Orektic Solutions is known for state-of-the-art cloud based, SAAS solutions which help our clients to transform the way they do 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

With our innovative thoughts, we have come up with new AppExchange Package which will help the Salesforce admin, and developers which provides drag and drop-able Lightning Web Components that can be used on the Home page, or App page, or Record details page to ease the use of record according to users custom requirement.

With most of the Salesforce users are moving to Lightning from Classic, and having limited standard features and components from Salesforce, admins, and developers find it difficult to customize, as per the requirements and need to create customized UI (Visualforce, AURA Components, etc.)

- Limitation to not able to show key information from the parent record.
- Page Layouts Limitations
- ListView Limitation
- Field Set Limitation

How to get started??

To get started, you need to make sure you are all good with the list of prerequisites/checklist mentioned below.

Prerequisite

As a prerequisite, Salesforce admin needs to make sure that the Salesforce org is already setup 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.

Checklist

Below is the checklist to complete, before installing the package to their Salesforce Org.

- [ ] Salesforce Org Setup with Domain Name
- [ ] The user installing the package must have admin access in Salesforce Org

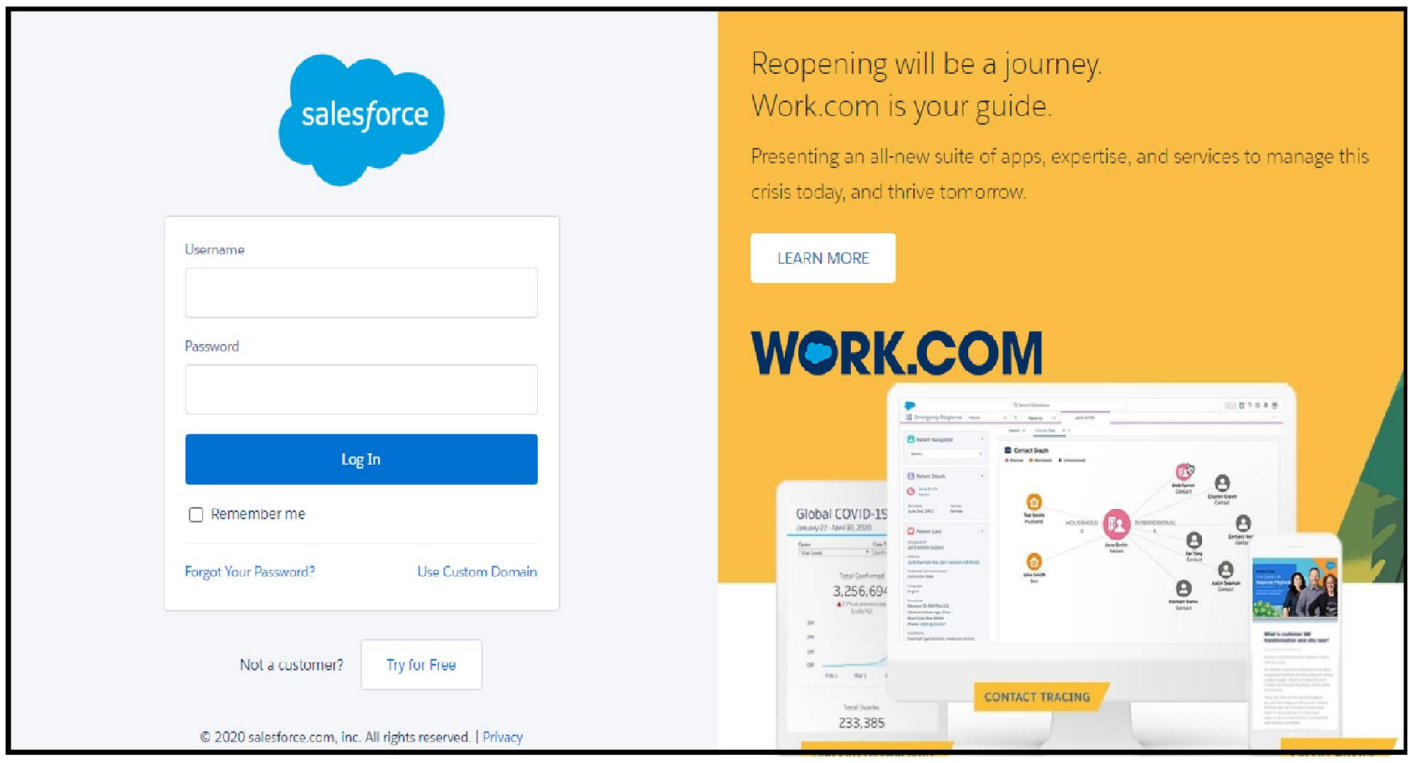
Once you confirm you are all set with the prerequisites and checklist, you need to install a managed package from AppExchange, and to do so please refer to the below article : Package Installation – AppExchange

In case if you have received a managed package link from our team, then to install the package please refer to the below article : Package Installation - URL

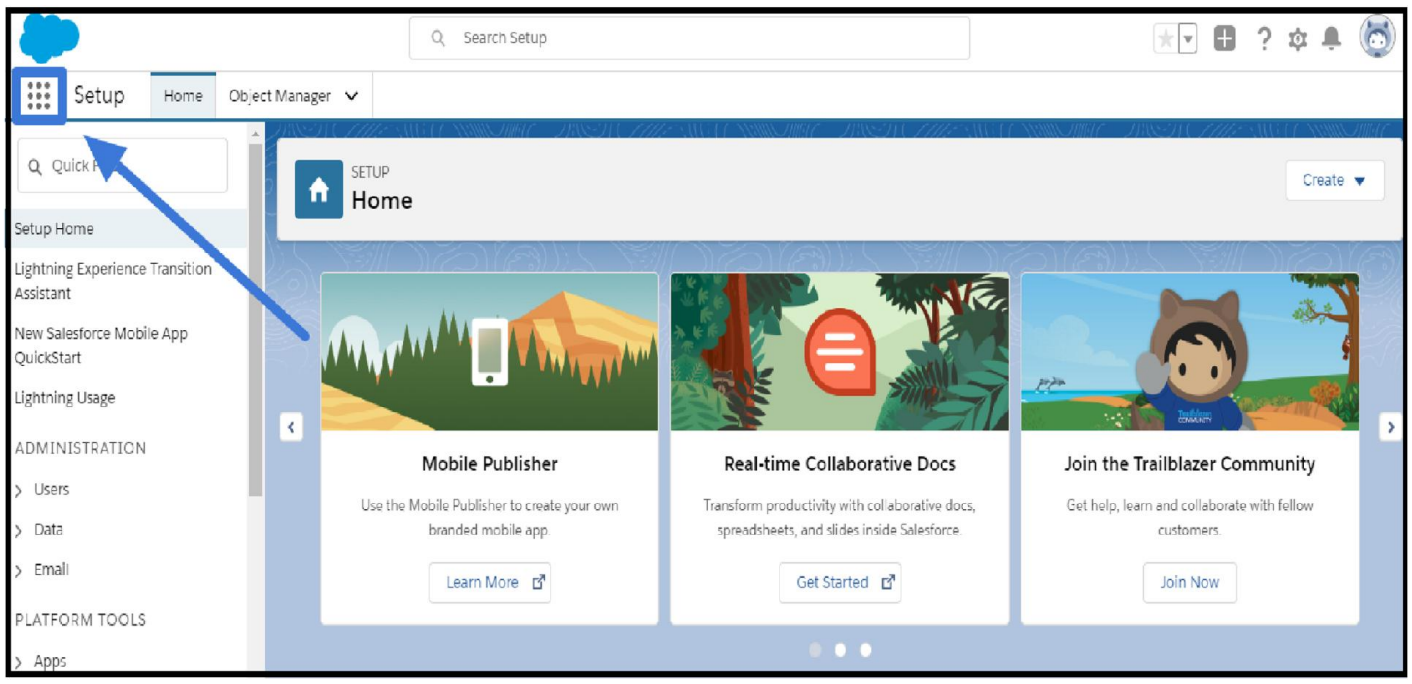
Package Installation - AppExchange

Please go through the steps below to successfully install the package to your Salesforce Org (Production/Sandbox).

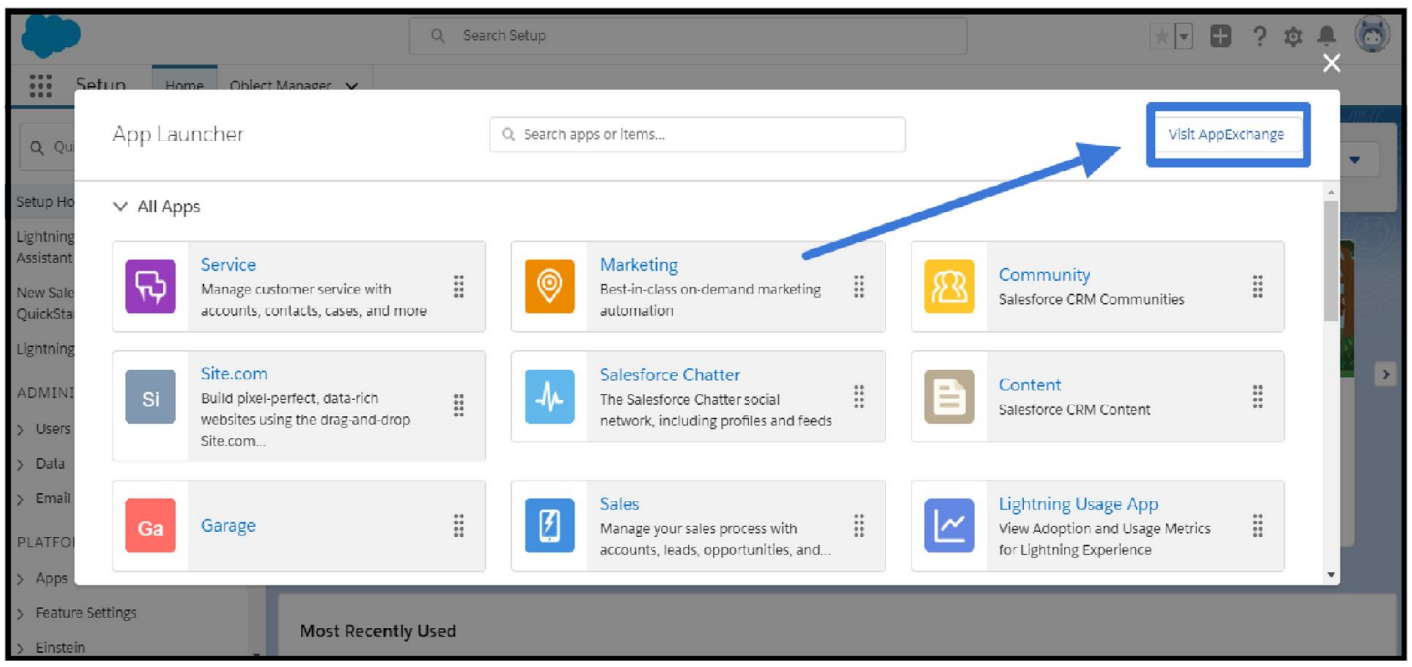
Login to Salesforce



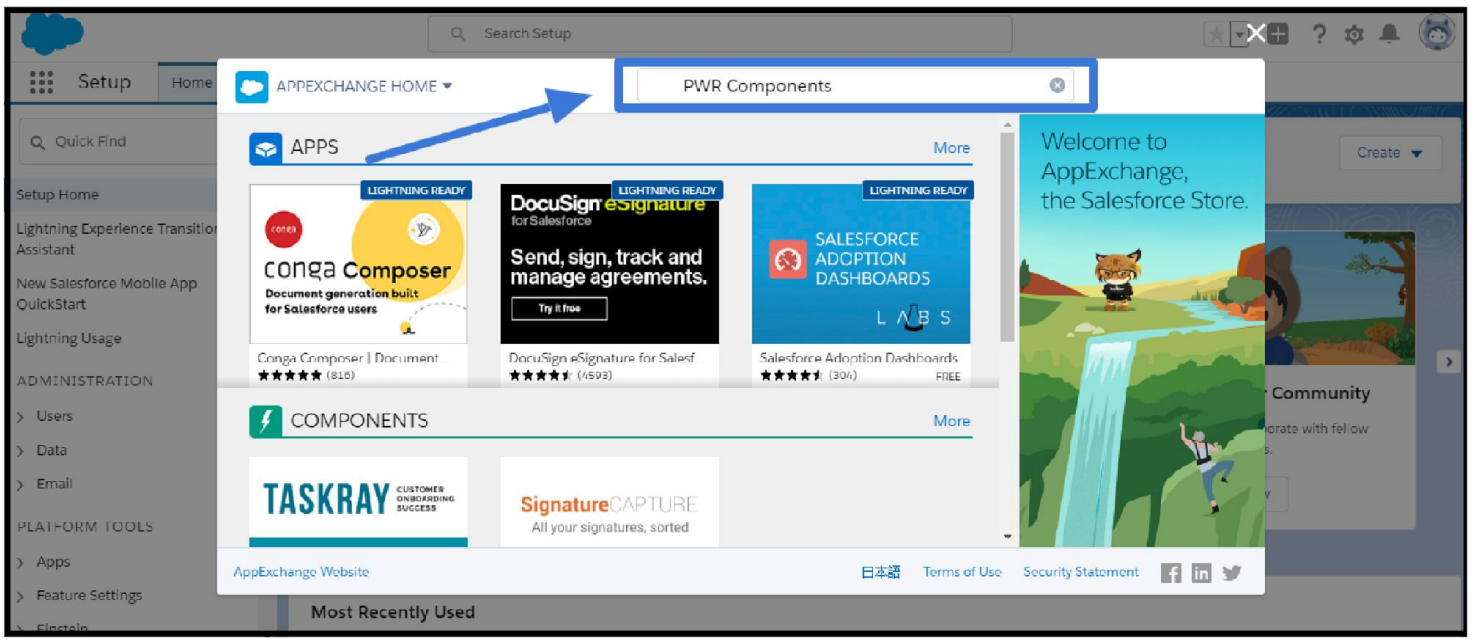
Click on Apps Icon



Click on AppExchange Button

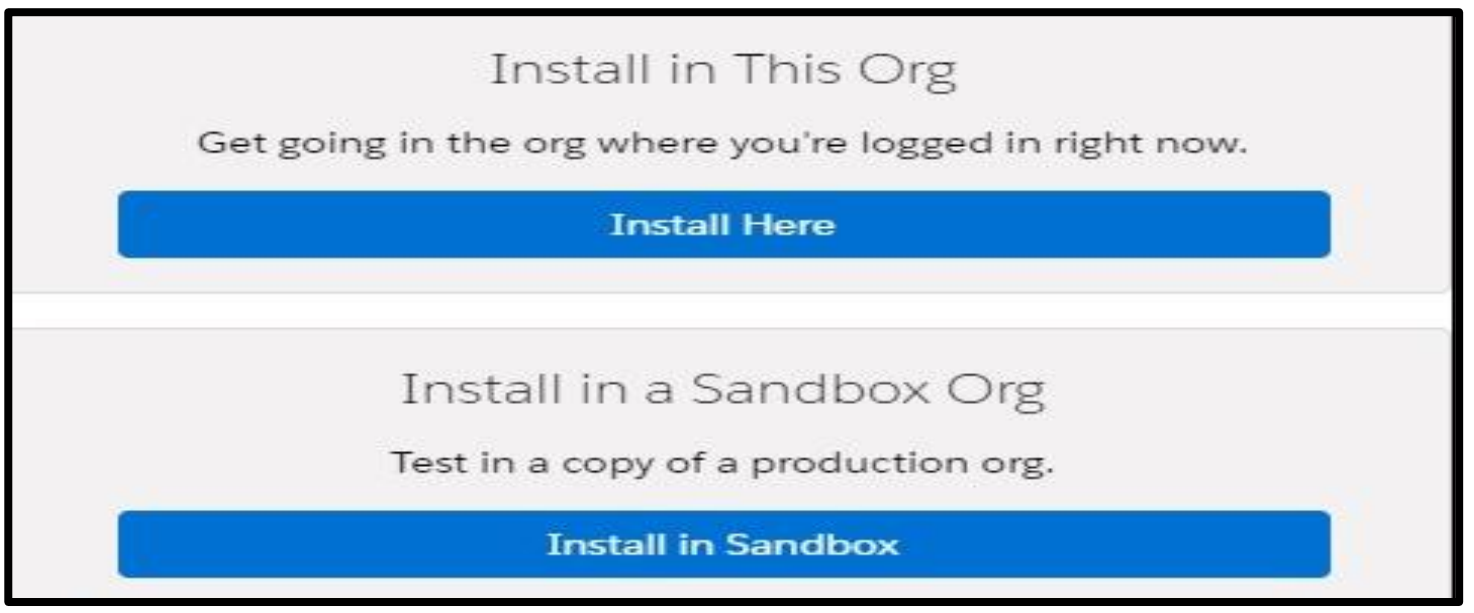


Search - PWR Component



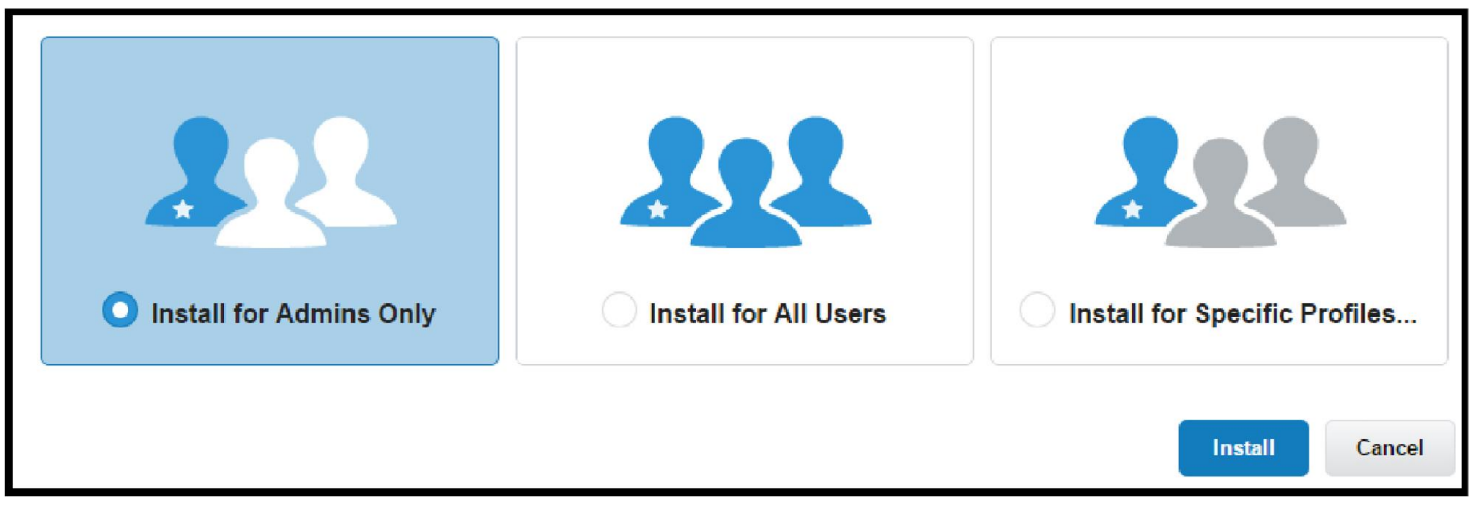
Click “Get It Now” Button, and then click “Open Login Screen”.

Select Org Type (Sandbox / Production)



Click on “Agree to Terms and Conditions”, and click “Confirm and Install”

Select Users, and Click Install



Click on Done once the installation is completed. In some cases, package installation takes more time than expected, in that case, the user who is installing the package will receive an email notification

You can find the PWR Components package in the Installed Package under Setup of your Salesforce instance.

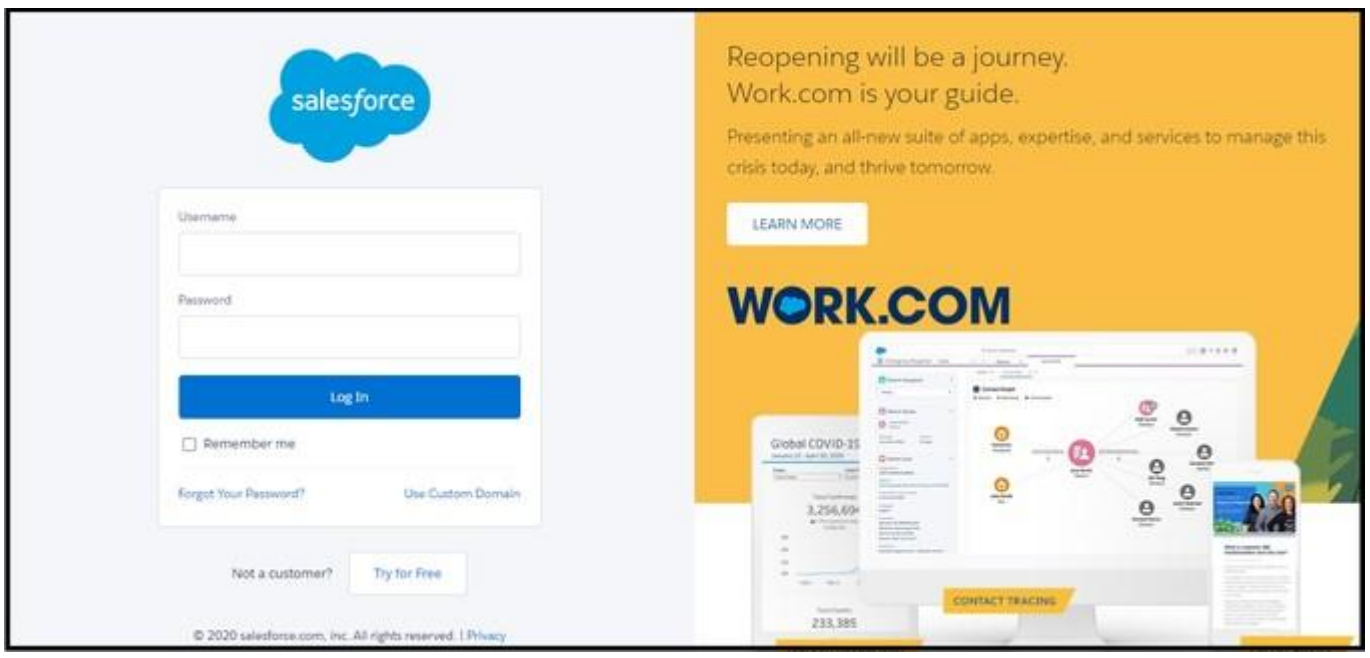
For more reference and details around package installation, please click here

In case if you encounter any issues or difficulties, please reach out to us at support@orketic.com

Package Installation - URL

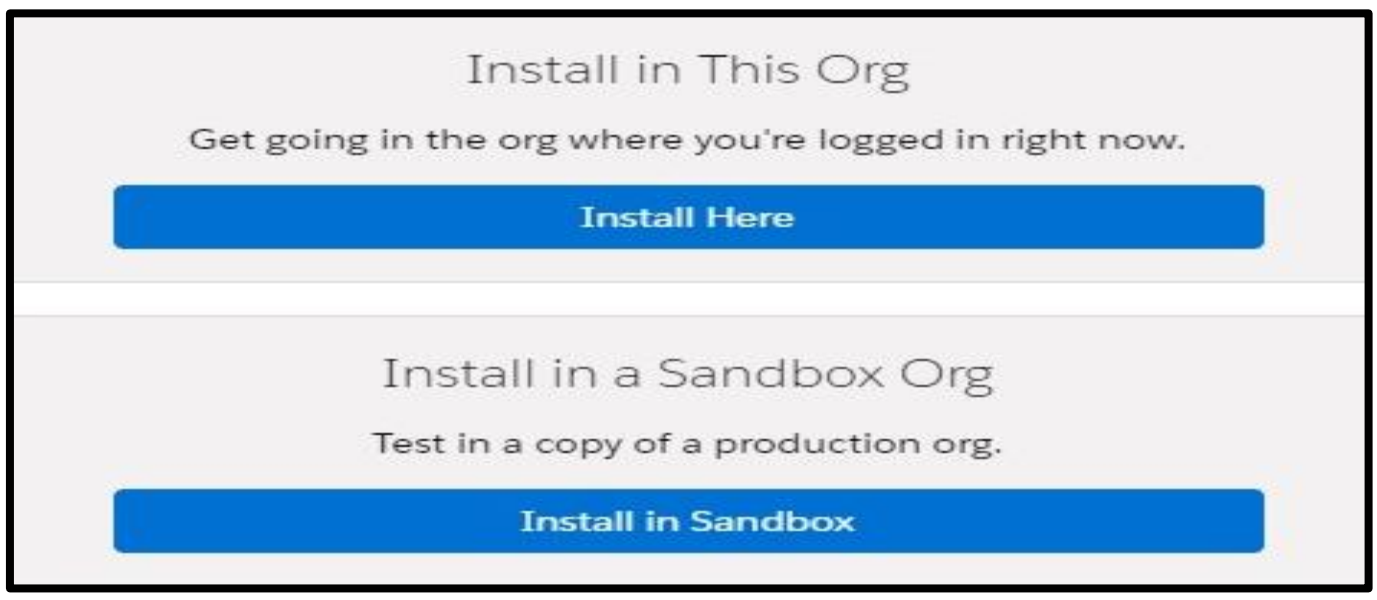
To install the PWR Components managed package, please do follow the steps given below: Click the package installation URL.

Log in to Salesforce



Click “Get It Now” Button

Select Org Type (Sandbox / Production)



Click on Agree Terms and Conditions and click Confirm and Install.

Select Users, and Click Install

Click on Done once the installation is completed. In some cases, package installation takes more time than expected, in that case, the user who is installing the package will receive an email notification

You can find the PWR Components package in the Installed Package under Setup of your Salesforce instance.

For more reference and details around package installation, please click here

In case if you encounter any issues or difficulties, please reach out to us at support@orketic.com

Components

PWR Components is a bundle of following lightning components, which helps the Salesforce admin/developers achieve their requirements and goals with easy drag and drop-able lightning components Below components help in overcoming the limitation of Salesforce configuration.

1. PWR Progress Bar
2. PWR Field
3. PWR Field Set
4. PWR Button
5. PWR ListView
6. PWR Related List

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.

PWR Progress Bar is of two types:

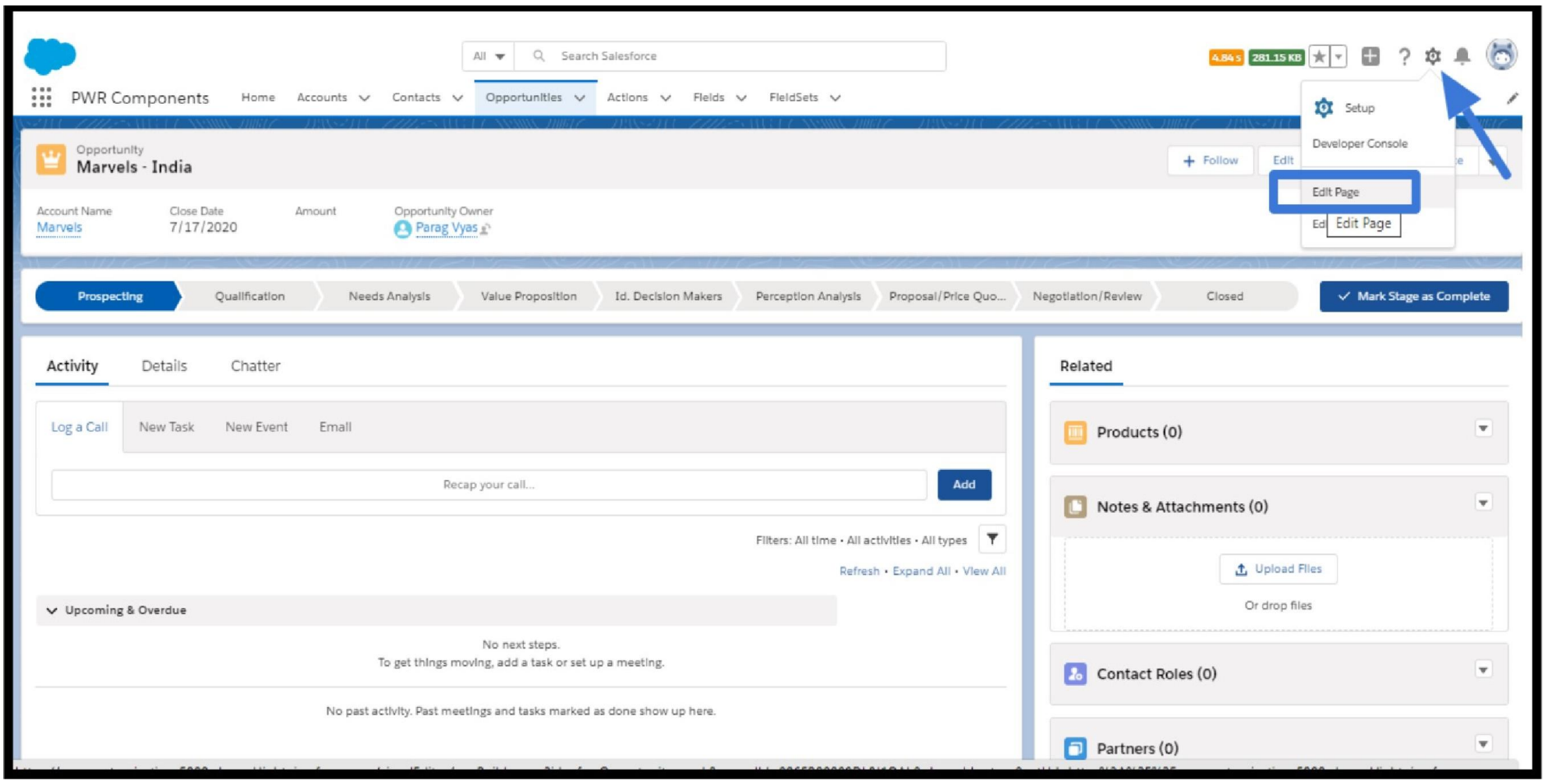
- **Date Progress Bar:** Date Progress Bar is used to measure the progress of the work between the range of the dates given.
- **Number Progress Bar:** Number Progress Bar is used to measure the progress of the work between the range of number fields.

*** PWR Progress Bar is mobile compatible. Number Progress Bar by default takes the start and end value as 0 to 100 respectively, but users can customize the start and end value as per their requirements*

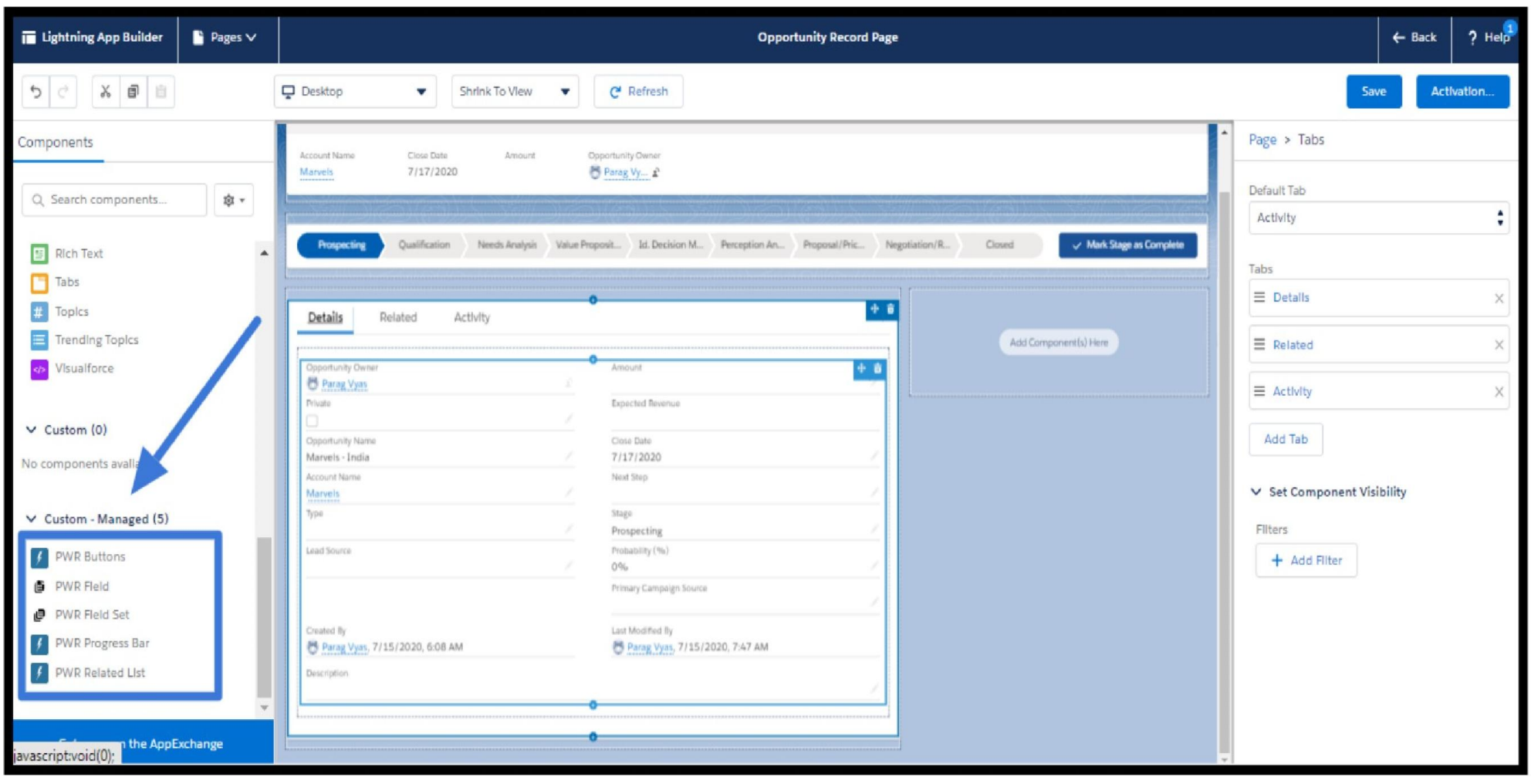
CONFIGURATION DETAILS

Here are the steps required to configure PWR Progress Bar on your Record Page, similarly you can configure the same on the Home page or App page in Salesforce Lightning.

To edit a record page, Click on Gear icon (setup) and click Edit Page



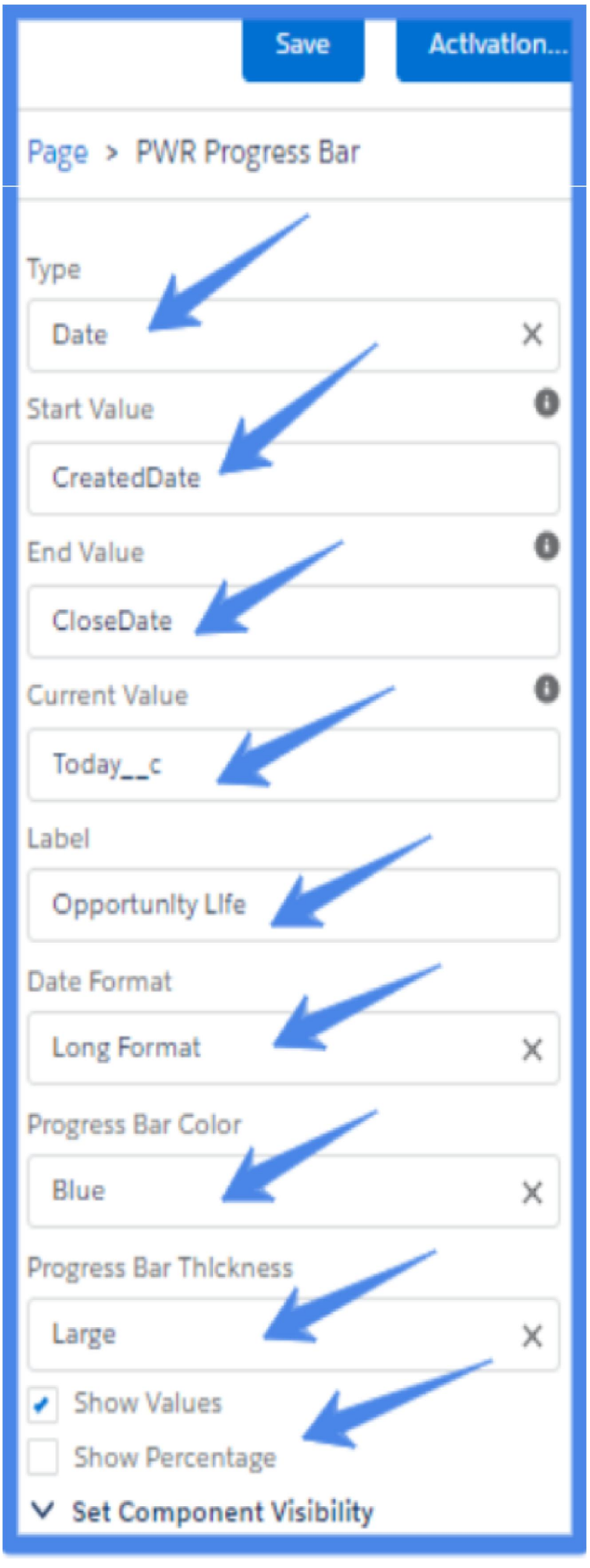
In Lightning App Builder, scroll down in Component panel (left panel), and at the bottom, you will find the list of all Custom Managed Components.



Drag and Drop Progress Bar in the desired section of the record page

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

As per the requirements, the user can select the type of progress bar, it can be Date or Number.



1. **Type:** Select the Type of the Progress Bar
   a. Date
   b. Number.

2. **Start Value:** Based on the type selected, users need to input the API name of the field

3. **End Value:** Based on the type selected, users need to input the API name of the field

4. **Current Value:** Based on the type selected, users need to input the API name of the field.

   Note**

   If the Type is Date, then the user should update the API name of either **Date** or a **Date /Time** field in Start Value, End Value, and Current Value.

   If the Type is Number, then the user should update the API name of either **Number** or a **Text** field in Start Value, End Value, and Current Value.

   In case, if no field is updated, then for Date Progress Bar default date is today's date, whereas for Number Progress bar Current Value is a mandatory field.

5. **Label:** Users can name the progress bar by entering the name in the Label.

6. **Date Format:** In case, of Date Progress Bar User can choose the Date Format to appear on the Progress Bar by selecting either.
   a. **Default Format** (2020-01-20)
   b. **Long Format** (20-Jan-2020)

7. **Progress Bar Color:** PWR Progress Bar can be visualized in two colors.
   a. Green
   b. Blue

8. Progress Bar Thickness: User can set their preference for the thickness of the Progress Bar a. Large
   b. Medium
   c. Small
   d. Extra Small

9. **Show Value:** In order to show or hide the Start Value and End Value on the PWR Progress bar, the Show Values checkbox can be checked and unchecked respectively.

10. **Show Percentage:** In order to show or hide the progress in percentage on the PWR Progress bar the Show Percentage checkbox can be checked and unchecked. respectively.

PWR Field

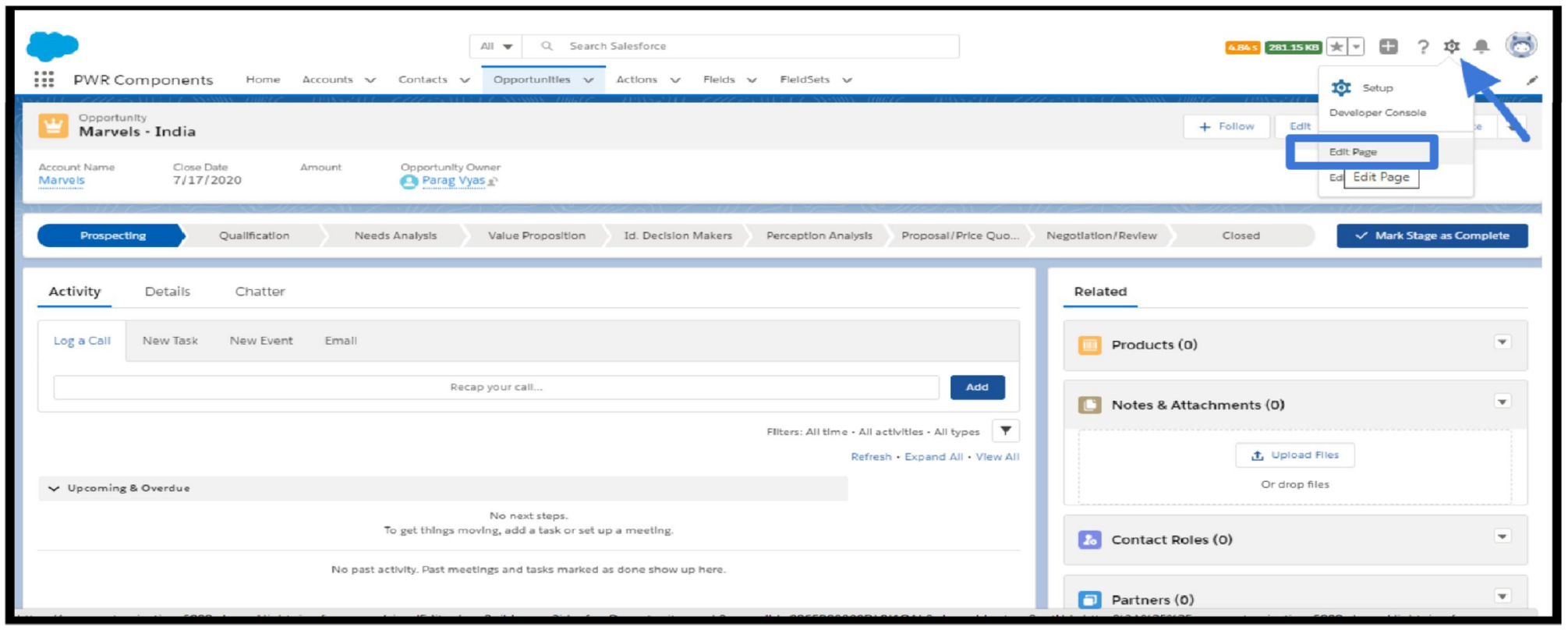
Users can use the PWR Field component to highlight and show any Field on the UI and the User does not have to do a lot of customization or need a lot of object relation knowledge. With PWR Field, user can display the information from related objects and go up to 3 levels up in the Object hierarchy

With the help of PWR Field components, users can show the field in either editable or read-only format. Also, it enables users to put restrictions on the number of picklist values to be shown.

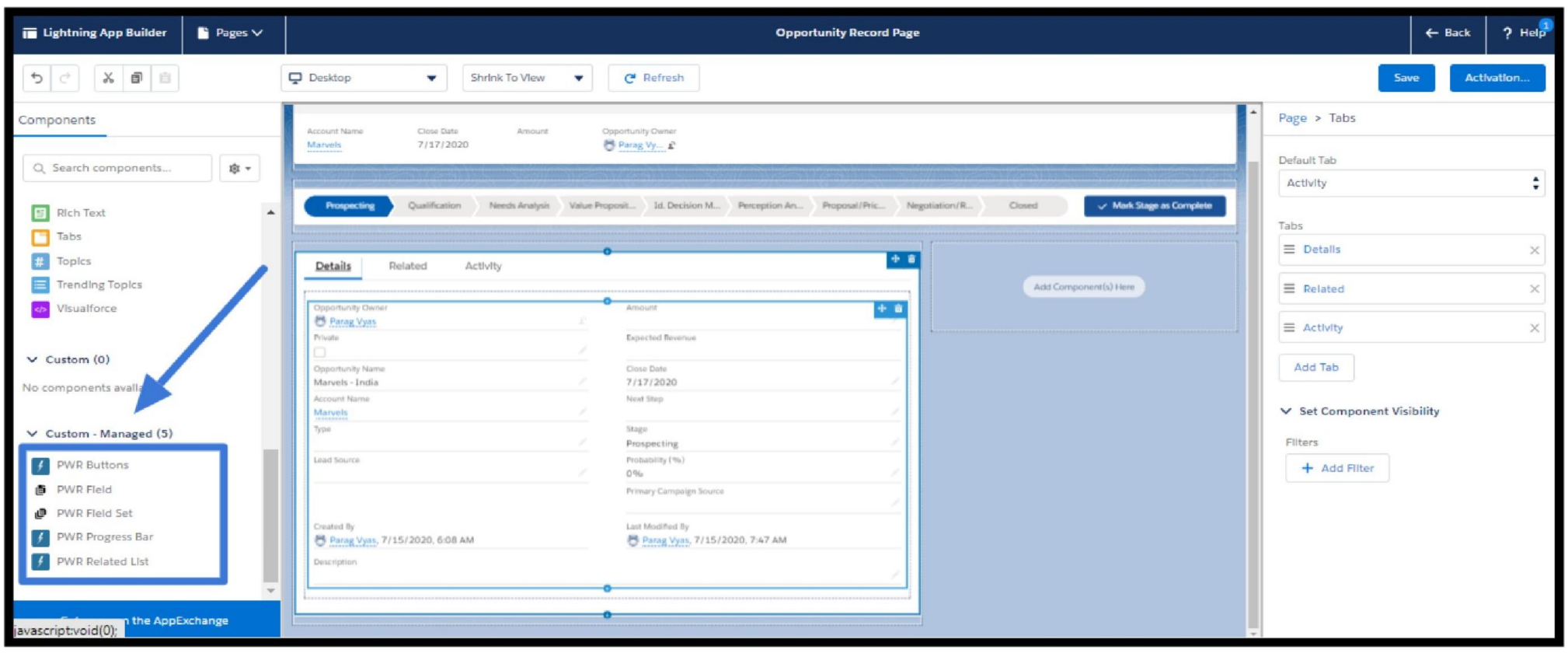
CONFIGURATION DETAILS

Here are the steps required to configure PWR Field on your Record Page, similarly you can configure the same on the Home page or App page in Salesforce Lightning.

To edit a record page, Click on Gear icon (setup) and click Edit Page



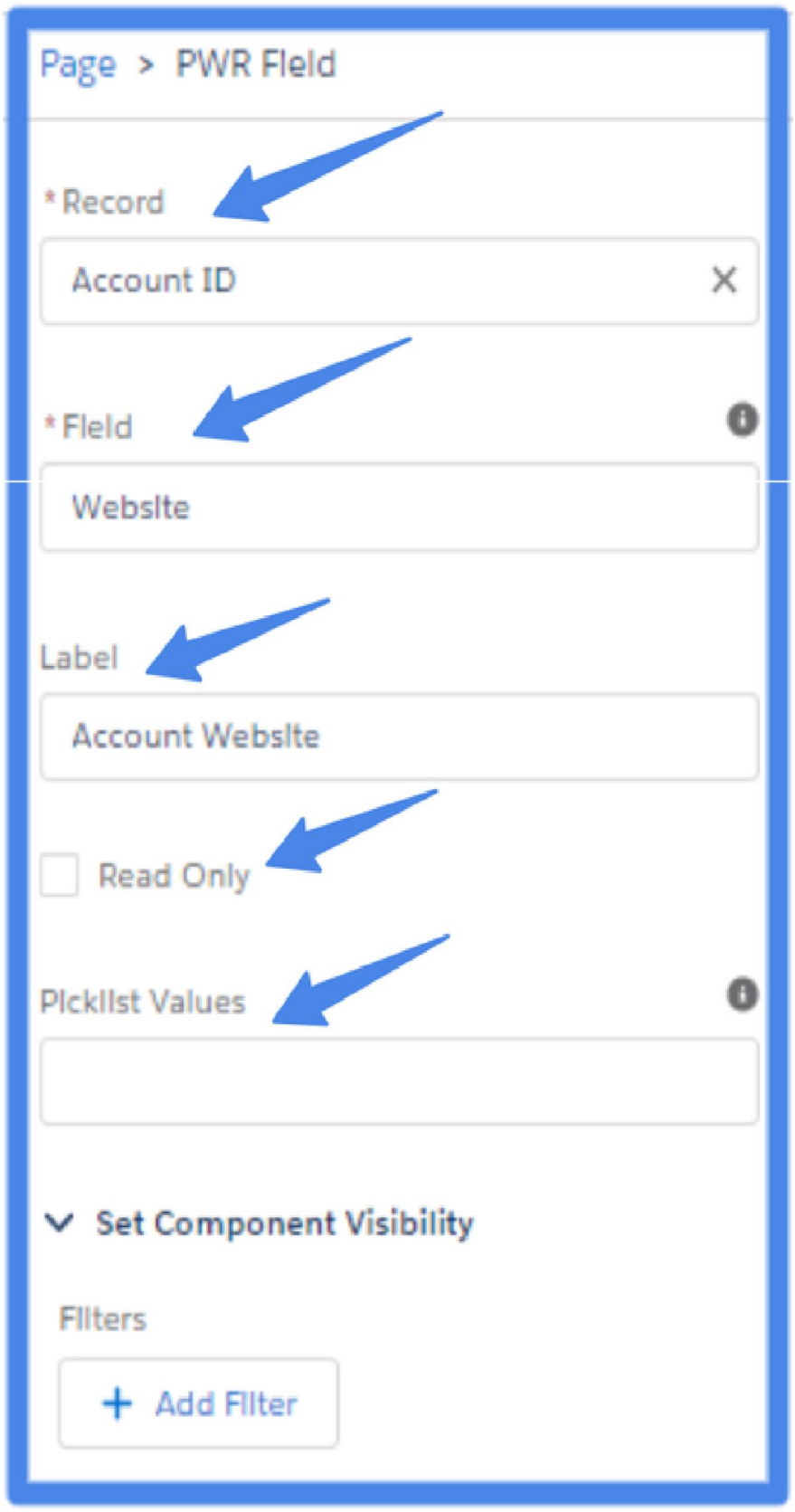
In Lightning App Builder, scroll down in Component panel (left panel), and at the bottom, you will find the list of all Custom Managed Components.



Drag and Drop PWR Field in the desired section of the record page

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

As per the requirements, the user can select the object from which they want to see the field



1. **Record:** Users can select the following records from the options which are the fields of either the current record or the parent objects record or the grand-parent record.
    a. **Current Record:** If the user is using the PWR Field component on the Contact object, then the user can show the fields from the Contact object. For example, Name, Phone, or Email.
    b. **Parent Record:** If the user is using the PWR Field component, then the user can also fetch fields from the Parent object. In our example, users can fetch fields from the Account object as well. User can show Account website, or Account HQ Address and other fields.
    c. **Grand Parent:** If the user is using the PWR Field component, then the user can also fetch fields from the Grand Parent object. In our example, if the user wants to show Account Owner details on Contact, same can be done using PWR Field component
2. **Field:** Users need to enter the API name of the field that needs to be displayed on the UI from the record user has selected (Record, Parent Record, or Grand Parent Record)
3. **Label:** Users need to enter the label which will be displayed on the Lightning Record Page (UI) to the end-users.
4. **Read Only:** In case the user wants to make the field Read-Only on Lightning Page, then the User can check the Read-Only checkbox.
5. **Picklist Values:** If the user has selected a picklist or multi-picklist field, and the user wants to show some conditional values from the picklist, then the user needs to populate the API name of the picklist values. All the API names should be entered in a comma-separated format.

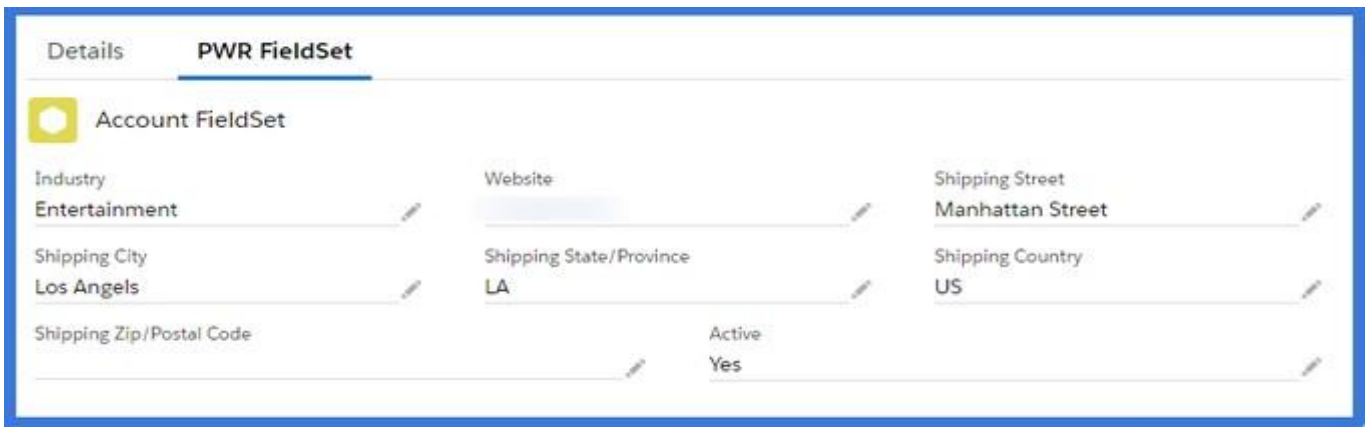
For example, for the Industry picklist on Account, the value needs to be entered as (Accounting,Advertising, Career & Hiring). If there are ten values defined on the field and the User wants to show fewer values, then the user can enter only those values.

**No spaces are valid between two values in the Picklist Values

Accounting, Advertising, Career & Hiring 

Accounting,Advertising,Career & Hiring 

PWR Field Set



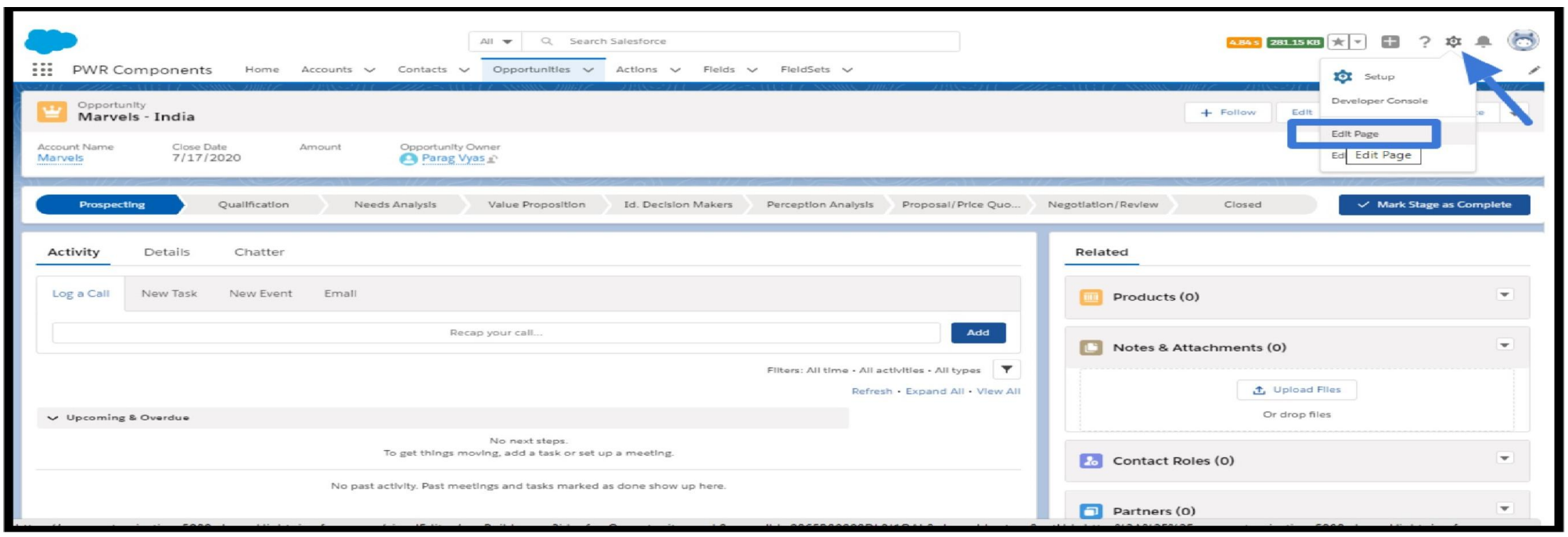
PWR Field Set component uses the standard Field Set provided by salesforce but the PWR Field Set extends the functionality of the Salesforce as an existing Field Set used in some Visualforce Page can be reused by the PWR Field Set show fields on the Lightning UI and as well as add extra fields to it.

With the help of this component User can show any number of fields on the record page. As per the standard restriction, PWR Field Set can show more than 8 fields on the UI, and along with that, this component supports Text Area, Formula, and Multi-Select picklist which are restricted in Standard Salesforce. PWR Field Set component is compatible for the Lightning Record Pages.

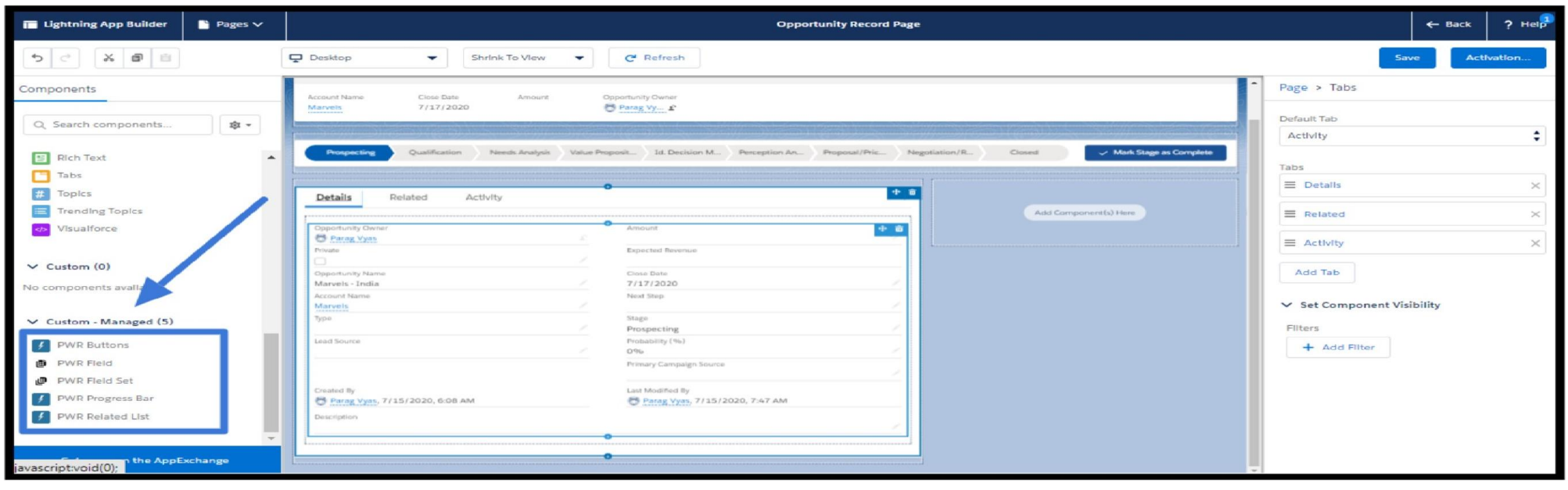
CONFIGURATION DETAILS

Here are the steps required to configure PWR Field Set on your Record Page, similarly you can configure the same on the Home page or App page in Salesforce Lightning.

To edit a record page, Click on Gear icon (setup) and click Edit Page



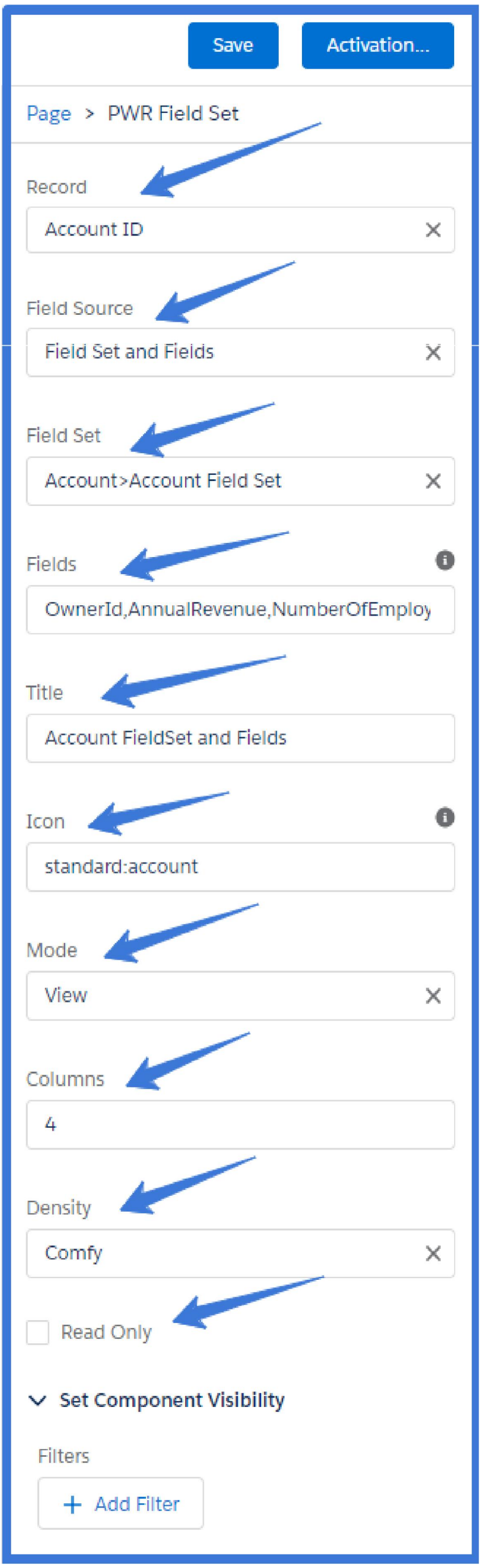
In Lightning App Builder, scroll down in Component panel (left panel), and at the bottom, you will find the list of all Custom Managed Components.



Drag and Drop PWR Field Set in the desired section of the record page

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

1. **Record:** Users can select the following records from the options for which are the fields can be from either the current record or the parent objects record or the grand-parent record.
    a. **Current Record:** If the user is using the PWR Field Set component on the Contact object, then the user can show the Fields and Field Set from the Contact object. For example, Name, Phone, or Email.
    b. **Parent Record:** If the user is using the PWR Field Set component, then the user can also fetch Fields and Field Set from the Parent object. In our example, users can select Account object.
    c. **Grand Parent:** If the user is using the PWR Field Set component, then the user can also fetch Fields and Field Set from the Grand Parent object.
2. **Field Source:** User needs to select the field source as either of the values given below.
    a. Field Set : When the 'Field Source' is chosen as 'Field Set' it will show fields configured in the Field Set selected.
    b. Field : When the 'Field Source' is chosen as 'Fields' it will show the order of the fields as mentioned in Fields property
    c. Field Set and Fields : When the 'Field Source' is chosen as 'Field Set and Fields' it will display the fields from the 'Field Set' and then the fields in the 'Fields' property.
    d. Fields and Field Set : When the 'Field Source' is chosen as 'Fields and Field Set' it will display the fields from Fields property and then fields configured in the selected Field Set.
3. Field Set: If the Field Source is selected as “Field Set” or “Fields and Field Set” or “Field Set and Fields” then user needs to enter the name of the Field Set exists on the object. The user needs to either create or reuse the existing Field Set on the object. Please refer to following link for creating a Field Set Click here.
4. Fields: If the Field Source is selected as “Fields” or “Fields and Field Set” or “Field Set and Fields” then user needs to enter the API names of the fields. Enter API names of fields in comma separated format in ‘Fields’. For example, on Contact object enter the Fields as Name, Phone, Email. ** No spaces between the comma are allowed
5. Title: Shows the title for the PWR Field Set on the UI.
6. Icon: This property is used to set an icon for the PWR Field Set. For more details around the icons, please refer the link here.
7. Mode: User can set mode for the Field Set as follows
    a. View Mode: It shows the Fields with an inline Edit symbol.
    b. Edit Mode: It shows the fields in edit mode on loading.
8. Column: User can configure or set the Number of Columns required to show the fields in the Field Set.
9. Density: User can set Density of the fields to be displayed as follows
    a. Comfy is a spacious view with labels on the top of fields and more space between page elements.
    b. Compact is a denser view with labels to the left of fields and less space between page elements.
10. Read Only: User needs to check the checkbox to make the fields Read Only on the UI.



PWR Button

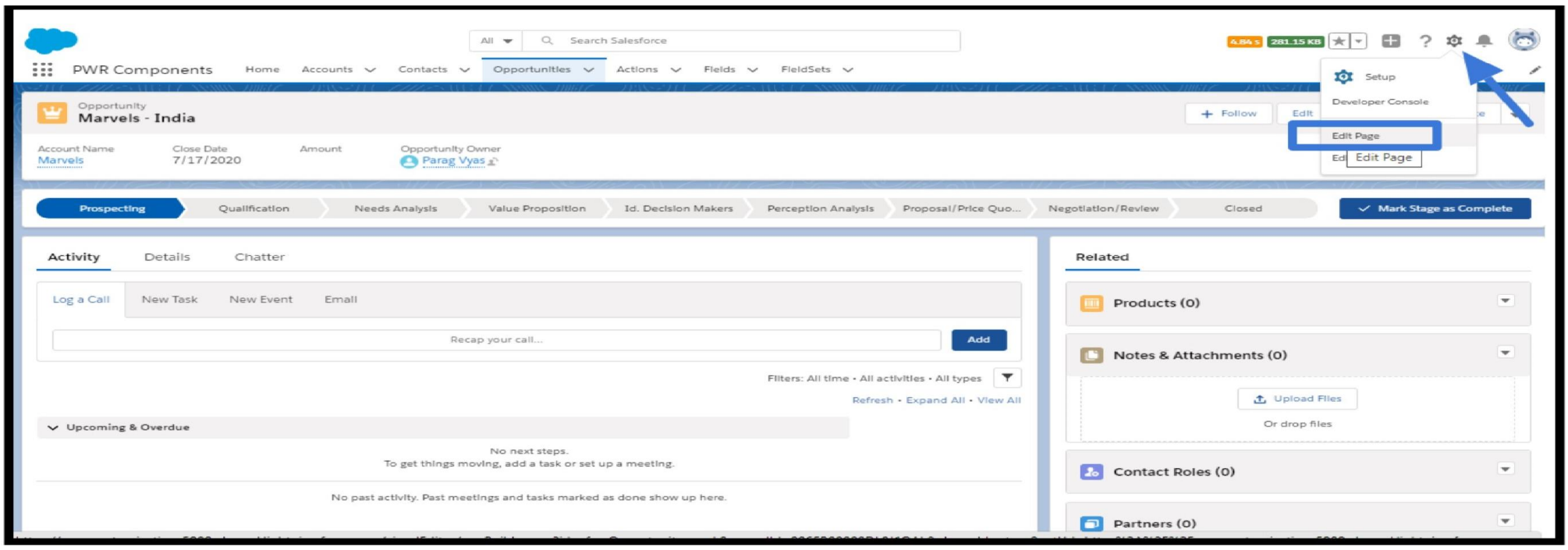
PWR Button are used for calling any actions on the Lightning UI. PWR buttons can be placed anywhere on the Lightning Pages. It looks like the button group in Salesforce. The advantage of using PWR Buttons is that it can be placed anywhere in the Pages and so it is easily accessible.

PWR Button supports Quick Actions, Global Actions and URL Actions. URL actions are supported on the Home Page, App Page and the Record Page, whereas Quick Actions, and Action Actions are supported only on the Record Page. PWR Buttons take input in JSON format and are Lightning compatible. PWR Button supports all the standard configurations which are provided by Salesforce.

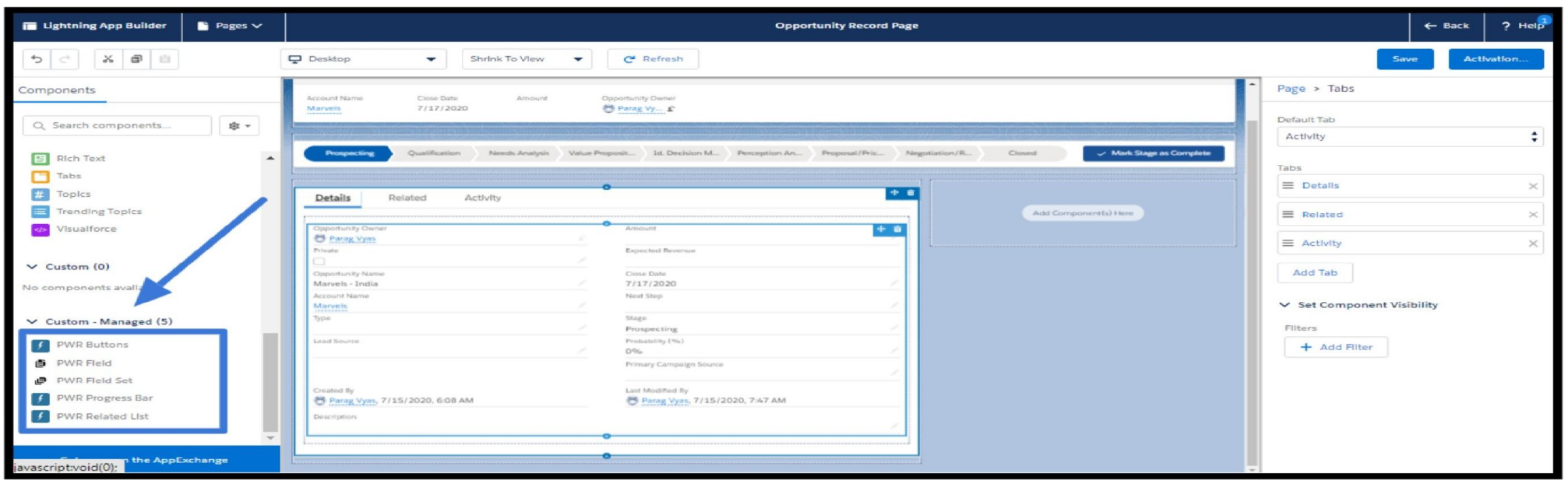
CONFIGURATION DETAILS

Here are the steps required to configure PWR Button on your Record Page, similarly you can configure the same on the App page in Salesforce Lightning.

To edit a record page, Click on Gear icon (setup) and click Edit Page



In Lightning App Builder, scroll down in Component panel (left panel), and at the bottom, you will find the list of all Custom Managed Components.

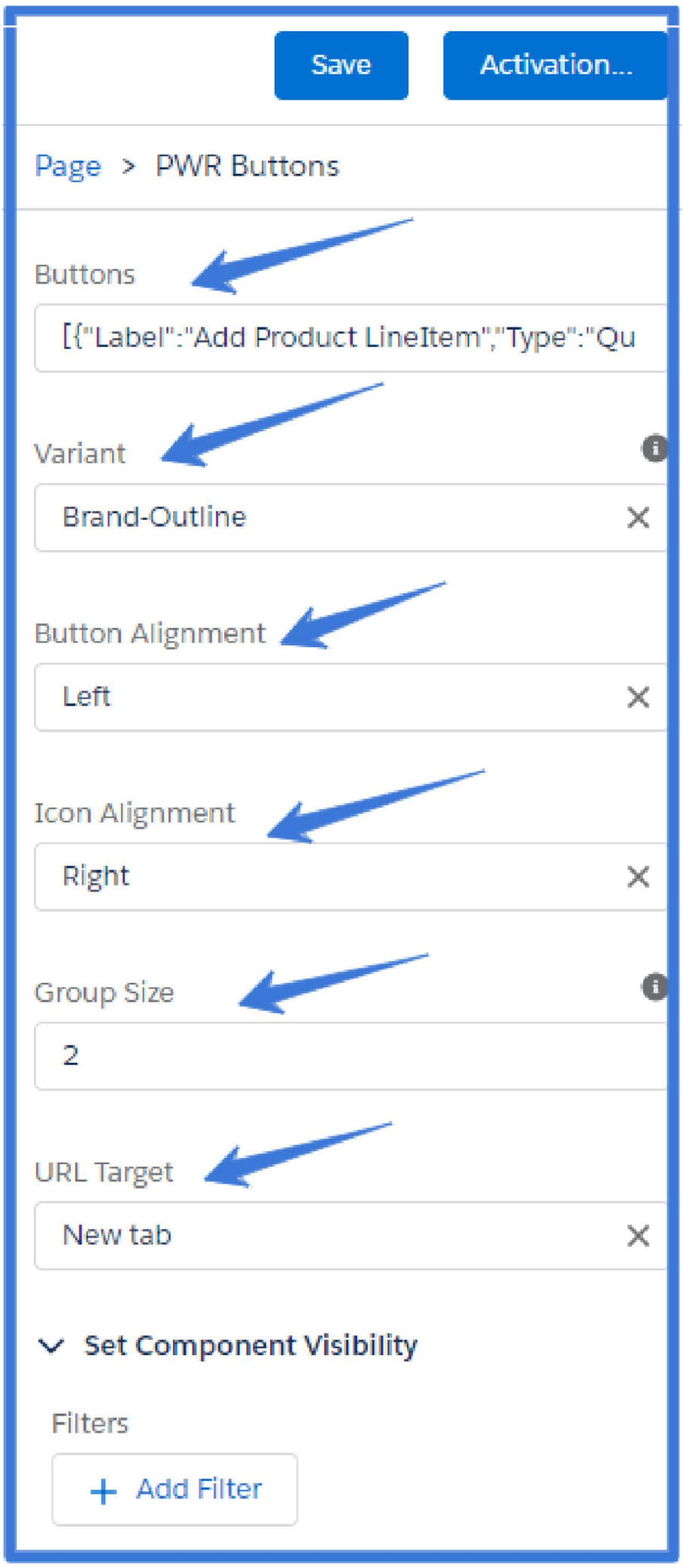


Drag and Drop PWR Button in the desired section of the record page

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

As per the requirements, the user can select the object from which they want to see the field

1. **Buttons :** User needs to input valid JSON for the Buttons property. Here are the params that are allowed in the JSON
    a. **Icon :** This is used to show Icon on the buttons. For icons you can refer the following link. This is an optional parameter in the JSON.
    b. **Label :** Stores the label for the buttons which will be displayed on the UI
    c. **Type :** This specifies the type of button. The 'Type' needs to be specified as ("Global action", "Quick action" or "URL").
    d. **ActionName :** Specify the API name of Action in case of "Global action" or "Quick action".
    e. **URL** - Specify the URL on which it is to be redirected when the **type** is "URL".
2. **Variant:** Users can change the appearance of the button by selecting one of the following options.
    a. **Base** - is a button without a border, which gives it the look of a plain text link.
    b. **Neutral** - is the default variant, a plain uncolored button.
    c. **Brand** - is a blue button, used to draw attention to the primary action on a page.
    d. **Brand-Outline** - is like brand but only the label and border are blue.
    e. **Success** - is a green button used to indicate a successful action.
3. **Button Alignment:** Users can set the alignment to either left or right, in which the buttons will be visible to the end users
4. **Icon Alignment:** Users can set the alignment of the icons on the button to either left or right, which will be visible to end users.
5. **Group Size:** If there are 5 buttons and the Group Size is set to 3 and then the rest of the 2 buttons will be shown in a drop-down menu.
6. **URL Target:** Users can set URL type buttons and can open the pages in either the same tab or on a new tab.



JSON EXAMPLE

[{"icon":"custom:custom55","Label":"Add Product LineItem","Type":"Quick action","ActionName":"Add_Product_LineItem"}, {"icon":"custom:custom14","Label":"Send Quote","Type":"Global action","ActionName":"Send_Quote"},{"icon":"standard:page","Label":"Visit Website","Type":"URL","URL":"https://google.com"}]

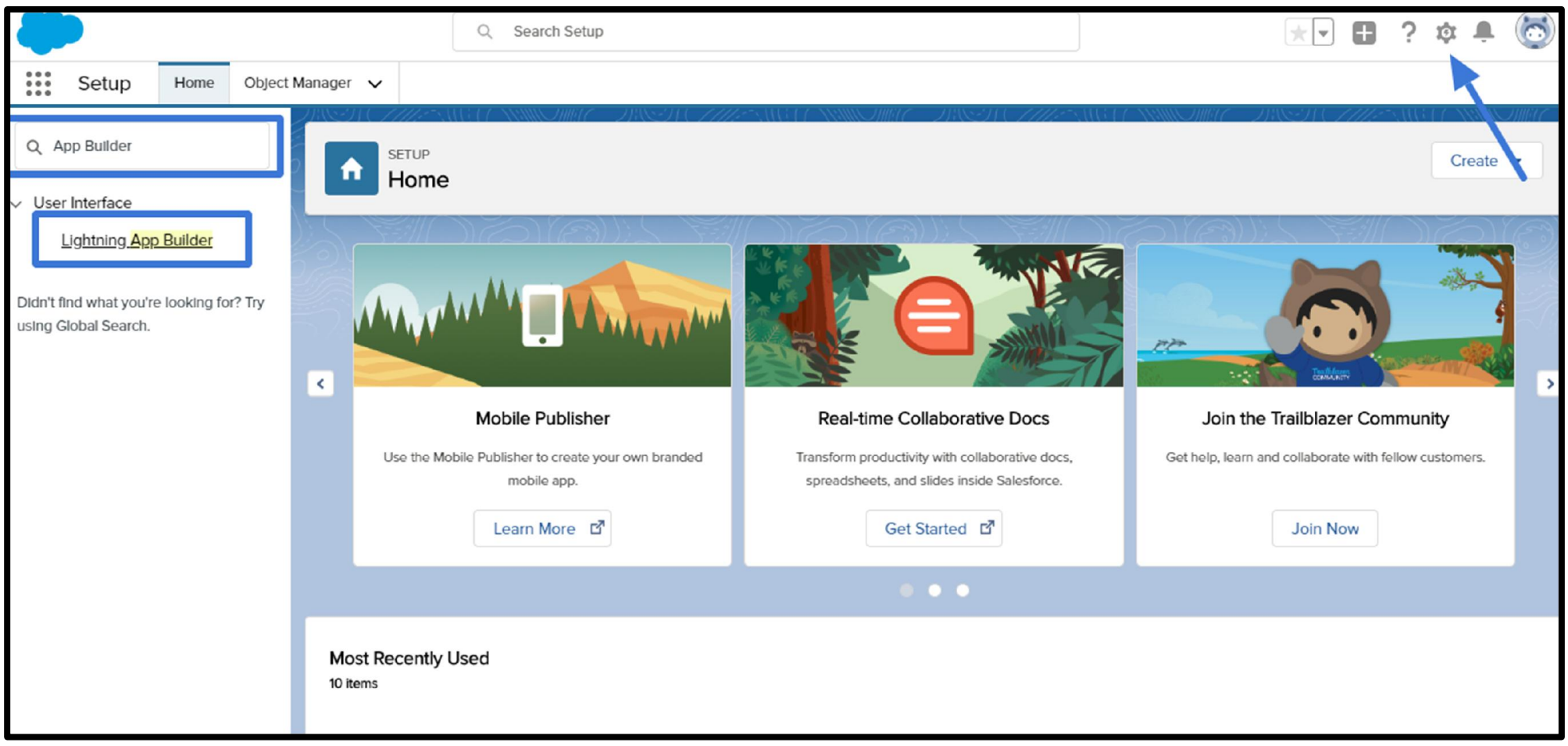
PWR ListView

PWR ListView is an easily configurable component which can be used to display the ListView on Home Page and Record Page. The ListView uses the Standard ListView and the User can configure it in order to display it on the Lightning UI by adding extra fields to the existing ListView. PWR ListView also allow users to create new ListView in case if there no existing ListView exists for an object. PWR ListView is Mobile compatible for the App Page.

PWR ListView supports Card View. ListView automatically gets converted into a Card when accessed through the mobile. The user can control the Actions to be shown on the ListView. The Action object is used to create actions like View, Edit, Delete and other custom Actions like opening a Lightning Tab or Lightning App.

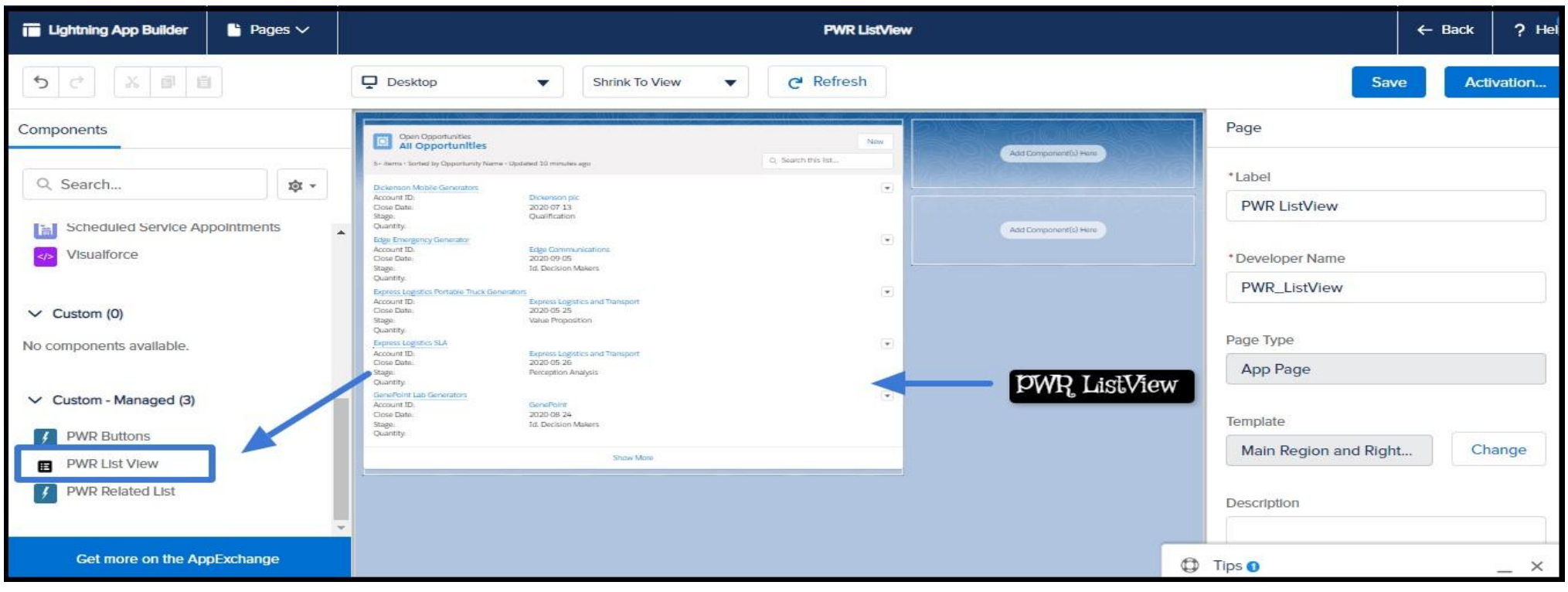
CONFIGURATION DETAILS

From Setup, enter App Builder in the Quick Find box, then select Lightning App Builder.



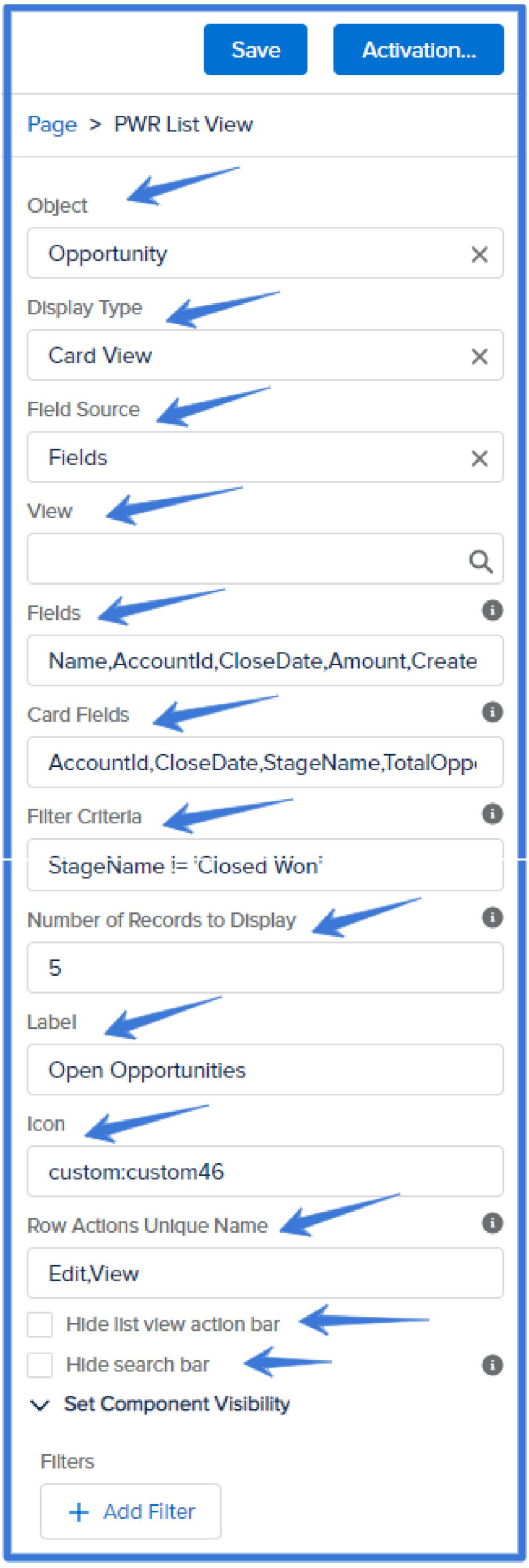
Click New and select any one of the App Page or Home Page and enter the Name of the page, select the screen layout and finish the steps of the wizard.

In the components pane under the Custom-Managed option PWR Button is present. Drag and drop it to the Record page.



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

1. **Object:** Select the object for which you wanted to create the ListView.
2. **Display Type:** Select the display type for the records, currently it supports two views as
   a. **Card View** : Records and the information are displayed in card format
   b. **Table View** : Records and the information are displayed in table format.
3. **Field Source:** User needs to select the field source as either of the values given below**.**
   a. Field : User needs to enter valid API name of the fields which user want to see in the ListView.
   b. View : User can select existing ListView and user will be able to see the fields which are configured in the existing ListView.
   c. Fields and View : It displays the fields configured in the Field property first, and then show all the fields in same sequence from the existing ListView selected
   d. View and Fields : It displays the fields from the existing ListView selected and then the fields which are configured in the Field property.
4. **View:** User can select existing ListView for the selected Object from this property. In case if they want to create, they need to create the ListView on the object first, and then access. This field become mandatory when the user select Field Source as View or Field and View or View and Field.
5. **Fields:** User needs to enter the API names of the Fields which the user want to see in the ListView. This field become mandatory when the user select Field Source as Field or Field and View or View and Field.
6. **Card Fields:** User needs to enter the API names of the Fields to be displayed on the card view. User can selectively pick up to 5 fields which they want to show on the Card View. If user don't update this property, the first 5 fields from the configuration will be displayed on the UI.
7. **Filter Criteria:** In case if the user wants to see the ListView with some specific filter criteria, then the user can put the criteria in SOQL format. For example - If user wants to see Not Closed Won Opportunities, then StageName != 'Closed Won'.
8. **Number of Records to display:** Stores the number of records needs to be displayed on first load of the page.
9. **Label:** Shows the label for the PWR ListView on the UI.
10. **Icon:** This property is used to set an icon for the PWR Field Set. For example, standard: account. For more details around the icons, please refer the link here.
11. **Hide List View Action Bar:** This property enables users to either show or hide actions buttons.
12. **Hide Search Bar:** This property enables user to either show or hide search bar.
13. **Row Actions:** User needs to enter the names of the Action records in comma separated format.
    For more details around Action, please click here.



PWR Related List

PWR Related List component can show the data directly derived from the related object like standard related list provided by Salesforce and is also capable of showing customized data derived from classes on the
Lightning UI. This component is Lightning and Mobile Compatible. PWR Related List has two views

1. Table View
2. Card View.

PWR Related List component allows the users to show fields from parent objects up to three levels. PWR Related List can show data which is present in filtered format and is defined in Apex business logic.

Example: If two objects data are combined and need to be shown in a single line which is defined in Apex class
can be easily configured and shown through the Related List.

In order to configure PWR Related List there are three objects named as **FieldSet**, **Field** and **Actions** are provided with the package. The Fields and FieldSet have a master-detail relationship. The user can create custom actions other than Edit, Delete and View under the object Actions. The Fields needs to be created under the FieldSet object. User can do the following customizations using the FieldSet and Field object:

FieldSet : For configuring the FieldSet, below properties needs to be set properly.

1. Name: Enter the FieldSet Name.
2. Object: Enter the API name of the object for which the FieldSet is being created.

** If the FieldSet is being created for displaying the data fetched using Apex Class then the API name of the object is not required.

Field : For configuring the Field, below properties needs to be set properly.

1. Label : The Label of the field can be changed for displaying on the Related List UI.
2. Sequence # : Sequence # of Field can be set and controlled.
3. Which field among the Fields in the FieldSet needs to be shown in the card view.
4. In case of Custom Class, the **Highlight** functionality can be used. If in a single related list there is data from two different objects and the first row represents data of object1 and the row below it shows the data from object2 you can Highlight any of the row depending on value present on the Fields record.
5. In the case of Custom Class, the **Link URL** functionality can be used. If in a single row of related list there is data from two different objects and the second column represents the field of Object2 and User wants the link to that object, then User needs to set it in the Link Field URL and Link Field Id.
6. The Date and Datetime fields can be shown in many custom formats using the **Custom Date Format** field. For example, Date(1/1/2019) can be shown as 31-Aug-2020.

Actions: This object is used to create actions like View, Edit, Delete and other custom Actions like opening a Lightning Tab, Lightning App and Open custom Component. Users can control the Actions to be shown on the Related Lists. If User has added Edit, Delete, action on the Related List and any user in the system does not have Edit or Delete Access then the Actions gets hidden for that user. The various Actions needs to be created using the following syntax:

Name	Operation Type	Component URL	Parameters	Description
View	View Record			This action is to View Records. This can be opened in the same tab or on a new tab.
Edit	Edit			This action is to Edit the Record.
Delete	Delete			This action is to Delete the Record.
Open App	Lightning App	c__PWR_Components		This action is to navigate to a Lightning App in the same tab or new tab. API name of the Lightning App, c__ as a prefix is mandatory

Record Relationship Page	Record Relationship Page		{
			"recordId": "{!Id}",
			"objectApiName": "Contact",
			"relationshipApiName": "Cases",
			"actionName": "view"
			}
				This Action is to open the current records related records page. For example if the record of Contact is open this action can navigate to the related Cases of it. The parameters in the JSON needs to be specified as follows:
				
				recordId - Current record id
				objectApiName - Current record's object api name needs to be mentioned.
				relationshipApiName - Specify the API name of the object's relationship field (look-up or master-details).
				actionName - The action name to invoke. Only 'view' is supported.

Open Object Page	Object Page		{
			"attributes": {
			"objectAPIName": "Contact",
			"actionName": "list"
			}
			}
				This Action is to open a page that navigates to a standard or custom object Page.The parameters in the JSON needs to be specified as follows:
				
				actionName - Specify the action name to invoke. Valid values are 'home', 'list', and 'new'.
				objectApiName - Specify the API name of the standard or custom object.

Custom Tab	Navigation Item Page	Test_VF_page		This Action displays the content mapped to a Custom tab. Specify the API name of the Custom Tab.

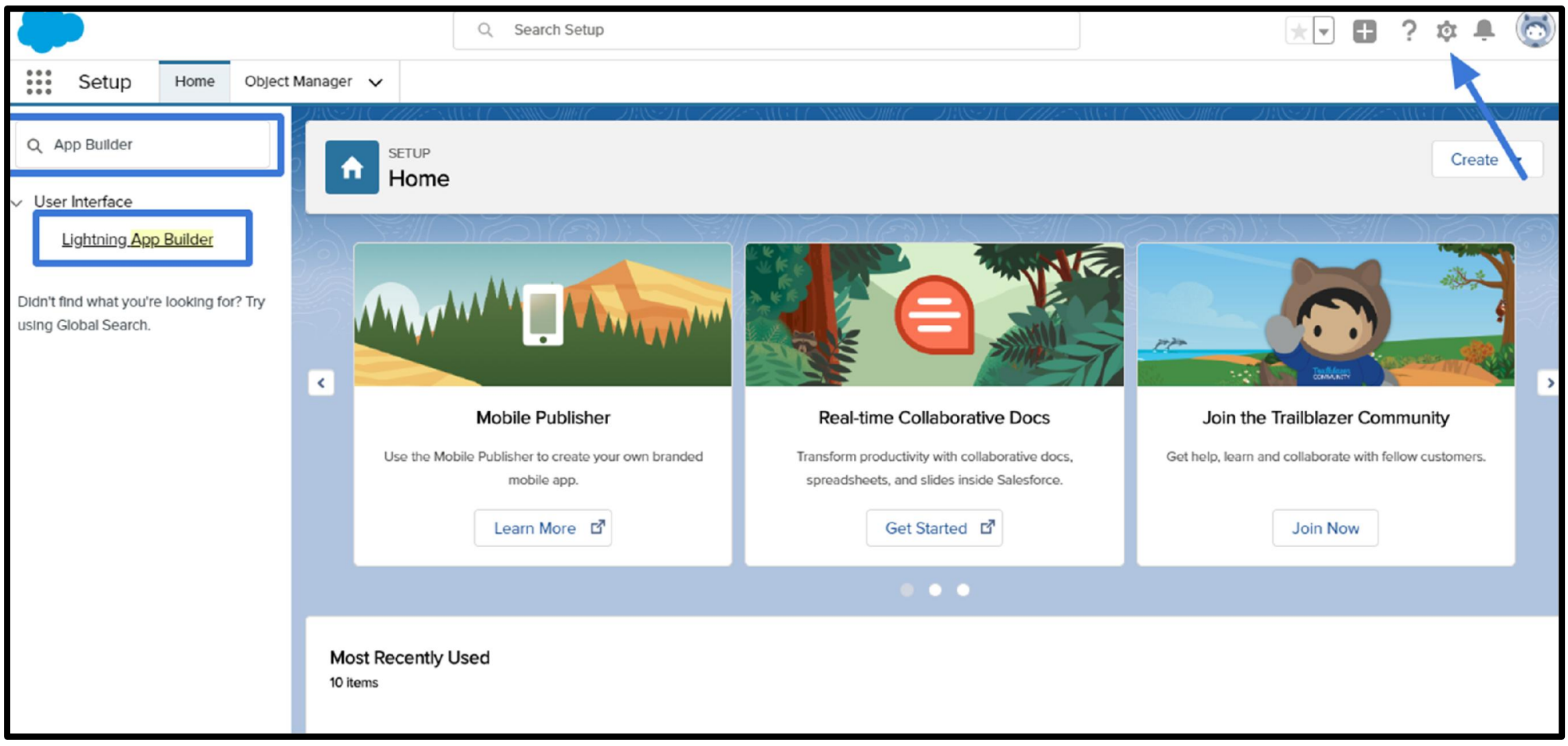
Knowledge Article	Knowledge Article		{
			"articleType": "Knowledge",
			"urlName": "Test-Knowledge-Article"
			}
				This action is used to open any 'Knowledge Article' in the JSON format provided. The parameters need to specify as follows:
				
				articleType - Specify the Knowledge Article record "articleType" API name.
				urlName- Specify the name of the Knowledge Article.

Open Component	Open Custom Component	c__partner	{
			"c__email":"{!Email}",
			"c__description":"{!Description}",
			"c__accountname":"{!Account.Owner.Name}"
			}
				This action is used to open the custom Lightning Aura component. The parameters that are to be passed to the component needs to be specified as provided in the JSON format.
				
				For example,
				Using value from current record's field, specify field the API name :
				{
				"c__email":"{!Email}"
				}
				
				Using static value :
				{
				"c__email":"xyz@abc.com"
				}

Standard Page	Standard Page	chatter		This action is used to navigate to a standard page. For Example Home,Chatter.
Webpage	Webpage	http://youtube.com OR /{!Id}		This action is to specify the URL to navigate on. The user can either give a static url as: http://youtube.com Or The user can dynamically bind the query-parameters value to the URL with binding syntax as: 'https://www.xyz.com?firstName={!fieldApiName}'

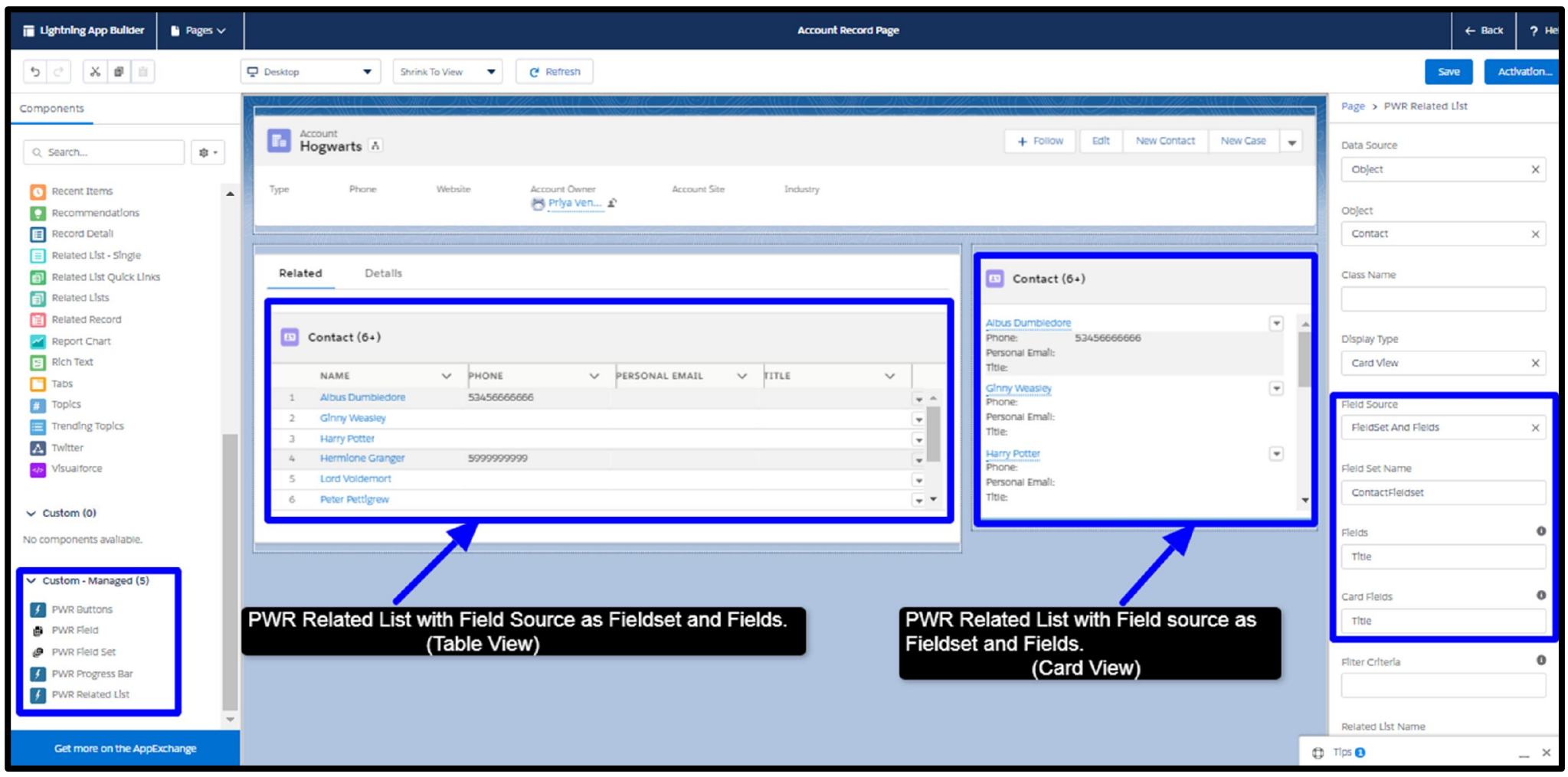
CONFIGURATION DETAILS

From Setup, enter App Builder in the Quick Find box, then select Lightning App Builder.



Click New and select any one of the Record Page, App Page or Home Page and enter the Name of the page, select the screen layout and finish the steps of the wizard.

In the components pane under the Custom-Managed option PWR Related List is present. Drag and drop it to the Record page.



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

Click Setup, Search App Builder in the Quick Find box, then select Lightning App Builder

1. Click New.
2. Select Record Page and enter the Name of the page, select the object and finish the steps of the wizard.
3. In the components panel (left side) under the Custom-Managed option PWR Related List is present. Drag and drop the PWR Related List.
4. Fill the Property Fields on the components pane to configure the PWR Related List:
    a. **Data Source**: User needs to select the data source as Object or Custom Class.
    b. **Object**: If the Data Source is chosen as Object then the User needs to select the object.
    c. **Class Name**: If the Data Source is Chosen as Custom Class then the Class Name of the Class needs to be entered.
        i. When the Data Source is chosen as Custom Class the Field Source should be Fieldset only.
    d. **Display Type**: The type can be Table view or Card View.
        i. Note: Even if the Display Type is chosen as Table View but by default it is switched to Card view in mobile.
    e. **Field Source**: The Field Source are Fields/ Fieldset/ Fieldset and Fields/ Fields and Fieldset.
        i. When the 'Field Source' is chosen as 'Fieldset' it will show the order of the fields as mentioned in the Field Set.
        ii. When the 'Field Source' is chosen as 'Fieldset and Fields' it will take the fields from the 'Fieldset' and then the User needs to enter the API names of the fields in the 'Fields'. While displaying it will display the fields of the Fieldset first and then the Fields.
        iii. When the 'Field Source' is chosen as 'Fields and Fieldset' it will take the fields from the 'Fieldset' and then the User needs to enter the API names of the fields in the ' Fields'. While displaying it will display the fields which are manually entered and then the Fields of the Fieldset.
    f. **Field Set Name**: The name of the Fieldset defined on the object Fieldset needs to be provided in this property.
    g. **Fields**: User needs to enter the API names of the Fields to be displayed.
        i. If the Field Source is chosen as Fields then user needs to enter API names of fields in Fields property and also User needs to enter in the card fields or else in the mobile view it will only show name,
    h. **Card Fields**: User needs to enter the API names of the Fields to be displayed on the card view.
        i. Note: User needs to enter the API names of the fields in the Fields property and then Card Fields or else it will not show up in the card view.
        ii. The card view supports only 5 fields at maximum.
    i. **Filter Criteria:** The input needs to be given in normal SQL format.
        i. This property only works for filtering data when the Data Source is Object.
        ii. If there is a Stage field on Opportunity and User wants to show only Opportunities at Stage ‘Prospecting’ and ‘Closed Won’. Users need to add the Filter Criteria as StageName IN ('Prospecting','Closed Won').
    j. Related List Name: This property is for naming the Related List as per Users convenience.
    k. Icon: This property is for setting an icon for the related List.
    l. Number of Records to Display: This property is for showing the number of records to display when the record is first opened.
    m. Row Actions: User needs to enter the names of the Action records in comma separated format.
    n. Hide Row Number Column: Users can hide the serial number column but check this checkbox.