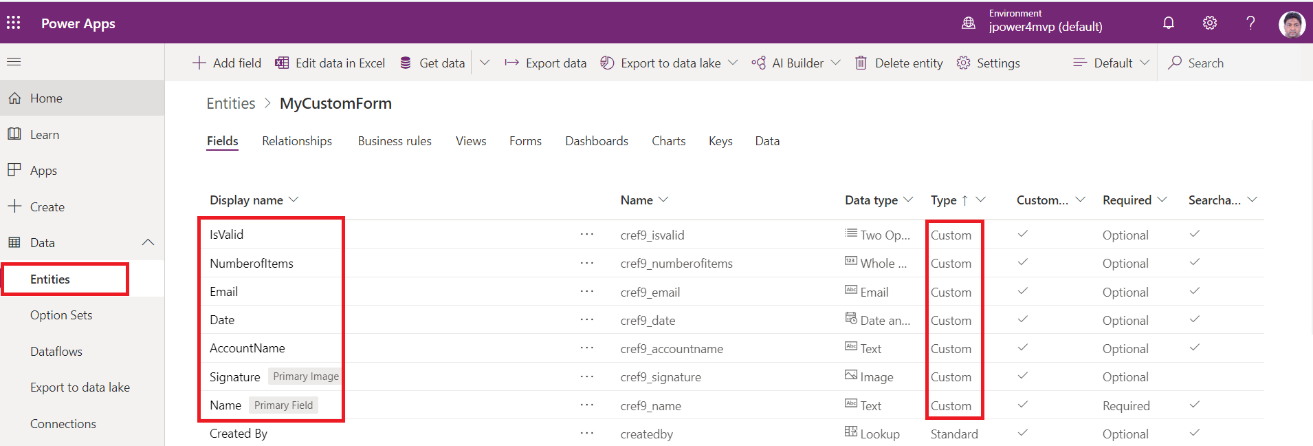
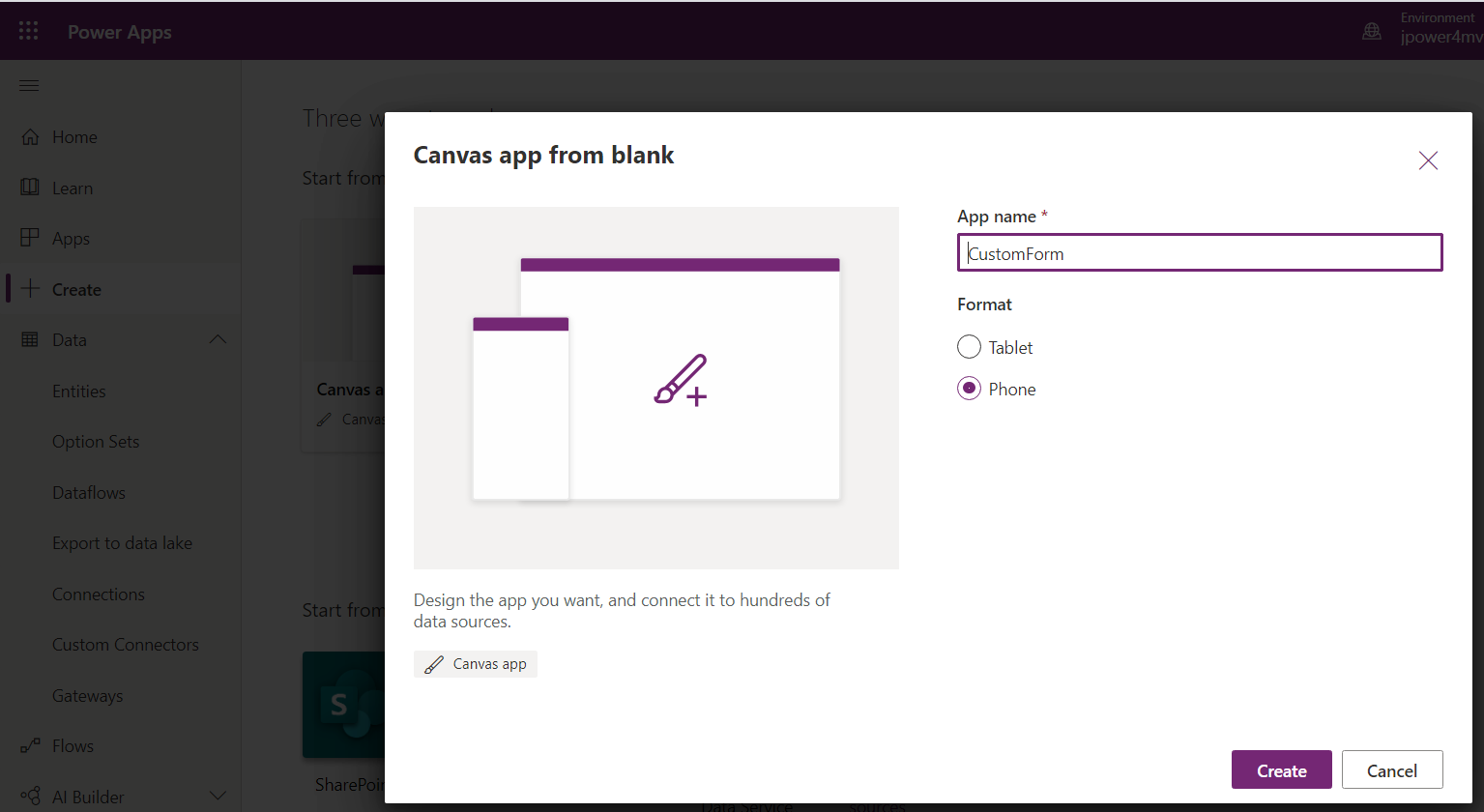
**Create the records in Common Data Service with the blank canvas app using Patch function**

Let’s create a Canvas app with few controls to create records in Dynamics CRM.

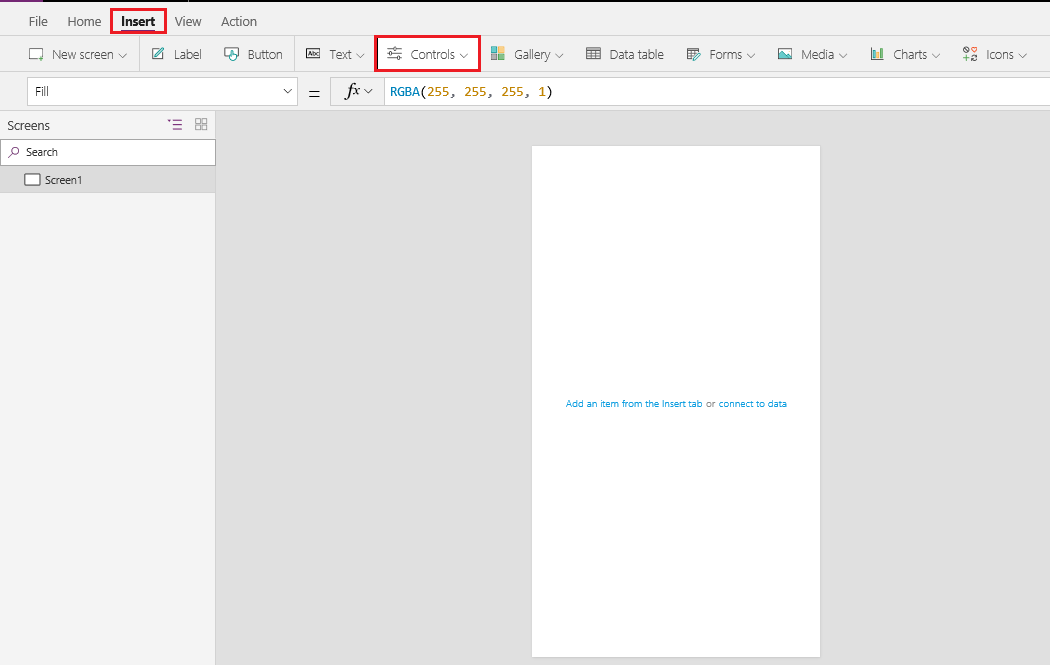
1. Sign in to the PowerApps
2. Below are the steps to create the records in Dynamics CRM
3. Here we are going to create the custom entity (**MyCustomForm**) record in CRM. You can use other entity as well.
4. Create a **MyCustomForm** Entity using common data service with below columns and refer the image below: -
   1. Name
   2. AccountName
   3. Email
   4. Signature
   5. Date
   6. IsValid
   7. NumberofItems



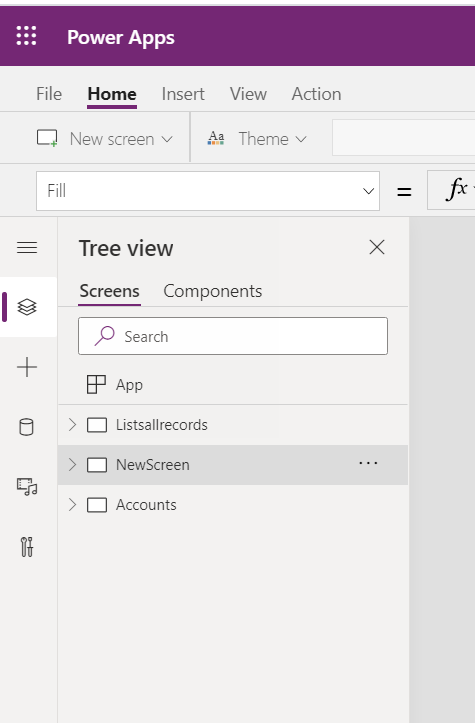
1. Click on Create an app and it will navigate to another screen. Then select Phone Layout under the blank Canvas blank.



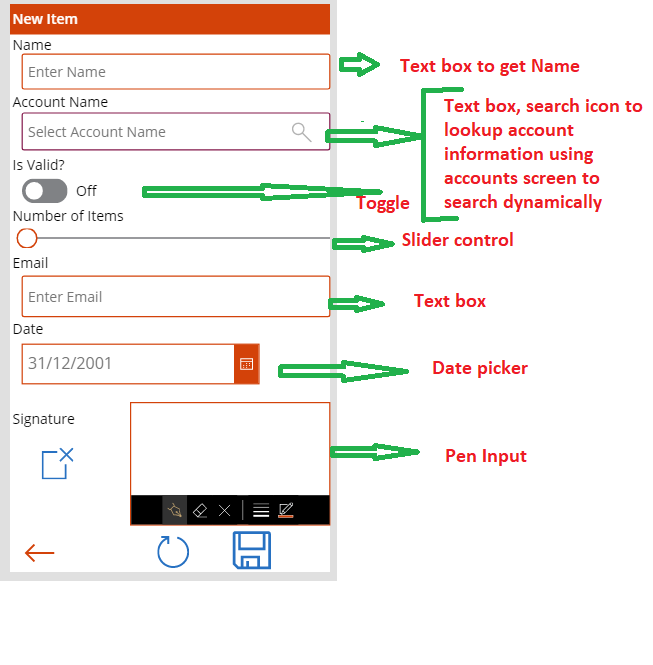
1. A blank screen will be appeared as below. You can rename it if you want,



1. I have created three Screens
   1. List all records
   2. New Item
   3. Accounts

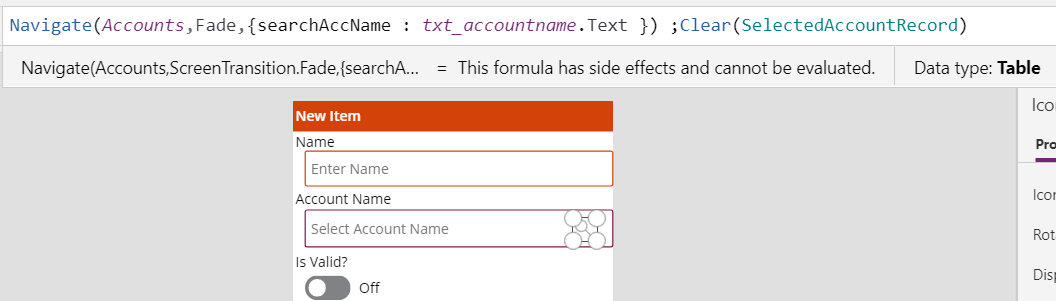


1. Add controls from the Insert Tab to the blank screen so that we can take input to save the record in CRM.



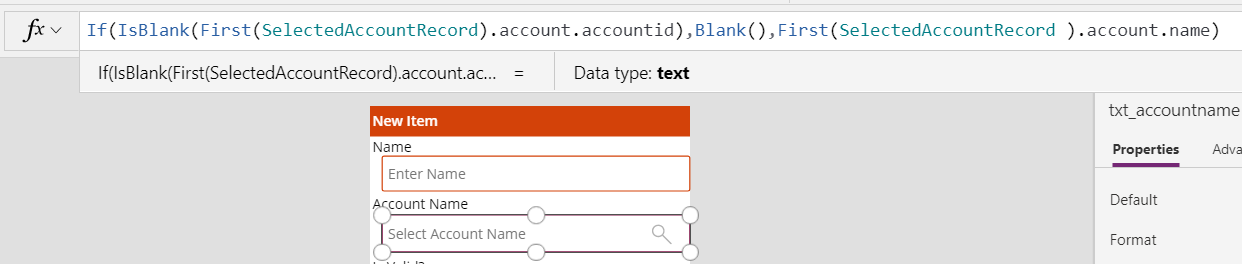
1. Set the **Min**, **Max**, **Default** property of slider control.
2. Configure the Account Lookup
   1. To set the Account lookup on mycustomform entity we added another screen to select the Account record
   2. To add blank screen: **Go to Insert -> New Screen -> select Blank Screen**.
   3. Add Gallery control to display the list of existing Account records in Gallery control.
   4. To add Gallery control; **Go to Insert -> select Gallery -> Select Vertical Gallery control** and layout Title alone
3. Now set the property for Account Lookup on textbox control
   1. Set **On Select** property of search icon to

*Navigate(Accounts,ScreenTransition.Fade,{searchAccName : txt\_accountname.Text }) ;Clear(SelectedAccountRecord)*

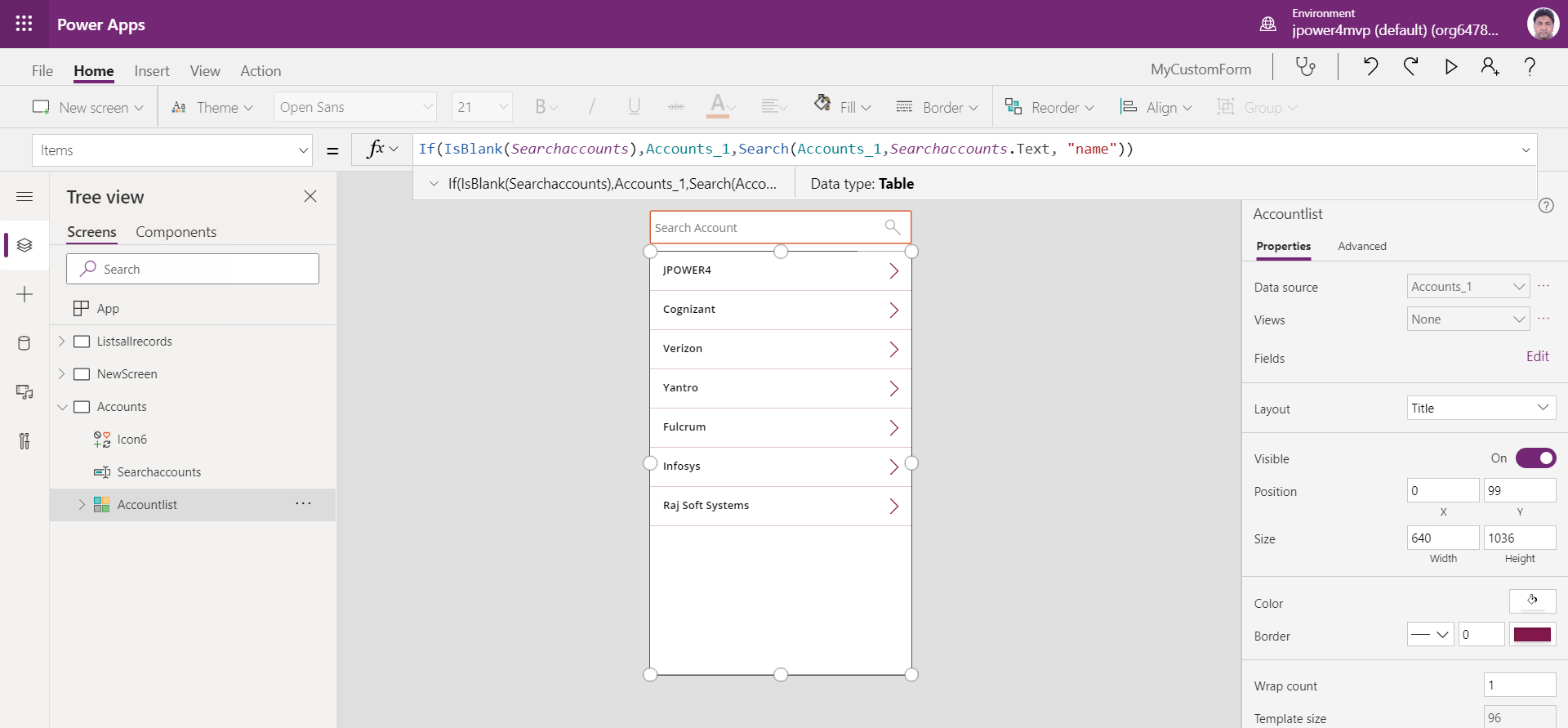


1. Here we are navigating the screen to select the Account record and passing the values **SearchAccName** of Select Account Textbox.
2. Set the **Default** property of Select Account textbox to

*If(IsBlank(First(SelectedAccountRecord).account.accountid),Blank(),First(SelectedAccountRecord ).account.name)*



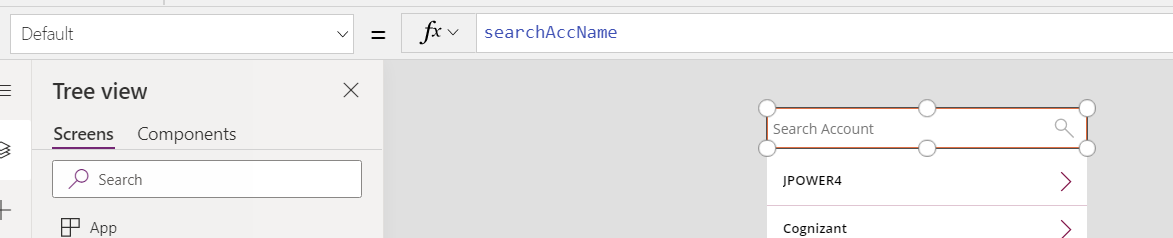
1. Here we are displaying the selected account name from the Account Gallery in the textbox. Initially it will display blank.
2. To select CRM Account Record, we added another blank screen.
   1. **Insert – > New Screen**
3. Now add Account data source: **Select view -> Data Source – > Select the connection and then add the Account Data Source**



1. Now set the Gallery Items Property to

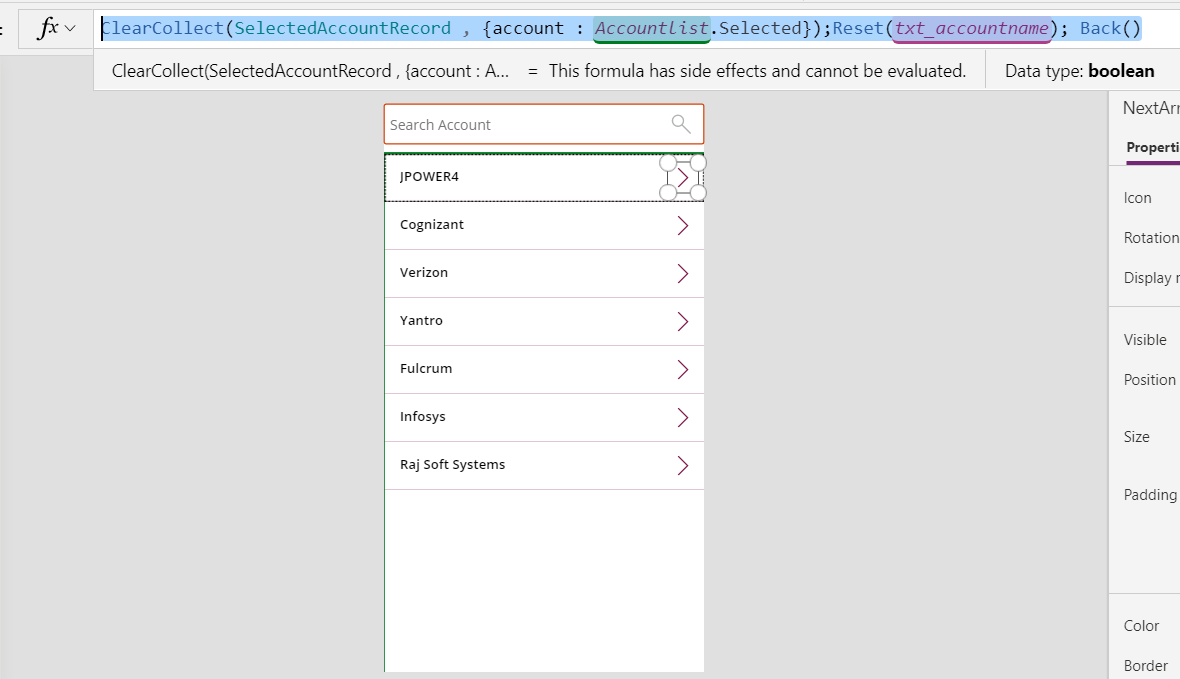
*If(IsBlank(Searchaccounts),Accounts\_1,Search(Accounts\_1,Searchaccounts.Text, "name"))*

* Here the Gallery will display the list of the accounts with matching name in **searchAccName** variable from previous screen. If **Searchaccounts** control is empty then it will show all accounts.

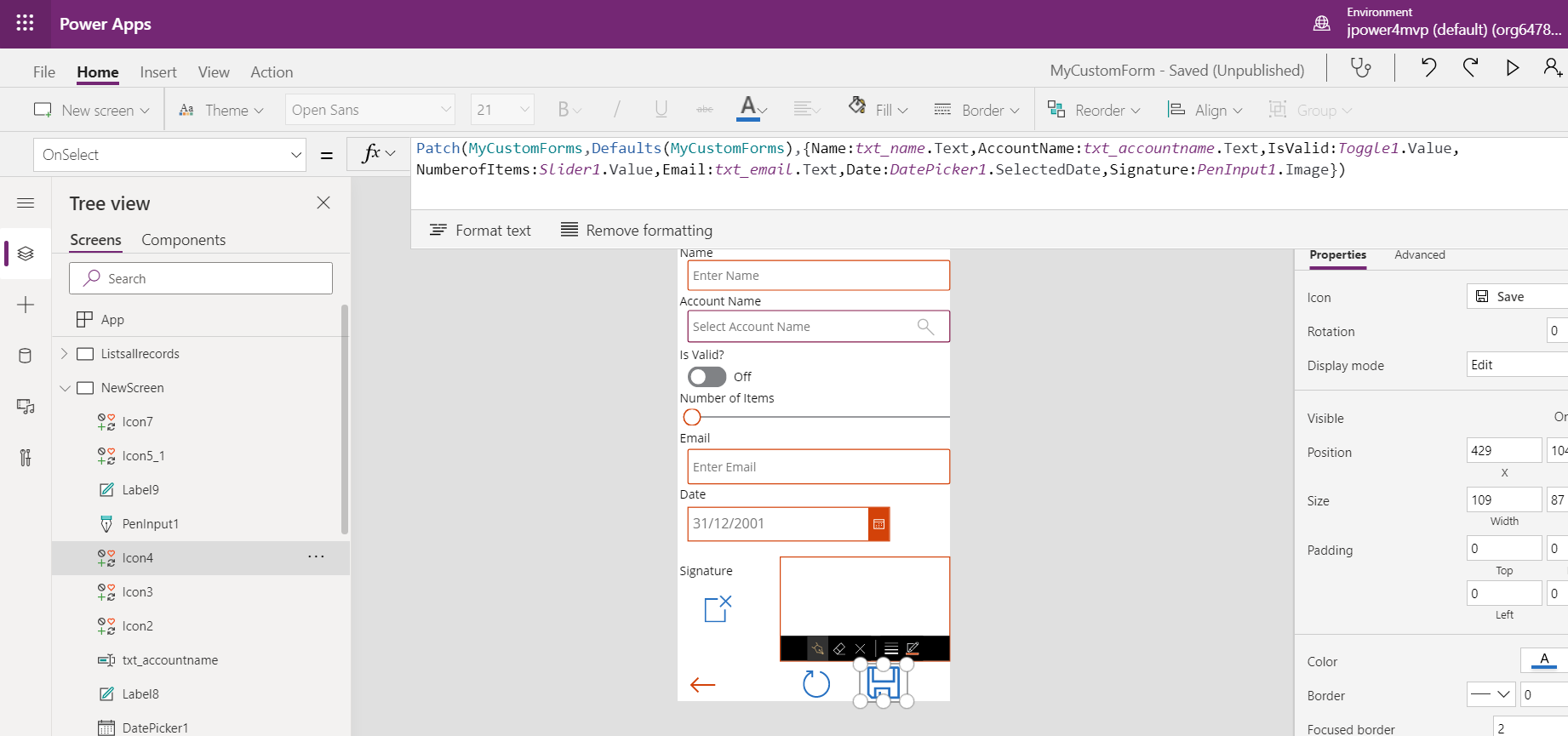


1. Set **On select** property of Arrow on Gallery to

*ClearCollect(SelectedAccountRecord , {account : Accountlist.Selected});Reset(txt\_accountname); Back()*

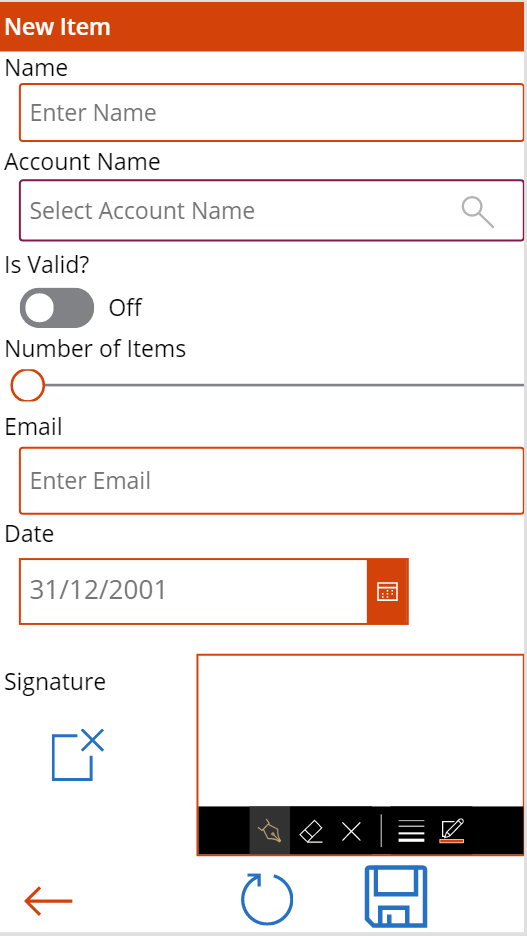


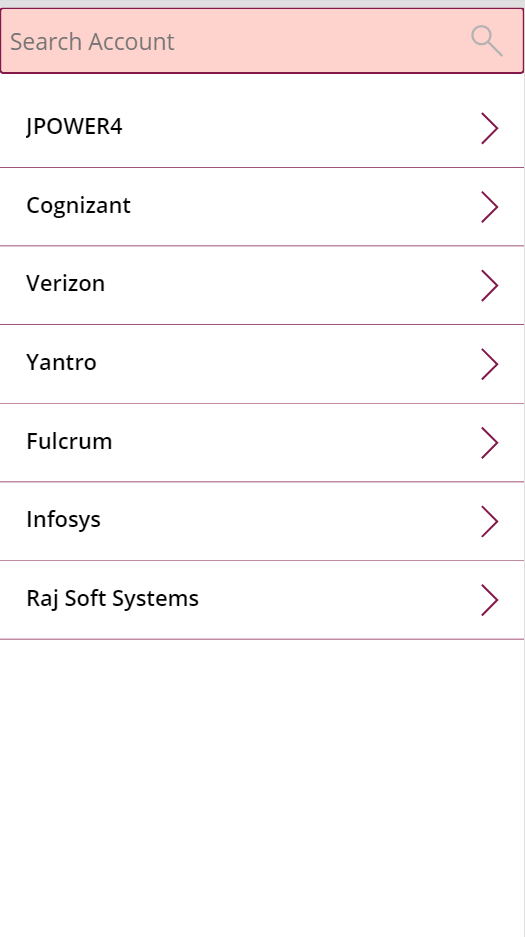
1. Here we are storing the selected account record in the separate collection i.e. SelectedAccountRecord.
2. Set the On select property of Save icon - Here we need to add the Data source for the Custom entity (myCustomForms)
   1. Set the On select property of **Save icon** to *Patch(MyCustomForms,Defaults(MyCustomForms),{Name:txt\_name.Text,AccountName:txt\_accountname.Text,IsValid:Toggle1.Value,NumberofItems:Slider1.Value,Email:txt\_email.Text,Date:DatePicker1.SelectedDate,Signature:PenInput1.Image})*
3. In Patch function we will set the values of controls to the myCustomForms data source columns.



1. Now save and publish the app to reflect the changes on mobile device. Now play the App from PowerApp Studio as you can see in the below screenshot:

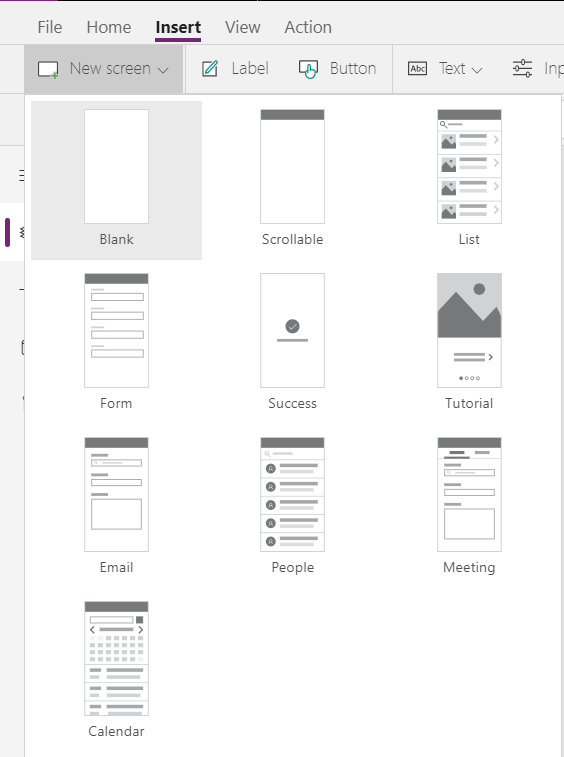
Output



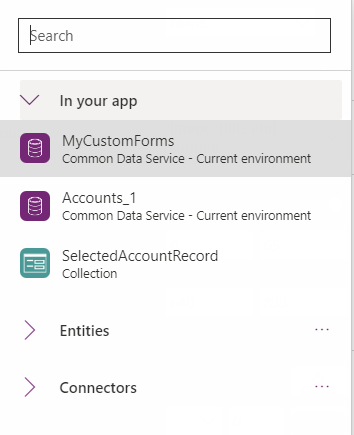


1. Create New screen to display submitted MyCustomForms

**Insert – > New Screen**



1. Now add Account data source: **Select view -> Data Source – > Select the connection** and then add the **myCustomForm** Data Source



1. Now add Gallery Control and select data source as **myCustomForms**
2. **Output – Data display from myCustomForms entity**

