Model Driven Apps In a Day

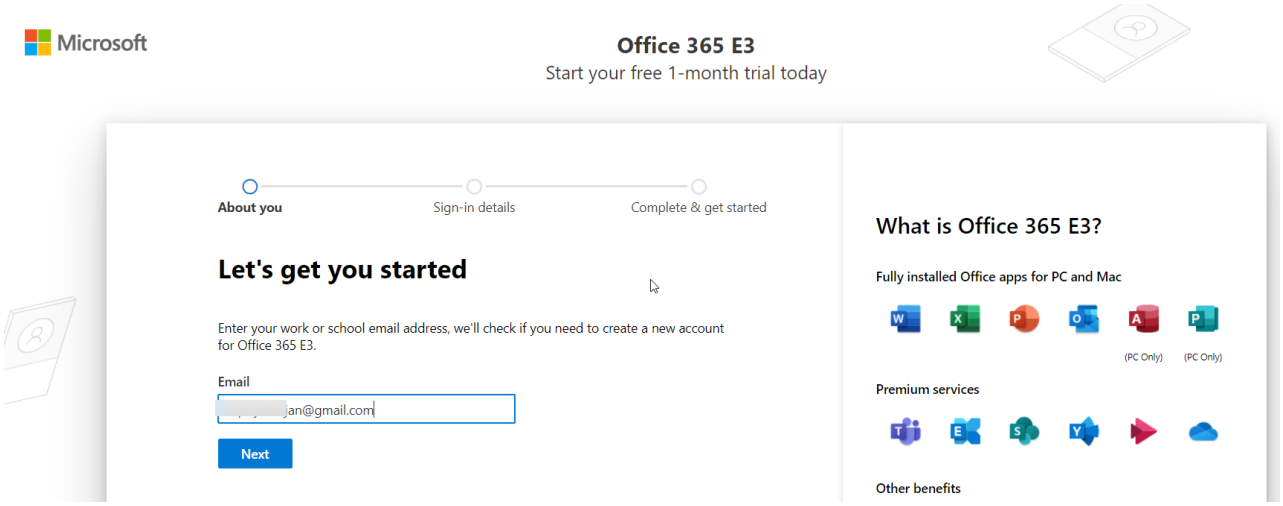
Author: Priyaranjan KS

Contents

Sign up for Office 365 E3 Tenant

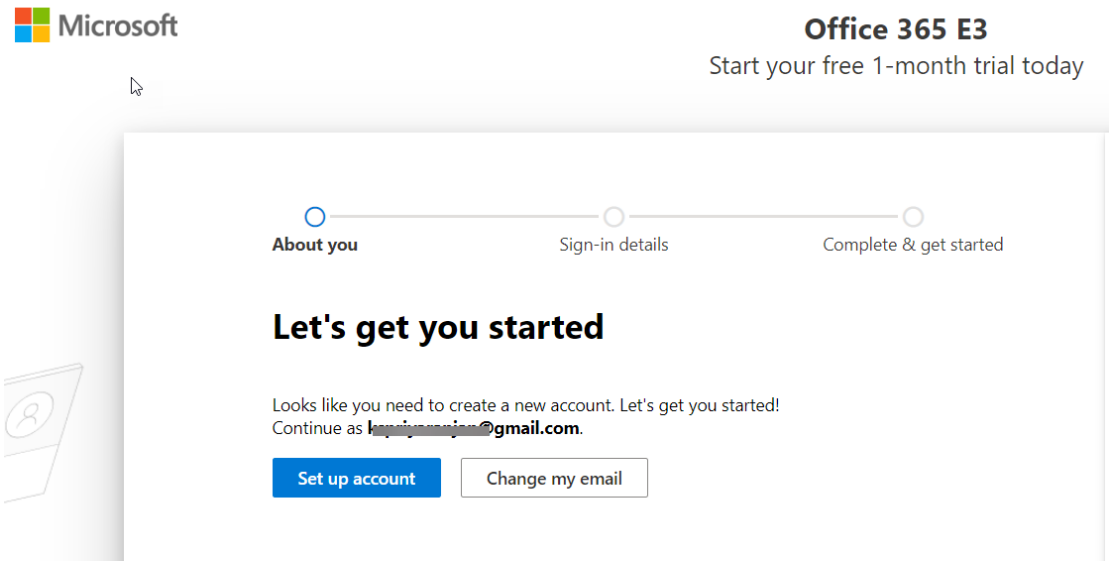
The sessions will involve hands on labs which you will be performing along with the instructor in your personal tenant. To provision the personal tenant, follow the below steps.

Step 1: Open browser Google Chrome (in Incognito mode) or Edge (in Private Window) and head over to the URL . Specify your Email ID and click on Next.

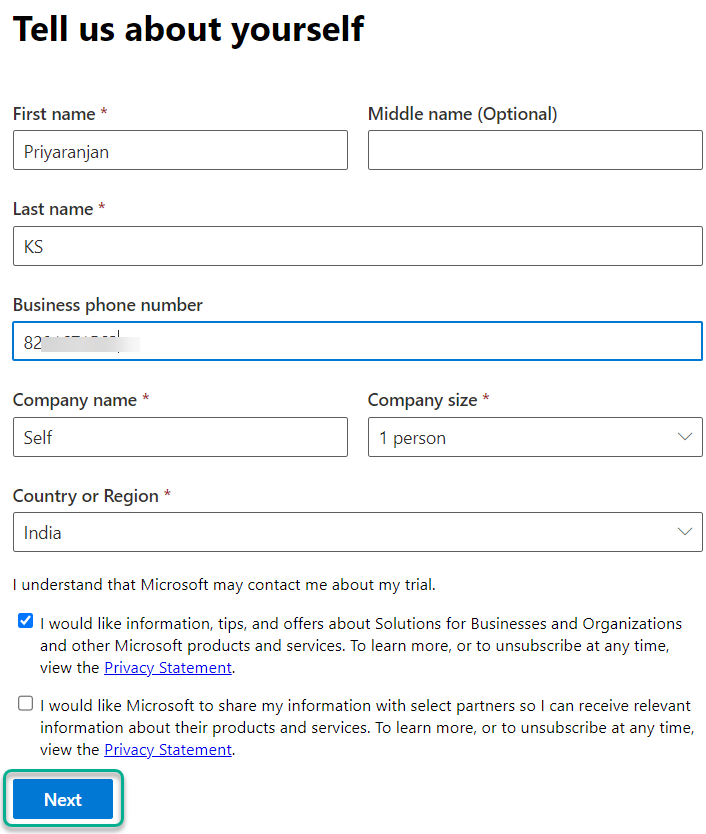


Note : In case you specify a personal email address like gmail, you will be asked to create a new work account.

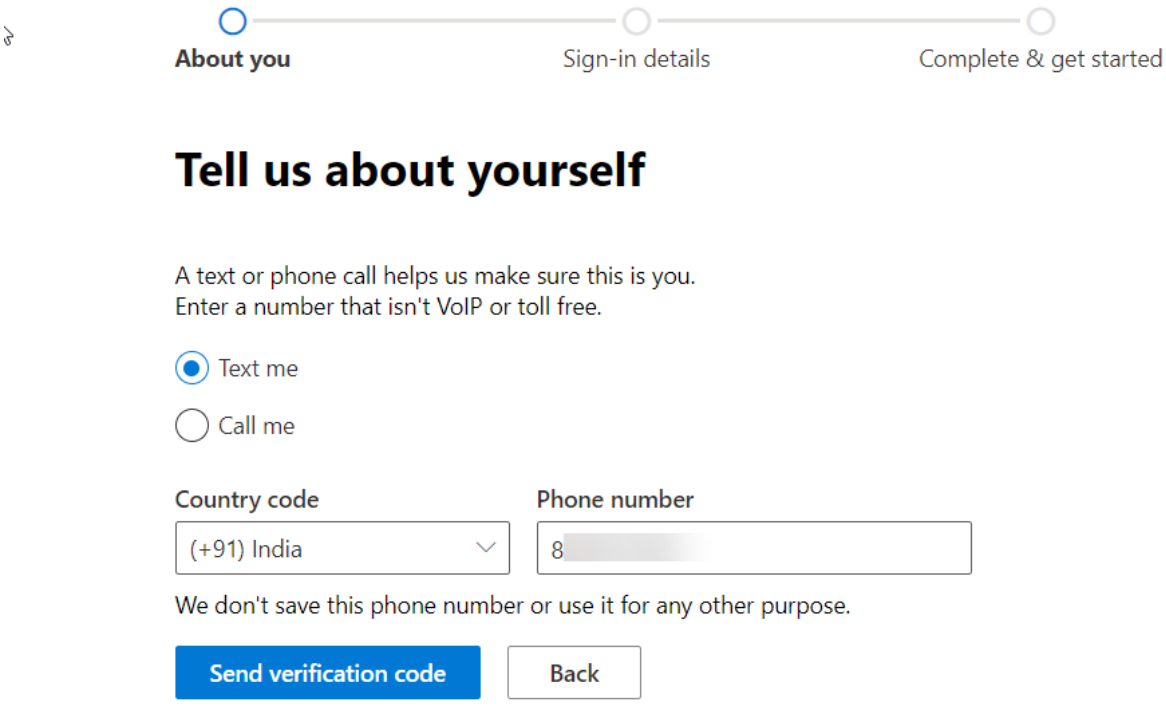
Step 2 : Click on Set up account.



Step 3 : Specify the personal details in the next screen and click on Next



Step 4: Specify the phone number to perform the verification step.

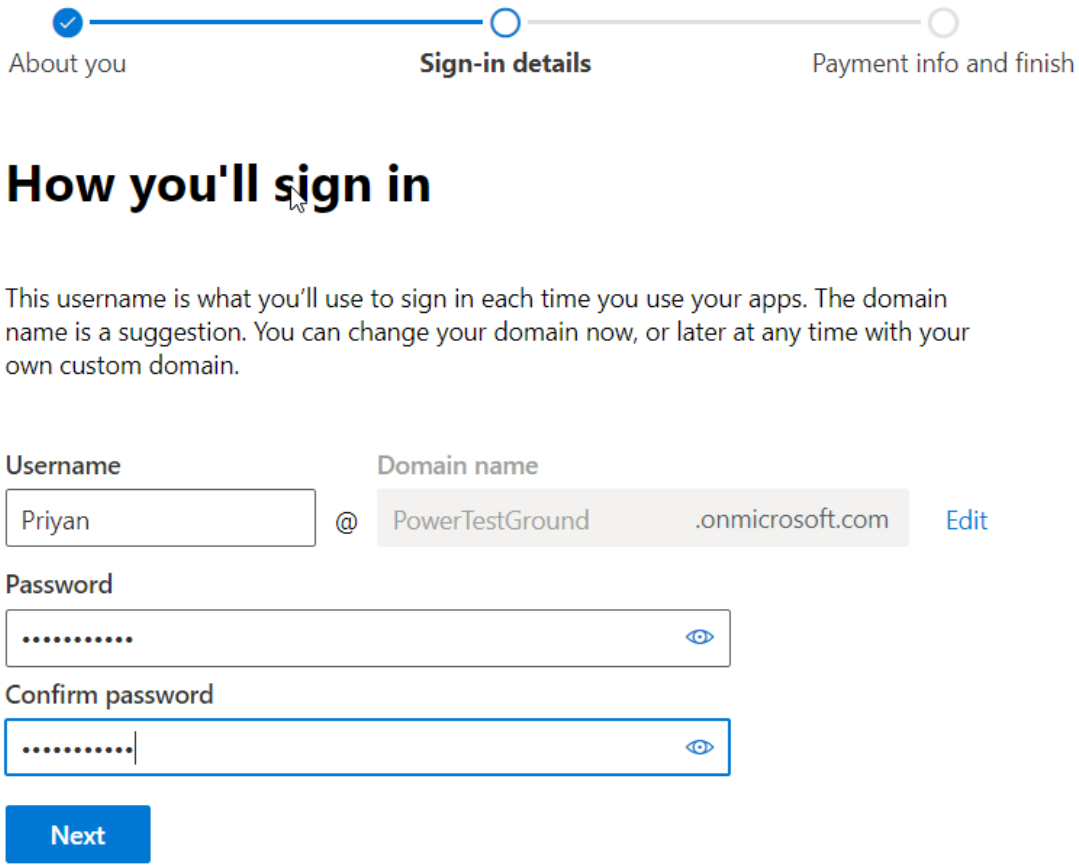


Step 5: Specify the username, domain and password in the next screen. The domain will have to be a

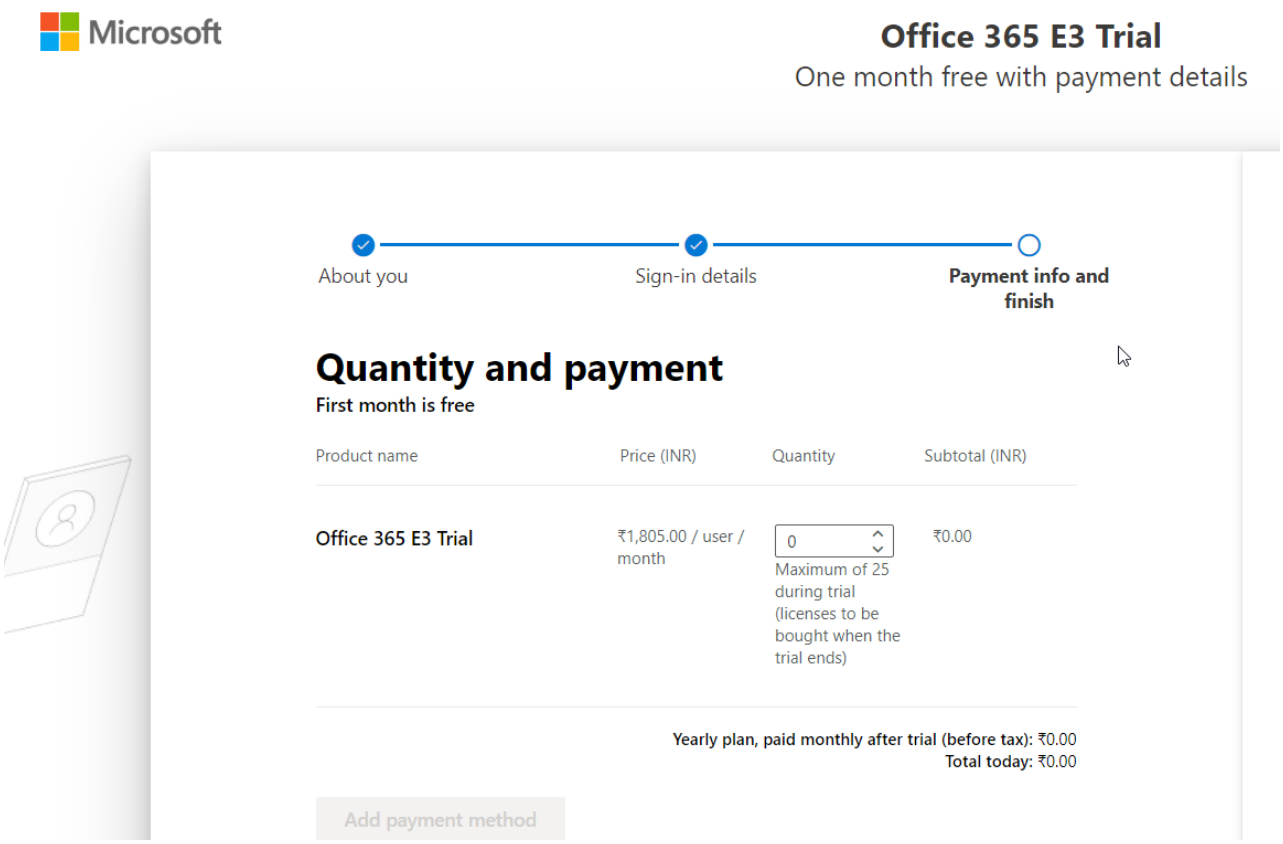
unique text. Click on Next.

Note:

- Make a note of the password entered in this step.

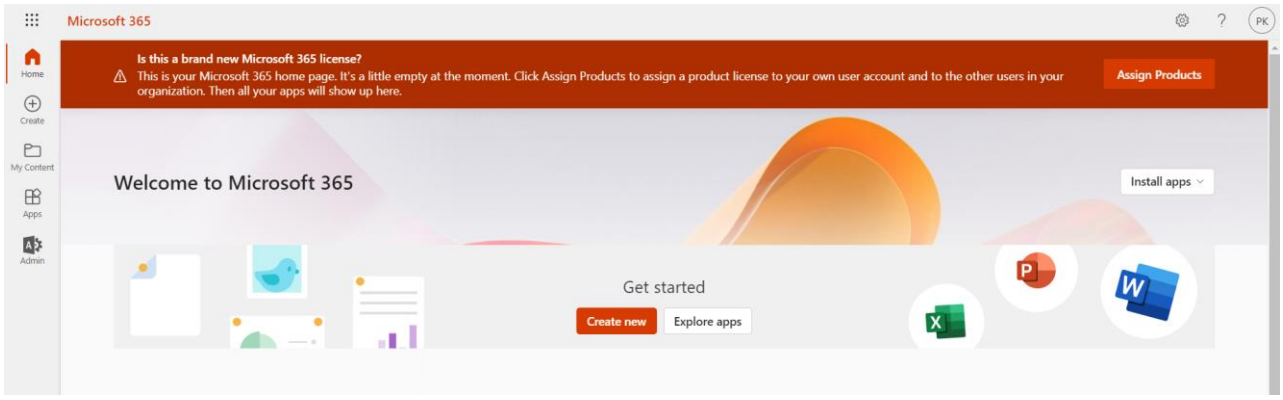


Note down the username and password for all future Office 365 Sign Ins.



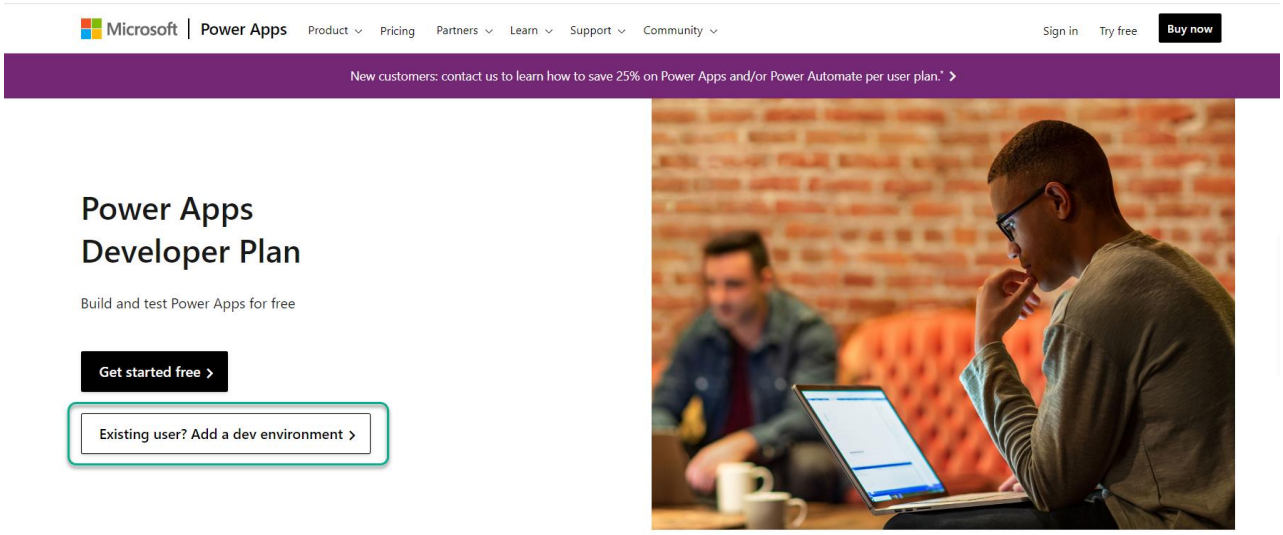
This will take you to the payment page. Since you just want to have hands on trial and if not looking to extend the trial subscription, you can skip the payment information addition step by closing the tab.

Open a new tab and head over to https://www.portal.office.com and sign in with the recently saved office 365 credentials.

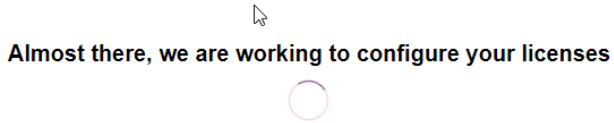


1. Apply Power Apps developer plan to Office 365 Tenant

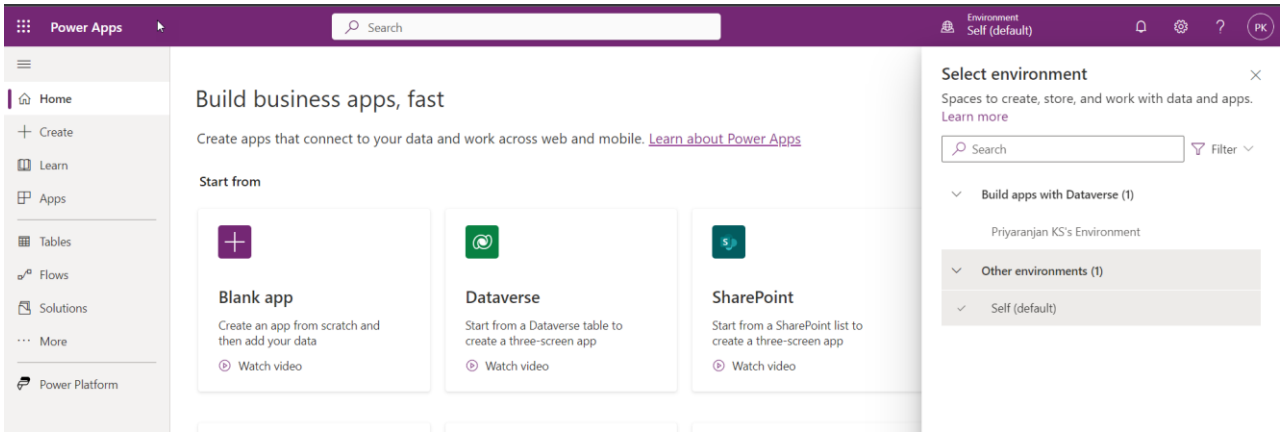
Header over to this link and click on existing user



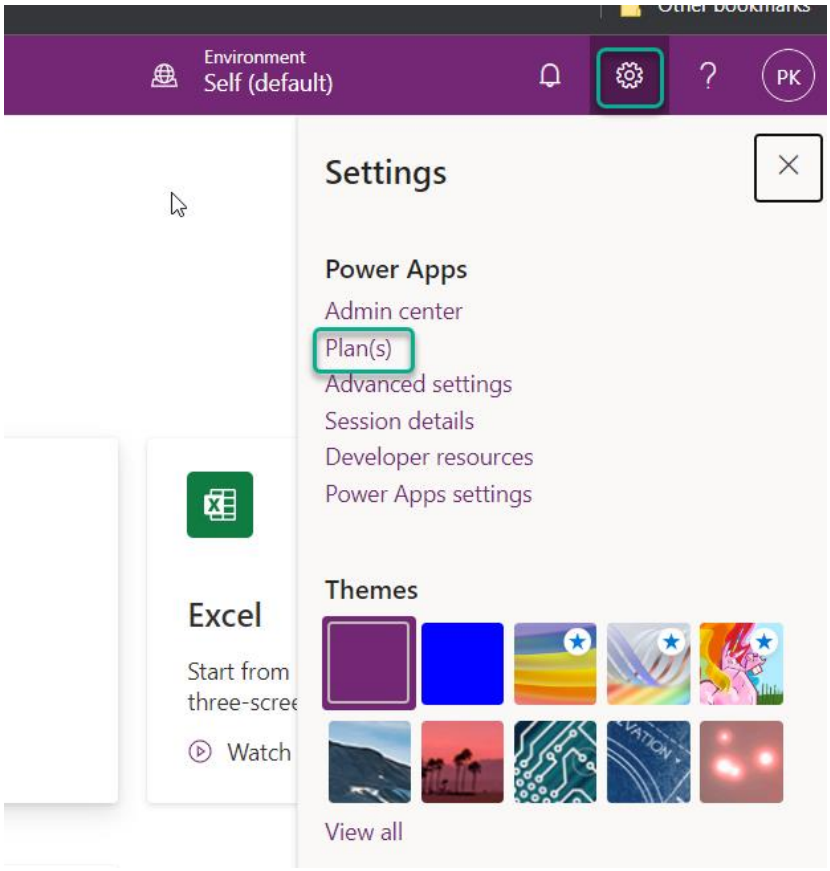
Select the recently created onmicrosoft.com account if the Login screen pops up. It will auto apply the developer license to the tenant.



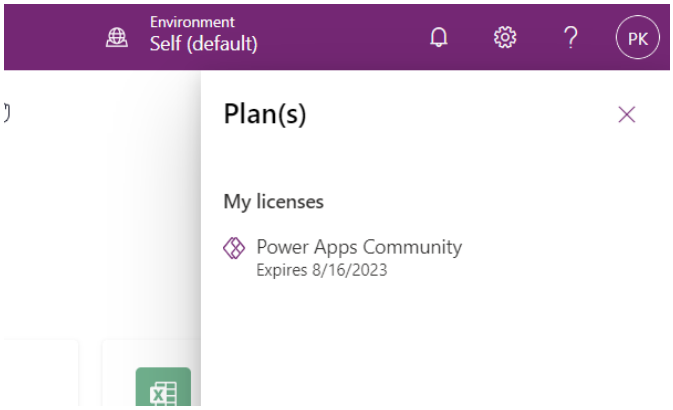
The developer license has been applied and it opens up the Power Apps Environment.



To ensure if the developer license has been applied, head over to the gear icon and select Plans

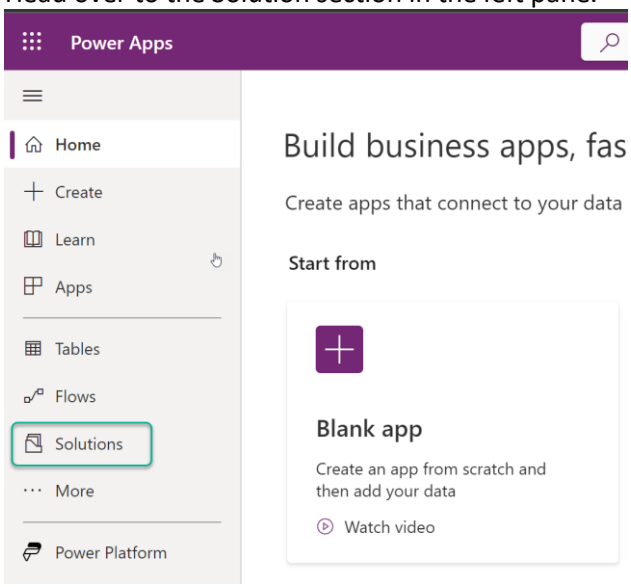


It will show the community plan as part of the developer license.

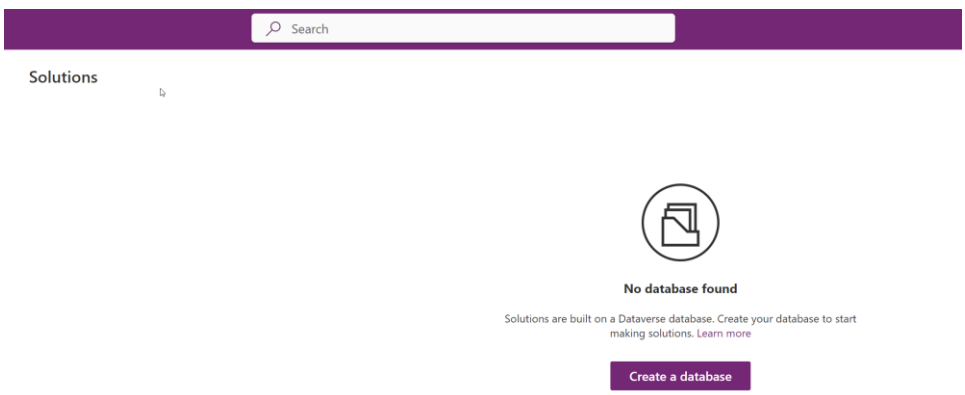


Create Solution

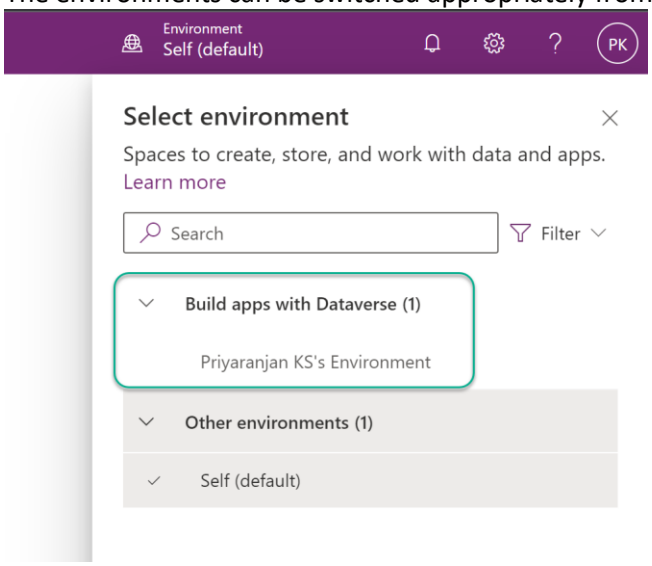
Now lets create a solution which will act as the container for the elements of the model driven application. Head over to the Solution section in the left pane.



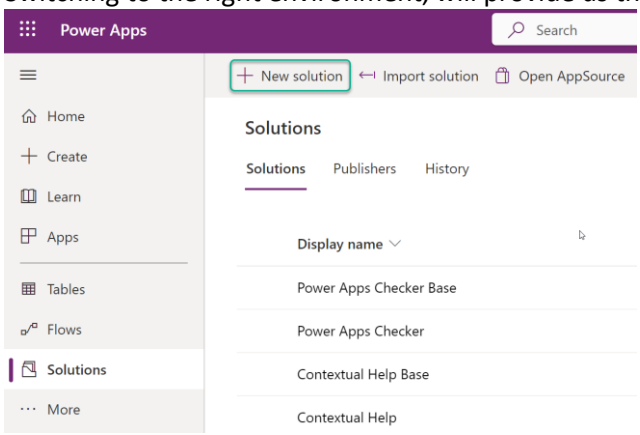
Ensure that you are in the right environment that has dataverse database associated with it, else you will get the below message to create a database



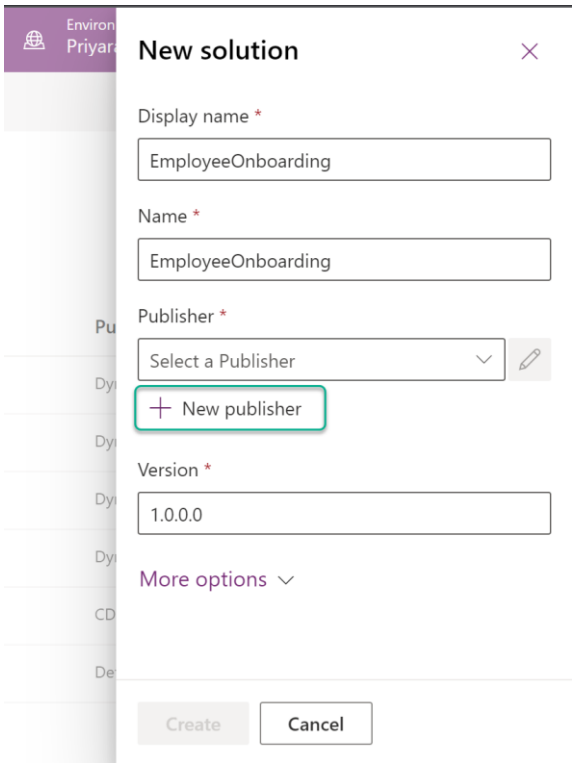
The environments can be switched appropriately from the top right ribbon.



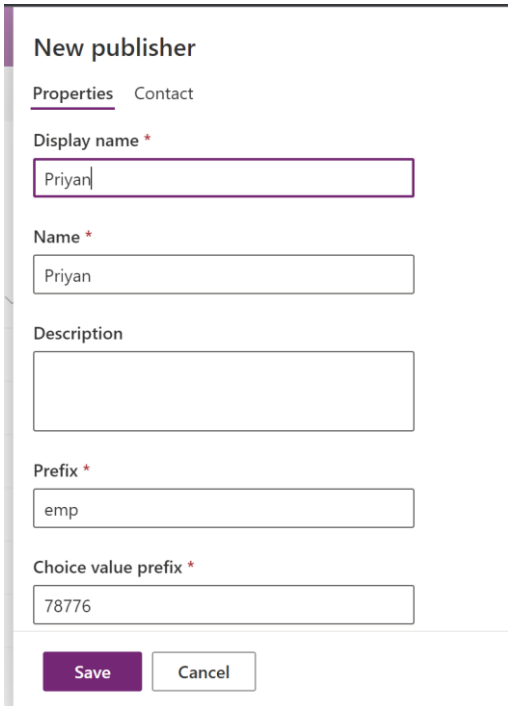
Switching to the right environment, will provide us the option to create a solution. Click on New Solution



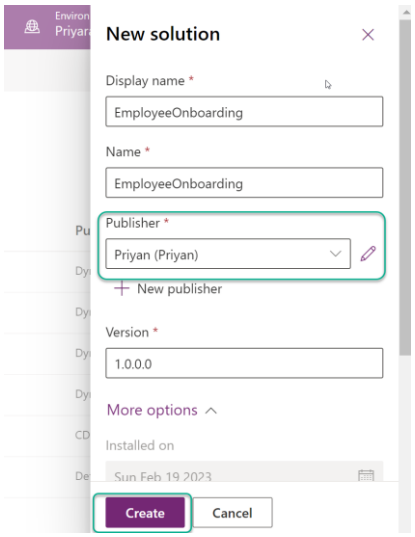
Specify the solution name as “Employee Onboarding” and click on Publisher to create a publisher that will be associated to the custom entities that are created.



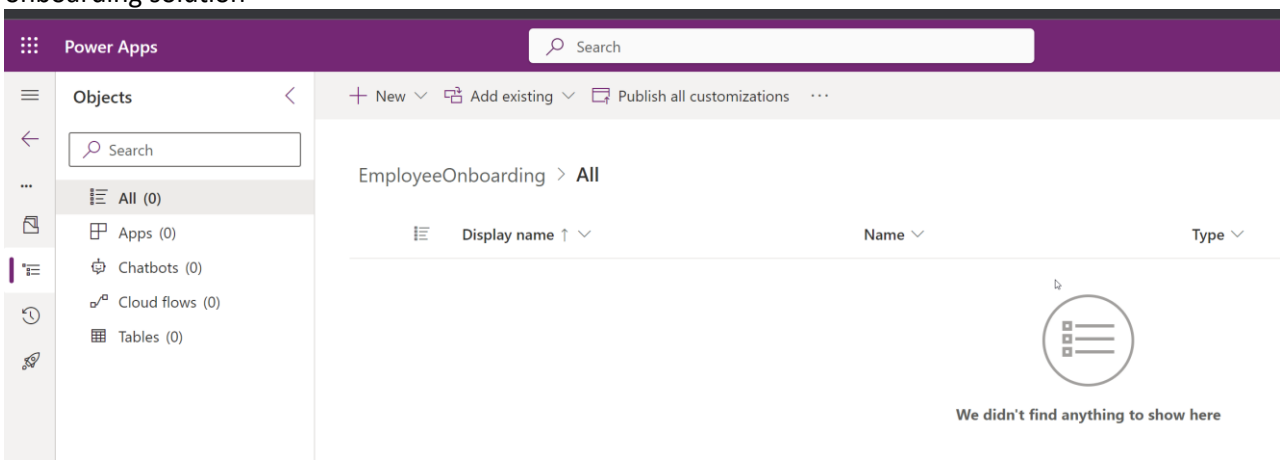
Specify the Name, display name. Mention the prefix which would be appended to the customer columns and tables created in the solution. Click on Save.



This will bring us back to the previous pane where we were creating the solution. Select the recently created publisher and click on “Create”



Thus the solution is created where we can add various elements that would make up the employee onboarding solution



Create Tables and Forms

We will be creating the below 5 tables and associated forms with these tables as part of the solution.

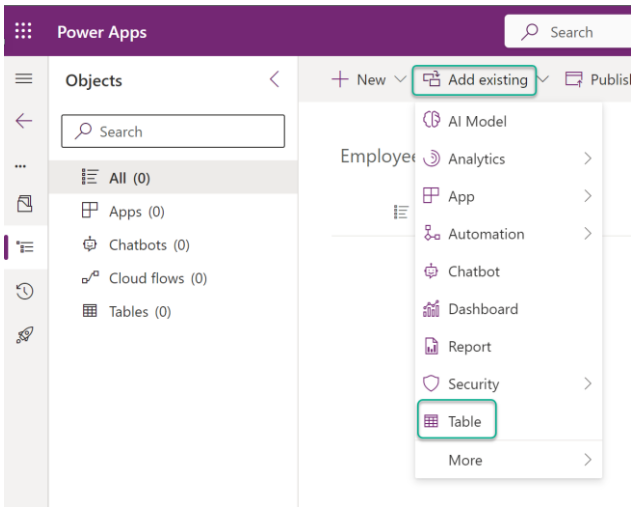
1. Employee – Hosts the information of the Interview Panel Employees
2. Candidate – Holds the information of the external interview candidates.
3. Job Entry – Contains information about the job opening and maps to the candidates appearing for the interview
4. Interview Feedback – Holds the candidate interview feedback details
5. Onboarding- Contains the details of the interview cleared candidates and their onboarding details

Table 1: Employee

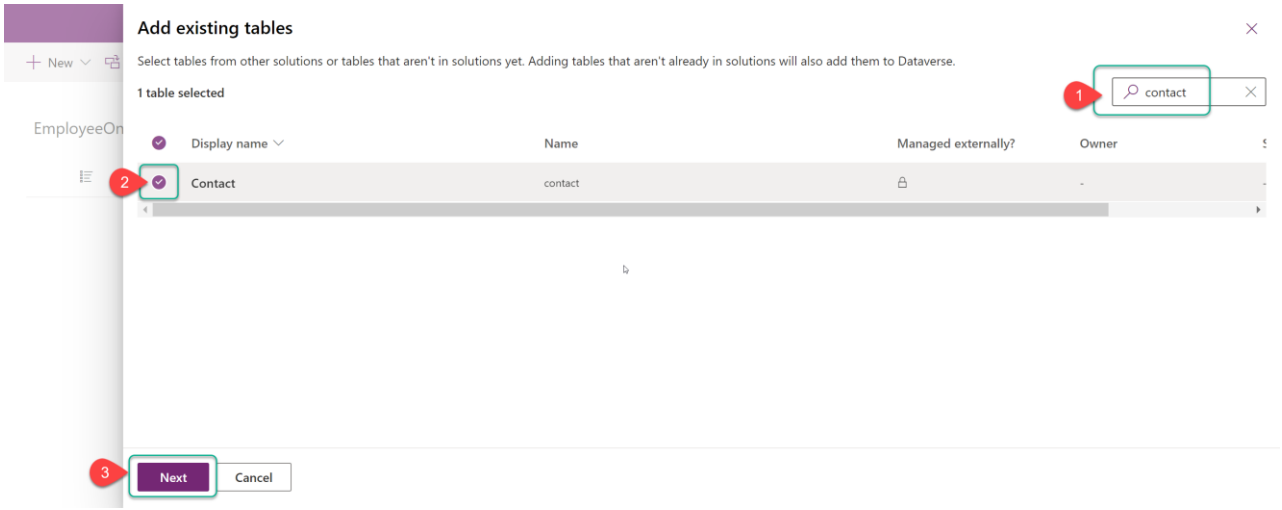
The employee table will hold the information of the internal employees of the company who will be part of the interview panel. So as to create the employee table , we will make use of the existing out of the box “Contacts” table which will contain columns that relate to a person (in our case the employee).

Create Employee Table

So as to create the table, head over to the recently created solution and select the option :
Add existing -> Table

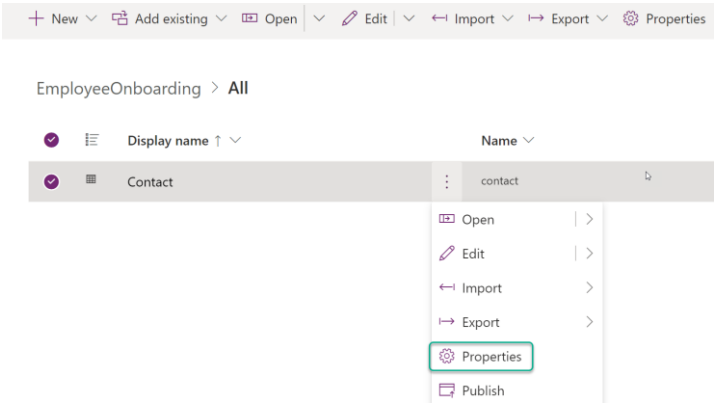


This will list out numerous out of the box tables available in the dataverse. To narrow down to the contact table that we would like to use, type in “contact” in the search box and select the contact table row. Click on Next.

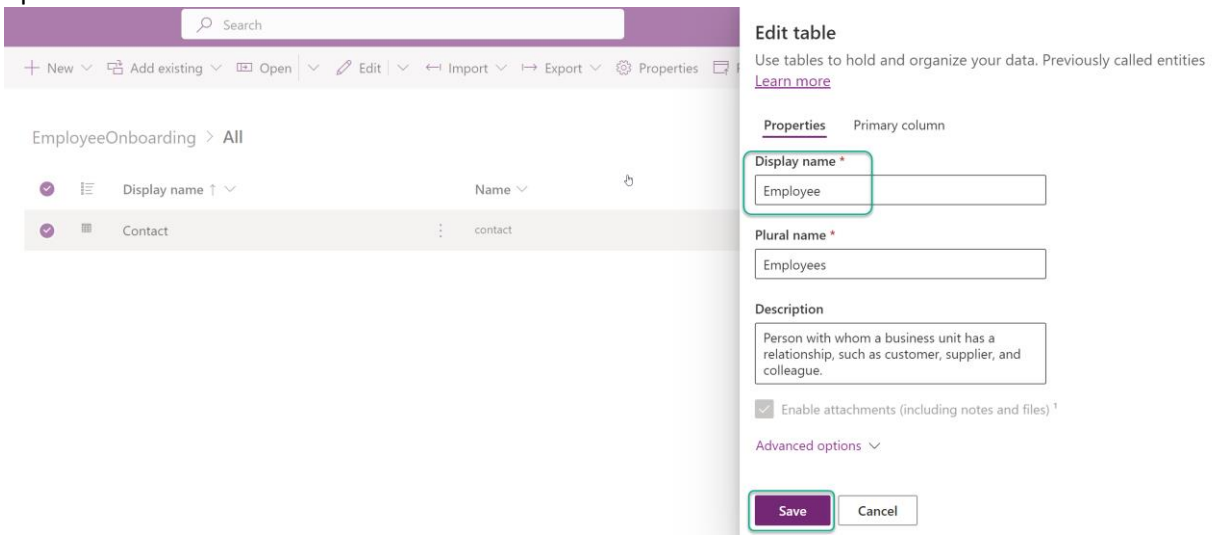


Keep the “Include all objects” and “Include table metadata” options unselected and click on “Add”

This will add the instance of the contacts table. Lets rename the table by selecting the “Properties” option



Let's rename the table to "Employee" in the display name field. The plural name will get automatically updated. Click on Save.

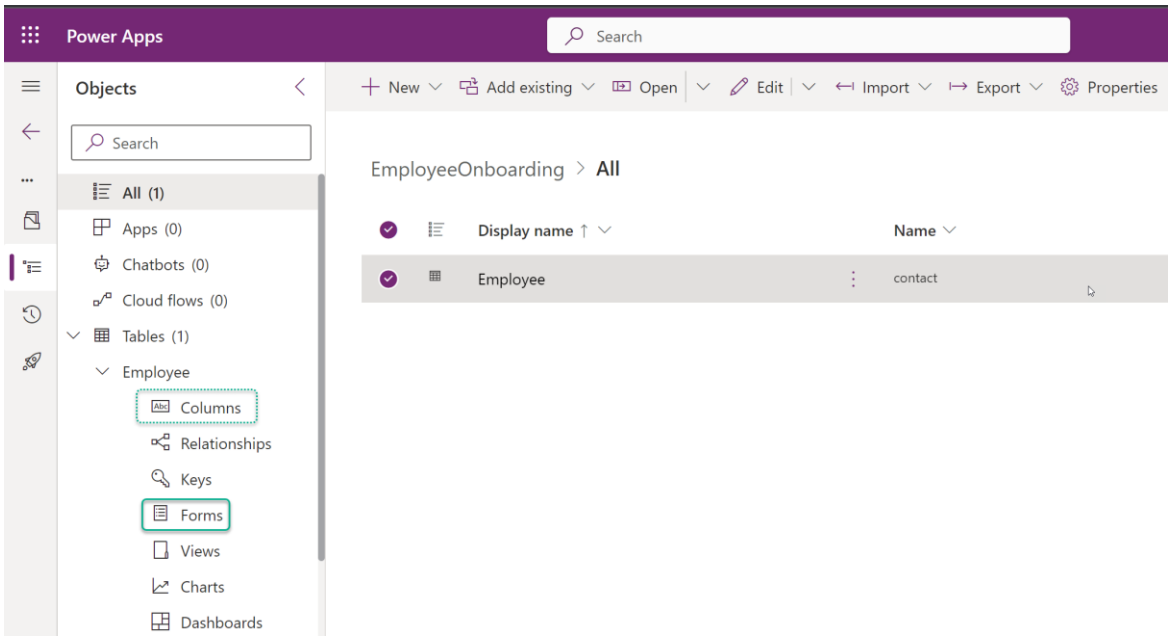


We will use the default available columns in the table to store the internal employee information. In case we need to add custom columns, we can do that by selecting the "Columns" option from the left pane.

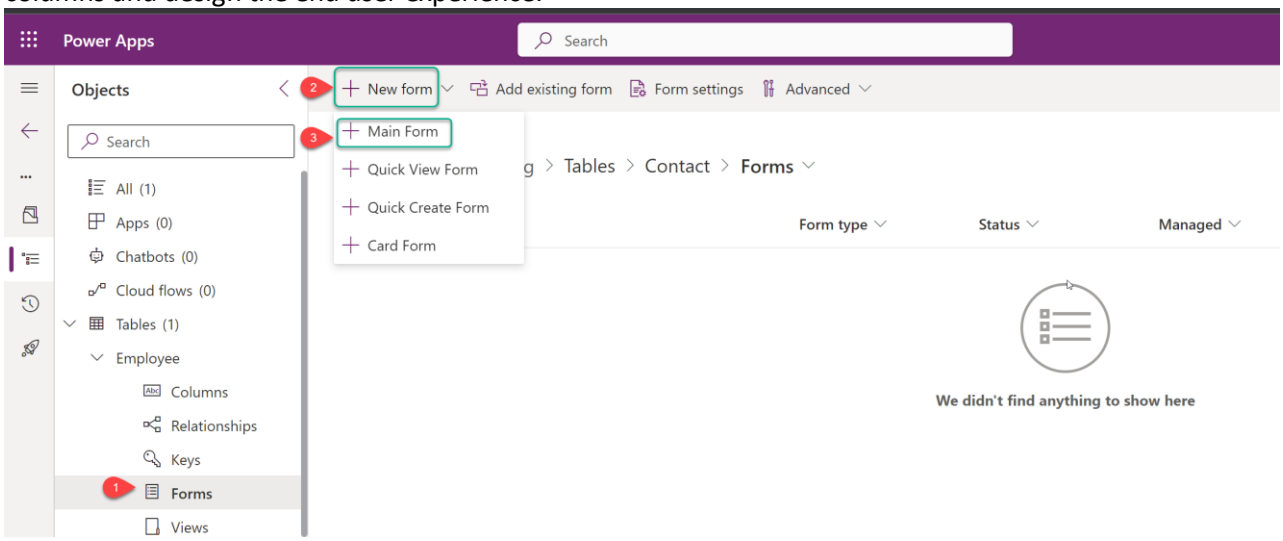
Employee Main Form

For now, we will use the existing columns and design the Form Experience for an end user who would want to add new internal employees into the interview panel.

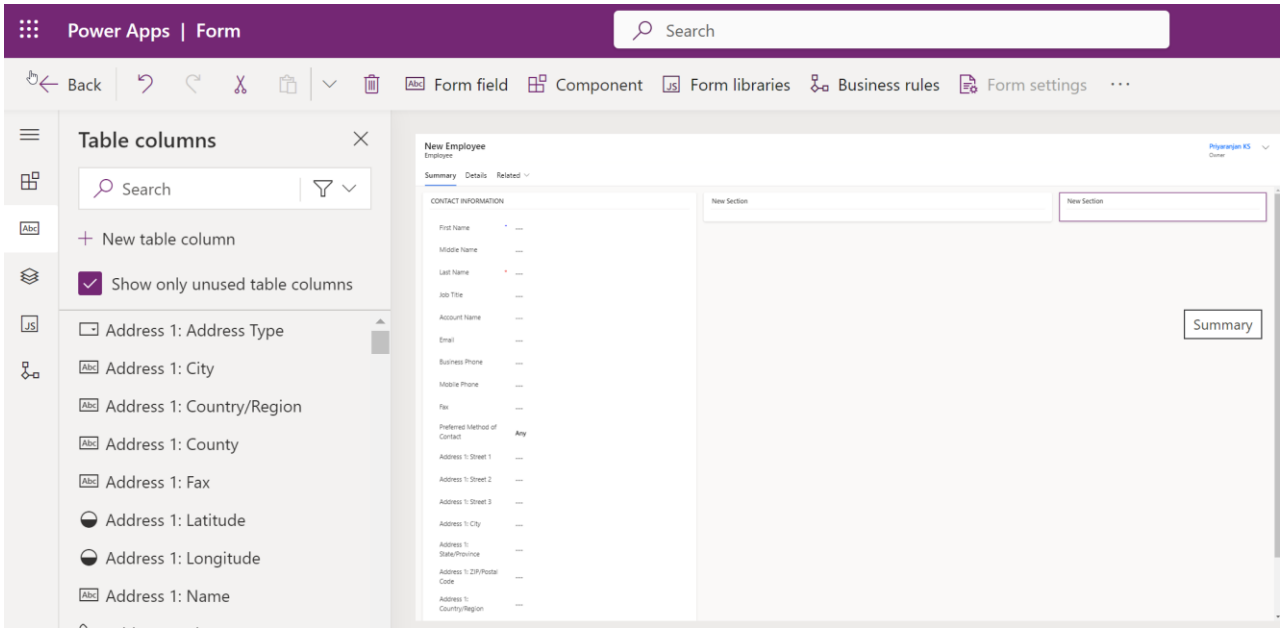
To do this, click on *Forms*.



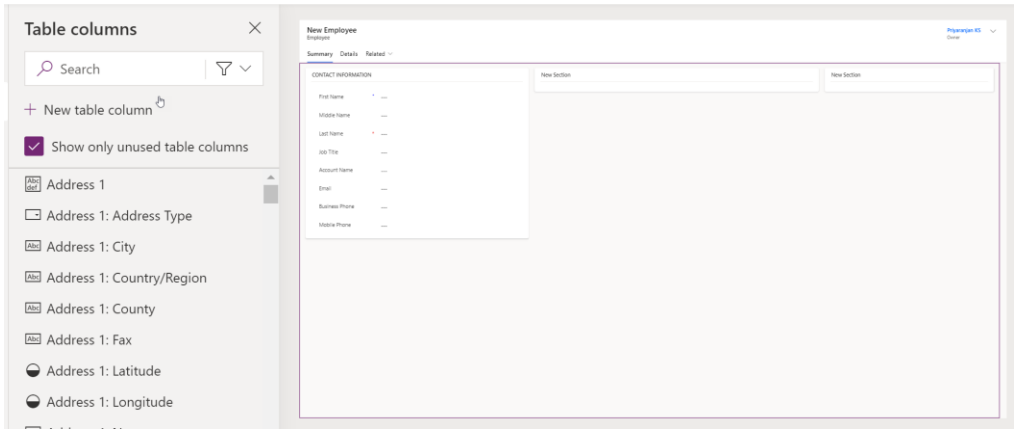
Select New form and click on Main Form to open up the form designer where we can drag and drop the columns and design the end user experience.



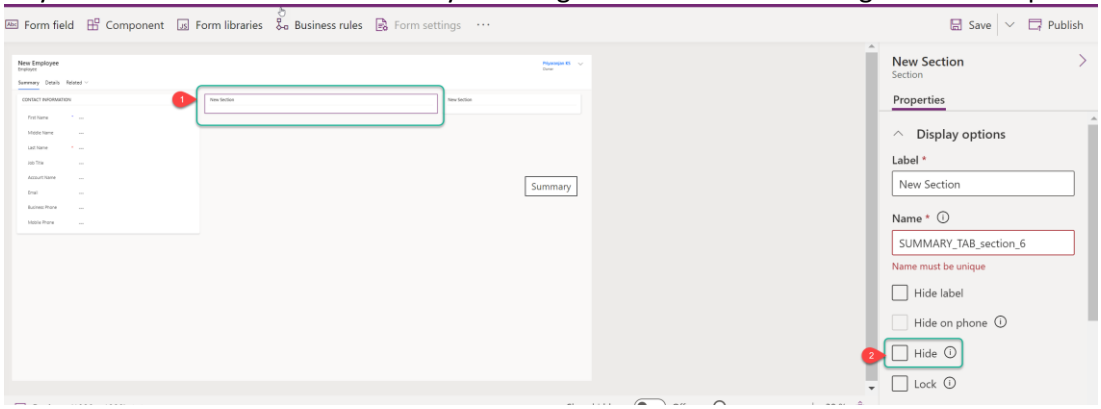
By default the form will have few sections and columns added to it as below :



Let's remove the unused columns and retain only the needed columns. We can select the unused columns and click on delete to remove it from the form.



Any unused sections can be hidden by selecting the section and selecting the "Hide" option.



Thus, we have completed the needed customizations for the employee table. Head back to the solutions by using the back button.

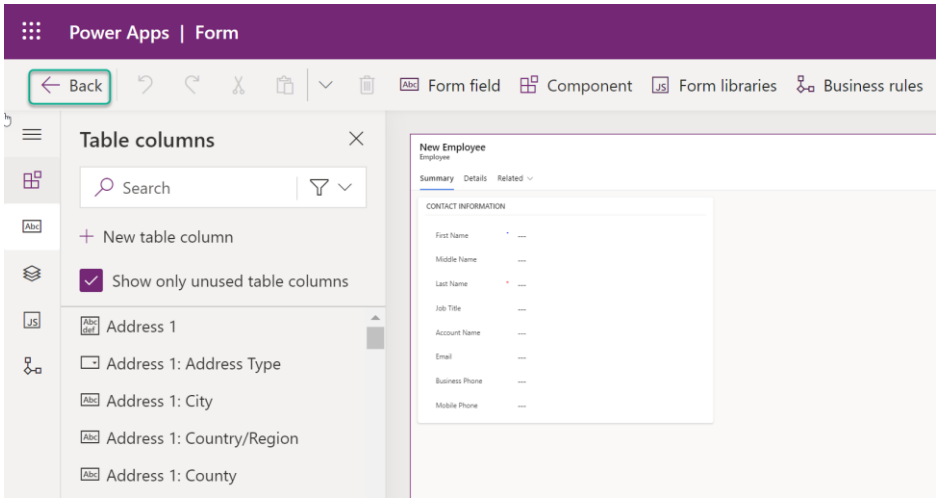
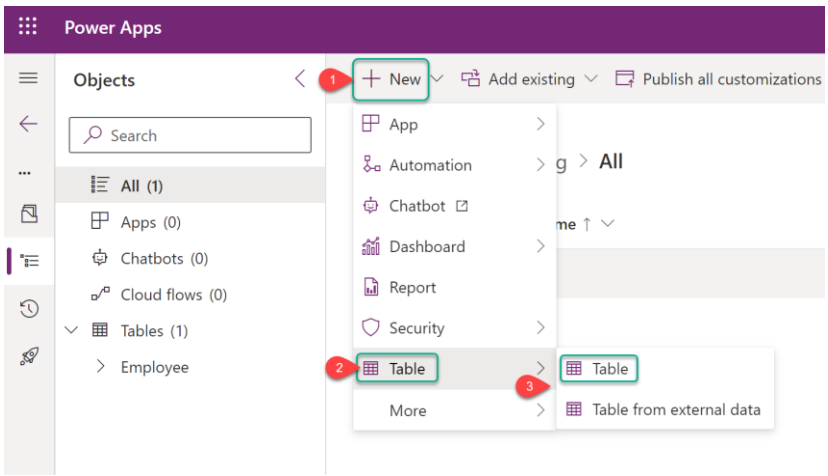


Table 2: Candidates

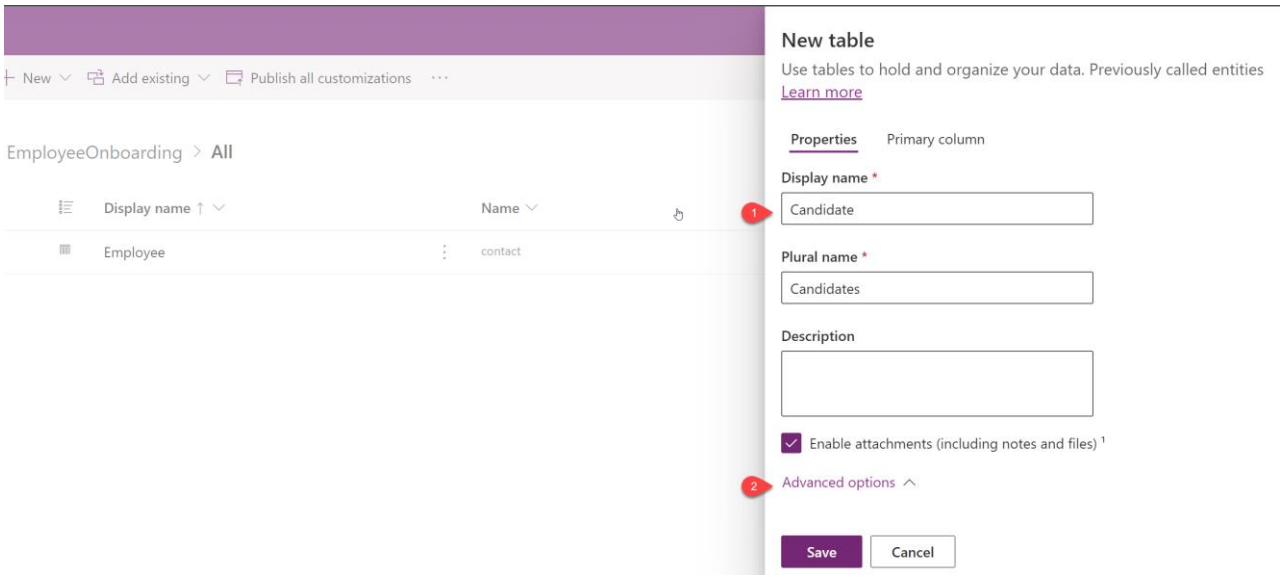
The candidates table will be used to store the details of the external candidates who are applying for a job position. To store the details, we will create a table from scratch and add custom columns to it.

Create Candidate Table

To add a new table, head over to the solutions and select *New -> Table ->Table.*

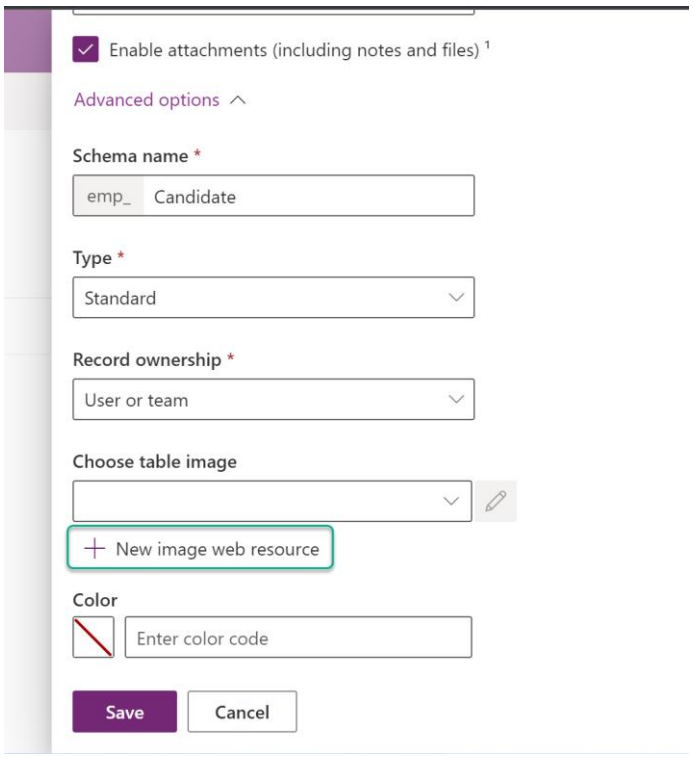


Specify the Display name as “Candidate” which will auto populate the Plural name as “Candidates” . Click on Advanced options to add more features to the table as needed.

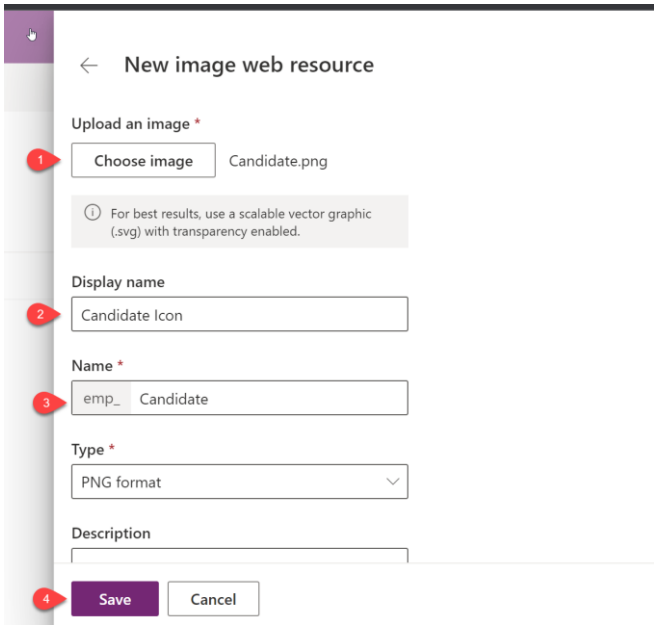


Add Table Image

We can add an icon to the table by selecting an image from the existing gallery of icons or by uploading an image icon. Select “New image web resource” to add a new custom image.

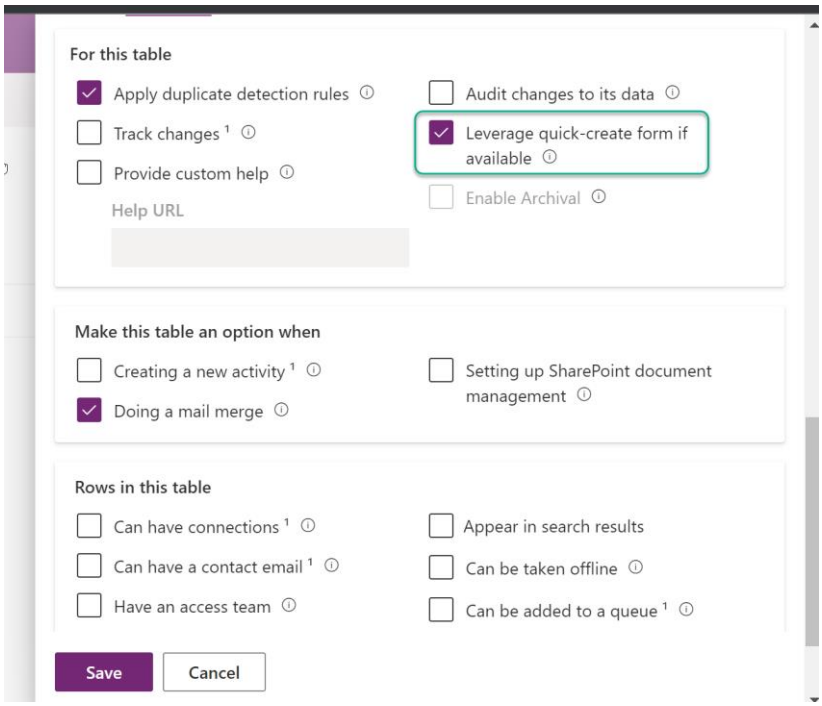


Upload the image by clicking on “Choose image” and selecting the image from the local file system. Mention the Display Name and Internal Names to be used for the image. Click on Save.

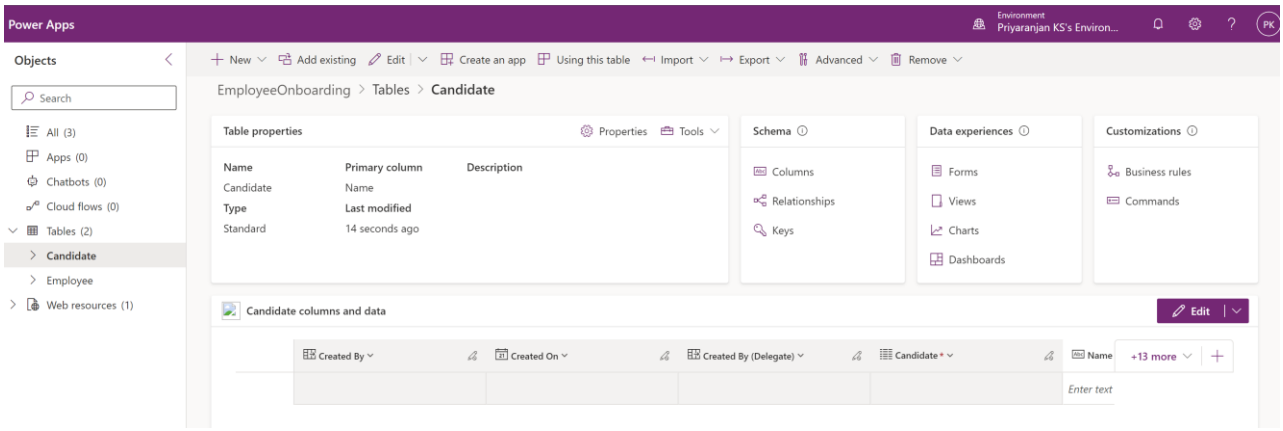


This will add the icon for the table. Now we can add some configurations specific to the table available for selection as below. For the time being, select “Leverage quick-create form if available” and click on Save.

The configurations marked with the superscript 1 cannot be updated once the table is created. Hence ensure that you plan the configurations well in advance before the table creation.



This will take us to the newly created table where we can make further modifications.

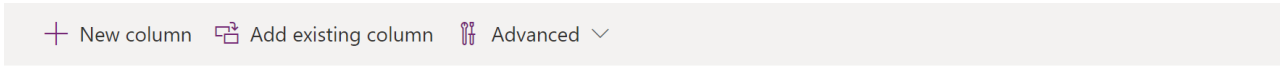


Create Columns for Candidate Table

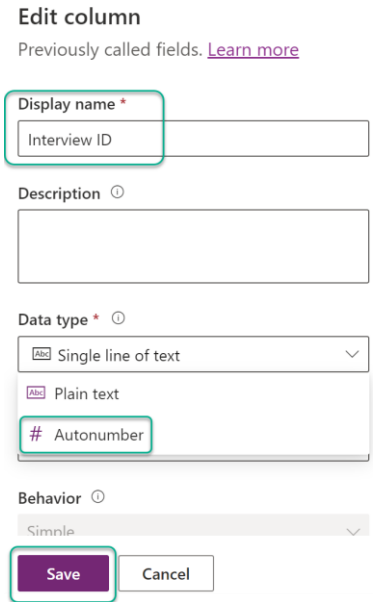
We will follow the below schema and add columns to the table.

Column Name	Data Type
First Name	Single Line of Text
Last Name	Single Line of Text
Email	Single Line of Text
Address	
Technical Expertise	
Experience	
Profile Picture	
Selected ?	

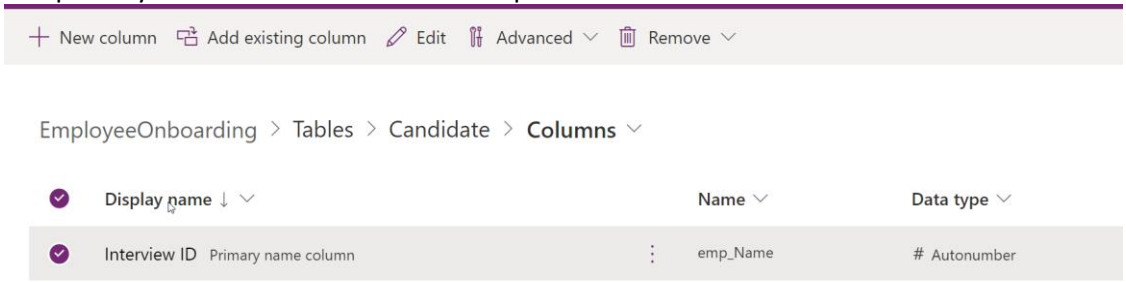
Before creating the custom columns, lets edit an existing column - “Name” which is the Primary Name column of the table.



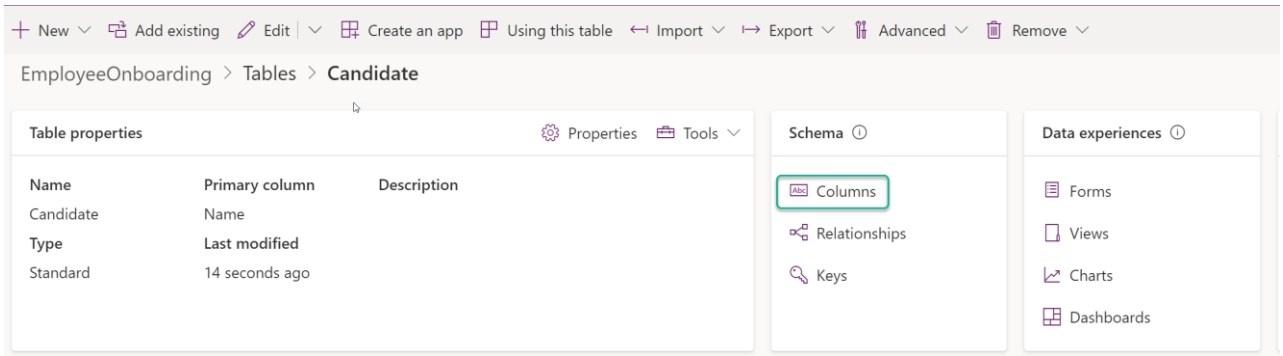
Click on Name to rename or change the configuration of the column. By default, it is of data type Single line of text and System required. We will rename it to *Interview ID* and change the data type to auto number so that when ever the record is created, the job id will be incremented automatically. We will also make the Required value to optional. Click on Save once done.



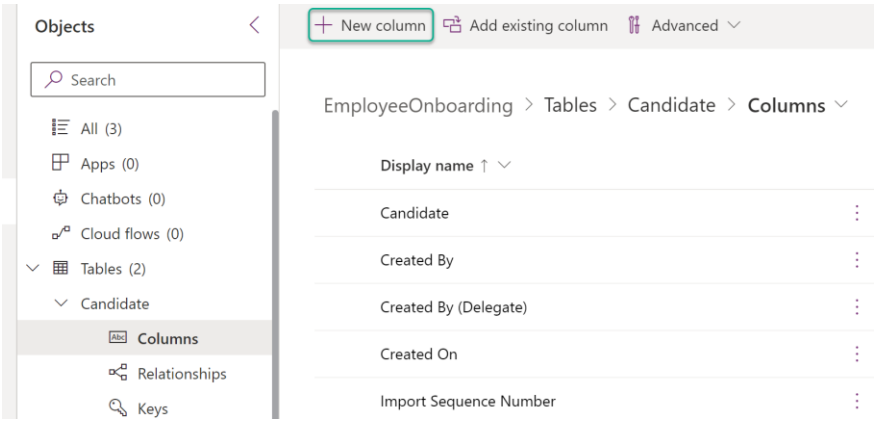
The primary name column has now been updated.



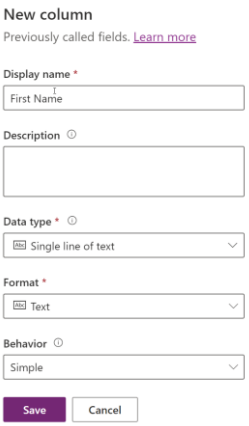
Now let's add the custom columns. Let's add the column *First Name* by selecting the Columns option.



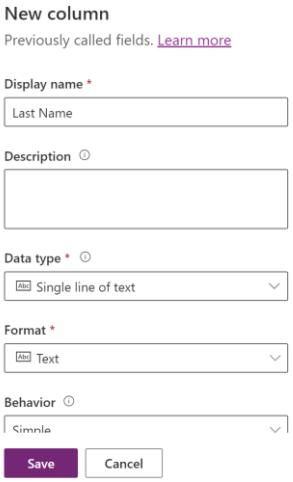
Click on New Column



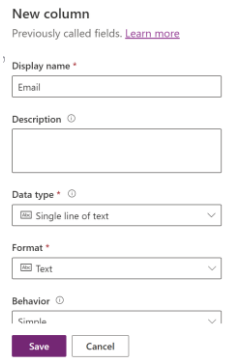
- Specify the Display Name as *First Name* and Select the Data Type as *Single Line of Text* and click on Save.



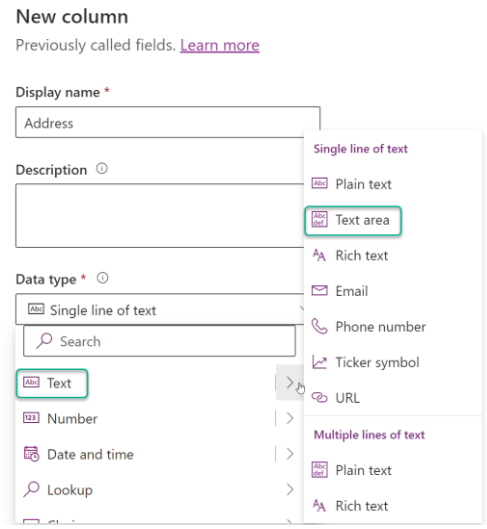
- Click on New Column and Specify the Display Name as *Last Name* and select the data type as *Single Line of text*



- Click on New Column and Specify the Display Name as *Email* and select the data type as *Single Line of text*

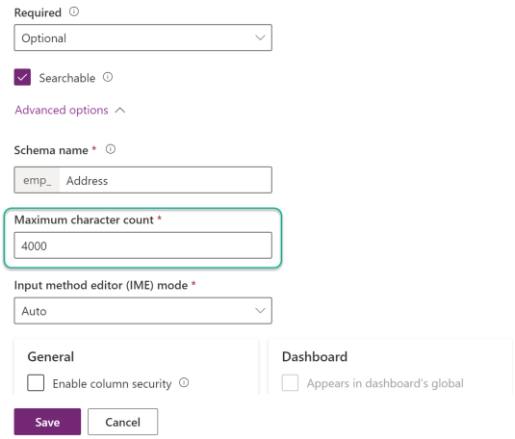


- Click on New Column and Specify the Display Name as *Email* and select the data type as *Text ->Text area*

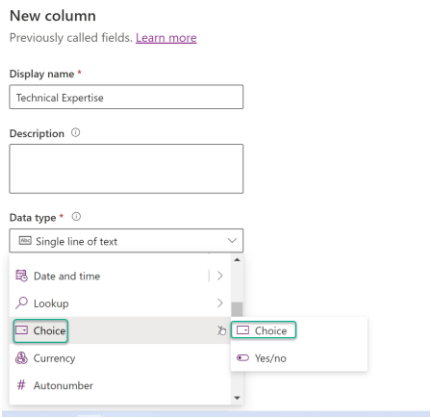


To ensure that there is ample space in the text area, head over to the *Advanced options* and type in 4000 as the maximum character count. Click on Save.

Note : You can enter a value of at the most as 4000.

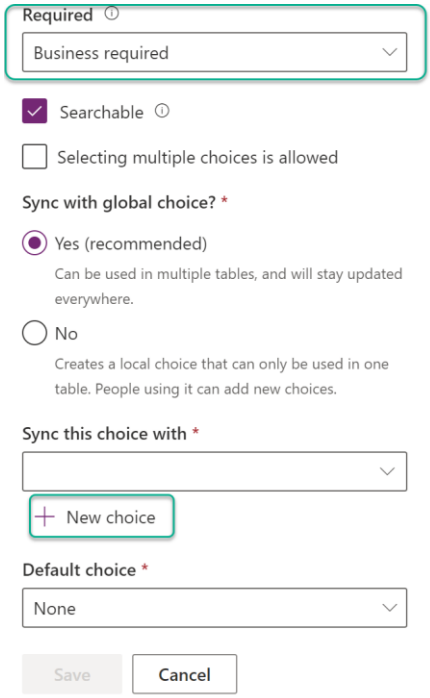


- Click on New Column and Specify the Display Name as *Technical Expertise* and select the data type as *Choice ->Choice*

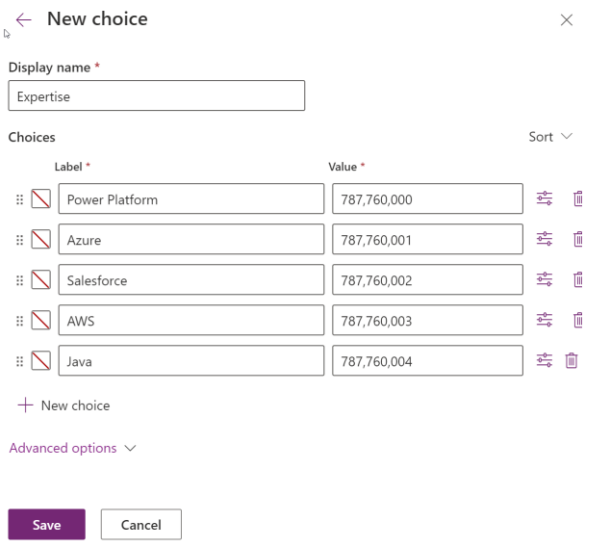


Let's make the column as mandatory by selecting the *Required* value as *Business Required* . To populate choice field, we can either create a local option set or create a globally usable option set so that it can reused and synchronized across multiple choice fields.

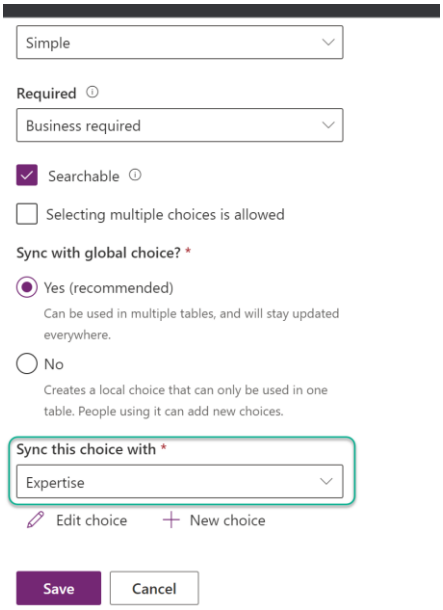
Here we will create a global choice set. Select *Yes* for "*Sync with global choice?"* and click on "*New choice*" to add the choice options.



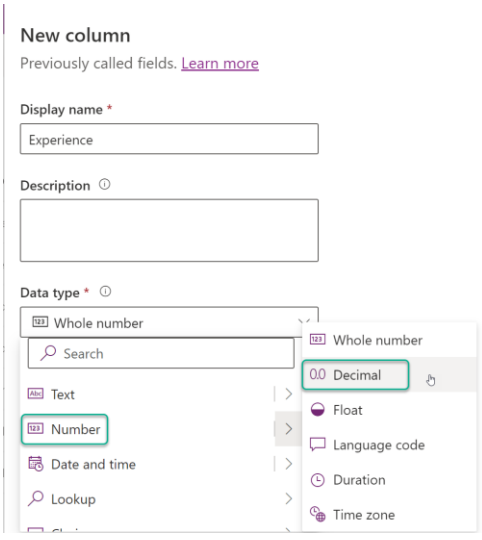
Specify the global choice field name as "Expertise" and add the choice values as below . Click on Save.



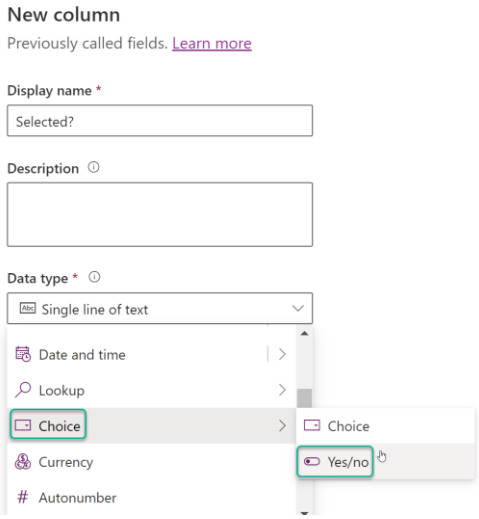
Select the recently created global choice in the field : “Sync this choice with” and click on Save.



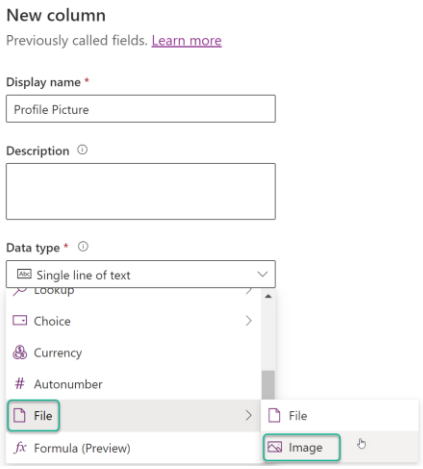
- Click on New Column and Specify the Display Name as *Experience* and select the data type *Number -> Decimal* . Click on Save.



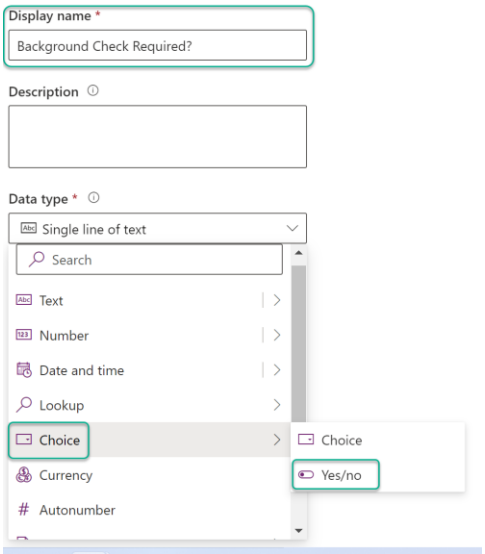
- Click on New Column and Specify the Display Name as *Selected?* and select the data type *Choice -> Yes/no* .Click on Save.



- Click on New Column and Specify the Display Name as *Profile Picture* and select the data type *as File-> Image* .Click on Save.

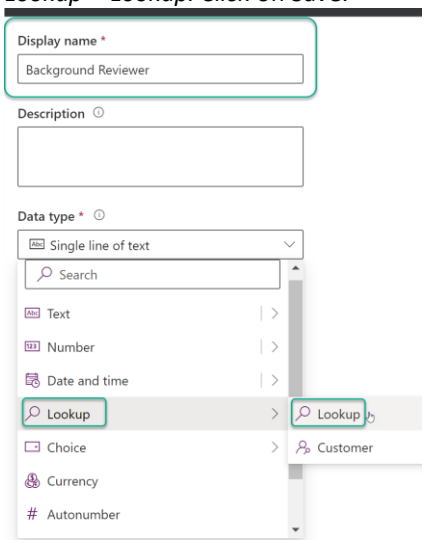


- Click on New Column and Specify the Display Name as *Background Check Required ?* and select the data type *as Choice-> Yes/No* .Click on Save.

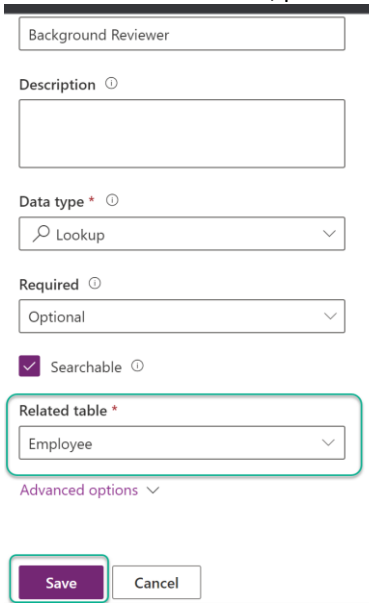


- Click on New Column and Specify the Display Name as *Background Reviewer?* and select the data type *as*

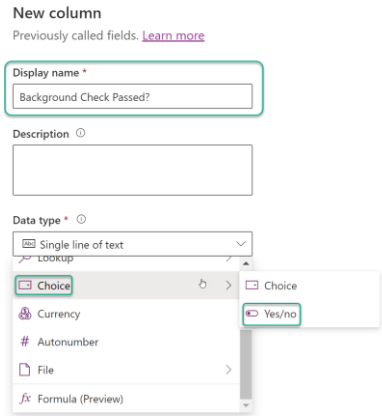
Lookup-> Lookup. Click on Save.



In the *Related table* field, point the look up to the *Employee* table and click on Save.

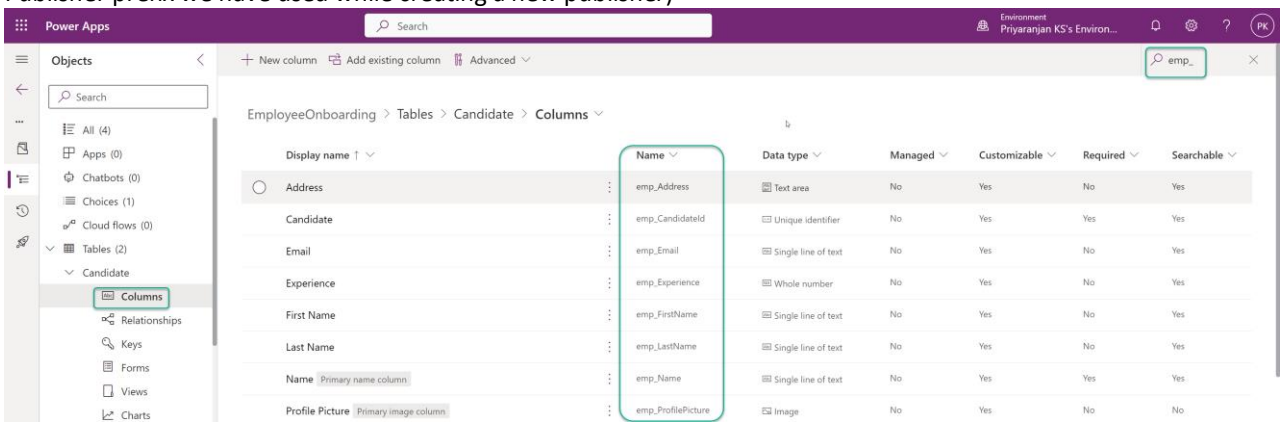


- Click on New Column and Specify the Display Name as *Background Check Passed ?* and select the data type *as Choice-> Yes/No* .Click on Save.



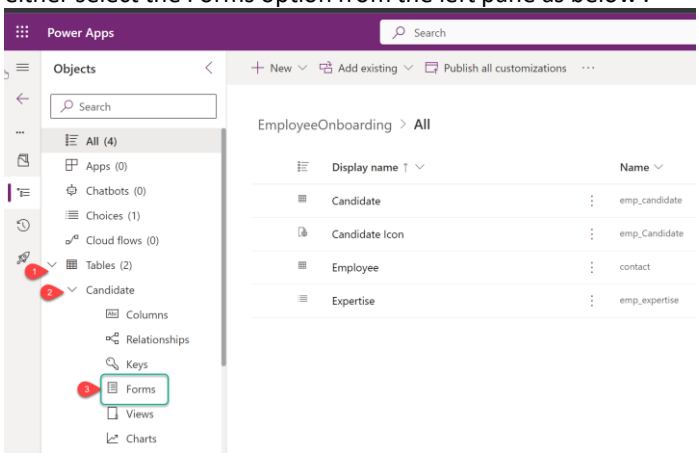
Thus, we have completed the addition of the needed columns for the Candidate Table. To see the custom columns

we have added to the table, we can head over to the solution's left pane and select the Candidate table from the Tables section. Click on Columns and type in "emp_" in the search box to filter the custom columns. (Emp is the Publisher prefix we have used while creating a new publisher)

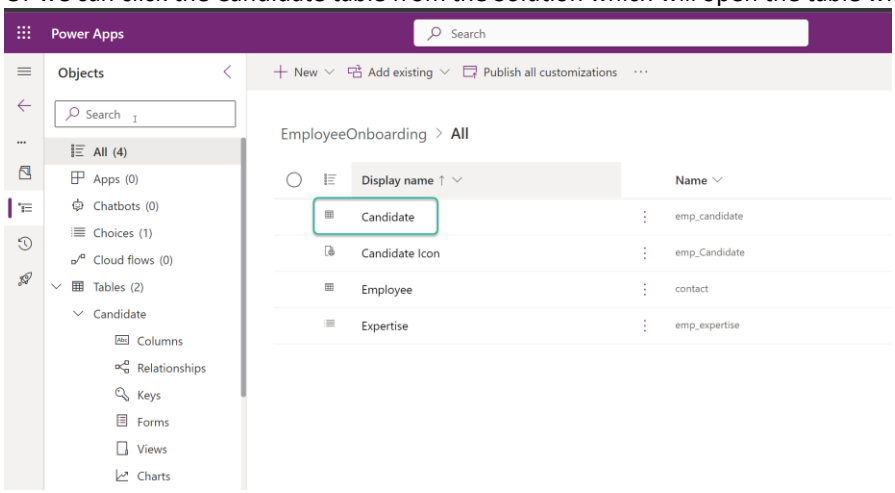


Create Main Form for Candidate Table

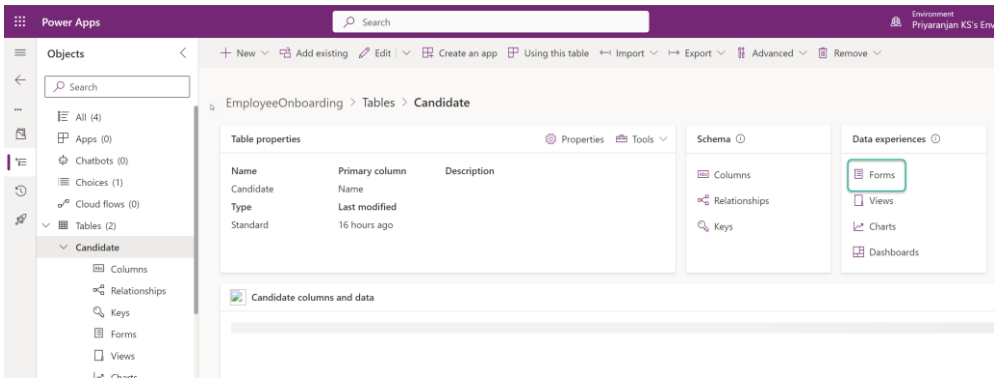
Now let's create the User experience for using the table by customizing the main form. To access the Forms, we can either select the Forms option from the left pane as below :



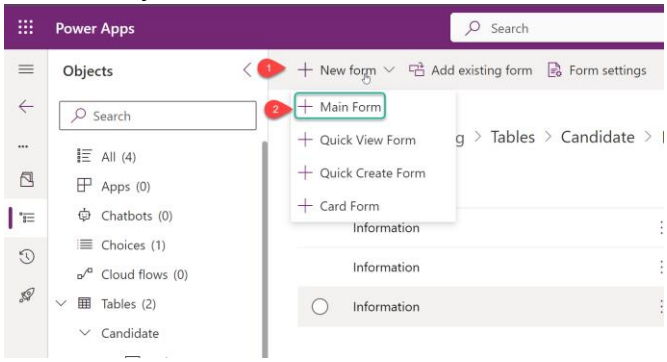
Or we can click the Candidate table from the Solution which will open the table with numerous options to work with.



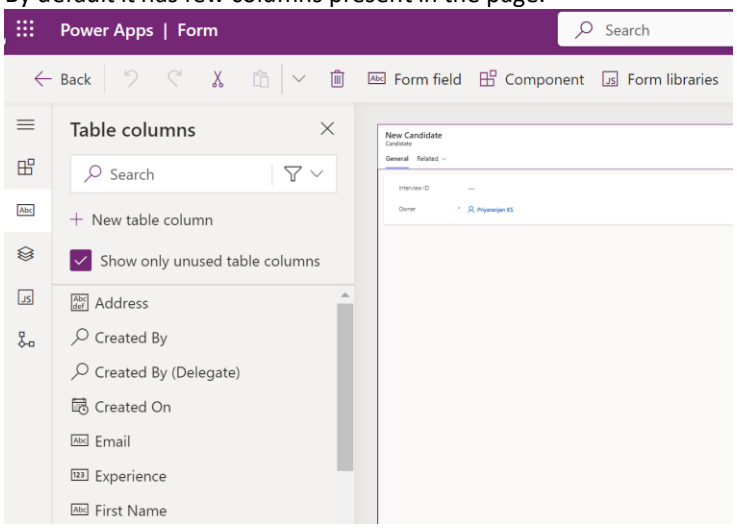
From within the table, select Forms.



This will open the page where we can see some of the existing forms that can be utilized for the candidate table. We will create a new Main form which is the primary data update UI for the table. So, let's go ahead and select *New Form->Main form.*

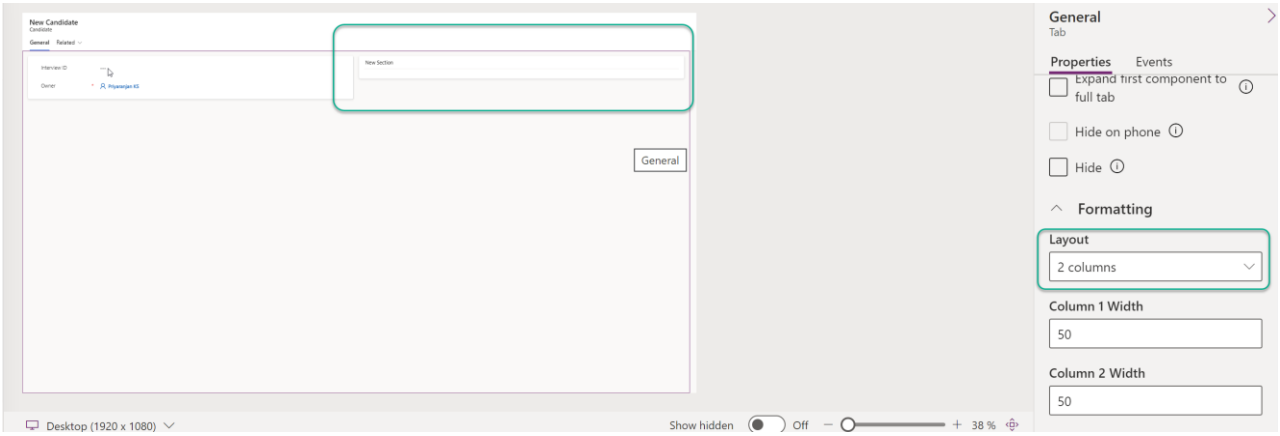


By default it has few columns present in the page.

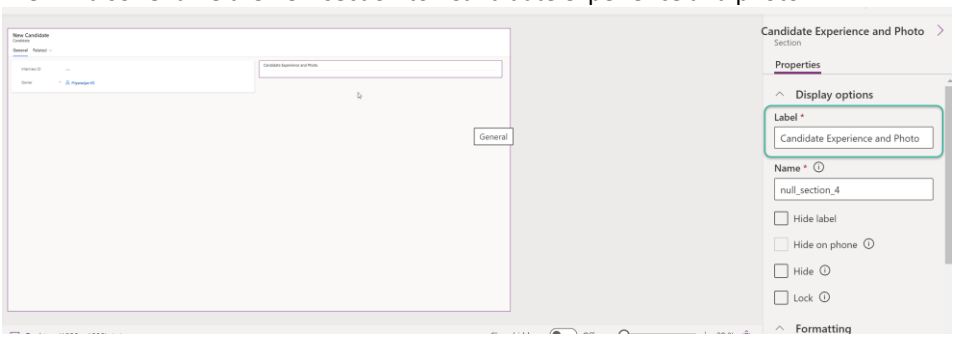


Before adding the remaining columns, lets divide the form into 3 sections so that we can divide the data present in the form according to their category.

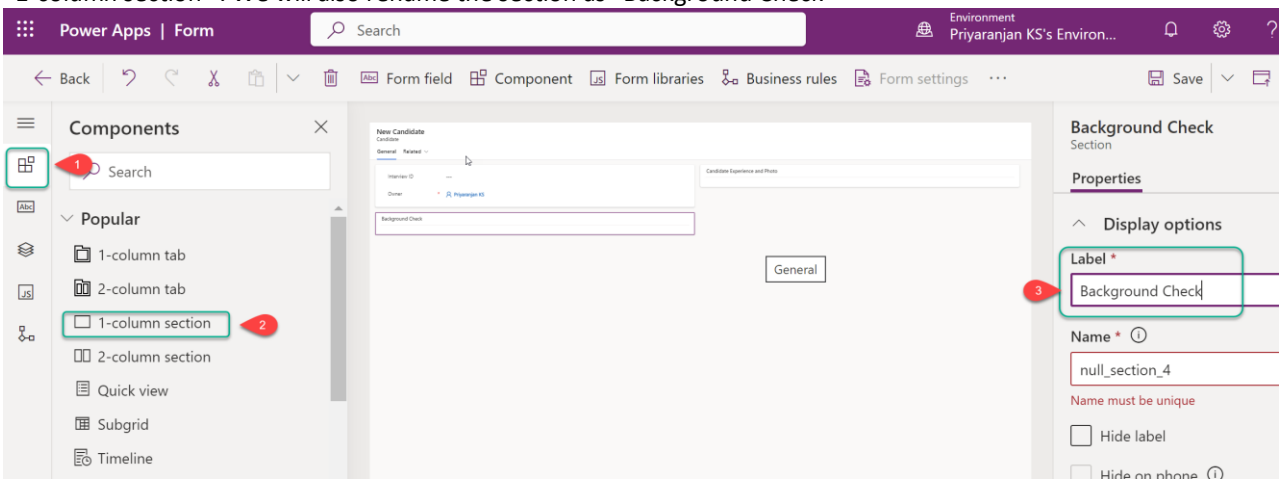
By default the form has a 1 column layout, we will divide it into a left and right section by changing it to 2 columns which will add the right section.



We will also rename the new section to “Candidate experience and photo”

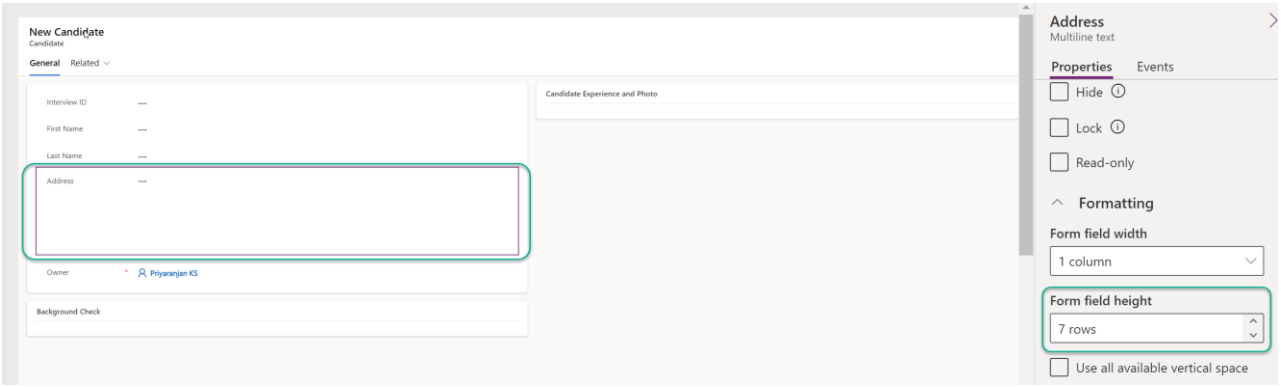


We will also add a section to the bottom for which we can click on the components section in the left pane and select “1-column section” . We will also rename the section as “Background Check”



Now let’s drag and drop the remaining columns that we have created recently onto the form to various sections as below.

After adding First Name and Last Name, we have added Address. To ensure that there is enough space to add the data, increment the form field height to 7 rows.



Lets drag and drop the remaining columns to the remaining sections and the final form design will look as below :

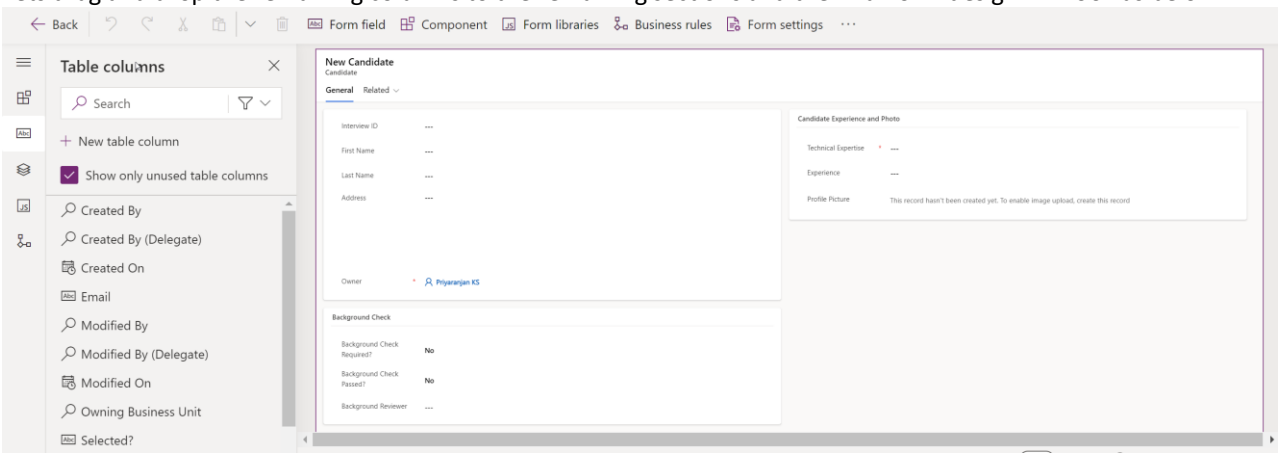
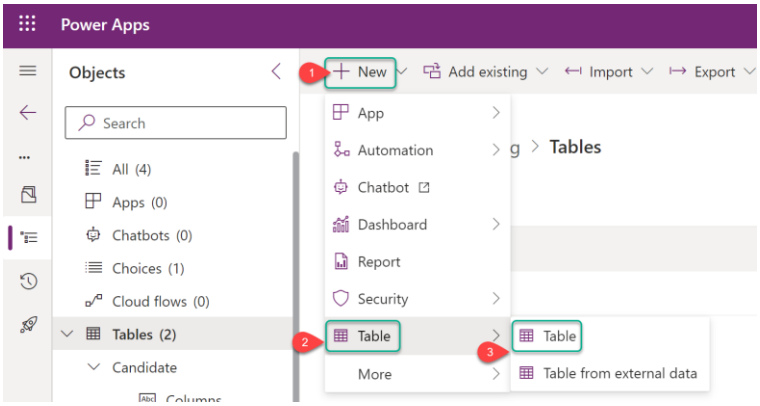


Table 3 : Job Entry

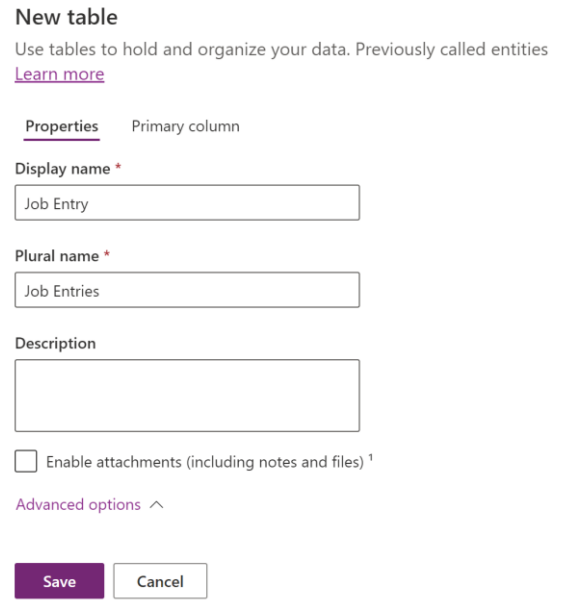
The Job entry table will contain the records related to the Job opening and the candidate interviewed for the job opening. For each new candidate interviewed, we will create a new record with the Job description and other details.

Create Job Entry Table

Head over to the solution and Select New -> Table -> Table.



Specify the table name as *Job Entry* and click on Save.

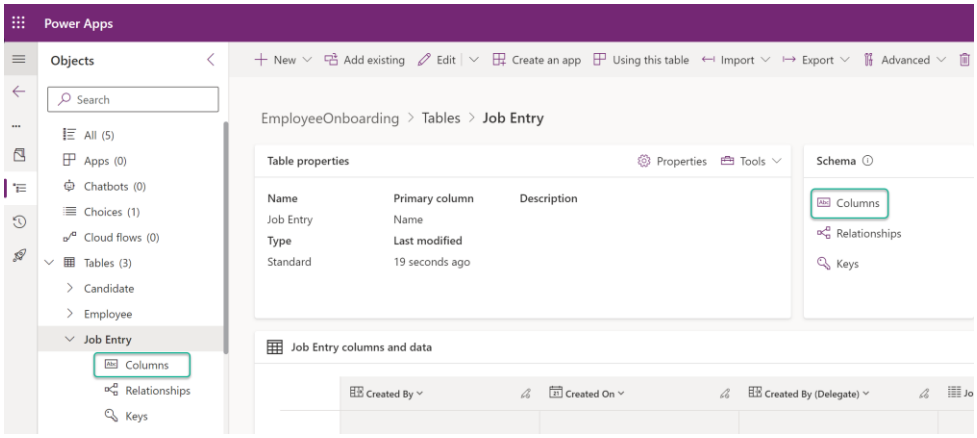


Create Columns for Job Entry Table

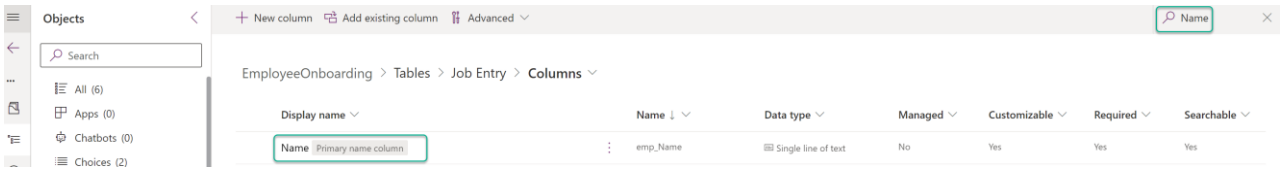
The column schema used for the Job Entry Table is as follows:

Column Name	Data Type
Technology	Choice
Minimum Years of Experience	Number
Department	Choice
Manager	Lookup
Job Description	Text
Candidate Name	Text
Interview Date	Number
Mode of Interview	Choice
Interview Panel	Lookup

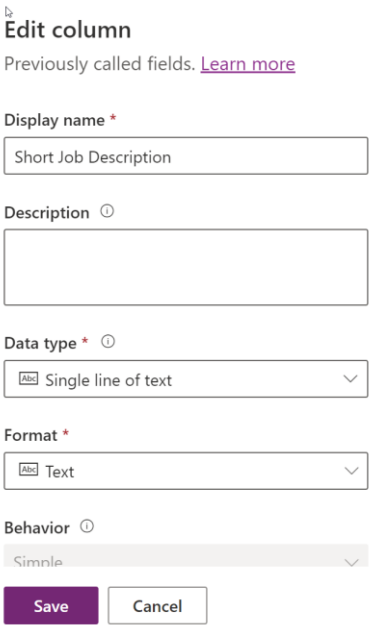
Before adding custom columns, lets modify the existing primary name column by selecting the *Columns* section from either the Solutions left pane or by going inside the Job Entry table.



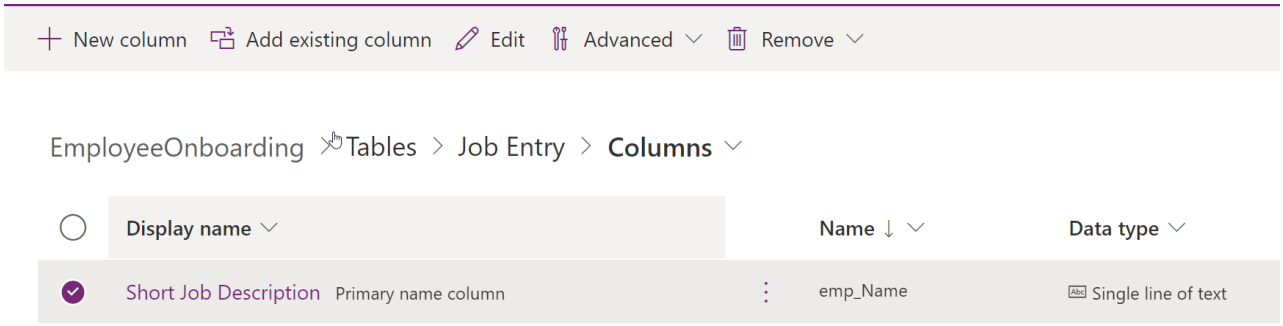
If we search by the title *Name* , we can see the primary name column listed as below



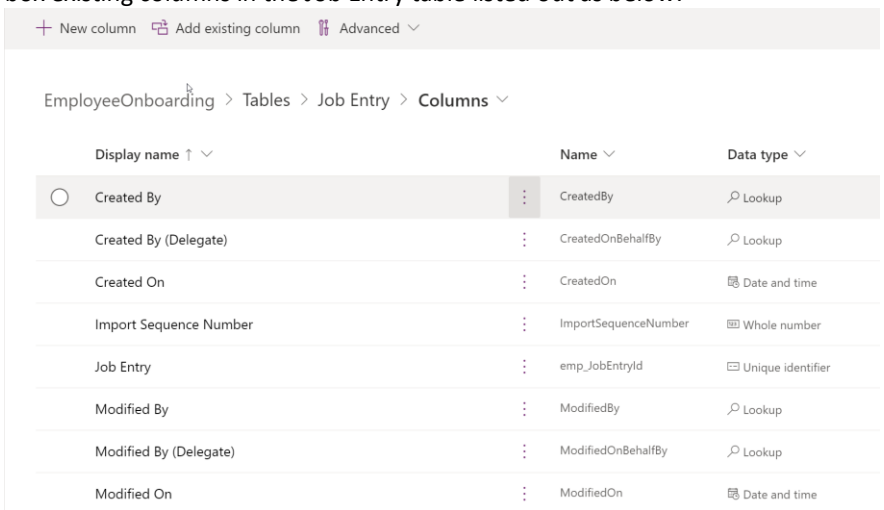
We will click on it and edit the column to rename it to *Short Job Description.* Click on Save



This will rename the existing primary name column as below:

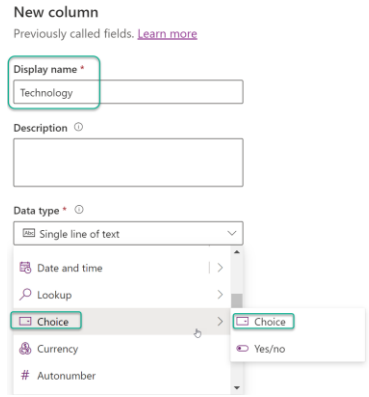


Now let's add the custom columns by heading back to the Columns section of the table. We can see the out of the box existing columns in the Job Entry table listed out as below.

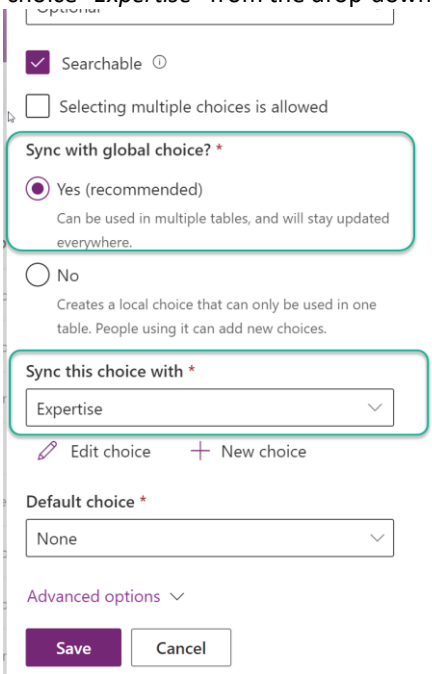


- Click on New Column and Specify the Display Name as *Technology?* and select the data type *as Choice->*

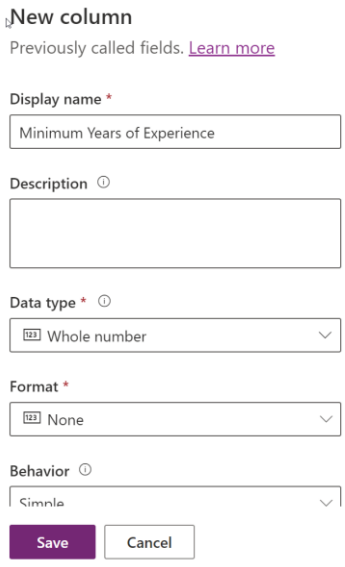
Choice .



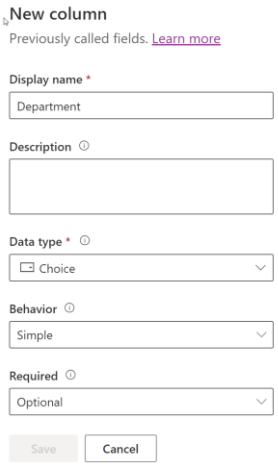
So as to make the Choice reusable in the solution, lets make it a global choice by selecting the option "Sync with global choice?" . Select the previously created(while creating the candidates table columns) global choice "*Expertise*" from the drop-down menu *Sync this choice with* .Click on Save.



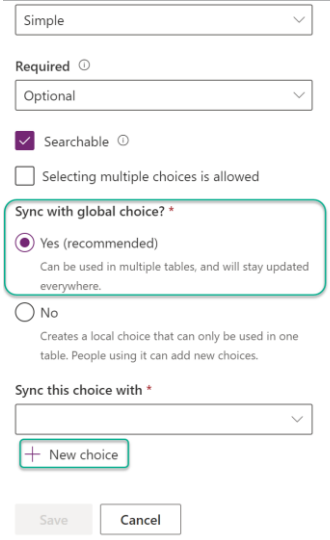
- Click on New Column and Specify the Display Name as *Minimum Years of Experience* and select the data type *as Whole number* .



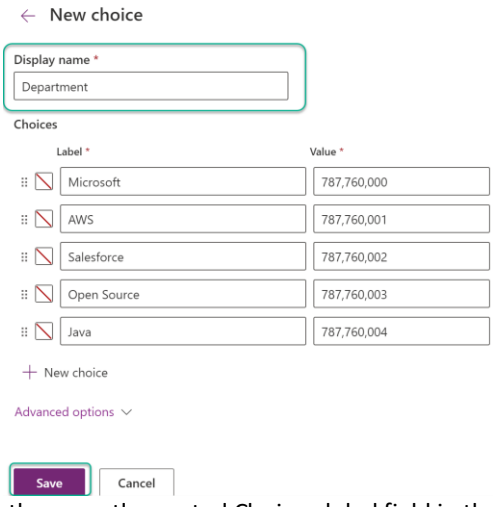
- Click on New Column and Specify the Display Name as *Department* and select the data type *as Choice -> Choice.*



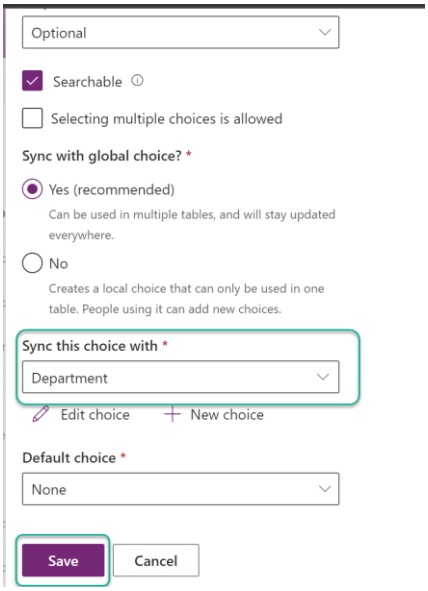
So as to populate the choice values, ensure that we make it a global choice by selecting the "Sync with global choice?" as Yes . Click on New choice to populate the global choice field.



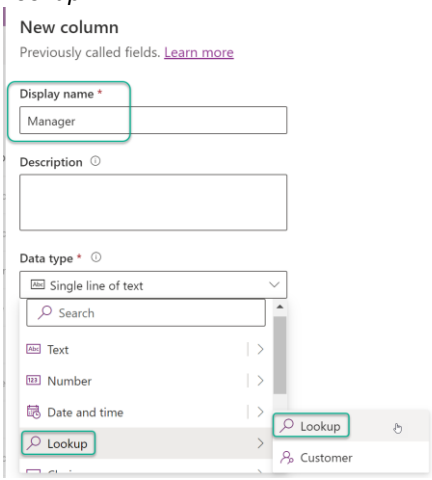
Specify the choice field name as "Department "and enter the field values. Click on Save.



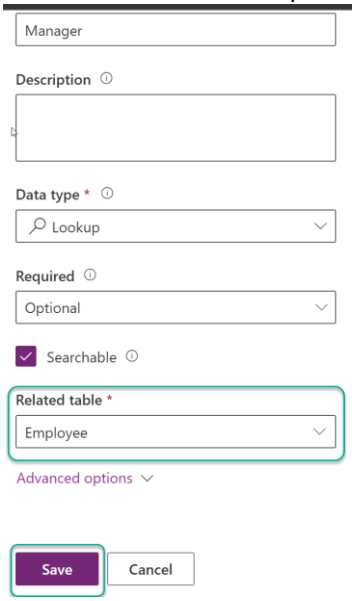
Specify the recently created Choice global field in the "*Sync this choice with*" drop down and click on Save.



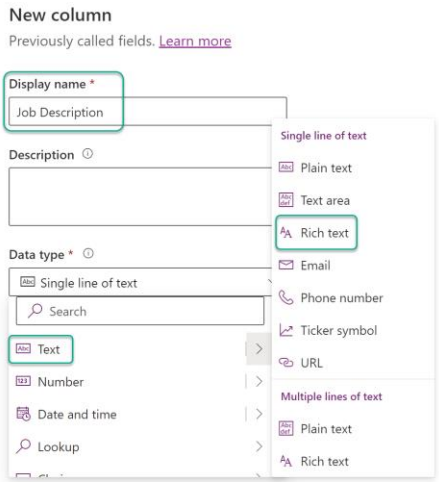
- Click on New Column and Specify the Display Name as *Manager* and select the data type *as Lookup -> Lookup.*



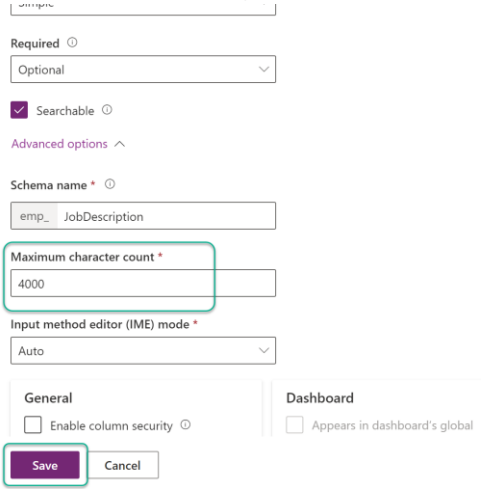
Mention the related lookup table as *Employee* and click on Save.



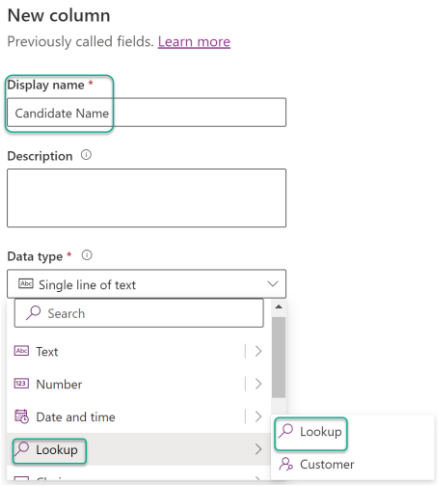
- Click on New Column and Specify the Display Name as *Job Description* and select the data type *as Text -> Rich Text.*



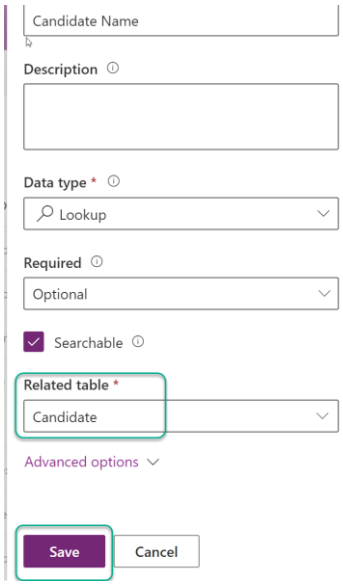
In the advanced settings, ensure to increment the *Maximum character count* to *4000* and click on Save.



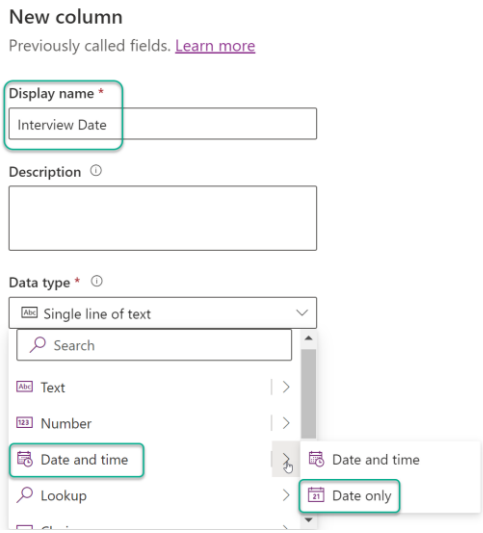
- Click on New Column and Specify the Display Name as *Candidate Name* and select the data type *as Lookup -> Lookup*



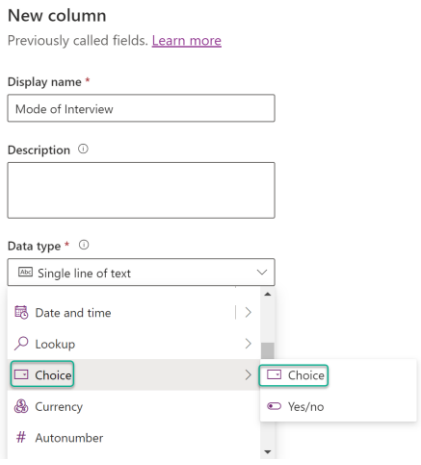
Select the lookup table as "*Candidate*" from the drop-down menu and click on Save.



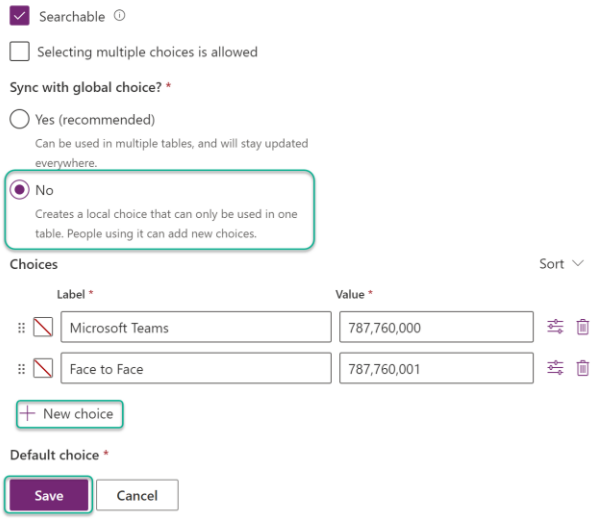
- Click on New Column and Specify the Display Name as *Interview Date* and select the data type *as Date and time -> Date only* . Click on Save.



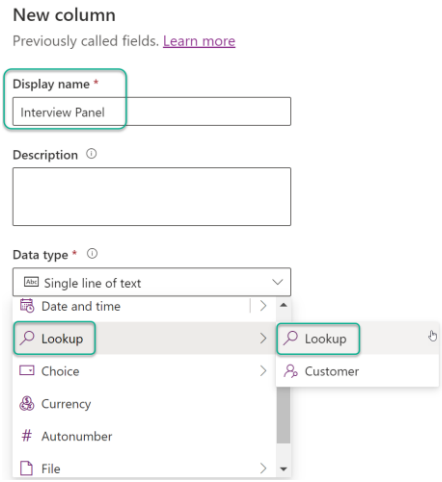
- Click on New Column and Specify the Display Name as *Mode of Interview* and select the data type *as Choice -> Choice* .



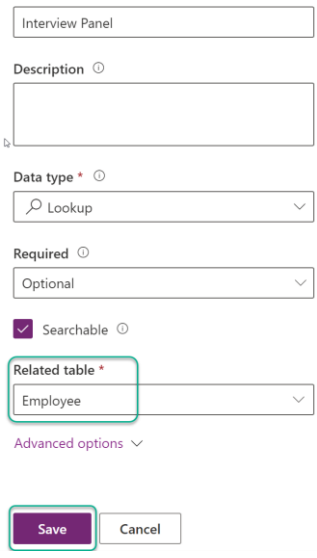
To populate the choice field, instead of global choice values, we will create a localized field value set by selecting *No* for the “Sync with global choice ?” option and enter the choice values by clicking on *New choice* . We have added 2 choices *Microsoft Teams* and *Face to Face*. Click on Save.



- Click on New Column and Specify the Display Name as *Interview Panel* and select the data type *as Lookup -> Lookup.*



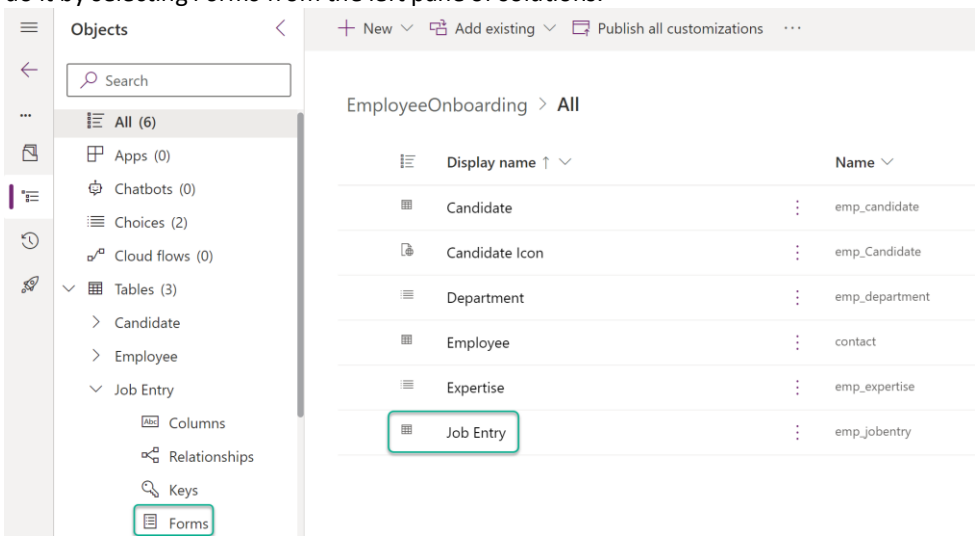
Select the look up table as *Employee* from the drop down.



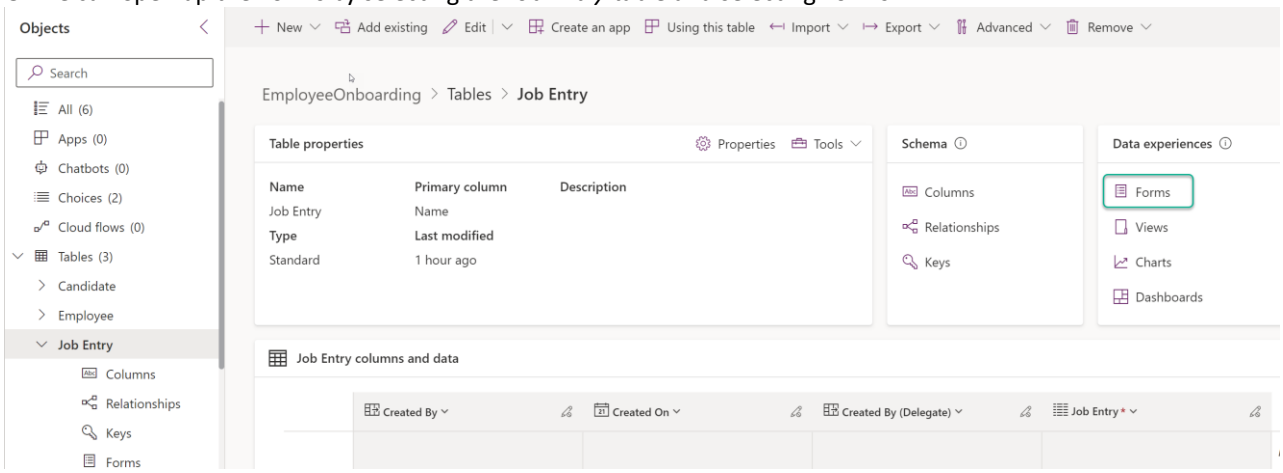
Thus, we have completed the creation of the custom columns for the Job Entry table.

Create Main Form for Job Entry Table

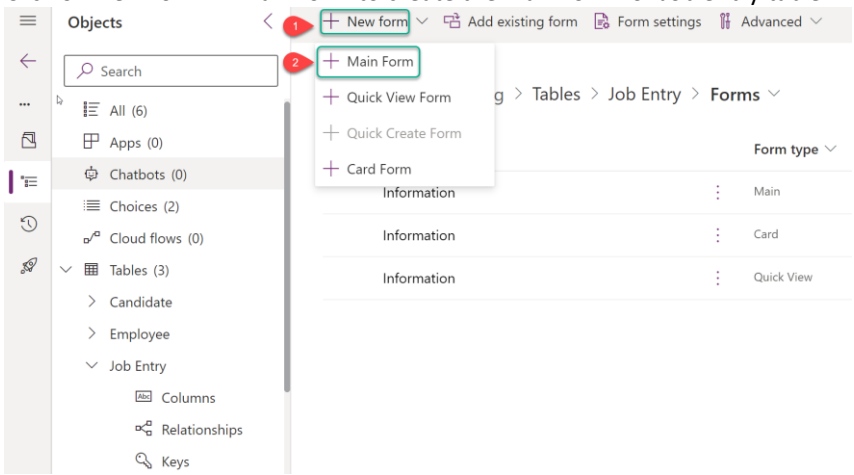
To provide a data entry UI experience for the end user, we will create a form for the Job Entry table. We can either do it by selecting *Forms* from the left pane of solutions.



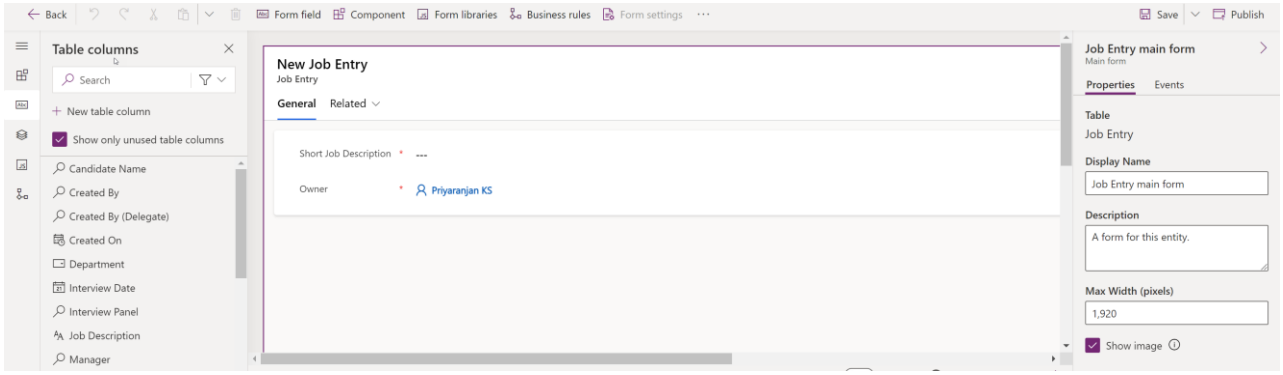
Or we can open up the Forms by selecting the *Job Entry* table and selecting Forms



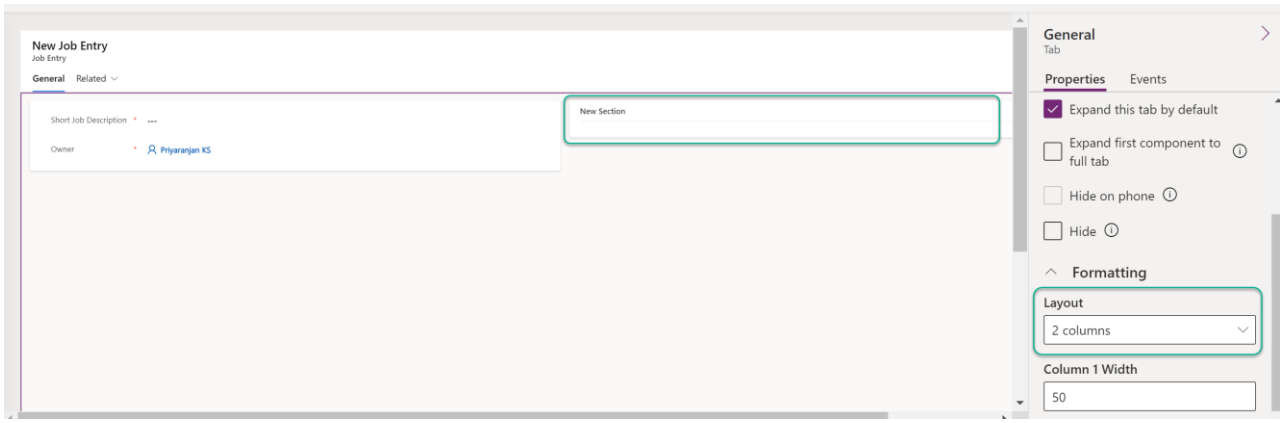
Click on New form -> Main form to create the main form for Job entry table.



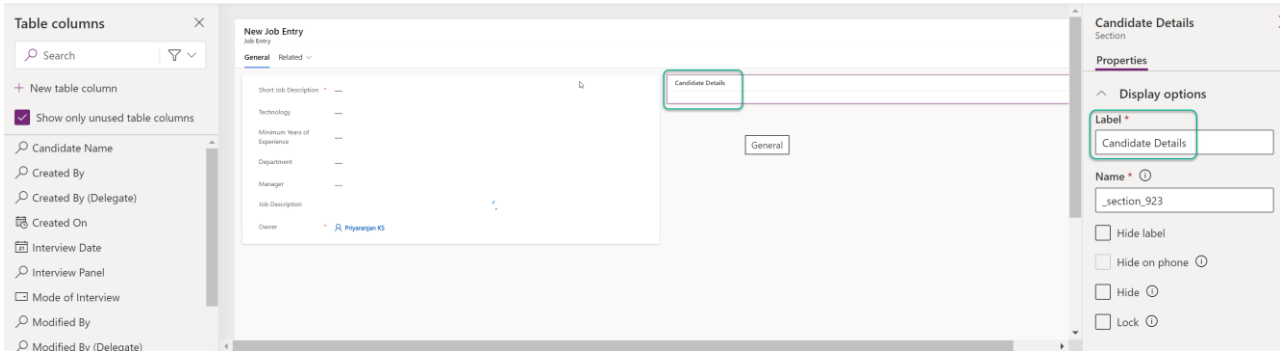
The main form will have few default columns listed out as below



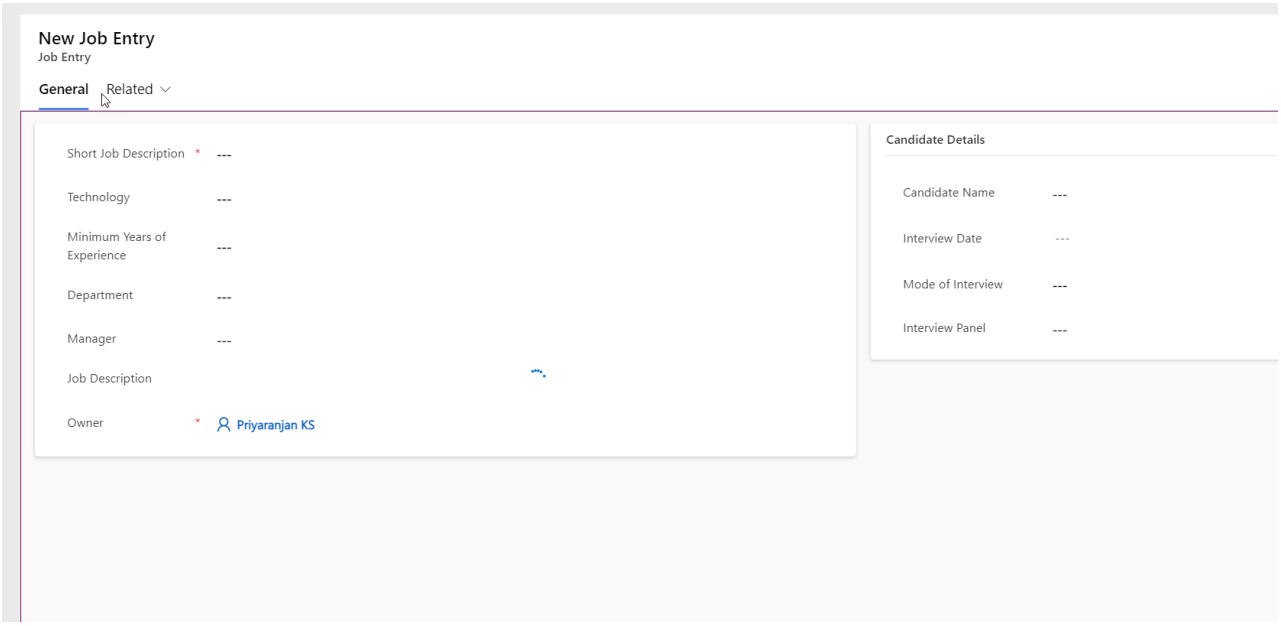
Let's create a 2 section view initially before dragging and dropping the columns to the main form. To do this, select the form and from the properties select the layout as *2 columns* which will add a new section to the right side of the form.



We have added the columns to the left as below and have renamed the right section as "Candidate Details"



Lets add the remaining columns to the Candidate Details section



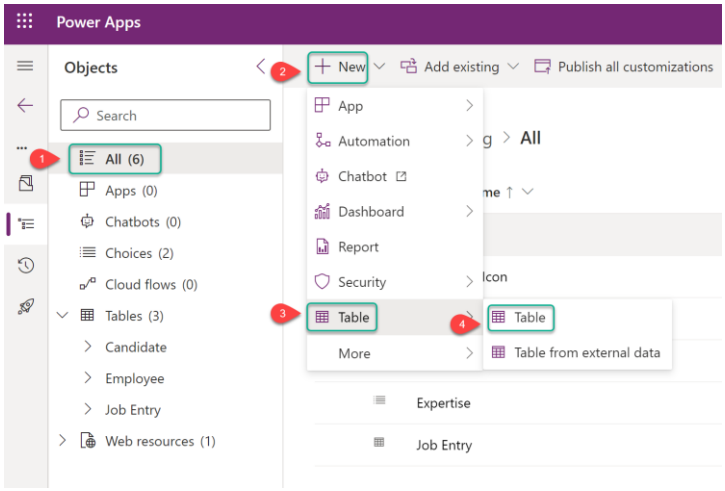
Thus, we have completed the design for the main form of the Job Entry table.

Table 4: Interview Feedback

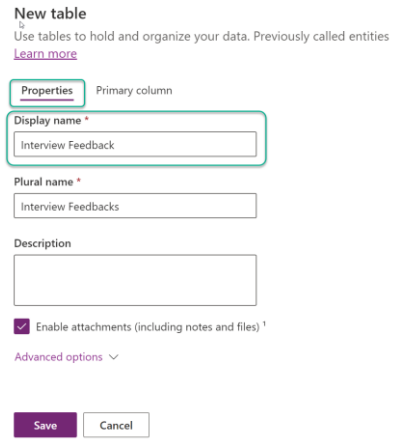
We will be creating the 4th table Interview Feedback to store the notes about the interview process of a candidate.

Create Interview Feedback Table

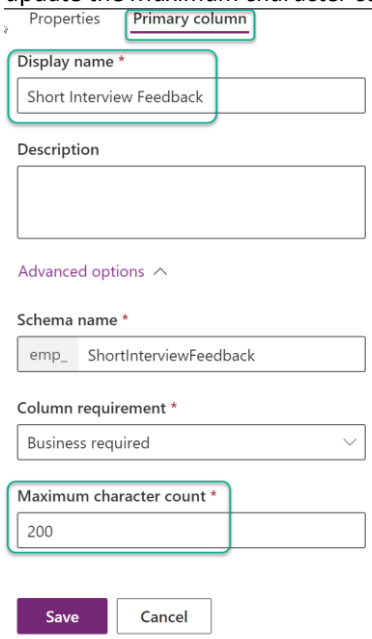
So as to create the table, head over to *All -> New -> Table -> Table*



From the properties tab, mention the Display name as *Interview Feedback*

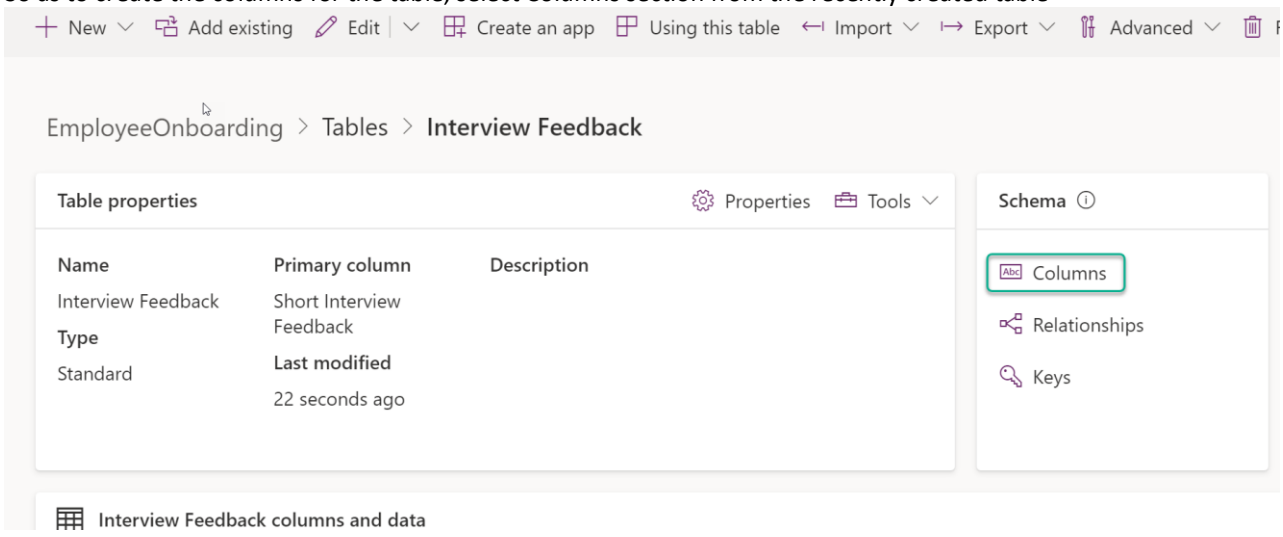


From the Primary column tab, specify the Display name for the Primary column as *Short Interview Feedback* and update the *Maximum character count* to *200*. Click on Save.

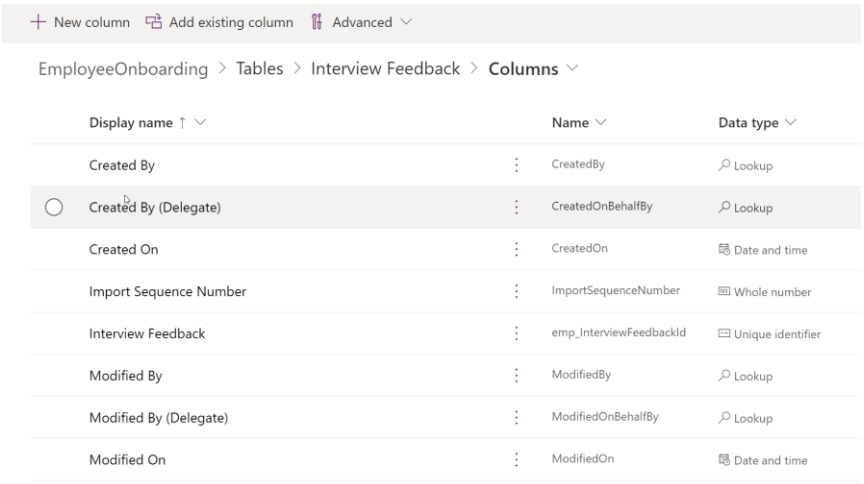


Create Columns for Interview Feedback Table

So as to create the columns for the table, select Columns section from the recently created table



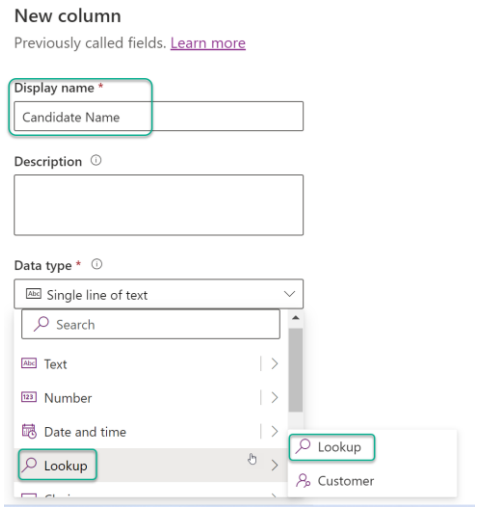
This will list the out of the box columns present in the table.



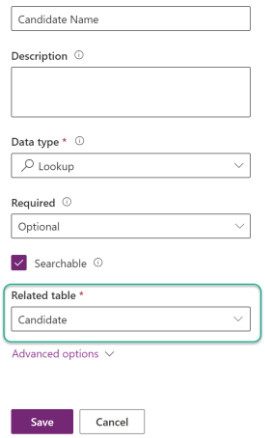
We will create the columns following the below schema.

Column Name	Data Type
Candidate Name	Lookup
Overall Interview rating	Number
Interview Detailed Notes	Text
Interview Date	Date
Interviewed By	Lookup

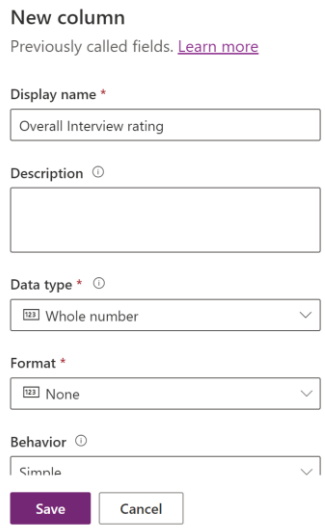
- Click on New Column and Specify the Display Name as *Candidate Name* and select the data type *as Lookup -> Lookup.*



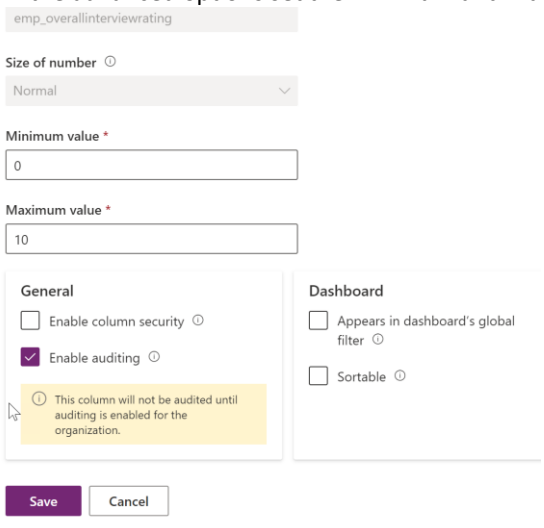
Select the lookup table as Candidate and click on Save.



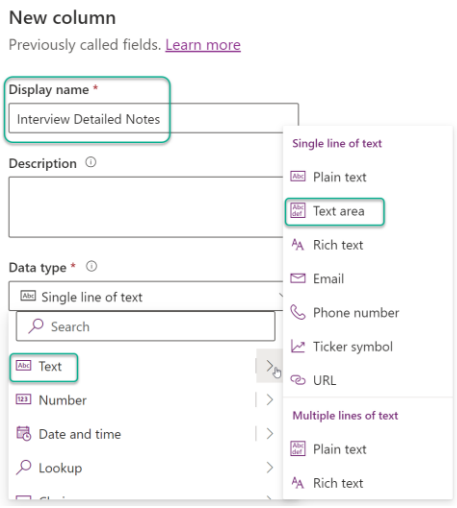
- Click on New Column and Specify the Display Name as *Overall Interview rating* and select the data type *as Number.*



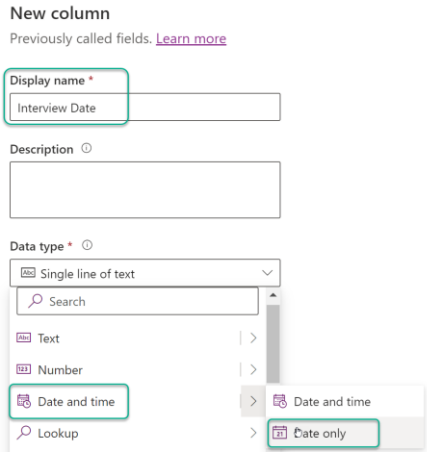
In the advanced options set the minimum and maximum values as below. Click on Save .



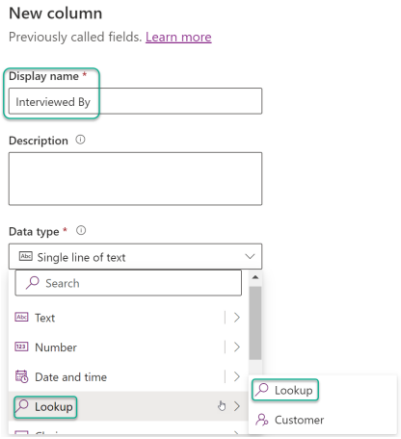
- Click on New Column and Specify the Display Name as *Interview Detailed Notes* and select the data type *as Text-> Text area.* Click on Save



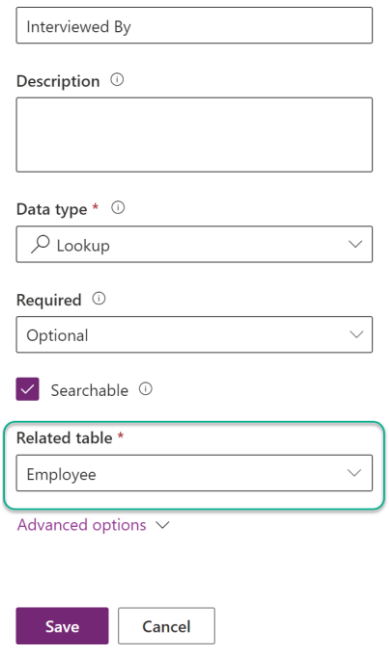
- Click on New Column and Specify the Display Name as *Interview Date* and select the data type *as Date-> Date only.* Click on Save



- Click on New Column and Specify the Display Name as *Interviewed By* and select the data type *as Lookup-> Lookup.*



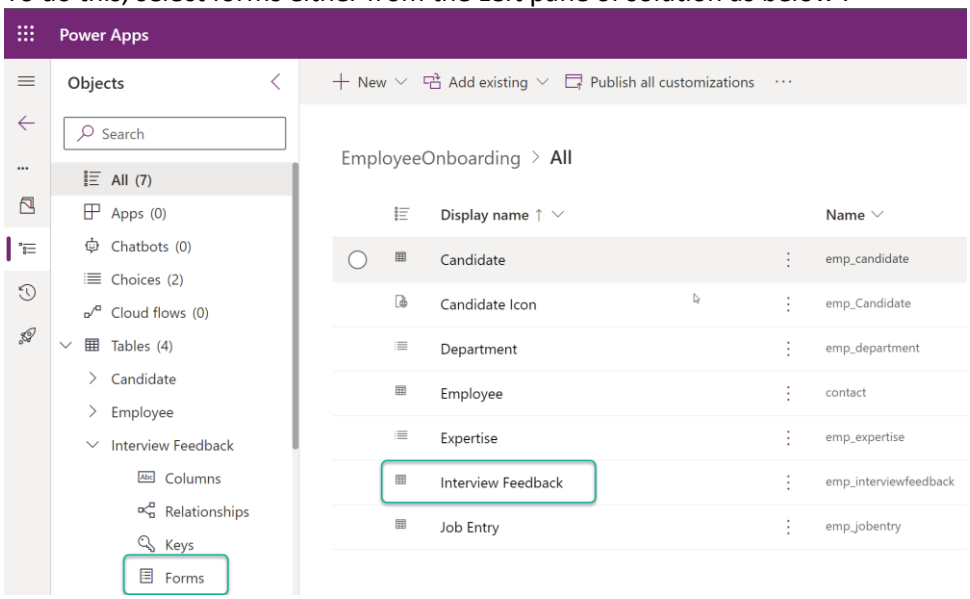
Specify the lookup table as *Employee* and Click on Save.



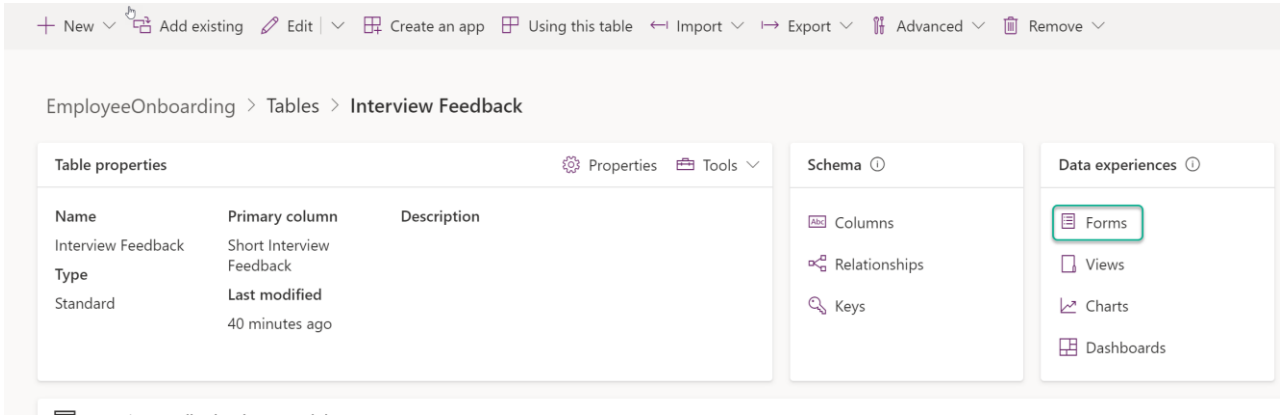
Thus, we have completed the creation of the columns needed for the Interview Feedback table.

Create the Main Form for Interview Feedback Table

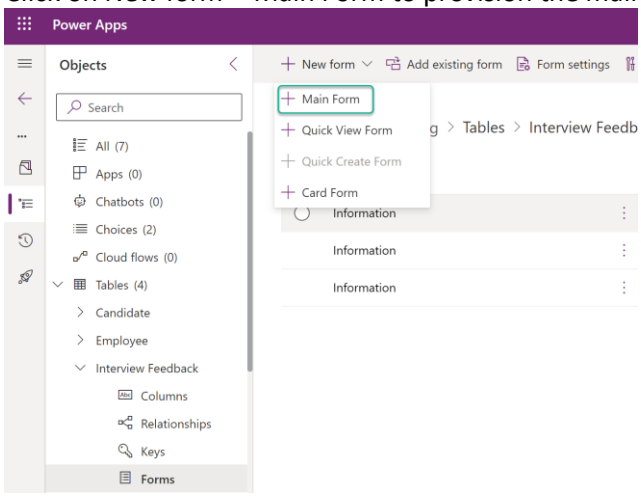
Now lets create the main form which will act as the main entry point of the information by the end user. To do this, select forms either from the Left pane of solution as below :



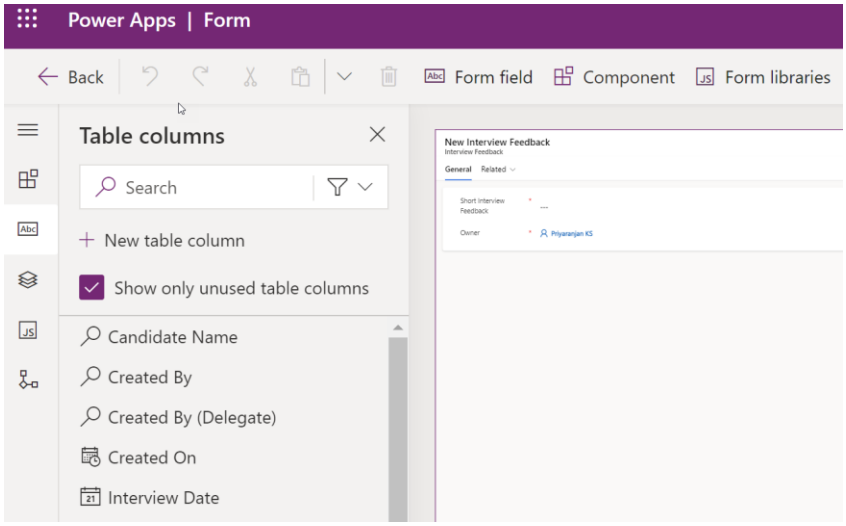
Or select the Interview Feedback table and select Forms as below.



Click on New form-> Main Form to provision the main form for the table

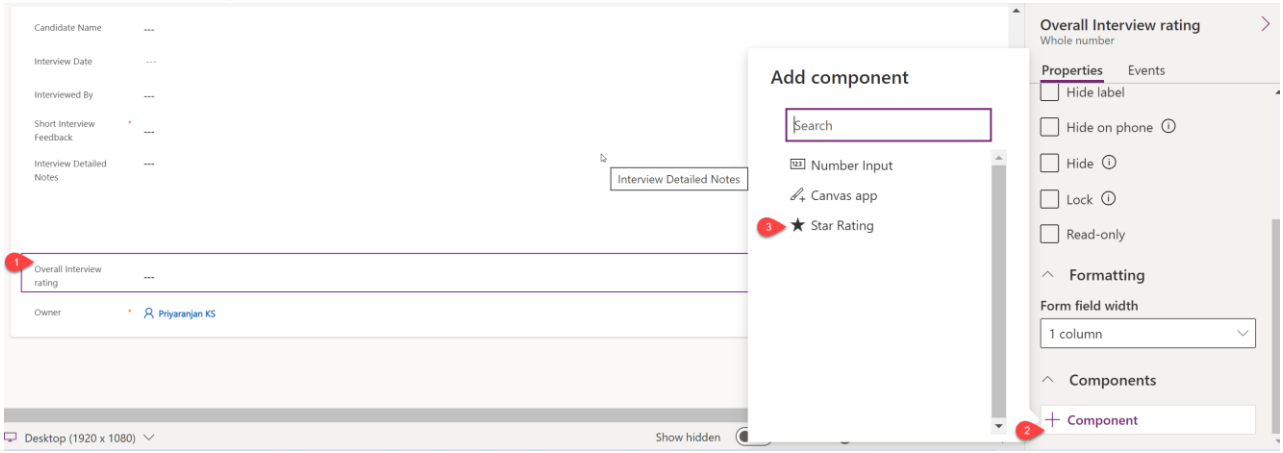


The default form view will look as below with few default columns.

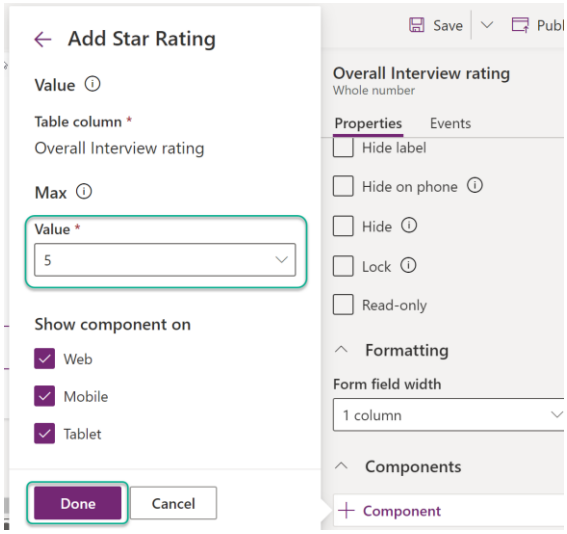


Lets drag and drop additional columns which we have created recently onto the form. For the *Interview Detailed Notes* field, increment the *Form field height* to 7 rows to make space for the detailed data entry

After adding the *Overall Interview rating*, click on *Component* from the Properties tab and select *Star Rating* to enable the star rating UI feature.



Select the maximum star value as 5 and click on Done



This completed the design of the main form for the Interview Feedback table

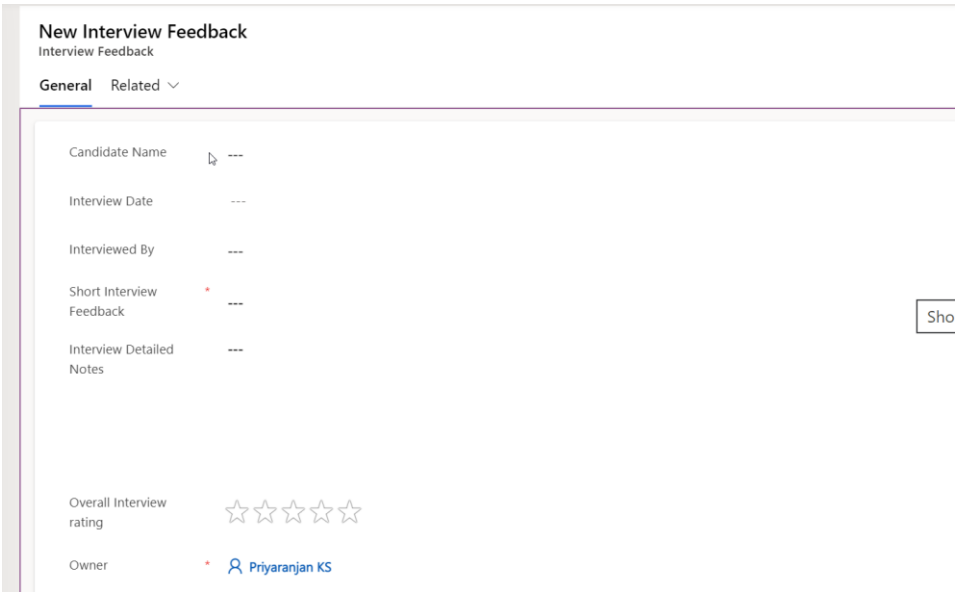
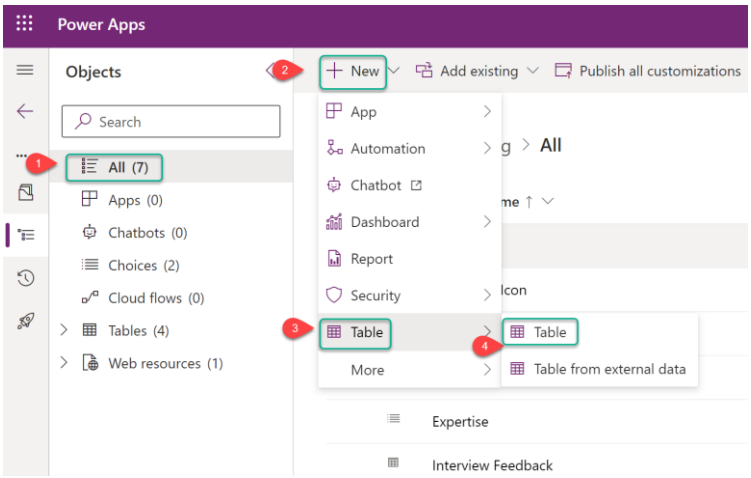


Table 5 : Onboarding

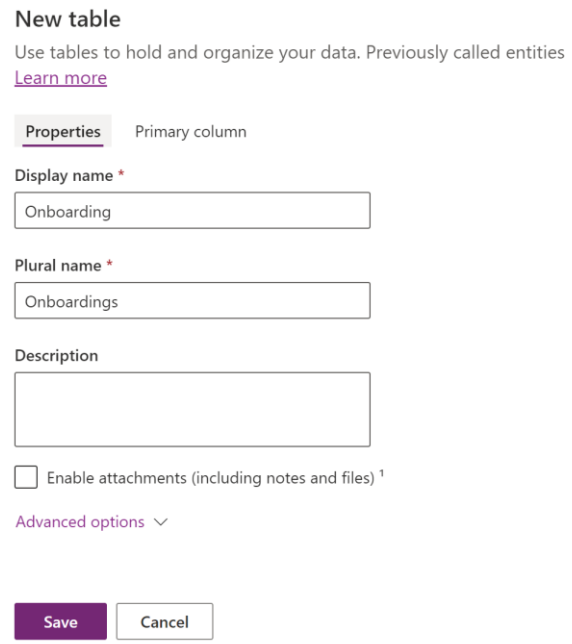
Lets create the final table Onboarding which will be used to store the details related to the candidates who have successfully passed the interview and their joining details.

Create Onboarding Table

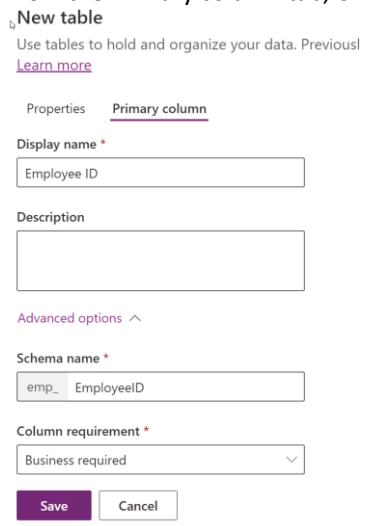
To create the table, lets head over to the solutions left pane and click on All. Then, select New -> Table -> Table.



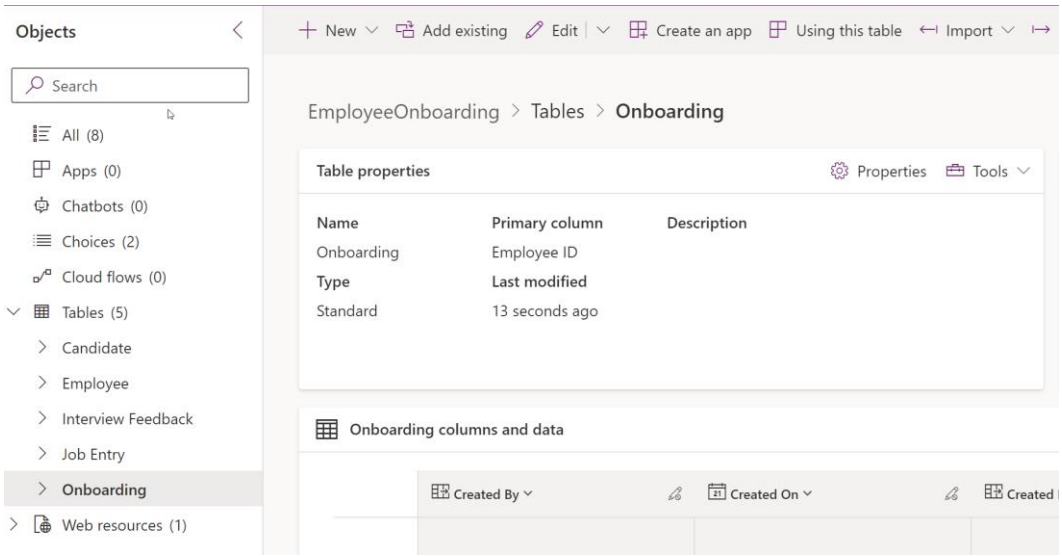
In the properties tab, Specify the table name as Onboarding.



From the Primary column tab, Change the display name from *Name* to *Employee ID* . Click on Save.

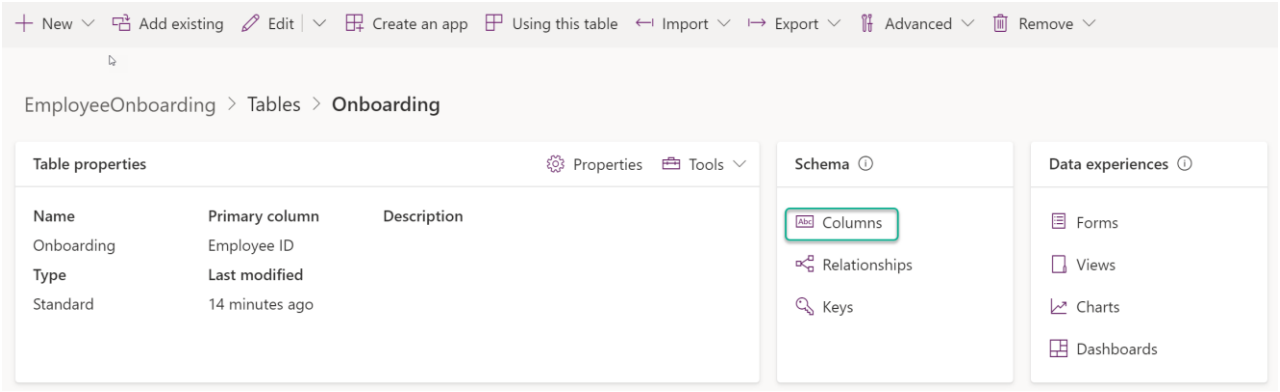


Thus the Onboarding table has been provisioned

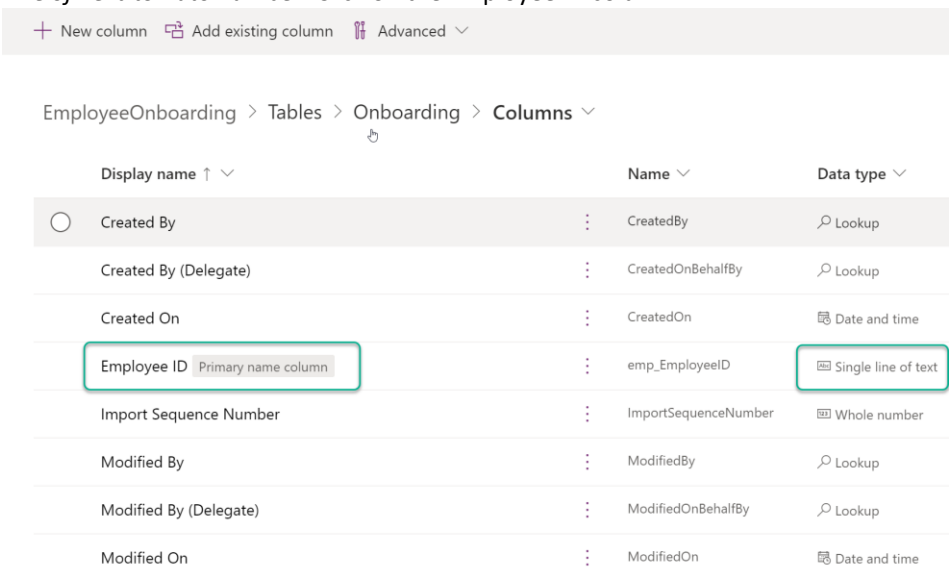


Create Columns for Onboarding Table

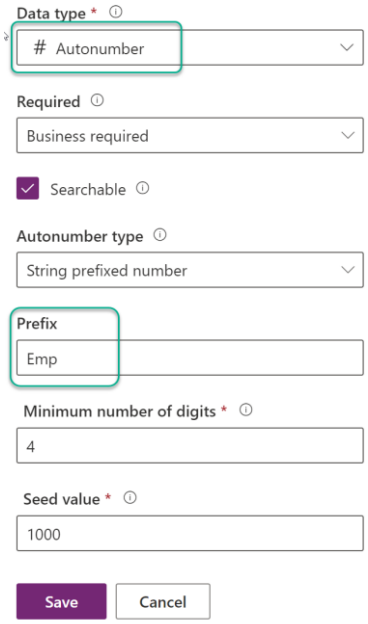
Let's create/update the columns needed to store onboarding information in the table by clicking on *Columns*



Before creating new custom column, we will change the datatype of the *Employee ID* primary column from *Single Line of Text* to *Auto number*. Click on the Employee ID column.



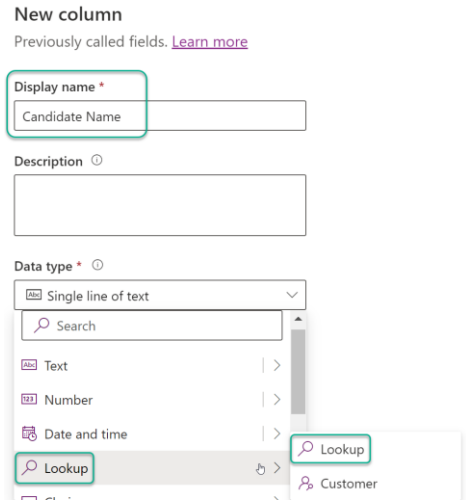
Change the data type to *Autonumber* and add the Prefix as *Emp* . The seed value indicates the starting value that would be used for incremental employee number. The ID would hence look like Emp-1000,Emp-1001 etc: . Click on Save.



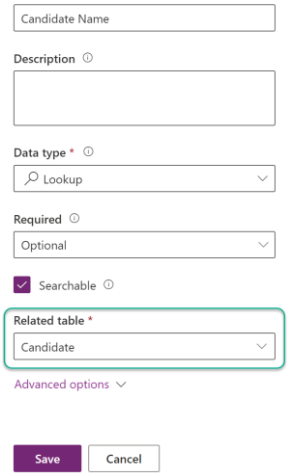
Let's create the rest of the custom columns using the below schema

Column Name	Data type
Candidate Name	Lookup
Department	Choice
Joining Date	Date
Position	Choice
Joining Location	Choice
Manager	Lookup
Short Interview Feedback	Lookup

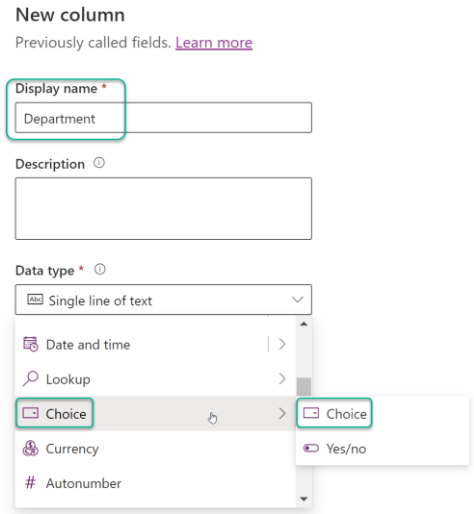
- Click on New Column and Specify the Display Name as *Candidate Name* and select the data type *as Lookup-> Lookup.*



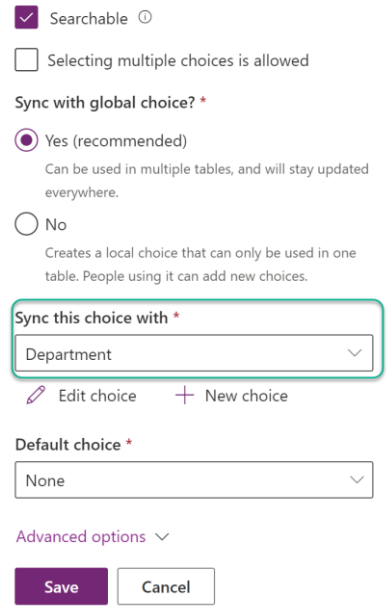
Select the lookup table as Candidate and click on Save.



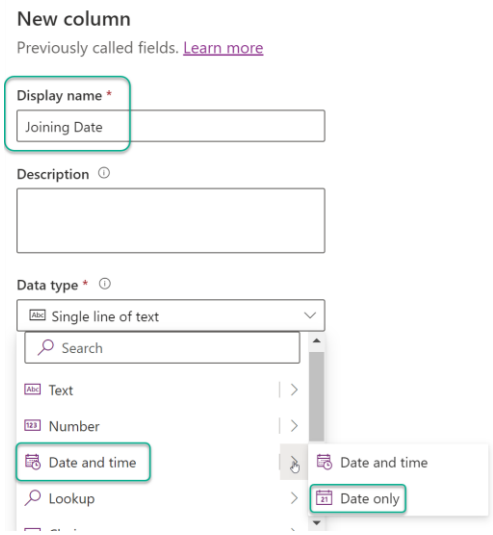
- Click on New Column and Specify the Display Name as *Department* and select the data type *as Choice-> Choice.*



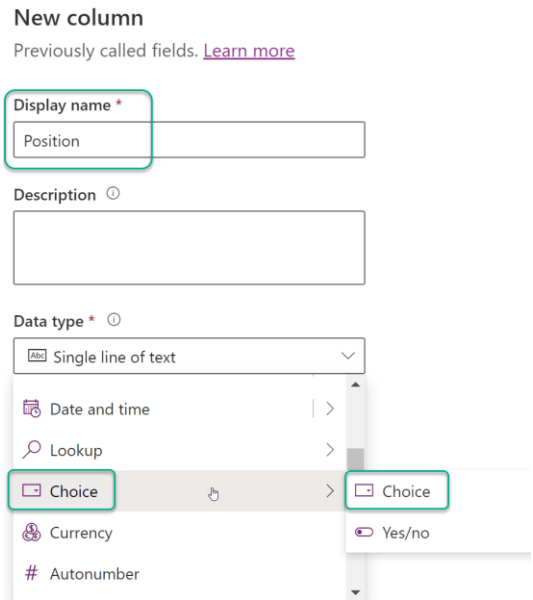
We will reuse the global choice field - *Department* we had created in the Job Entry table. Click on Save.



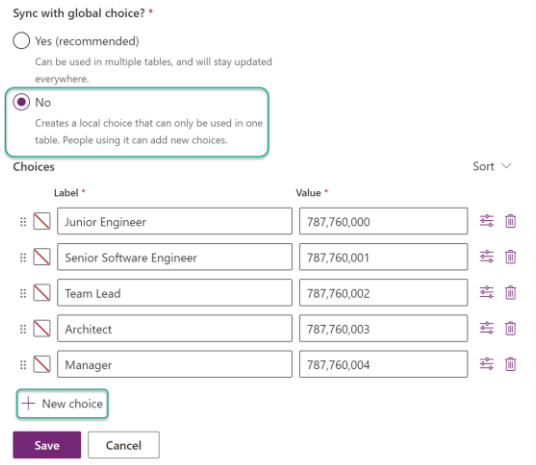
- Click on New Column and Specify the Display Name as *Joining Date* and select the data type *as Date and time-> Date only*



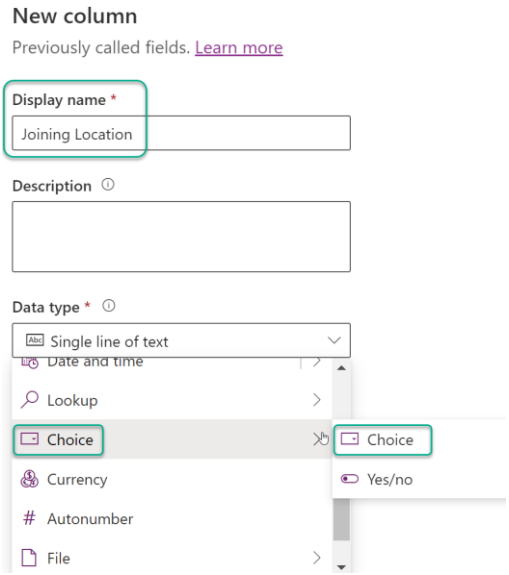
- Click on New Column and Specify the Display Name as *Position* and select the data type *as Choice-> Choice.*



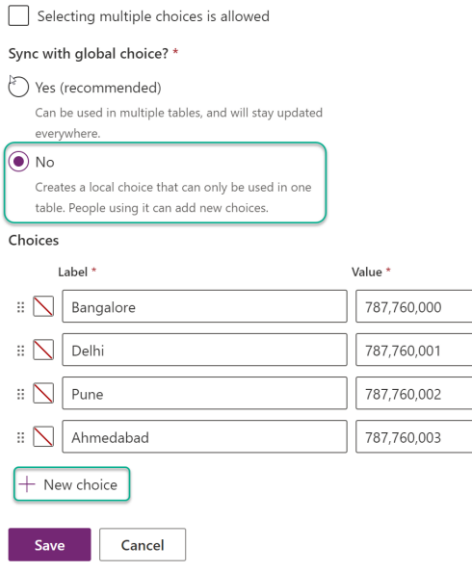
We will use the below local choice value set which we can add by clicking *New choice* .Click on Save.



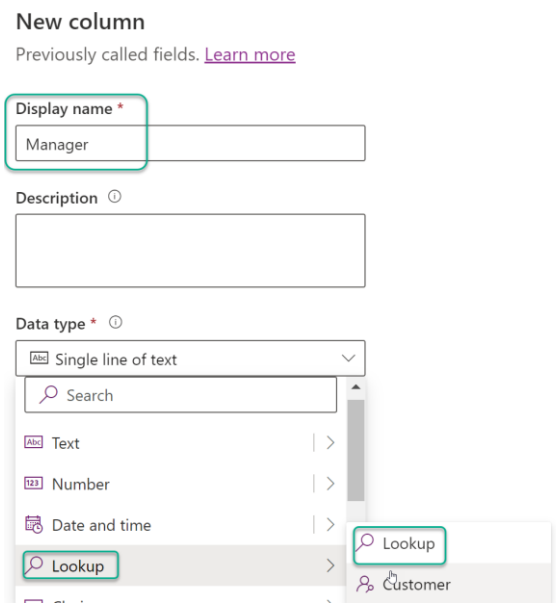
- Click on New Column and Specify the Display Name as *Joining Location* and select the data type *as Choice-> Choice.*



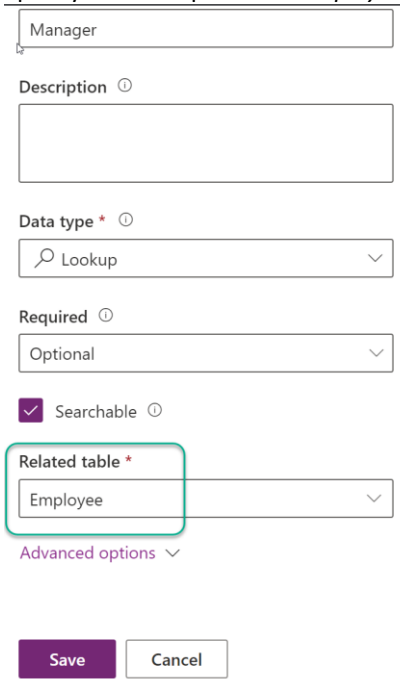
We will use the below local choice value set which we can add by clicking *New choice* .Click on Save.



- Click on New Column and Specify the Display Name as *Manager* and select the data type *as Lookup-> Lookup.*



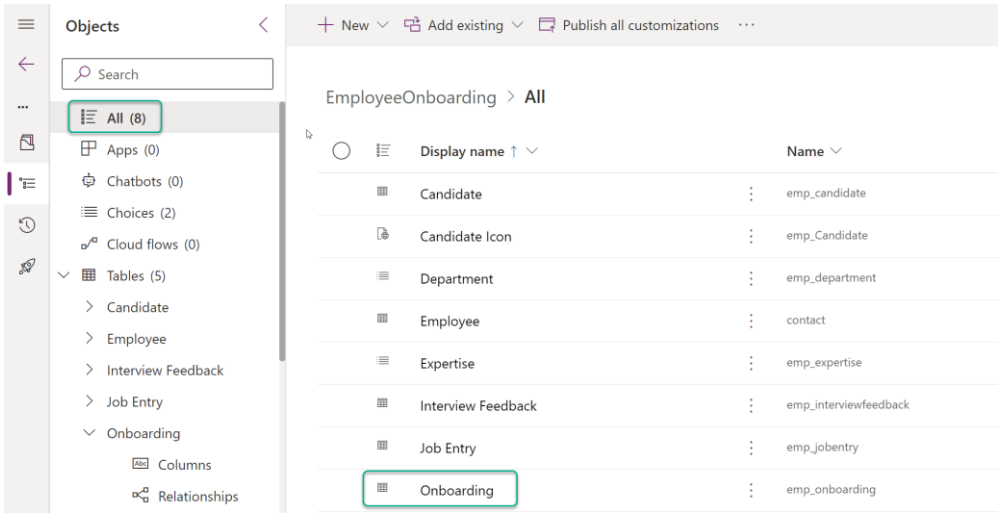
Specify the lookup table as *Employee* and click on Save.



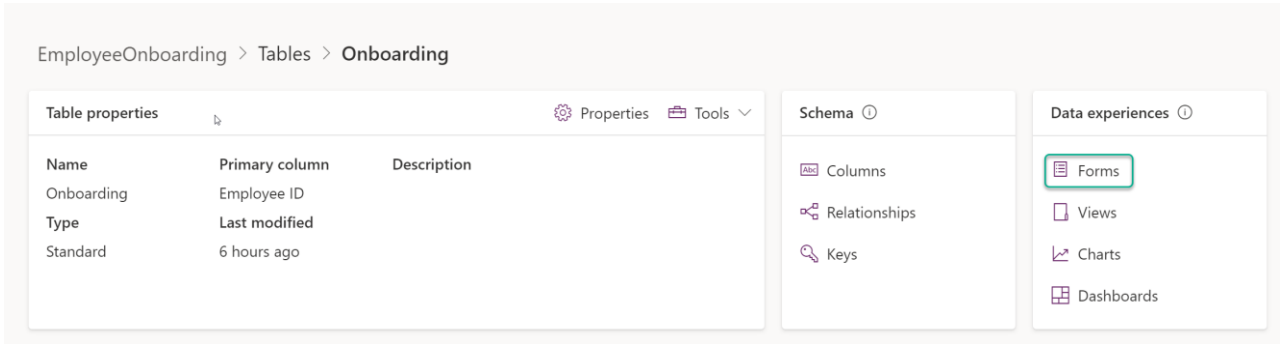
Thus, we have created the columns needed for the Onboarding table.

Create Main Form for Onboarding Table

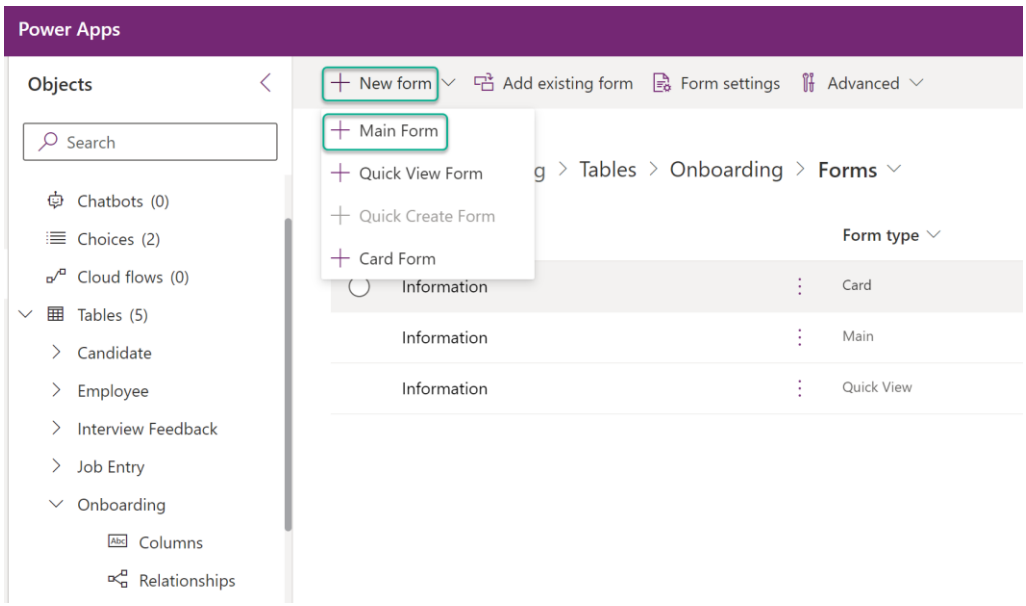
Now let's create the main form that would act as the front end for end users to input the onboarding data. To do this, Lets open the Forms section of the Onboarding table by going to All -> Onboarding



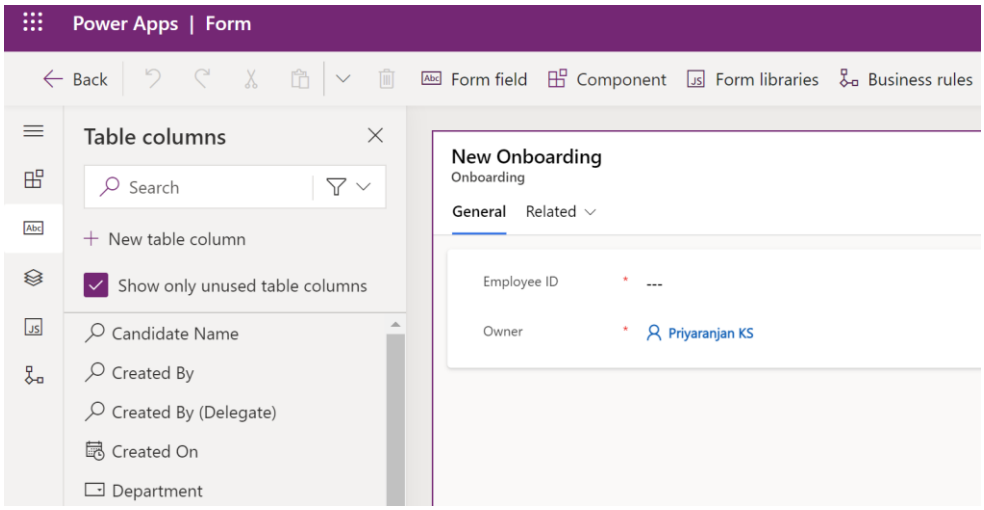
Select Forms from the table.



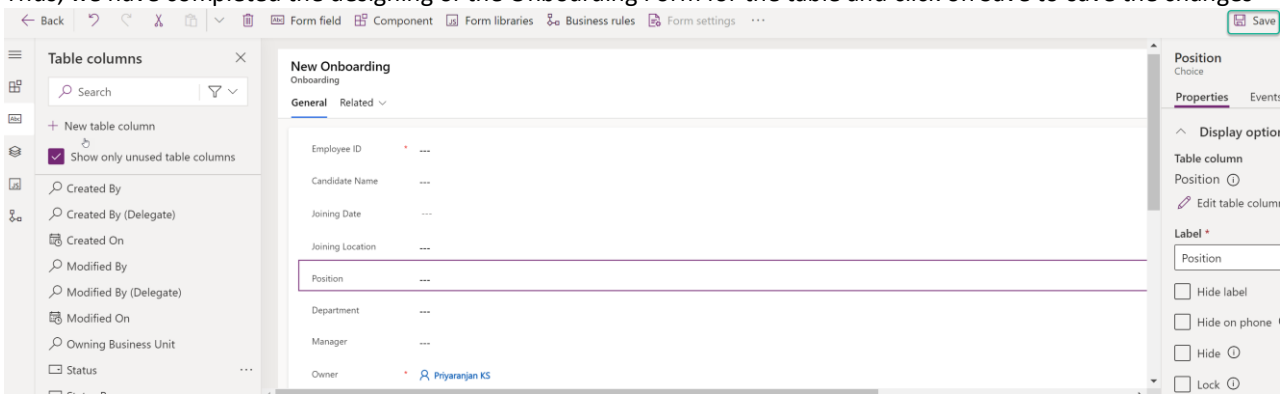
Create the main form by selecting New form -> Main Form



The Main form would show with few default columns. Lets add more columns to the form by dragging and dropping it from the left pane.



Thus, we have completed the designing of the Onboarding Form for the table and click on Save to save the changes

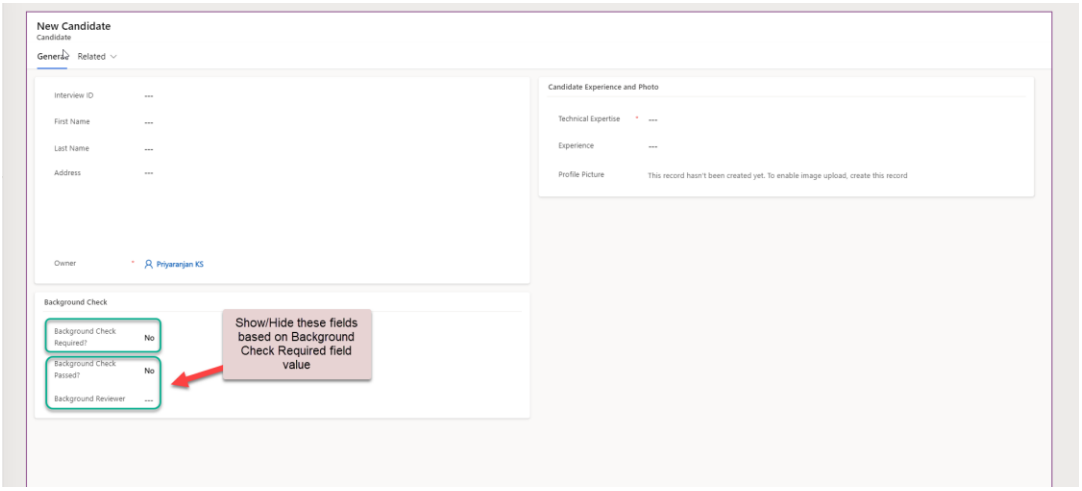


Business Rules

Business rules enables us to implement column level business logic. It can be applied on Main forms as well as Quick Create Forms. Some of the possibilities of Business Rules are :

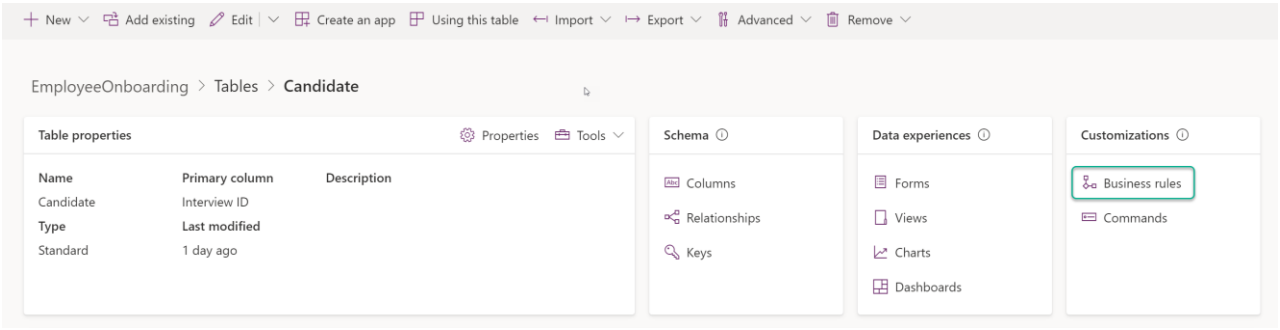
- Set column values.
- Clear column values
- Set column requirement levels.
- Show or hide columns.
- Enable or disable columns.
- Validate data and show error messages.
- Create business recommendations based on business intelligence.

In our hands on, we will implement Business rules in the Candidate table. The candidate table contains a Background Check Section. In this section, we will show/hide the *Background check passed?* And *Background check reviewer* based on the value of *Background check required?*

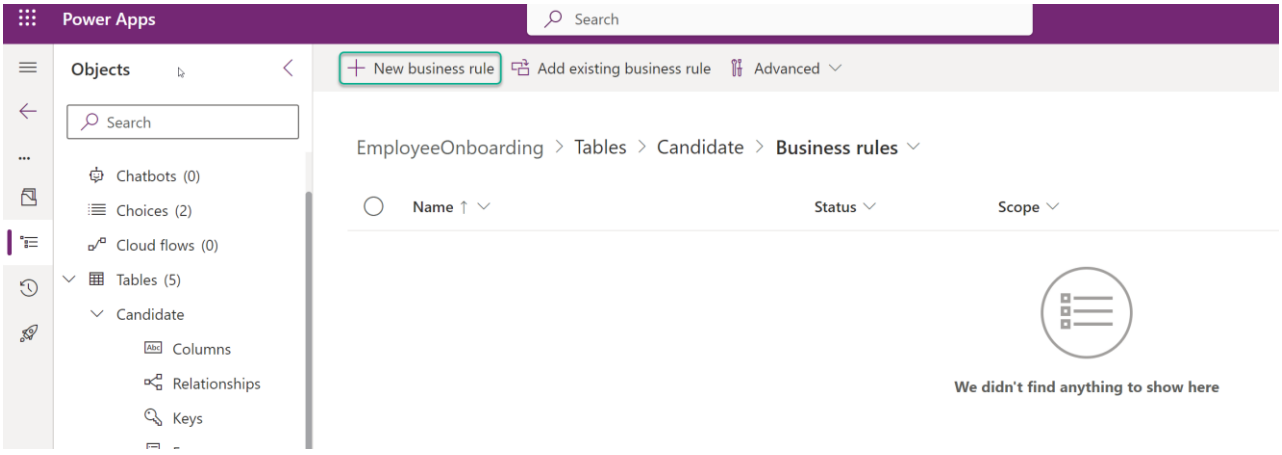


Add Business Rule

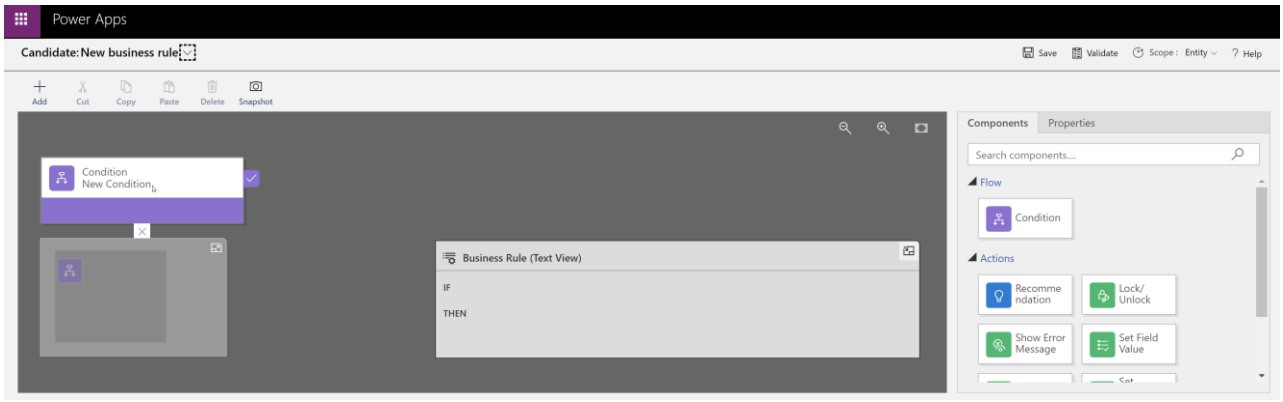
To add a new business rule, head over to the Candidates table and select the Business rules option.



Click on New business rule to add a new conditional rule.

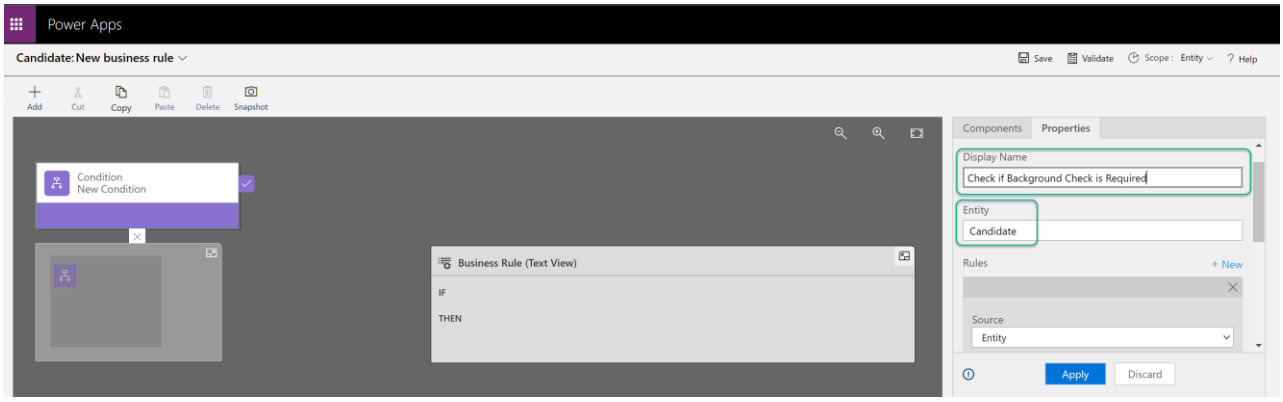


This will open the business rule designer where we can design the logic.



Every rule starts with a condition and by default a condition is added for you when the designer is opened. Business rule takes one or more actions based on that condition.

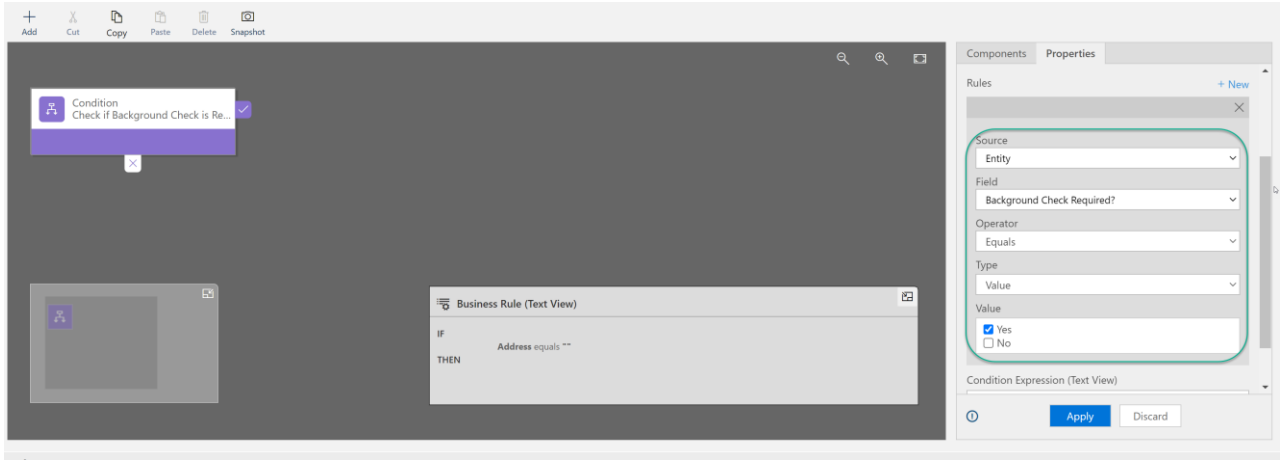
Lets change the Display Name of the condition to "*Check if Background Check is required*". The entity will determine the table on which the rule will run. In our case it is the candidate table.



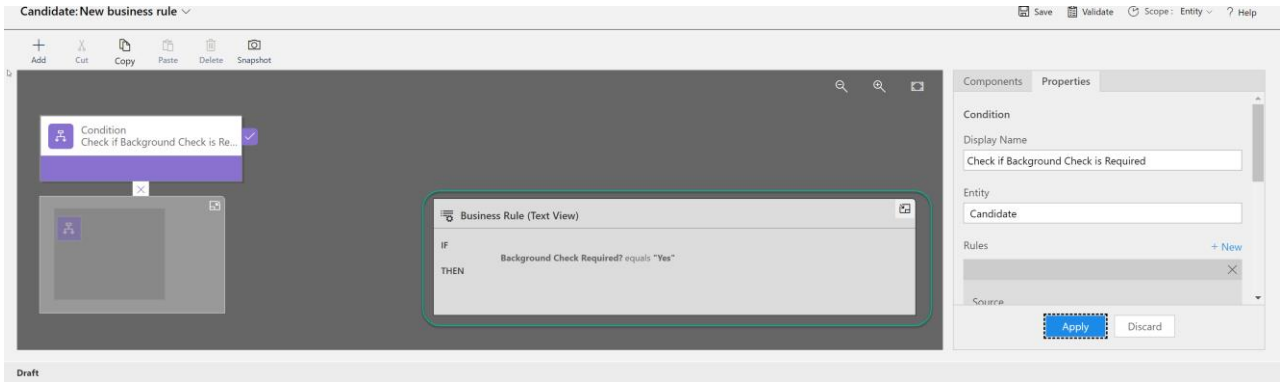
In the rules section, specify the field on which the condition has to be applied, which in our case is the "Background check required ?" field.

In the type , mention "Value" as we are evaluating if the field is equal to some value. Based on the requirement, we can also check if One field is equal to another field

Click on Apply.

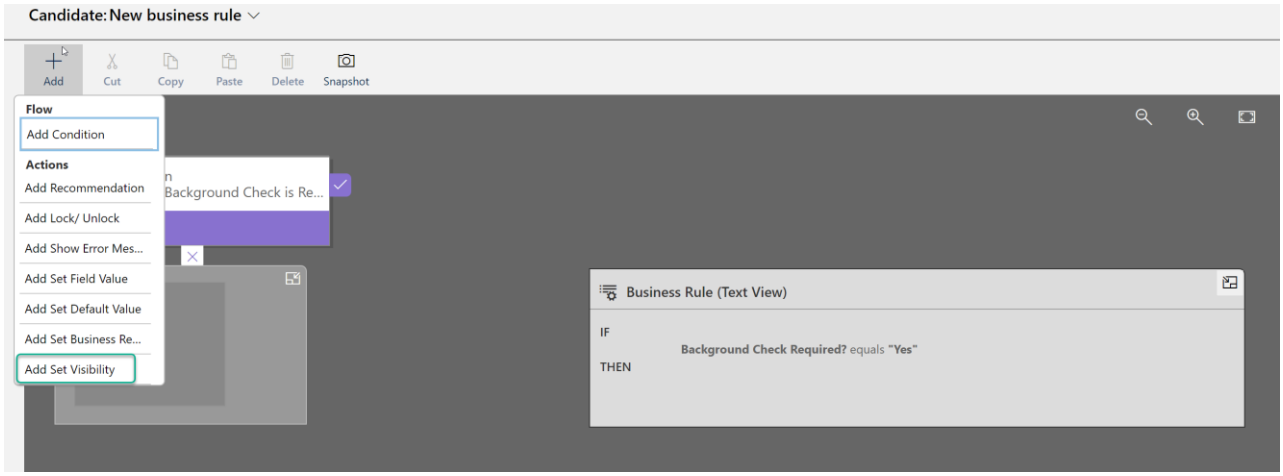


The condition has been saved and the equivalent logic can be viewed from the below highlighted bottom section.

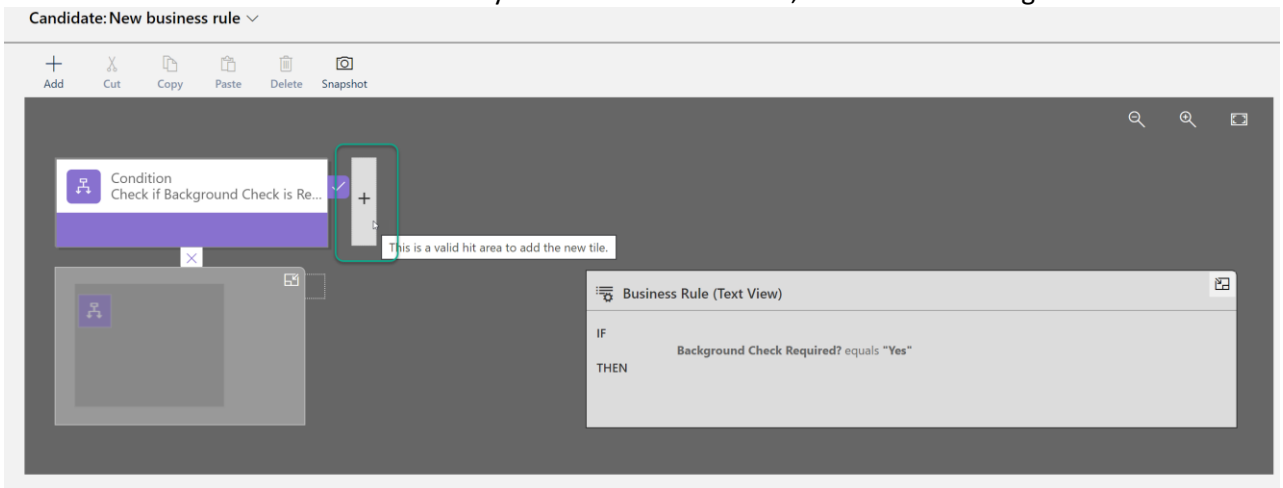


Now lets add the Actions to the right side of the condition, which will be executed if the condition evaluates to true.

Click on Add-> Add Set Visibility

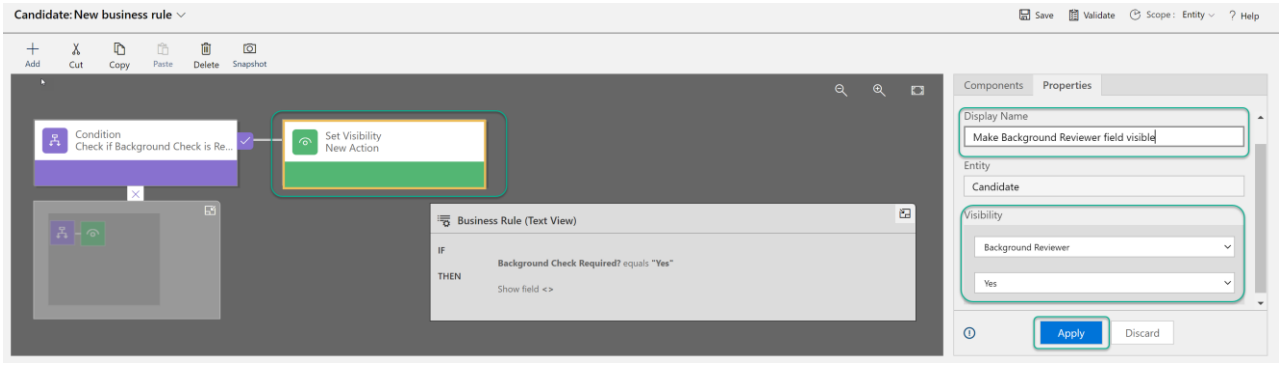


To set the location where the Set Visibility action has to be added , click on the Plus sign

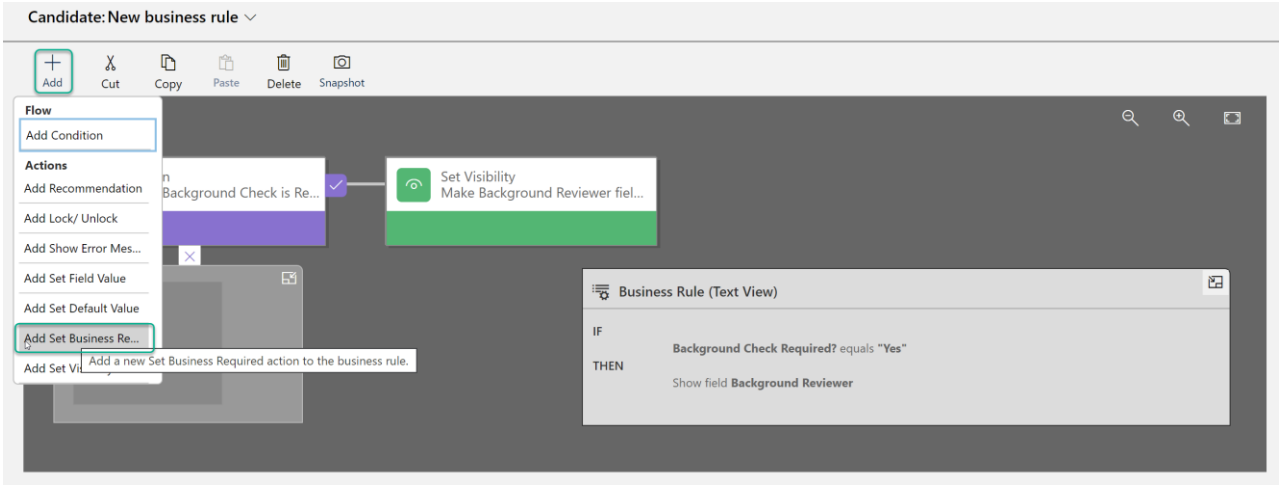


This will add the action tile in the Yes block. Let's rename it to "*Make background reviewer field visible*" and select the column "*Background reviewer*" in the visibility section and set it to "*Yes*"

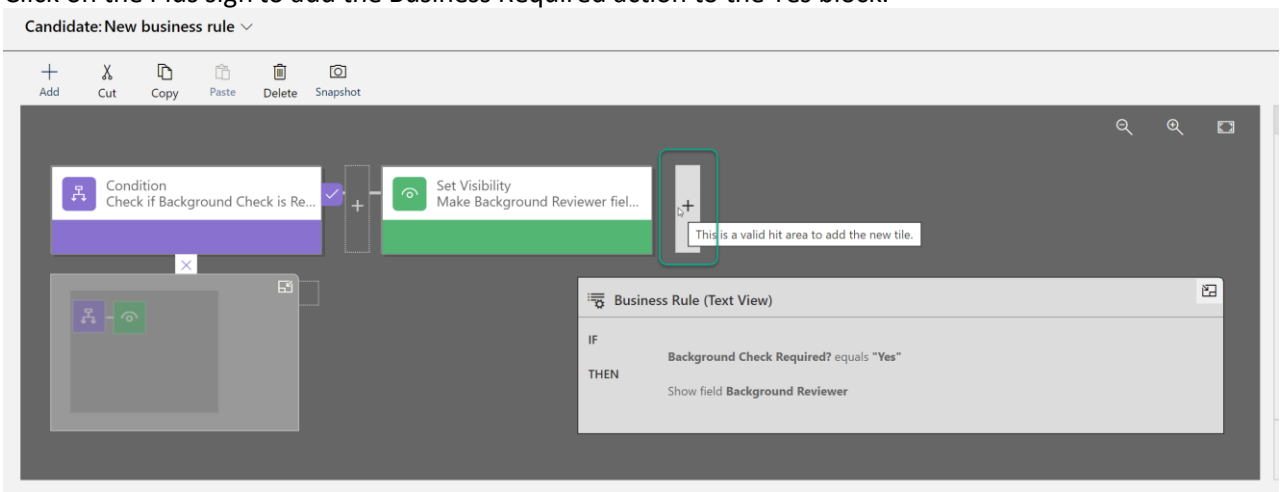
Click on *Apply*.



Let's add another action in the Yes block by clicking on Add. Select "Add Set Business Required"

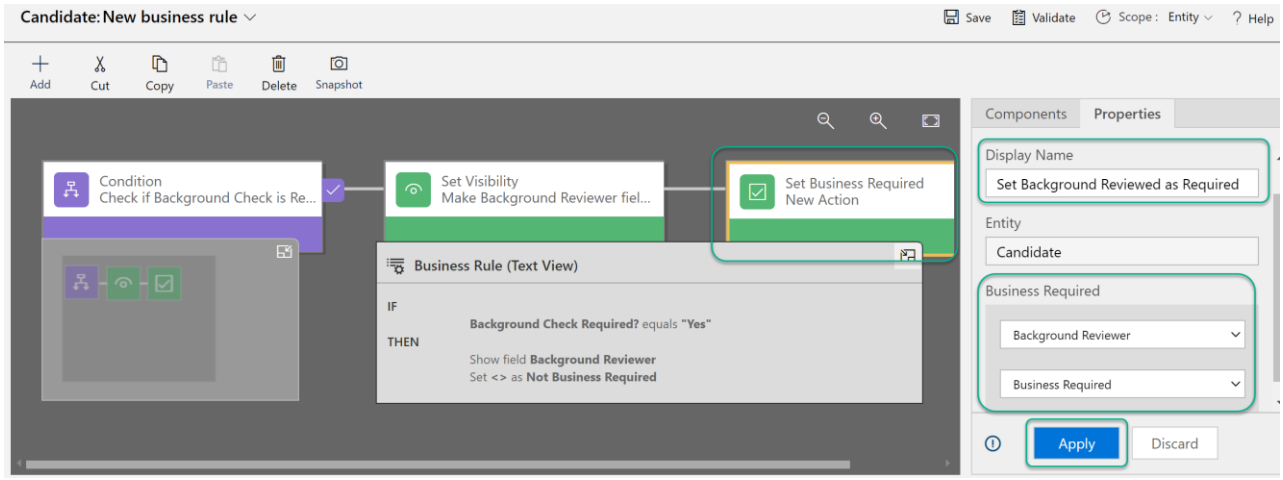


Click on the Plus sign to add the Business Required action to the Yes block.

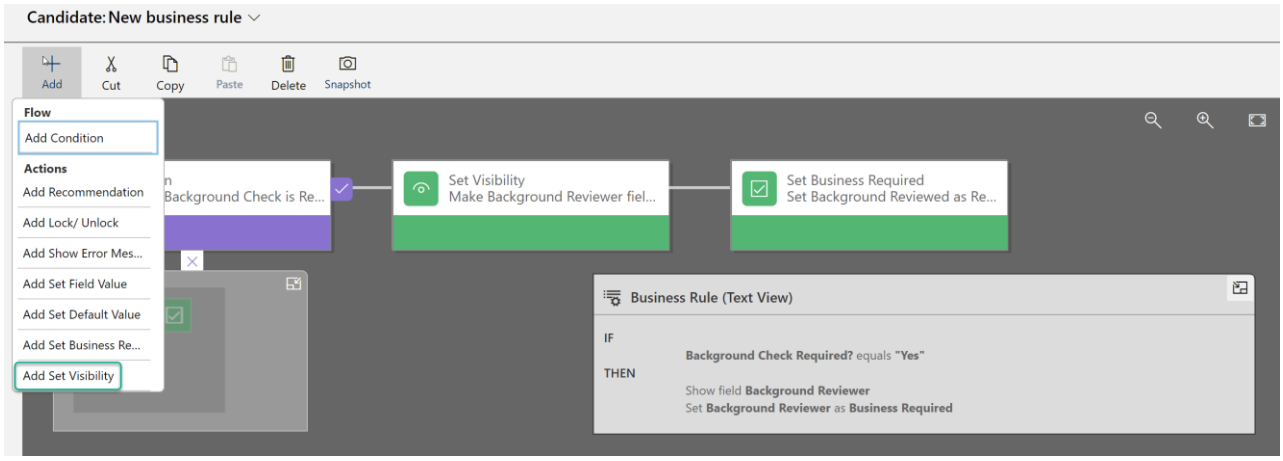


Lets rename the action to "Set Background Reviewer as Business Required". In the Business Required section select the column as "*Background Reviewer*" and set it to *Business Required.*

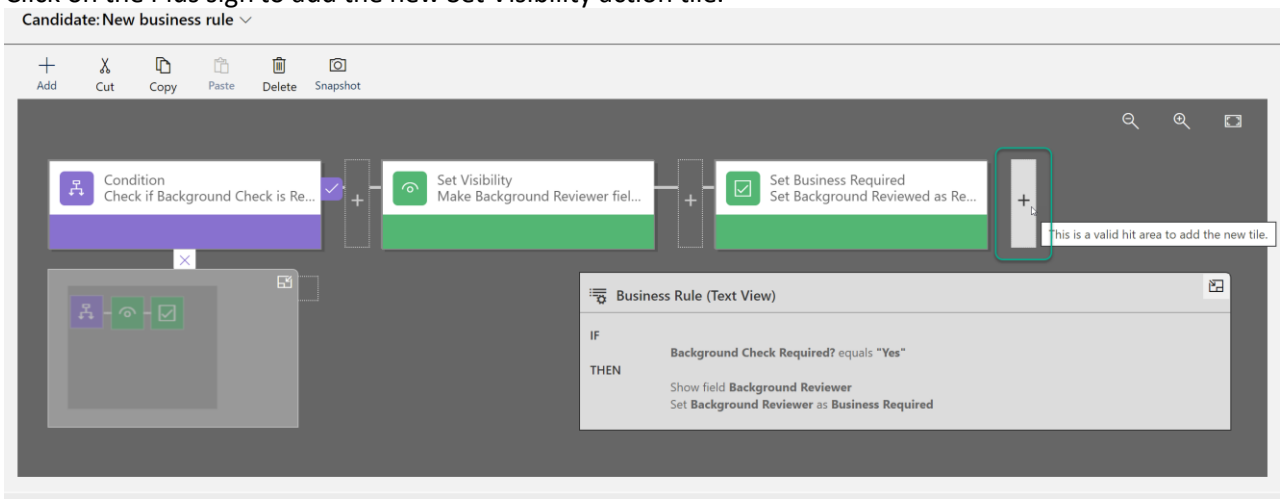
Click on Apply.



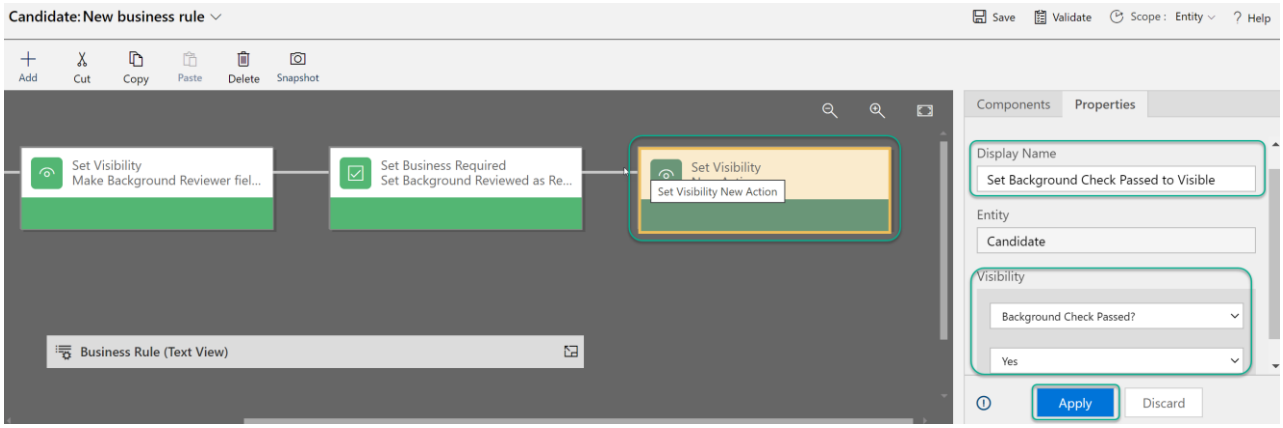
Now lets add another action by clicking on “Add” and select “Add Set Visibility”



Click on the Plus sign to add the new Set Visibility action tile.

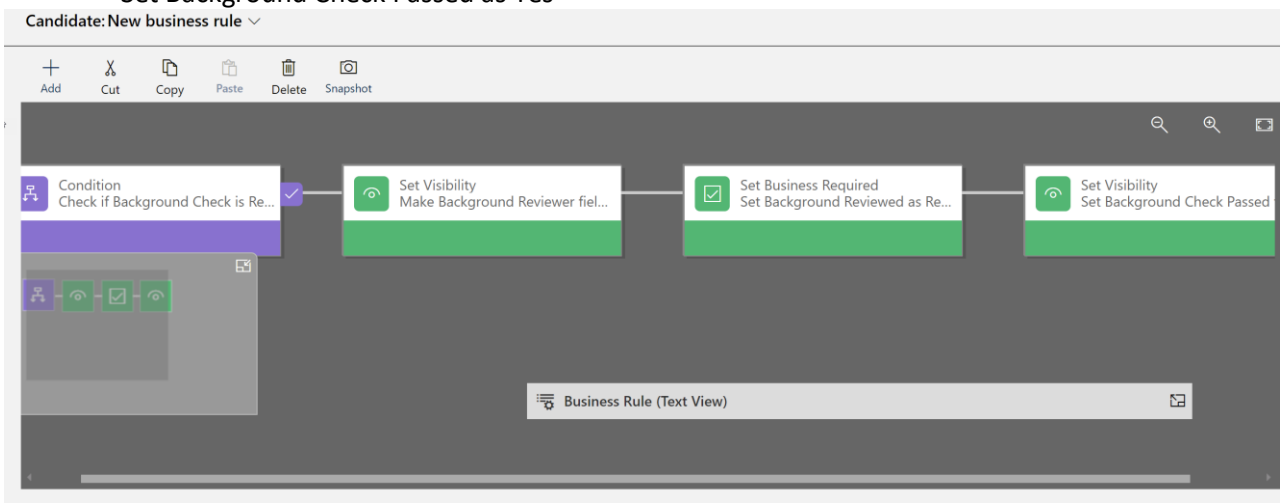


Rename the display name to “*Set Background Check Passed to Visible*” and in the Visibility section select the “*Background Check Passed?”* column to *Yes.* Click on Apply.



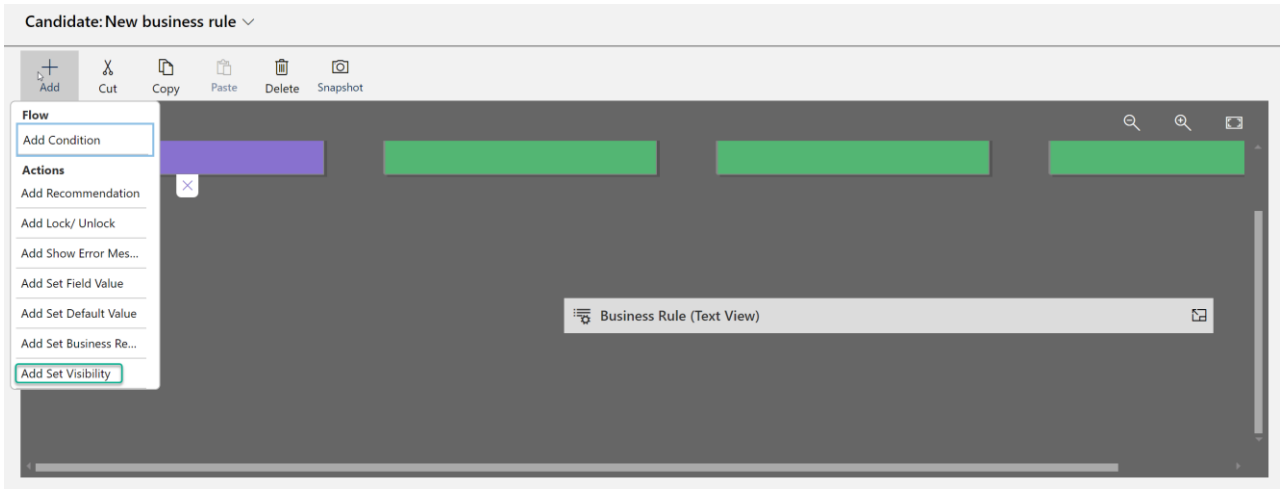
Thus, the Yes Block of the Business Rule Condition has been updated with the below actions

- Set Background Reviewer Visibility to Yes
- Set Background reviewer as Business Required
- Set Background Check Passed as Yes

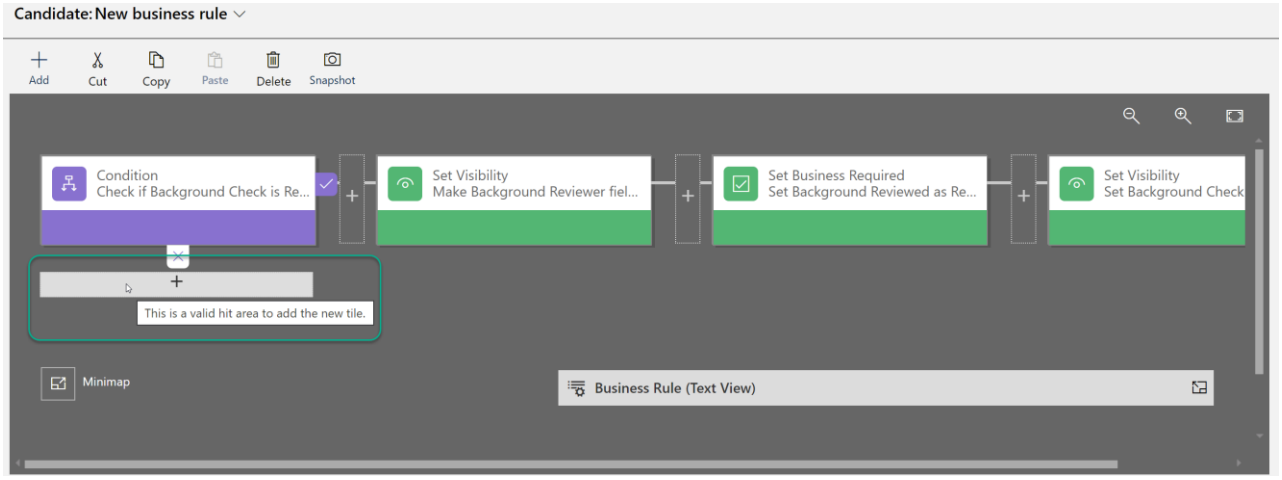


Now lets populate the No Block of the condition with opposite actions. The reason for populating the No block with Turning off the visibility Or Turning off Business required is because if a user toggles the Yes/No field on which the business rule is running, then the equivalent fields should also be turned on/off in response.

To do this, lets click on Add and select *Add Set Visibility*

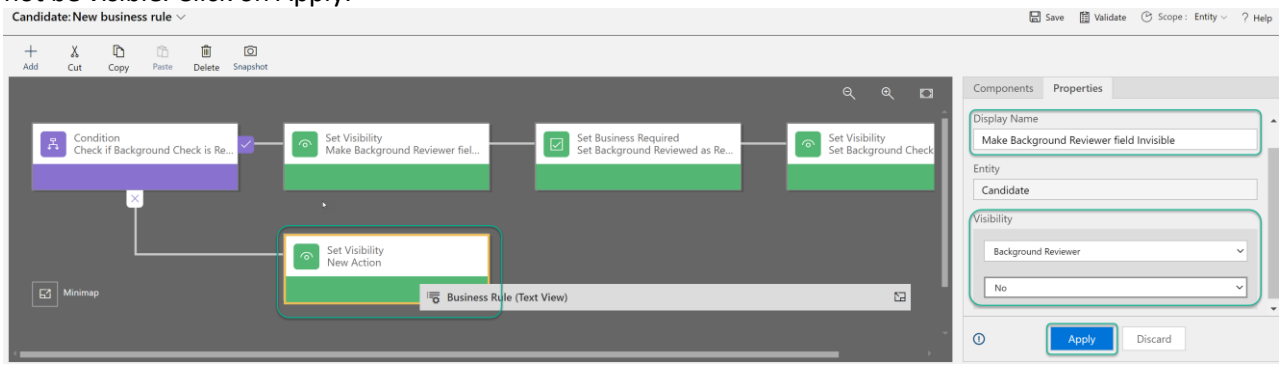


Click on the Plus sign to add the Visibility action so that we can configure it.

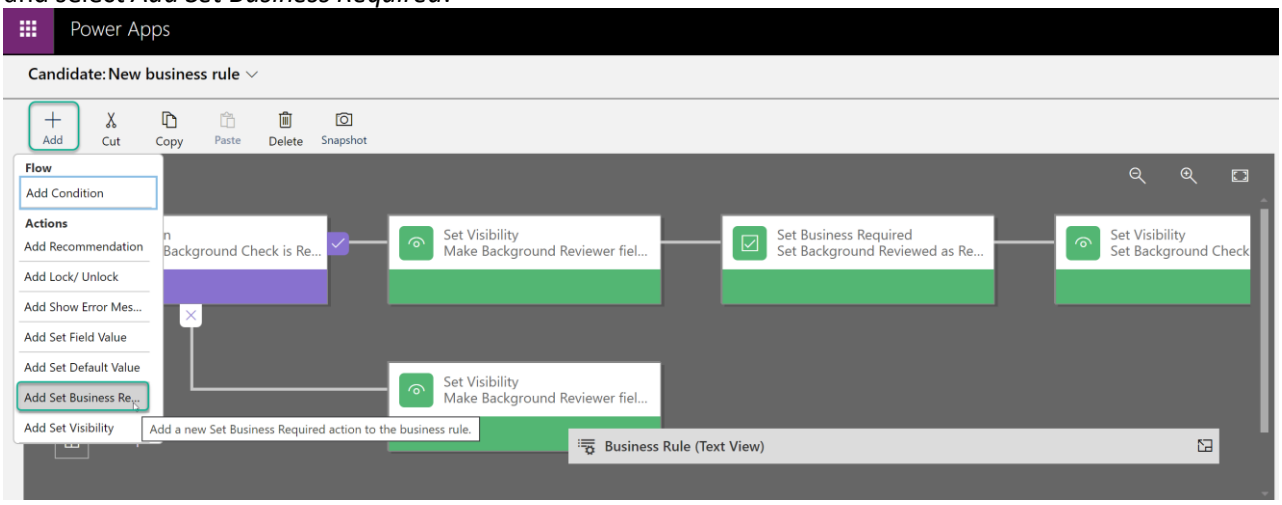


Let's rename the field to "*Make background reviewer field invisible*" and in the visibility section, select the column "*Background Reviewer"* from the dropdown and set it to *No*.

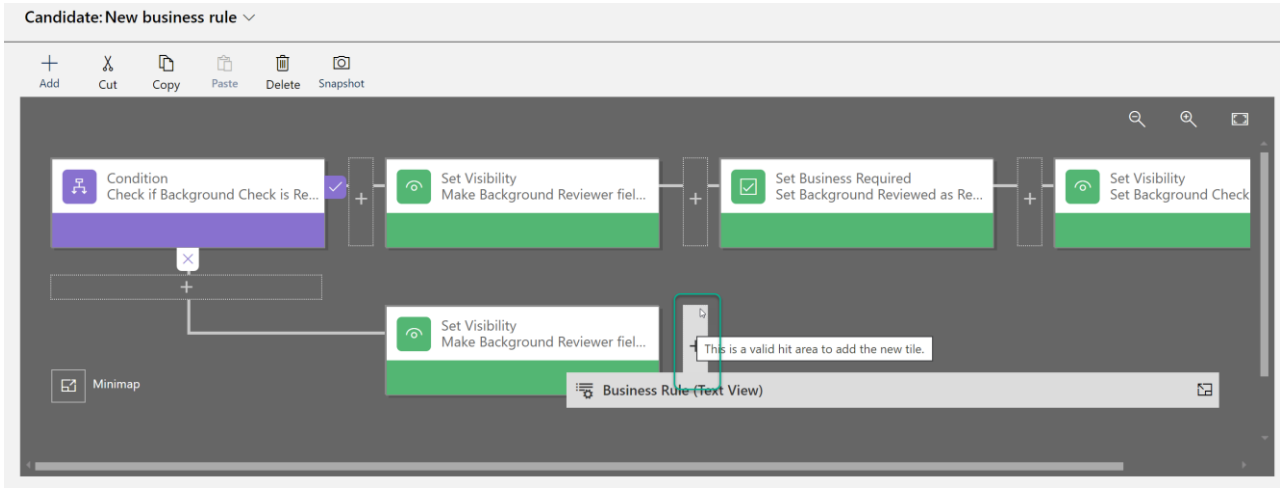
This will ensure that if the *Background Check* field is selected as *No*, then, *Background Reviewer* field will not be visible. Click on Apply.



Now let's add an action to set the *Background reviewer* as not business required. To do this, click on *Add* and select *Add Set Business Required*.

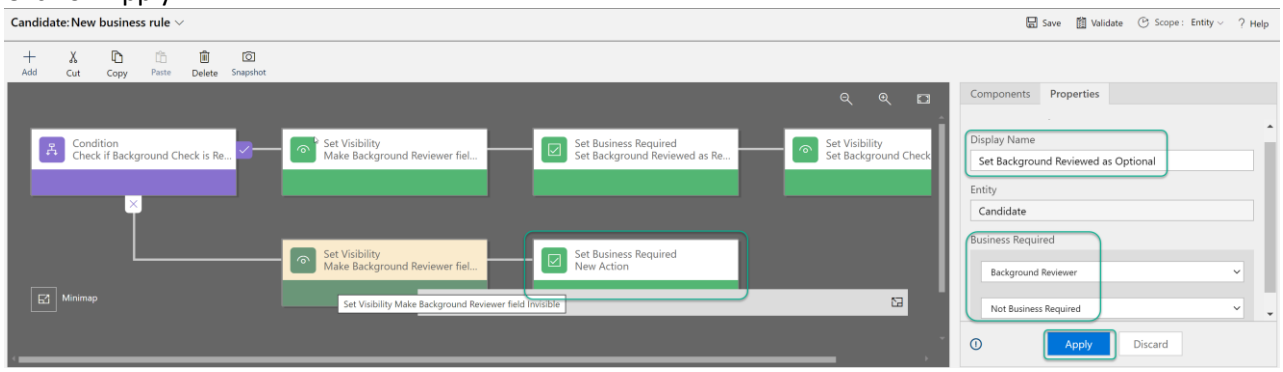


Click on the Plus sign to add the action to the designer.

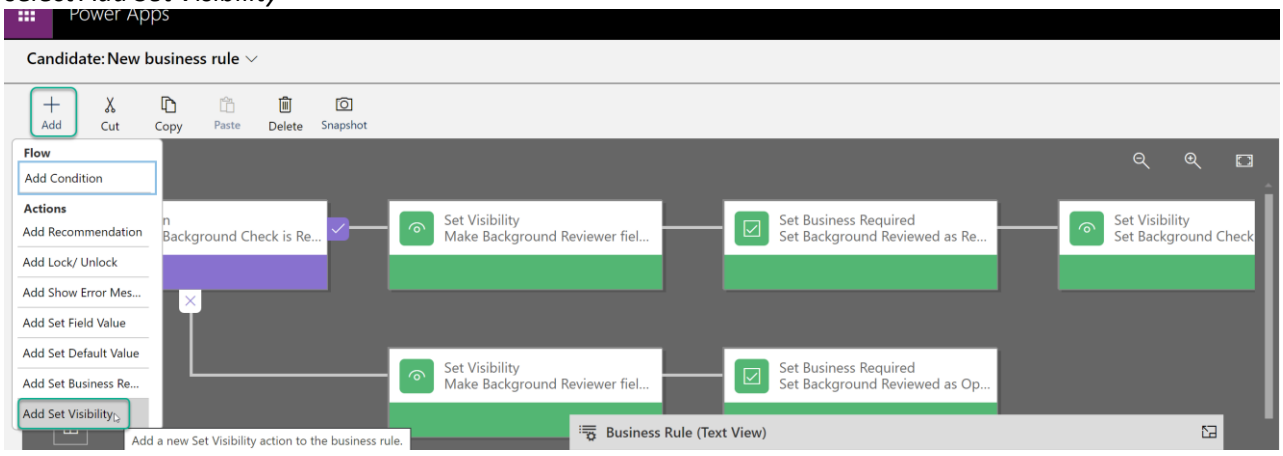


Rename the action to "*Set Background Reviewed as Option*" and in the Business Required section, select the column "*Background Reviewer*" from the dope down and set it to *Not Business Required*

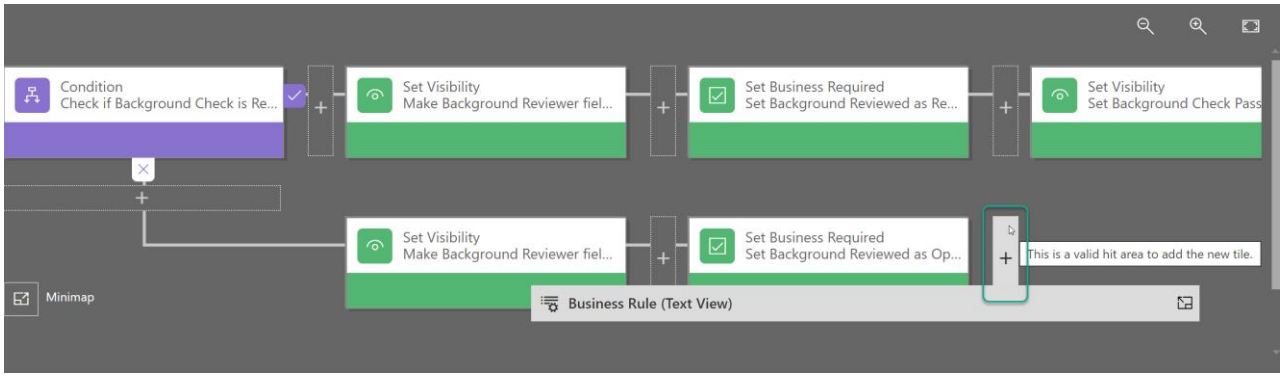
Click on Apply



Lets add the final action to set the Visibility of *Background Check Passed* field to hidden. Click on *Add* and select *Add Set Visibility*

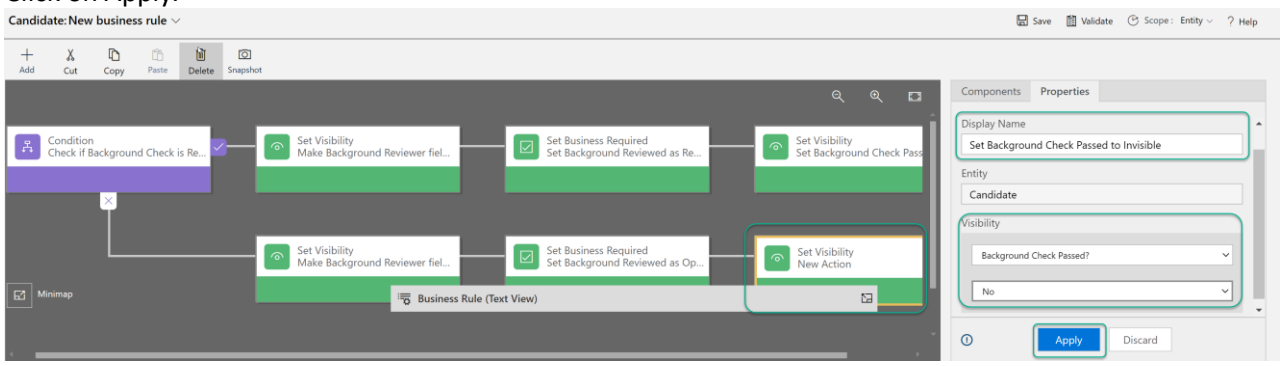


Select the Plus sign to add the action on to the designer.



Rename the action to "*Set Background Check Passed to Invisible*" and in the Visibility section, select the column "*Background Check Passed ?"* and set it to *No*

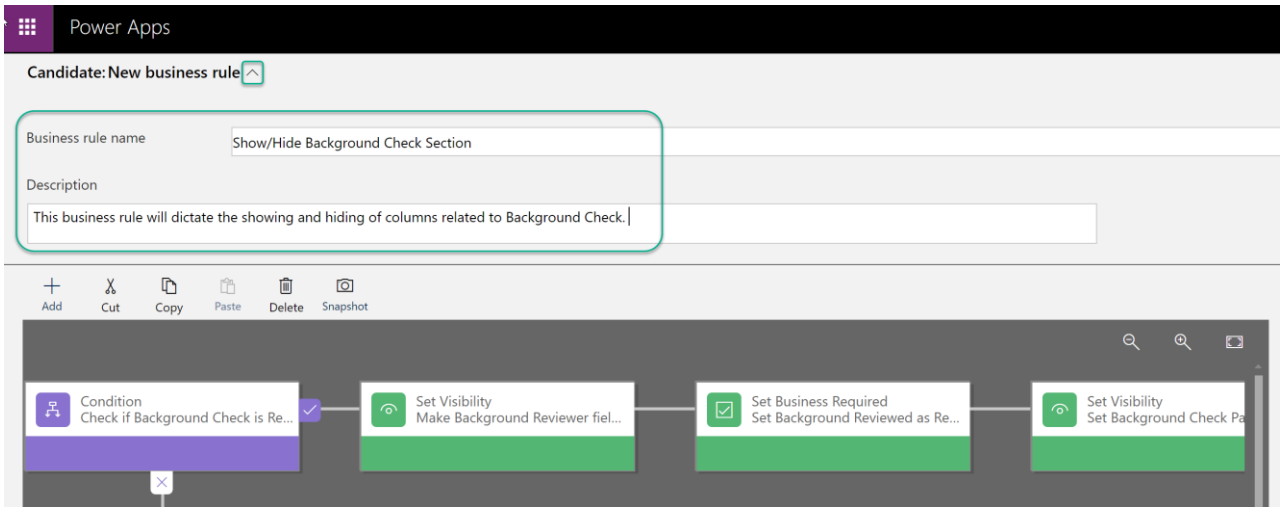
Click on Apply.



Thus in the No block , we have added the below actions :

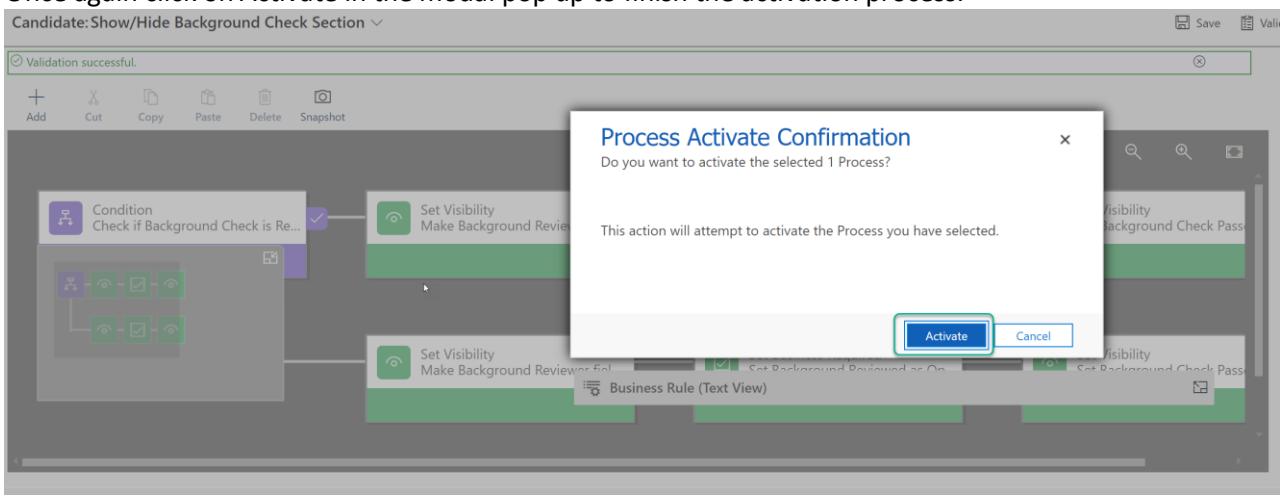
- Set the Background Reviewer to Hidden
- Set the Background Reviewer to Not Business Required
- Set the Background Check Passed ? to Hidden

Let's give a name to the Business Rule by clicking on the Arrow next to "New business rule" . Click on Save.

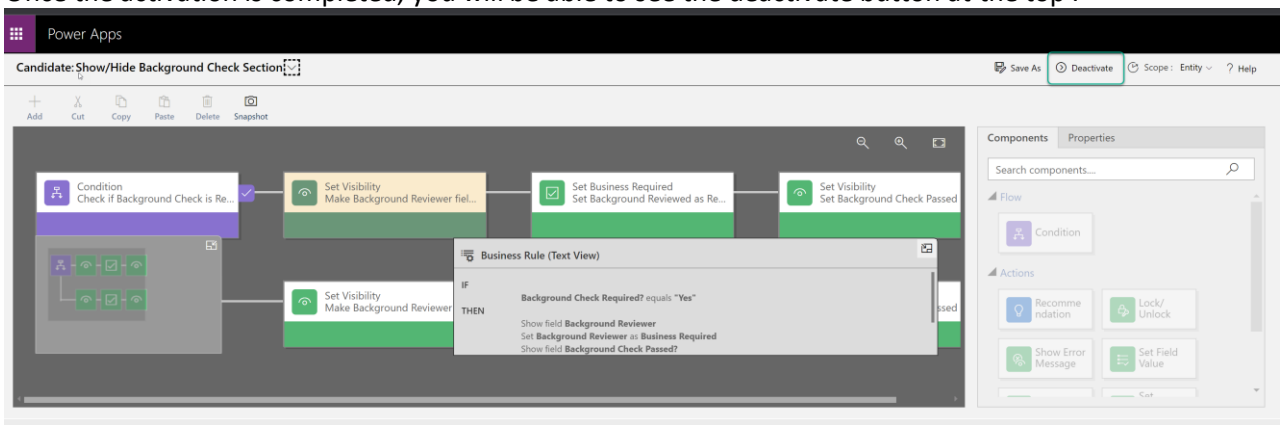


The overall Business Rule will now look like below. Once you have clicked on Save, Click on Activate to publish the business rule which will make it available for use to end users.

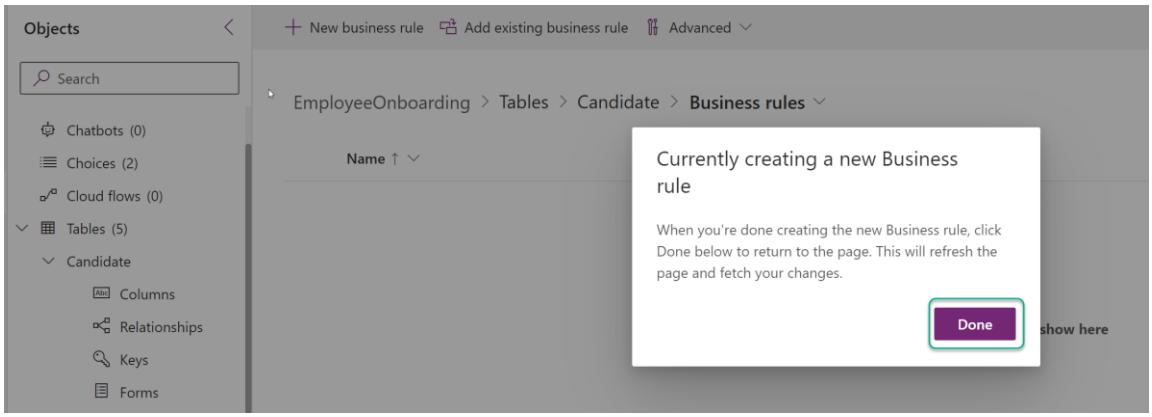
Once again click on Activate in the modal pop up to finish the activation process.



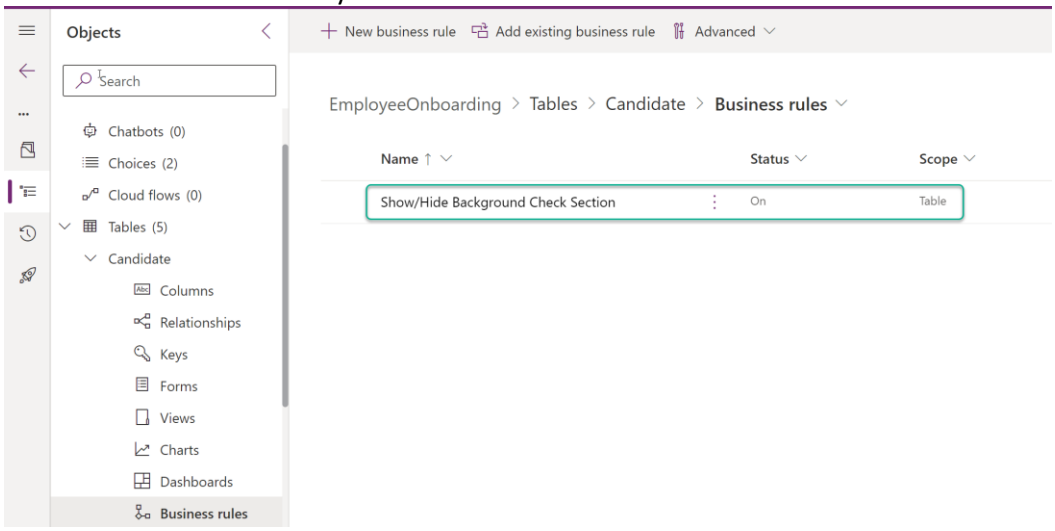
Once the activation is completed, you will be able to see the deactivate button at the top .



Head back to the solution from which the business rule designer was launched and click on Done.



Now we can see the recently created business rule associated with the candidate table.



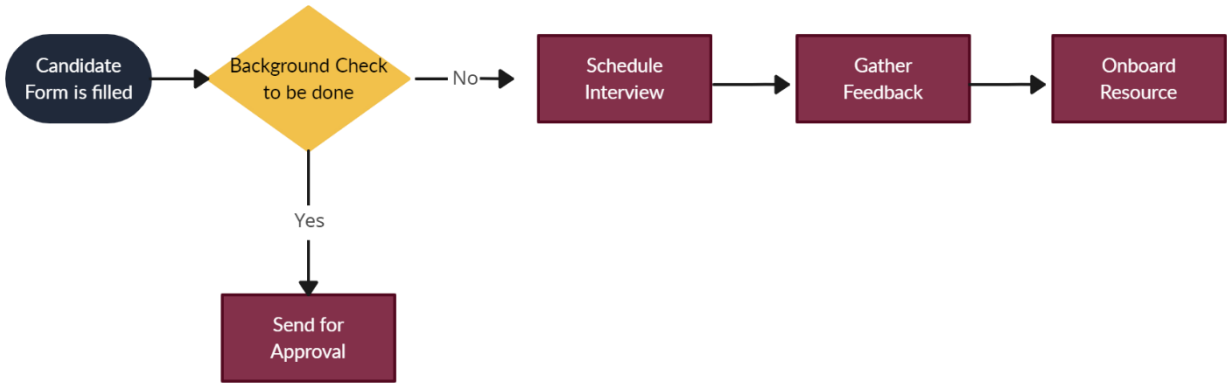
This completes the creation of the Business rule for the candidate table.

Business Process Flow

Business processes provide a step-by-step timeline for the stages of a process. We can open the menu for the active stage, enter each column of required and business-critical data, and then select the next stage when we are ready.

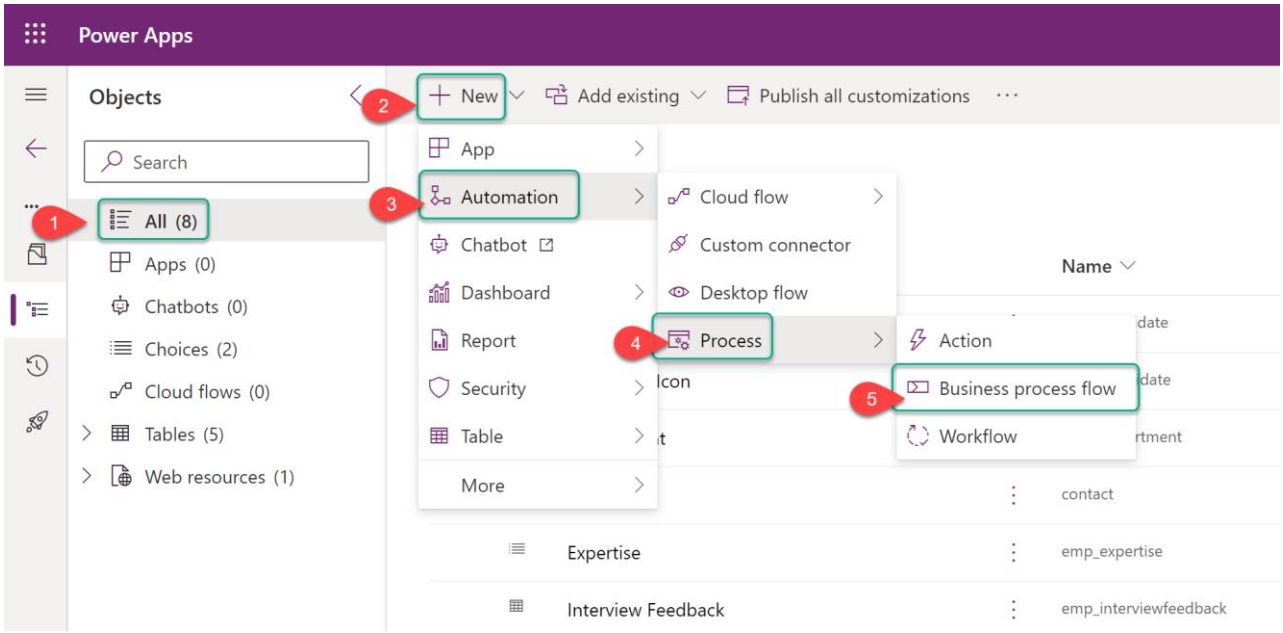
The biggest benefit of Business Process flow is that it helps us to view each step that we need to take to create the new row and fill out the required information according to your organizations business process. So it provides a step by step process guideline to complete the actions.

In our scenario, the business process flow will follow the below flow.

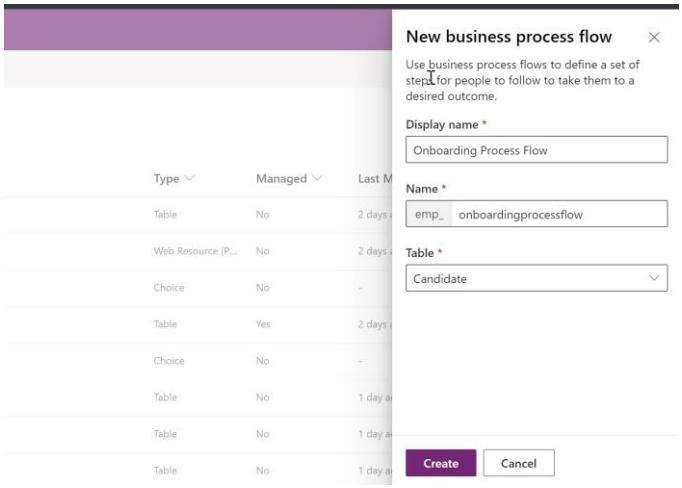


Create Business Process Flow

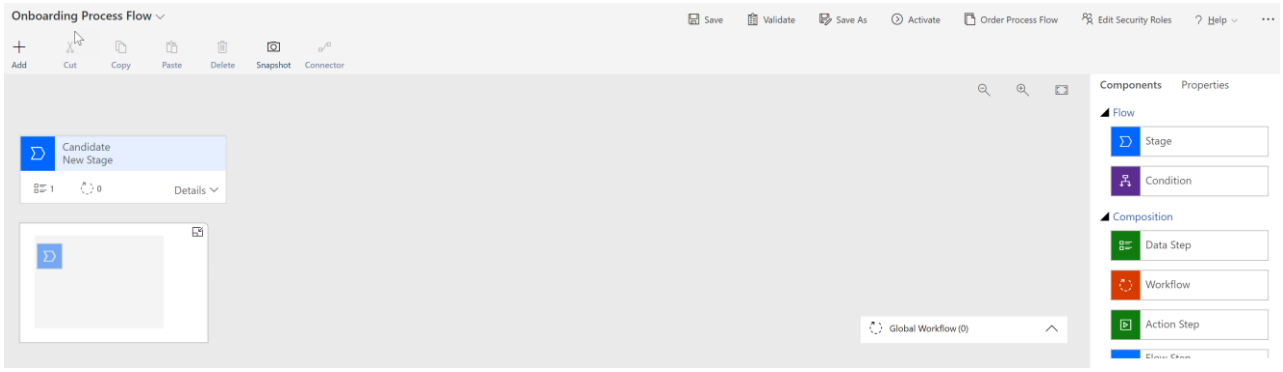
We will try to achieve this process flow in the solution by creating a business process flow. To do this, head over to the Solution and Select All-> New -> Automation -> Process -> Business process flow.



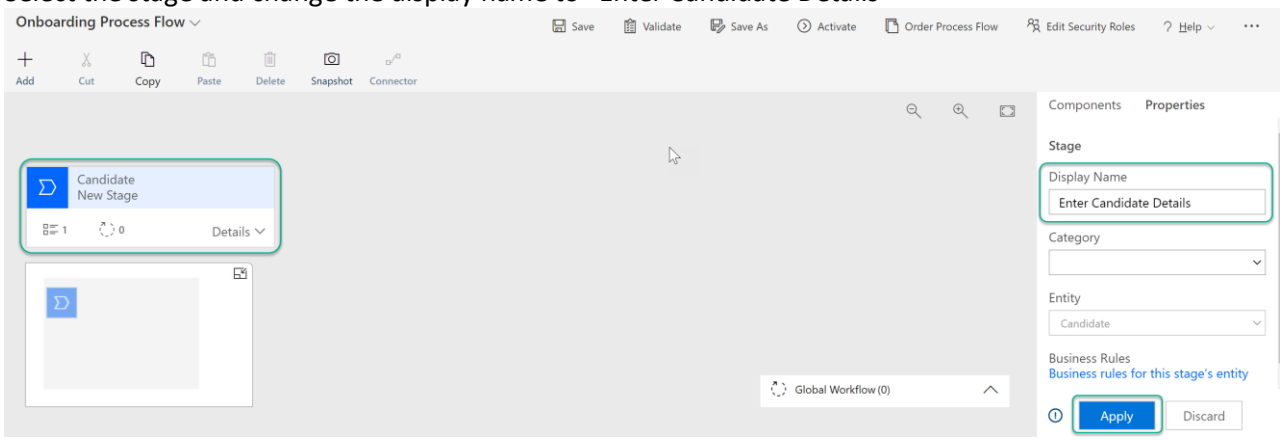
Let's name the flow as Onboarding Process flow and set the destination table on which the process flow will run as *Candidate* table



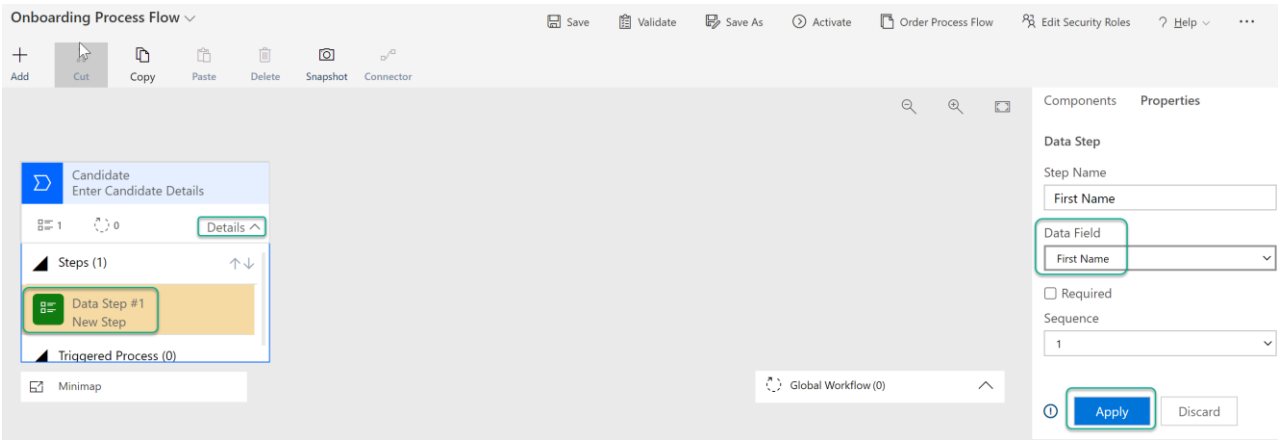
The default design view would look as below:



Select the stage and change the display name to “Enter Candidate Details”

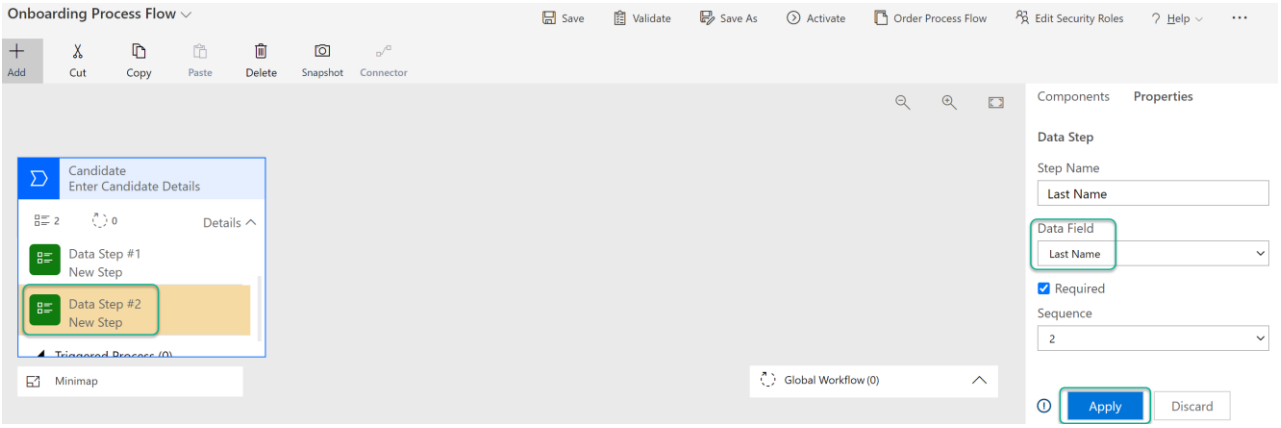


Expand the details section and select *Data Step #1* . In the properties section, select First Name from the Data Field Drop down and click on Apply

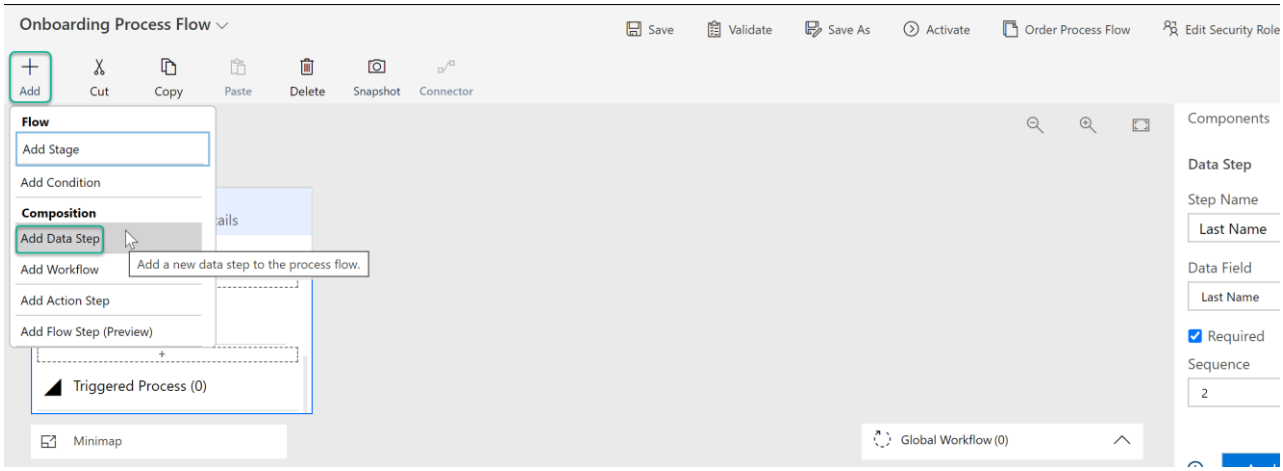


Lets add another data step, by selecting components tab and dragging and dropping Data Step to the Plus sign as below.

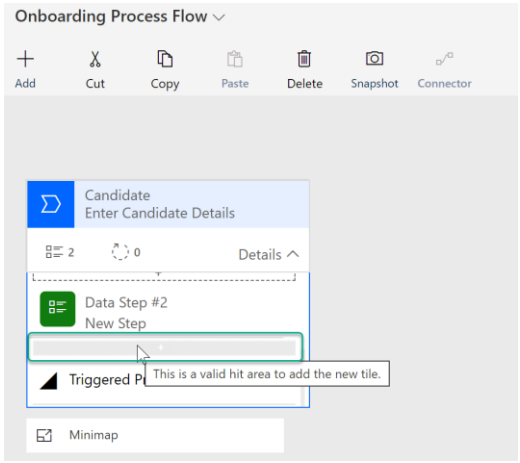
Select the newly added *Data Step #2* and update the Data Field in Properties to *Last Name*. Click on Apply.



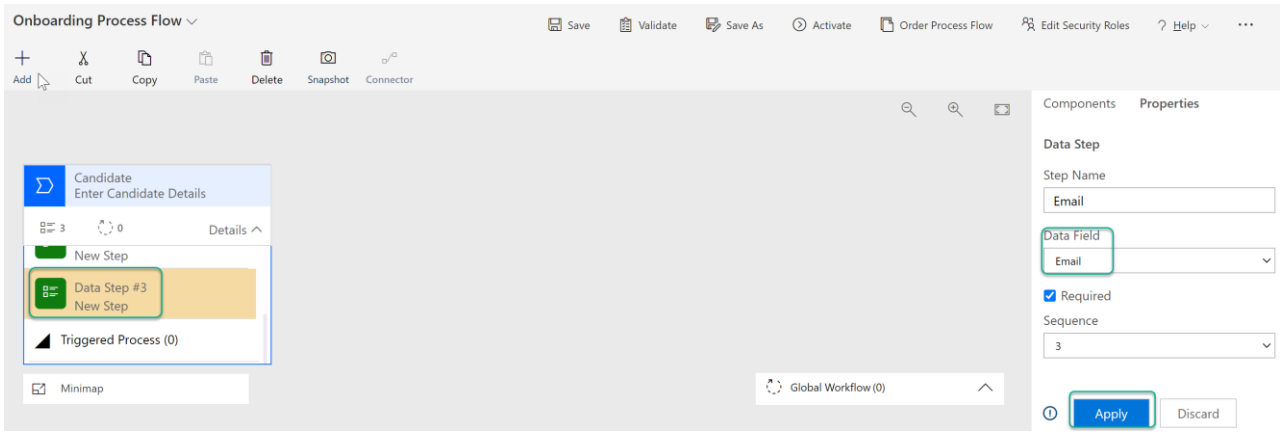
Lets add another Data Step. This time rather than selecting from components, lets explore another option. Click on Add and select “Add Data Step”



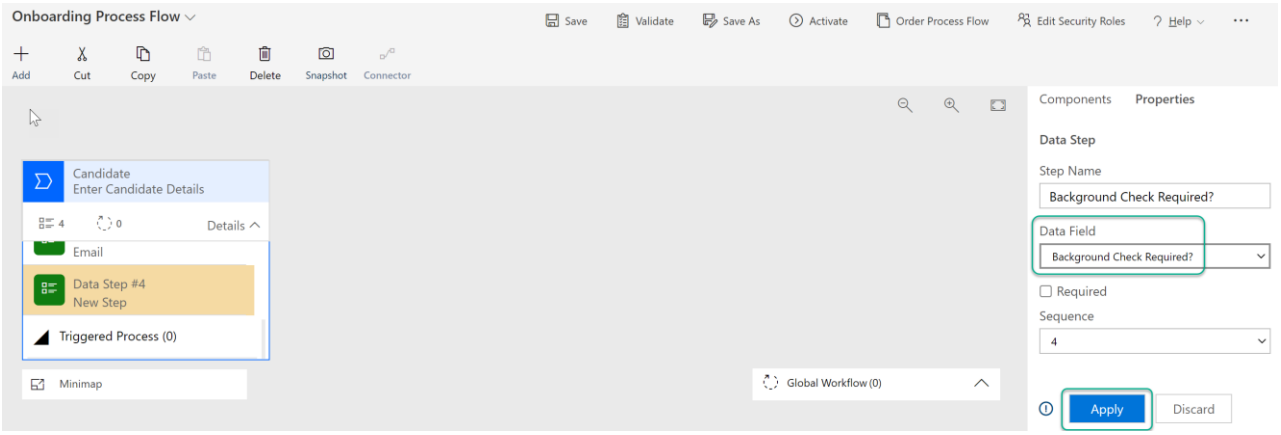
Below the Data Step #2, select the Plus icon, so that the data step will get dropped in this location.



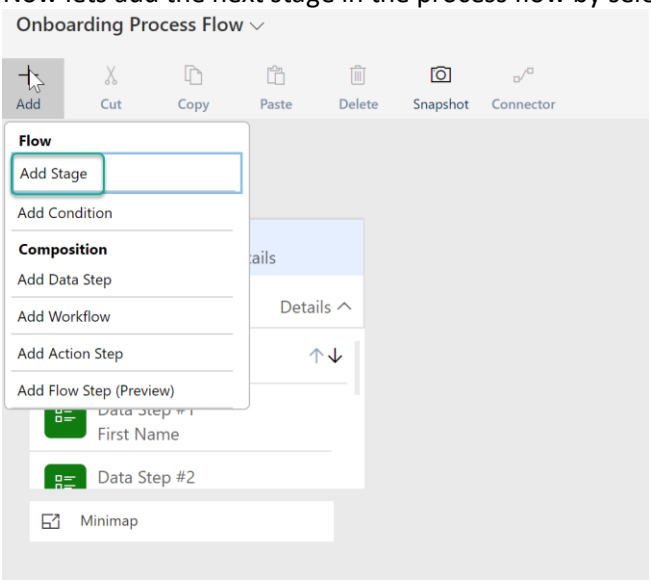
Select the data field and from the properties tab, Update the *Data Field* with the drop-down field value *Email.* Click on Apply.



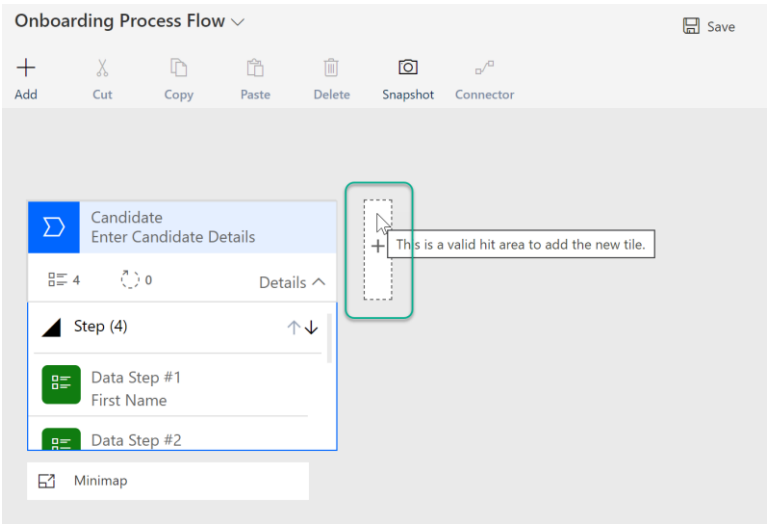
Add one more final field in this stage by dragging and dropping the data stage from the Components section. Click on Apply



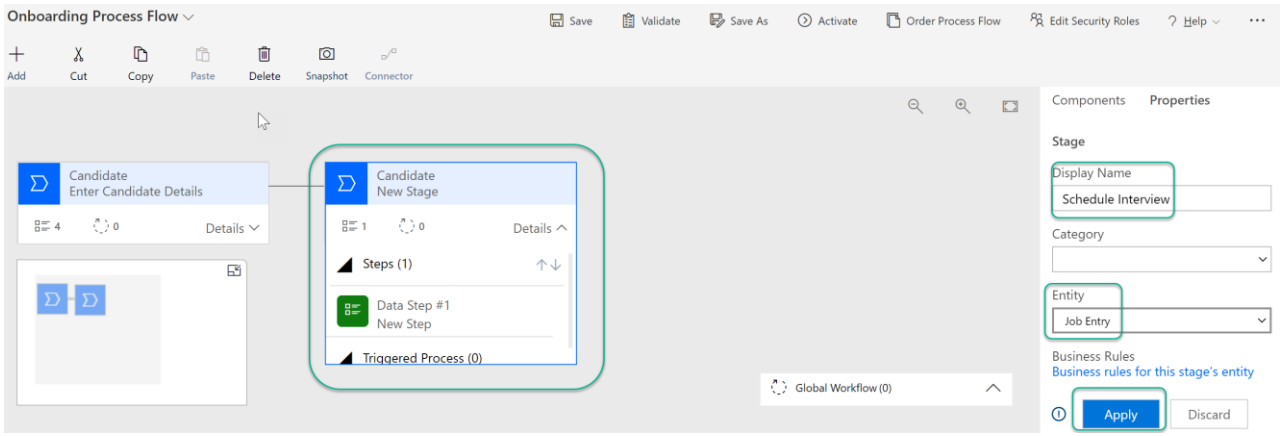
Now lets add the next stage in the process flow by selecting Add -> Add Stage



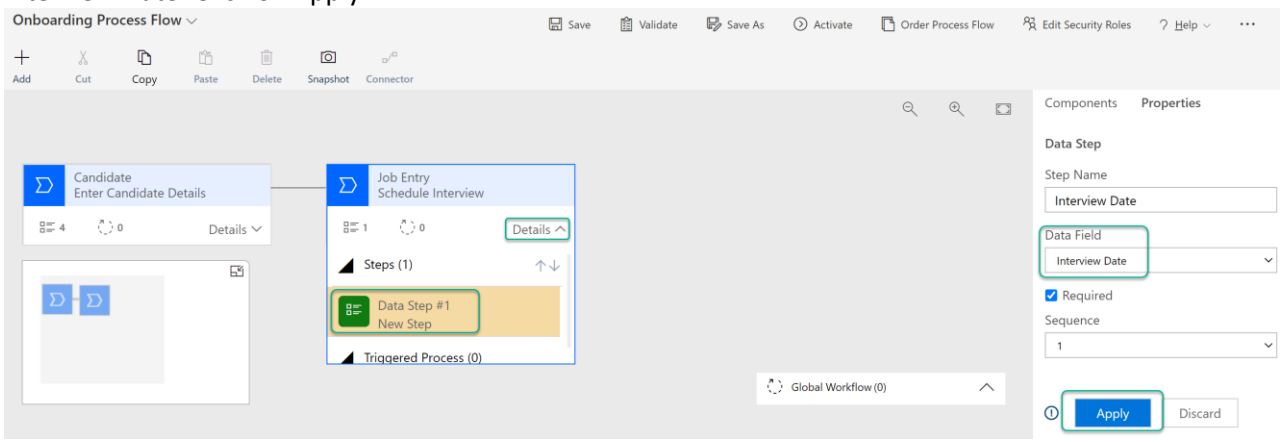
Click on the Plus sign to drop the newly selected stage on to the designer.



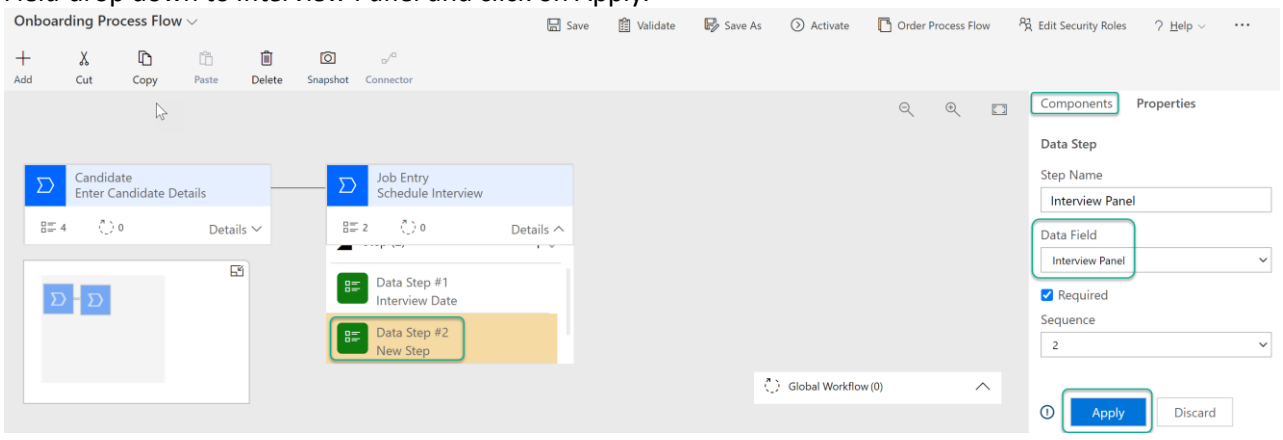
Select the newly added stage and change the display name to “Schedule Interview” and update the Entity field to “Job Entry” as we are updating back the data to this table. Click on Apply.



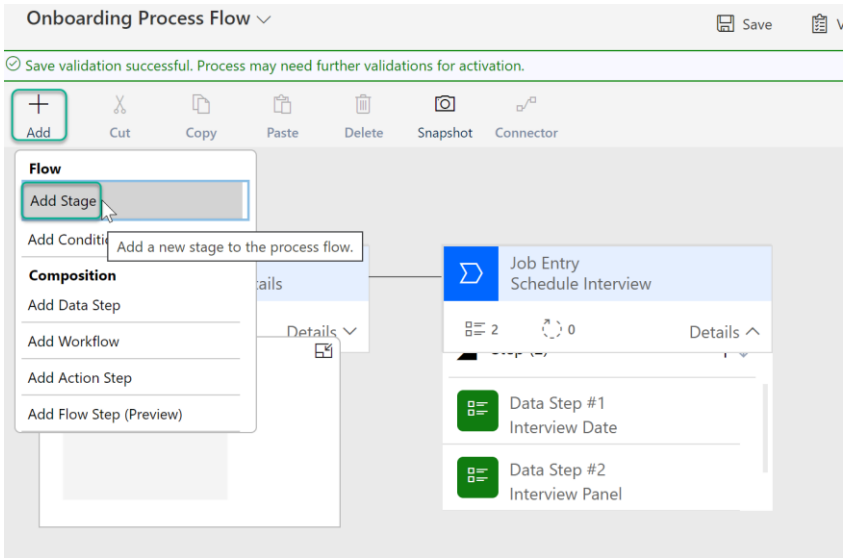
Expand the Details section, select *Data Step #1* and update the *Data Field* in the properties tab to *Interview Date*. Click on Apply



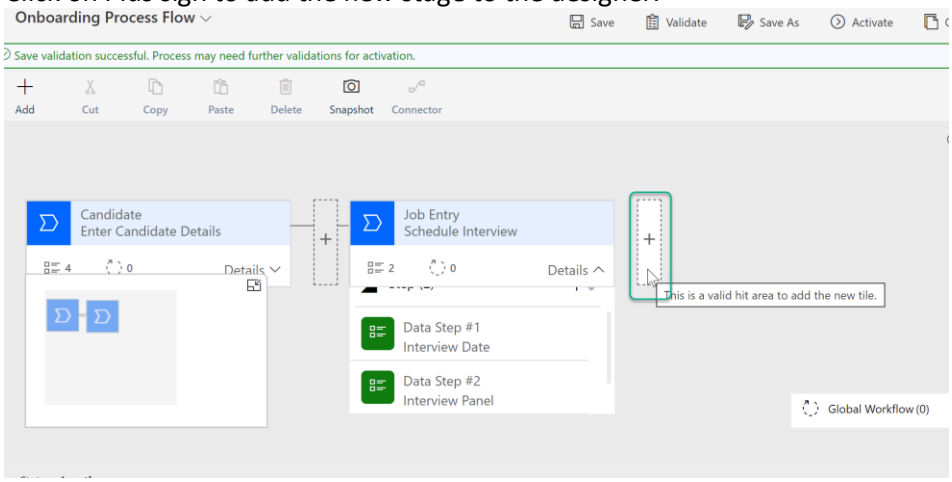
From components section, drag and drop Data Step to the stage. Select *Data Step #2* and select the *Data Field* drop down to *Interview Panel* and click on Apply.



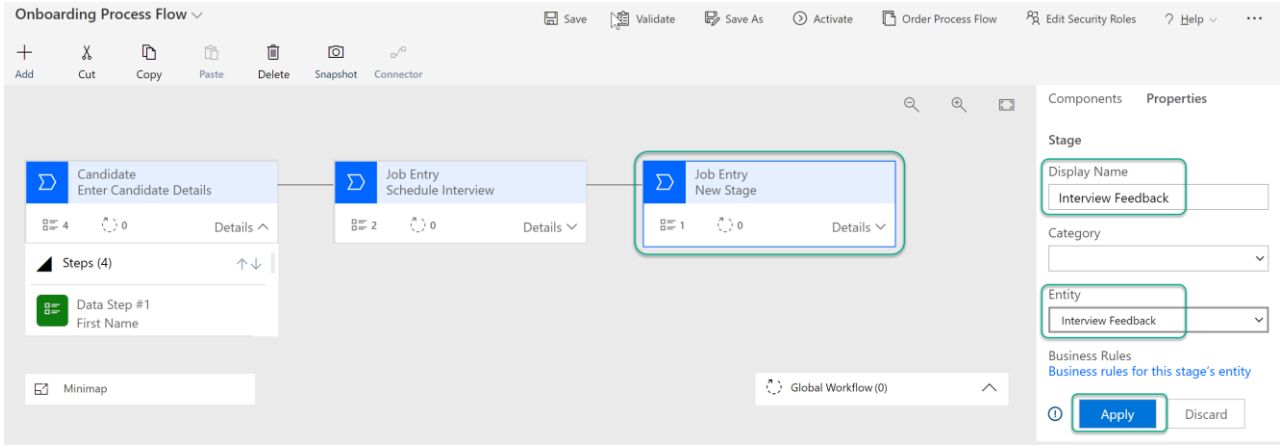
Now let's add another stage to the Flow designer by clicking on Add-> Add Stage



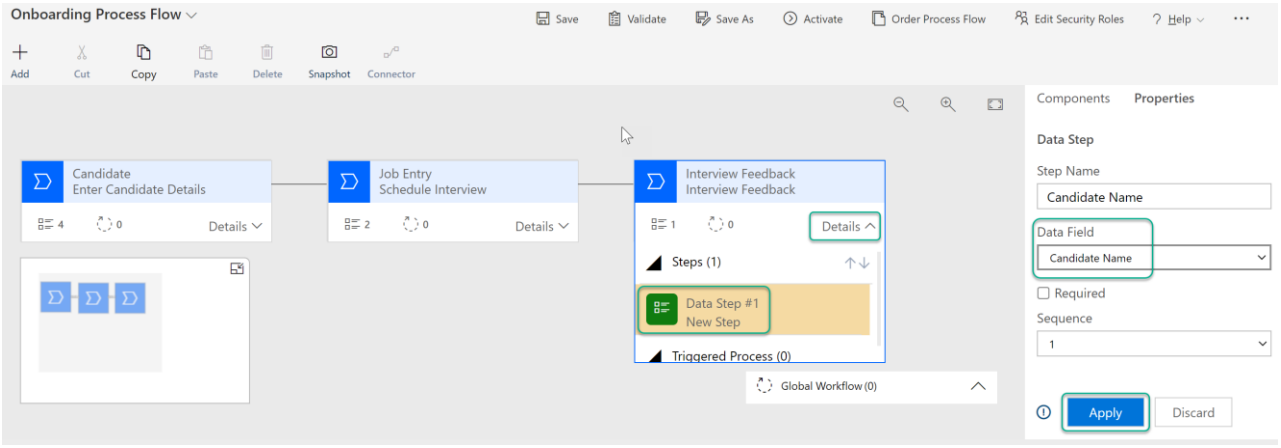
Click on Plus sign to add the new stage to the designer.



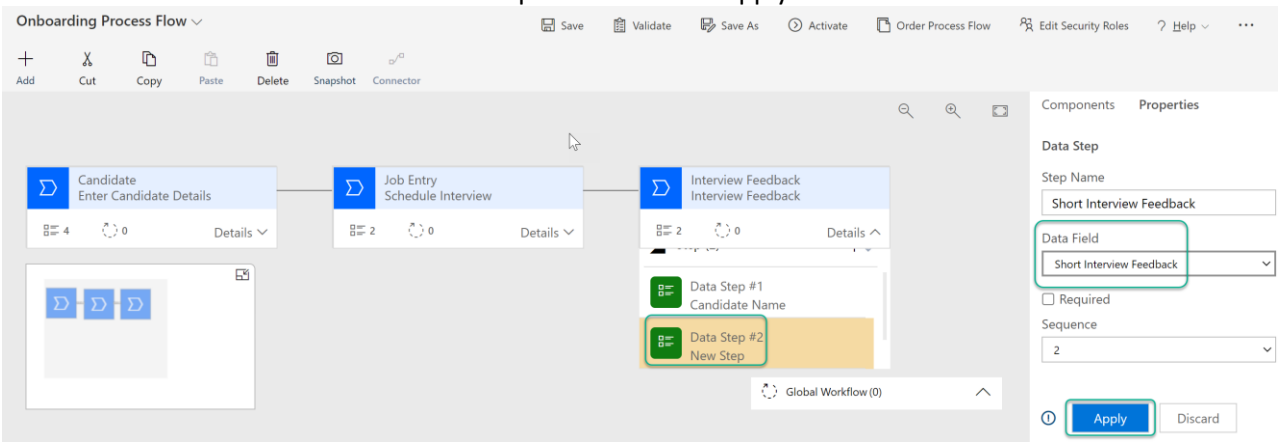
Select the newly added stage and from the properties tab change the Display Name to Interview Feedback and select the Entity(table) as Interview Feedback. Click on Apply



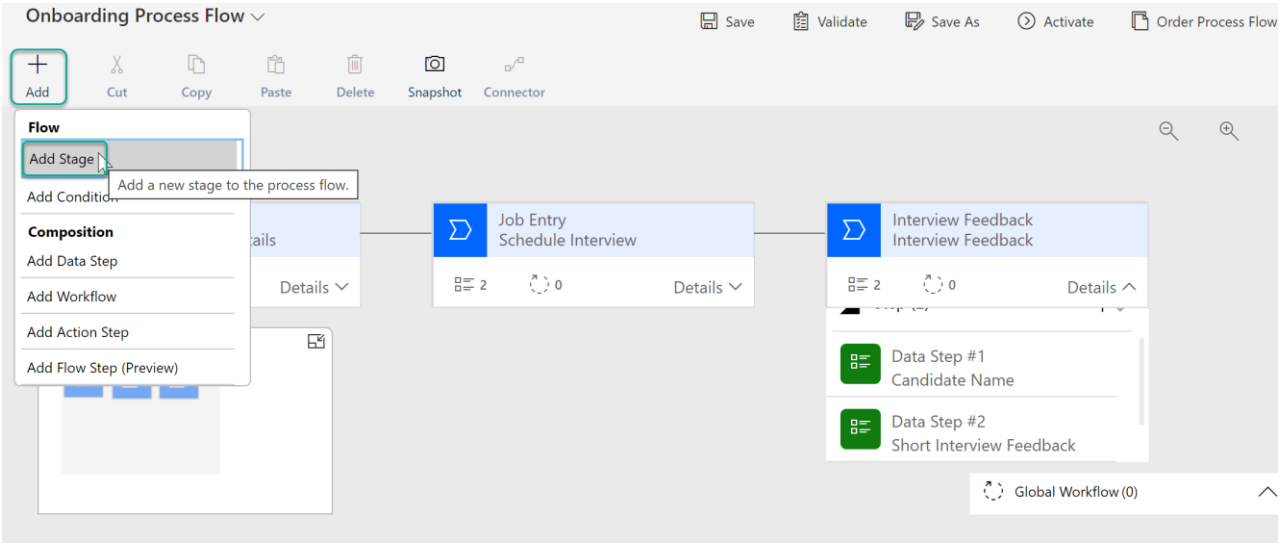
Now let's add the Data steps to the stage. Click on Details and Select Data Step #1. From the Data Field drop down, select Candidate Name. Click on Apply



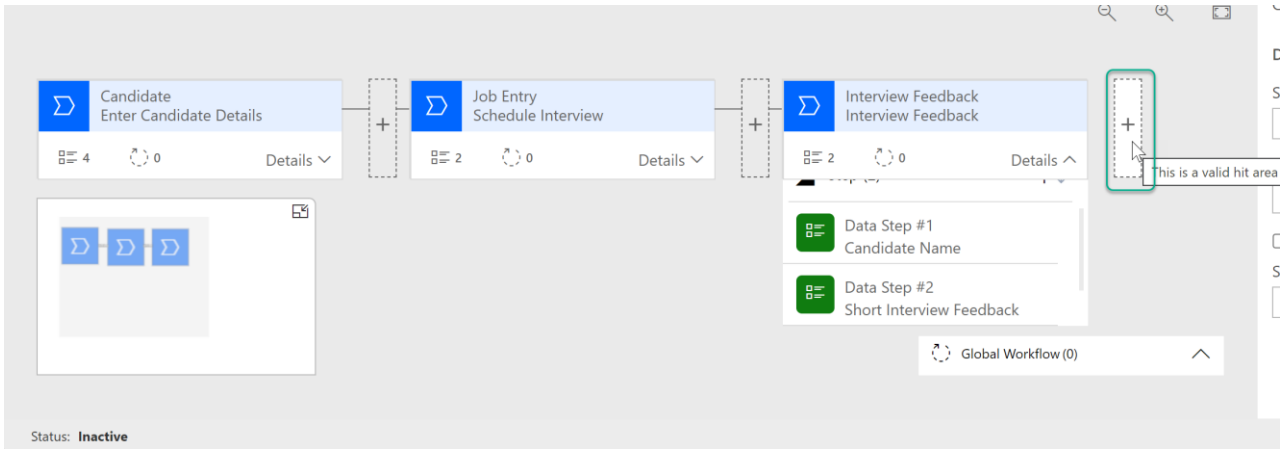
From Components tab, Drag and Drop Data Step to the stage. Select Data Step #2 and select *Short Interview Feedback* from the Data Field drop down. Click on Apply



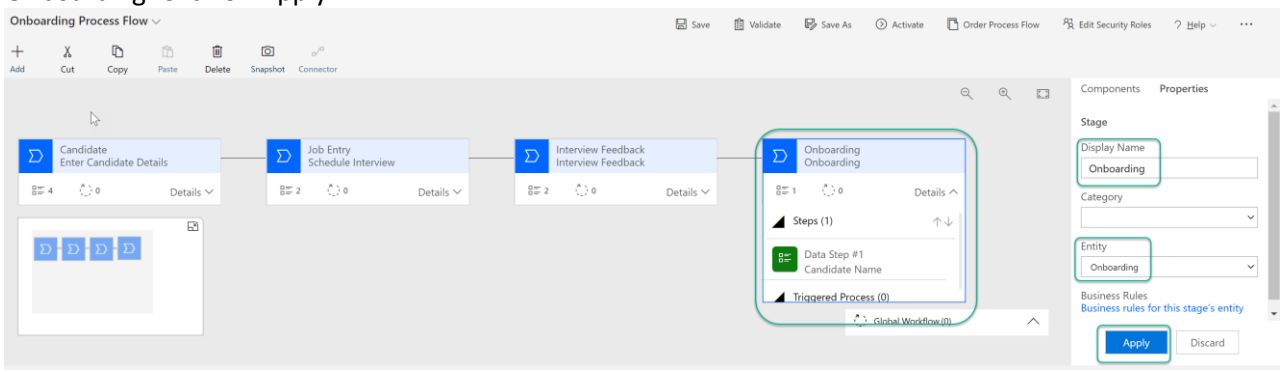
Let's add one last stage from Add -> Add Stage



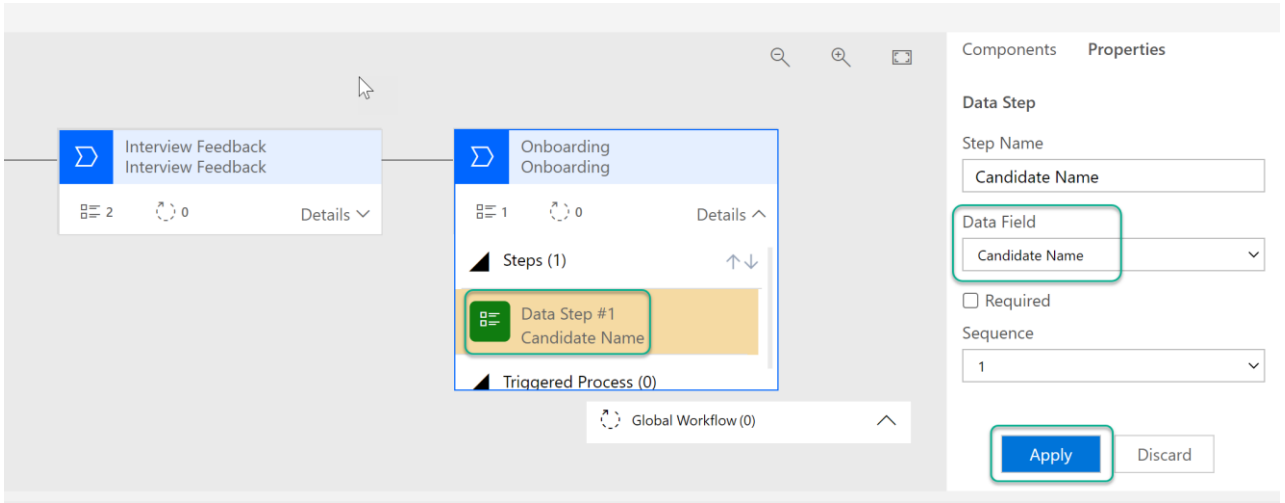
Click on Plus sign to add the Stage to the flow designer.



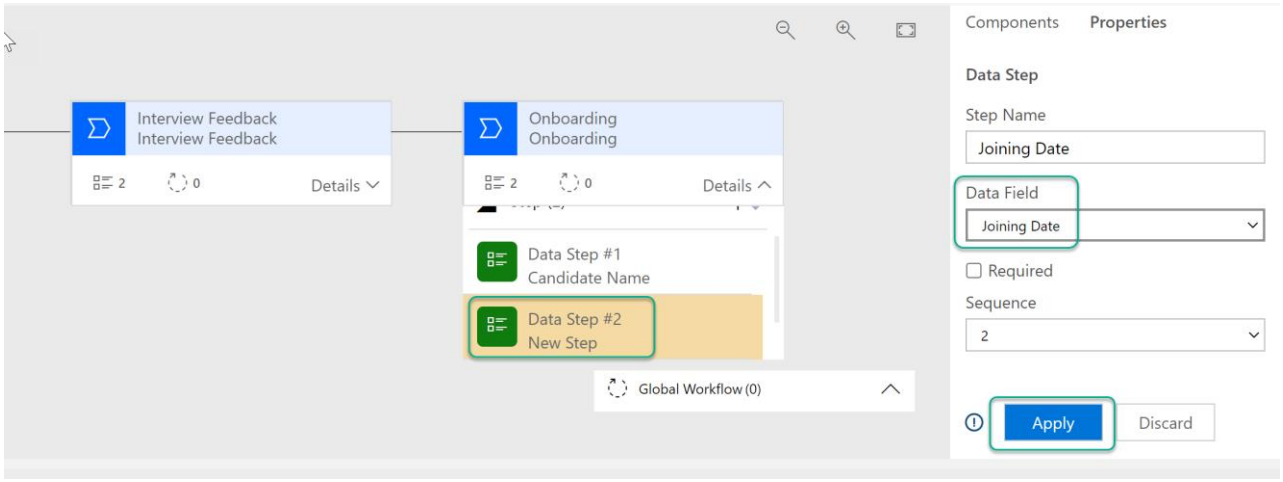
Select the stage and rename it to Onboarding and select the entity on which the updates will happen as Onboarding. Click on Apply



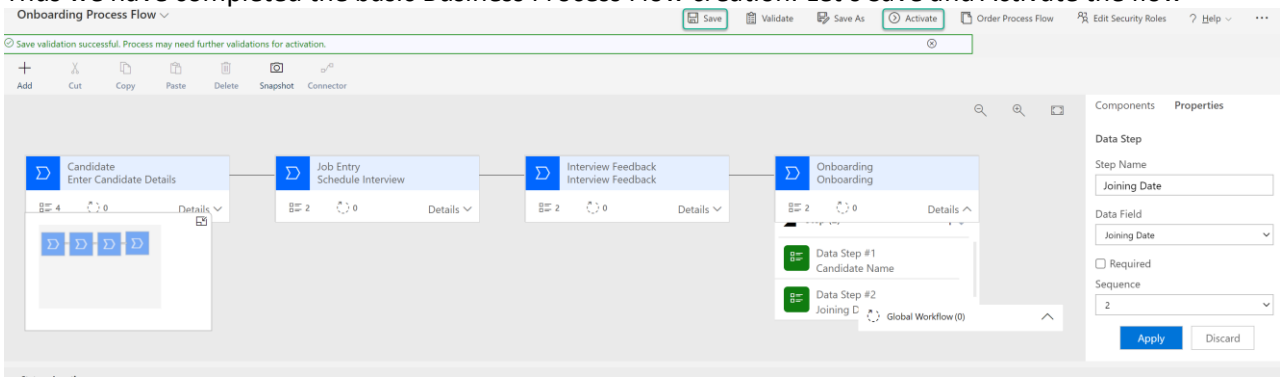
Now let's add the data steps to the stage. Select the default *Data Step #1* and Select the Data field drop down as *Candidate Name*. Click on Apply.



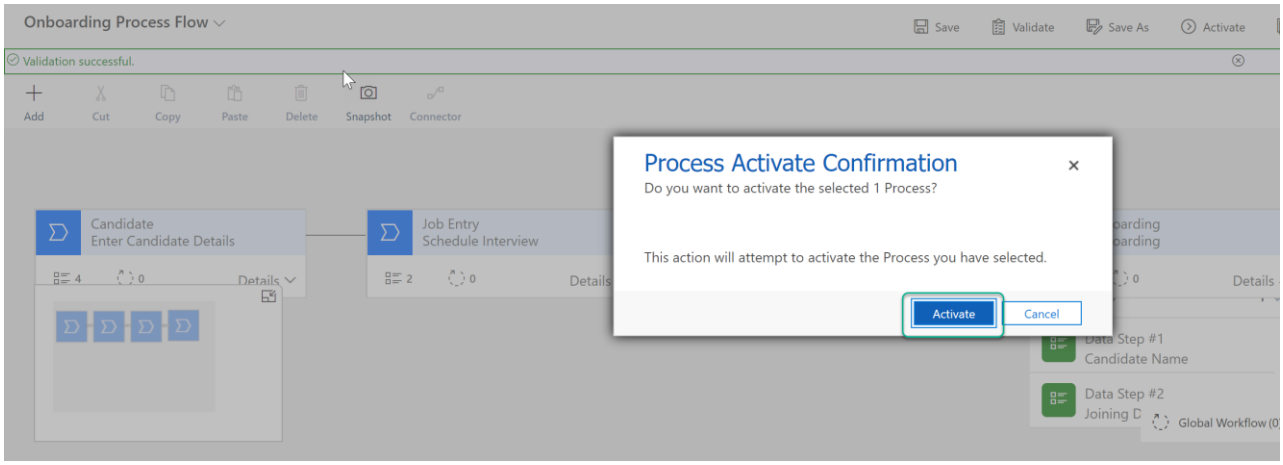
Lets add one more Data Step by dragging and dropping it from the Components tab. Select Data Step #2 and select the Joining Date column from the drop down. Click on Apply.



Thus we have completed the basic Business Process Flow creation. Let's Save and Activate the flow



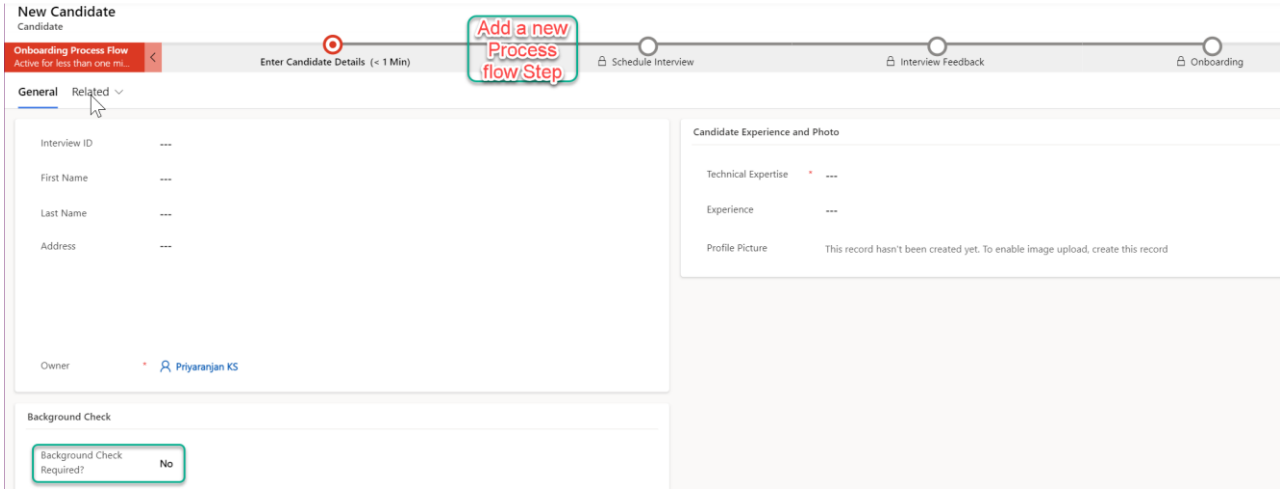
Click on Activate from the Pop up to complete the activation process.



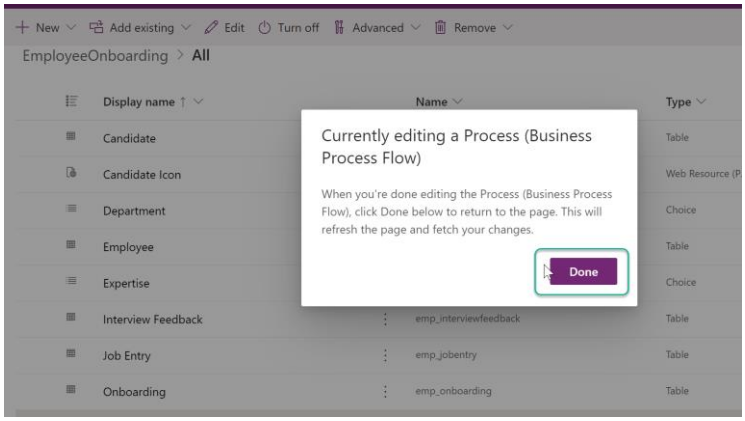
Call Power Automate from Business Process Flow

In our scenario, Once the candidate details are entered in the Form, we have an option to toggle the "Background Check Required ? " field to Yes/No . If the user toggles it to Yes, then we will initiate a Background check process via power automate where an internal employee will get the notification to do the manual check on the candidate.

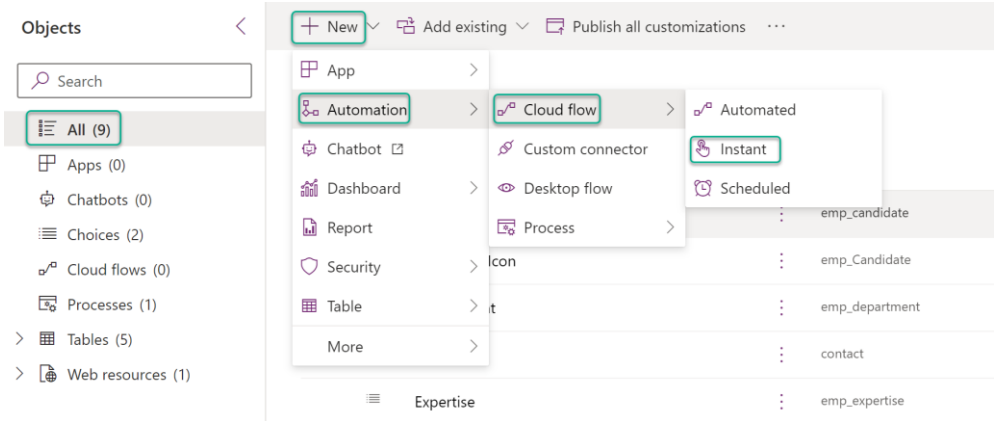
To do this , we will create a Power Automate and conditionally call it from the Business Process Flow



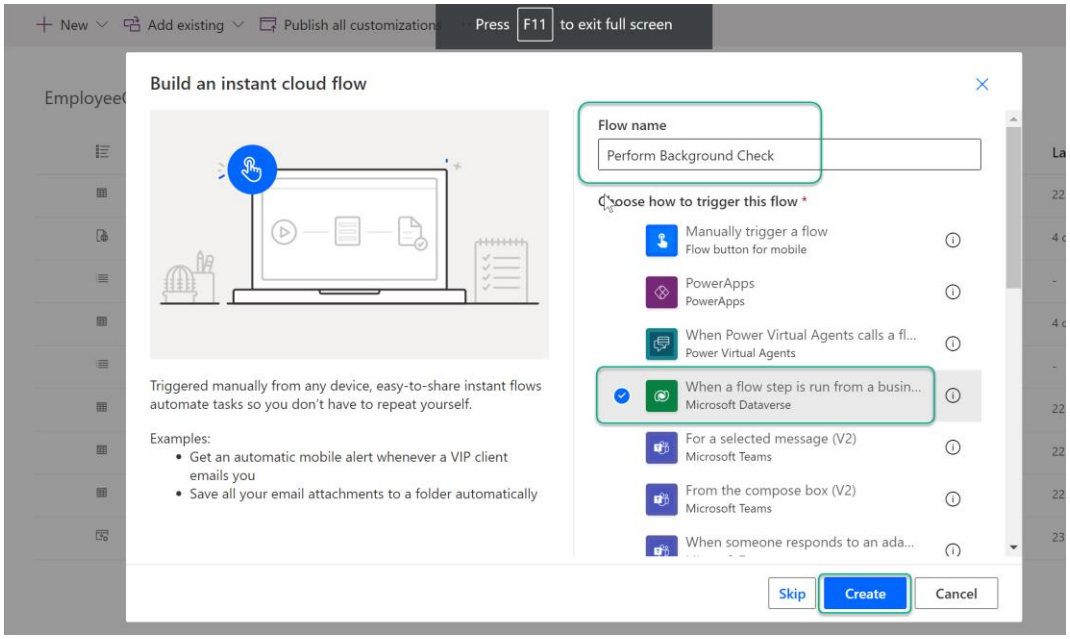
Let's head back to the solution and click on *Done* to indicate that we have completed editing the Business Process Flow.



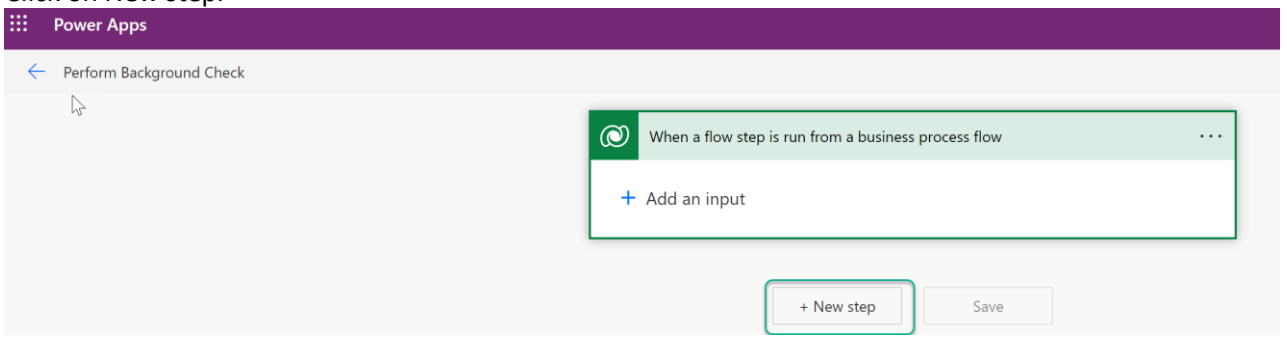
To add the Power Automate flow, Click on All - > New -> Automation -> Cloud flow -> Instant



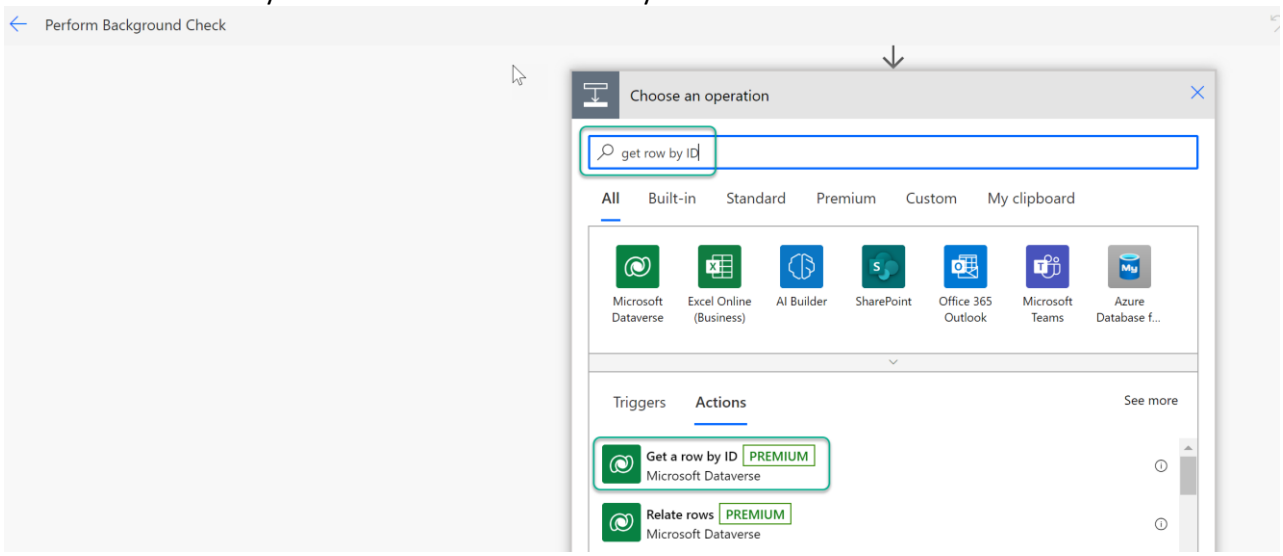
Specify the Flow name as "*Perform Background Check*" and select the trigger as "*When a flow step is run from a business process flow*" . Click on *Create*



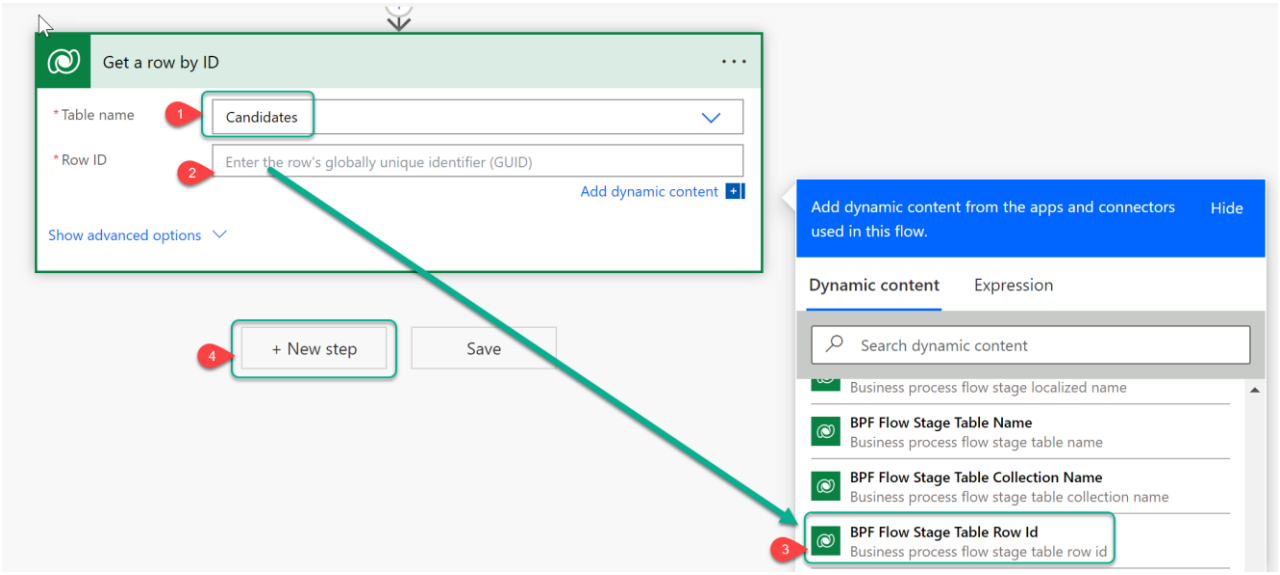
This will open the Power Automate Designer where we can add the actions corresponding to the trigger. Click on New step.



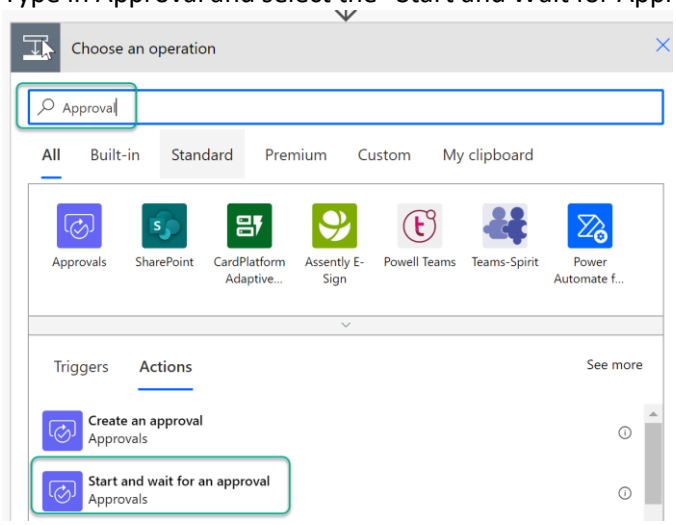
Search for "Get row by ID" and select the Get a row by ID action



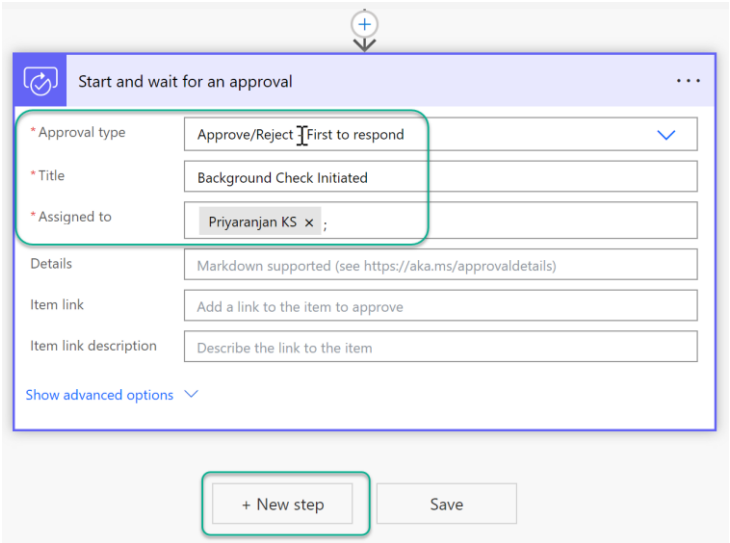
Select the table name as "Candidates" and for Row ID select the Dynamic content "BPF Flow Stage Table Row Id" . Click on *New step*



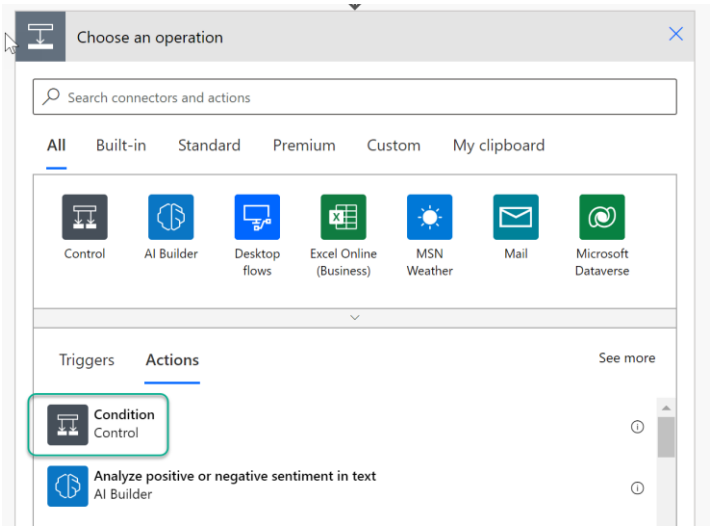
Type in Approval and select the “Start and Wait for Approval” action.



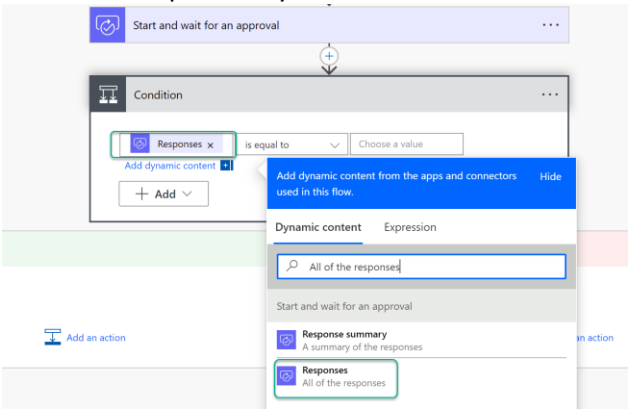
Update the Approval type, Title and Assigned to fields and click on New step.



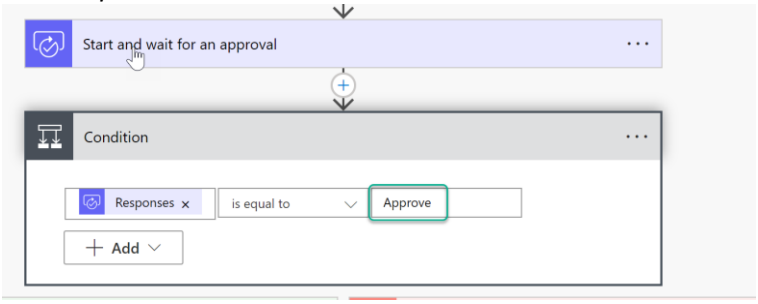
Select the *Condition* action from the Choose an Operation Block.



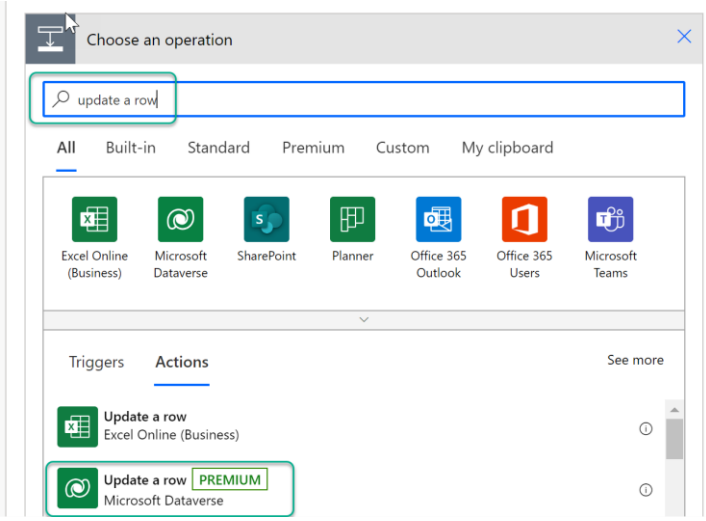
Select the Responses Dynamic Content in the first cell of the condition



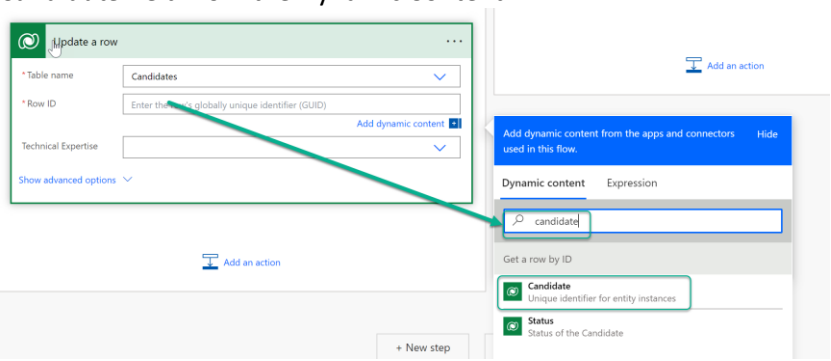
So as to evaluate the value returned from Responses field, add the text *Approve* to the third cell . Click on *New Step*



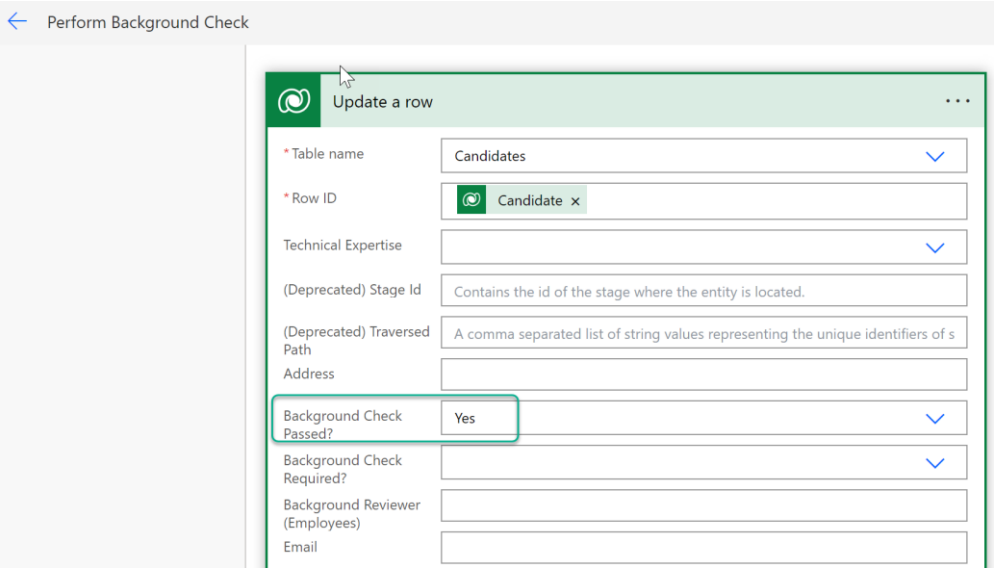
Select *Update a row* from the *Choose an operation* block.



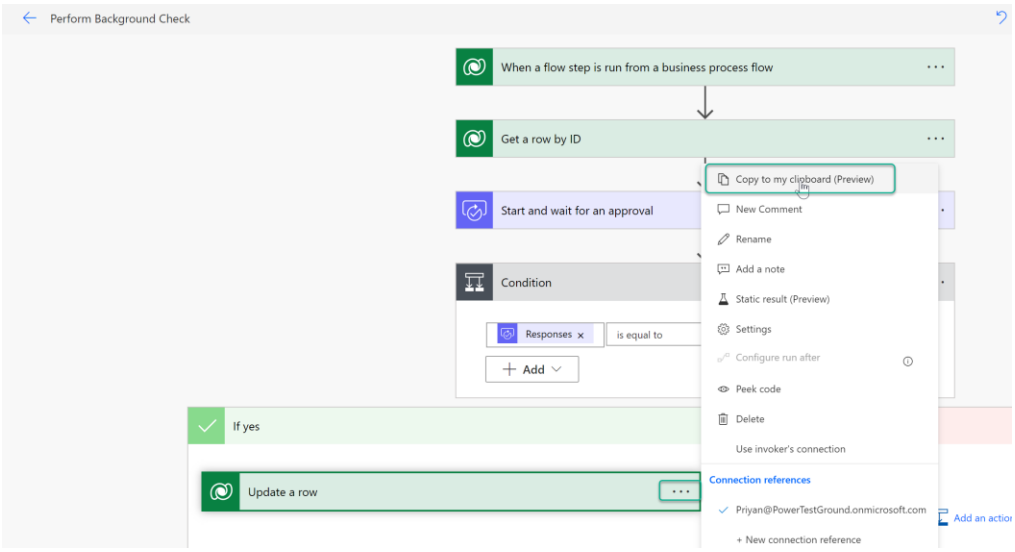
In the *Table name* field,select the table *Candidates* from the drop down. For the *Row ID*, select the *Candidate* field from the Dynamic Content.



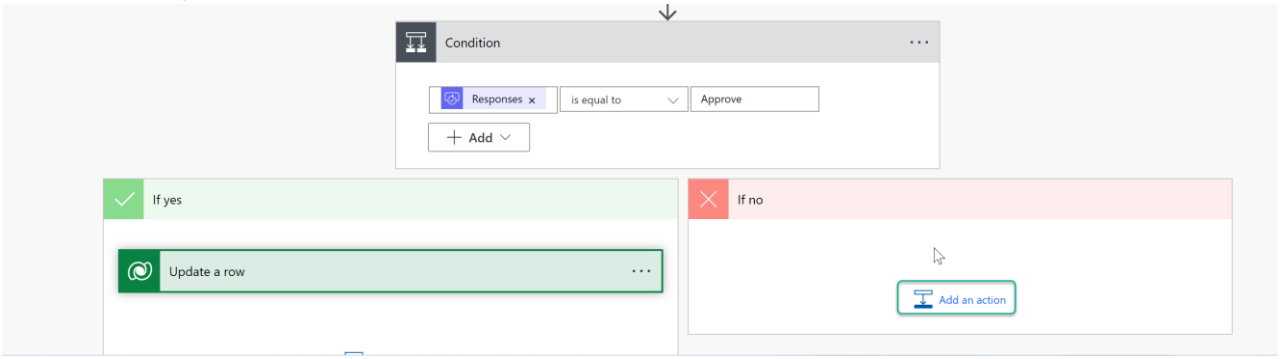
In the *Background Check Passed ?* field select the value as Yes/No based on the successful completion of the background check.



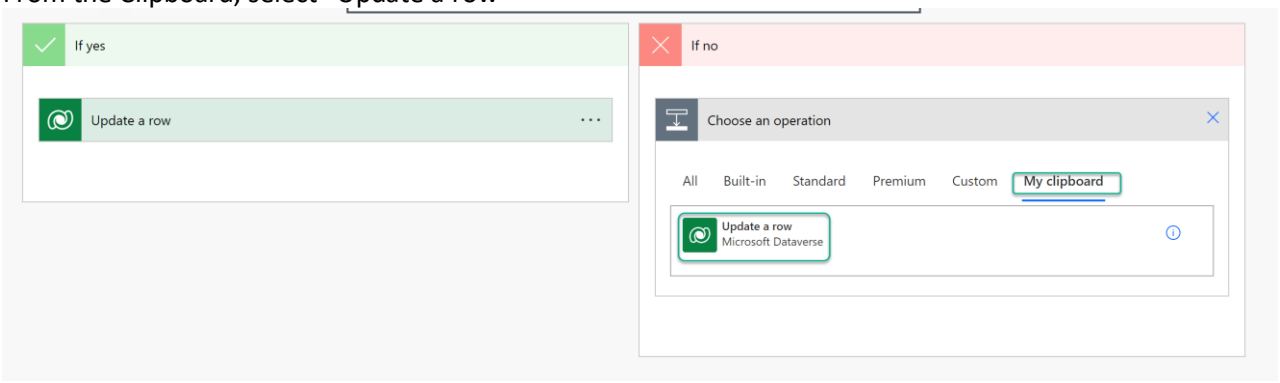
So as to add the opposite row update in the No block, lets copy the “Update a row” to the clipboard by selecting the 3 dots and select *Copy to my clipboard*



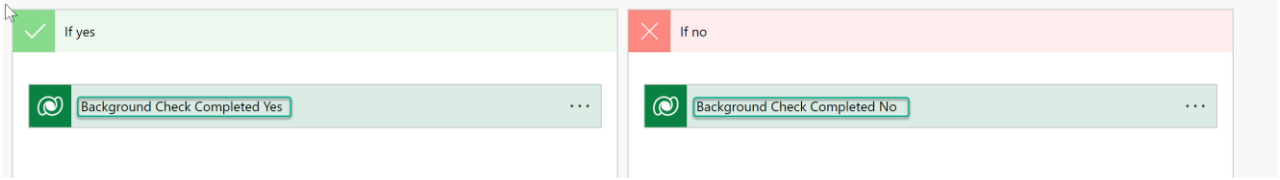
In the No block, click on *Add an action.*



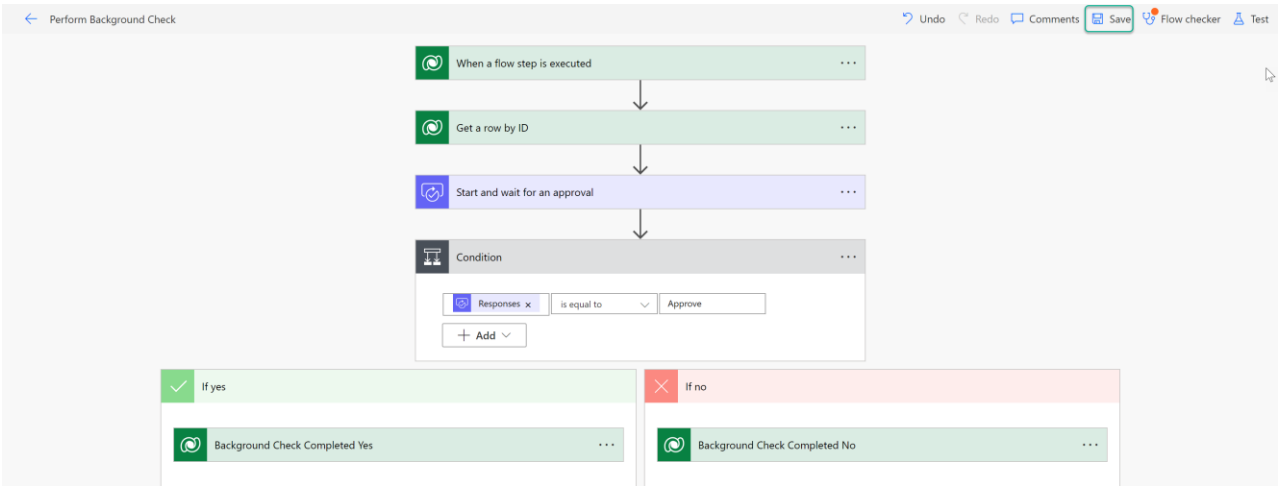
From the Clipboard, select “Update a row”



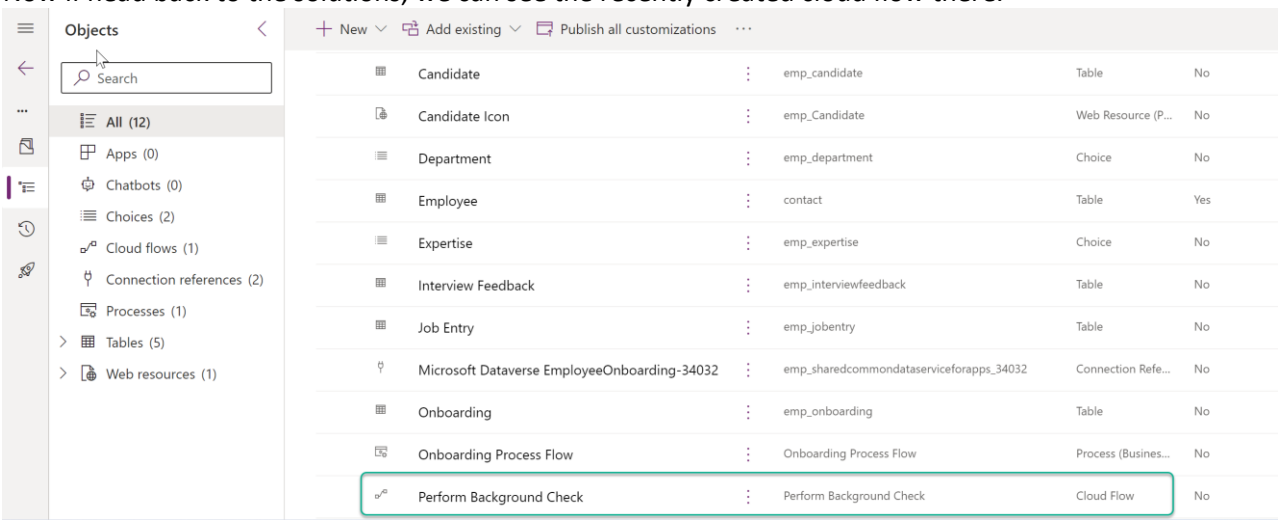
Rename the Dataverse row update actions by clicking on the 3 dots and selecting rename



Thus, the completed flow will look like below. Click on Save.



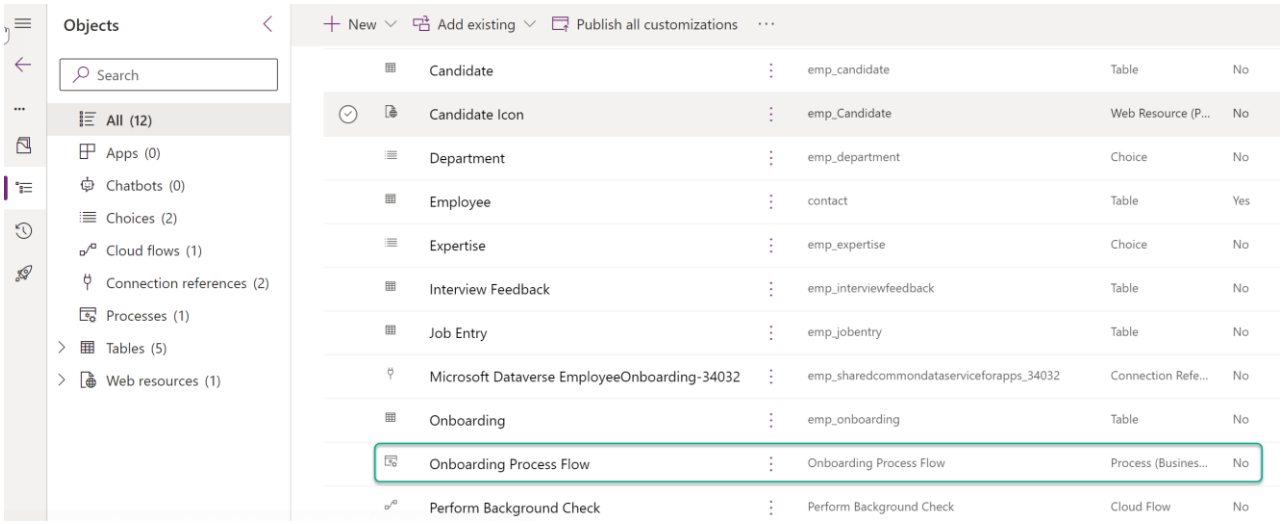
Now if head back to the solutions, we can see the recently created cloud flow there.



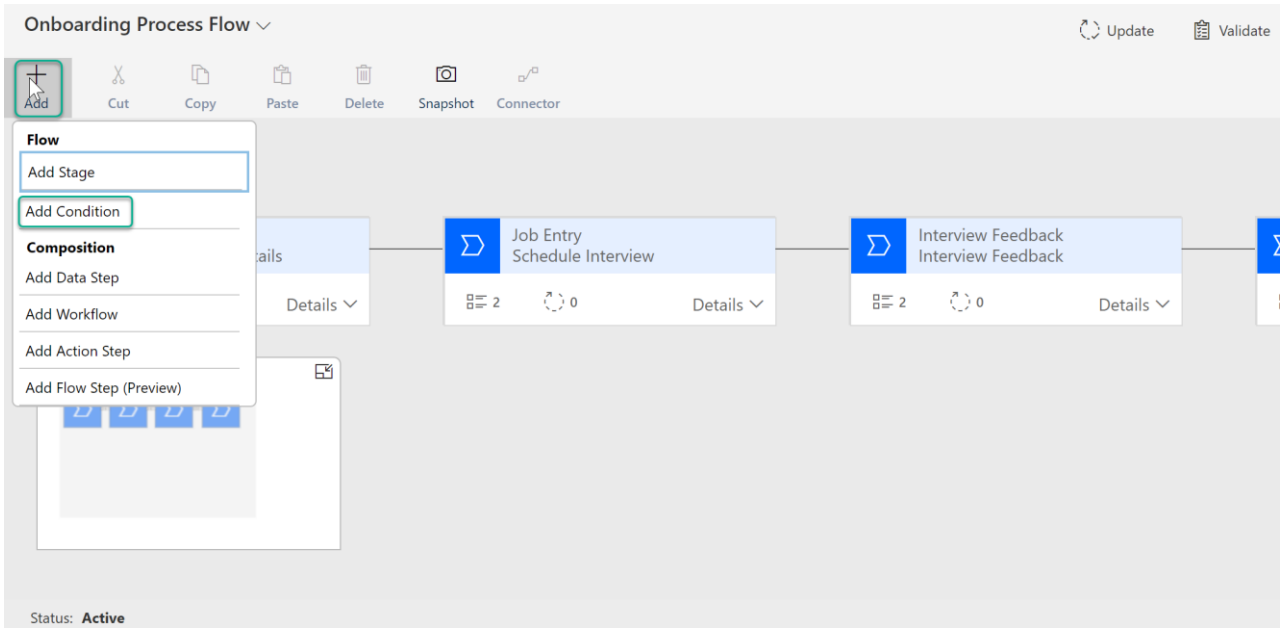
Update the Business Process Flow with Approval

Now we will update the existing business process flow to include the back ground check Power Automate so that whenever while filling the Candidate form, if the Background Check Required ? column is set to Yes, the Business Process flow will show the additional stage to call the Power Automate

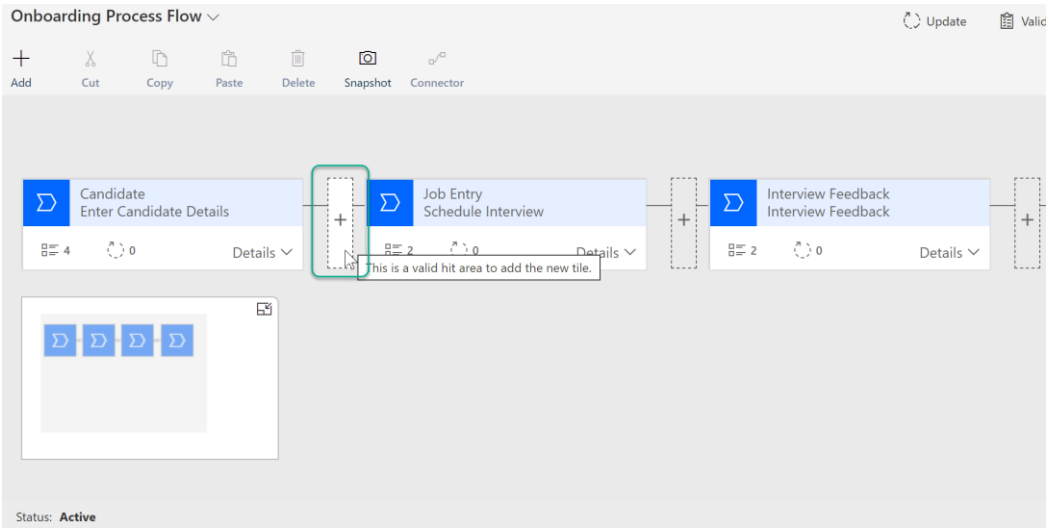
To do this, lets open the previously created Business Process Flow by clicking on *Onboarding Process Flow*



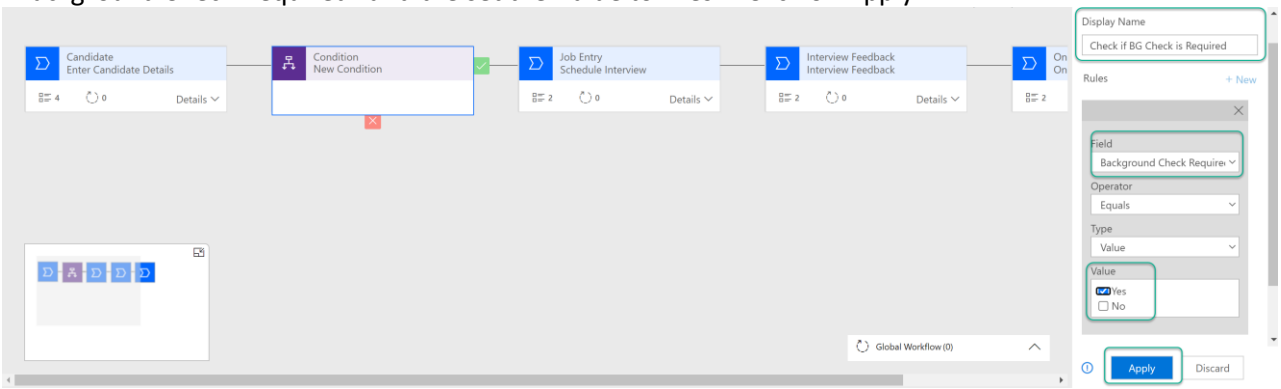
Click on Add -> Add Condition to drop the condition block on to the designer.



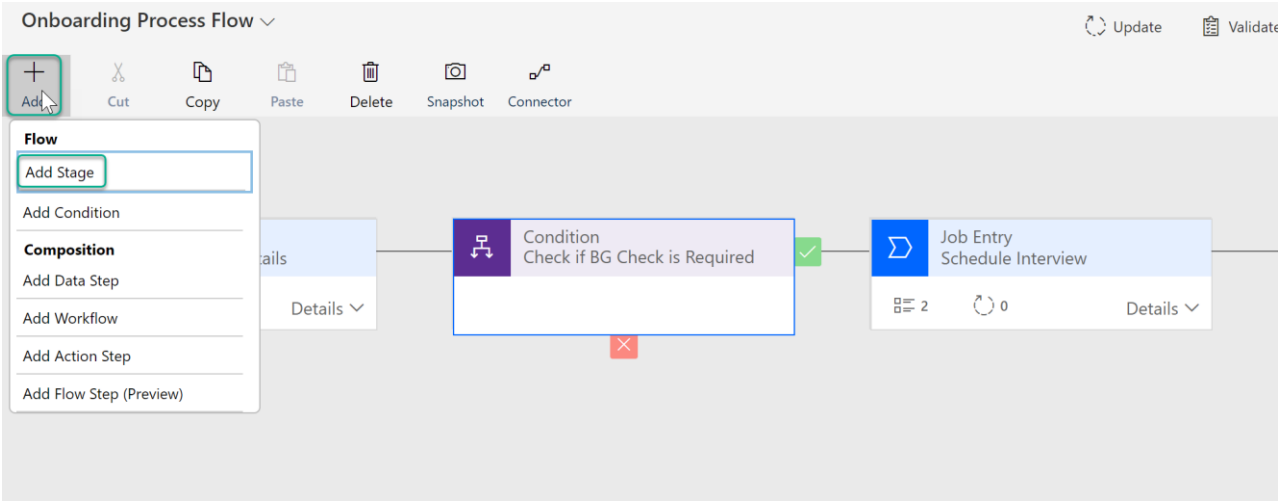
Click on the Plus Sign to drop the Condition action on to the form before the Job Entry stage.



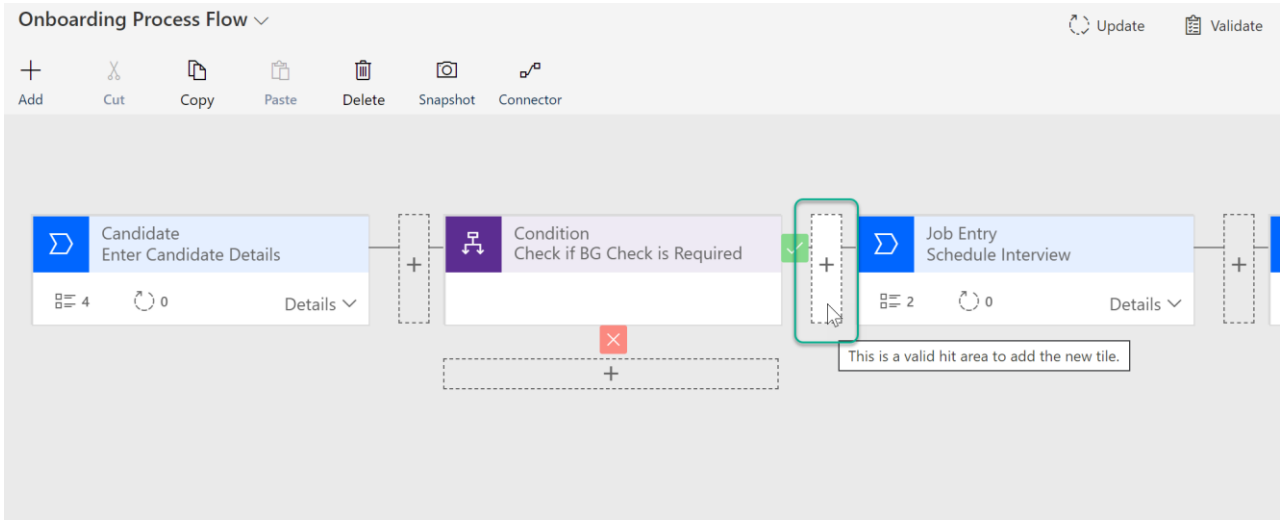
Select the Condition and rename the Display Name to “Check if BG is Required” and select the field to “Background Check Required” and the set the Value to “Yes” . Click on Apply.



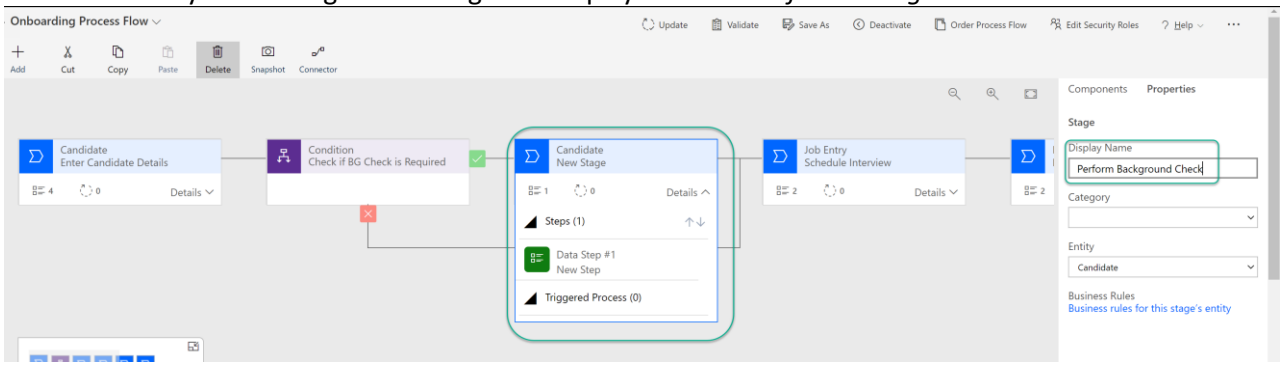
Now lets add a new stage that will be run if the condition evaluates to Yes. Click on Add -> Add Stage



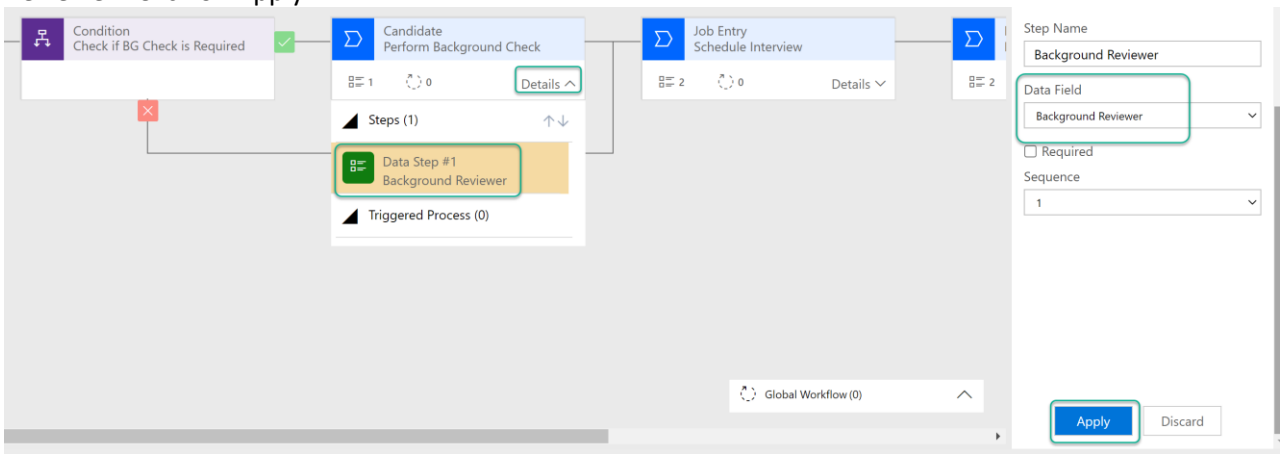
Select the Plus sign to add the new stage.



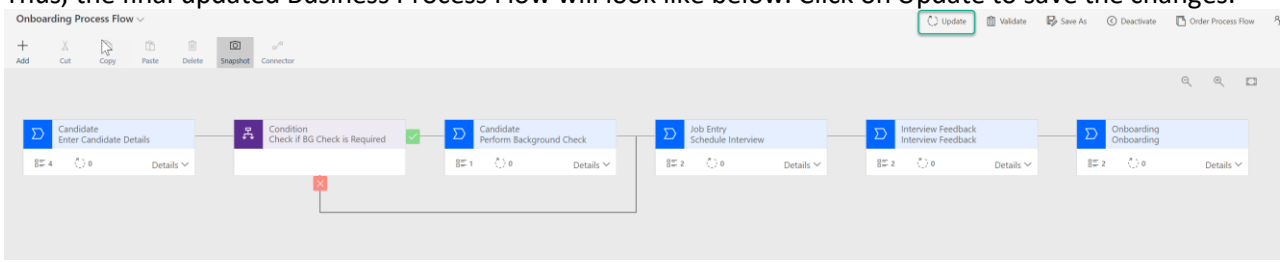
Select the newly added stage and change the Display name to *Perform background check*



Expand the Details section of the stage and select *Data Step #1* and select the Data Field as Background Reviewer . Click on Apply



Thus, the final updated Business Process Flow will look like below. Click on Update to save the changes.



Quick View Form

Quick View forms helps us bring in related data from other tables on to the main form for an unrelated table so that we have the wholistic data at a single place. This also reduces the need to navigate away from the main form of a table's row to get related data from another table's main form. So in general, Quick View Form helps in giving a summary view of the related data at a single place.

In our sample, in the Onboarding Table's Main Form, we have the Onboarding information of the candidate who has completed the Interview process successfully. To give the wholistic data related to the candidate,

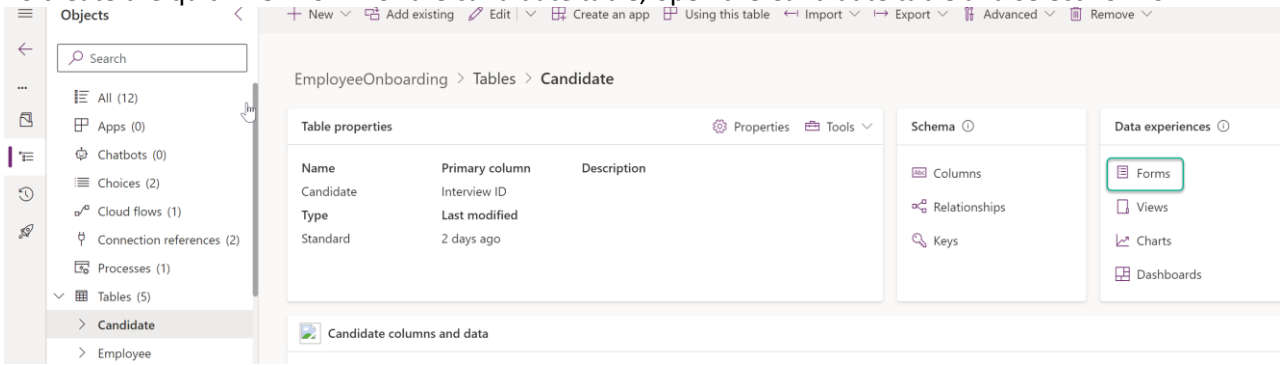
- We will create a Quick View Form that shows further details about the candidate by picking information from the Candidate table.

Finally we will show them on the Main form of the Onboarding Table so that we have complete information about the candidate.

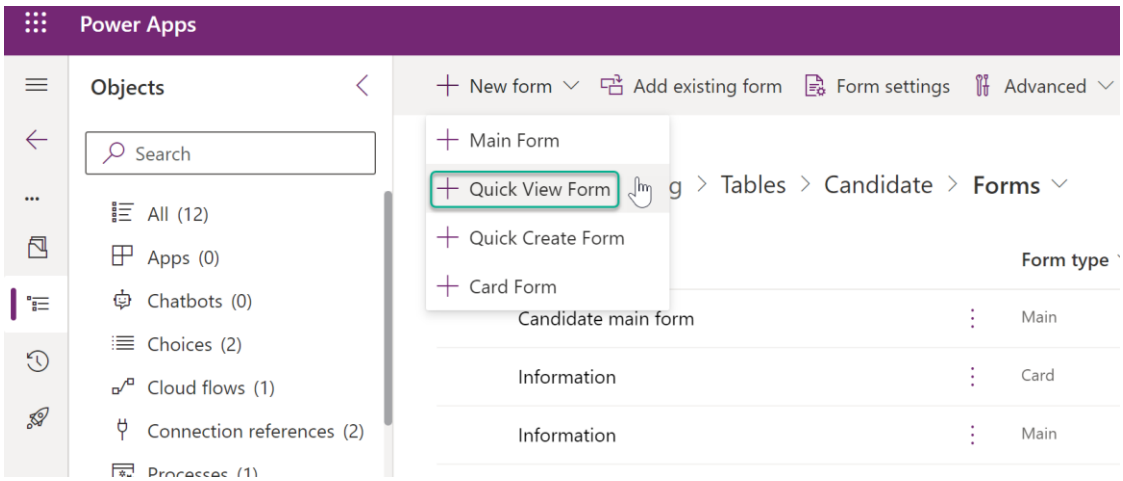
To create the Quick View Forms, lets head over to the respective tables.

Quick View Form for Candidate Table

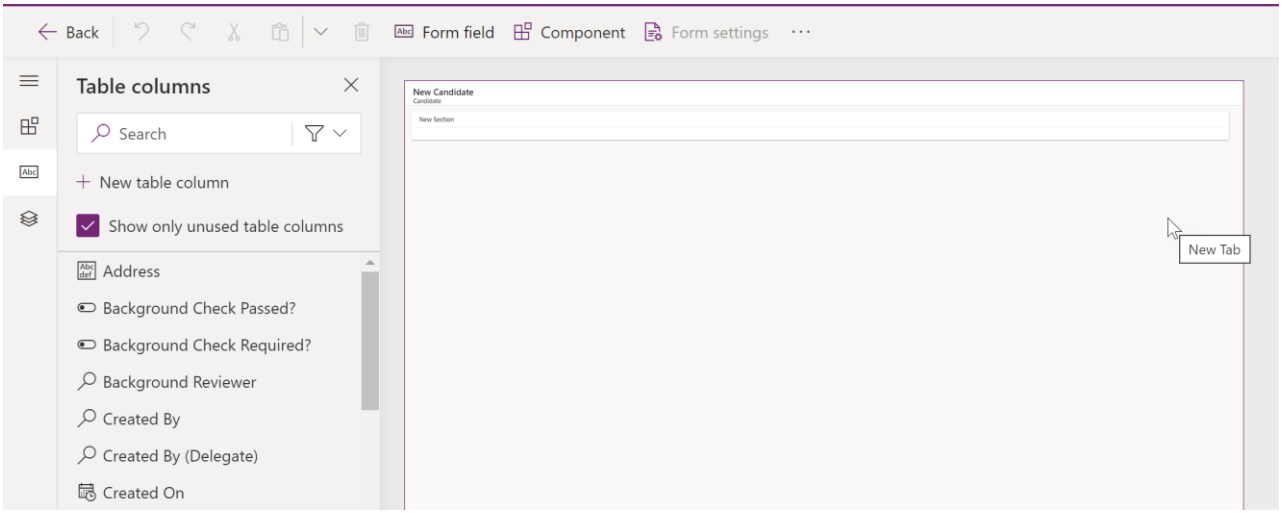
To create the quick view form for the candidate table, open the Candidate table and select Forms.



Select New form -> Quick View Form



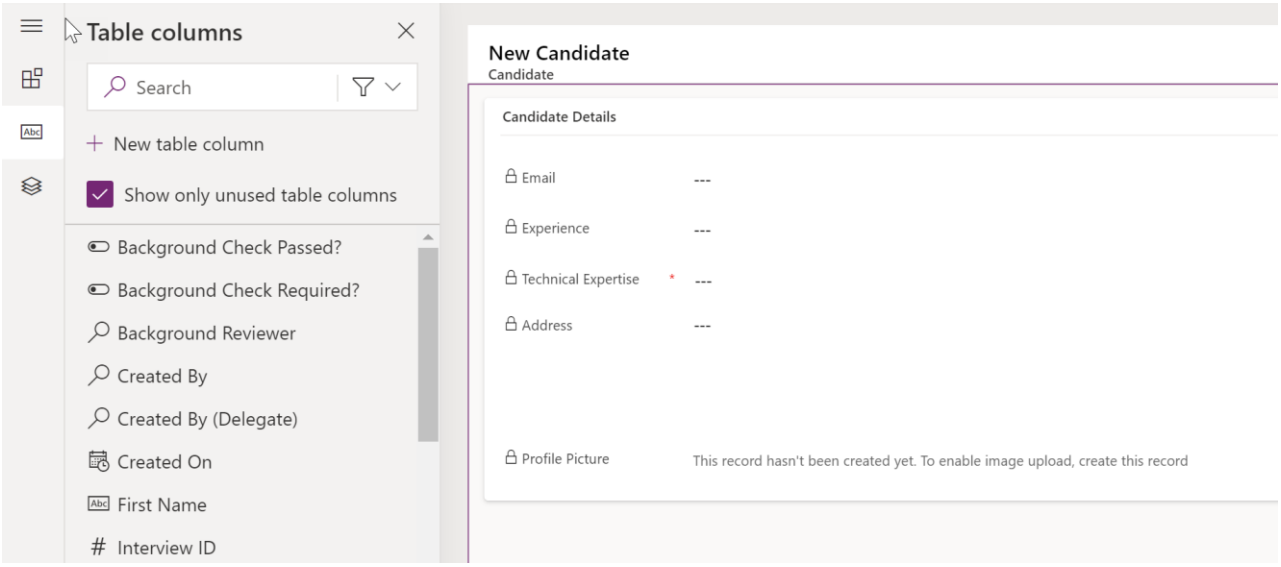
The default quick view design will look like below



Let's rename the section as *Candidate Details*.

We will add the needed columns to the new Section by dragging and dropping the columns to the form.

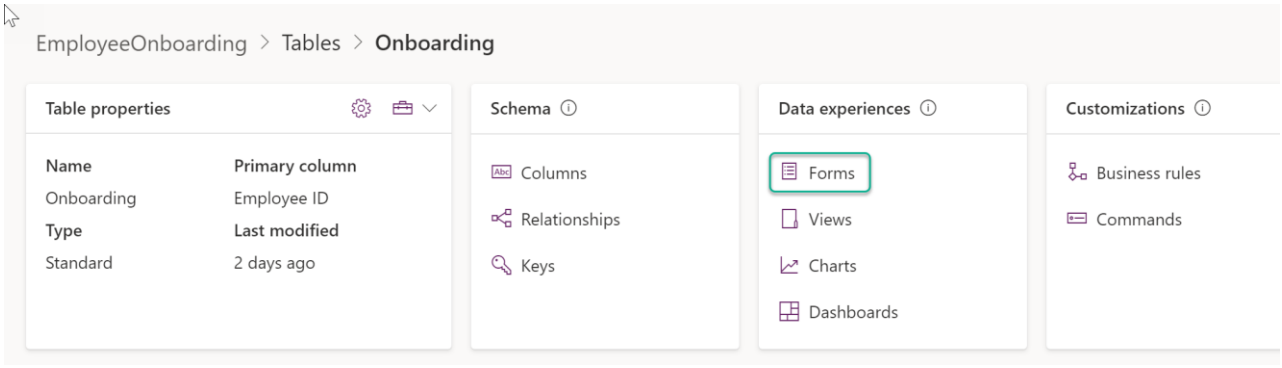
Once done with the column addition, click on Save.



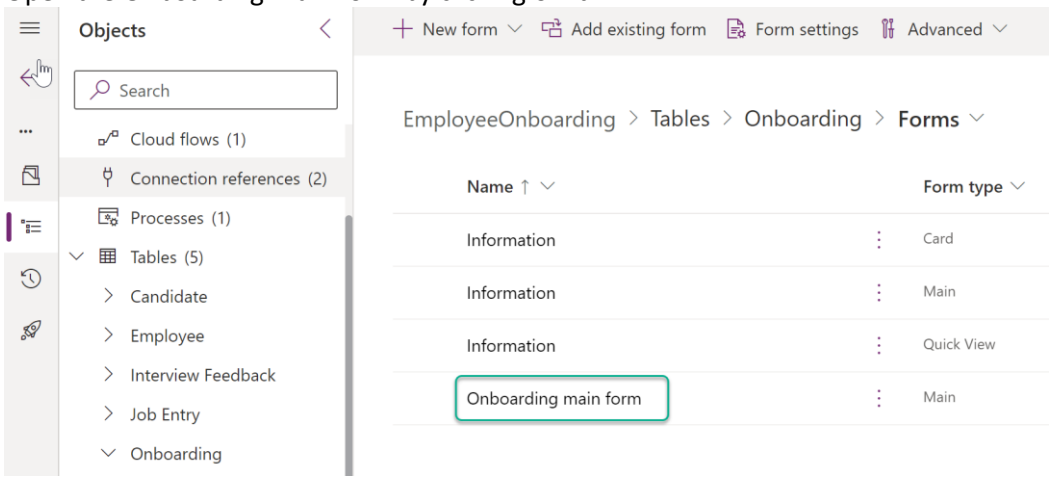
Thus we have created the Quick View Form for Candidate table.

Add Candidate Quick View Form to Onboarding Form

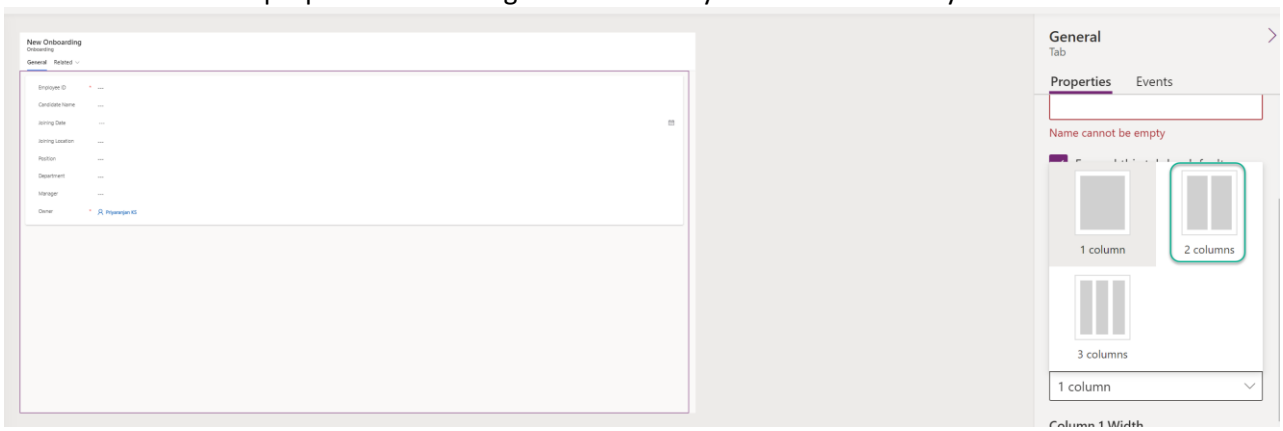
Lets add the previously created Quick View Forms to the Main form of the Onboarding Form. Head over to the Onboarding Table and Select Forms.



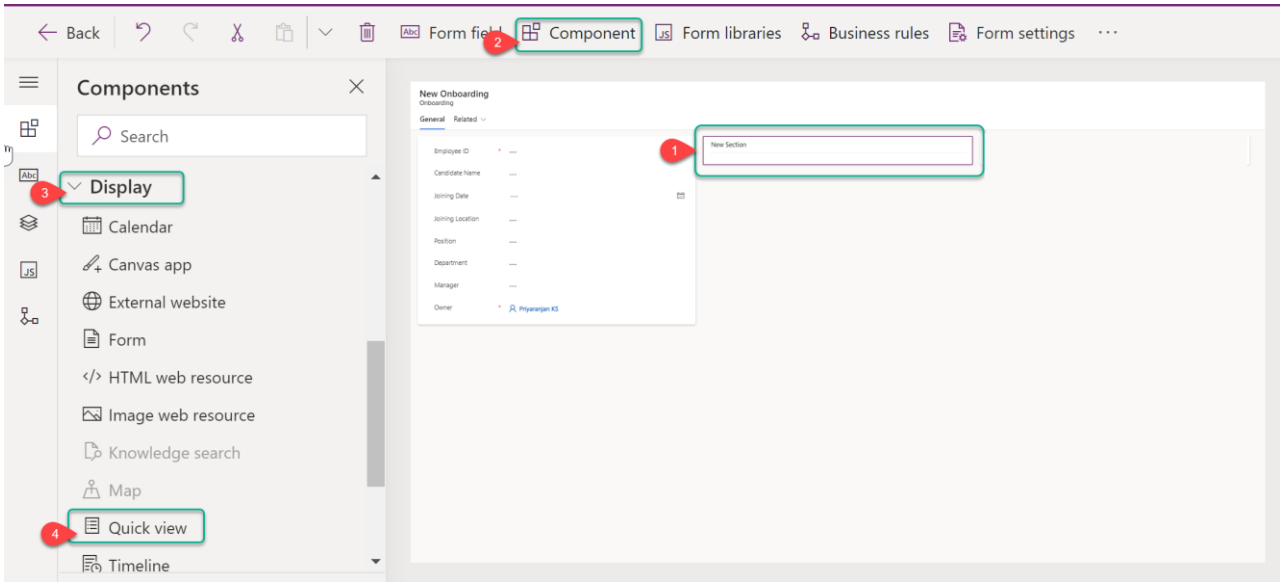
Open the Onboarding Main form by clicking on it.



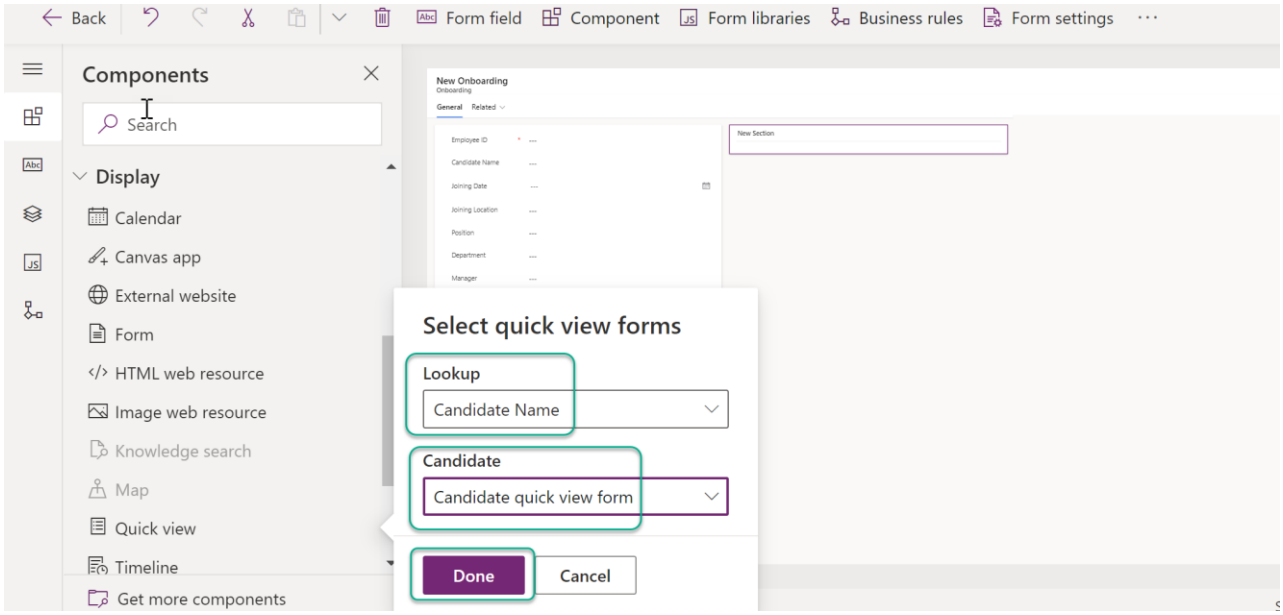
Lets head over to the properties and change the current layout to *2 Columns* layout



Select the *New Section* and click on *Components* which will open up the left pane control list. Select *Display* and Click on *Quick view*.



From the Dialog, select the look up column as *Candidate Name* and the Pick the *Candidate quick view form* from the drop down. Click Done.



The Candidate Quick View Form has come up in the Onboarding Main Form. Let's hide the *New Section* label by selecting the section and checking *Hide Label*. Click on Save.

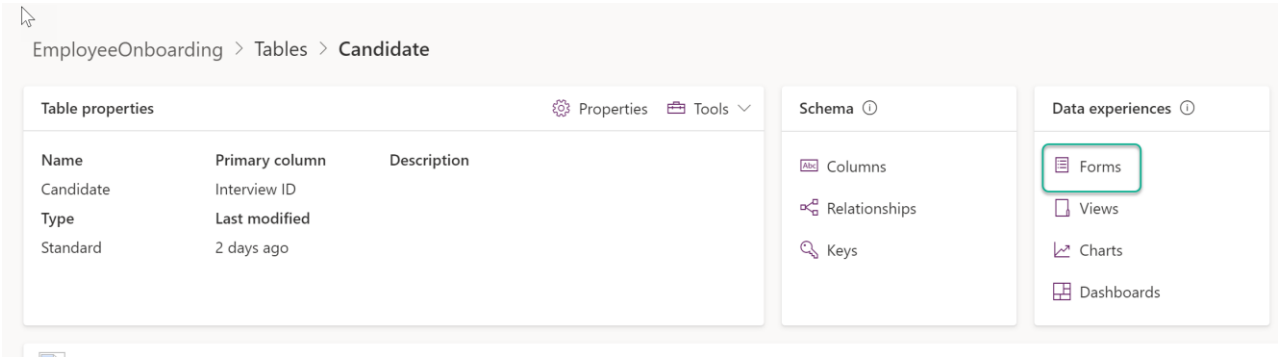
Thus we have added the quick view form of the Candidate table on to the Main form of the onboarding

table.

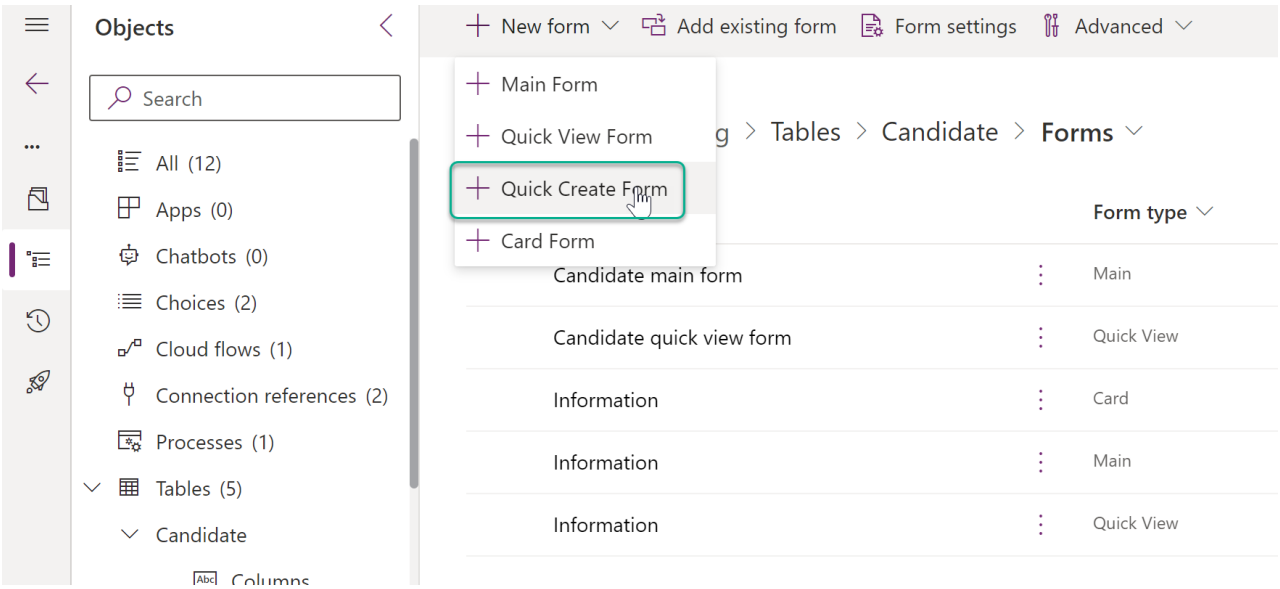
Quick Create Form

The quick create form provides a sleek narrow window for faster record creation where by you can just add the mandatory fields to the form for faster record creation process.

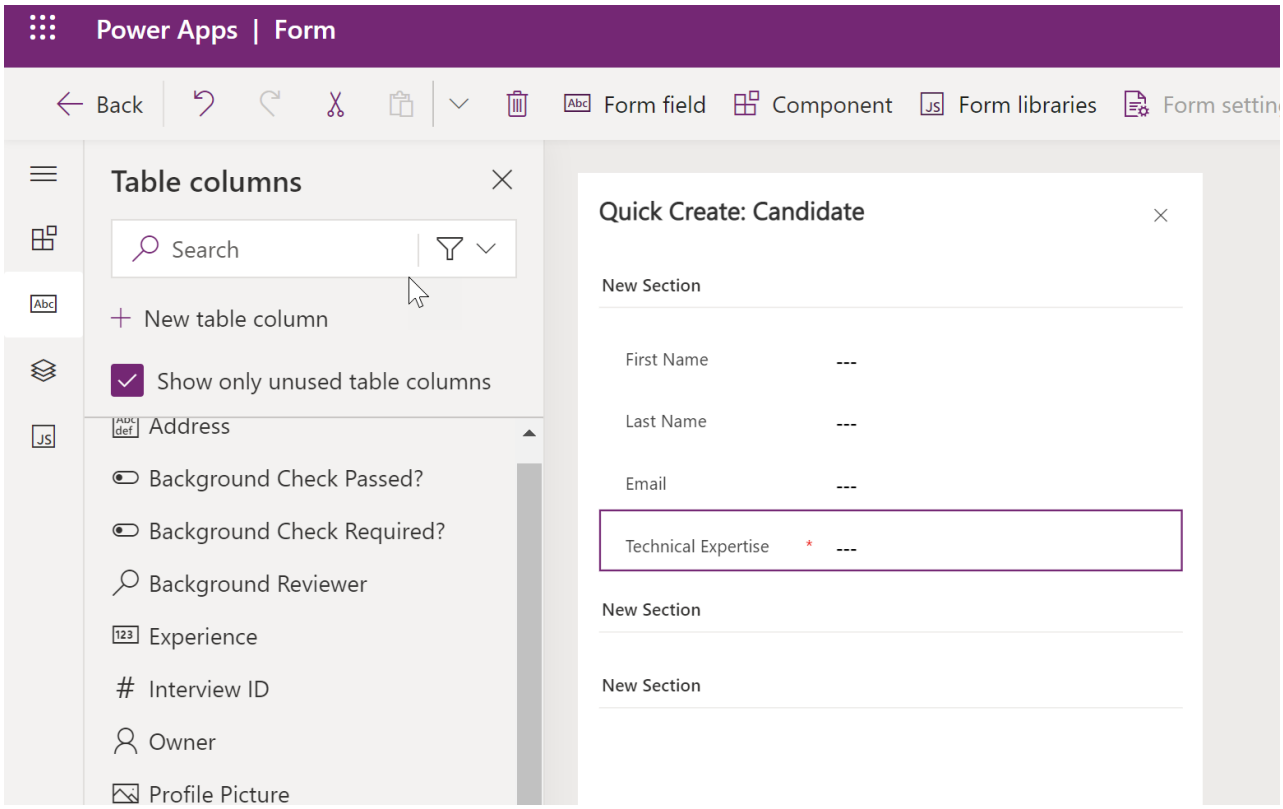
The Quick Create Form can be created from the Forms section of the table, lets create a Quick Create form for Candidates tables by going inside the table and selecting Forms.



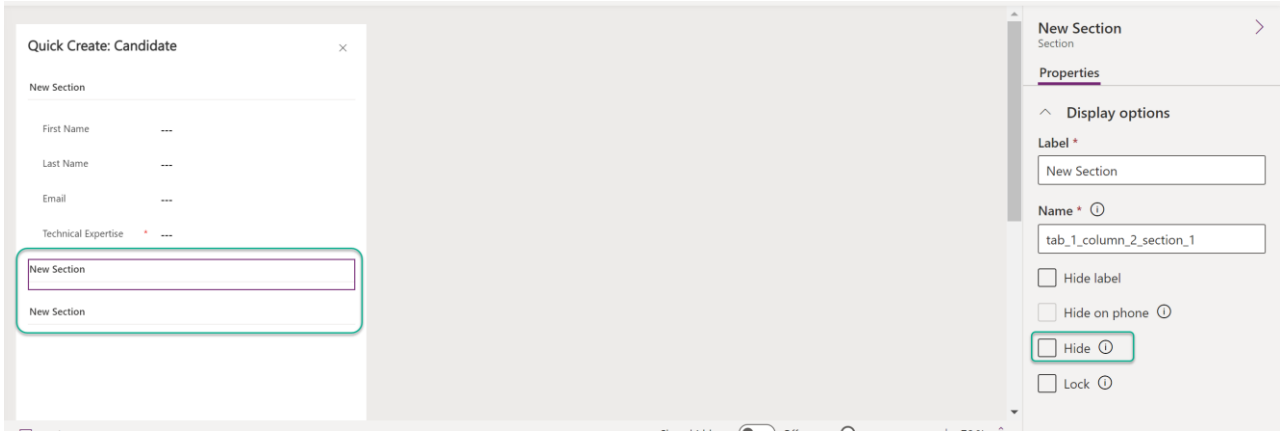
Select New form ->Quick Create Form



Lets drag and drop minimal required fields to the Quick Create Form Designer as below



The rest of the two sections can be hidden by checking Hide from the Properties tab. Click on Save.



Thus we have completed the creation of the Quick Create form for the Candidates table.

Views

Lets create 2 views for the Candidates and the Onboarding table.

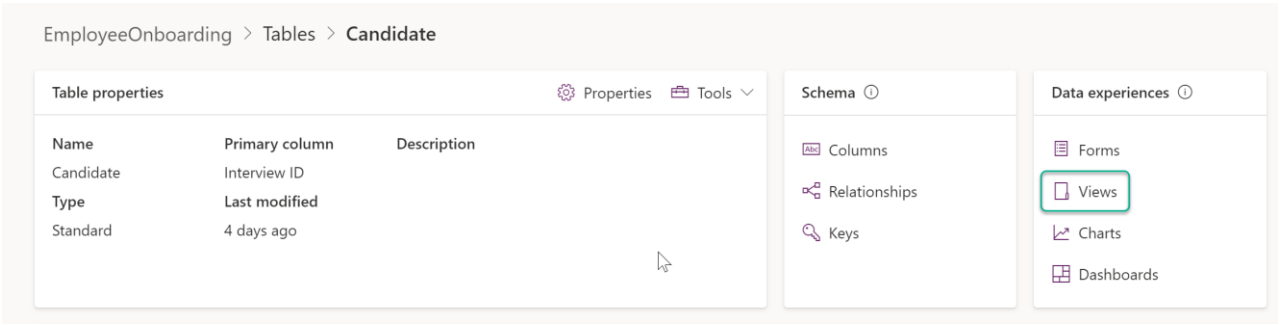
Followed by that we will also create 2 views that will show the details of the selected and rejected candidates so that we can show them in the Model Driven App.

We will finally create one view that will be used as the source of a subgrid that shows all the interview details of the candidate.

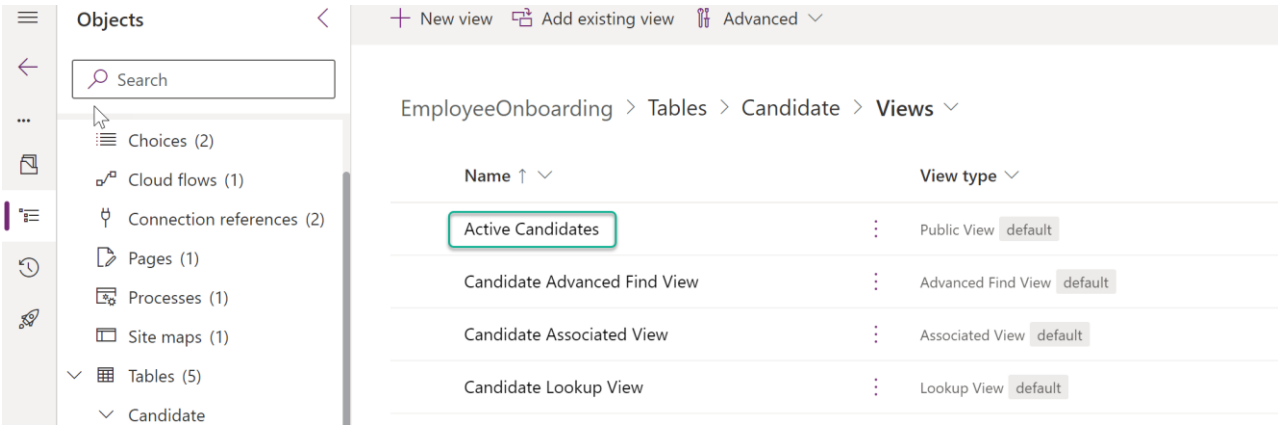
Create Candidates View

We will be creating the a View from the Candidates Table. Let's head over to the Candidates table and

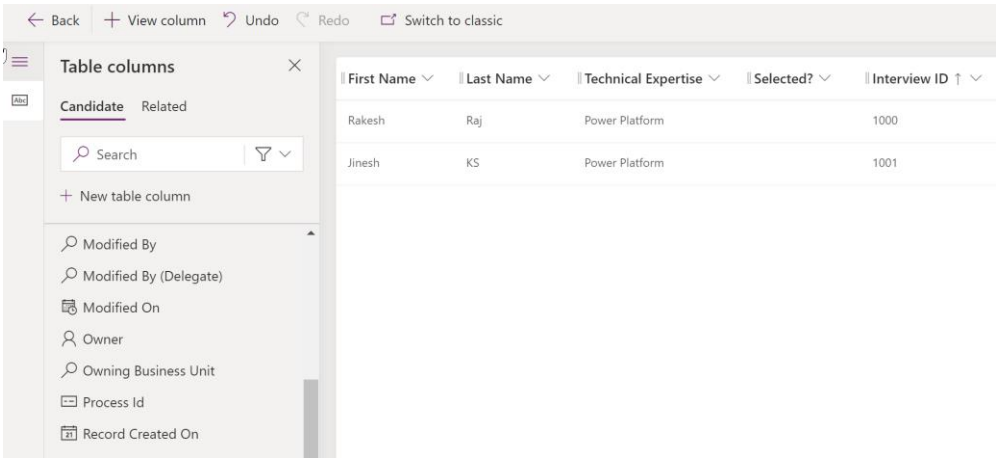
select Views.



Select the Active Candidates View

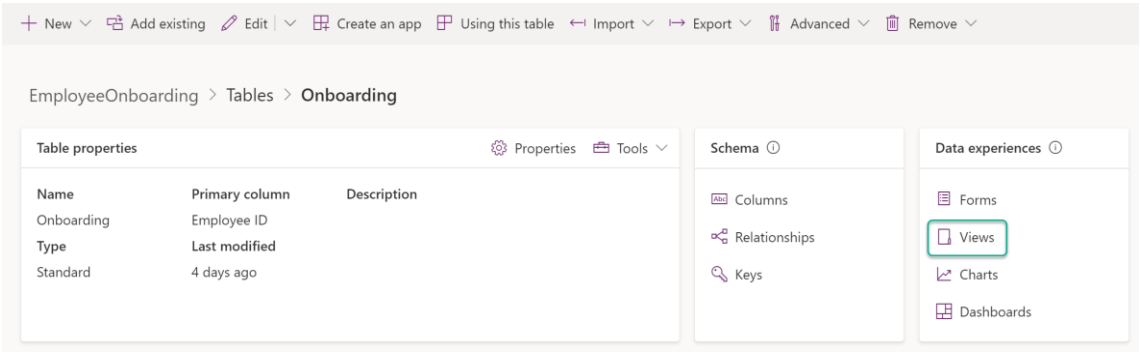


Drag and drop the needed columns on to the table . Click on Save.

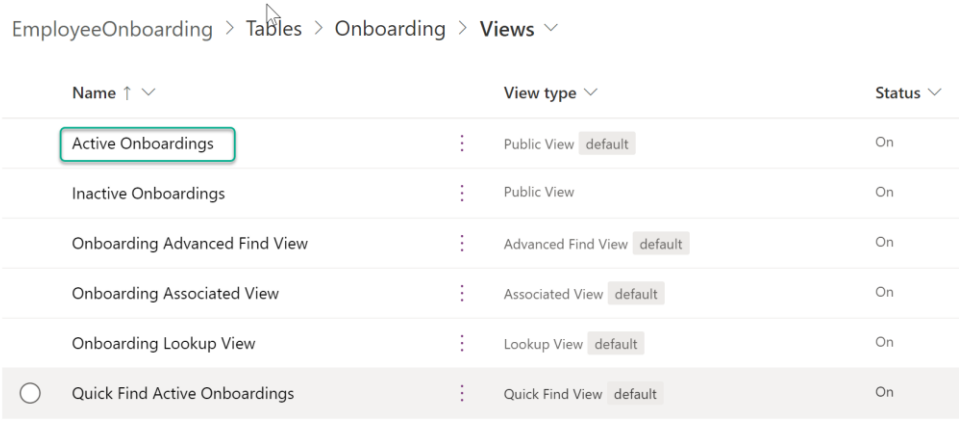


Create the Onboarding View

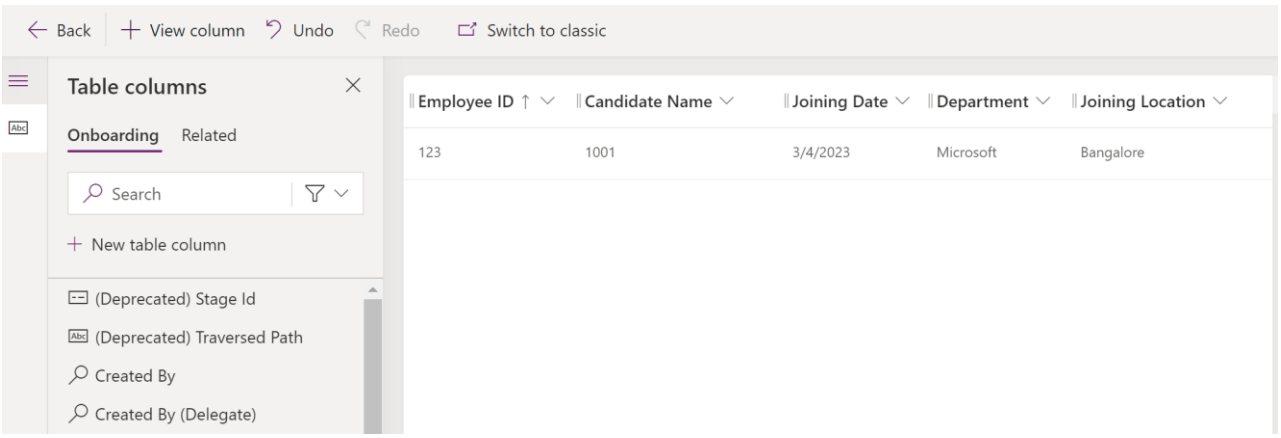
Head over to the Onboarding table and select Views



Select the Active View from the list

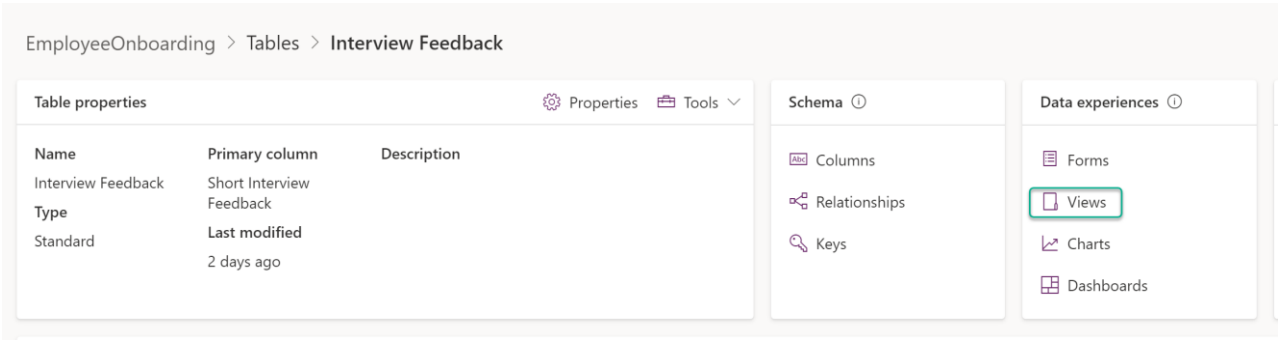


Drag and Drop the needed columns on to the View and Save

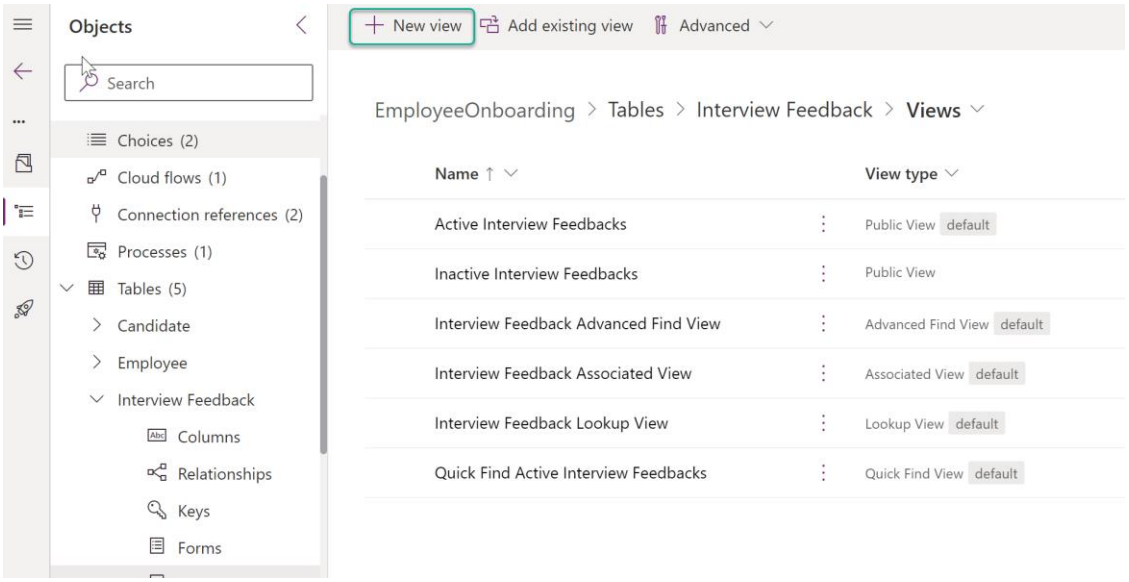


Create Selected View

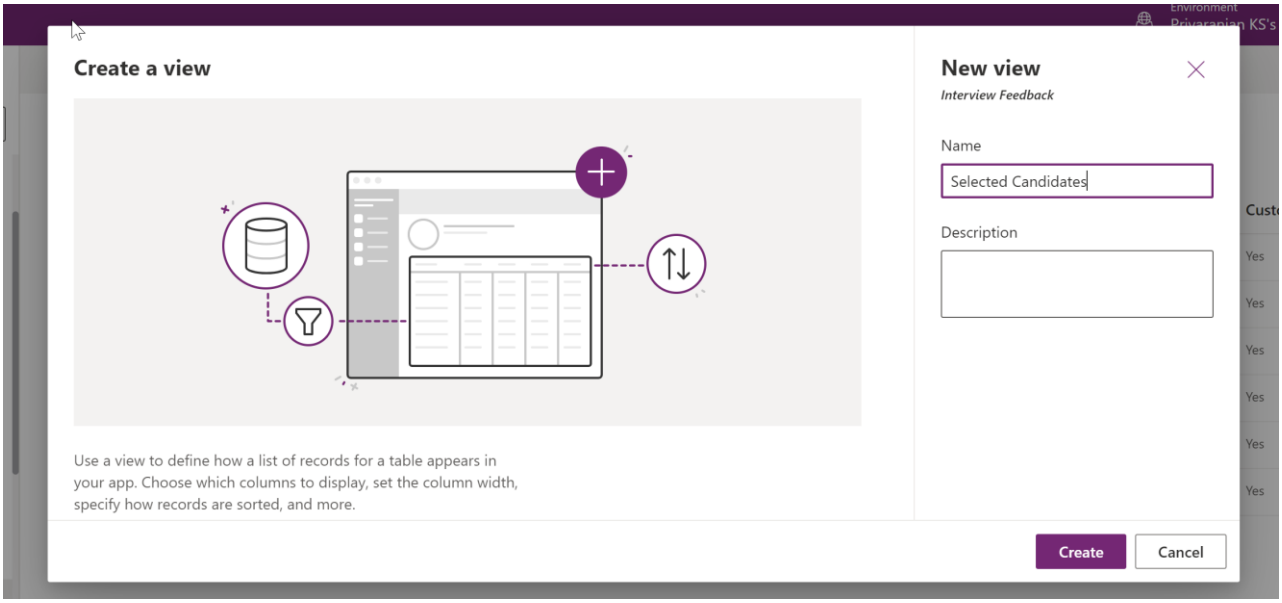
We will be creating the selected View from the Interview Feedback Table. Lets head over to the Interview Feedback table and select Views.



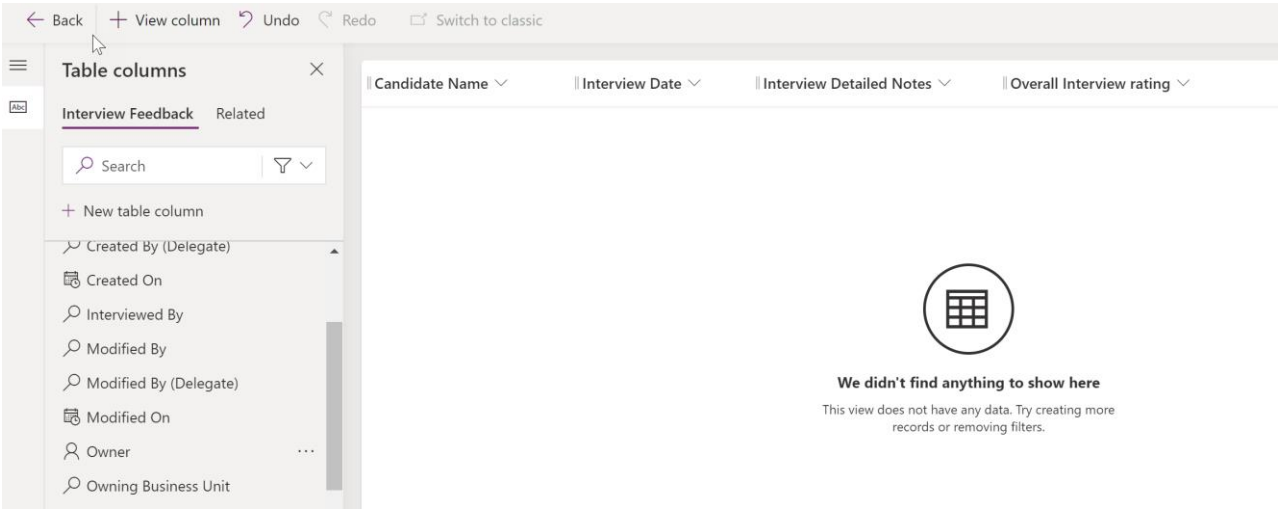
Select *New view* .



Specify the view name and click on Create.

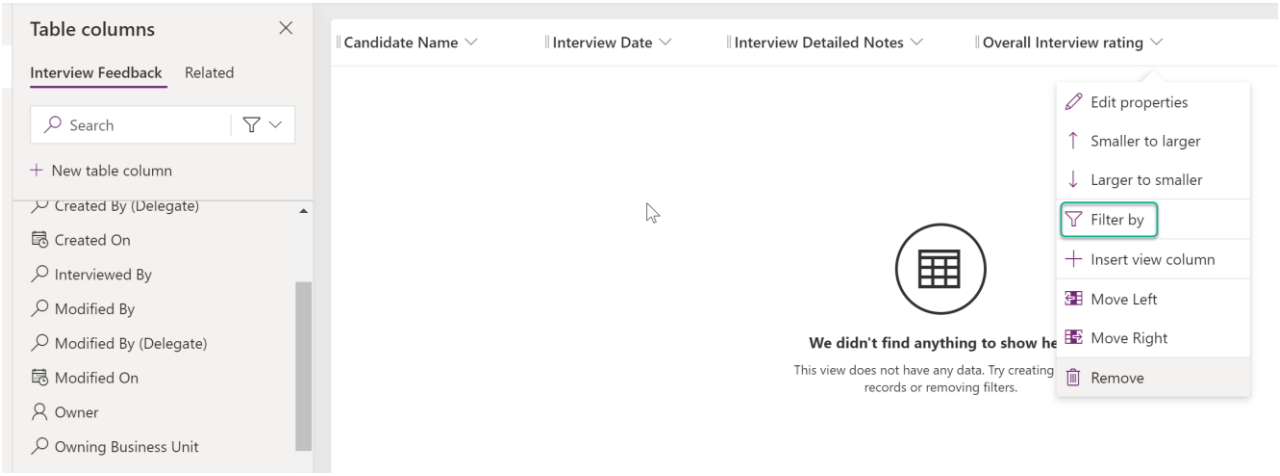


Drag and drop the needed columns to the view

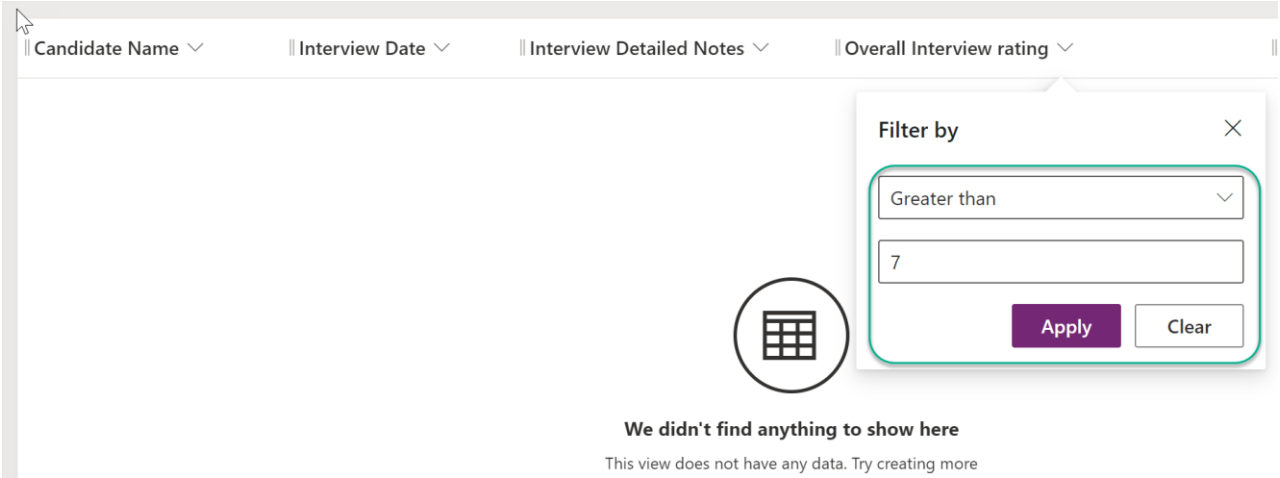


So as to fetch only the selected candidates in the view, we will now add a filter. The HR has given a mandate that only candidates scoring more than 7 out of 10 rating will be considered as selected. So We will apply a filter on the *Overall Interview rating* column (which has a min value of 0 and max value of 10)

Select Filter by from the Overall Interview rating column.



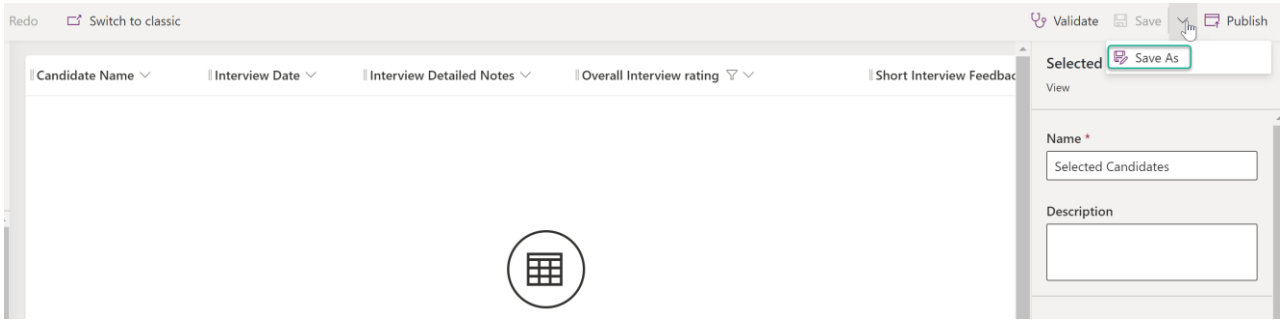
Set the filter as Greater than 7 and click on Apply.



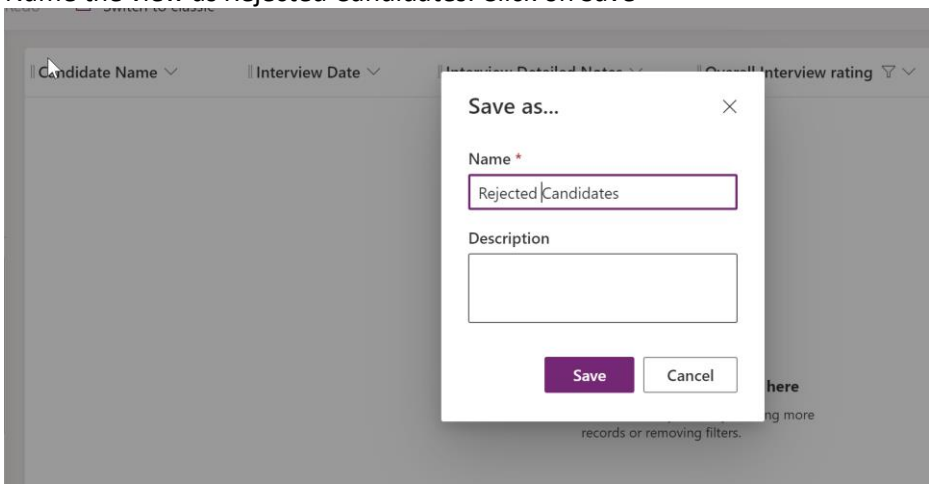
Click on Save to create the view.

Create Rejected View

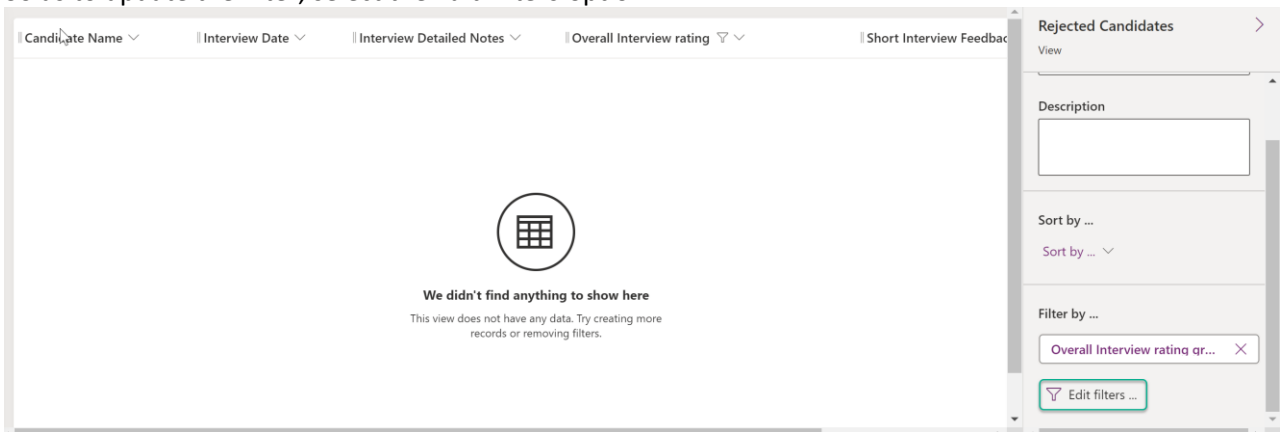
Click on Save as to create a duplicate of the selected view. We will modify the filter to create the Rejected view.



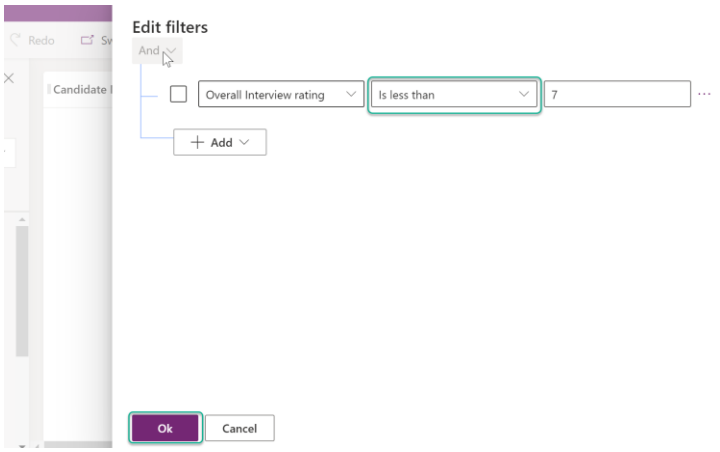
Name the view as Rejected Candidates. Click on Save



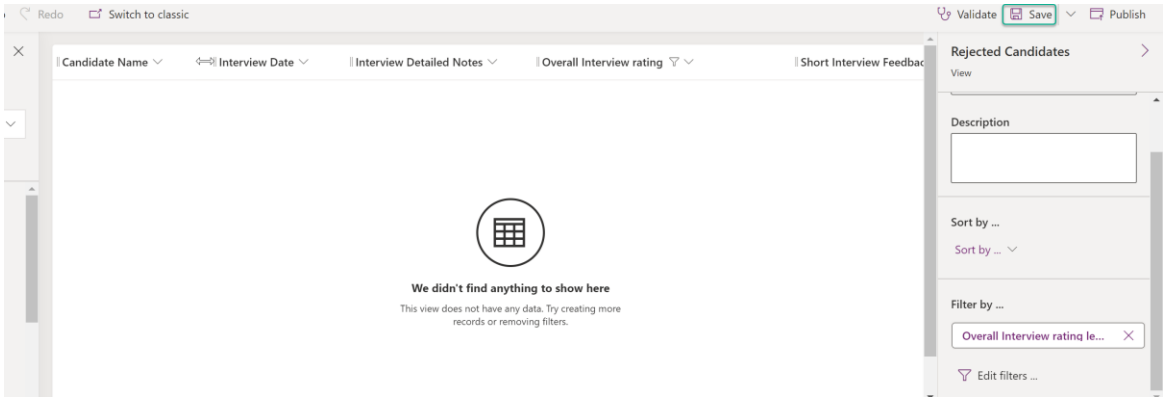
So as to update the filter, select the *Edit Filters* option



In the Filter, Change “Is Greater than” to “ Is Less than” and click Ok



Click on Save.

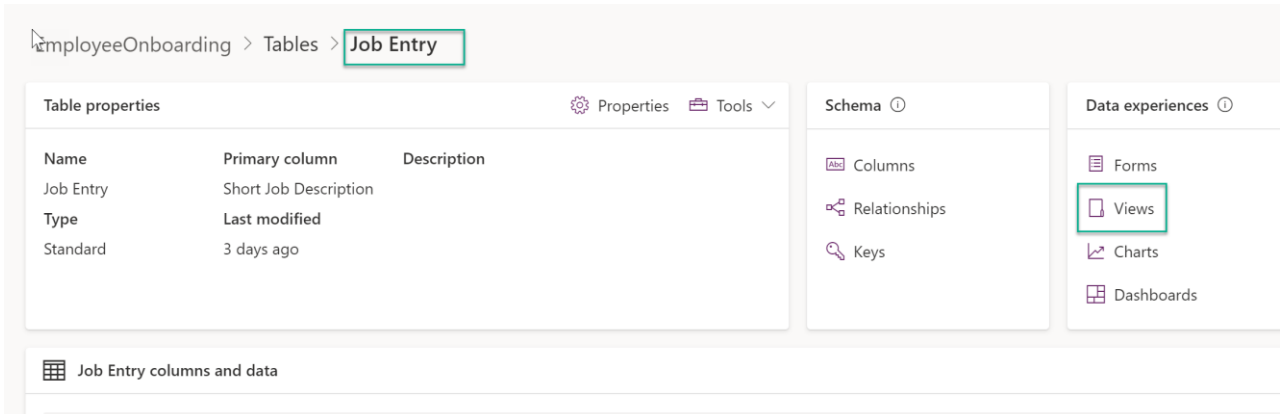


Thus we have completed the creation of the Selected and Rejected Candidate Views.

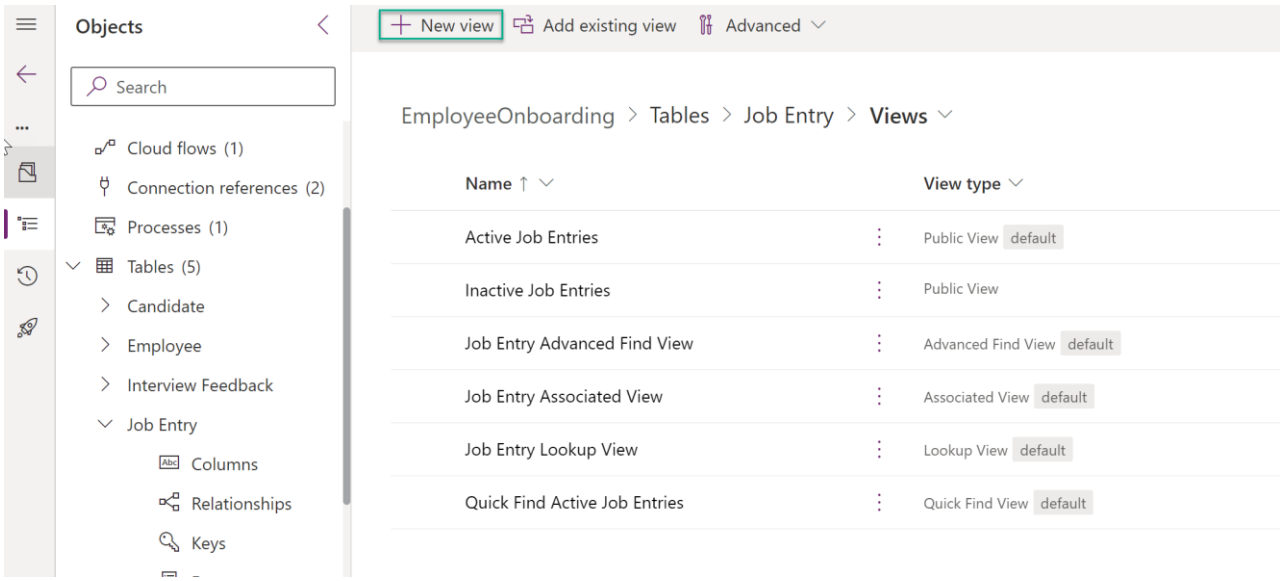
Create Previous Interviews View

This will will be used as the source of a subgrid where we will show the previous interview details of a specific candidate. To do that we will create the view and pass the candidate name from the subgrid to auto filter the view.

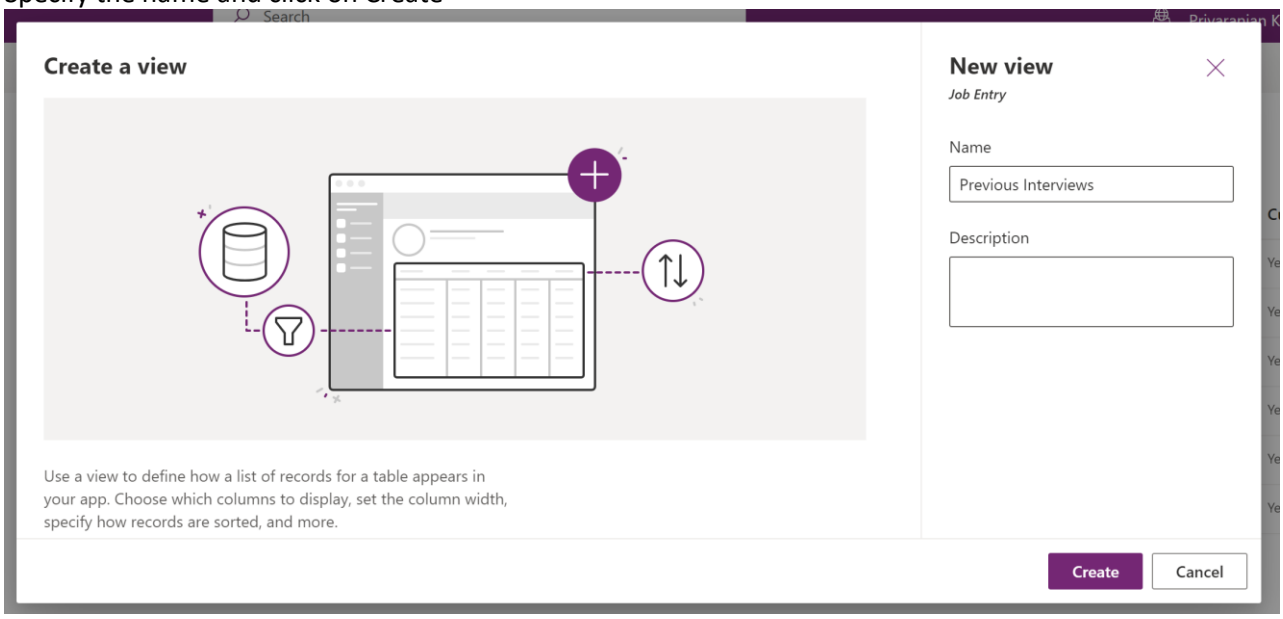
Lets create the view on the Job Entry table by selecting Views.



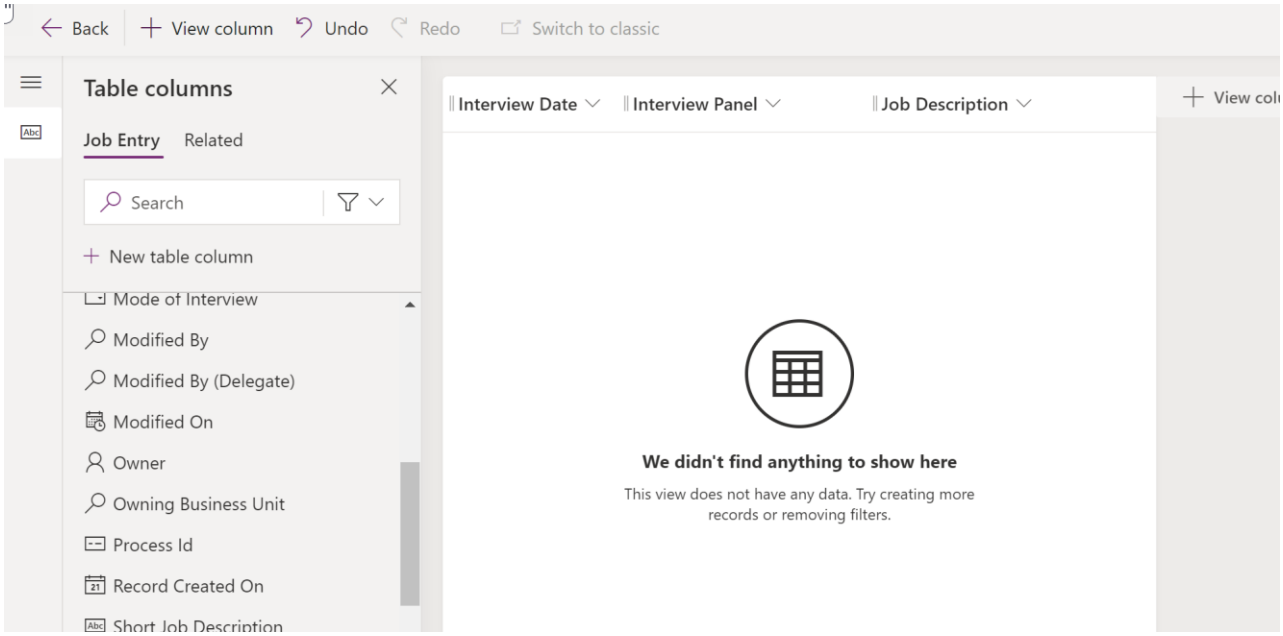
Select New View



Specify the name and click on Create



Drag and drop the needed columns on to the View designer .Click on Save.



Thus we have create the view needed for the subgrid.

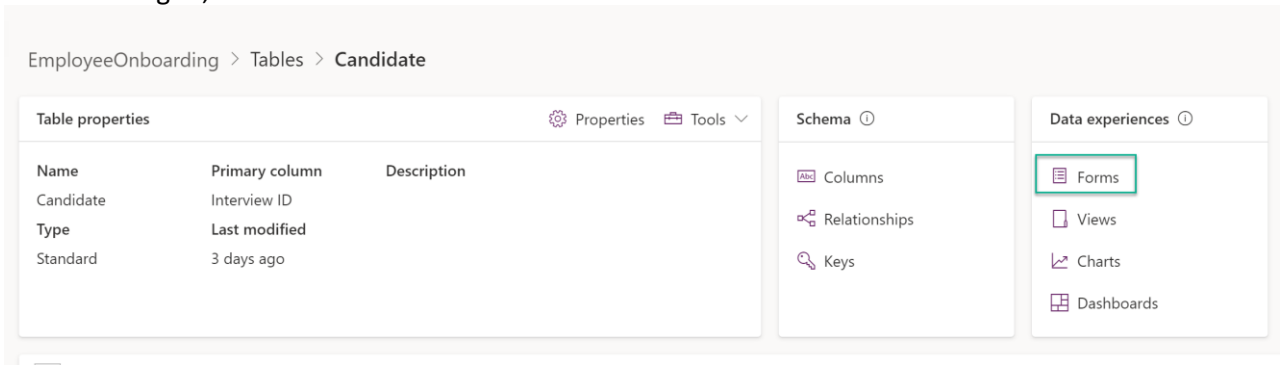
Subgrids

Subgrids provides a native way to show related records from other tables as a grid within the main form. This way we will be able to show information that are related by a row value.

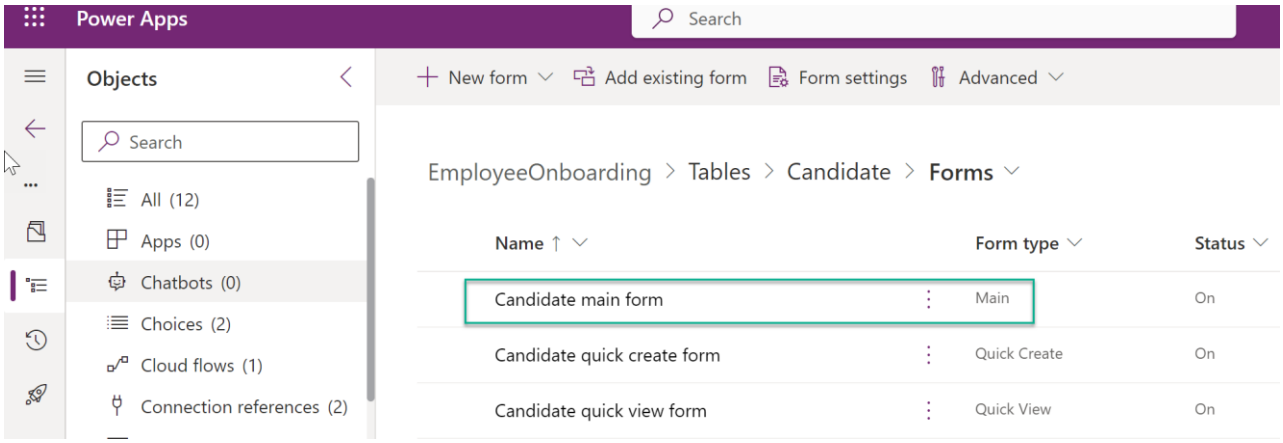
In our exercise, we will add a subgrid to the candidate main form. This way a single candidate's form will show his previous appearances for interview which is stored in the Job Entry Table. A single candidate can have multiple job entries to which he would have applied resulting in a 1 – many relation from the candidate to Job Entry table. Thus picking all those previous interview records and showing it against the candidate will give a complete picture of the interview history.

Create Subgrid

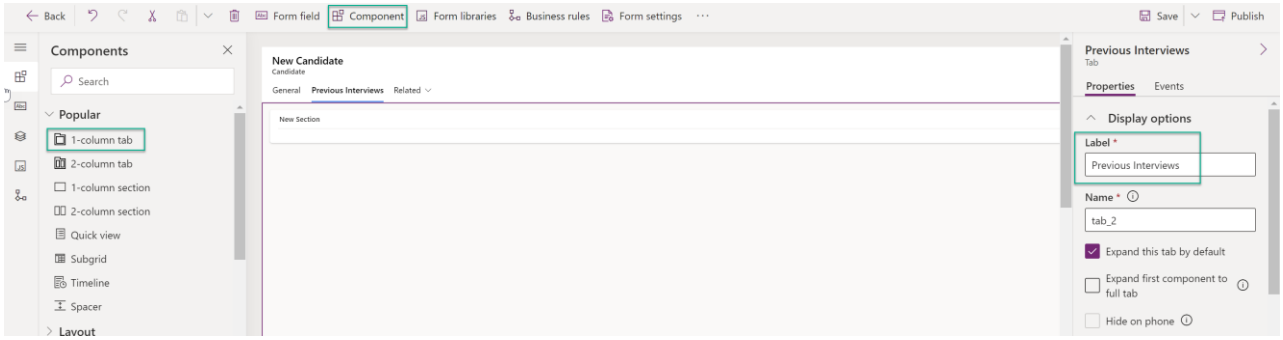
To add a subgrid, lets head over to the Candidate table and select Forms.



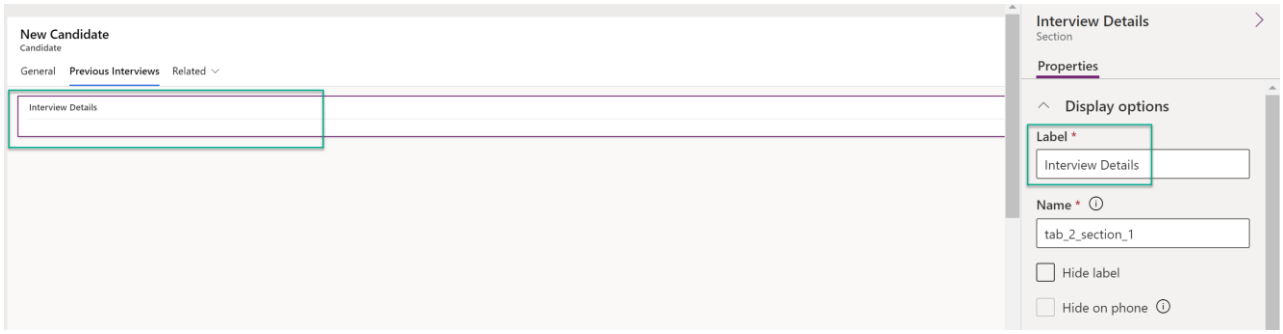
Select Candidate Main Form



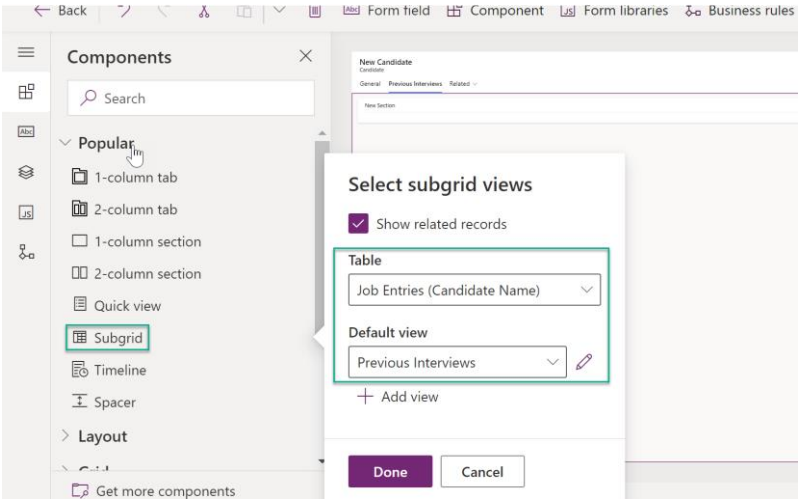
From the *Component* section, select *1-column tab* and specify the label as *Previous Interviews*.



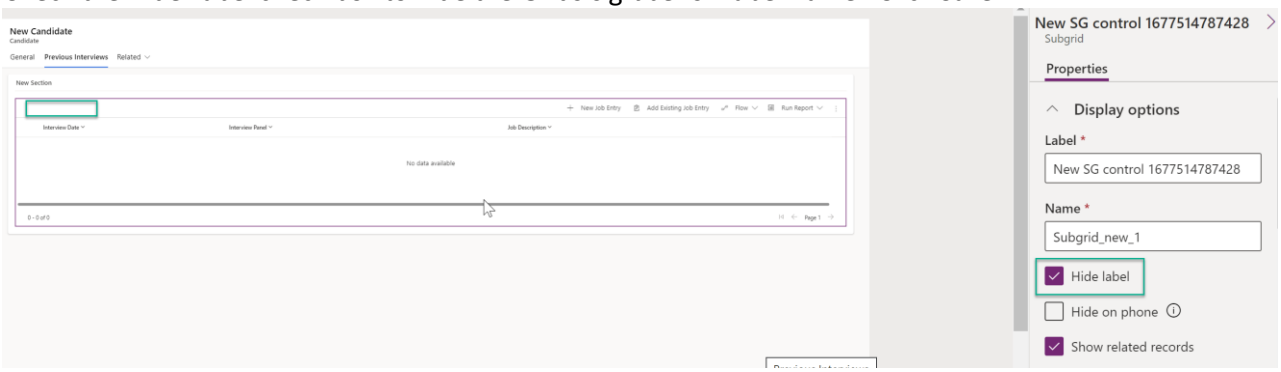
Select the *New Section* and set the label as *Interview Details*



In this renamed section, Select Subgrid and check the checkbox *Show related records* . Specify the table as Job Entry and select the view that has to be shown in the subgrid. Click on Done



Check the Hide Label check box to hide the Grids's gibberish label name . Click Save

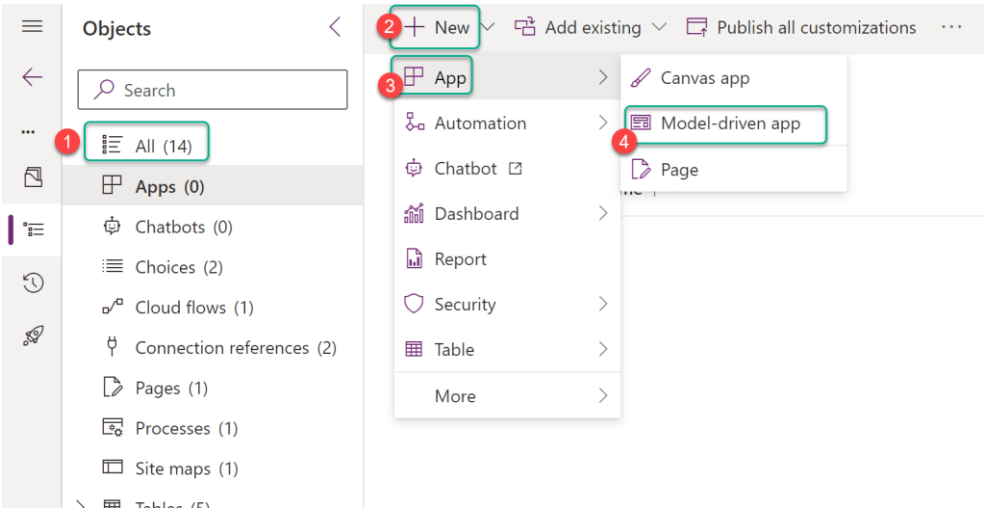


Thus, we have completed the addition of the subgrid as a new tab

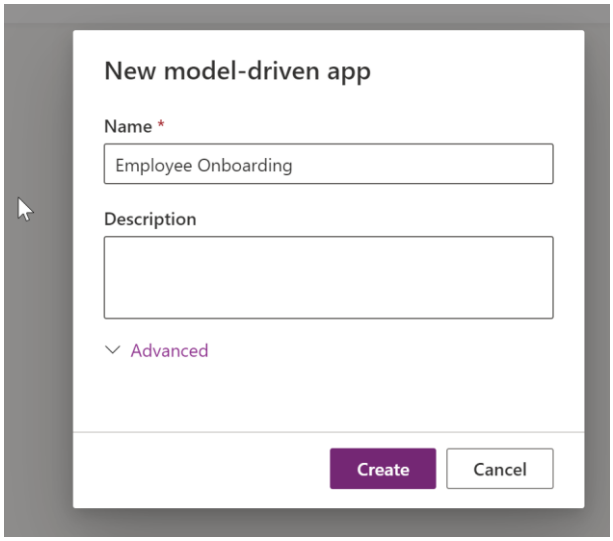
Create a Model Driven App

So far we have created the basic components needed for creating a model driven app. Now lets create the model driven app and plug in the components we have created so far.

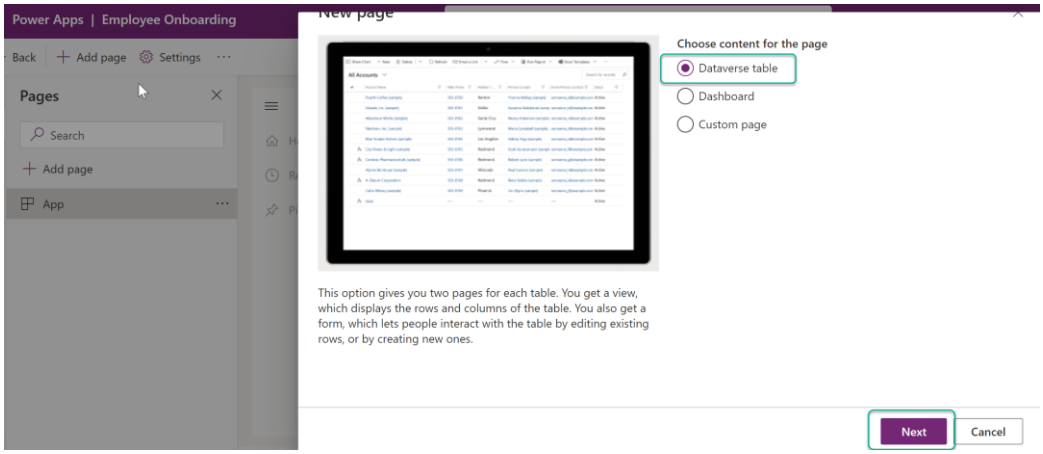
Select *All -> New -> App -> Model-driven app*



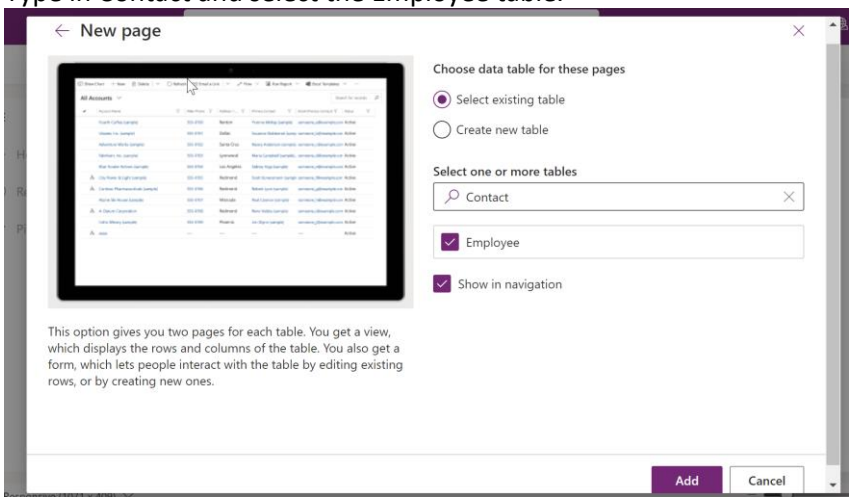
Specify the app name and click on Create



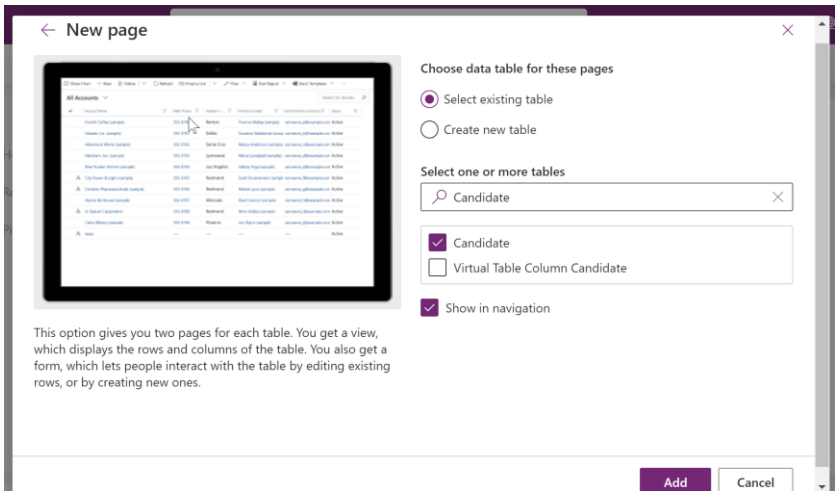
Select Dataverse table and click on Next



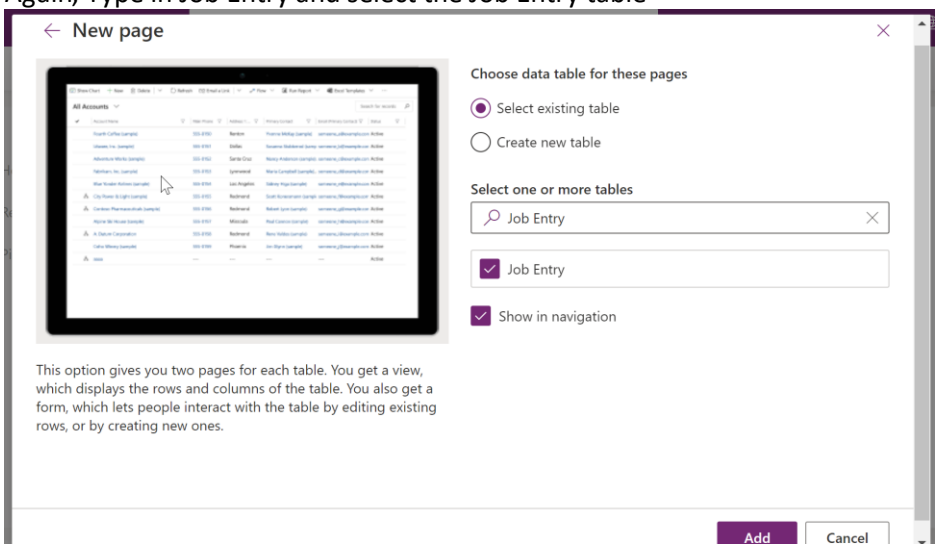
Type in Contact and select the Employee table.



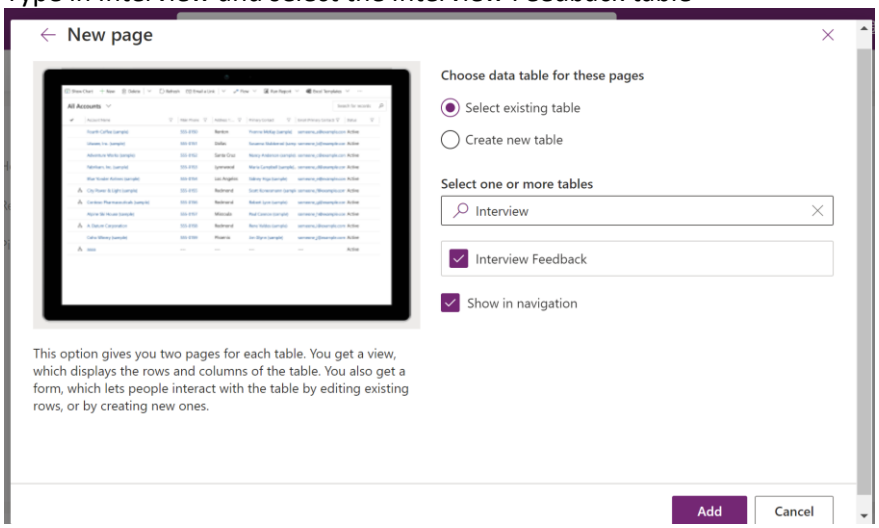
Again, type in Candidate and Select the Candidate table .



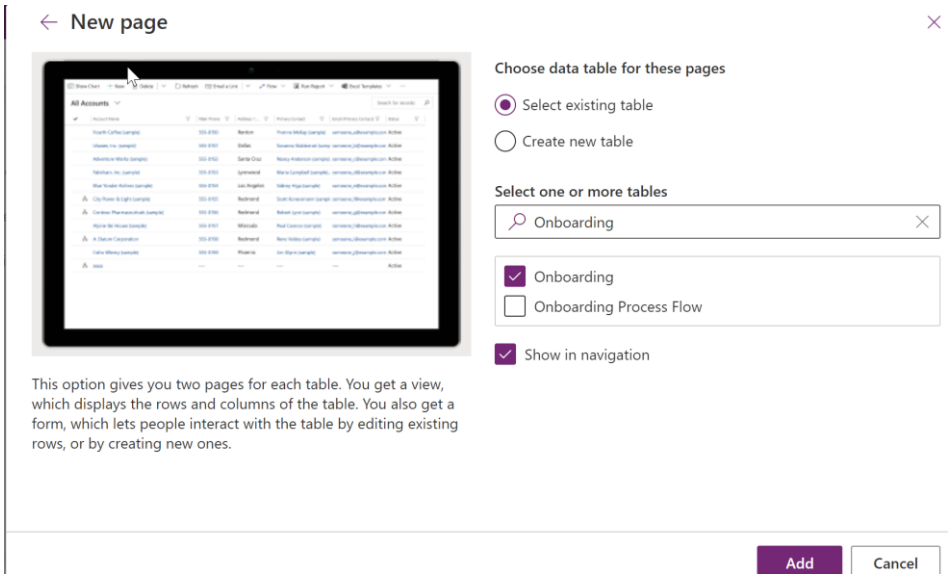
Again, Type in Job Entry and select the Job Entry table



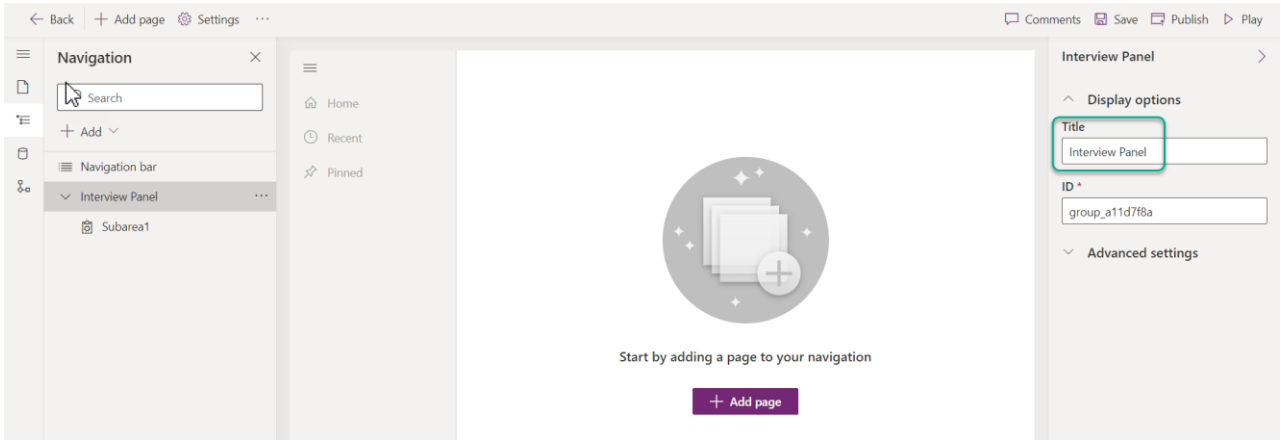
Type in Interview and select the Interview Feedback table



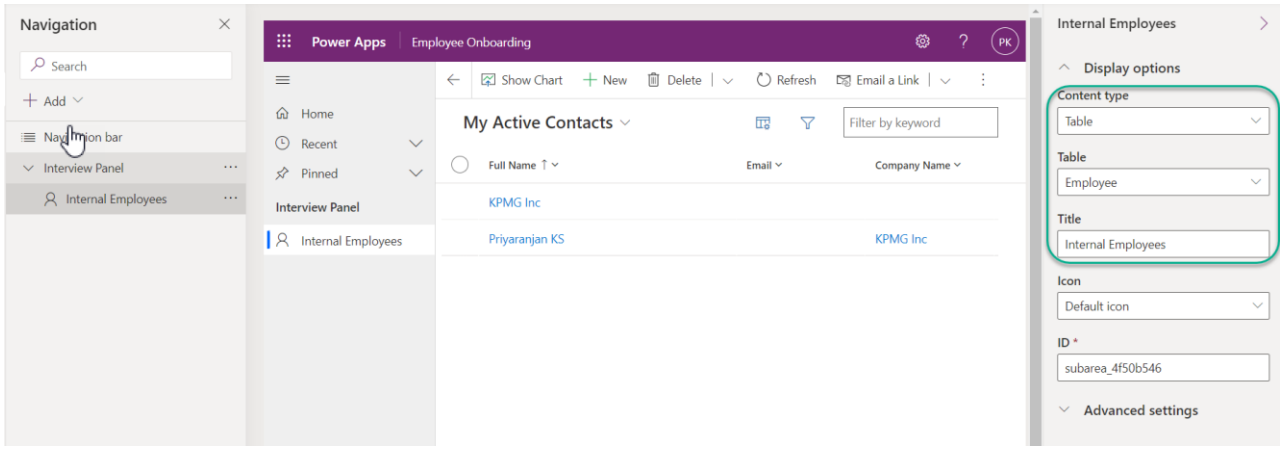
Finally select the Onboarding table ,Uncheck the *Show in navigation checkbox and* click on Add.



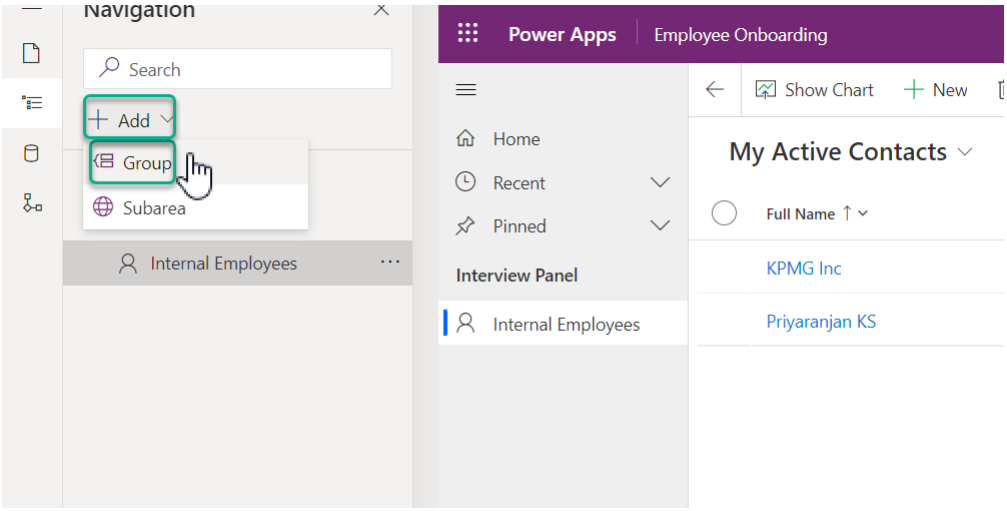
Select *Group 1* and Specify the Title as *Interview Panel*



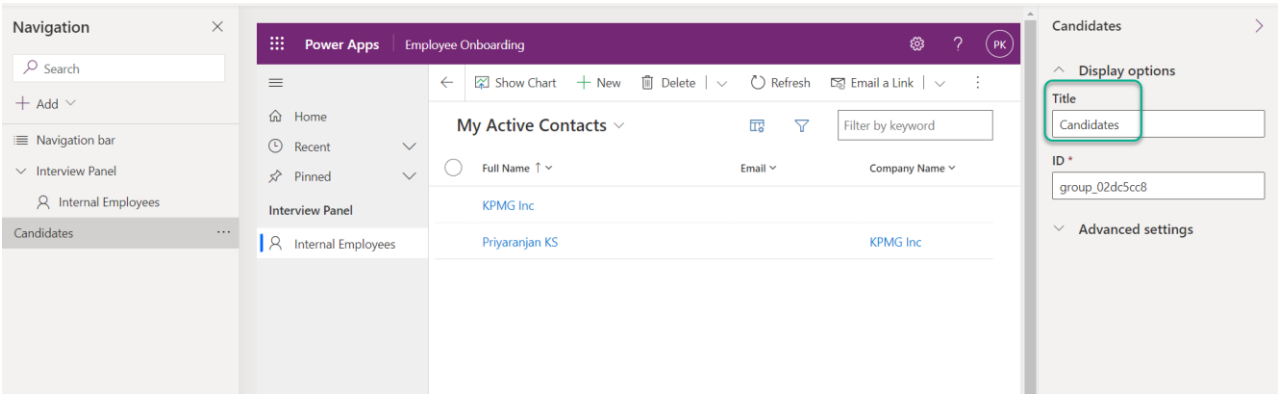
Select the Subarea 1 and specify the Content Type as *Table*. Select the *Employee* Table and mention the title as *Internal Employees*



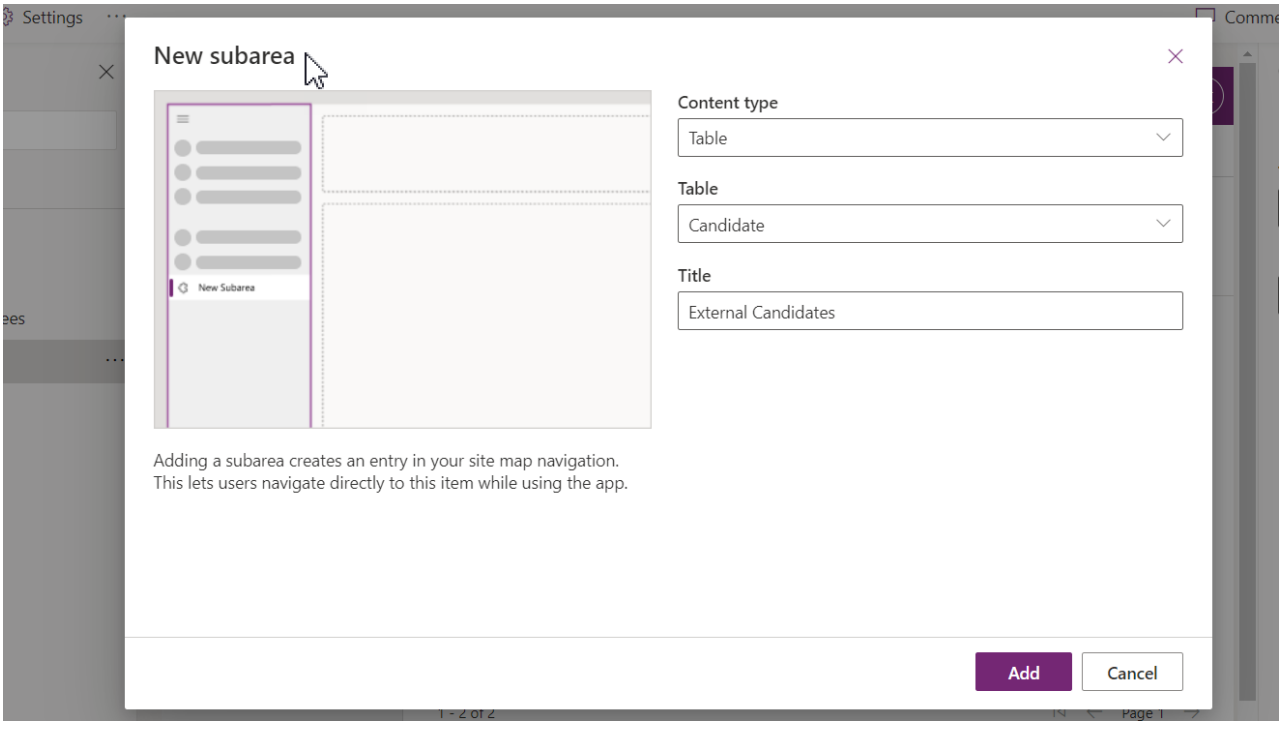
Select Add -> Group



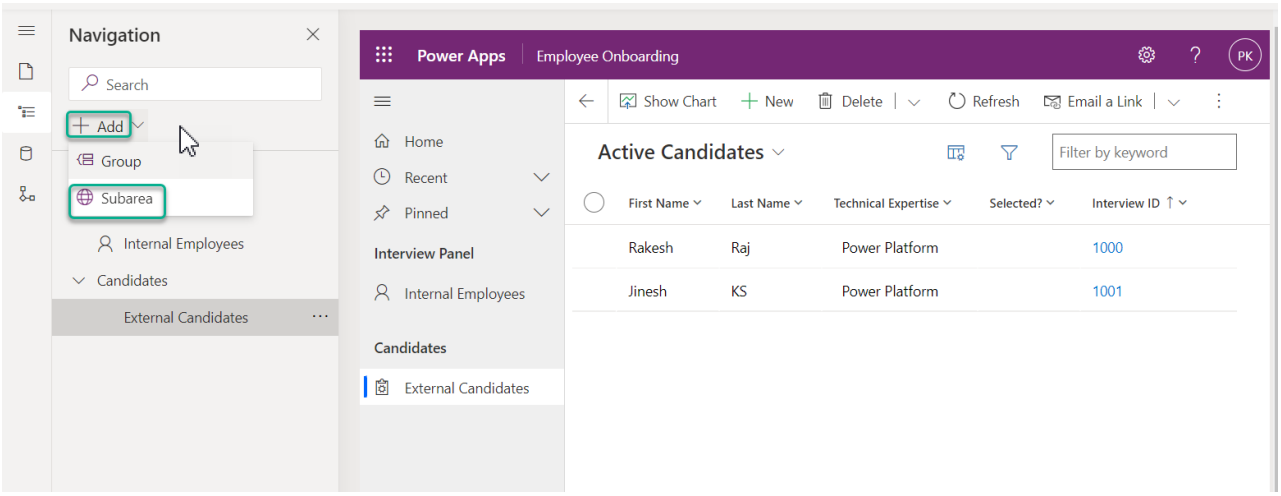
Mention the title as Candidates



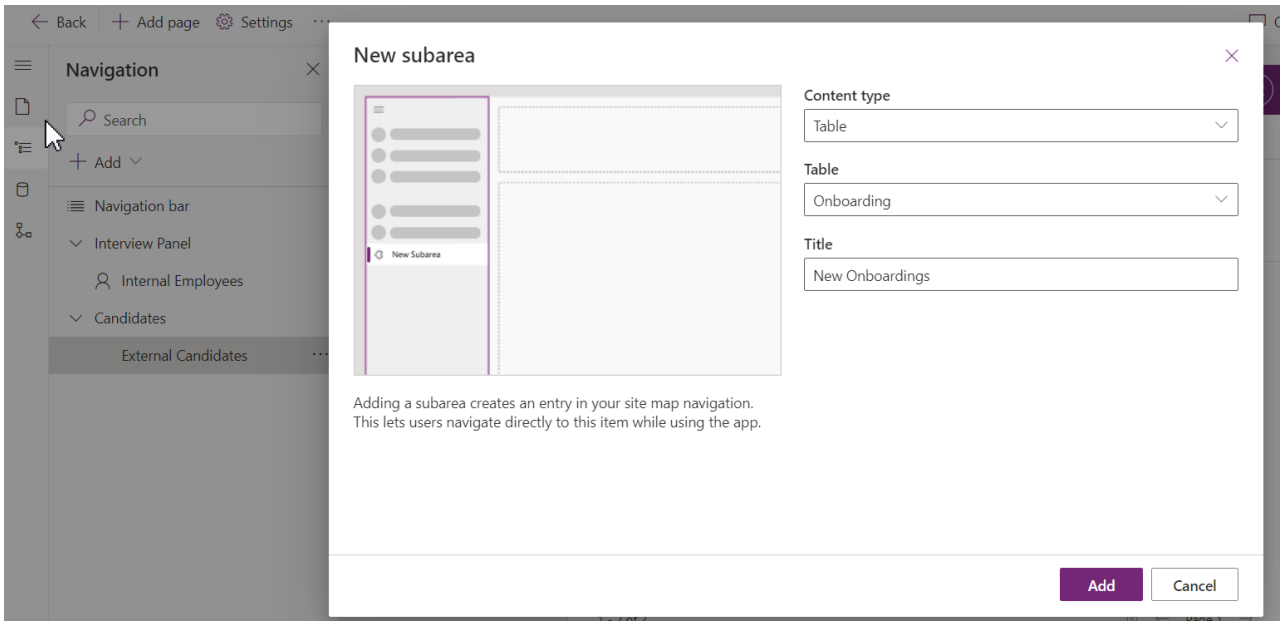
Click on Group -> Subarea and select the Content Type as Table. Select the Candidate Table gives the title as External Candidates.Click on Add



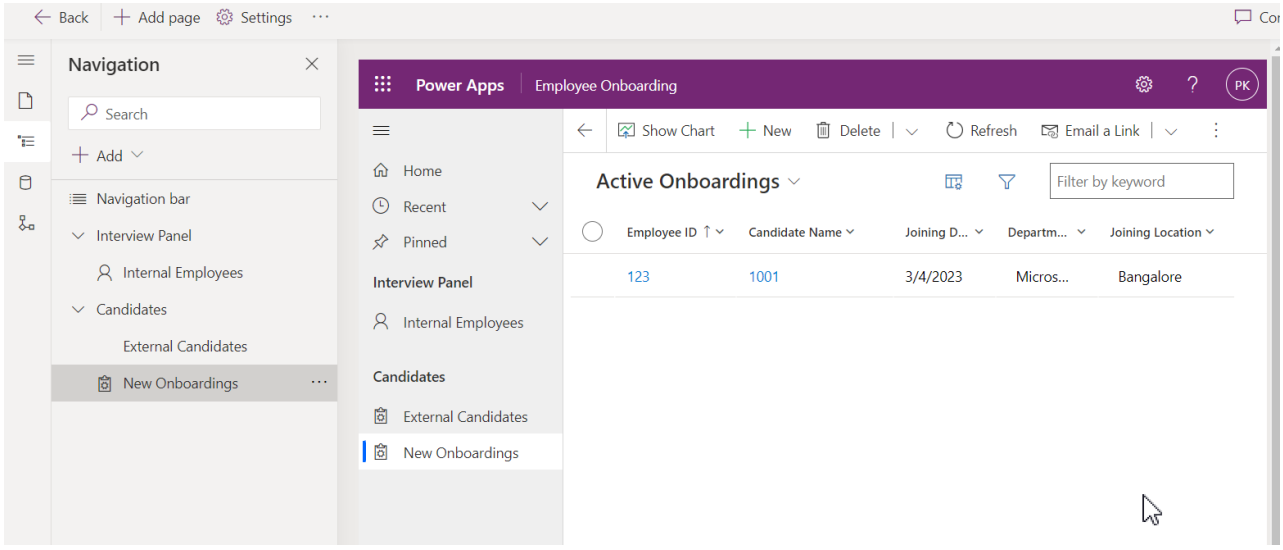
Select Add-> Subarea to add the Onboarding Details.



Mention the content type as Table and select the Onboarding Table. Specify the Title as New Onboardings and click on Add.



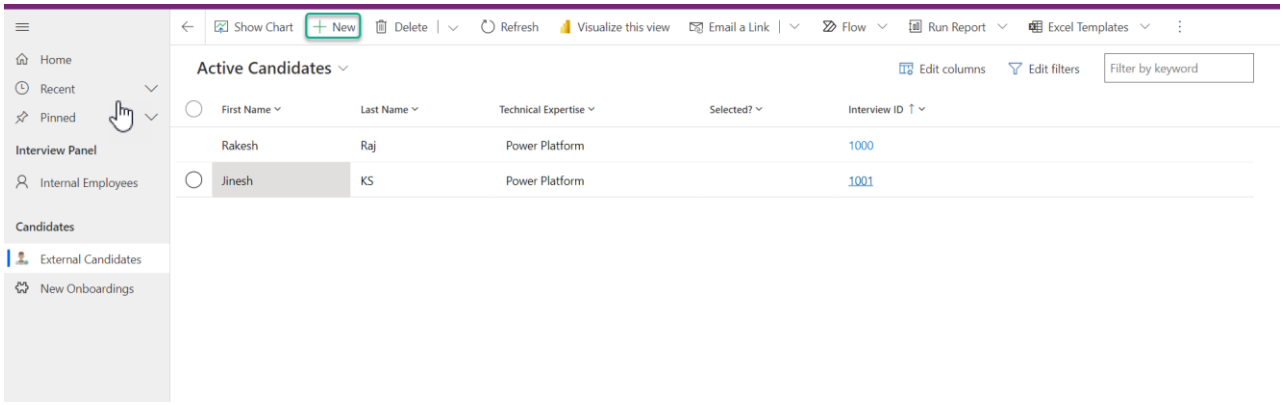
Thus we have added the needed components of the Model Driven App.



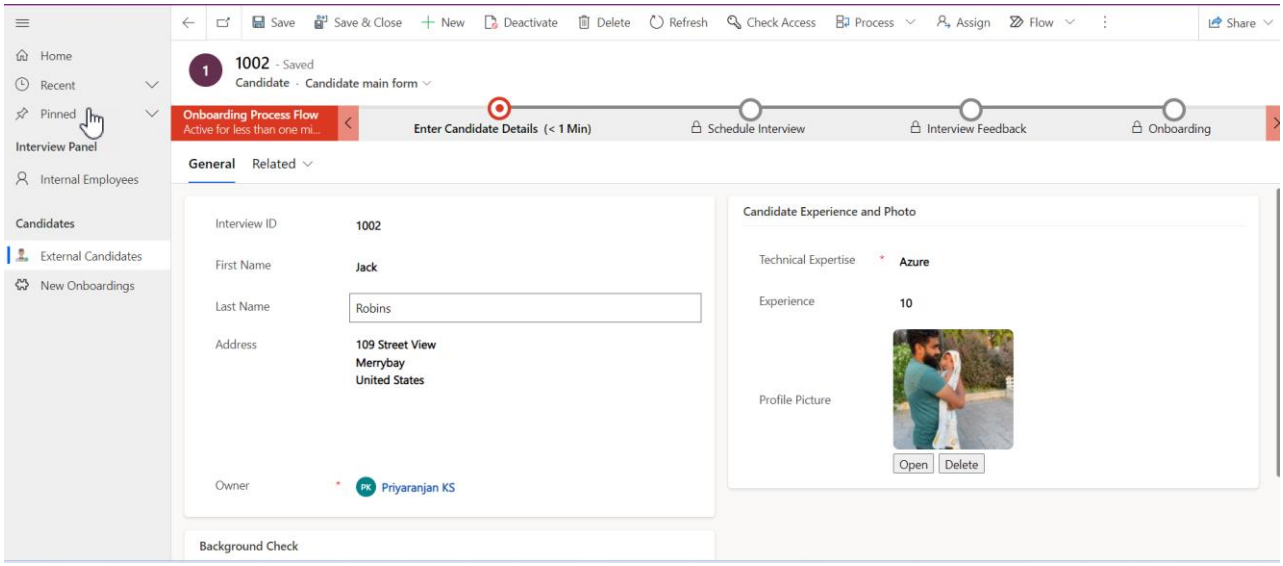
Test the Model Driven App

Publish the app so that it becomes available for end users and lets play it to see how the flow of the app looks like.

Select External Candidates and Click on New

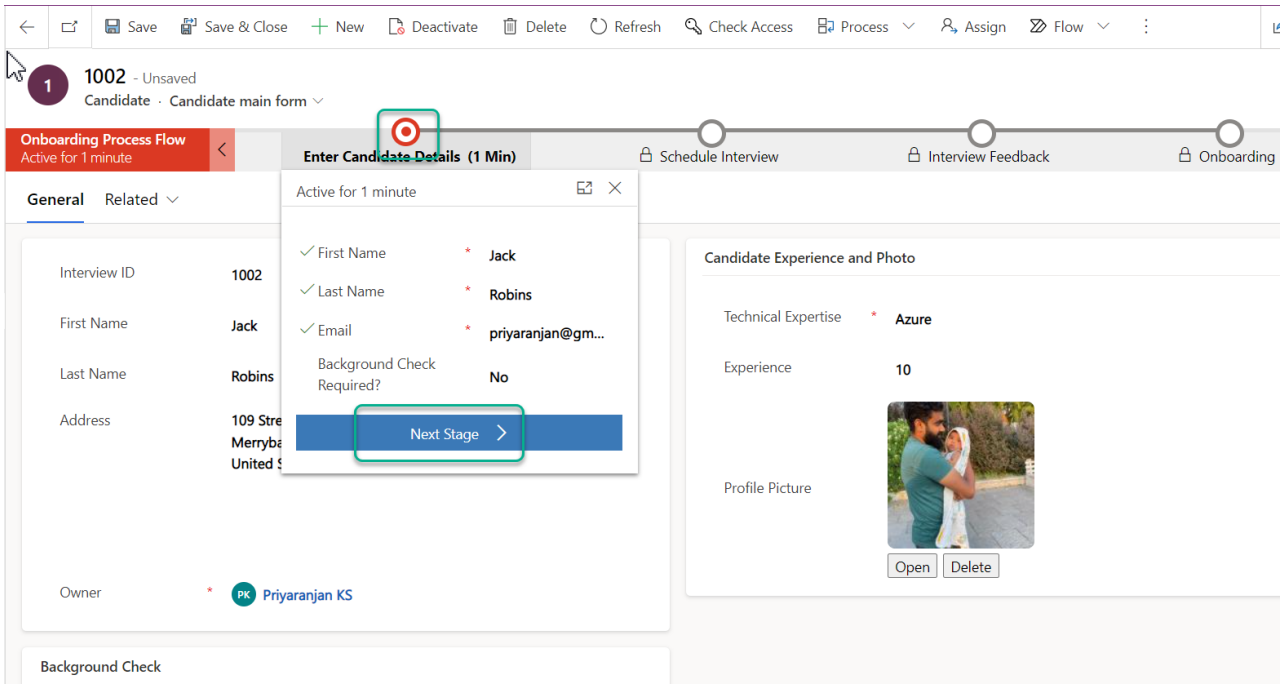


Enter the details of the candidate and click on Save

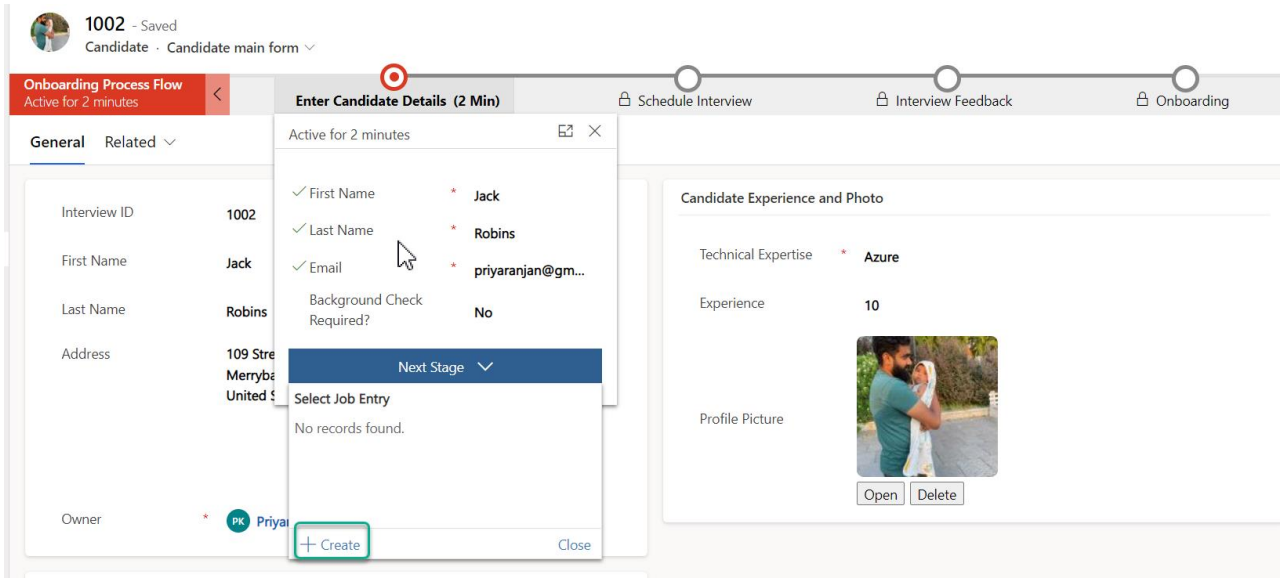


Note : You can upload the photo only once you have saved the record once.

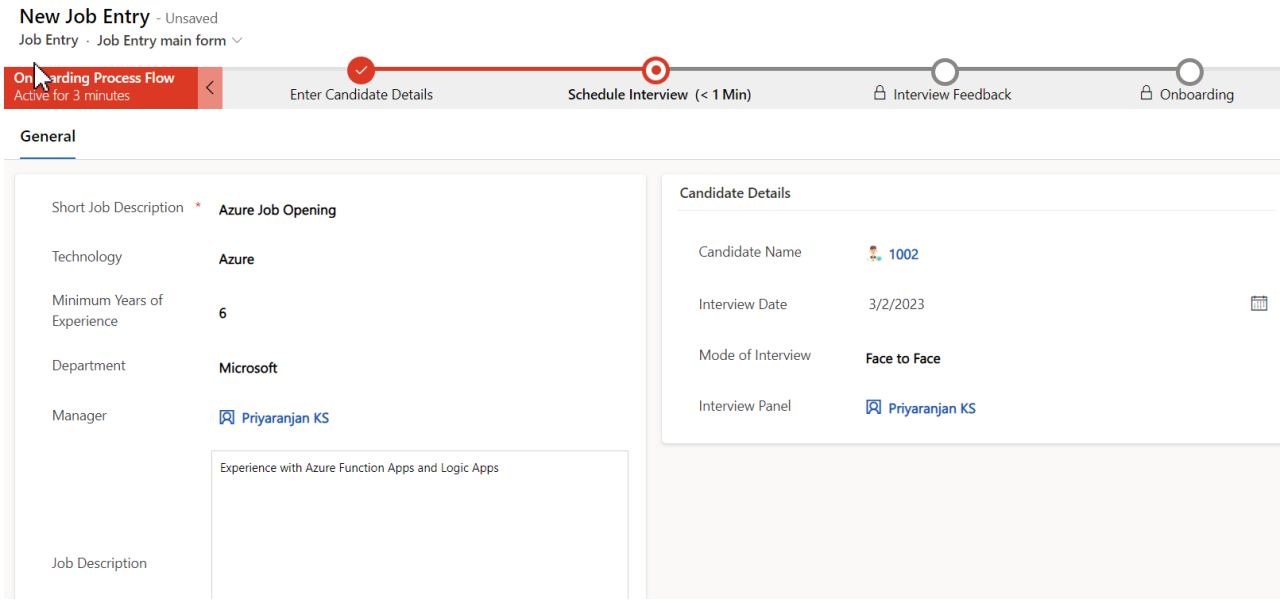
Click on the First stage Red Dot and ensure that all the mandatory fields have been filled in and click on Next Stage



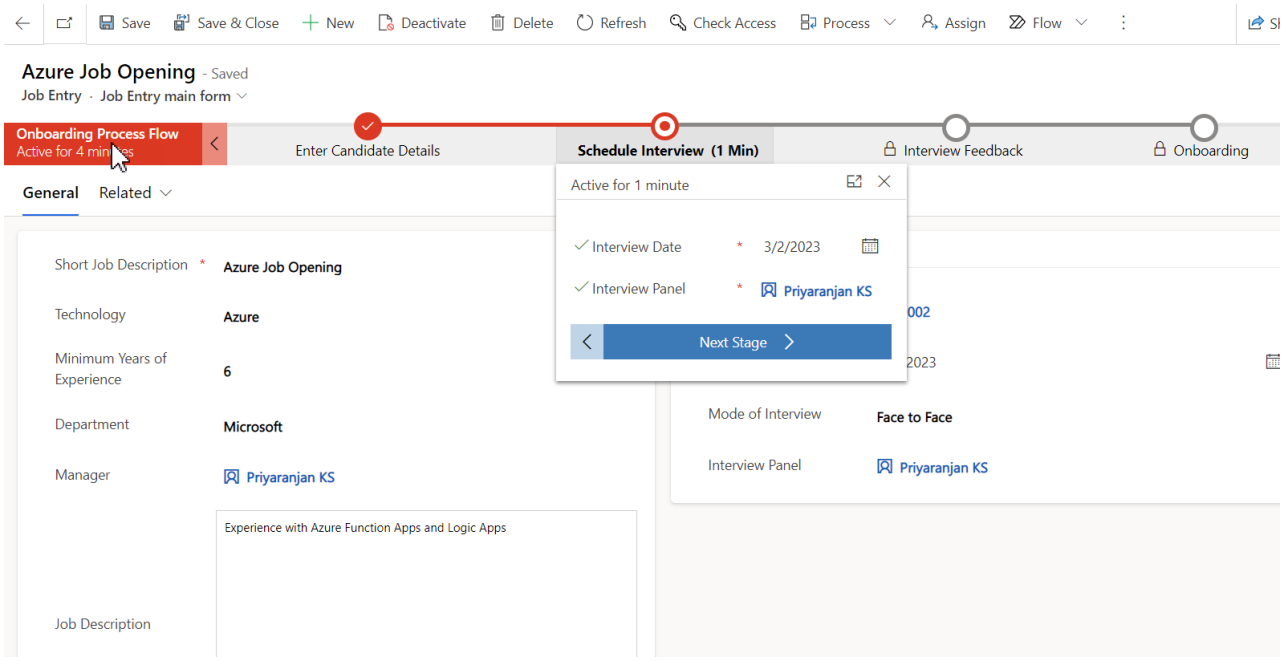
Click on Create to add a new Job entry record for the candidate



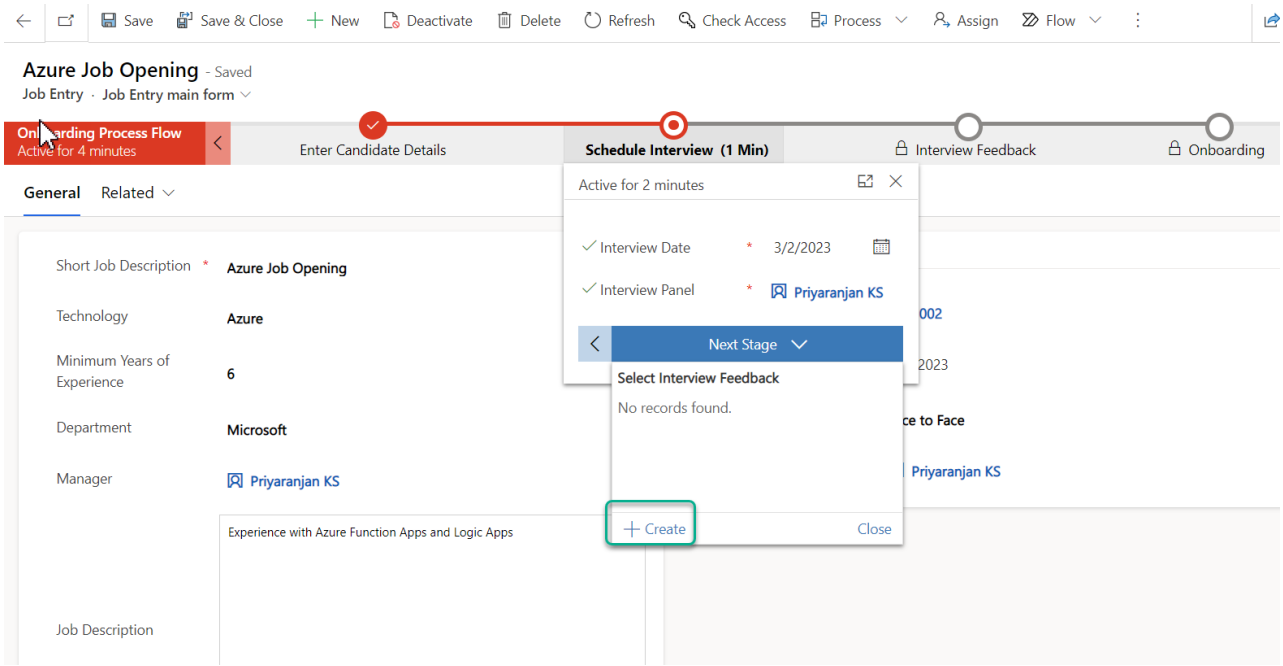
Specify the needed details.



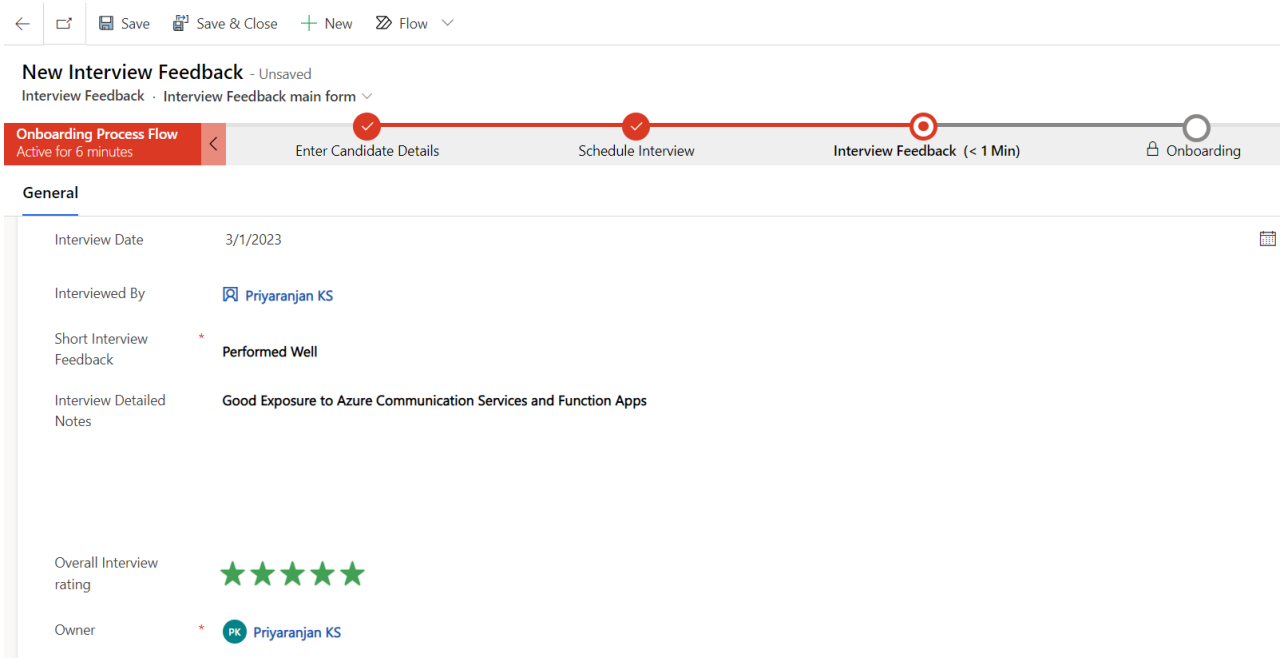
Save the record and move to the next stage



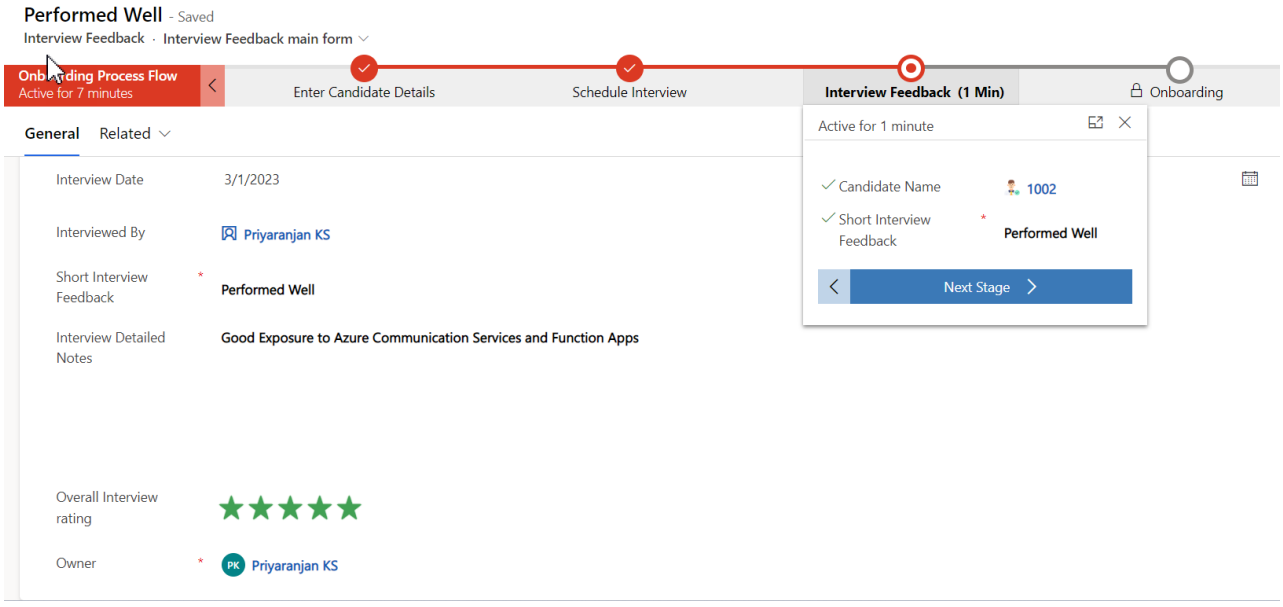
Click on Create to add a new interview feedback record once the interview process has been completed.



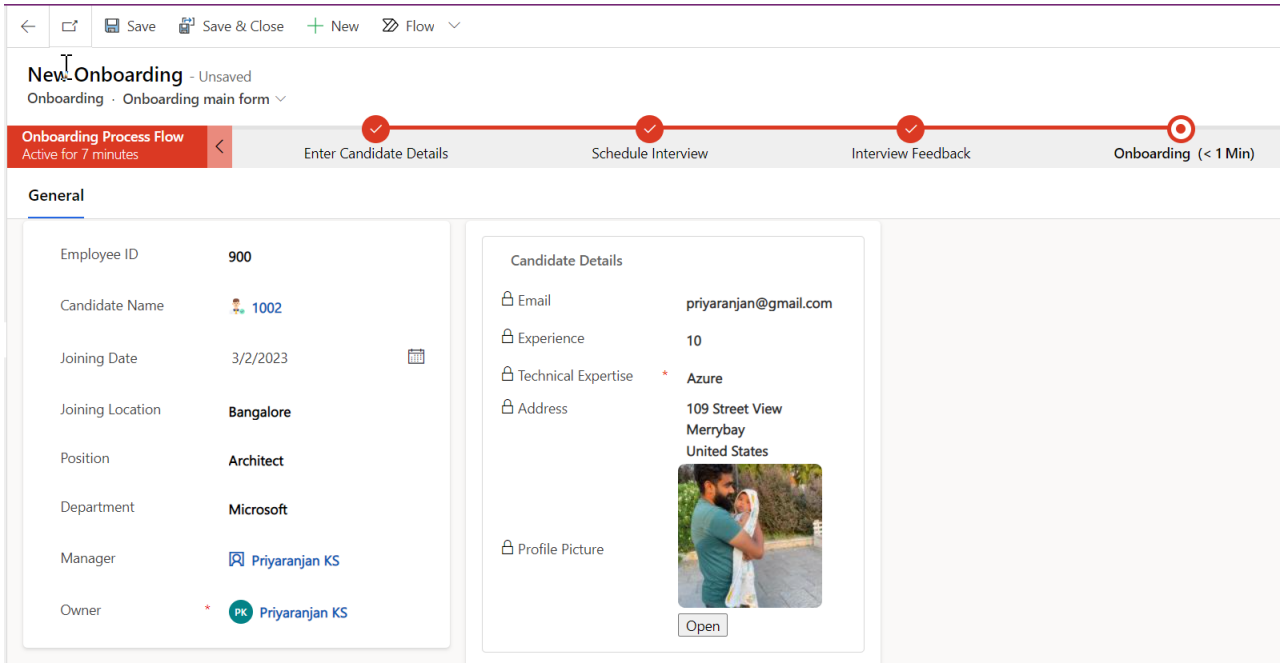
Enter the needed details as below



Click on Save and then move to the next stage



In the final stage of onboarding specify the onboarding details and click on Save to complete the process flow.



Summary

Thus we have seen how we can create a Model driven app by plugging in the various components that will make up the app. This E Book will be constantly updated and evolved with more advanced topics, stay tuned for more interesting reads.