page:

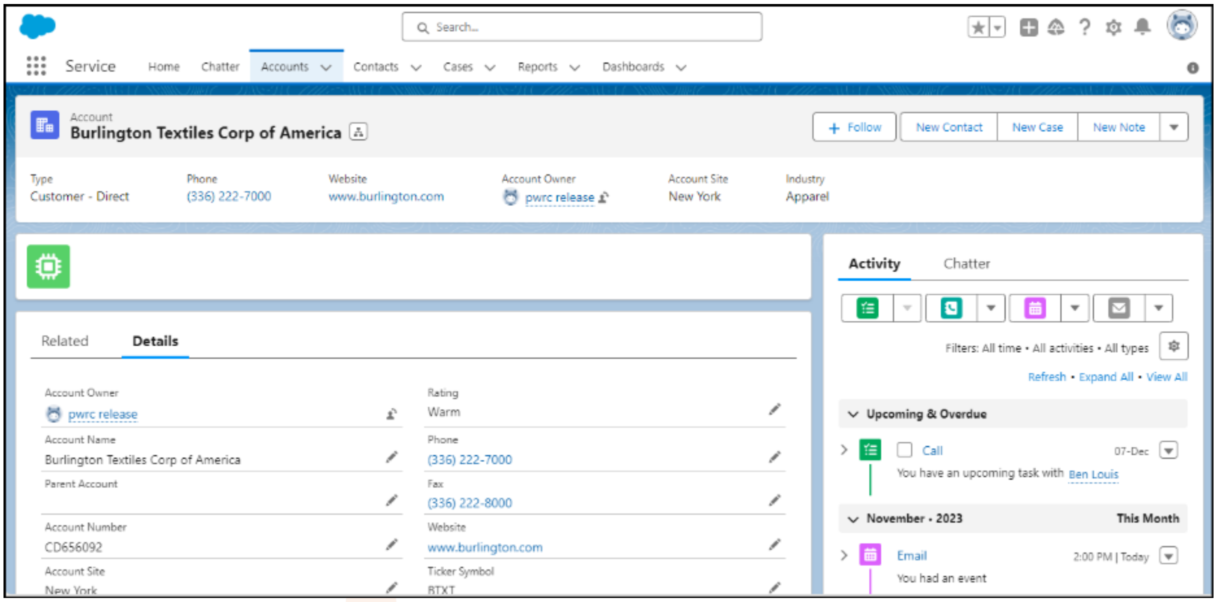


Figure 43: PWR Icon on Record Page

* PWR Icon on Home Page:

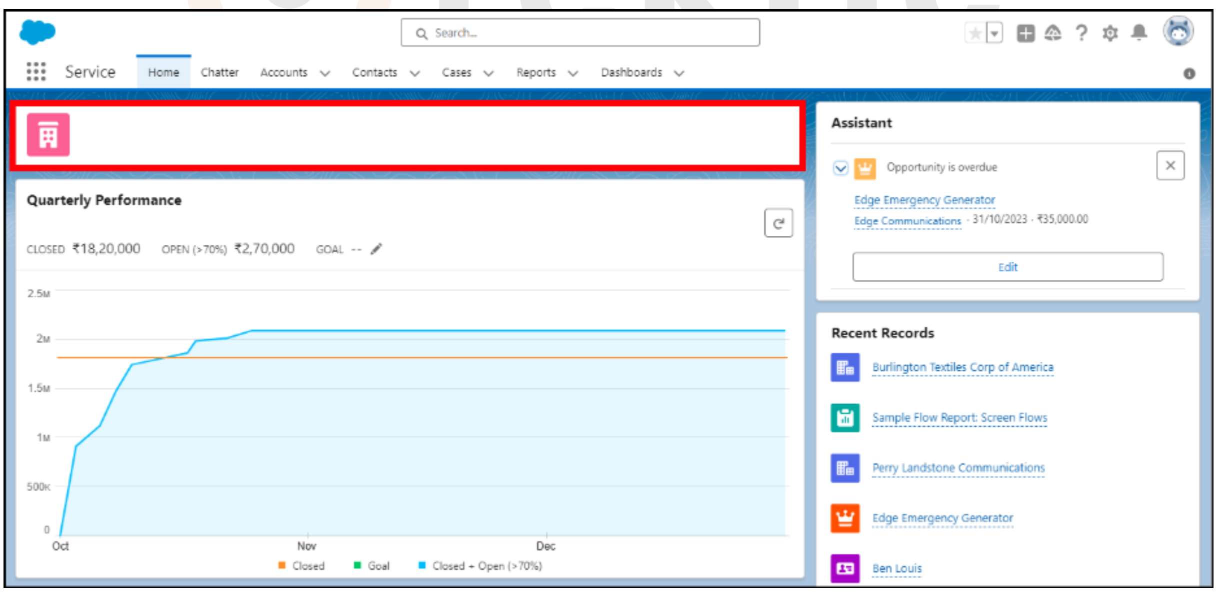


Figure 44: PWR Icon on Home Page

### V.III.3 PWR Progress Bar:

– **Introduction:** Progress Bar displays the progress of an operation. Generally, the Progress Bar displays horizontally from left to right indicating the progress of an operation.

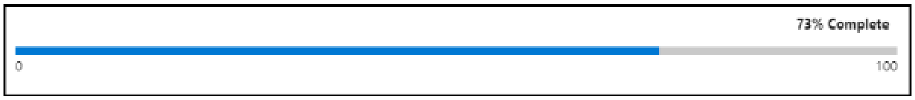


Figure 45: PWR Progress Bar

– **Steps to create progress bar config:**

  ⋆ Go to the designer tab

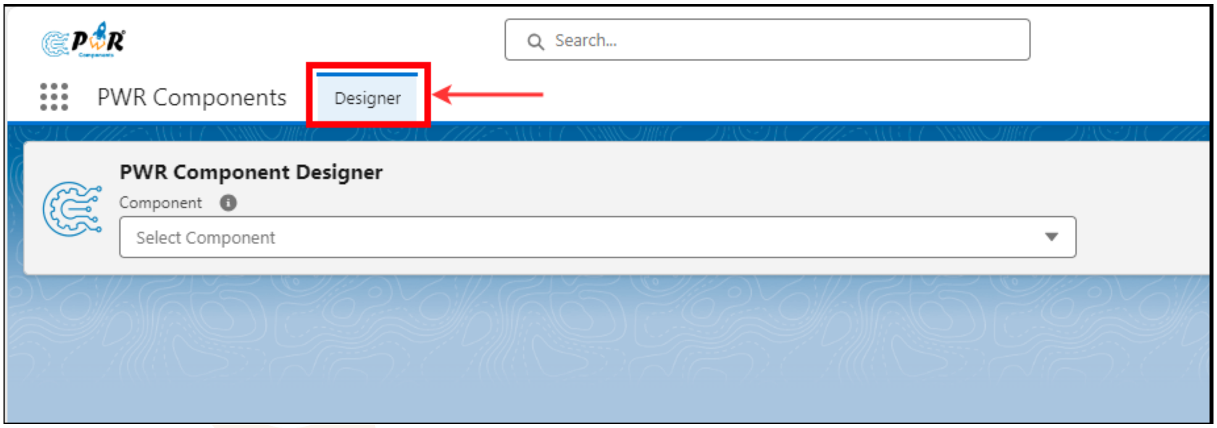


Figure 46: Designer

  ⋆ Select any component from the list of components

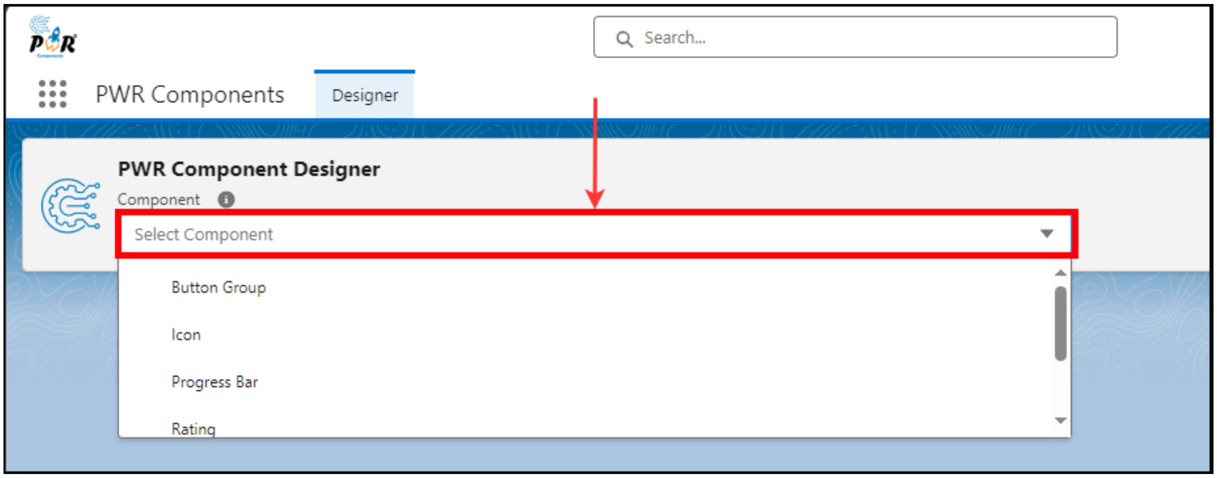


Figure 47: Select Component

  ⋆ Users can use existing Config, or users can create a new Config

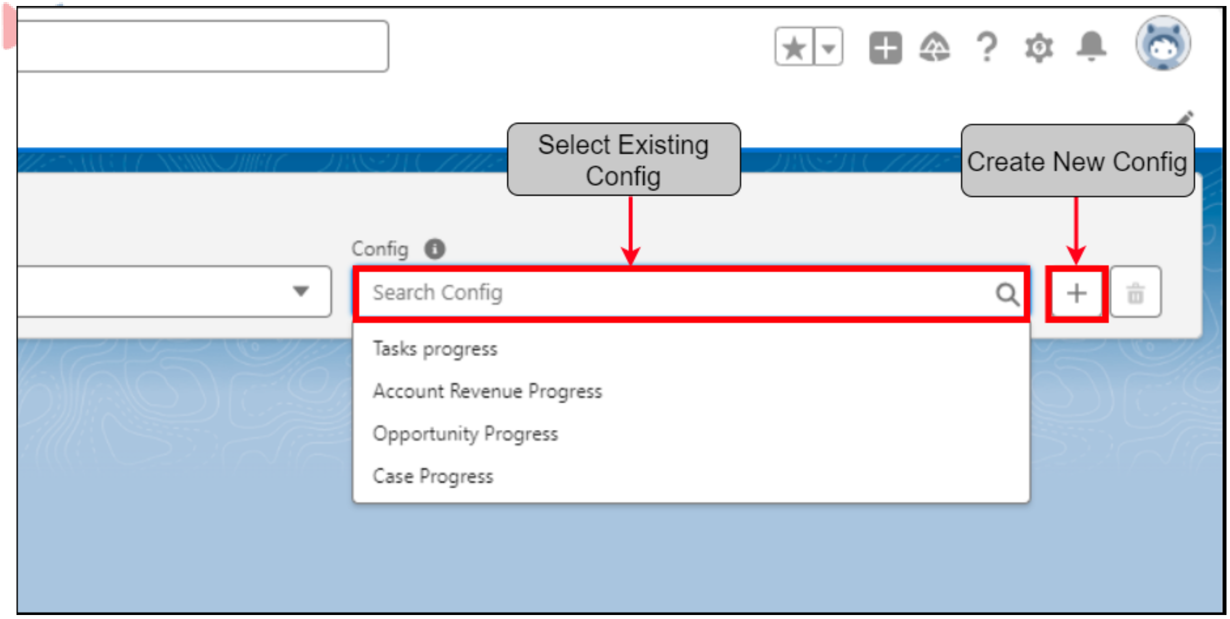


Figure 48: Select Config

* If the user creates a new config

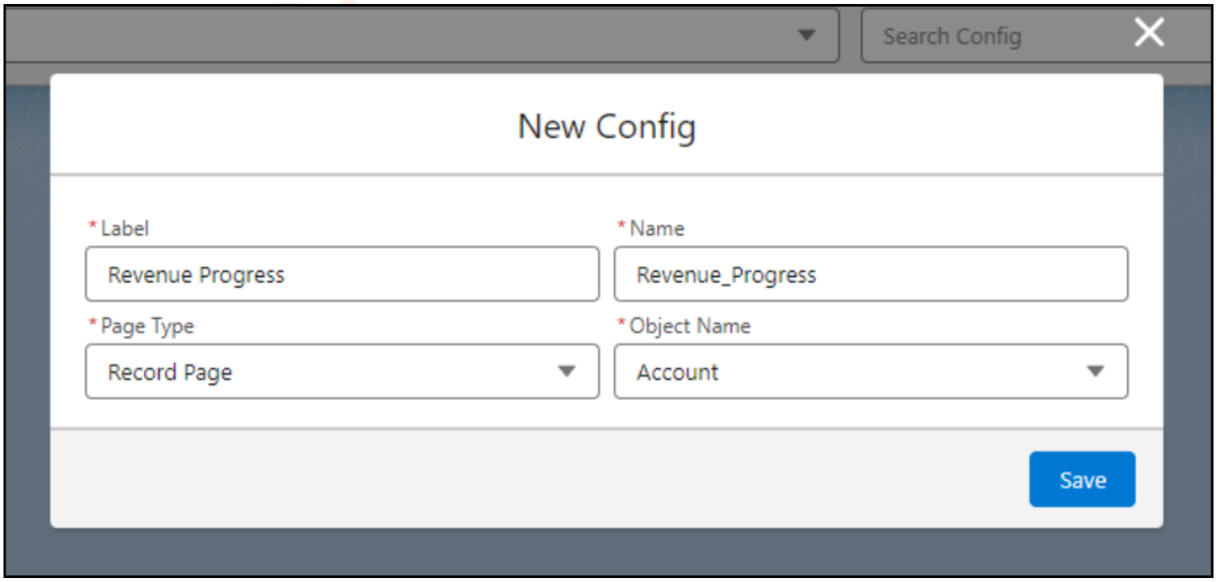


Figure 49: Create Config

- **Label:** Here user can set the title for the config of the component
- **Name:** The API name of the label is shown here
- **Page Type:** Here user can select page type as record page or non-record page; depending on which page the user wants to show the component
- **Non-Record Page:** By choosing the non-record page user can use this config on the home page and app page

- **Record Page:** By choosing the record page user can use this config on a related record object
- **Object Name:** Here user can choose any object on which that created config will be shown

**Here are the details around different attributes, and which valid information can be input for the same.**

– **Attributes:**

* **Record Object:** The object name is auto-populated here as you select the object at the time of config creation.
* **Record To Preview:** This feature helps you preview and evaluate the progress bar's appearance according to the selected record.

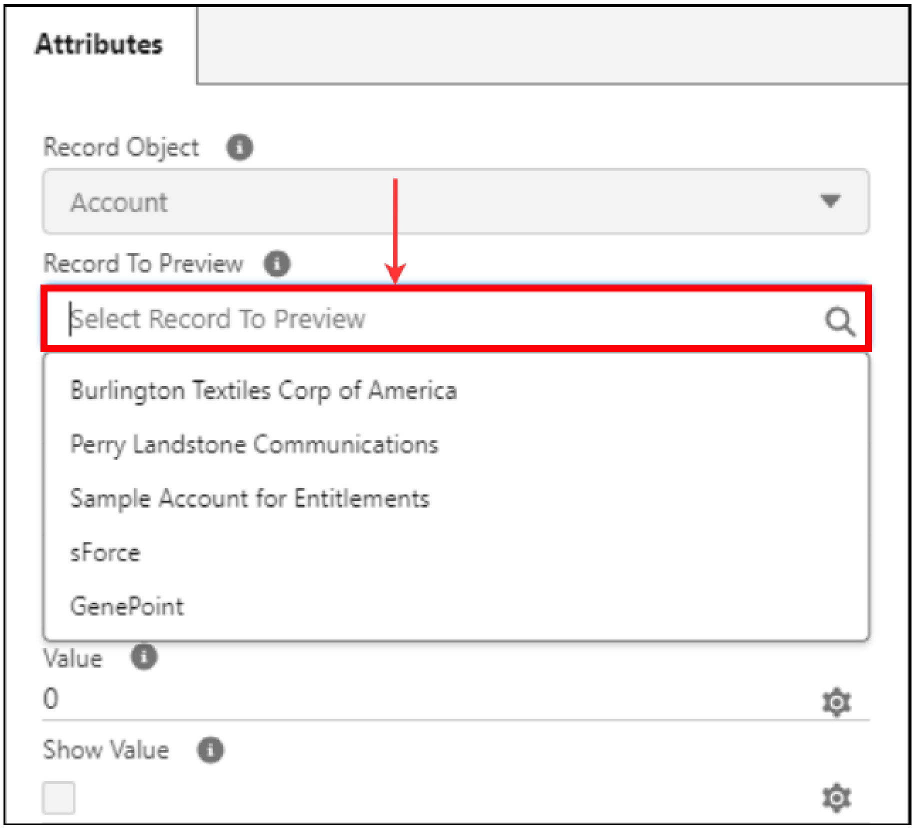


Figure 50: Select Record to Preview

* **Label:** Here user can set the title for the component. User can set the label by given value sources in the property binder.

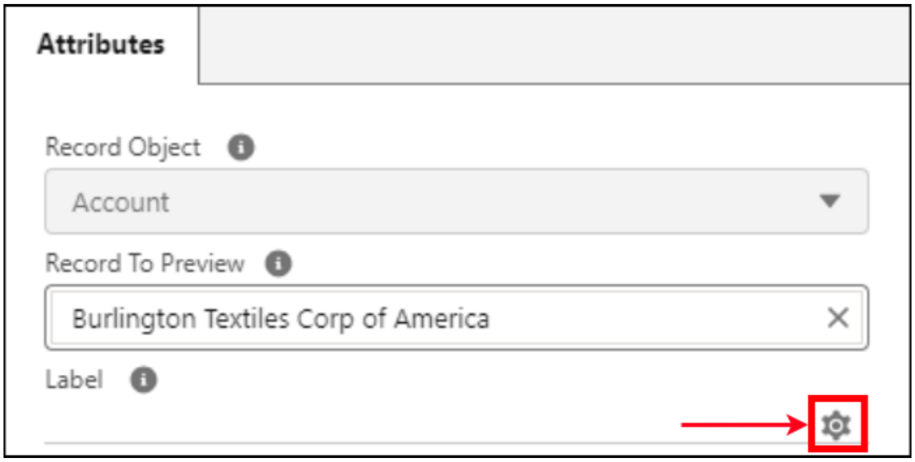


Figure 51: Label Config

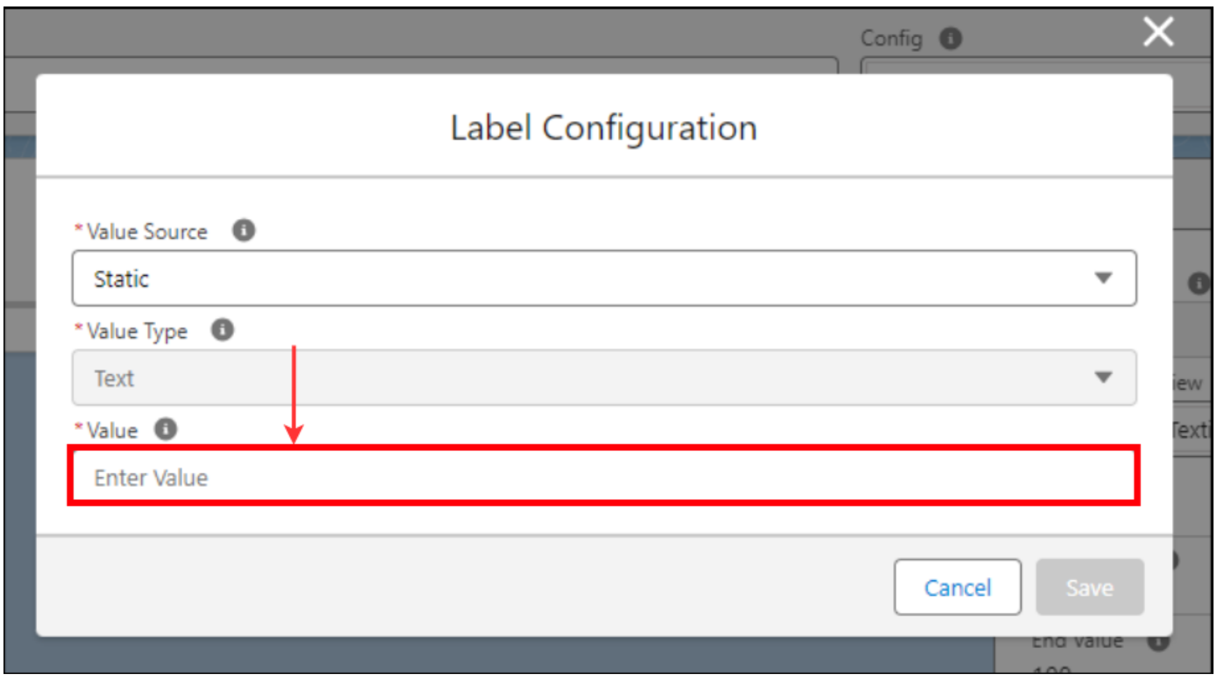


Figure 52: Label Value

* **Start Value:** "Start Value" refers to the initial point or position on a progress bar where the tracking of progress begins. You can set the start value for the progress bar by given value sources in the property binder, by default start value is set as 0.

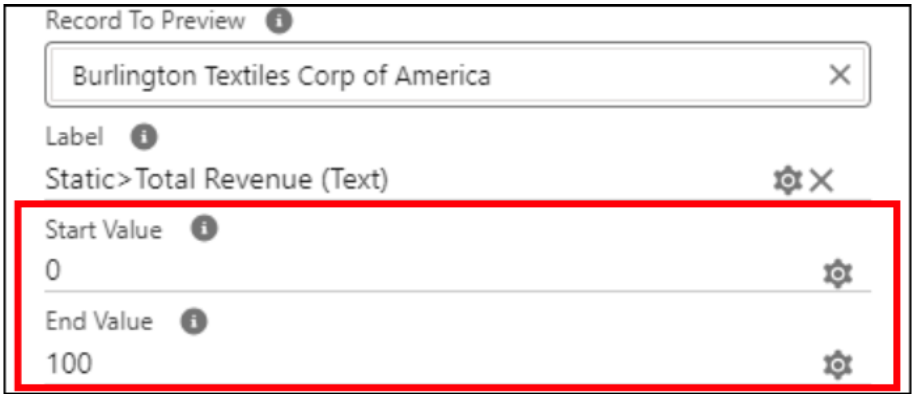


Figure 53: Start & End value Config

* **End Value:** ”End Value” refers to the maximum or final point of progress indicated on a progress bar. User can set the end value for the progress bar by giving value sources in the property binder, by default end value is set as 100.
* **Value:** ”Current Value” refers to the current progress within a process or task that is represented by a progress bar. User can set the progress value for the progress bar by given value sources in the property binder.

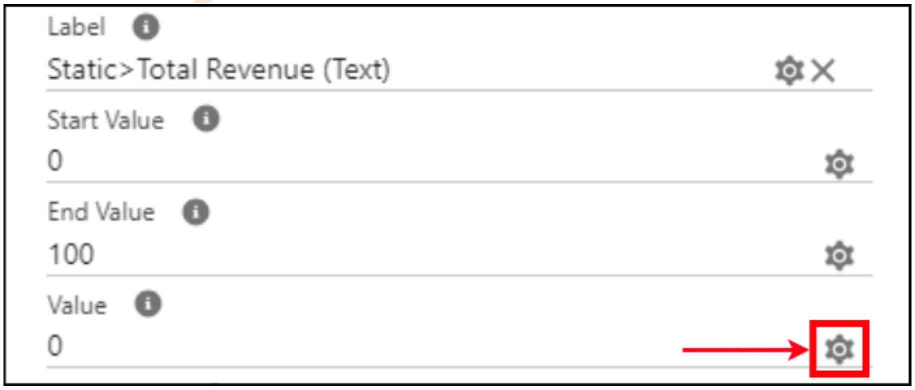


Figure 54: Value Config

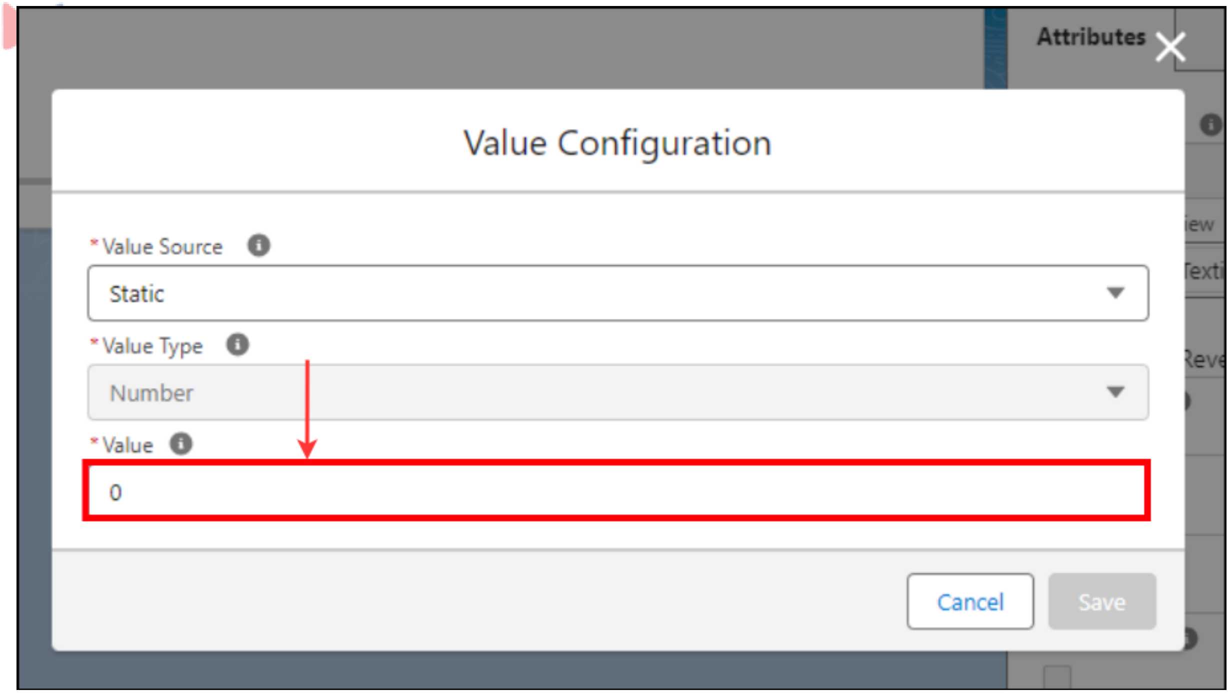


Figure 55: Progress Value

* **Show Value:** "Show Value" is a feature in a progress bar that allows you to display the start value and end value on the progress bar.

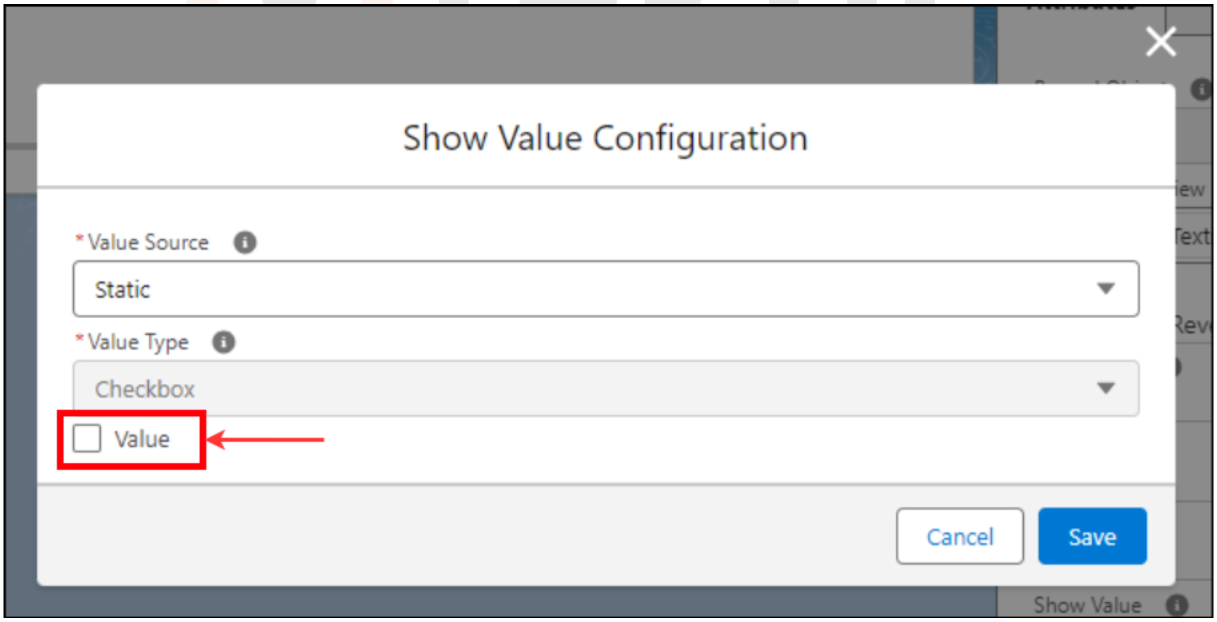


Figure 56: Show Value Config

* **Show Percentage:** "Show Percentage" is a feature in a progress bar that allows you to display the completion percentage on the progress bar. User can set the show progress for the progress bar by given value sources in the prop

erty binder.

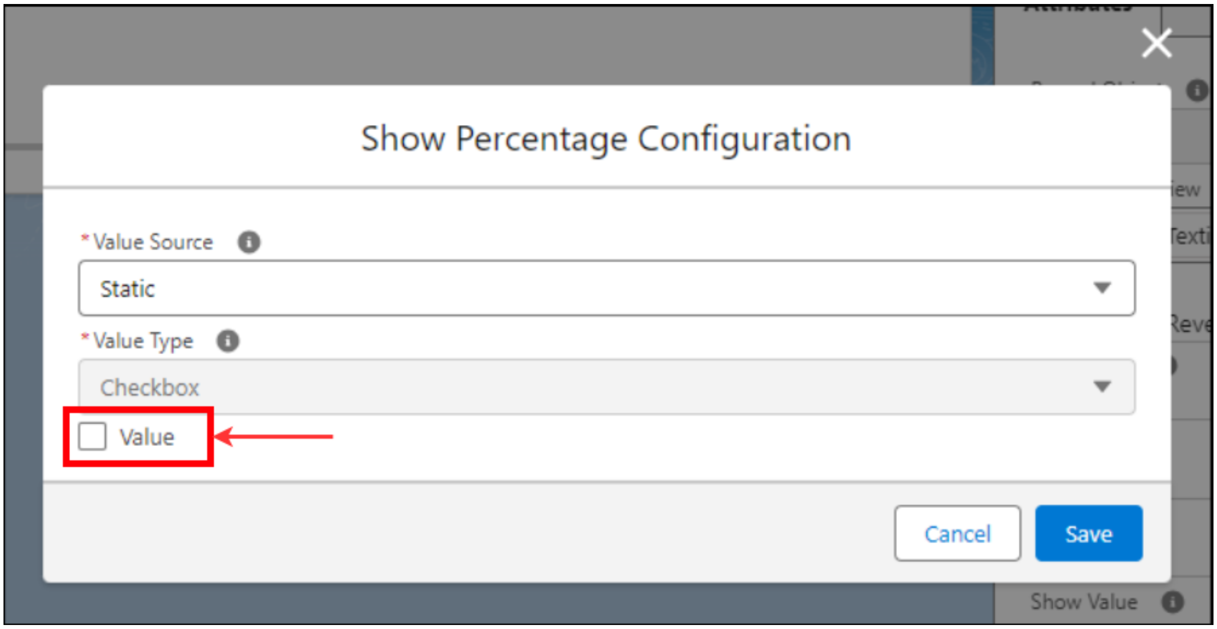


Figure 57: Show Percentage Config

- **Is Vertical:** This checkbox property enables the user to show their progress bar vertically. User can set the progress value for the vertical by given value sources in the property binder.

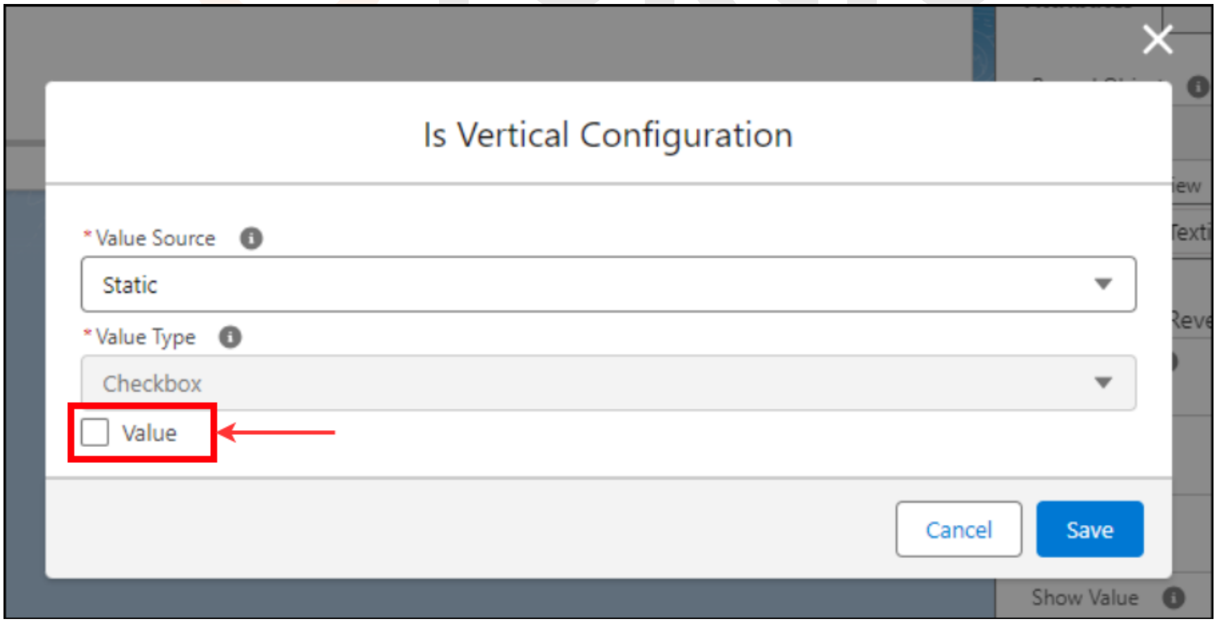


Figure 58: Is Vertical Config

- **Show Radius:** This property enables the user to show their progress bar with rounded corners. Users can set the progress value for the show radius by given value sources in the property binder.

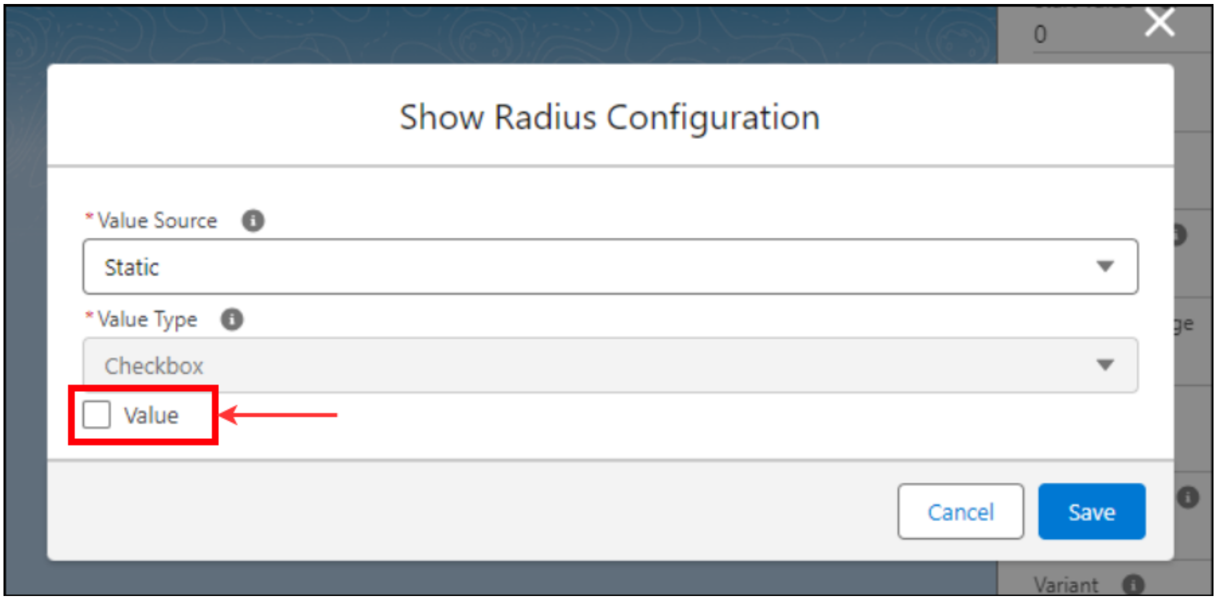


Figure 59: Show Radius Config

* **Variant:** Variants display the Progress Bar with different colors to convey different meanings, user can change the appearance of the progress bar by selecting one of the options given, by default Brand variant is selected. Progress bar comes in five variants:

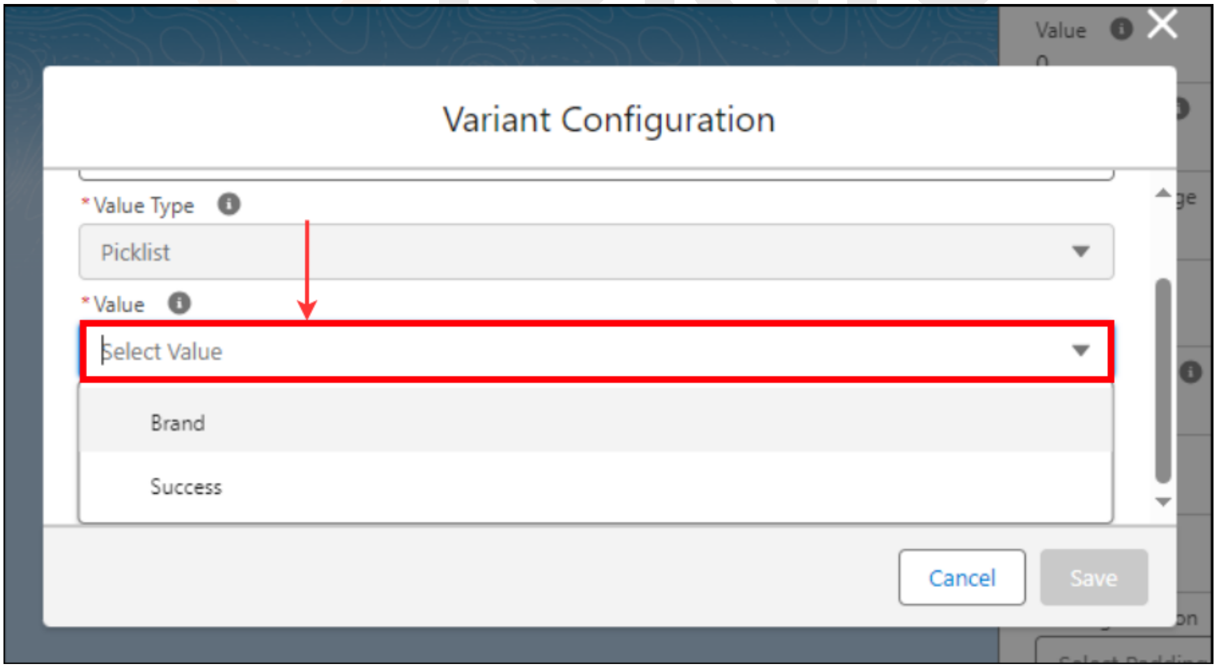


Figure 60: Varient Config

- **Brand:** Shows Progress Bar in Blue color

- **Success:** Shows Progress Bar in green color

* **Size:** User can set their preference for the thickness of the Progress Bar; Users can get the size through the different value sources in the property binder.

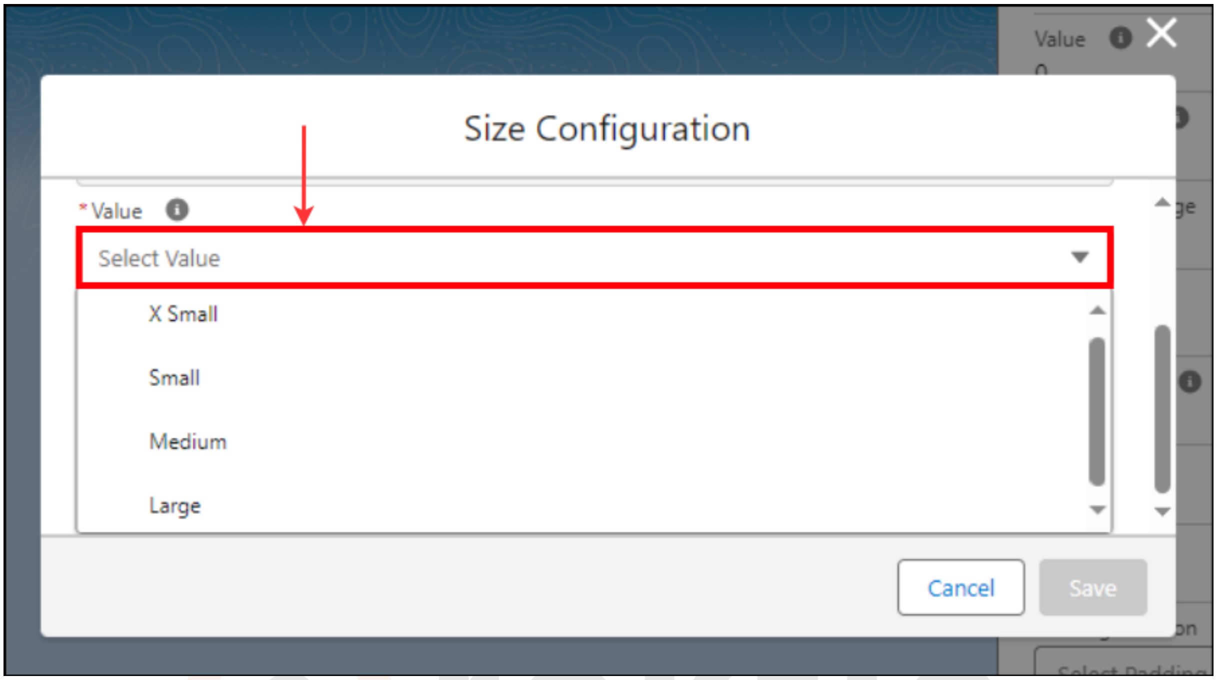


Figure 61: Size Config

Progress Bars are available in four sizes i.e.

- x-small
- small
- medium
- large

* **Padding Location:** Defines the position of the padding for a component. The padding creates extra space within a component.

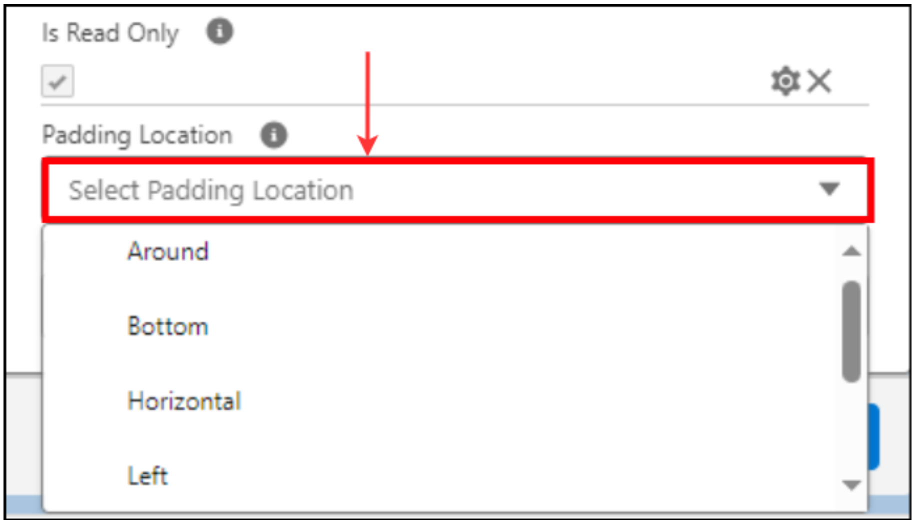


Figure 62: Padding Location Config

- **Around:** Creates padding around the component
- **Top:** Creates padding at the top of the component
- **Left:** Creates padding at the left side of the component
- **Bottom:** Creates padding at the bottom of the component
- **Right:** Creates padding at the right side of the component
- **Horizontal:** Creates padding horizontally
- **Vertical:** Creates padding vertically

* Padding Size: The padding size of the component can be set to:

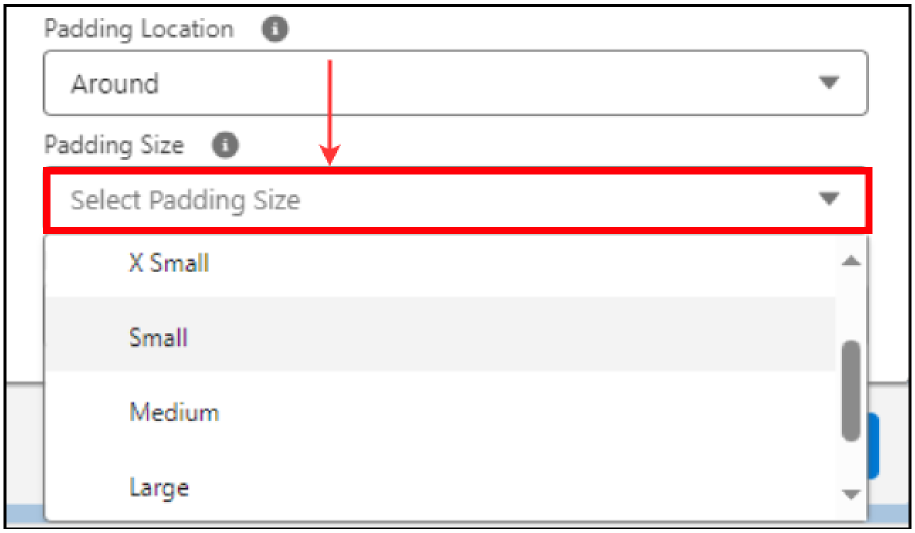


Figure 63: Padding Size Config

- None
- xxx-small
- xx-small
- x-small
- Small
- Medium
- Large
- x-large

* **Margin Location:** Defines the position of the margin for a component. margin creates extra space around a component.

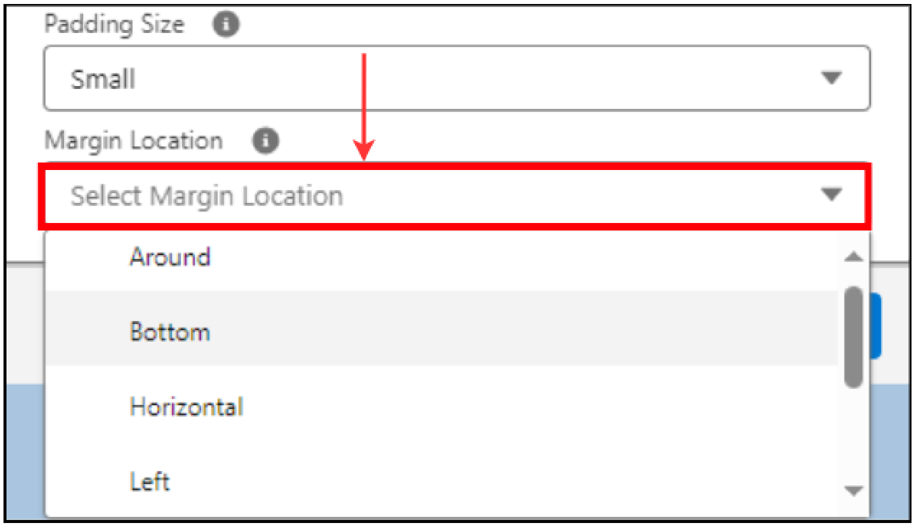


Figure 64: Margin Location Config

- **Around:** Margin gets added around the component
- **Top:** Margin gets added at the top of the component
- **Left:** Margin gets added at the left side of the component
- **Bottom:** Margin gets added at the bottom of the component
- **Right:** Margin gets added at the right side of the component
- **Horizontal:** Margin gets added horizontally
- **Vertical:** Margin gets added vertically

* **Margin Size:** The margin size of the view can be set to:

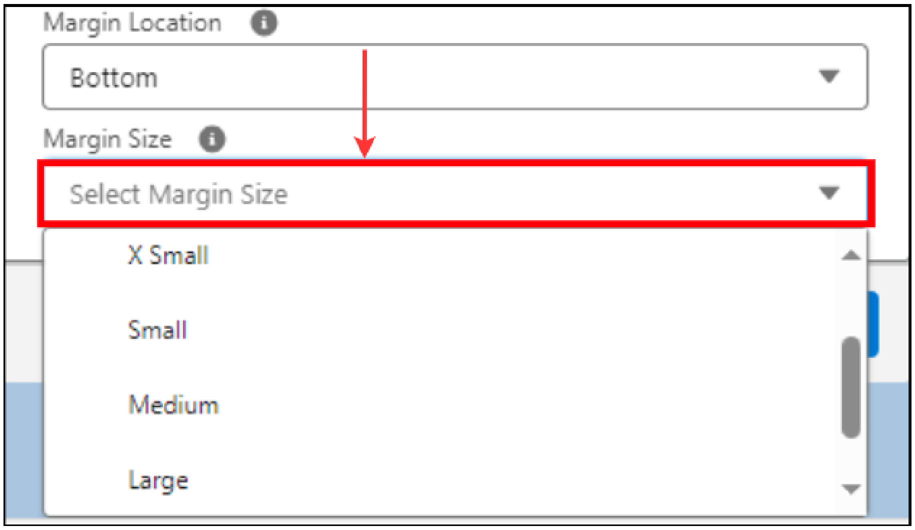


Figure 65: Margin Size Config

· None
· xxx-small
· xx-small
· x-small
· Small
· Medium
· Large
· x-large

After clicking on the Save button the config will get saved on the component config object and the user can view this button on the Record page, App page, Home Page, Flows, Community.

– **Preview Section:**

After the user has selected and filled the fields according to their requirements, they can preview the components in the Preview section before saving them on the Record Page, App Page, Flows, and Community Page.

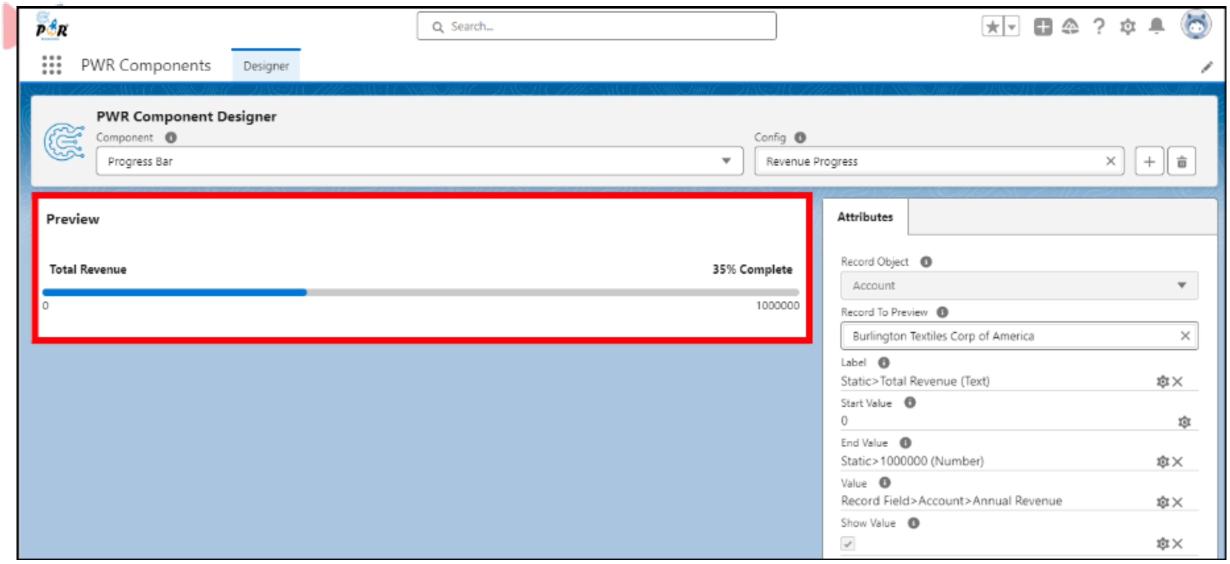


Figure 66: Preview Section

– **PWR Progress Bar screenshots:**

* PWR Progress Bar on Record Page:

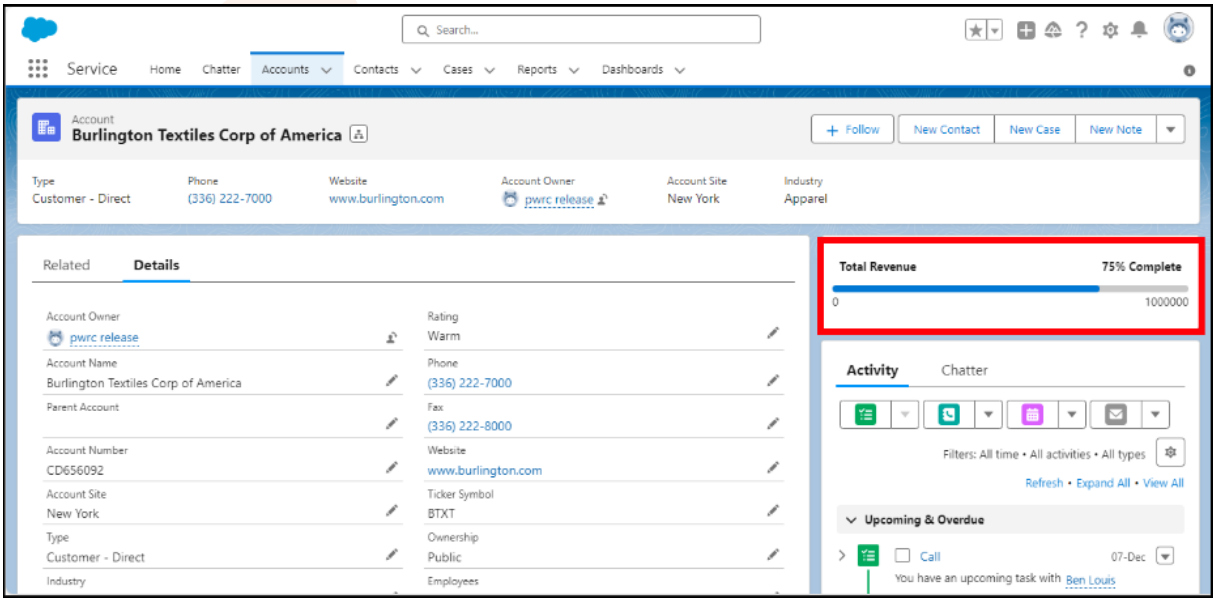


Figure 67: PWR Progress Bar on Record Page

* PWR Progress Bar on Home Page:

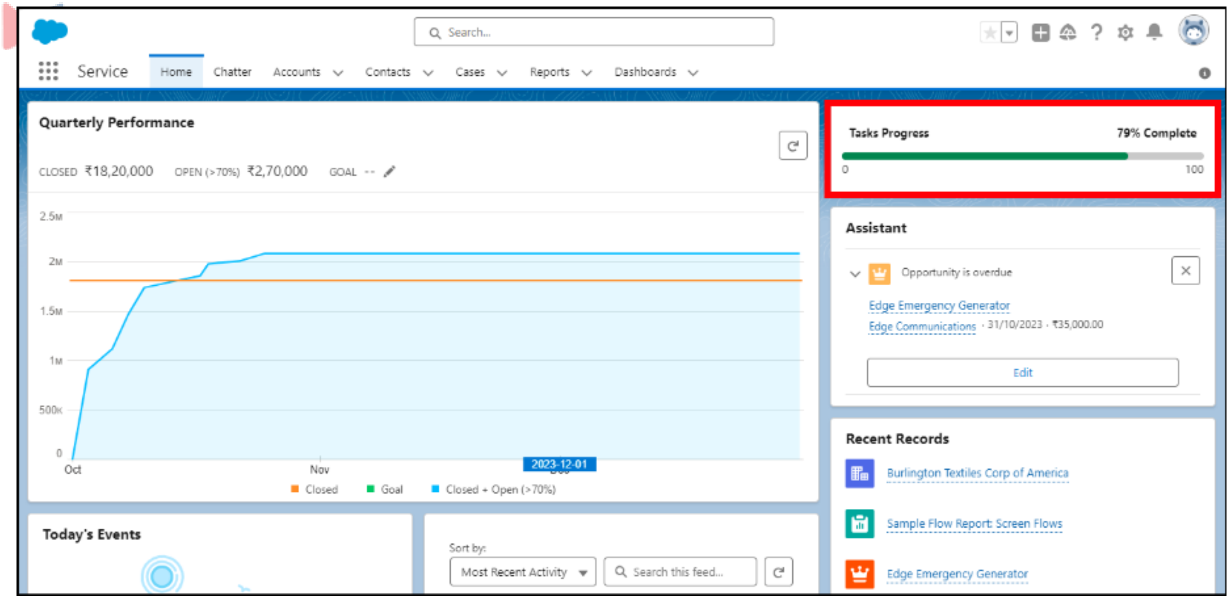


Figure 68: PWR Progress Bar on Home Page

### V.III.4 PWR Rating:

- **Introduction:**

  The PWR Rating component is used for rating purposes. It can be used for giving ratings to an object, fields, etc. We can display the rating on the Home Page, Record Page, Flows, Community Page, etc.


Figure 69: PWR Rating

- **Steps to create rating config:**

  * Go to the designer tab

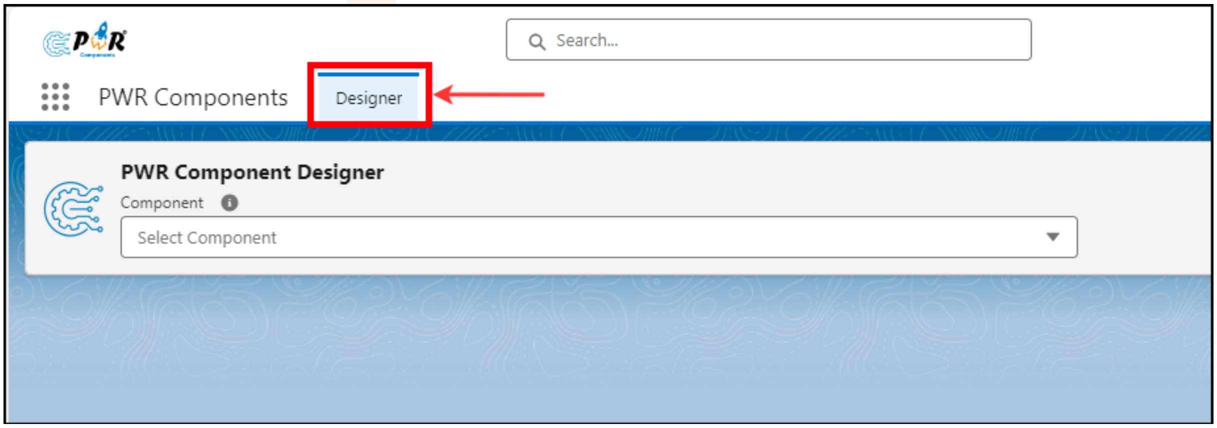


Figure 70: Designer

  * Select any component from the list of components

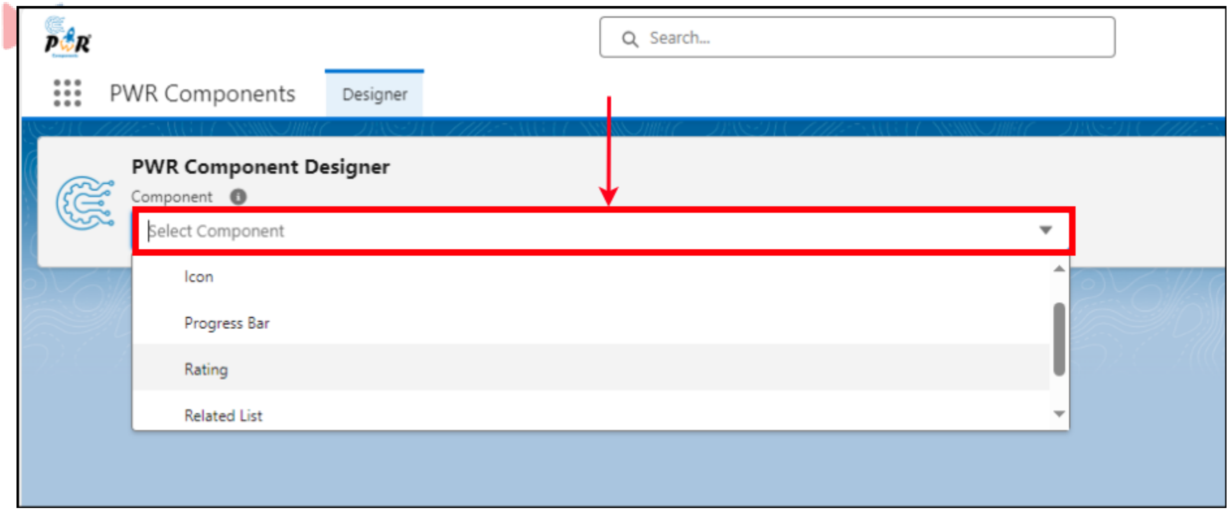


Figure 71: Select Component

* Users can use existing Config, or users can create a new Config

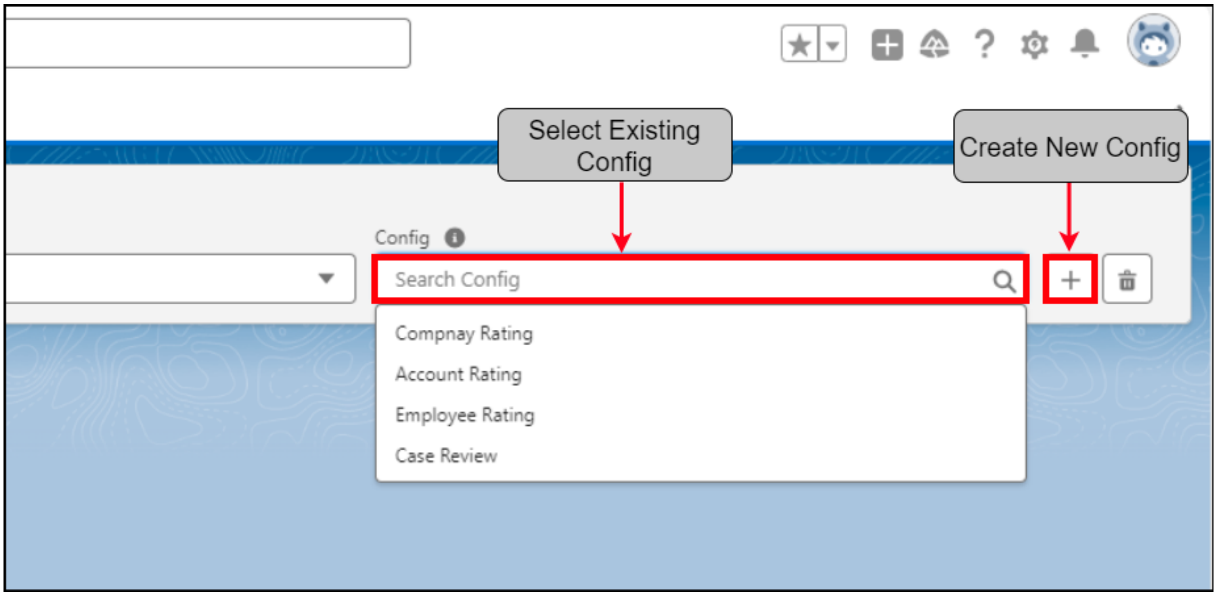


Figure 72: Select Config

* If the user creates a new config

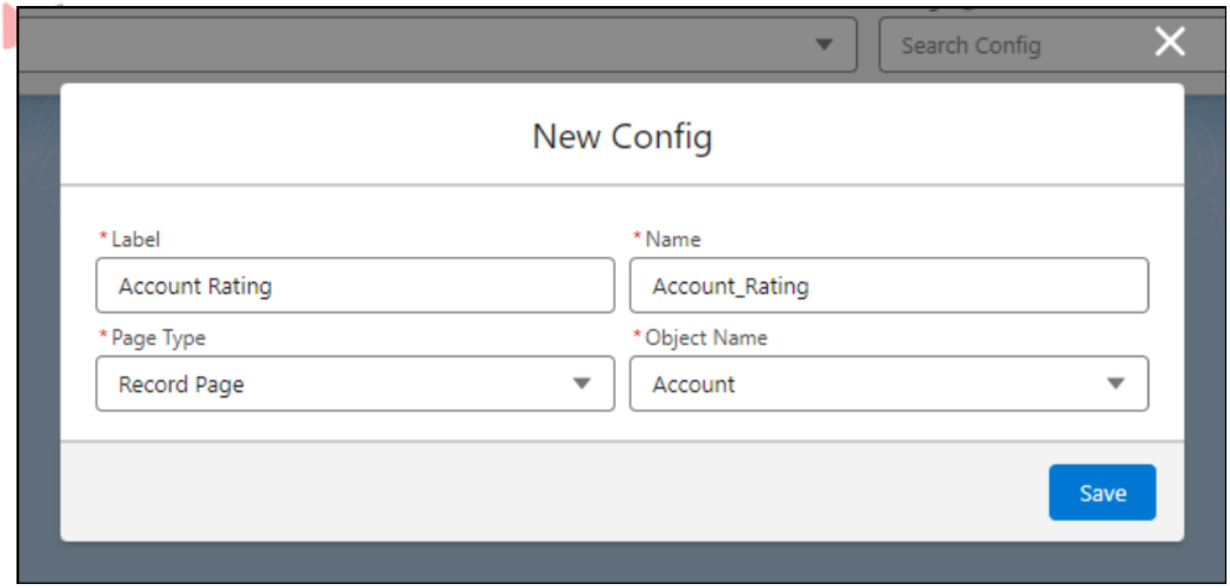


Figure 73: Create Config

- **Label:** Here user can set the title for the config of the component
- **Name:** The API name of the label is shown here
- **Page Type:** Here user can select page type as record page or non-record page; depending on which page the user wants to show the component
- **Non-Record Page:** By choosing the non-record page user can use this config on the home page and app page
- **Record Page:** By choosing the record page user can use this config on a related record object
- **Object Name:** Here user can choose any object on which that created config will be shown

Here are the details around different attributes, and which valid information can be input for the same.

– **Attributes:**

  * **Record Object:** The object name is auto-populated here as you select the object at the time of config creation.
  * **Record To Preview:** This feature helps you preview and evaluate the rating's appearance according to the selected record.
  * **Label:** Here user can set the title for the component. Users can set the label by given value sources in the property binder.

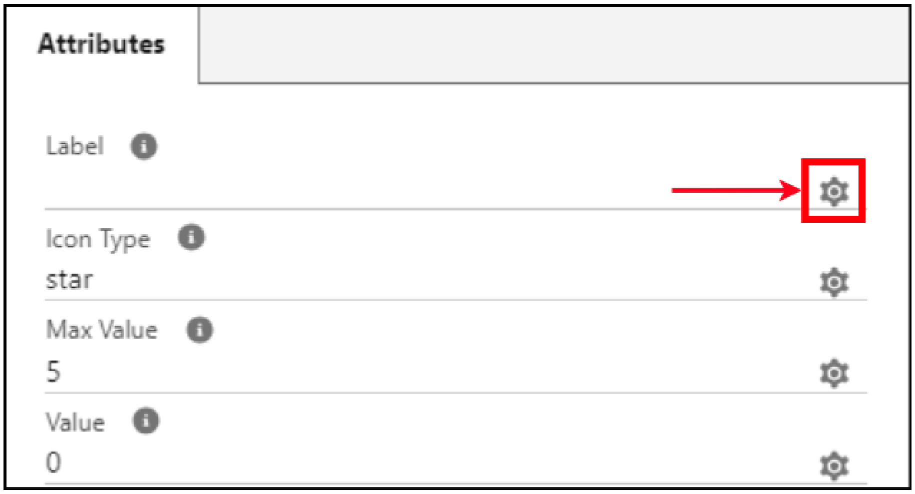


Figure 74: Label Config

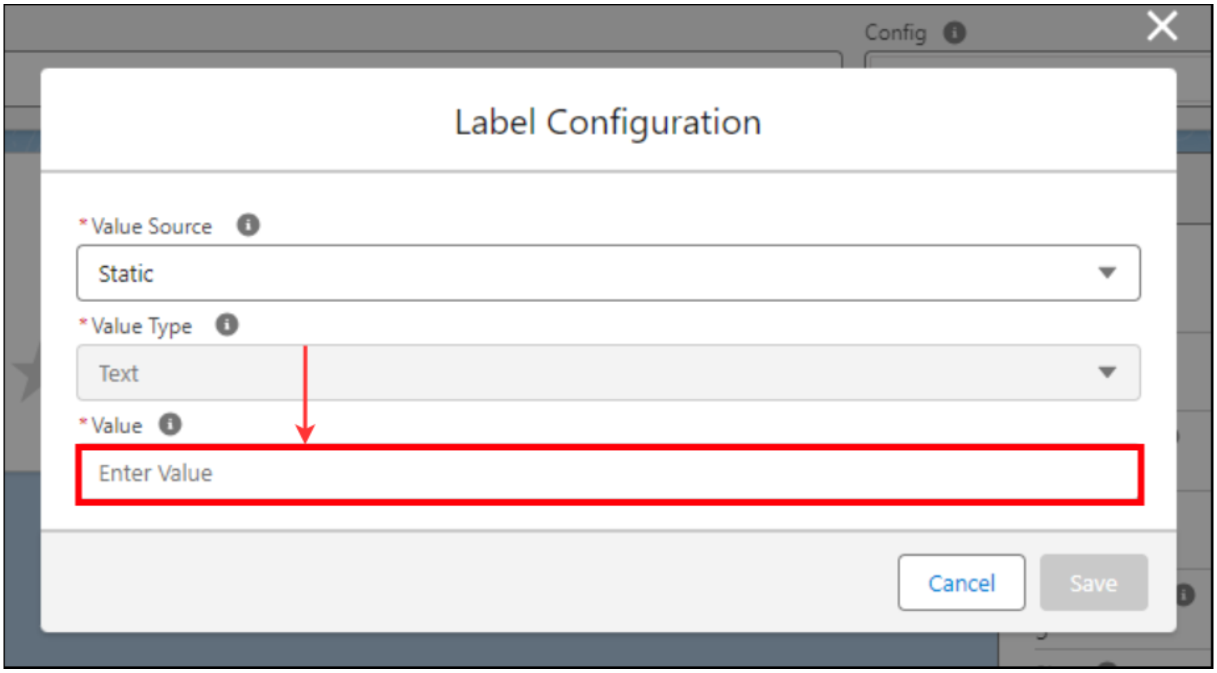


Figure 75: Label Value

* **Icon Type:** This feature allows users to choose the style of icons or symbols that will be displayed to users when they rate or provide feedback. Users can set the icon type by given value sources in the property binder.


Figure 76: Icon Type Config

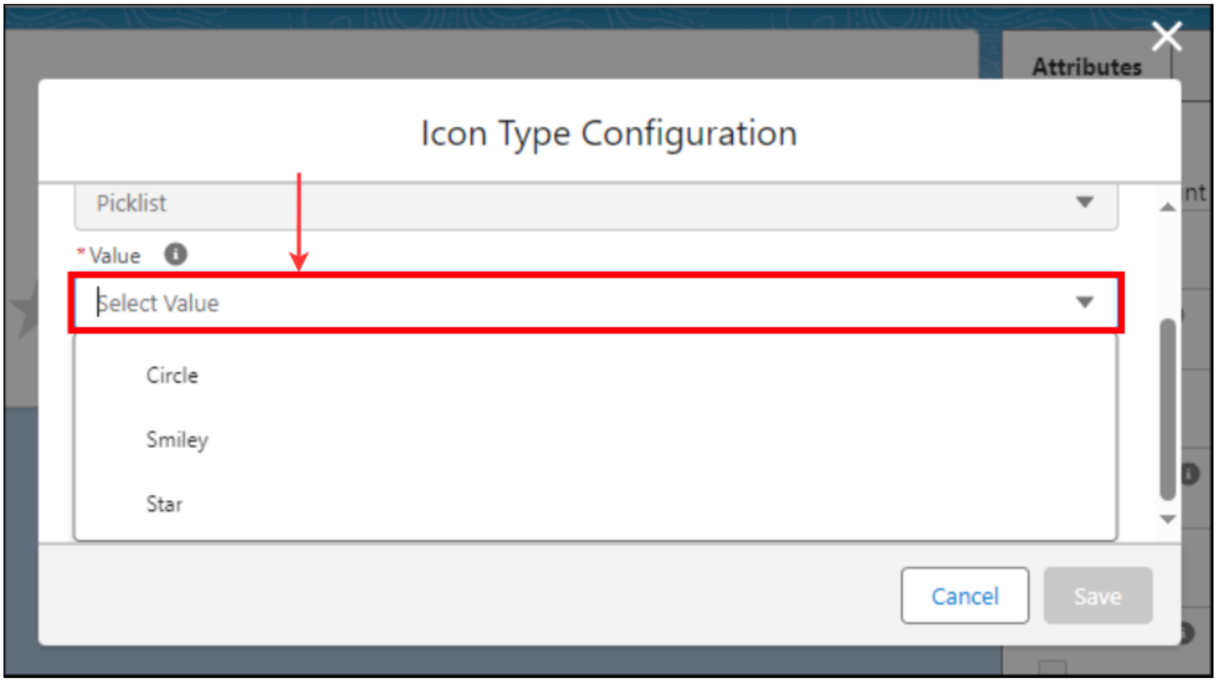


Figure 77: Icon Type Value

There are three Icon types, in which the user can fill in the feedback/ ratings by default star icon type is selected

- **Circle:** If the user selects the icon type value as a circle, then the circle will display
- **Smiley:** If the user selects the icon type value as smiley, then smiley will display
- **Star:** If the user selects the icon type value as a star, then the star will display

* **Max Value:** "Max Value" in ratings refers to the highest or maximum achievable value within a rating scale. For example, on a rating scale of 1 to 10 where 10 represents the "Max Value", by default max value is set as 5. Users can set the max value through the different value sources in the property binder.

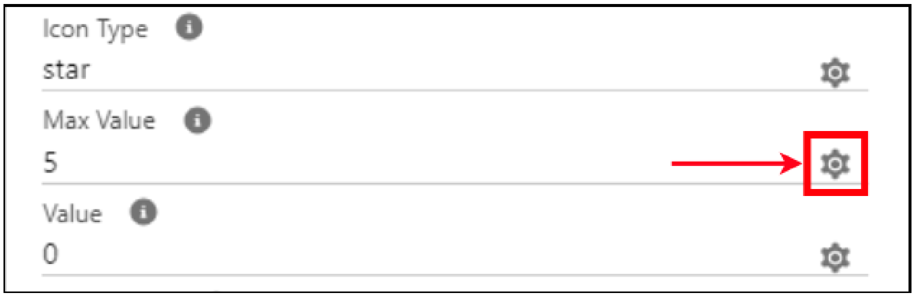


Figure 78: Max Value Config

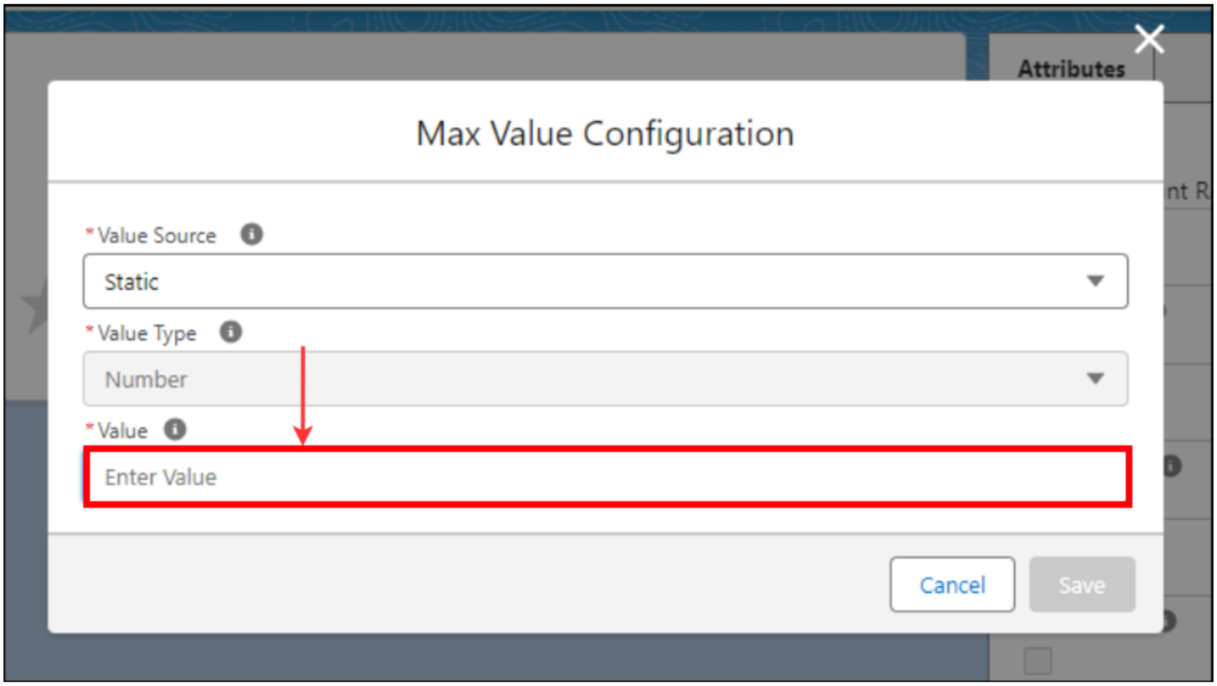


Figure 79: Max Value

* **Value:** The rating will be measured by the value in this field. Users can get the value through the different value sources in the property binder. By default, the value is set as 0. Users can set the value through the different value sources in the property binder.

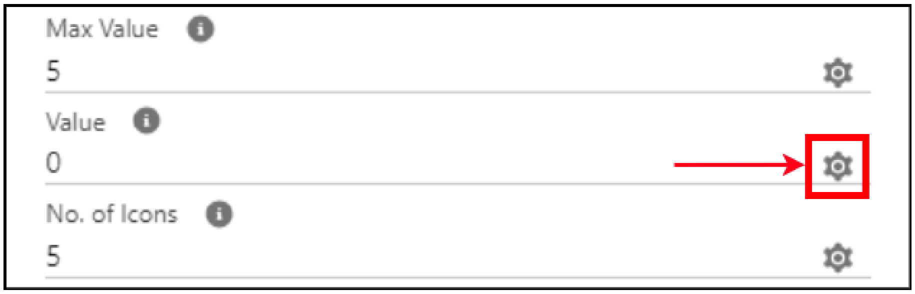


Figure 80: Value Config

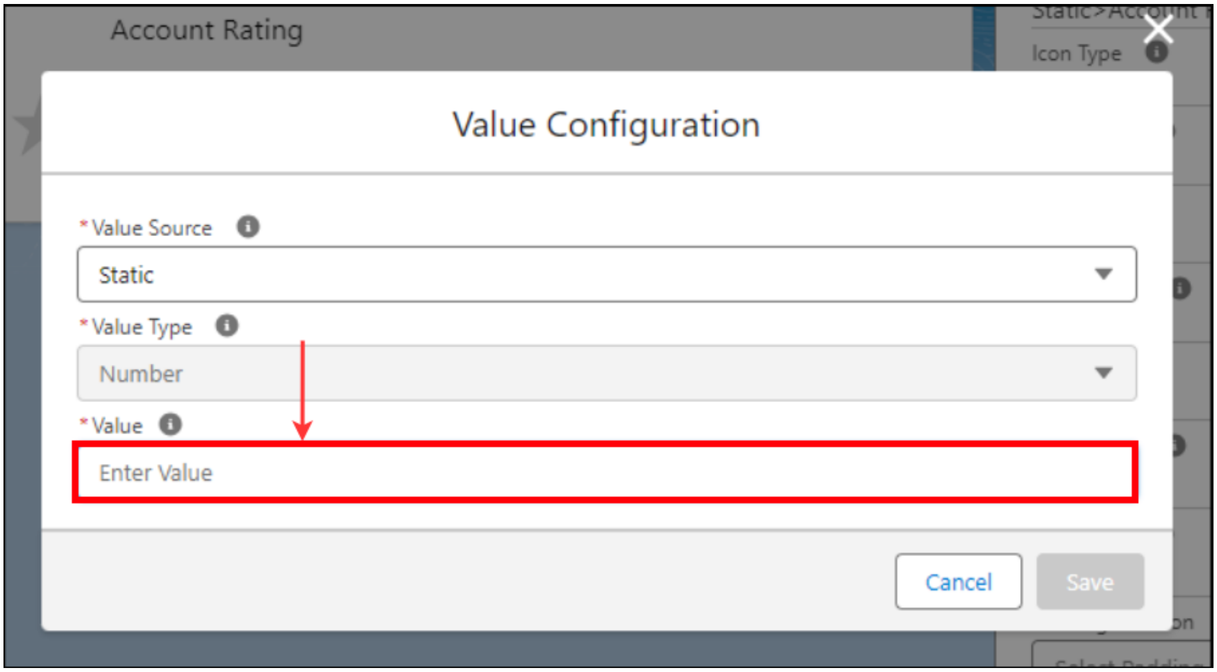


Figure 81: Progress Value

* **No. of icons:** This feature allows users to determine how many individual icons or symbols will be displayed to users when they provide feedback, by default no. of icons is selected as 5. Users can get the no. of icons through the different value sources in the property binder.

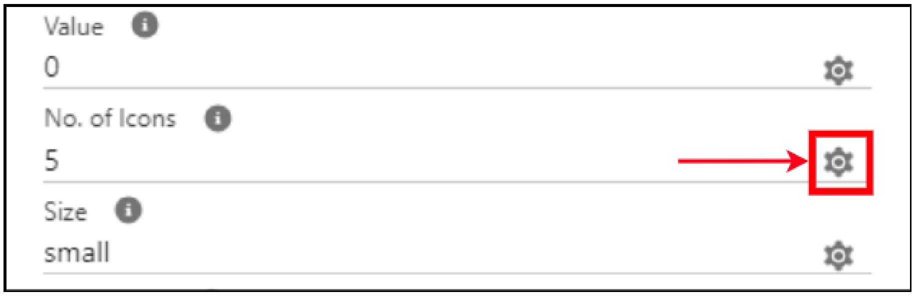


Figure 82: No of Icons Config

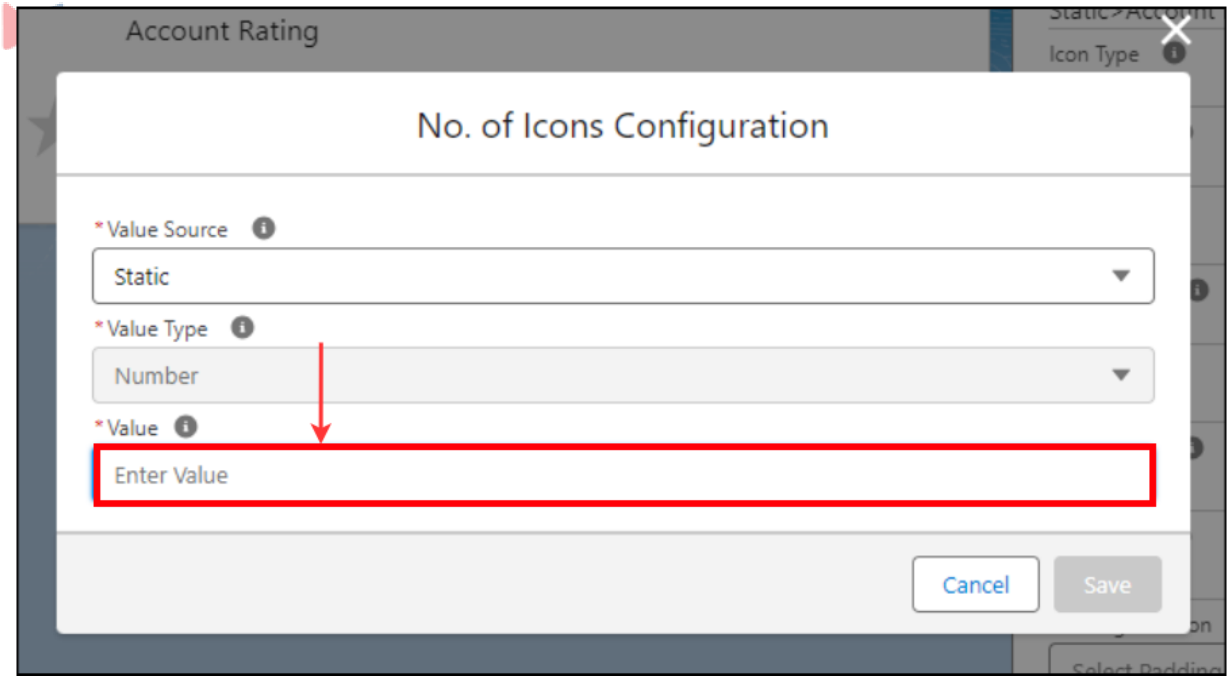


Figure 83: No of Icons Value

- **Size:** User can set the size of the rating icons; Users can get the size through the different value sources in the property binder.


Figure 84: Size Config

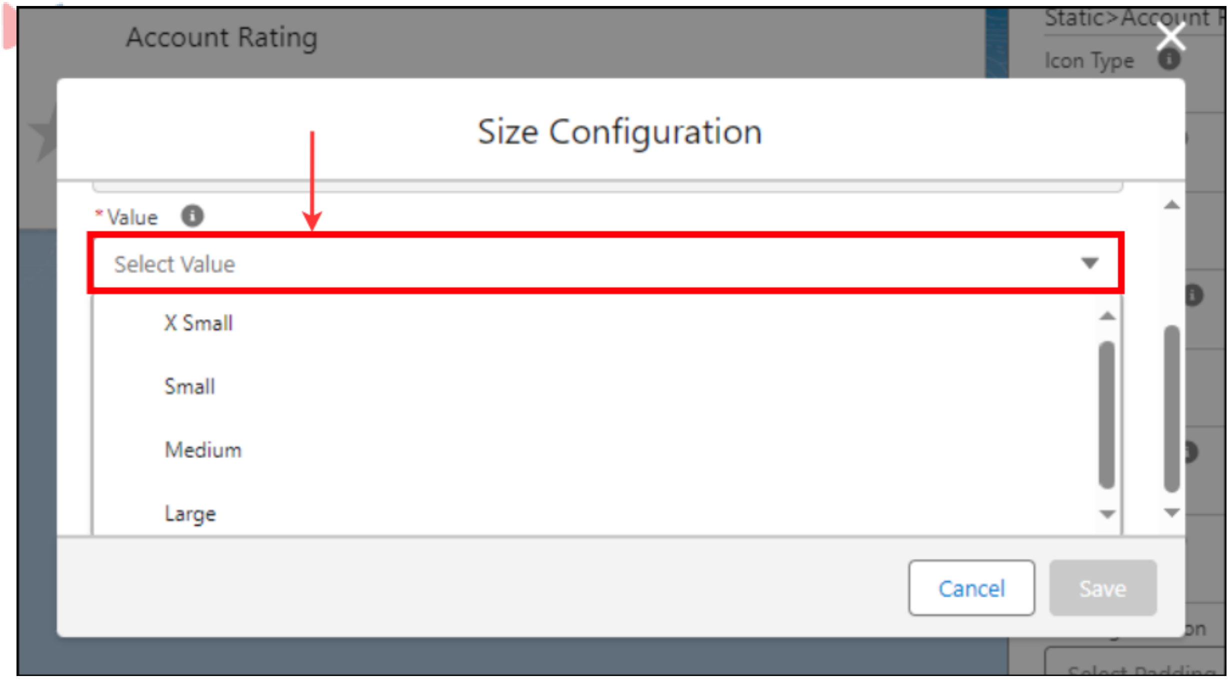


Figure 85: Size Value

Rating sizes are available in four sizes i.e., x-small, small, medium, and large

- **Show Value:** "Show Value" is a feature in a rating that allows the user to display the rating value on the rating component. Users can set the show value through the different value sources in the property binder.

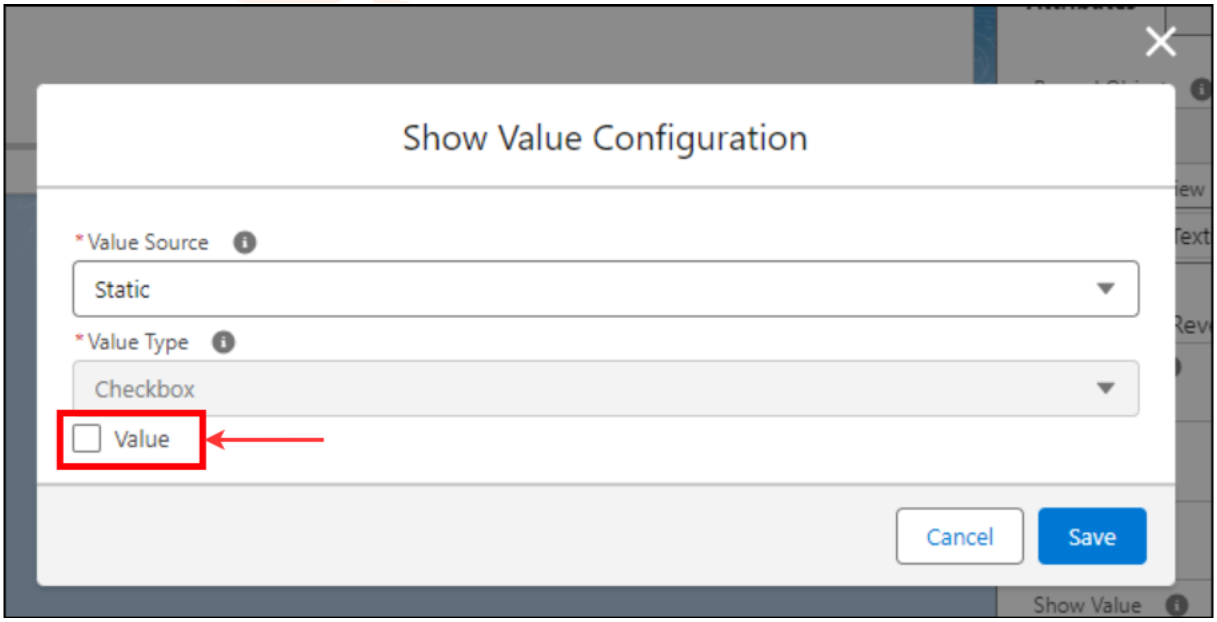


Figure 86: Show Value Config

- **Read Only:** If the value is checked as true then the user cannot change/edit the value from the page/ preview. Users can set the read-only through the different value sources in the property binder.

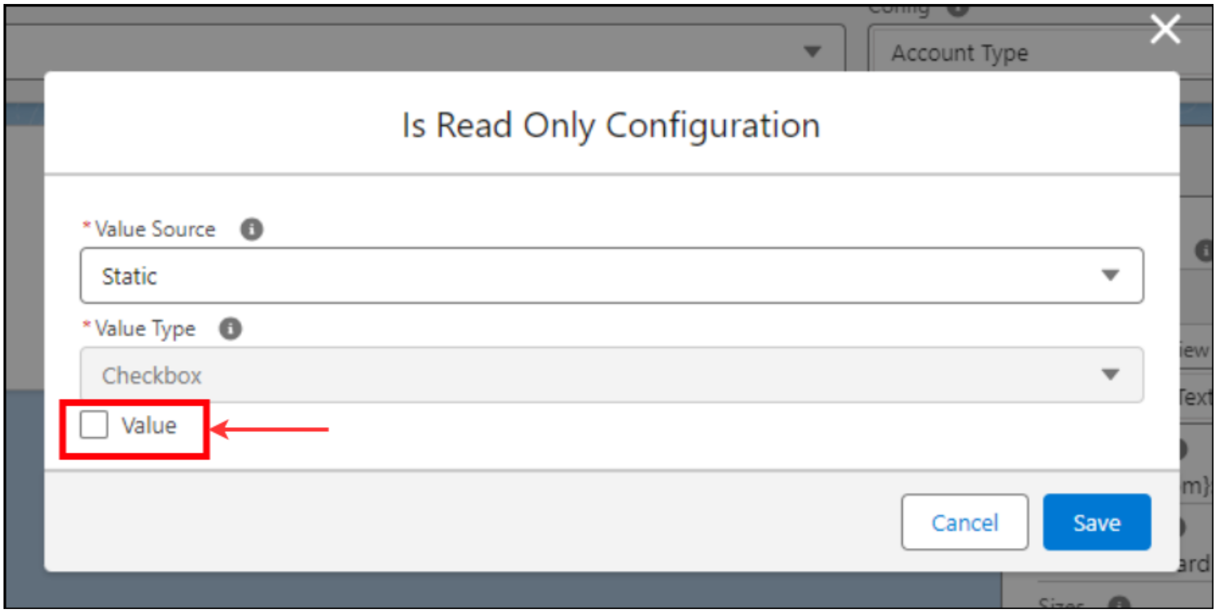


Figure 87: Read Only Config

* **Padding Location:** Defines the position of the padding for a component. The padding creates extra space within a component.

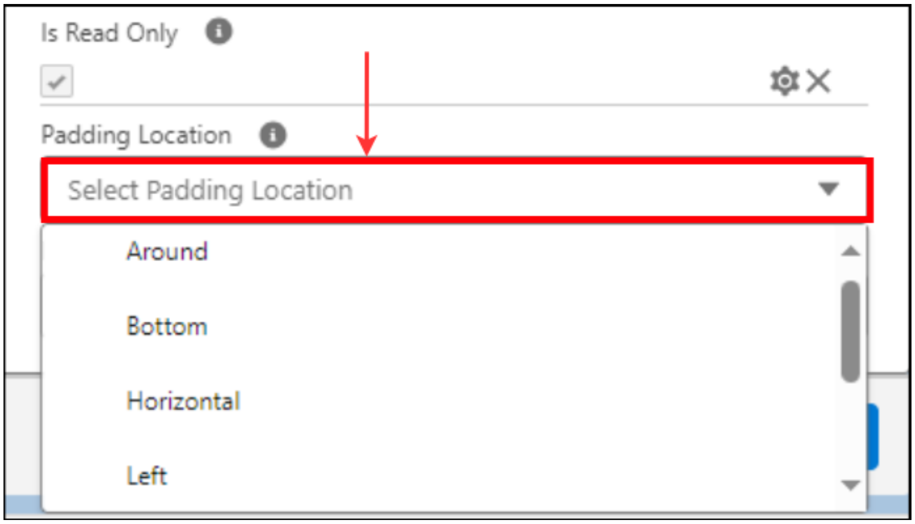


Figure 88: Padding Location Config

- Around: Creates padding around the component
- Top: Creates padding at the top of the component
- Left: Creates padding at the left side of the component

- Bottom: Creates padding at the bottom of the component
- Right: Creates padding at the right side of the component
- Horizontal: Creates padding horizontally
- Vertical: Creates padding vertically

* **Padding Size:** The padding size of the component can be set to:

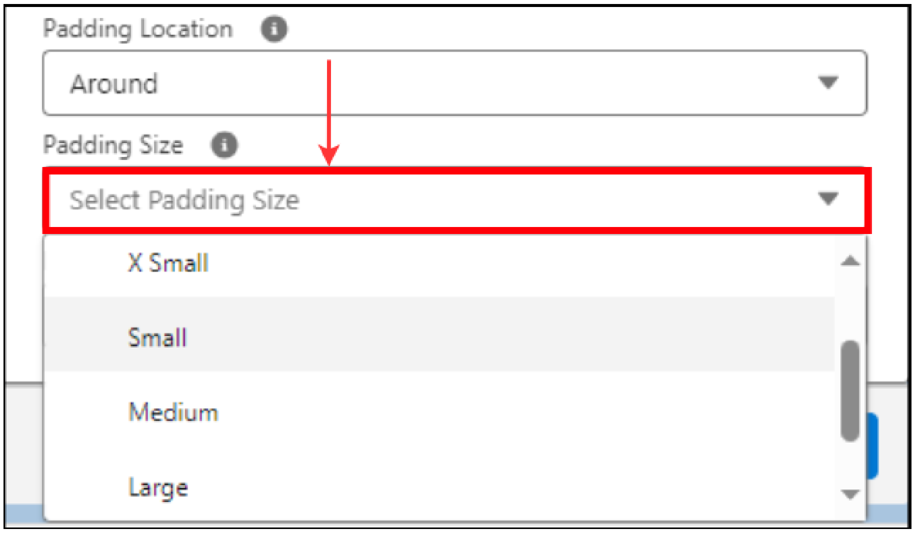


Figure 89: Padding Size Config

- None
- xxx-small
- xx-small
- x-small
- Small
- Medium
- Large
- x-large

* **Margin Location:** Defines the position of the margin for a component. margin creates extra space around a component.

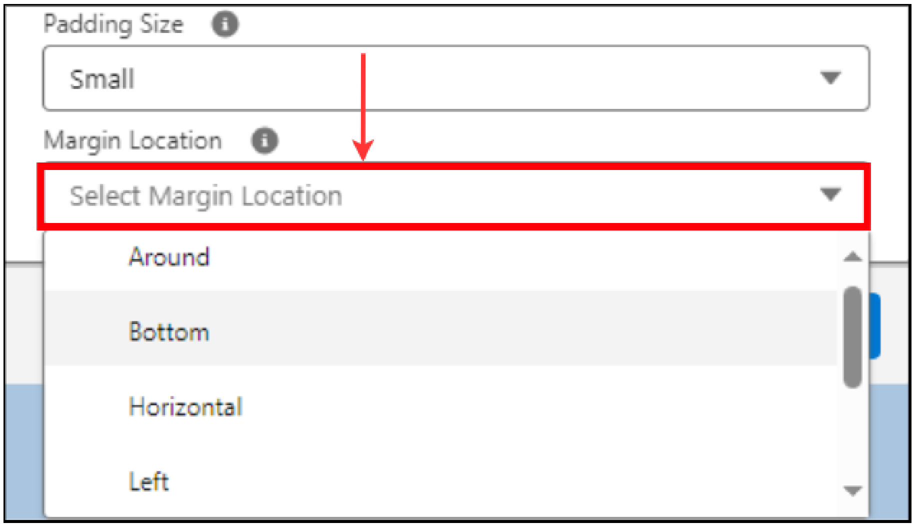


Figure 90: Margin Location Config

- **Around:** Margin gets added around the component
- **Top:** Margin gets added at the top of the component
- **Left:** Margin gets added at the left side of the component
- **Bottom:** Margin gets added at the bottom of the component
- **Right:** Margin gets added at the right side of the component
- **Horizontal:** Margin gets added horizontally
- **Vertical:** Margin gets added vertically

* **Margin Size:** The margin size of the view can be set to:

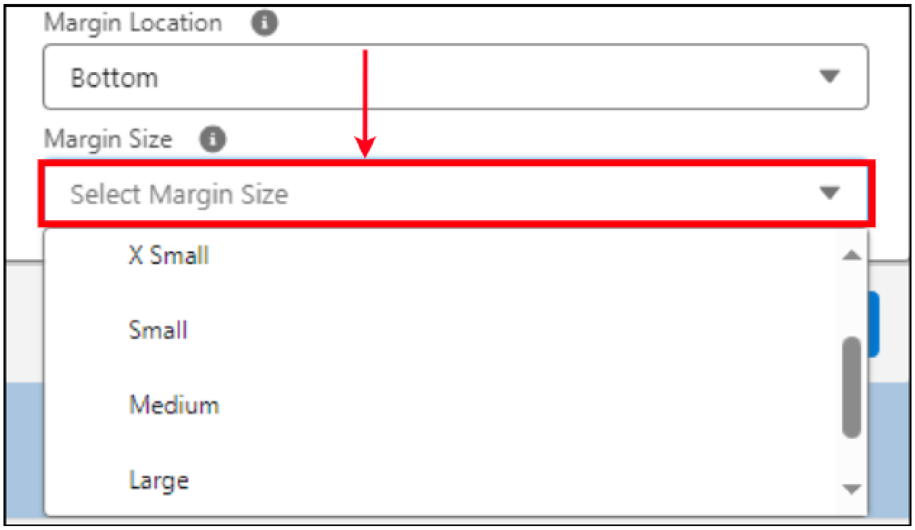


Figure 91: Margin Size Config

- None
- xxx-small
- xx-small
- x-small
- Small
- Medium
- Large
- x-large

After clicking on the Save button the config will get saved on the component config object and the user can view this button on the Record page, App page, Home Page, Flows, and Community.

– **Preview Section:**

After the user has selected and filled the fields according to their requirements, they can preview the components in the Preview section before saving them on the Record Page, App Page, Flows, and Community Page.

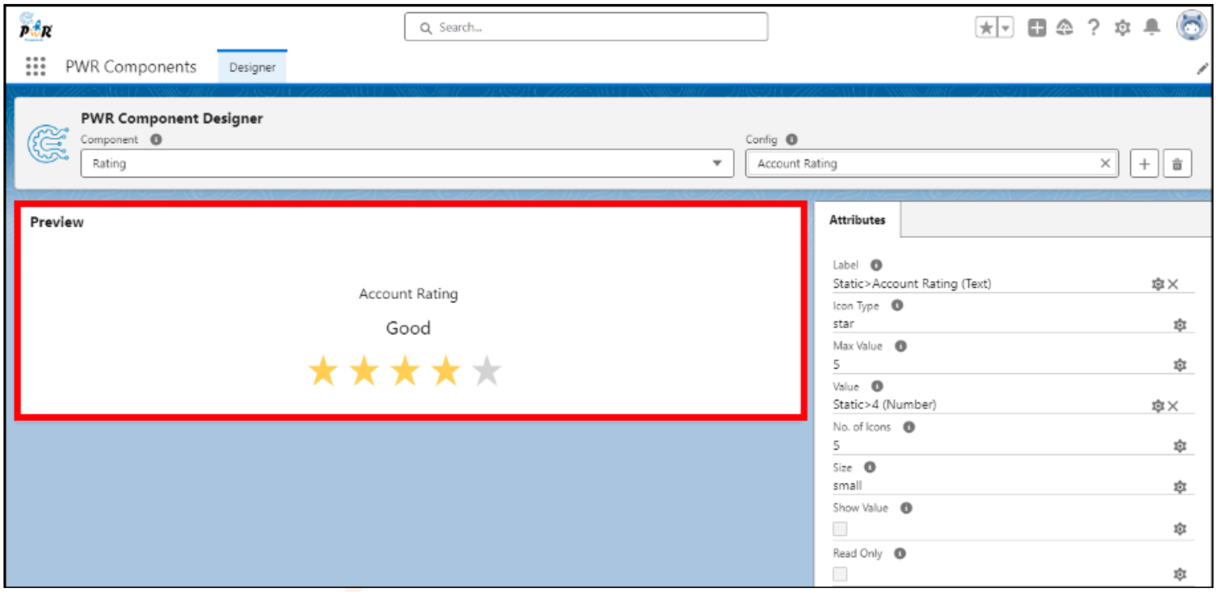


Figure 92: Preview Section

– **PWR Rating screenshots:**

⋆ PWR Rating on Record Page:

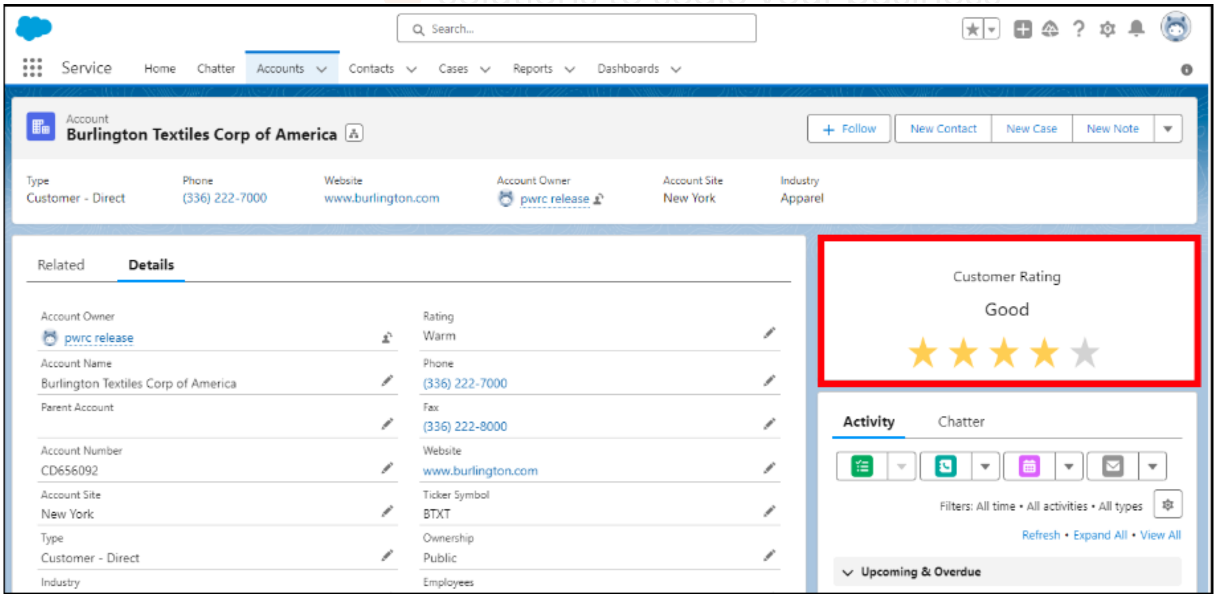


Figure 93: PWR Rating on Record Page

⋆ PWR Rating on Home Page:

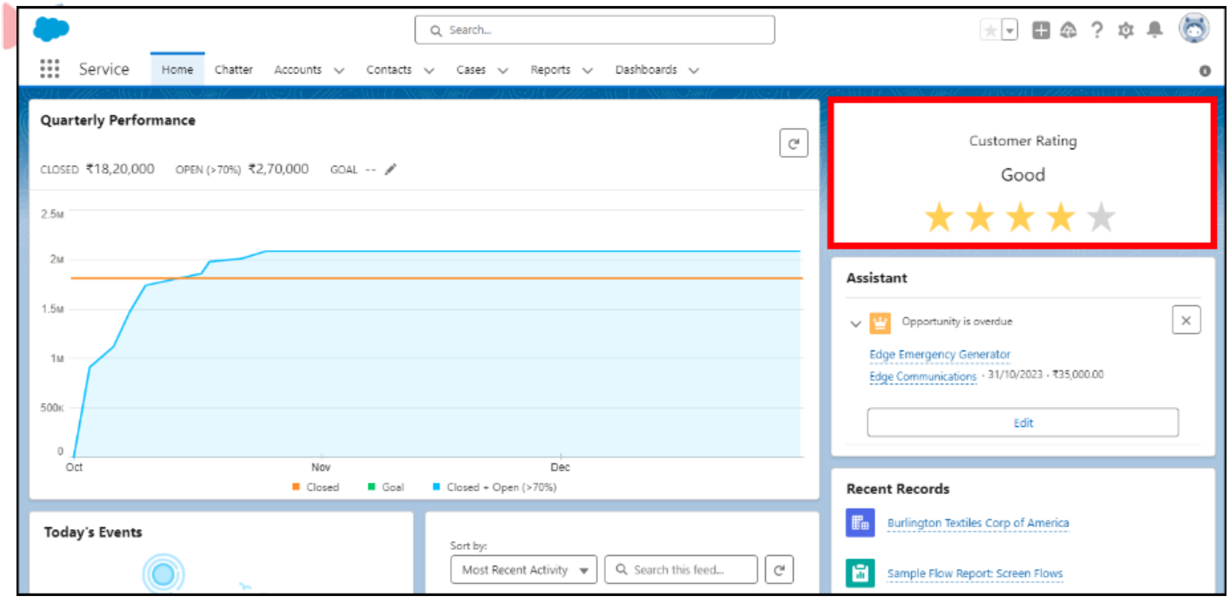


Figure 94: PWR Rating on Home Page

## V.III.5 PWR Related List:

– **Introduction:**

With the help of a related list, the user can show the related data of objects and the user can show the data up to 3 the level hierarchy.

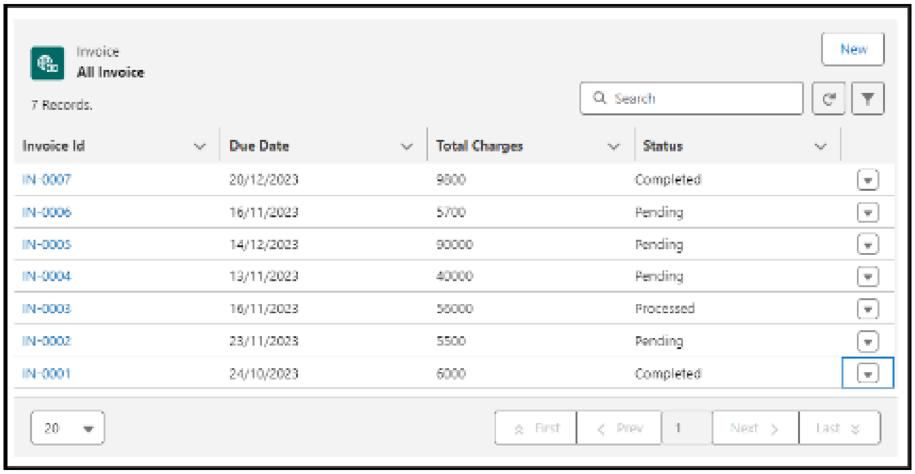


Figure 95: PWR Related List

– **Steps to create related list config:**

- Go to the designer tab

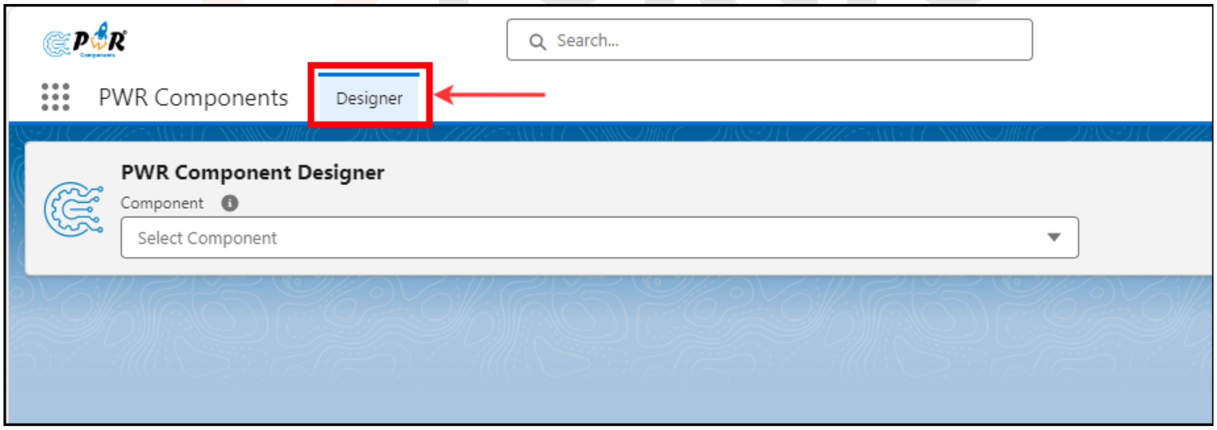


Figure 96: Designer

- Select any component from the list of components

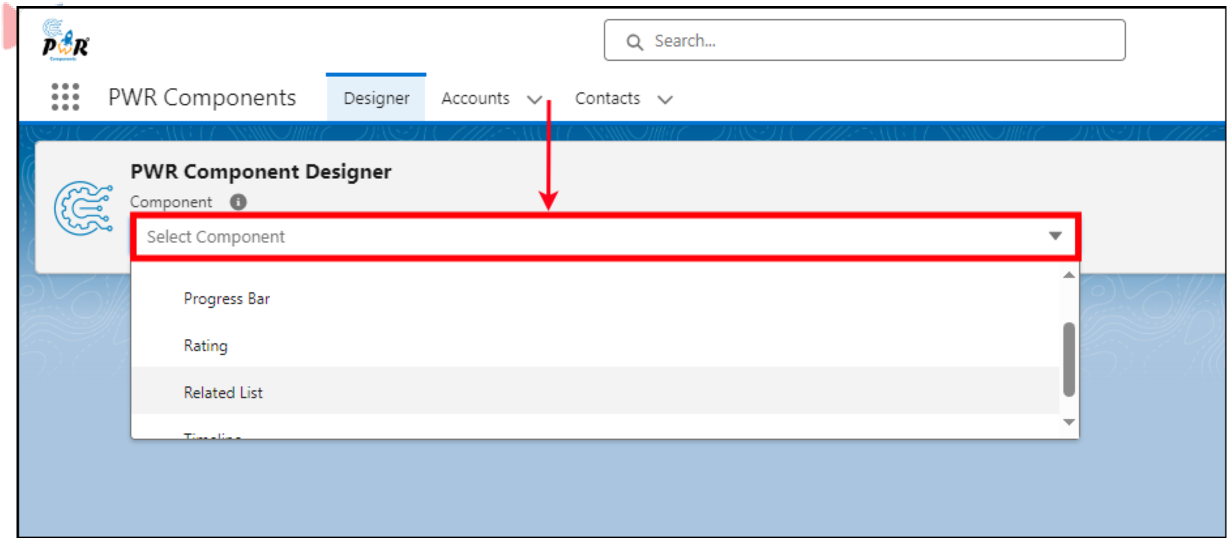


Figure 97: Select Component

* Users can use existing Config, or users can create a new Config

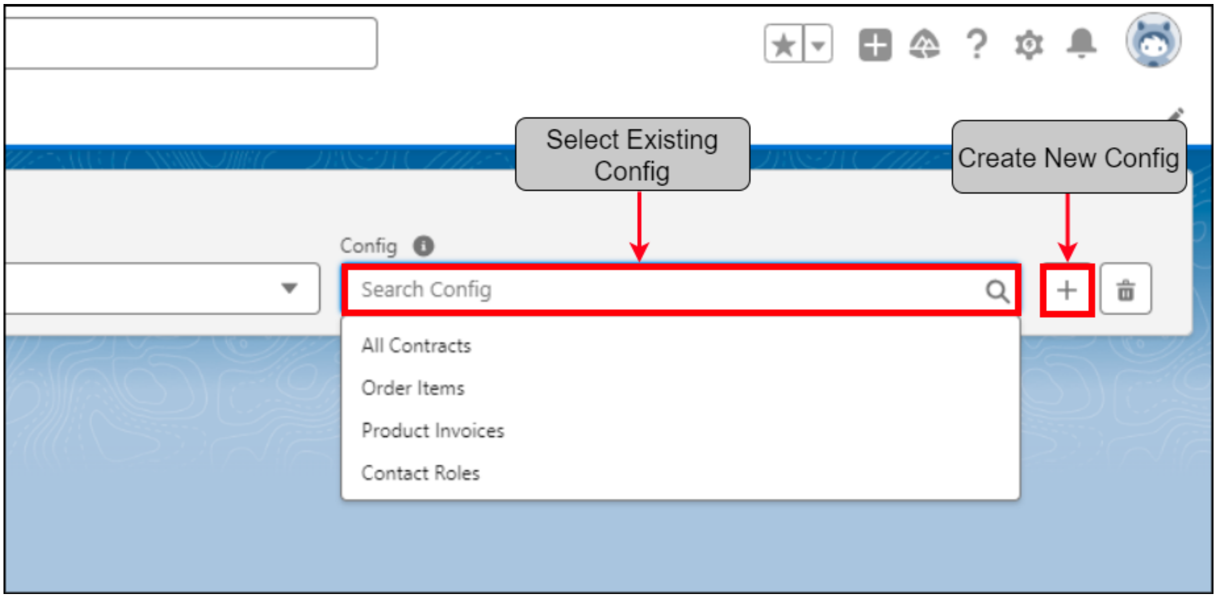


Figure 98: Select Config

* If the user creates a new config

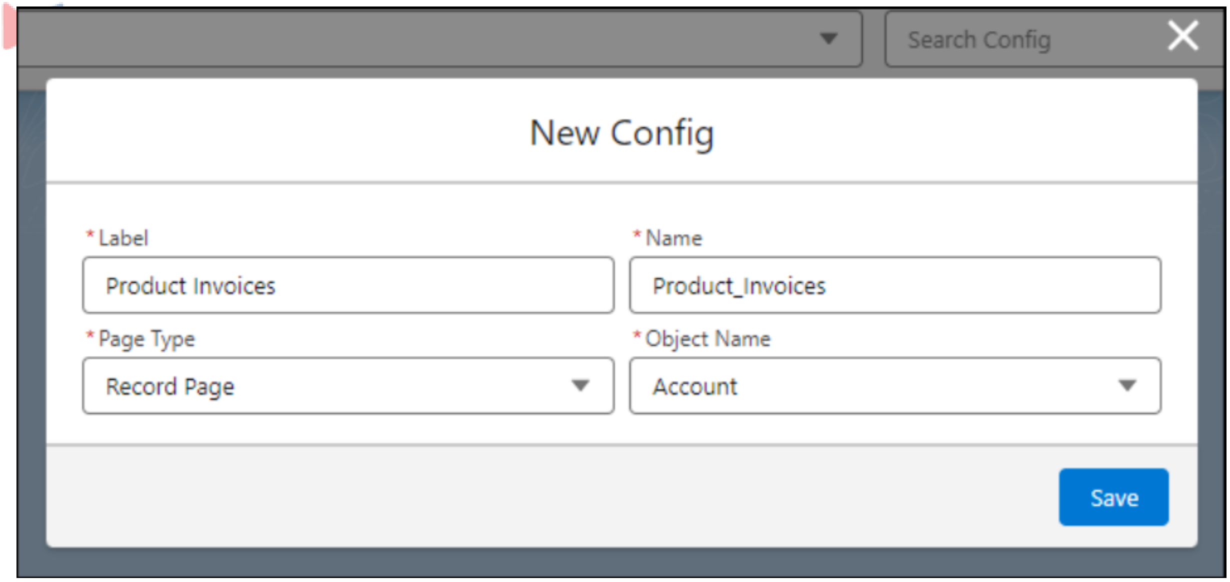


Figure 99: Select Config

- **Label:** Here user can set the title for the config of the component
- **Name:** The API name of the label is shown here
- **Page Type:** Here user can select page type as record page or non-record page; depending on which page the user wants to show the component
- **Non-Record Page:** By choosing the non-record page user can use this config on the home page and app page.
- **Record Page:** By choosing the record page user can use this config on a related record object
- **Object Name:** Here user can choose any object on which that created config will be shown

**Here are the details around different attributes, and which valid information can be input for the same.**

– **Attributes:**

  * **Record Object:** The object name is auto-populated here as you select the object at the time of config creation.
  * **Record To Preview:** This feature helps you preview and evaluate the related list's appearance according to the selected record.

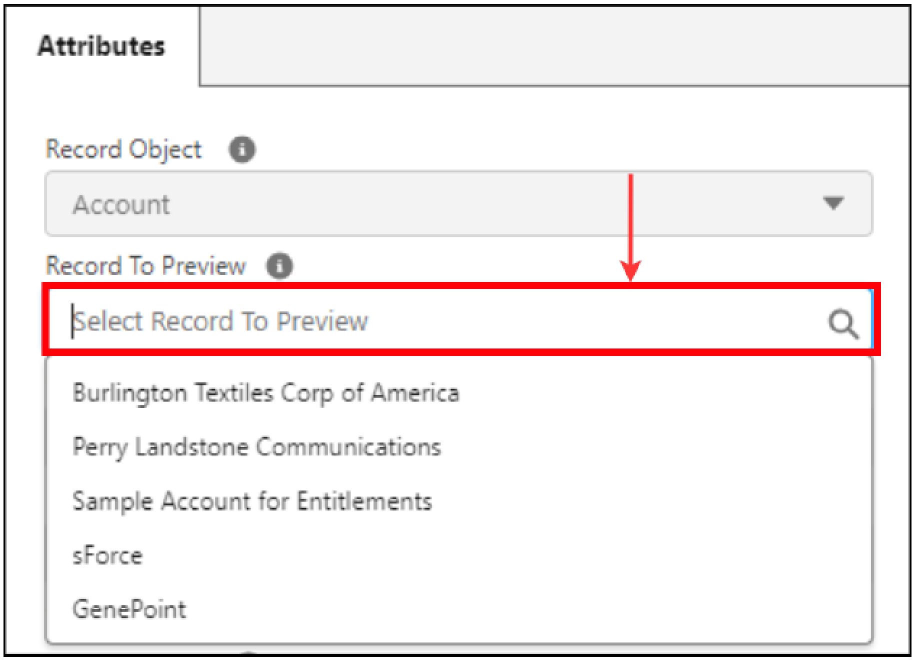


Figure 100: Select Record to Preview

- **Child relationship:** The ”Child relationship” field allows you to choose the child object you want to display in the related list.

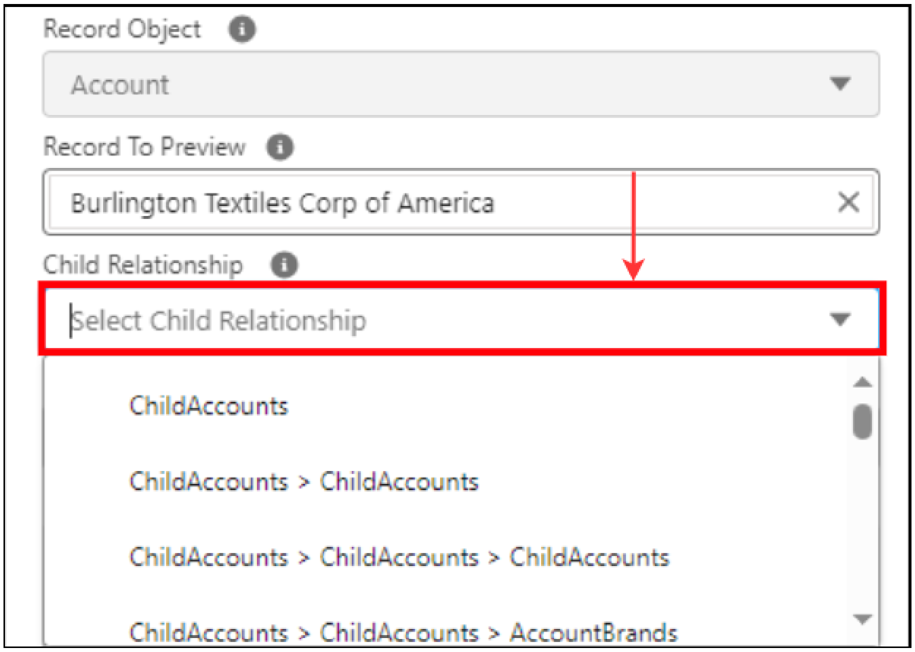


Figure 101: Child Relationship Config

- **Object Name:** Selected child relationship object name will be auto-populated here in read-only format. If no child relationship object is selected, then by default, the record object is auto-populated in this field, or else here you have to choose an object.

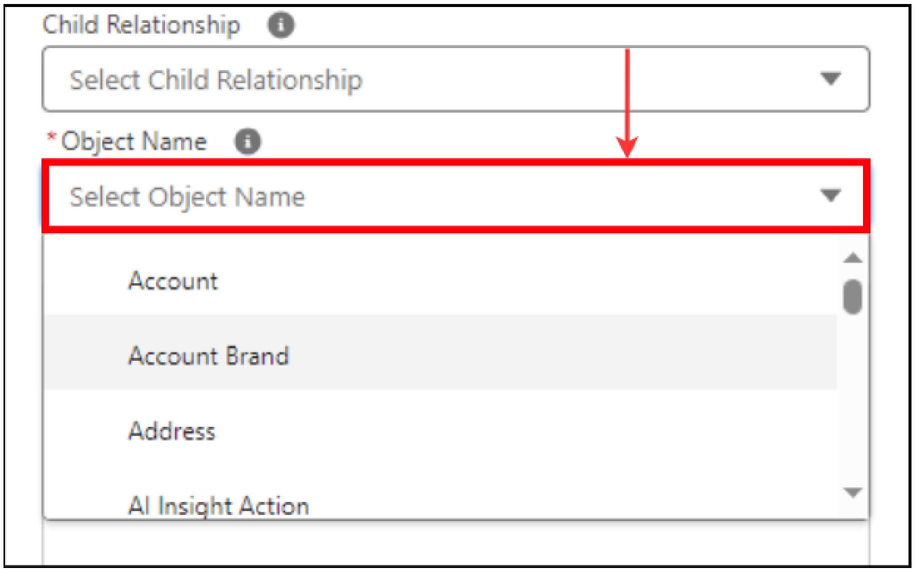


Figure 102: Object Config

* **Display Fields:** Display Fields allow you to choose the fields from the display fields that you want to display in the related list. The fields that you selected here are displayed in the selected fields.

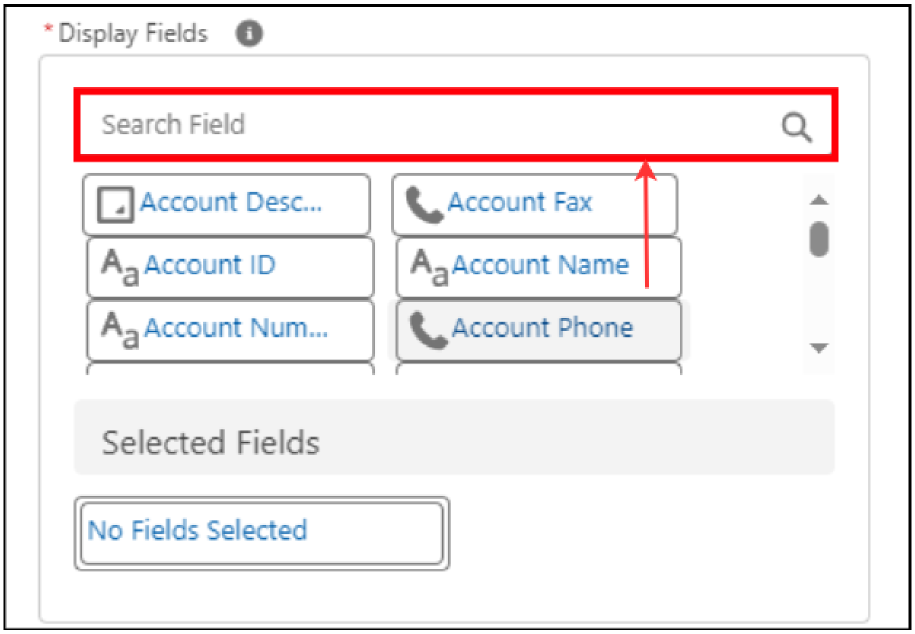


Figure 103: Display Fields Config

* **Show Search:** The "Show Search" checkbox allows you to enable or disable the search functionality within the related list. Search Bar will be displayed at the top of the related list.

* **Search Fields:** ”Search Fields” allows you to customize the search experience by choosing which fields you want to be included in the search results for this related list.

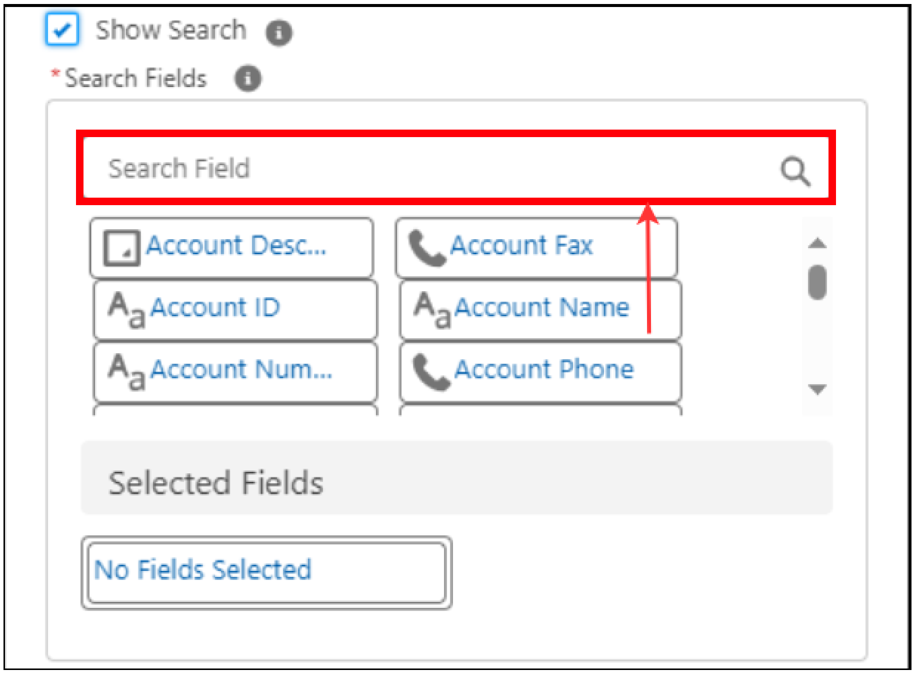


Figure 104: Search Fields Config

* **Filter Criteria:** The ”Filter Criteria” section allows you to customize the display of the related list based on specific criteria. Criteria can be set from multiple value sources


Figure 105: Filter Criteria Config

* **Record Selection:** The “Record Selection” is a picklist type field in which it contains “No Selection”, “Single Select”, “Multi Select” and by default, the “No Selection” is selected
* **Single Select:** The Single Select activates the radio button for choosing a single record

- ⋆ **Multi Select:** The Multi-select activates the checkboxes for choosing multiple records at once
- ⋆ **Max Selected Record Count:** The ”Max Selected Record Count” field allows you to set a maximum limit for the number of records that can be selected in this related list.
- ⋆ **Infinite Scrolling:** Enabling the ”Infinite Scrolling” checkbox will assist users in scrolling through the data table and retrieving records according to scrolling.
- ⋆ **Page Size:** The ”Page Size” option allows you to specify the number of records displayed per page in the related list.

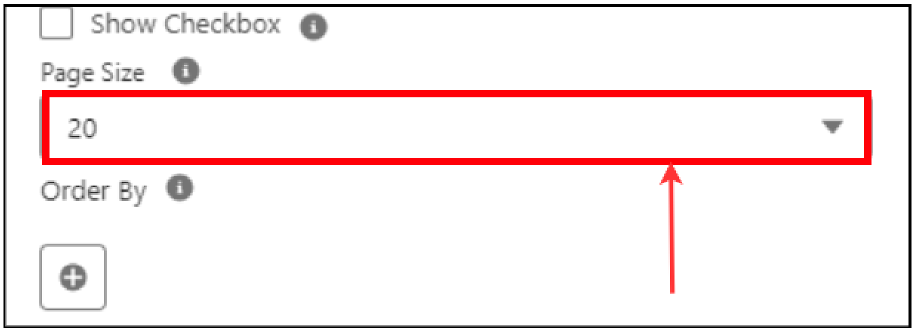


Figure 106: Page Size Config

- ⋆ **Summarize Field:** ”In Related List, we employ a ’summarize’ field to calculate data based on various filters such as static, all records, and custom filters. Users have the flexibility to choose from different aggregate types, including sum, min, max, average, count, and distinct count. Depending on the selected aggregate type and the applied filter conditions, the ’summarize’ field generates calculated results. This functionality allows users to obtain specific insights and metrics from the data table by aggregating information according to their preferences and requirements”
- ⋆ UI of the summarize field using this UI user can easily config the summary metrics

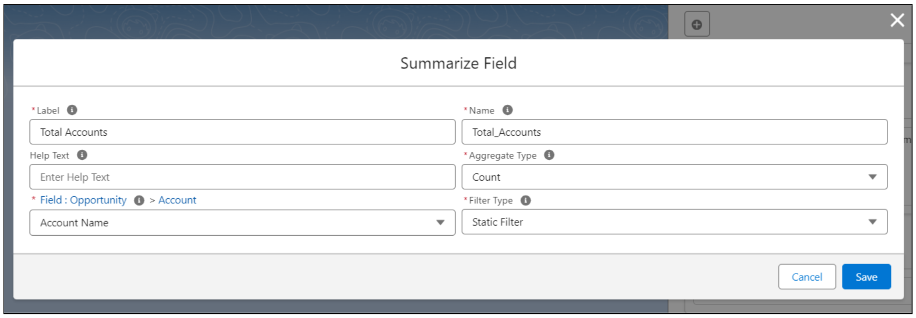


Figure 107: Summarize field Config

- ⋆ From here user can give the label, and help text to the summarize field

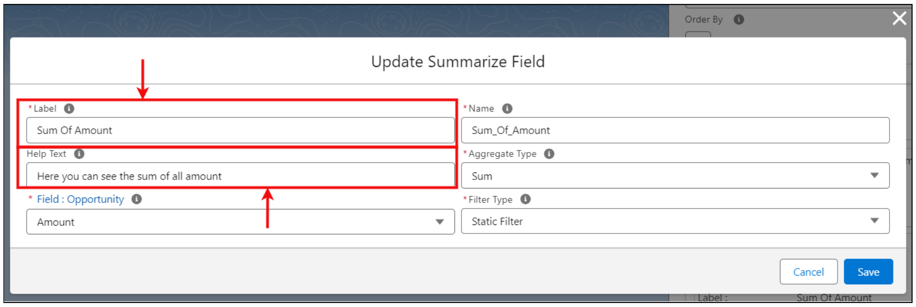


Figure 108: Summarize field Config

* Using Aggregate Type user can select the aggregate type like sum, max, min, etc., and apply the summary metrics on the data

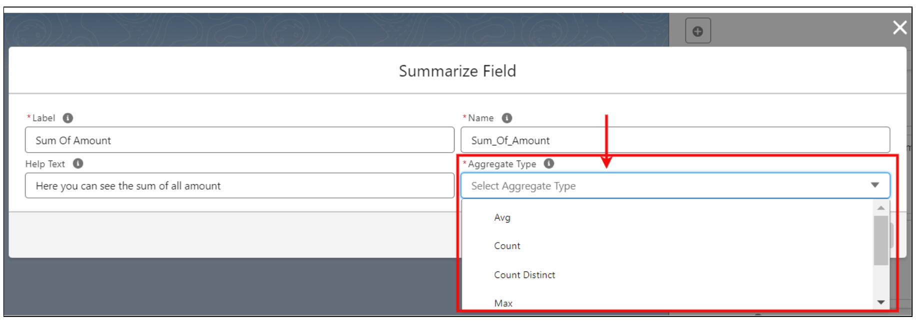


Figure 109: Aggregate of Summarize field

* Here user can select the field on which the user wants to create summary metrics so, from the picklist user can select the object field

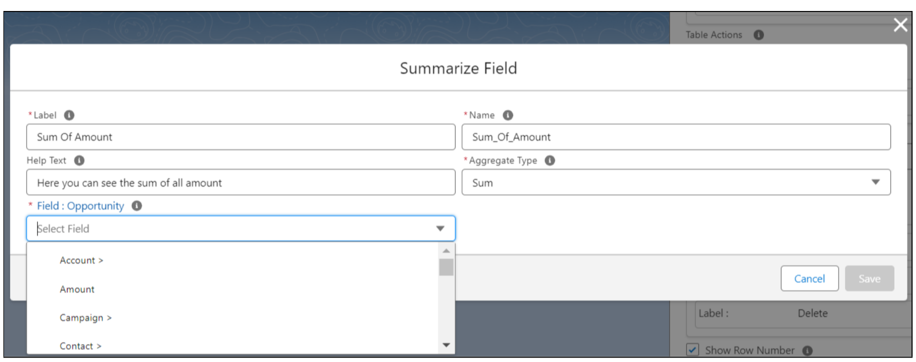


Figure 110: Field for Summarize config

* From here user can select the filter type as mentioned below:

1. Custom filter
2. All records
3. Static filter

so, based on this filter user can show summary metrics using a data table.

* **Custom filter:** When the user selects the custom filter then the user can apply the custom logic for the summary metrics and this logic overrides the static filter applied on the data table and shows the summarize metric
* **All records:** When the user selects the all records filter the summarize filter is applied to the total data present in the object and then the summary calculation proceeds
* **Static Filter:** When the user selects the static filter then summarizes fieldwork according to the static filter and shows the data according to the static filter

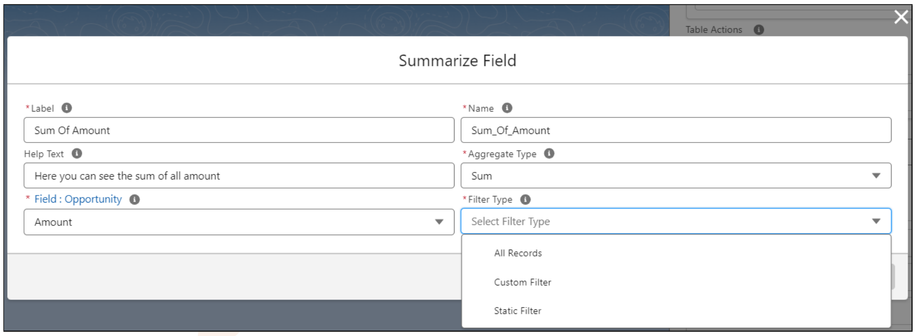


Figure 111: Filter of summarize config

* Here user can see the summary metrics of all the config of the summarize field with the proper name of the config and calculation of the summary field

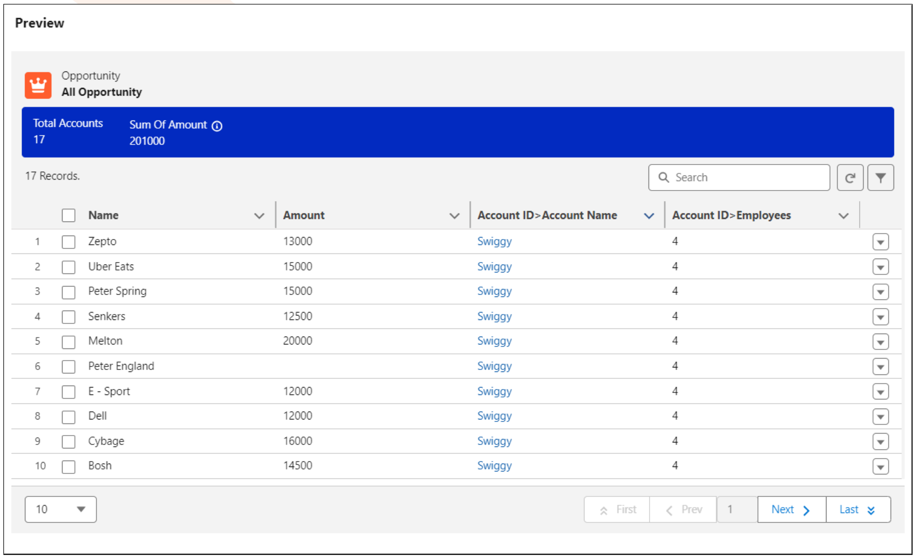


Figure 112: Summarize Column

* **Order By:** "Order By" allows the user to specify the way they want the related list items to be displayed; "Order " allows the user to organize the items in this related list either in ascending or descending, by default ascending is selected.


Figure 113: Order By Config


Figure 114: Order By value



* **Table Action:** The user can create multiple table actions on the related list. By default, only new actions are displayed, which are visible in the top right corner.


Figure 115: Table Action Config

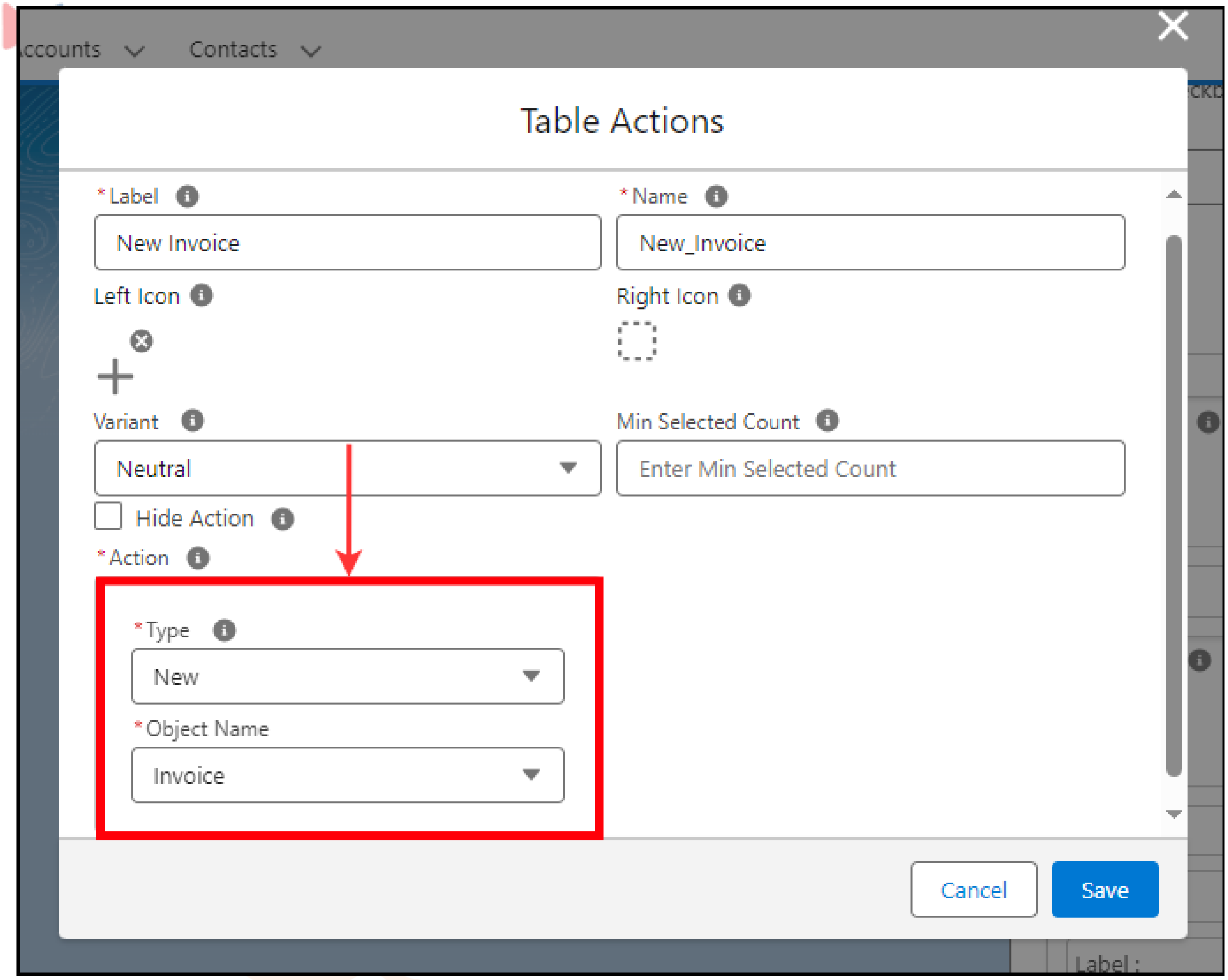




Figure 116: Table Action Value

- **Label:** The Label provide a name or description for a specific table action
- **Name:** API name of the label is shown here.
- Left Icon: This feature enables you to display an icon on the left side of an action, adding visual context and clarity to the action it represents.
- **Right Icon:** This feature enables you to display an icon on the right side of an action, adding visual context and clarity to the action it represents.
- **Variant:** Allows to show Buttons in different color formats.
- **Neutral:** Neutral is the default variant of the Button which is displayed in white color
- **Brand:** Brand is a blue Button, used to draw attention to the primary action on a page
- **Outline-brand:** Outline-brand is similar to the brand variant, but the color is used for the label and border only, not the Button color

- **Destructive:** Destructive is a red Button used to warn that the action has a negative effect
- **Success:** Success is a green Button used to indicate a successful action
- **Text Destructive:** This button is used to indicate a destructive action to the user

* **Min Selected Count:** The ”Min Selected Count” option allows the user to specify the minimum number of items that must be selected in the related list.
* **Hide Action:** The Hide Action feature allows the user to control the visibility of specific items within a related list.
* **Action:** User can create the actions here. For example, creating new records, navigating to any URL, editing any record, etc.
  - **Type:** Here user can select any specific type for that particular action
  - Bulk Delete
  - Custom Component
  - Edit
  - Global Action
  - List View
  - New
  - Object Page
  - Quick Action
  - Redirect
  - Tab
  - View
  - Delete
* **Row Actions:** User can create the row actions here, Actions to be performed at the record level for the respective record by default three actions are created view, edit, delete.

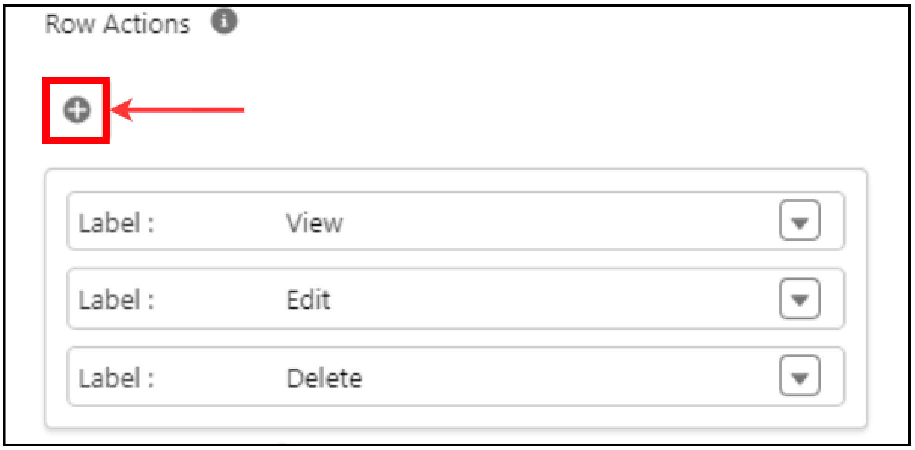


Figure 117: Row Action Config

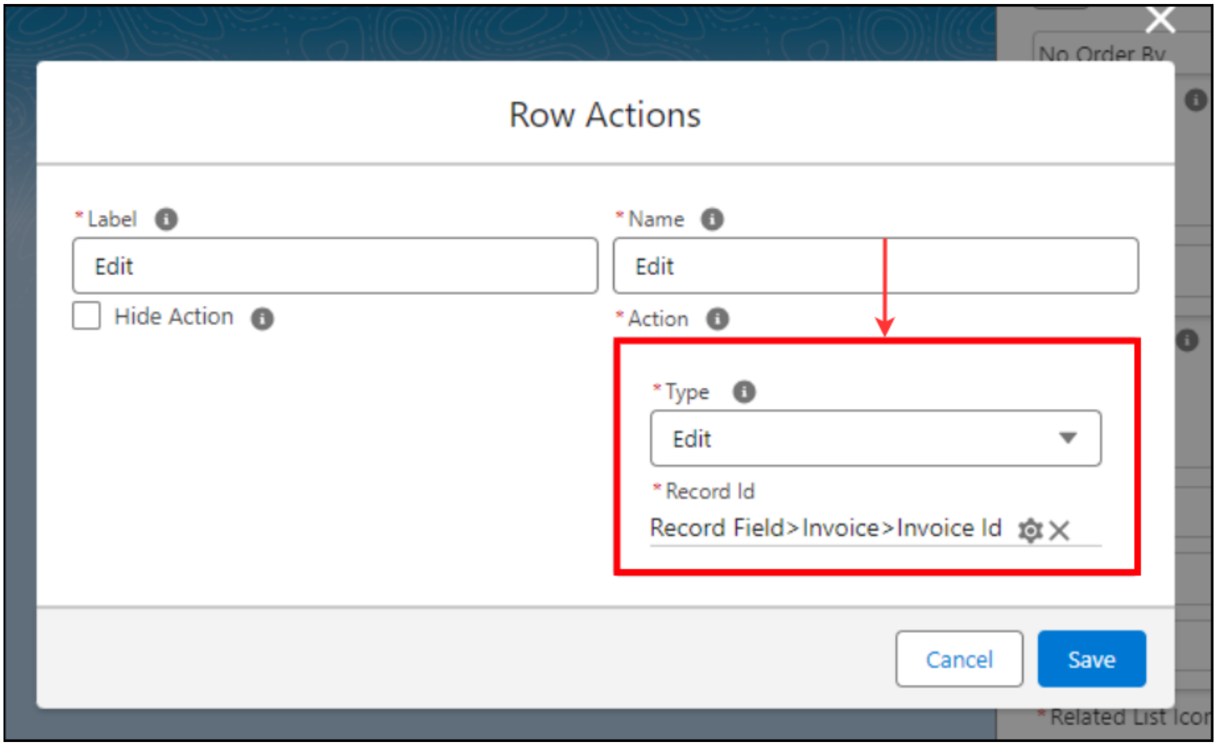


Figure 118: Row Action Value

- **Label:** The Label provide a name or description for a specific table action.
- **Name:** API name of the label is shown here.
- **Hide Action:** The Hide Action feature allows you to control the visibility of specific items within a related list.
- **Action:** You can create the actions here. For example, creating new records, navigating to any URL, editing any record, etc.
- **Type:** Here user can select any specific type for that particular action

After clicking on the Save button the config will get saved on the component config object and the user can view this button on the Record page, App page, Home Page, Flows, and Community.

* **Preview Section:** After the user has selected and filled the fields according to their requirements, they can preview the components in the Preview section before saving them on the Record Page, App Page, Flows, and Community Page.

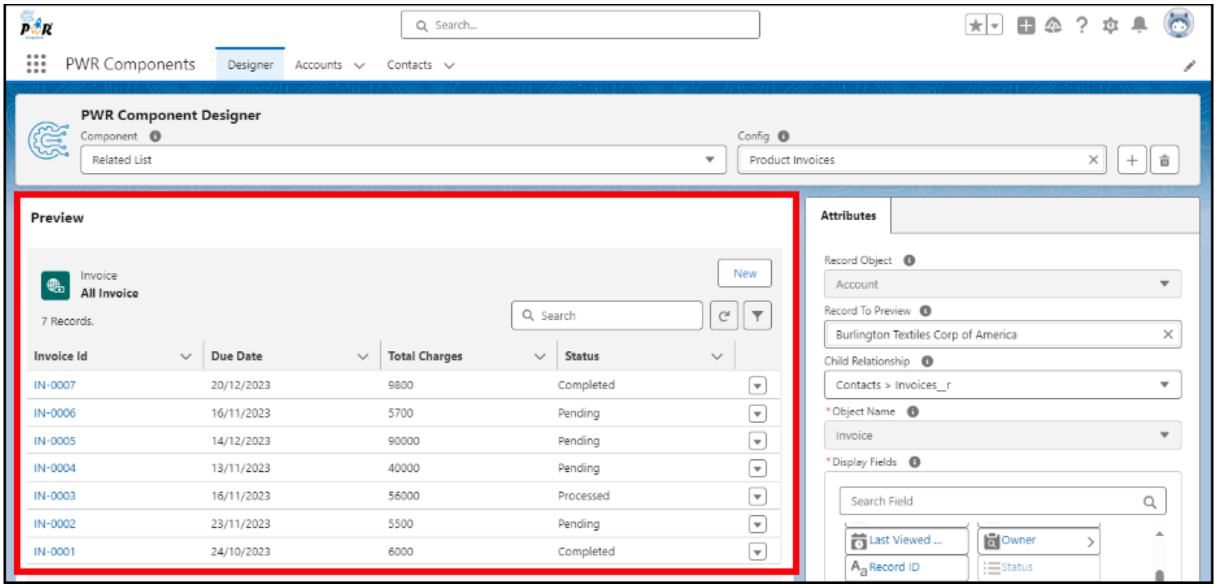


Figure 119: Preview Section

## – PWR Related screenshots:

* PWR Related List on Record Page:

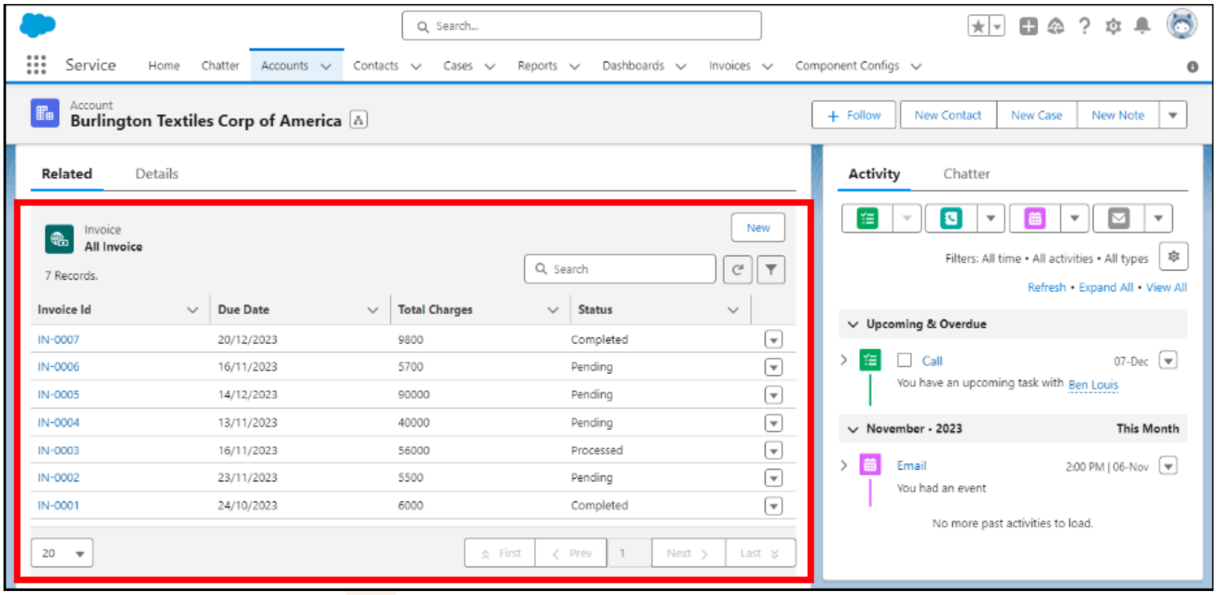


Figure 120: PWR Related List on Record Page

* PWR Related List on Home Page:

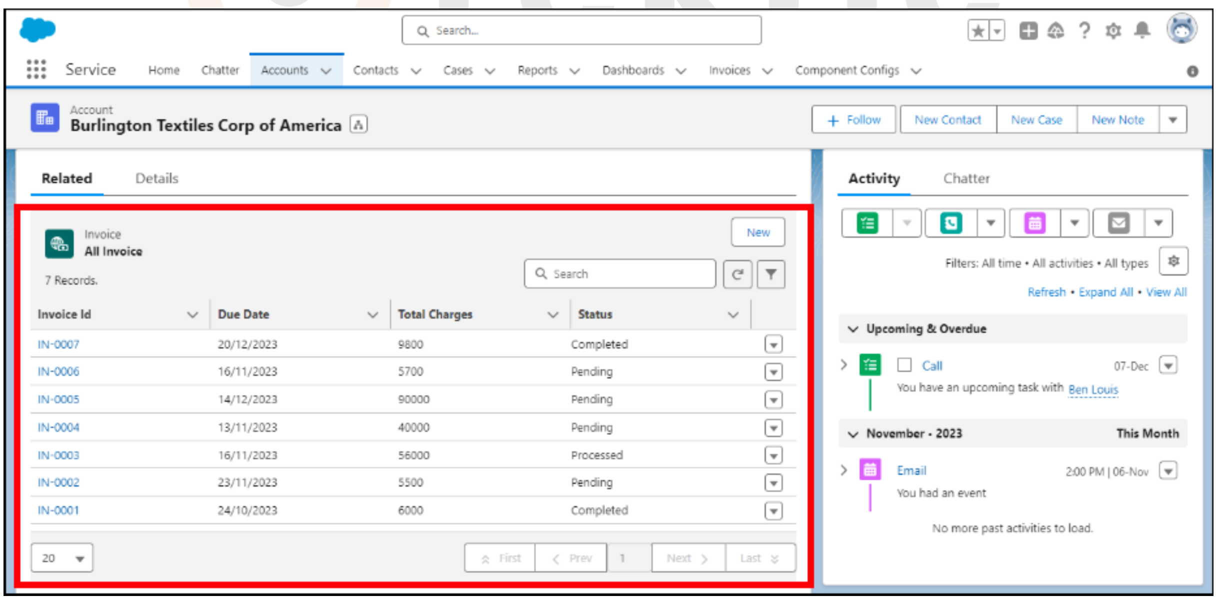


Figure 121: PWR Related List on Home Page

## V.III.6 PWR Timeline

– **Introduction:**

The timeline in Salesforce displays a chronological list of these activities, making it easier for users to understand the history and progression of a record.

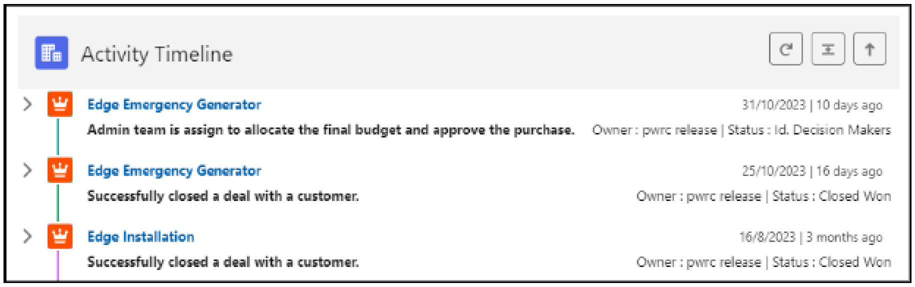


Figure 122: PWR Timeline

– **Steps to create timeline config:**

* Go to the designer tab

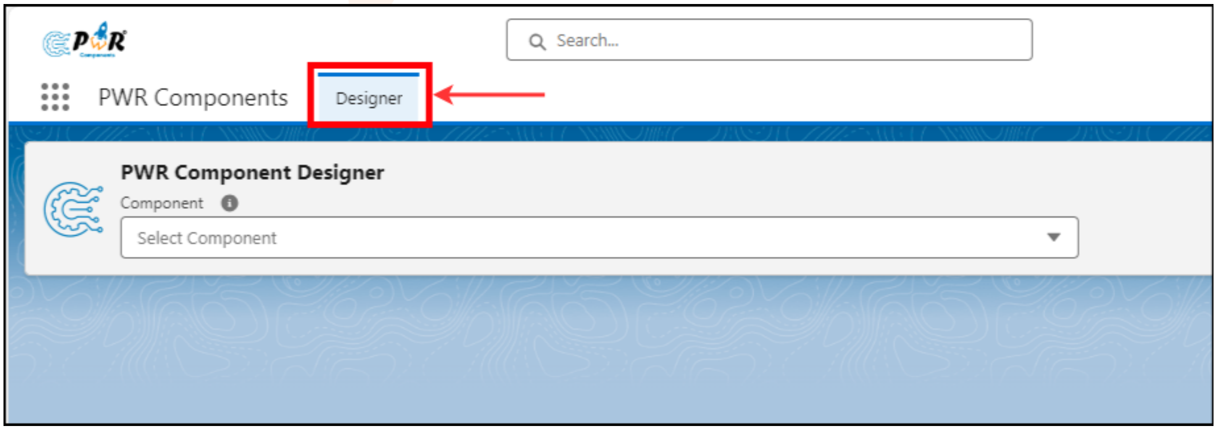


Figure 123: Designer

* Select any component from the list of components

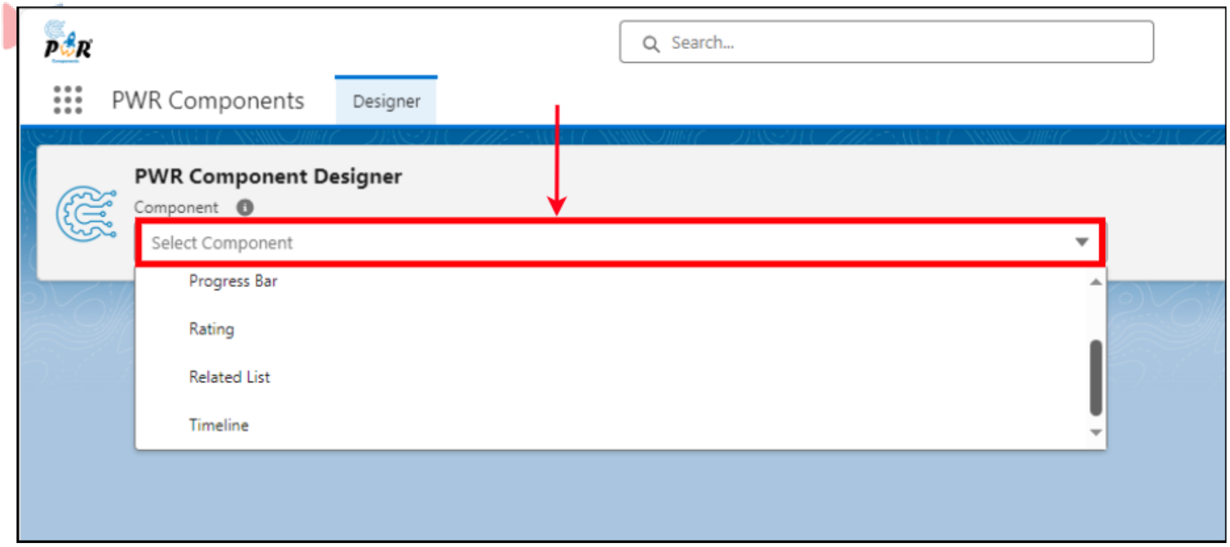


Figure 124: Select Component

* Users can use existing Config, or users can create a new Config

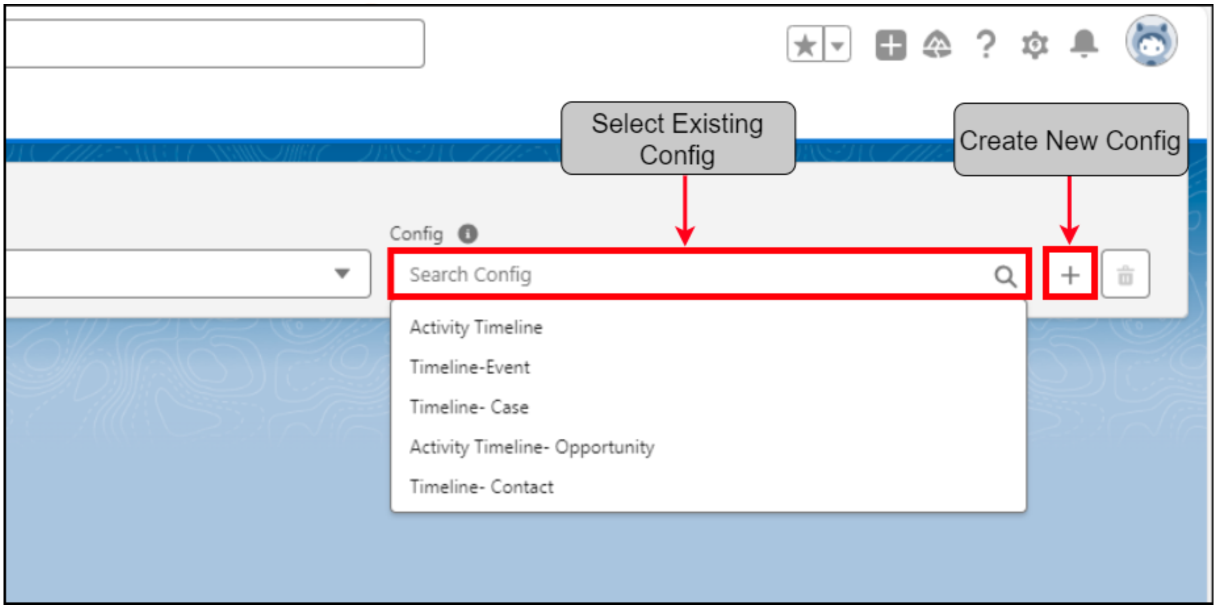


Figure 125: Select Config

* If the user creates a new config

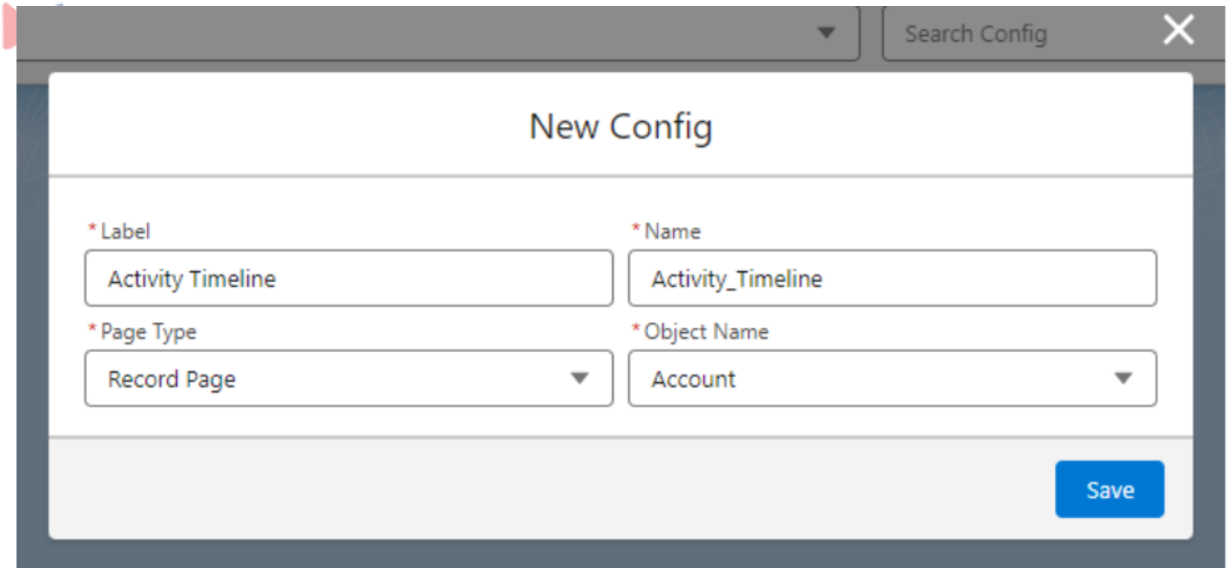


Figure 126: Create Config

- **Label:** Here user can set the title for the config of the component
- **Name:** The API name of the label is shown here
- **Page Type:** Here user can select page type as record page or non-record page; depending on which page the user wants to show the component
- **Non-Record Page:** By choosing the non-record page user can use this config on the home page and app page
- **Record Page:** By choosing the record page user can use this config on a related record object
- **Object Name:** Here user can choose any object on which that created config will be shown

**Here are the details around different attributes, and which valid information can be input for the same.**

– **Attributes:**

  * **Record Object:** The object name is auto-populated here as you select the object at the time of config creation.
  * **Record To Preview:** This feature helps you preview and evaluate the related list's appearance according to the selected record.

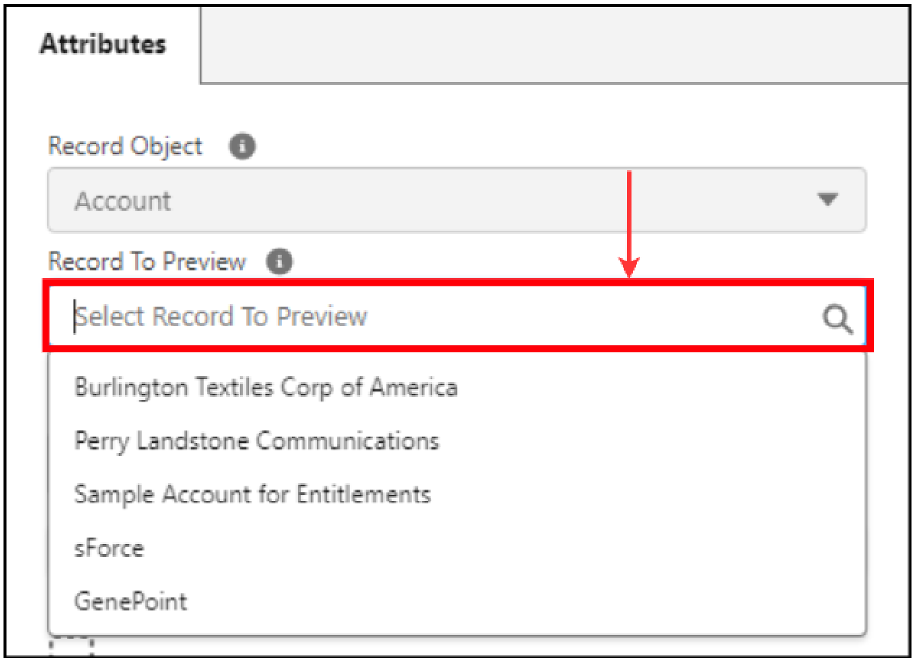


Figure 127: Record to Preview Config

* **Related Object:** This is an array type where we can configure activity details for different objects, as we can see multiple activities of different objects on a single timeline, once clicked, we will have the following things to configure:


Figure 128: Related Object Config

- **Child relationship:** The "Child relationship" field allows you to choose the child object you want to display in the related list.

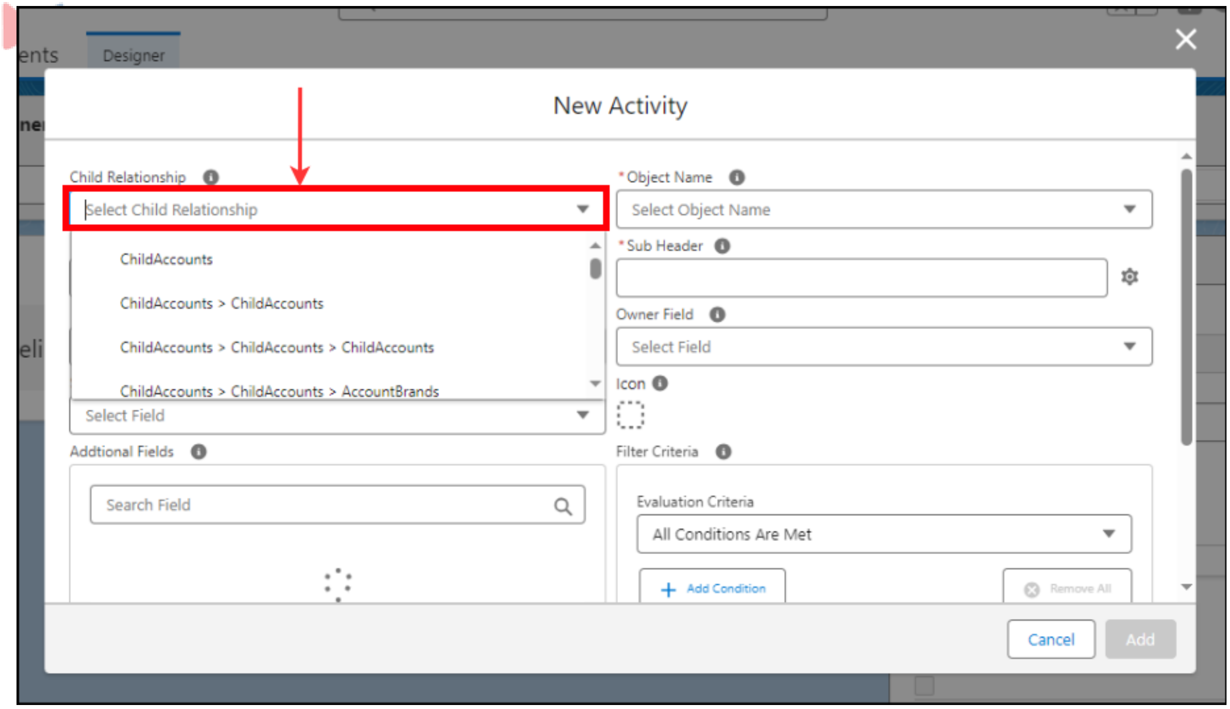


Figure 129: Child Relationship Config

- **Object Name:** Selected child relationship object name will be auto-populated here in read-only format. If no child relationship object is selected, then by default, the record object is auto-populated in this field, or else here you have to choose an object.

* Merge Text: When user clicks on the gear icon of the header and Sub Header then,

  - For the Activity Timeline, the user requires functionality that merges the text and displays the merged values.
  - This will be a collection of property binders that will allow a user to use multiple property binder values along with some static ones.
  - To provide the sub header and header user will need this merge text component because sub header contains some static text and some dynamic text which changes according to records.

* **Header:** We can use merge text for the Header, we can add static text along with some values from the field, we can use merge text component to create a custom text for the header, also we can see if we can make the whole header as a hyperlink of the record whose details are being shown

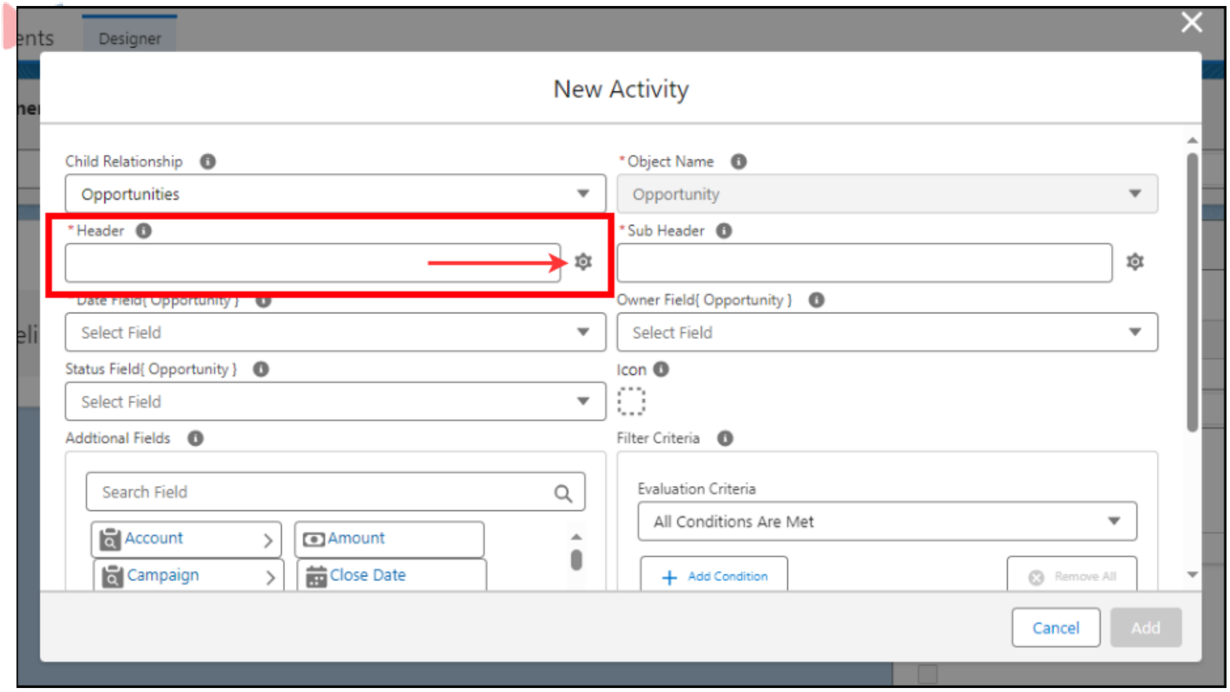


Figure 130: Header Config

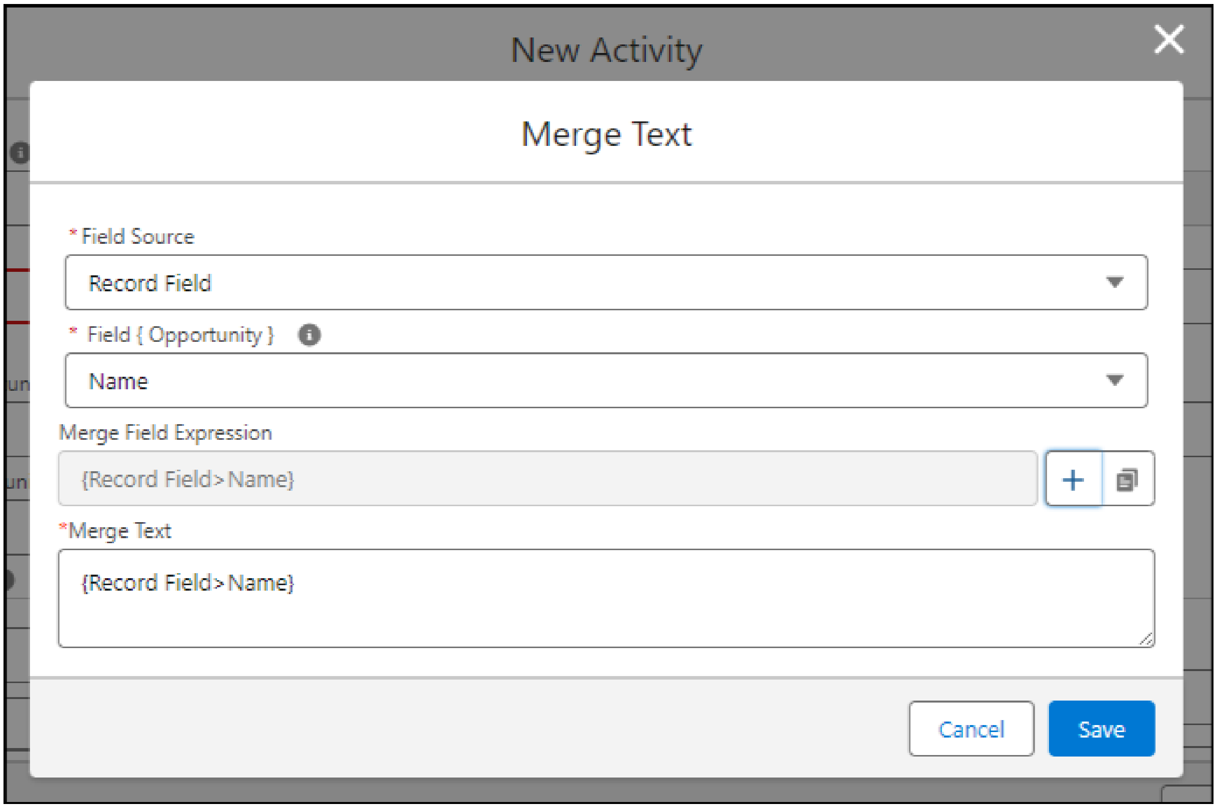


Figure 131: Merge text

* **Sub header:** User can use merge text for the sub header so that the user can

add static text along with the dynamic fields.

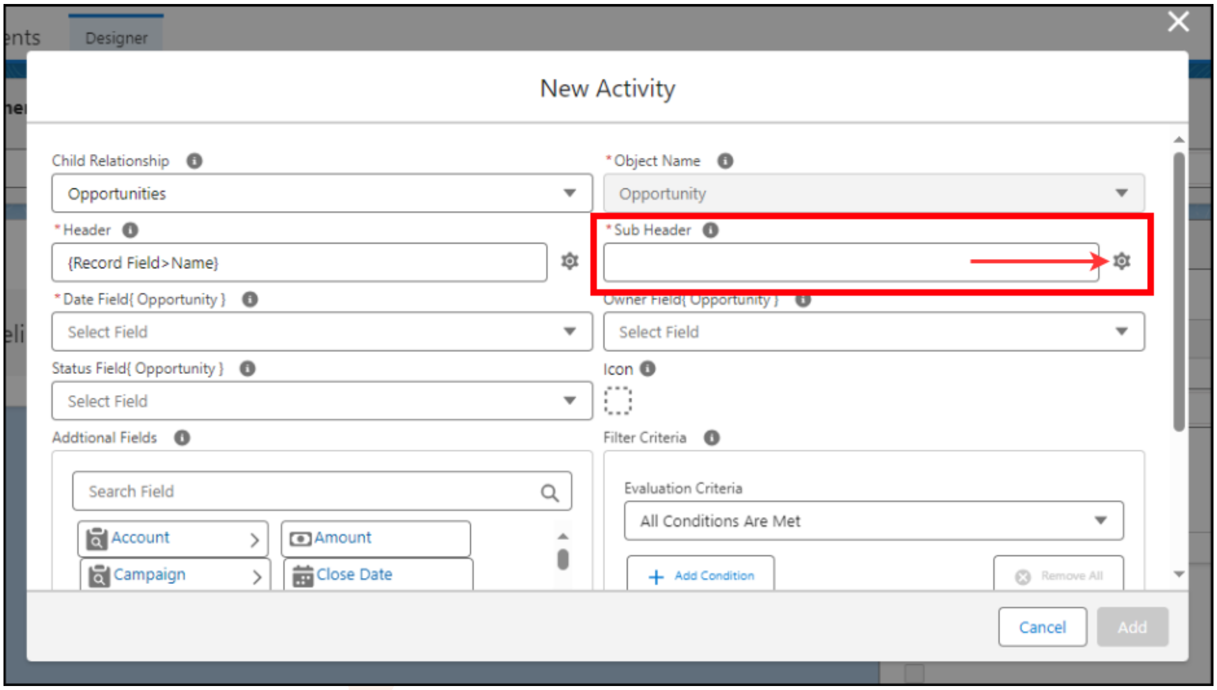


Figure 132: Sub Header Config

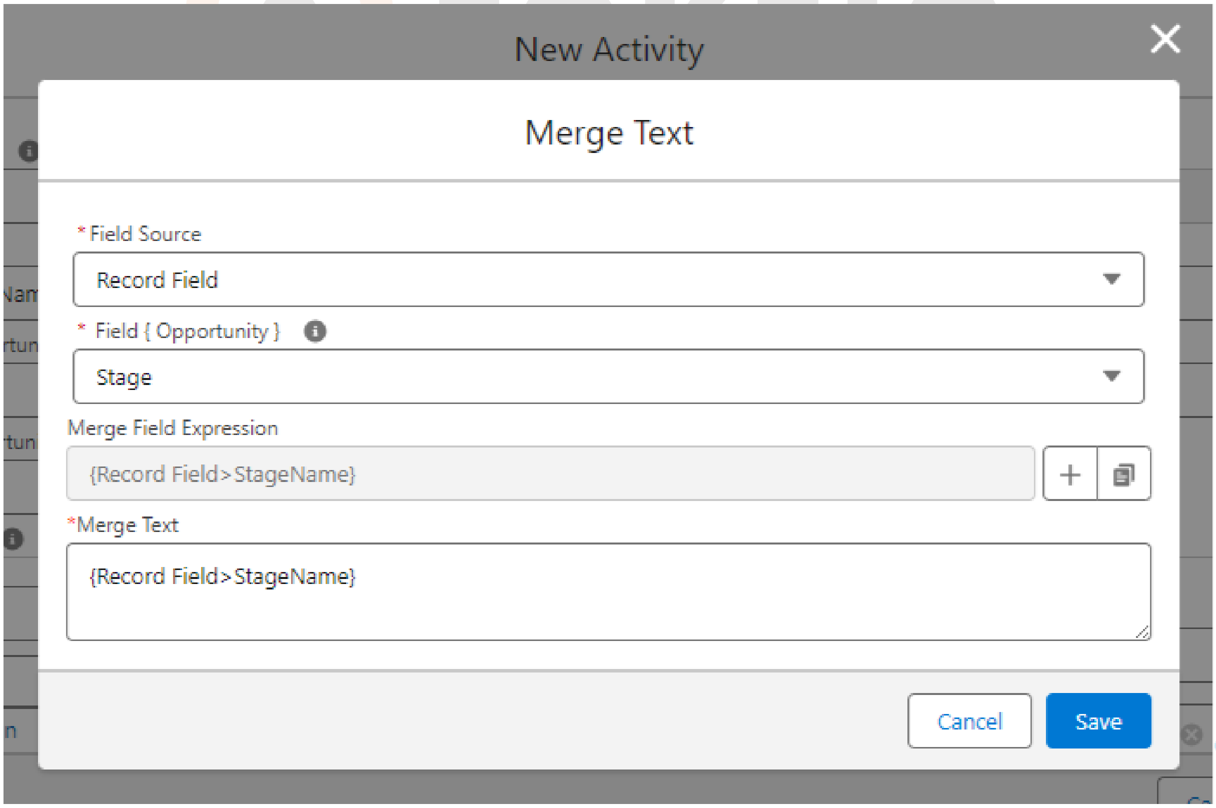


Figure 133: Merge text

* **Date Field:** This is the field that will be used to order the records on the Timeline, if we consider the example of Case as a source object and from it if we select the date field as a close date then the records in the timeline will be sorted using the close date field only.

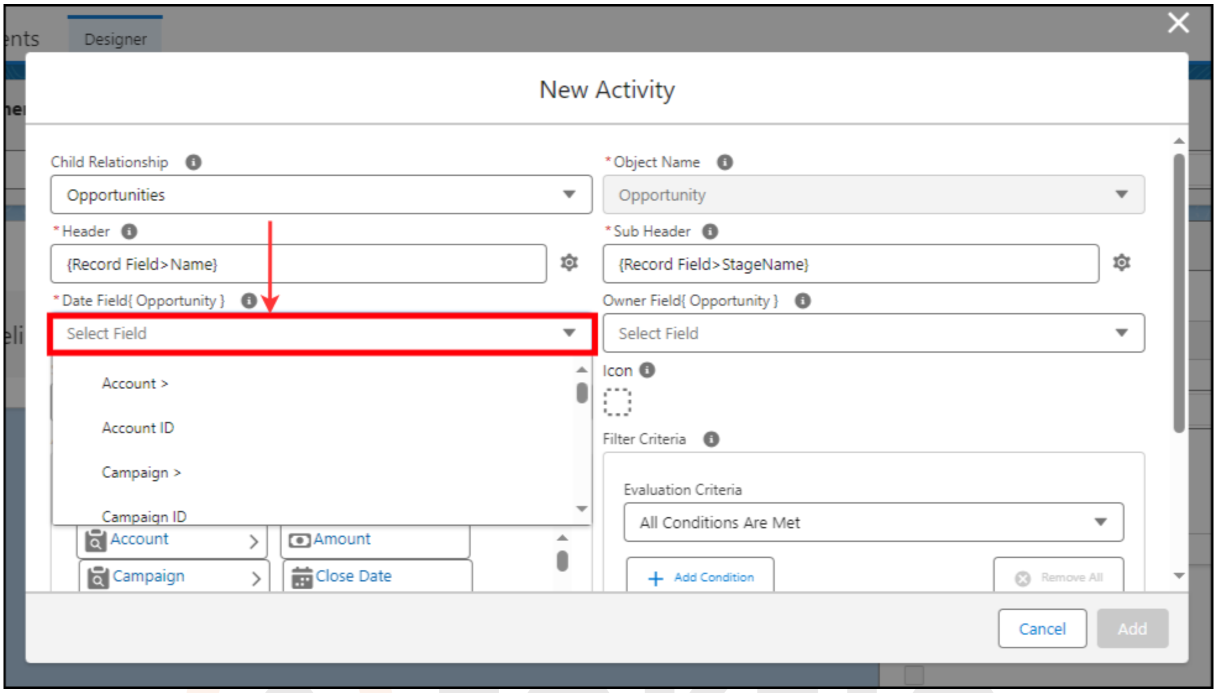


Figure 134: Date Field Config

* **Owner Field:** Users need to map one field from the source object that will be used to show who owns the activity, if we consider the same example of the case then we will map the case owner field to this one so we will be able to see that who was the owner of the case directly in the timeline

* **Status:** Users need to map one status field from the source object so that it will be displayed on the timeline.

* **Icon:** This is the field that will decide which icon needs to be shown for each activity in the timeline

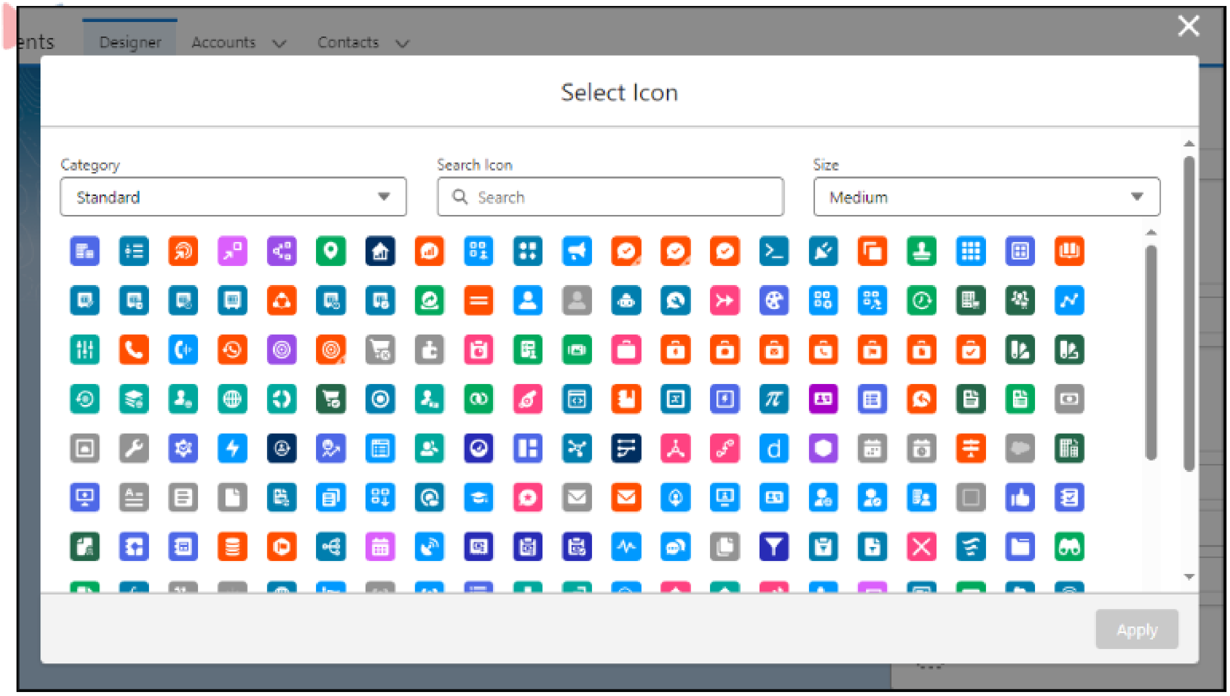


Figure 135: Icon Config

* **Additional Fields:** Users need to show some fields when user expand the activity to provide more details about the activity from the record, we will use our field selector which will allow us to select the field of the source object and its upward parent objects up to 3 levels.

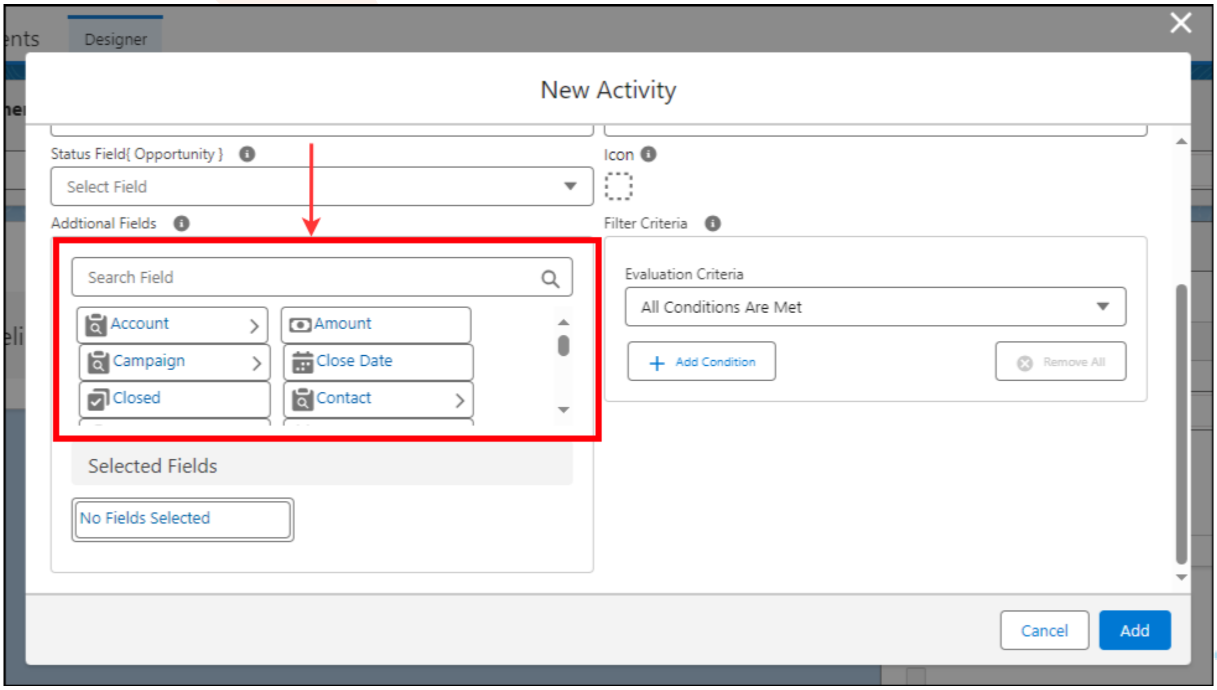


Figure 136: Additional Fields Config

* **Filter Criteria:** The ”Filter Criteria” section allows users to customize the display of the timeline based on specific criteria.

  · **Evaluation Criteria:** This will have three options, The first one is “All Conditions Are Met” which means it works as an AND operator. The second one is “Custom Logic Is Met” where we can give custom logic like (1 AND 2) OR 3 and the third one will be “Any Condition Is Met” which works as an OR operator

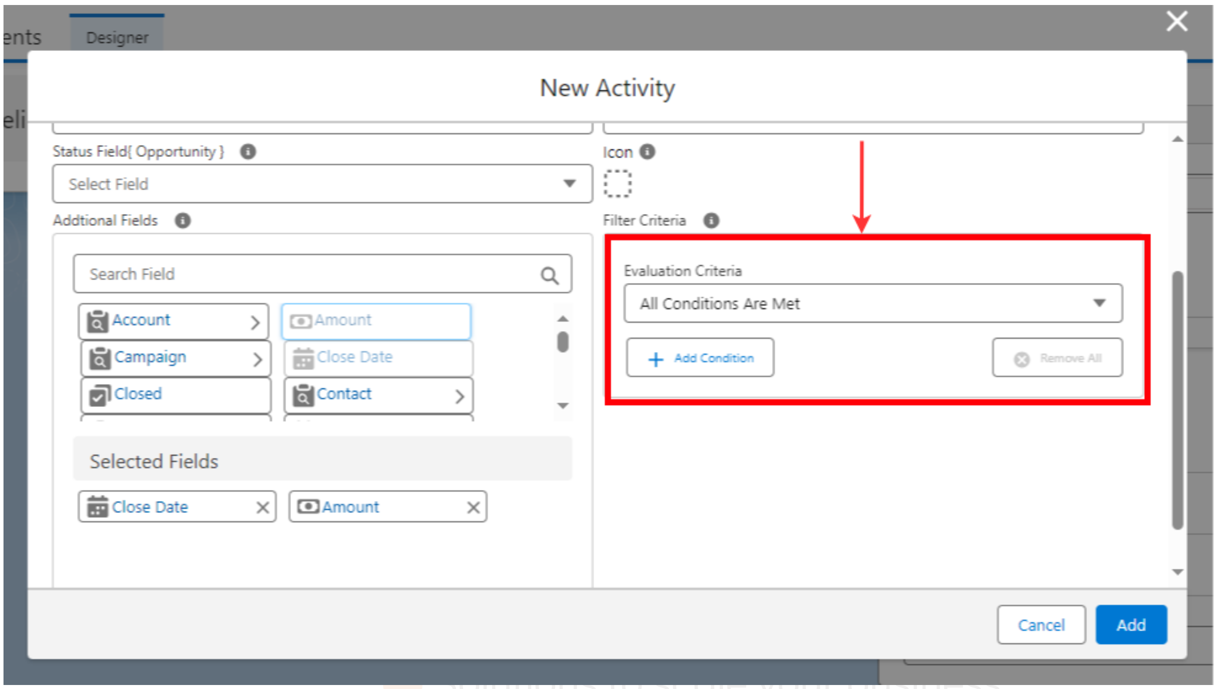


Figure 137: Evaluation Criteria Config

User will have two buttons one is Add Condition which is used to provide a condition, once clicked the UI will open to add a condition, another button is Remove All which will remove all the applied conditions

  · **Field** - When the user clicks on the field section, the user will see the list of fields from the source field and then the user can walk to the parent records up to 3 levels and can select any field to give condition.

  · **Operator** - In Operator user will have options like the following- Equals, Not Equals, Contains, Does Not Contains, Starts With, Ends With.

  · **Field Source** - In this user will have two options Static and Record Field, when the user selects static, one more text box will open to select static values and when the user selects Record Field, then the user will have a source fields and also user can walk through up to 3 levels from there and it will fetch the value from that field to validate.

* **Show Age:** This will be a checkbox type, and this will show the relative time as 2 days ago.

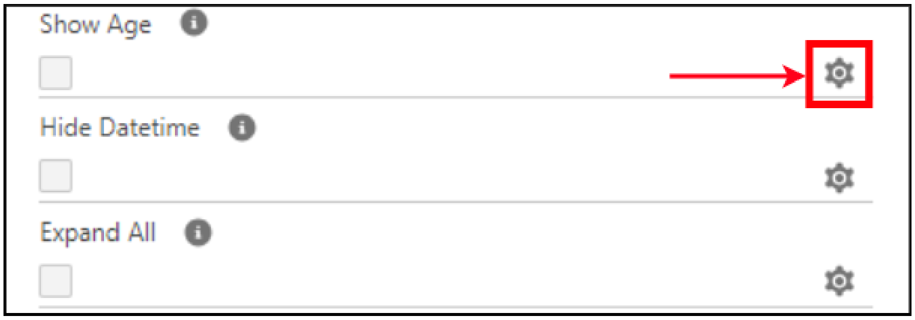


Figure 138: Show Age Config

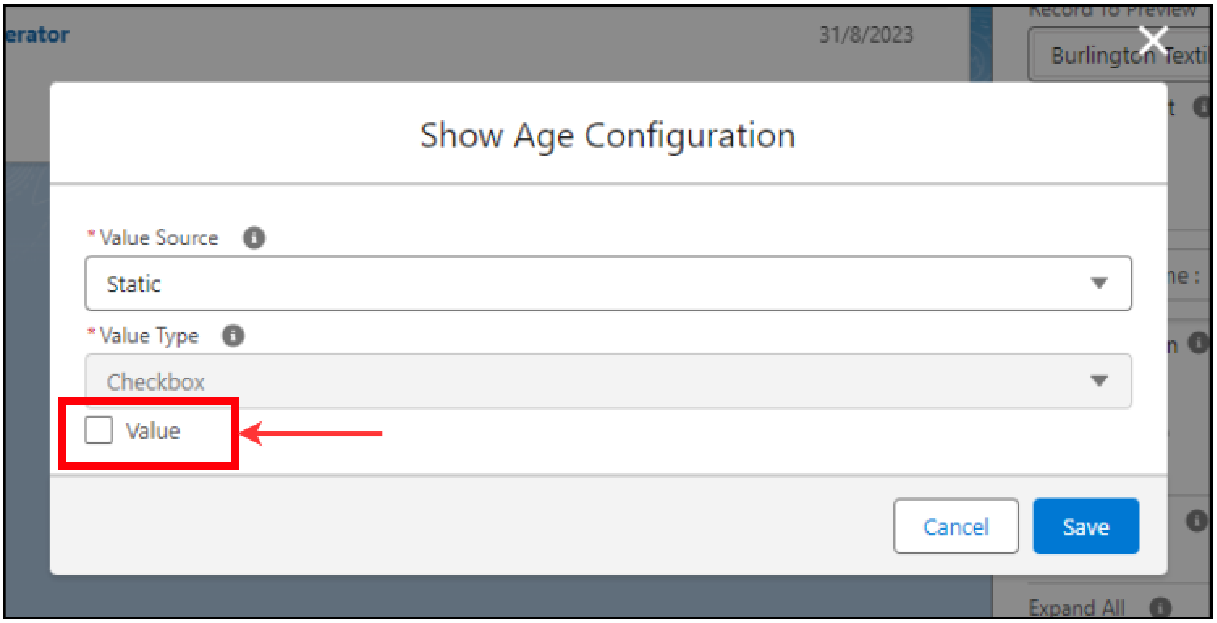


Figure 139: Show Age Value

- **Hide Datetime:** In Timeline user uses one field which will be mapped to the date field, this field derives the sorting of activities in the timeline, this hide option will be of checkbox type that will confirm if the user wants to show that date & Time on a timeline or not by default this will be Unchecked.

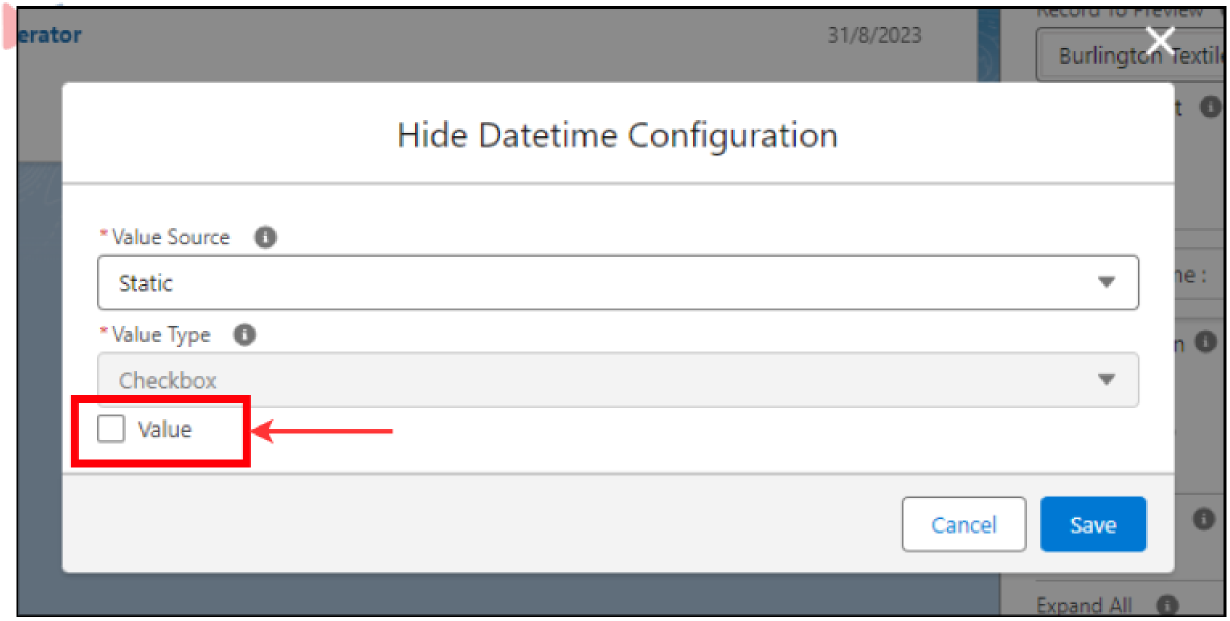


Figure 140: Hide date time config

* **Expand All:** This option will allow users to expand all the activities and collapse all activities, after refreshing the page this will go away to the original config.

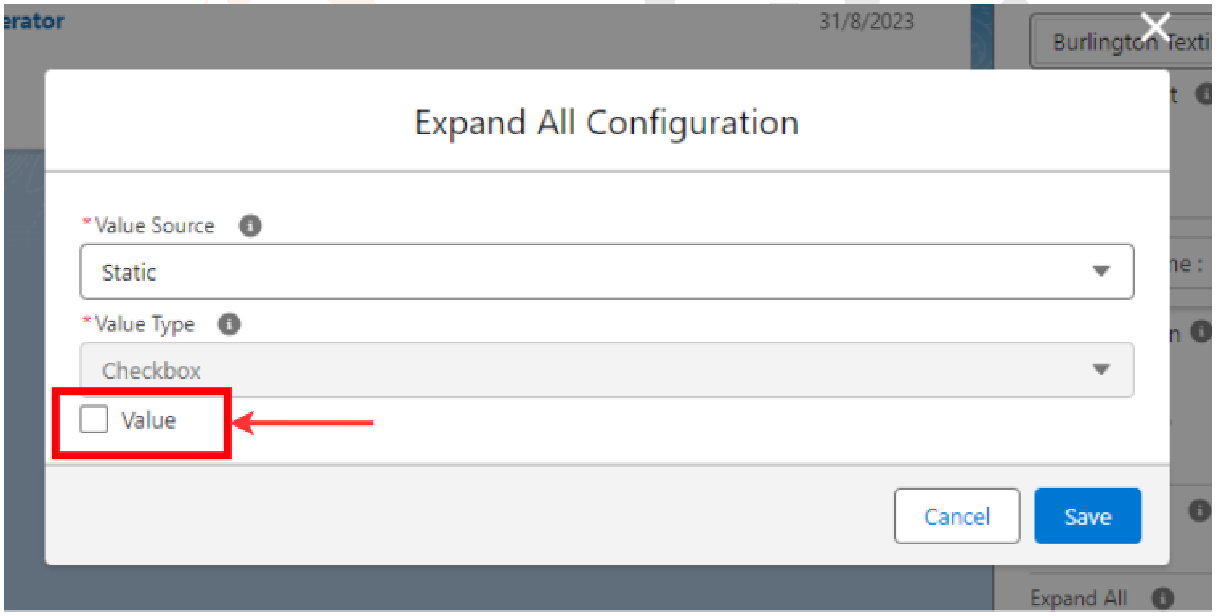


Figure 141: Expand All config

* **Sort As:** User can sort the records using the date field only because the timeline is based on the date field, so the user can sort from OldestDateFirst and NewestDateFirst, as per the name when a user says oldest first then it will be sort in ascending order and newest first means it will be ordered in descending order if user don't give anything then by default it will be selected as Newest-

DateFirst.

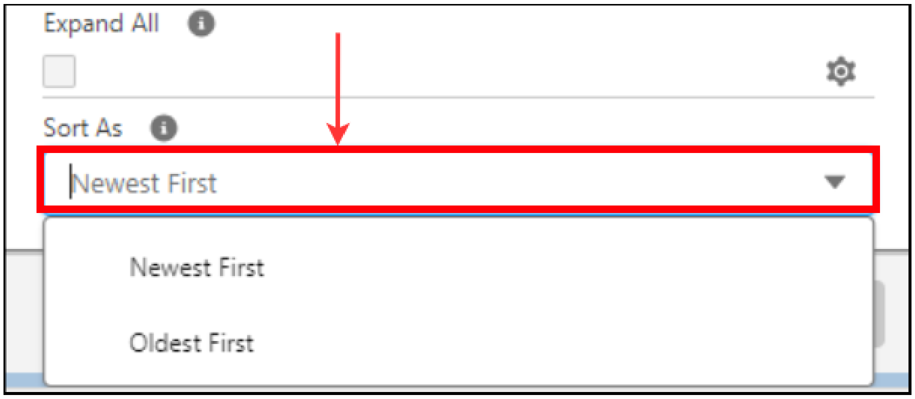


Figure 142: Sort As config

* **Refresh:** This will be an option to refresh the component and whatever changes we applied to our component on runtime will be reversed back as per config like filters and sorting.
* **Show Search:** There are many chances that the user doesn't apply filters for objects, so the timeline might have too many records combining all objects to show, so the user needs to have a setting to fetch the records and show it on a timeline, so when user clicks on show more button than next records including current records on a timeline will show.

After clicking on the Save button the config will get saved on the component config object and the user can view this button on the Record page, App page, Home Page, Flows, and Community.

– **Preview Section:**

After the user has selected and filled the fields according to their requirements, they can preview the components in the Preview section before saving them on the Record Page, App Page, Flows, and Community Page.

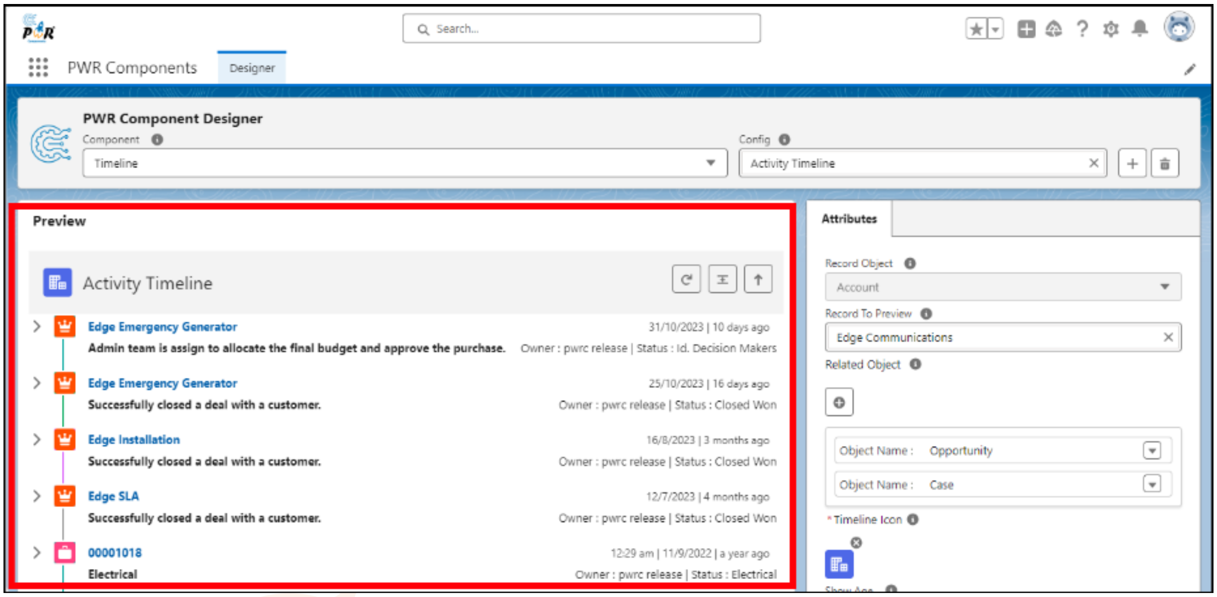


Figure 143: Preview Section

## – PWR Timeline screenshots:

⋆ PWR Timeline on Record Page:

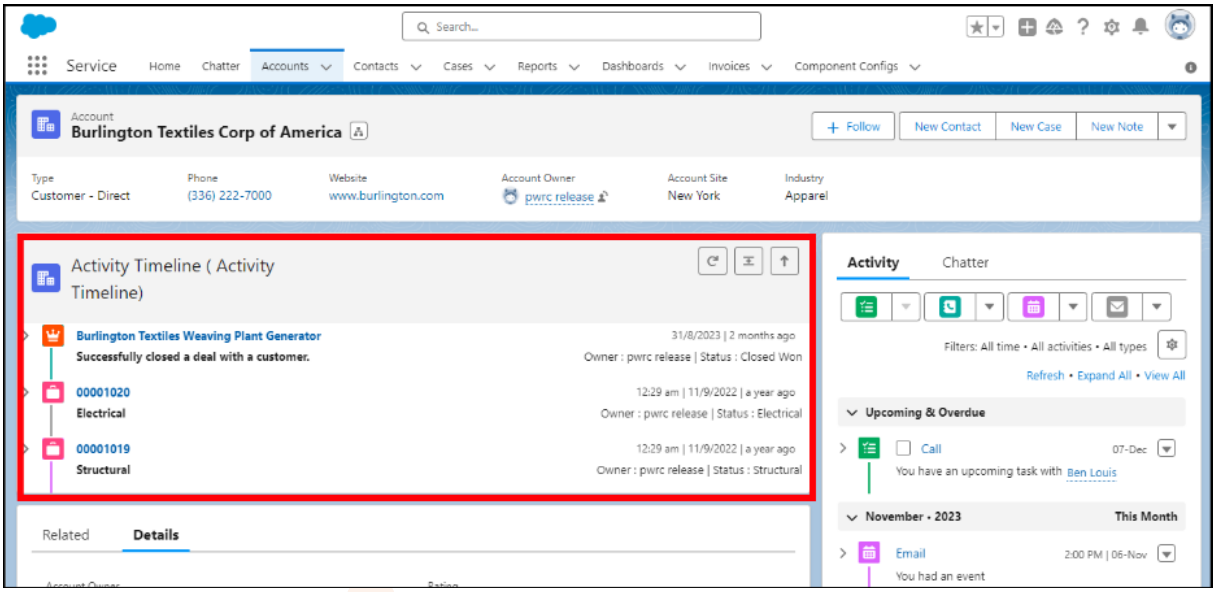


Figure 144: PWR Timeline on Record Page

⋆ PWR Timeline on Home Page:

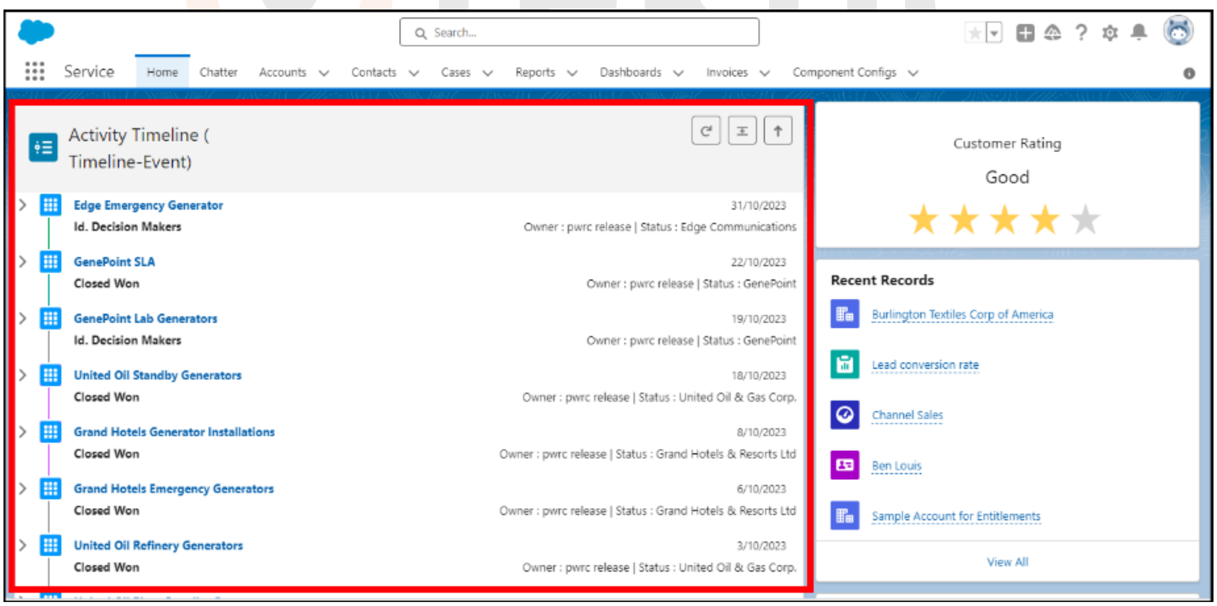


Figure 145: PWR Timeline on Home Page

## V.IV Steps To Configure the PWR Components on the Record Page, Home Page, and Flows:

– Select where to Configure the Required Component: User need to select where they want to configure or show the required/ any Component, i.e. if they want it on the Home Page or Record Page of an Object. If they want to Configure/ show the Component on the Home Page or App Page or Record Page, then they must select it from the Lightning App Builder

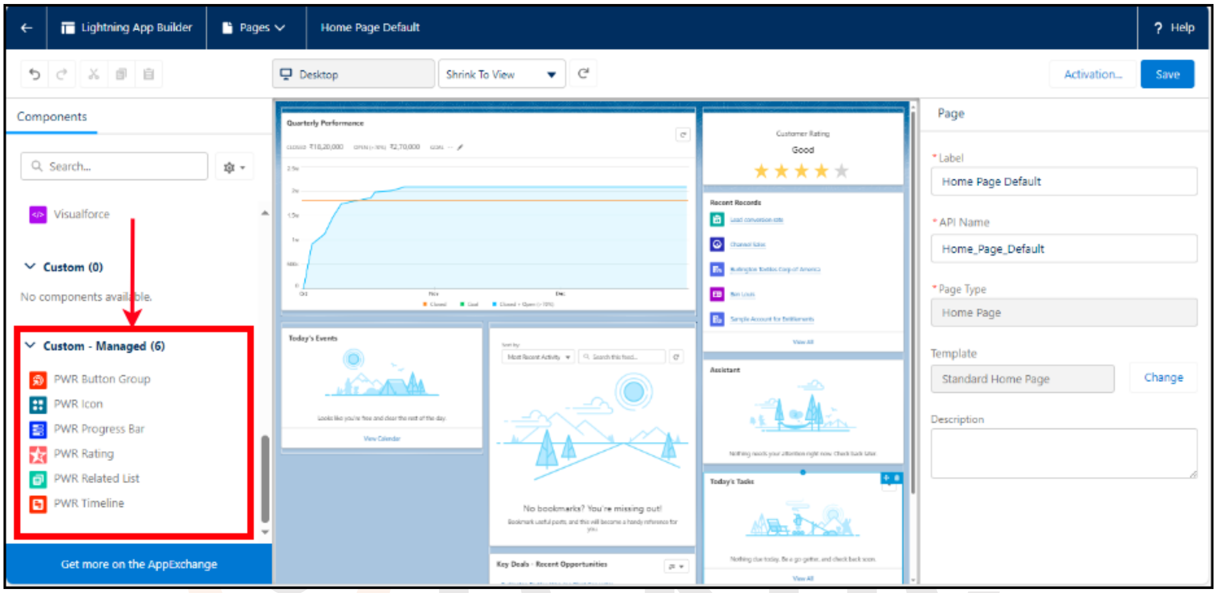


Figure 146: Edit Lightning Page



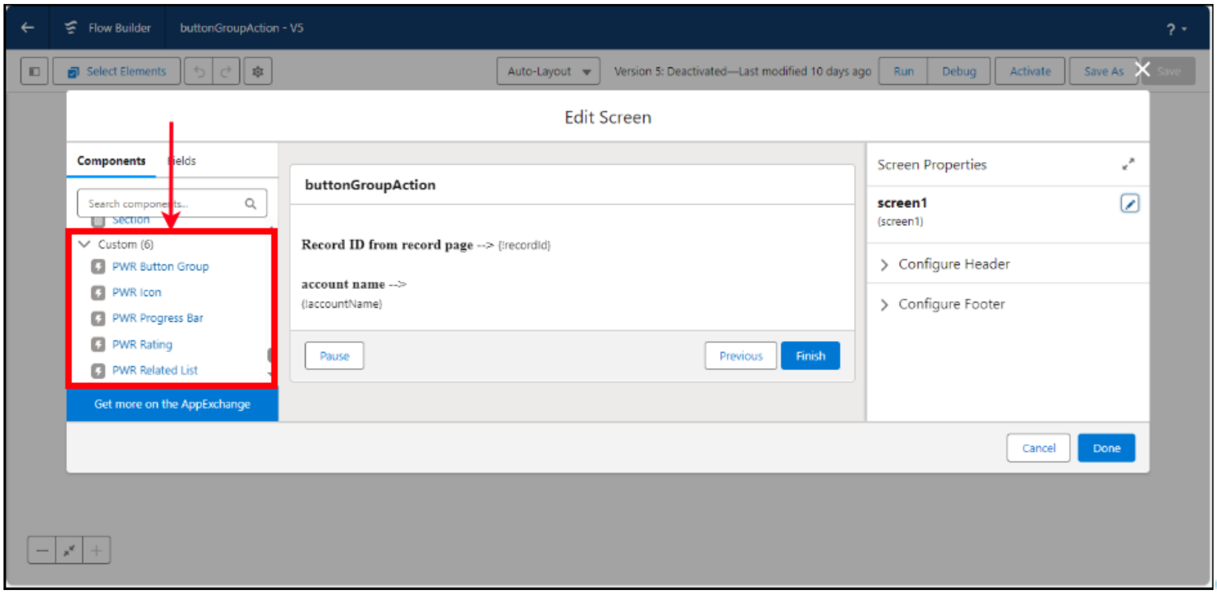


Figure 147: Edit Flow Page

- **Drag and Drop the Component:** Now users will have to Drag and Drop the required Component from the Custom Components section of the list of Components on the section according to their requirements.

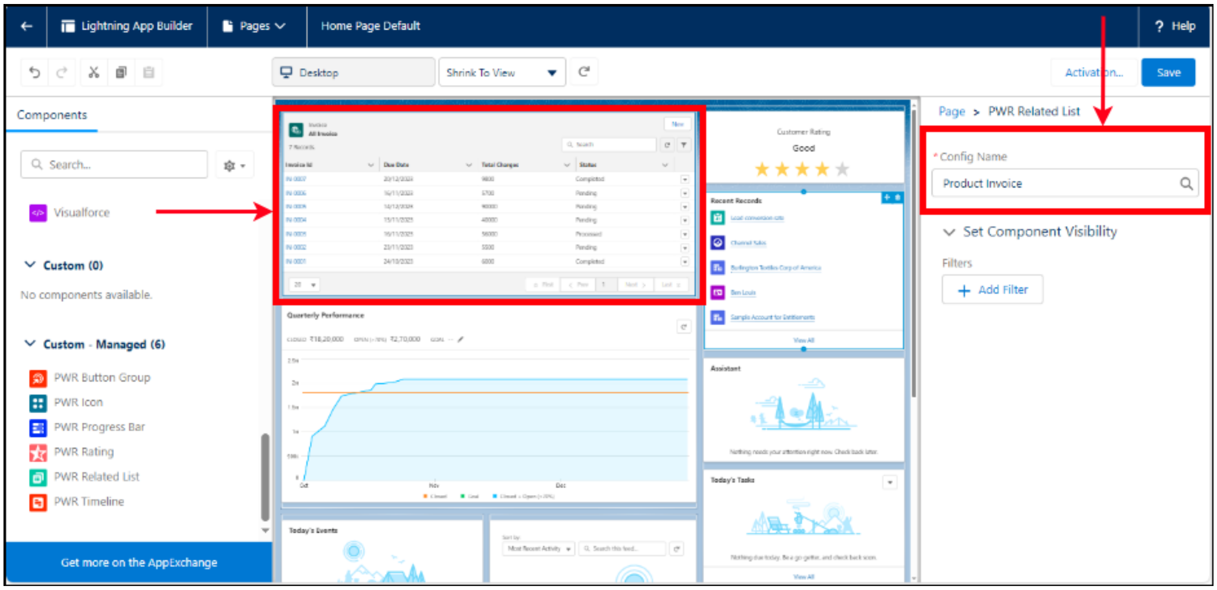


Figure 148: Drag and drop component

- **Select Config:** Once they have Dragged and Dropped the Component on the Page, they have to Select a Config name for that Component, if the User has selected the Home-Page or App-Page they can only see or select the Non-Record Page Config, but if they have selected Record-Page of an Object, then they can see and select Configs of all record page configs as well as non-record page configs.

- **Save the Page:** Now User has to save the Page so that the changes and the components they have added are saved and can be viewed on the specific Pages.

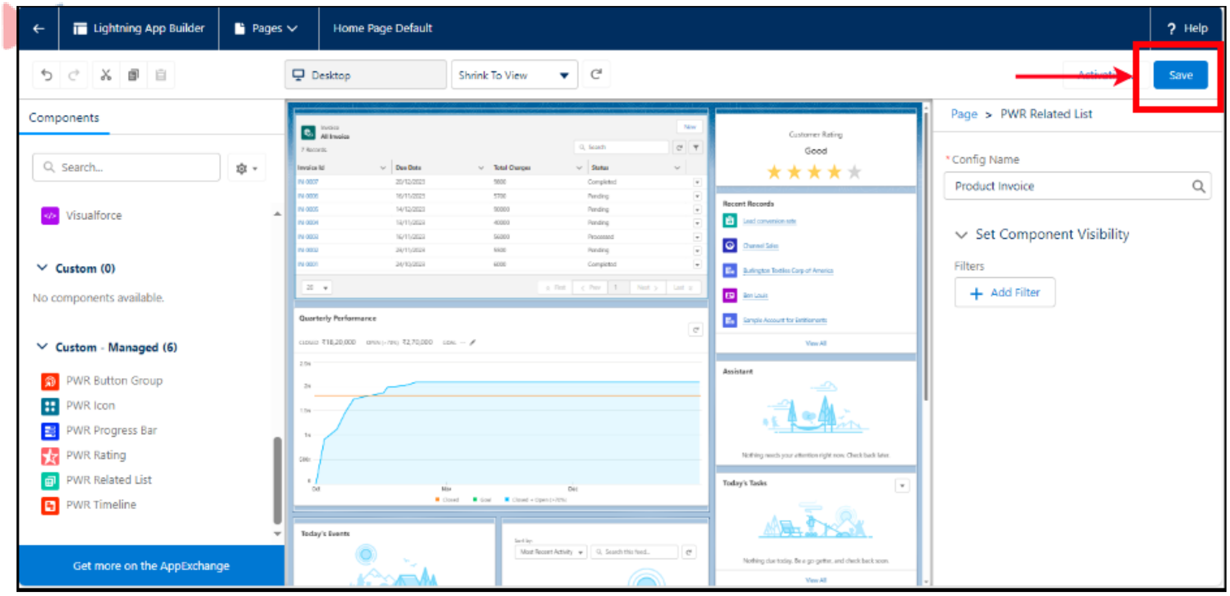

Figure 149: Save

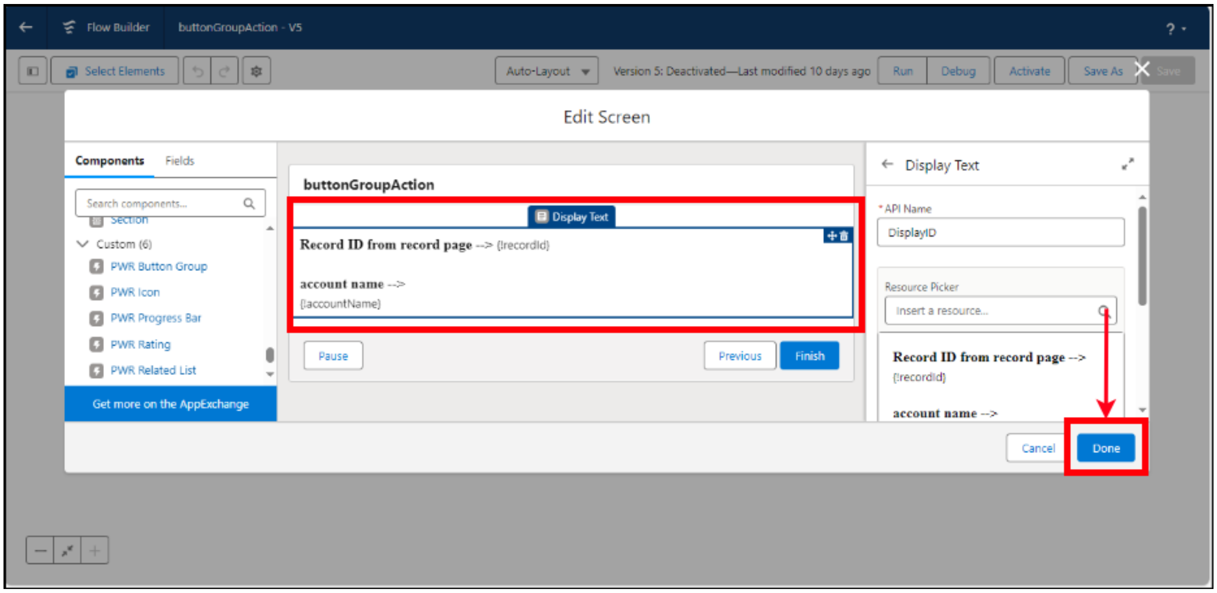

Figure 150: Save

# VI Contact Us

For more information visit www.orektic.com

You can also schedule a product demo to know more about **PWR Forms** simply by filling out the form here.

If you have any concerns or queries then please contact us at support@orektic.com