**\*\*\* Tabbing \*\*\***

Create 2 buttons ad TAB1 and TAB2, than create a rectangular box which will cover below area of both buttons

=========

create 2 lables (we can create more lable,button,gallery etc.) for TAB1, select both lables by ctrl key than select event **visible** and put variable **tab1** in function.

Similaraly create 2 lables (we can create more lable,button,gallery etc.) for TAB2, select both lables by ctrl key than select event **visible** and put variable **tab2** in function.

=========

Select 1st button TAB1, select event **OnSelect** than put below code in function –

UpdateContext

({

tab1:true,bgtab1:RGBA(255,255,255,100),colortab1:RGBA(0, 0, 0, 1),

tab2:false,bgtab2:RGBA(56, 96, 178, 1),colortab2:RGBA(255, 255, 255, 1)

})

Select 1st button TAB1, select event **Color** than put variable **colortab1** code in function.

Select 1st button TAB1, select event **Fill** than put variable **bgtab1** code in function.

=========

Similaraly Select 2nd button TAB2, select event **OnSelect** than put below code in function –

UpdateContext

({

tab1:false,bgtab1:RGBA(56, 96, 178, 1),colortab1:RGBA(255, 255, 255, 1),

tab2:true,bgtab2:RGBA(255,255,255,100),colortab2:RGBA(0, 0, 0, 1)

})

Select 2nd button TAB2, select event **Color** than put variable **colortab2** code in function.

Select 2nd button TAB2, select event **Fill** than put variable **bgtab2** code in function.

=========

Select screen than select event **OnVisible** and put below code in function :

UpdateContext

({

tab1:true,bgtab1:RGBA(255,255,255,100),colortab1:RGBA(0, 0, 0, 1),

tab2:false,bgtab2:RGBA(56, 96, 178, 1),colortab2:RGBA(255, 255, 255, 1)

})

**Note**: UpdateContext() method creates Local variables. Its scope remains a particular screen

**Some example of tabbing code:**

updateContext //on screen visible

({

varDeleteDepartment:false,varDeleteSubDepartment:false,varDeleteDocumentType:false,varDeleteLevels:false,varDeleteRoles:false,varDeleteBusiness:false,varDeleteLocation:false,varDeletePermission:false,varDeleteEmailTemplate:false,varWhiteBground:RGBA(255,255,255,100),varGreenBground:RGBA(197,227,182,100),varBlueColor:RGBA(0,139,210,100),varBlackColor:RGBA(51,51,51,100)

});

UpdateContext //on selection of tabbing

({

varGrpDepartment:true,bgroundDepartment:varWhiteBground,colorDepartment:varBlueColor,

varGrpSubDepartment:false,bgroundSubDepartment:varGreenBground,colorSubDepartment:varBlackColor,

varGrpDocumentType:false,bgroundDocumentType:varGreenBground,colorDocumentType:varBlackColor,

varGrpLevels:false,bgroundLevels:varGreenBground,colorLevels:varBlackColor,

varGrpRoles:false,bgroundRoles:varGreenBground,colorRoles:varBlackColor,

varGrpBusiness:false,bgroundBusiness:varGreenBground,colorBusiness:varBlackColor,

varGrpLocation:false,bgroundLocation:varGreenBground,colorLocation:varBlackColor,

varGrpPermission:false,bgroundPermission:varGreenBground,colorPermission:varBlackColor,

varGrpEmailTemplate:false,bgroundEmailTemplate:varGreenBground,colorEmailTemplate:varBlackColor

});

**\*\*\* Direct Navigate to Another Screen \*\*\***

Create a button(we can create any thing), select button, select event **OnSelect** and put **Navigate(**Screen2**)** in function where **Screen2** is the screen name where we want to navigate.

**\*\*\* Navigate to Another Screen By Component \*\*\***

**1st method:**

In component section, select a particular component, in right section under properties tab **Access app scope** must be on, than apply Navigate() function to every menu/button etc. In screen section, select a particular screen, click on Insert menu, than select the created component.

**2nd method:**

In component section, select a particular component, in right section under properties tab **Access app scope** must be on, than click on **New custom property**, set display name, set Property Type **Input** and Data Type **Screen,** than click on setted display name of screen under New custom properties and put screen name in function where we want to navigate, than similaraly create custom properties for all screen and put screen name in function.

(Getting some issue)

**\*\*\* Show Popup \*\*\***

Create a button(we can create any thing), as show popup, select button, select event **OnSelect** and put below code in function:

**UpdateContext({popup:true})**  where popup is variable.

Than create a popup with background, dialog box, lable, ok button, cancel button, close button etc.

(we can create circle button height:40px, width:40px and border-radius:20px, i.e. height and width will same and border radius will half.)

Select all popup components by ctrl key than select event **Visible** and put variable **popup** in function.

Select close button , select event **OnSelect** and put below code in function:

**UpdateContext({popup:false})**  where popup is variable.

Than select screen, select event **OnVisible** and put below code in function:

**UpdateContext({popup:false})**  where popup is variable.

**\*\*\* If condition \*\*\***

If(condition1, Statement1)

**\*\*\* If-else condition \*\*\***

If(condition1, Statement1, else statement)

**\*\*\* Multiple If condition \*\*\***

If(condition1, Statement1, condition2, Statement2, condition3, Statement3)

**\*\*\* Multiple If condition with else part\*\*\***

If(condition1, Statement1, condition2, Statement2, condition3, Statement3, else statement)

**\*\*\* Set() function \*\*\***

Set(varA, "Welcome"); Set(varB, "User");

Where **varA** and **varB** both are variables while **Welcome** and **User** both are values. We can use double/both function by terminating ;

**Note**: Set() function creates Global variables. Its scope remains entire Apps.

**\*\*\* Current Date \*\*\***

Now() **OR**  Text(Now(),DateTimeFormat.LongDate)

**\*\*\* Good Morning, Afternoon, Good Evening \*\*\***

If(Hour(Now())<12, "Good Morning,", Hour(Now())<17, "Good Afternoon,", "Good Evening,")

**\*\*\* Navigate screen by Component \*\*\***

https://www.youtube.com/watch?v=Ve7cTm66XVo // reference for navigate by component

**\*\*\* User Profile Details \*\*\***

User().FullName //For User name

User().Email //For User email

User().Image // For profile image

**\*\*\* No Record Found if Gallary doesn’t have record \*\*\***

Select event **Visible** and put below code in function

If(CountRows(glyDocumentList.AllItems)=0,true,false)

Where **glyDocumentList**  is gallery name

**\*\*\* Some conditions for form mode \*\*\***

Set(formModeEmail,"Edit"); Set(formModeEmail,"New");

If(formModeEmail = "New",New,formModeEmail = "Edit",Edit) // for form mode

**\*\*\* CRUD Operation(Add,Show Records,Edit,Delete) with Validation Using form\*\*\***

Create Tables(Entity) ,

Than goto insert >> Edit form >> select the form >> Select Entity by Data Source under properties in right section.

Than select the form>>click on “Edit Fields” under properties in right section >> we can add new field(column) by searching, Remove field, add custom card for creating space(click on three dots>>Add new custom card) and change order of existing fields(by drag & drop or click on three dots of field>>move down).

**Validation:**

Select the field >> select event “OnChange” and put below code in function:

If(

IsBlank(DataCardValue5\_1.Text),

UpdateContext({varDepartment: "Required fields must be filled in."}),

UpdateContext({varDepartment:""})

)

Where **DataCardValue5\_1** is name of input field and **varDepartment** is a new declared variable.

Than, in left side select “ErrorMessage” field (abobe the input field) >> Select event “Text” and put **varDepartment** variable in function. It will same process for all fields.

**Validation code for PAN number:**

Select the field >> select event “OnChange” and put below code in function:

If(IsBlank(DataCardValue5.Text) || IsEmpty(DataCardValue5.Text),UpdateContext({varpan:"Required Field"}),  
      !IsBlank(DataCardValue5.Text) && IsMatch( DataCardValue5.Text,  
      "[A-Z]" & "[A-Z]" & "[A-Z]" & "[A-Z]" & "[A-Z]" & Digit & Digit & Digit & Digit & "[A-Z]"),  
UpdateContext({varpan:""}),  
UpdateContext({varpan:"Pan format is not correct"}))

Where **DataCardValue5** is name of input field and **varpan** is a new declared variable.

Than, in left side select “ErrorMessage” field (abobe the input field) >> Select event “Text” and put **varpan** variable in function. It will same process for all fields. Code will be change.

**Validation code for Mobile number:**

Select the field >> select event “OnChange” and put below code in function:

If( IsBlank(DataCardValue6.Text),  
    UpdateContext({varemError: "Required field must be field in."}),

!IsMatch(DataCardValue6.Text, Digit&Digit&Digit&Digit&Digit&Digit&Digit&Digit&Digit&Digit),  
  
    UpdateContext({varemError: "Number should be of ten digits"}),  
    UpdateContext({varemError: ""})  
)

Where **DataCardValue6** is name of input field and **varemError** is a new declared variable.

Than, in left side select “ErrorMessage” field (abobe the input field) >> Select event “Text” and put **varemError** variable in function. It will same process for all fields. Code will be change.

**Validation code for Email Id:**

Select the field >> select event “OnChange” and put below code in function:

If(

    !IsMatch(Text(DataCardValue7.Text),Email),

      UpdateContext({varemail:"email format is not correct"}),

          UpdateContext({varemail:" "})

)

Where **DataCardValue7** is name of input field and **varemail** is a new declared variable.

Than, in left side select “ErrorMessage” field (abobe the input field) >> Select event “Text” and put **varemail** variable in function. It will same process for all fields. Code will be change.

**Validation code for Adhar number:**

Select the field >> select event “OnChange” and put below code in function:

If( IsBlank(DataCardValue8.Text),

    UpdateContext({varaaerr: "Required field must be field in."}),     !IsMatch(DataCardValue8.Text, Digit&Digit&Digit&Digit&Digit&Digit&Digit&Digit&Digit&Digit&Digit&Digit),

    UpdateContext({varaaerr: "AAdhar should be of valid"}),

    UpdateContext({varaaerr: ""})

)

Where **DataCardValue8** is name of input field and **varaaerr** is a new declared variable.

Than, in left side select “ErrorMessage” field (abobe the input field) >> Select event “Text” and put **varaaerr** variable in function. It will same process for all fields. Code will be change.

**Validation code for GST number:**

Select the field >> select event “OnChange” and put below code in function:

If(IsBlank(TextInput3.Text) || IsEmpty(TextInput3.Text),UpdateContext({vargst:"Required Field"}),

      !IsBlank(TextInput3.Text) && Value(Left(TextInput3.Text,2))<=35 || IsMatch( TextInput3.Text, "[A-Z]" & "[A-Z]" & "[A-Z]" & "[A-Z]" & "[A-Z]" & Digit & Digit & Digit & Digit & "[A-Z]" & Digit & "[Z]" & Digit),

UpdateContext({vargst:""}),

UpdateContext({vargst:"Gst number is not correct"})

)

Where **TextInput3** is name of input field and **vargst** is a new declared variable.

Than, in left side select “ErrorMessage” field (abobe the input field) >> Select event “Text” and put **vargst** variable in function. It will same process for all fields. Code will be change.

**Add(Submit form):**

Create a submit button >> select the button >>select event “OnSelect” and put below code in function:

If(

IsBlank(DataCardValue5\_1.Text),

UpdateContext({varDepartment: "Required fields must be filled in."}),

IsBlank(DataCardValue6\_1.Text),

UpdateContext({varDepartmentCode: "Required fields must be filled in."}),

SubmitForm(formDepartment);

ResetForm(formDepartment);

Notify("New Department Added Sucessfully",NotificationType.Success);

)

**Note**: This is a form submission code for two fields.

Where **DataCardValue5\_1** and **DataCardValue6\_1** are the names of input field , **varDepartment** and **varDepartmentCode**  are new declared variables , **formDepartment** is a name of form and "Required fields must be filled in." is a error message.

It will same process for more fields.

**Show Records:**

Create gallery (Goto insert >> select vertical/horizontal galllary) >> select the gallery >> in right side under properties tab, select Data source(Entity) by searching.

In gallery, in first row, insert lables according to entity(column) and also insert view details, edit and delete icon/image >> than select every lable one by one >> select event “text” and put **ThisItem.Title** in function where **Title** is a column name.

Similaraly we can put below codes:

**Text(ThisItem.Created,"dd-mmm-yyyy")**  where **Created** is a colum name to display date.

**View Details:**

1st Method (By using Form):

To show all/more colums data, we create a popup, then create the form in edit mode ,then add data source, then add all/more colums by using “Edit Fields” option than Select the form >> select event “Items” then write below code in function

**Gallary.Selected**

Where **Gallary** is the name of Gallary where all records are showing.

2nd Method (Without using Form):

To show all/more colums data, we create a popup, then create 2 lable (one for heading and second for data), then we put heading on first lable then select second lable >> select event “**Text**” then put below code in function:

**Gallary.Selected.Name**

Where **Gallary** is name of the **Gallary** where all records are showing and **Name** is the colum name of table.

**Edit:**

**Note:**

We will play two buttons “Add” and “Update” and will display condition basis.

Select screen >> Select event “OnVisible” >> put **Set(formModeDepart,"New");**  in function, where **formModeDepart** is a new declared variable.

Select edit icon on gallery first row >> select event “OnSelect” >> put **Set(formModeDepart,"Edit");** in function

Select the form >> Select event “DefaultMode” and put **If(formModeDepart = "New", FormMode.New,formModeDepart = "Edit", FormMode.Edit)** in function.

Select the form >> Select event “Item” and put below code in function

**glyDepartment.Selected**

OR

**First(DropColumns(Table(**glyDepartment**.Selected),**"varRownum"**))**

OR

**LookUp(**Registrations**,** UserEmail= glyDepartment**.Selected.**UserEmail**)**

Where **glyDepartment** is gallery name, **Registrations** is table name, **UserEmail** is colom name , **varRownum**  is self defined variable on gallery item by using With() method, for details With() method, you can read about With() method in this file.

Select the update button >> select event “OnSelect” and put below code in function:

If(

IsBlank(DataCardValue5\_1.Text),

UpdateContext({varDepartment: "Required fields must be filled in."}),

SubmitForm(formDepartment);

ResetForm(formDepartment);

Notify("Department Updated Sucessfully",NotificationType.Success);

UpdateContext({varAddDeptForm:true,varUpdateDeptForm:false}); //To display Add and Update button on condition basis

Set(formModeDepart,"New");

)

**Note:**

UpdateContext({varAddDeptForm:false,varUpdateDeptForm:true});

To display “Add” and “Update” button on condition basis. (Set variables on Visible event of buttons)

**Delete:**

By clicking on delete button show popup with Delete and Cancel button.

Select Delete button >> select event “OnSelect” and put below code in function:

**Remove('Master Department standards',glyDepartment.Selected);** //To Remove record

**Notify("Selected Department Deleted Succesfully",NotificationType.Success);** // To show Notification

**UpdateContext({varDeleteDepartment:false});** // To hide popup

Where **Master Department standards** is table(entity) name and **glyDepartment** is gallery name.

**\*\*\* Fill data in combo box \*\*\***

1st Method (To show default colom data):

Select combo box, select event “Items” and put table name in function.

2nd Method (To show perticular colom data or Choice field data):

Create form , add source(attach table) , add colums by edit field option then select particular dropdown field or combo box field or choice field >> selelct event “items” >> copy the code which is wtitten in function.

------------------

Select the combo box (where we want to show particular colom data or choice field data) >> select event “Items” and put the code in function which was copied.

**\*\*\* Fill data in dropdown\*\*\***

To show perticular colom data or Choice field data

Create a dropdown >> select the dropdown >> select the event ‘items’ >> put below code in function:

**Choices(SubjectNew)**

Where Choices() is predefined function and SubjectNew is choice field name which is created by developer in dataverse.

**\*\*\* Search \*\*\***

Syntax:

Search(Table\*, SearchString, Column1 [, \*Column2\*, ... ] )

Note: it will show all related data.

**\*\*\* Filter \*\*\***

We can show another table’s multiple colum data in our current table. It will show exact data and lookup function also show exact data.

**>>Filter By text field (For a single column , data type must be ‘Single line of text’):**

Select gallery, select event **Items** and put below code in function:

Filter(Registrations , StartsWith(UserEmail,search.Text))

**Note**:

1. Where **Registrations** is table name, **UserEmail** is column name and **search** is text field name.
2. It is case sensitive.
3. .Text is valid if data type of colum is ‘Single line of text’

**>>Filter By text field (For double/multiple multiple column , data type must be ‘Single line of text’):**

Select gallery, select event **Items** and put below code in function:

Filter(Registrations , StartsWith(UserEmail,search.Text) || StartsWith(Name,search.Text))

**Note**:

1. Where **Registrations** is table name, **UserEmail** is colum name , **Name** is column name and **search** is text field name.
2. It is case sensitive.
3. .Text is valid if data type of colum is ‘Single line of text’

**>>Filter By text field (For a particular column , data type must be ‘Single line of text’) and single combobox:**

Filter(

    Registrations,

    StartsWith( UserEmail,search.Text),

     IsBlank(ComboBox1.Selected.Value)||IsEmpty(ComboBox1.Selected.Value)|| UserInterest = ComboBox1.Selected.Value

     )

Where **Registrations** is table name, **UserEmail** is colom name(where we want to search email), **search** is text field name, **ComboBox1** is combo box name and **UserInterest** is colum name (where we want to apply filter).

**>>Filter By text field (For a particular column , data type must be ‘Single line of text’) and multiple combobox:**

Filter(  
    Registrations,  
        StartsWith( UserEmail,search.Text),

IsBlank(ComboBox1.Selected.Value)||IsEmpty(ComboBox1.Selected.Value)|| empBloodGroup.Value = ComboBox1.Selected.Value,  
     IsBlank(ComboBox2.Selected.Value)||IsEmpty(ComboBox2.Selected.Value)||empLocation.Value = ComboBox2.Selected.Value  
)

Where **Registrations** is table name, **UserEmail** is colom name(where we want to search email), **search** is text field name, **ComboBox1** is first combo box name and **empBloodGroup** is colum name (where we want to apply filter), **ComboBox2** is second combo box name and **empLocation** is colum name (where we want to apply filter).

**>>Filter By text field (For a single column , data type must be ‘choice field’):**

**1st Method**

Filter(AddColumns(Registrations,'newcol','ColorField'),StartsWith( newcol, *txtsearch*.Text))

Where **Registrations** is table name, **newcol** is user defined colom name(we can put any name), **ColorField** is a colom of table (data type is choice), txtsearch is name of text input field and rest is predefined keywords.

**>>Filter By text field (For a 2 column , data type must be ‘Single line of Text’ and 2nd is choice field):**

**1st Method:**

Filter(Registrations,StartsWith( Name, *txtsearch*.Text),'ColorField'= *cmbSearch*.Selected.Value)

Where **Registration** is table name, **Name** is 1st colum name with data type ‘Single line of Text’, **txtsearch** is name of text input field, **ColorField** is 2nd Colum name with data type choice field, **cmbSearch** is name of combobox field and rest is predefined.

Note:

1. We need to create one text input field and one combo box field.
2. We need to fill data in combo box field by using choice field.

**2nd Method:**

Filter

(

    AddColumns(Registrations,'newcol','ColorField'),

    StartsWith( Name, *txtsearch*.Text) || StartsWith( newcol, *txtsearch*.Text)

)

Where **Registrations** is table name, **newol**  is user defined colom (we can put any name), **ColorField** is name of colum with data type choice field, **Name** is name of colom with data type ‘Single line of Text’, **txtsearch** is name of text input field and rest is predefined keywords.

**\*\*\* Responsive View \*\*\***

Go to setting from bottom left bar >>Display / Orientation>> Turn off ‘Scale to fit’ and Turn off ‘Lock Orientation’>>

Click on Apply button (If appeared).

--------------------------

Select event as per requirement and put below code in function:

If( App.ActiveScreen.Size = 3 || App.ActiveScreen.Size = 4, true,false)

**Note**: 3 for Tablet view , 4 for Laptop view, 1 & 2 for Mobile view

-------------------------

**\*\*\* Collection \*\*\***

**Create Collection:**

Insert three (according to requirement) text input field, Add three buttons. (first for adding collection, second for adding current record rest will remove and third for removing collection )

1. Select first button, select event **onSelect**, then put below code in function to add record by Collect method in collection:

Collect(colEmpdat,{EmpName:TextEmpName.Text,Salary:TextSalary.Text,Designation:TextDesignation.Text});

**Note**:

Where **colEmpdat** is self defined collection name, **EmpName , Salary** and **Designation** is self defined colom name,

**TextEmpName** is the name of first text input field, **TextSalary** is the name of second text input field and **TextDesignation** is the name of third text input field.

1. Select second button, select event **onSelect**, then put below code in function to add record by Patch Function in collection:

Patch(colEmpdat,Defaults(colEmpdat),{EmpName:"New",Salary:"45000",Designation:"Clerk"});

**Note**:

Where **colEmpdat** is self defined collection name, **EmpName , Salary** and **Designation** is self defined colom name,

“New”,”45000” and “Clerk” are static values.

1. Select second button, select event **onSelect**, then put below code in function to add current record and rest will remove:

ClearCollect(colEmpdat,{EmpName:TextEmpName.Text,Salary:TextSalary.Text,Designation:TextDesignation.Text});

1. Select third button, select event **onSelect**, then put below code in function to delete all records in collection:

Clear(colEmpdat);

**Display Collection Data (in Gallary):**

Insert a gallery, select the gallery , select event “Items” and Put / Select collection name in function then it will show collection data in gallery. We can also put colom name in first row of gallary like: ThisItem.Designation where Designation is self defined colom name.

**Display Collection Data (in Dropdown):**

Insert a dropdown, select the dropdown, select event “Items” and Put / Select collection name in function then it will show collection data in dropdown.

**Edit Collection Data (in Gallary):**

Insert a text input field, select the text input field, select event **Default** and put **ThisItem.EmpName** in function, where EmpName is colom name of collection, than select event **OnChange** and put **Patch(colEmpdat,ThisItem,{EmpName:TextInput2.Text})** in function, where **colEmpdat** is collection name, **EmpName** is colom name and **TextInput2** is name of input field which was inserted in gallary.

Run the file, when we will change in text input field , it will update to particular colom in collection. Similaraly ,we can update other coloms.

**\*\*\* High light active row of gallary \*\*\***

Insert a image in first row of gallery, it will show in all row >> Select the image , right click on image then select **Reorder** than select **send to back.**

To hight light active row:

Select the image >> select event **Visible** >> put below code in function:

If(ThisItem.IsSelected,true,false)

**\*\*\* Alternate background row colors of gallary \*\*\***

Insert a image in first row of gallery, it will show in all row >> Select the image , right click on image then select **Reorder** than select **send to back.**

To show background color for alternate row:

Select the image >> select event **Fill** >> put below code in function:

If(Mod(ThisItem.ID,2)=1,RGBA(56, 96, 178, 1),RGBA(253, 222, 207, 1))

OR

If(Mod(Value(ThisItem.ID),2)=1,RGBA(56, 96, 178, 1),RGBA(253, 222, 207, 1))

**Note**:

1. Where **ID** is a colom name which data type is **Autonumber**.
2. Mod() function is used to get module(reminder) of ID, which is devided by 2.

**\*\*\* Alternate background row colors of gallery with Filter() method by using With() method\*\*\***

Select the gallery, select event **items** and put below code in function:

With({varRecords: Filter(Registrations, StartsWith(UserEmail, search.Text),

IsBlank(ComboBox1.Selected.Value) || IsEmpty(ComboBox1.Selected.Value) || UserInterest = ComboBox1.Selected.Value)},

ForAll(Sequence(CountRows(varRecords)), Patch(Last(FirstN(varRecords,Value)), {varRownum: Value})))

==========

Then select the gallery , select event **TemplateFill ,** and put below code in function:

If(Mod( ThisItem.varRownum,2)=1, Color.Red,Color.Blue)

We can also use image instead of **TemplateFill** event and set condition on **Visibility**.

**Note**:

1. Where **varRecords** is selft defined variable, **Registrations** is table name, **UserEmail** is colom name, **search** is name of text/search input field, **ComboBox1** is name of combobox, **UserInterest** is colom name, **Value** is predefined variable,

And **varRownum** is self defined variable which stores serial no. of row.

1. ForAll() method is providing default behaviour of table.
2. If we want to use without Filter() method then we will use below code:

With({varRecords: Registrations},

ForAll(Sequence(CountRows(varRecords)), Patch(Last(FirstN(varRecords,Value)), {varRownum: Value})))

**\*\*\* Distinct() Method \*\*\***

To stop repeatation for value of a particular colom , we use Distinct() method, it takes two parameter first is table name and second is colom name.

Exp:

**Distinct(**Country , Country\_Name**);**

Where **Country** is table name and **Country\_Name** is colom name.

**\*\*\* Sort() Method \*\*\***

**\*\*\* SortByColumns() Method \*\*\***

**\*\*\* With() Method \*\*\***

In this function we can declare variable one time and use it multiple times with in With() function.

**Example**:

With({varD:User()},"Name is "&varD.FullName &"Email is"&varD.Email &"Name is"&varD.FullName)

============

We can declare multiple variables:

**Example**:

With({varD:User(),varM:5},"Name is "&varD.FullName &"Email is"&varD.Email &"Name is"&varD.FullName&varM)

**Note**:

We can not use the variable in outside of With() method. Its scope remains a particular block or section.

**\*\*\* Unique Validation for a Particular colom\*\*\***

Select event **OnChange** and put below code in function:

If(  
    CountRows(Filter(Helpdesks,'Created By'.'Primary Email' = cUserName,Category = DataCardValue6\_1.Selected.Value))>0,  
    UpdateContext({varCategory : "You already have a request for this category. Kindly change the category and try again"}),  
    UpdateContext({varCategory:""})  
)

Than, Select error variable, select event text and put the variable **varCategory** in function.

Note:

1. **Helpdesks** is table name, **Created** **By** and **Primary** **Email** are colom name, **cUserName** is a variable in which login user name is stored, **Category** is colom name , **DataCardValue6**\_**1** is the name of field and **varCategory** is a self defined variable.

**\*\*\* Format Validation for attachment\*\*\***

Select event **OnAddFile** and put below code in function:

If(  
    ".pdf" in Concat(DataCardValue19\_1.Attachments.Name,Name&",") ||  
    ".png" in Concat(DataCardValue19\_1.Attachments.Name,Name&",") ||  
    ".jpeg" in Concat(DataCardValue19\_1.Attachments.Name,Name&",")||  
    ".jpg" in Concat(DataCardValue19\_1.Attachments.Name,Name&","),  
            UpdateContext({varAttachmenterror:""}),  
            UpdateContext({varAttachmenterror:"Only .jpg, .jpeg, .png, & .pdf document type is allowed."})  
)

Than, Select error variable, select event text and put the variable **varAttachmenterror** in function.

**\*\*\* HTML FORMAT\*\*\***

"<div style='background-color: rgba(0,0,0,0); box-shadow: 1px 2px 5px 0px rgba(89,87,87,0.25); margin: 10px; width: " & HtmlText1\_31.Width - 20 & "px; height:" &HtmlText1\_31.Height - 25 & "px; border-radius: 10px'>"& "<table style='border-collapse: collapse; width: 100%;'>  
<tr style=' background:#FFFFFF;'>  
<td style='border: 1px solid #fff; padding:1px; width:110px; height: 35px; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>Cancellation Type</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:333px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>--</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:110px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>Payment type</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:333px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>NEFT</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:110px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>Medical Charges</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:333px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>--</span></td>  
</tr>  
<tr style=' background:#FFFFFF;'>  
<td style='border: 1px solid #fff; padding:1px; width:110px; height: 35px; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>Partial Withdrawal Amount</span></td>  
<td style='border: 1px solid #fff; padding:1px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>--</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:110px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>  
 Surrender Value</span></td>  
<td style='border: 1px solid #fff; padding:1px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>₹ 27,351.56</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:110px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>Surrender Reason</span></td>  
<td style='border: 1px solid #fff; padding:1px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>Financial Emergency</span></td>  
</tr>  
<tr style=' background:#FFFFFF;'>  
<td style='border: 1px solid #fff; padding:1px; width:110px; height: 35px; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>Amount to be Transferred</span></td>  
<td style='border: 1px solid #fff; padding:1px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>--</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:110px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>Transaction ID</span></td>  
<td style='border: 1px solid #fff; padding:1px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>--</span></td>  
<td style='border: 1px solid #fff; padding:1px; width:110px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:normal;display:block; font-size:11px; padding:5px;'>Add Bank Account</span></td>  
<td style='border: 1px solid #fff; padding:1px; text-align:left; line-height:12px;'><span style='color:#000; font-weight:bold;display:block; font-size:11px; padding:5px;'>Yes</span></td>  
</tr>  
  
  
</table>" &  
 "</div>"

**\*\*\* TEMPORARY TABLE CREATION\*\*\***

Table

(

{ MenuName:"New Individual Timesheet", MenuIcon:'MicrosoftTeams-image (22)' },

{ MenuName:"New Teams Timesheet", MenuIcon:'MicrosoftTeams-image (23)' },

{ MenuName:"Approvals", MenuIcon:'MicrosoftTeams-image (24)' }

)

Note:

1. MenuName and MenuIcon both are colom name.
2. We do not use image format for value, like: .png , .jpg etc

**\*\*\* Manually creating combobox inside datacard of Form than concat combobox data to text field\*\*\***

1. Select combobox, select event “ITEMS” than paste below code in function:

**SortByColumns(Distinct(AddColumns('Company Short Codes',"**CompanyCodesnew**",'Company Short Code Data'),Name).Value,"Value",SortOrder.Ascending)**

Where **Company Short Codes** is table name , **CompanyCodesnew** is manually created colom , **Company Short Code Data**

Is unique identifier colom name and **Name** is colom name which data we want to show

1. Select the text field, select event **Default** and paste below code inside function:

**Concat(ComboBox2.SelectedItems,Value,",")**

Where **ComboBox2** is the name of ComboBox.

**\*\*\*To create ‘image uploading field’ and ‘Attchment’ field manualyy\*\*\***

For image uploading field:

Go to instert menu >> search ‘Add Picture’ field and select it.

For attachment field:

Go to insert menu >> Edit form >> select the table where the attachment field already created.>> select the attachment field >> copy and paste it out side form >> delete the form

**\*\*\*Patch() function to insert record\*\*\***

Patch('Table name',Defaults(' Table name '),

{

Colom1: Value1,

' Colom2': Value2,

' Colom3': Value3,

' Colom4': Value4

}

)

**Example**:

Patch(Registrations,Defaults(Registrations),{

    Name: *name*.Text,

    MobilenoNew: *mobile*.Text,

    SubjectNew: *subject*.Selected.Value,

    Profilepic: *UploadedImage1*.Image,

    Attachment:

    { FileName:First(*attachment*.Attachments).Name,

        Value:First(*attachment*.Attachments).Value

    },

    Gender: *gender*.Text,

    Description: *description*.Text,

    Color: *colorCombobox*.Selected.Value

})

Note:

1. **Patch()**  is predefined function, **Registrations**  is table name and **Selected** is predefined keyword.
2. Name, MobilenoNew, SubjectNew ,Profilepic, Attachment, Gender, Description and Color all are colum names of table.
3. name, mobile, gender, city and description all are name of the text field .
4. subject and city both are name of the dropdown field.
5. UploadedImage1 is the name of Add picture field.
6. Attachment is the name of attachment field.
7. Text, Selected, Attachments, Name, Value all are predefined keywords.
8. For attachment: { FileName:First(attachment.Attachments).Name, Value:First(attachment.Attachments).Value }

Note: where FileName, First(), Name, Value and Attachments are predefined keywords and attachment is name of the field.

**\*\*\*Patch() function to view details of perticular record\*\*\***

**Show Records: same as form with submit function.**

**View Details (set visibility mode view or edit):**

Create all fields with lable manually >> select the field >> select event “Default” >> put the below code with in function:

*glyrecords*.Selected.Name

Note:

1. Where **glyrecords** is name of gallery, **Selected**  is preferred keyword and **Name** is the name of colum inside table.
2. For dropdown field we need to fill all options in dropdown, to fill all options select the dropdown field, than select the

event **items** and put below code with in function:

Choices(<put choice name>)

1. For image, we need to select event **Image** and we need to take image to show image.
2. For PDF, we need to select event **Document** and we put below code with in function:

*glyrecords*.Selected.Attachment.Value

where **glyrecords** is the name of gallery, **Selected** is predefined keyword, **Attachment** is the name of colum inside table and **Value** is predefined keyword (we can use **FileName** instead of Value)

**\*\*\*Patch() function to update particular record\*\*\***

Patch(Registrations,

*glyrecords*.Selected,

{

    Name: *name2*.Text,

    MobilenoNew: *mobile2*.Text,

    SubjectNew: *subject2*.Selected.Value,

    Profilepic: *UploadedImage22*.Image,

    Attachment:

    { FileName:First(*attachment2*.Attachments).Name,

        Value:First(*attachment2*.Attachments).Value

    },

    Gender: *gender2*.Text,

    CityNew: *city2*.Selected.Value,

    Description: *description2*.Text,

    Color: *coloCombobox2*.Selected.Value

//'Color': colorchoice.Blue //if we want to put static value from userdefined choice

//'Color': 'coloCombobox2(Registrations)'.Active //if we want to put static value

from predefined choice

})

OR

Patch(Registrations,

LookUp(Registrations,Registration = *glyrecords*.Selected.Registration),

{

    Name: *name2*.Text,

    MobilenoNew: *mobile2*.Text,

    SubjectNew: *subject2*.Selected.Value,

    Profilepic: *UploadedImage22*.Image,

    Attachment:

    { FileName:First(*attachment2*.Attachments).Name,

        Value:First(*attachment2*.Attachments).Value

    },

    Gender: *gender2*.Text,

    CityNew: *city2*.Selected.Value,

    Description: *description2*.Text,

    Color: *coloCombobox2*.Selected.Value

//'Color': colorchoice.Blue // if we want to put static value

//'Color': 'coloCombobox2(Registrations)'.Active //if we want to put static value

from predefined choice

})

Note:

1. **Patch()**  is predefined function, **Registrations**  is table name, **glyrecords** is gallery name, **Selected** is predefined keyword.
2. Name,MobilenoNew,SubjectNew,Profilepic,Attachment,Gender,CityNew,Description and Color all are colum names of table.
3. name2, mobile2, gender2, city2 and description2 all are name of the text field .
4. subject2 and city2 both are name of the dropdown field.
5. UploadedImage22 is the name of Add picture field.
6. Attachment2 is the name of attachment field.
7. Text, Selected, Attachments, Name, Value all are predefined keywords.
8. **Colorchoice** is a user defined choice name and **Blue** is one of the value of this choice.
9. **coloCombobox2** is name of dropdown field and **Active** is one of the value of predefined choice.
10. 

When we create table than canvas app automatically create a unique(primary) field with same name of table.

i.e. **Registrations** is table name and **Registration** is the unique(primary) field name, according to this field we need to update the particular record.

**\*\*\* Createing Dropdown and combobox with Dataverse \*\*\***

Create a solution >> create table >> click in table name >> New colum >> Put display name >> select data type **CHOICE** >> CHOICE >> scroll down >>click on “New Choice” >> put display name >> put your options one by one in label by clicking on “New Choice” >> click on “Save”>>Search and select the field name which is recently putted by you on “**Sync this choice with**” >> click on “Save” >> refresh the table >> create a form >> attach table >> manage it by **edit field** option.

Note: (1) We can not create check boxes similar to html.

1. For combobox, select the field than enable **allow multiple selection** and **allow searching.**

**\*\*\* Createing lookup datatype colum with in dataverse \*\*\***

Suppose that we have a table and we want to show another table’s colom’s data in our current table, than that cases we use look up data type colum.

Open the current table in dataverse >> click on colum >> click on “+ new colum” >> put any name >> select datatype Lookup >> Lookup >> Related table >> select the table which data we want to show in the current table >> Open the current table >> appear the colum which was recently created >> double click on every colum and select the data (it will provide Only primary colum data in our current table)

**To show lookup colum data in gallary:**

Create a gallery >> select the gallery >> select event ‘Item’ >> put table name in function >> insert a lable in first row >> select the lable >> select the event ‘Text’ and put below code in function:

ThisItem.colorcodelookup.colorcontact

Where **colorcodelookup** is lookup colom of current table and **colorcontact** is another table’s colom’s name which data we want to show.we can use another colom name instead of colorcontact .

**\*\*\* ForAll() function \*\*\***

Create a gallery and show records , create a button , select this button , select event **Onselect** and put below code with in function:

ForAll(*Gallery2*.AllItems,

Patch(

        Registrations,ThisRecord,{

        Name:"Test"

    }

))

Where **Gallery2** is name of gallery, **Registrations** is name of table, **Name** is colom name of table, **Test** is static value and rest are predefined keywords.

Note: it will update Name’s colom value is Test only for all showing records inside gallery.

**\*\*\* Loader \*\*\***

<https://loading.io/>

email: videshi.muduli@infusai.com

pwd: Videshi@12345