

Mastering SharePoint list relationships in PowerApps

Overview:

The project involved developing a solution using PowerApps application with multiple SharePoint list that are related with each other. This project showcasing Power Apps working with List Relationships (Establishing a 1:N relationship between projects and tasks in SharePoint/Microsoft Lists.).

Technologies:

SharePoint, PowerApps (Canvas Apps),

Project Description:

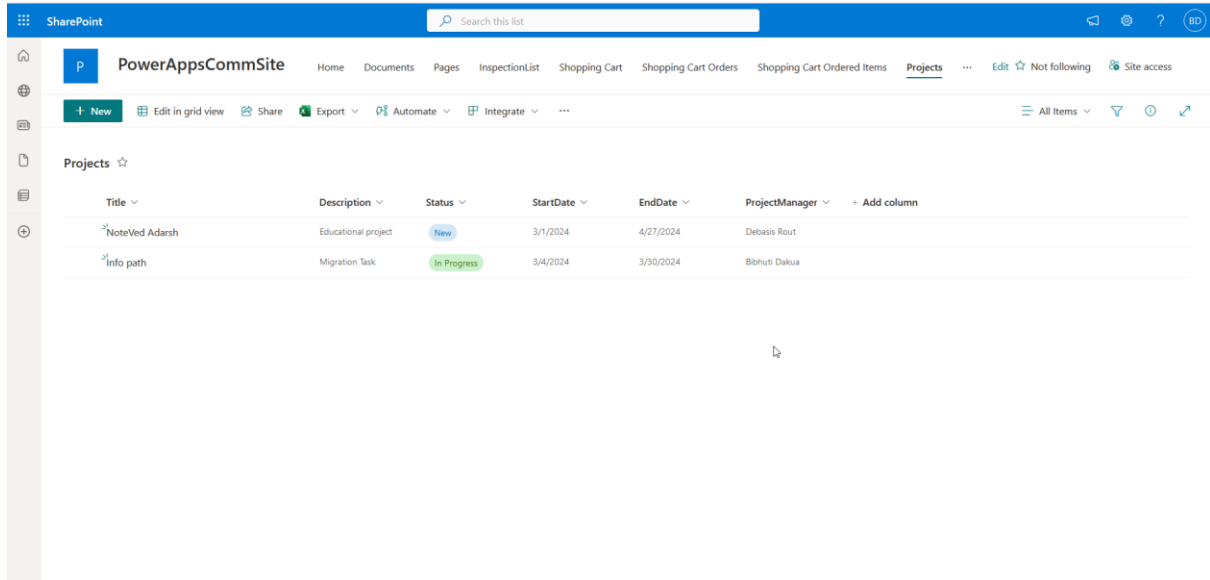
This PowerApp is connected to multiple SharePoint lists that are related to each other. The home screen showcases data from the project list. A user can select a project, look at the details of the project, edit the details, add new projects, delete projects, and perform full CRUD operations. For a specific project, the user can go to its related tasks, and we will get an editable grid-like experience that showcases the tasks that are related to that selected project. The user can make changes to the data; they can click save, and the updates are made to the data source. We can add new tasks for specific project saves; the entire PowerApps system will be fully responsive.

Sample and Design :

Steps:

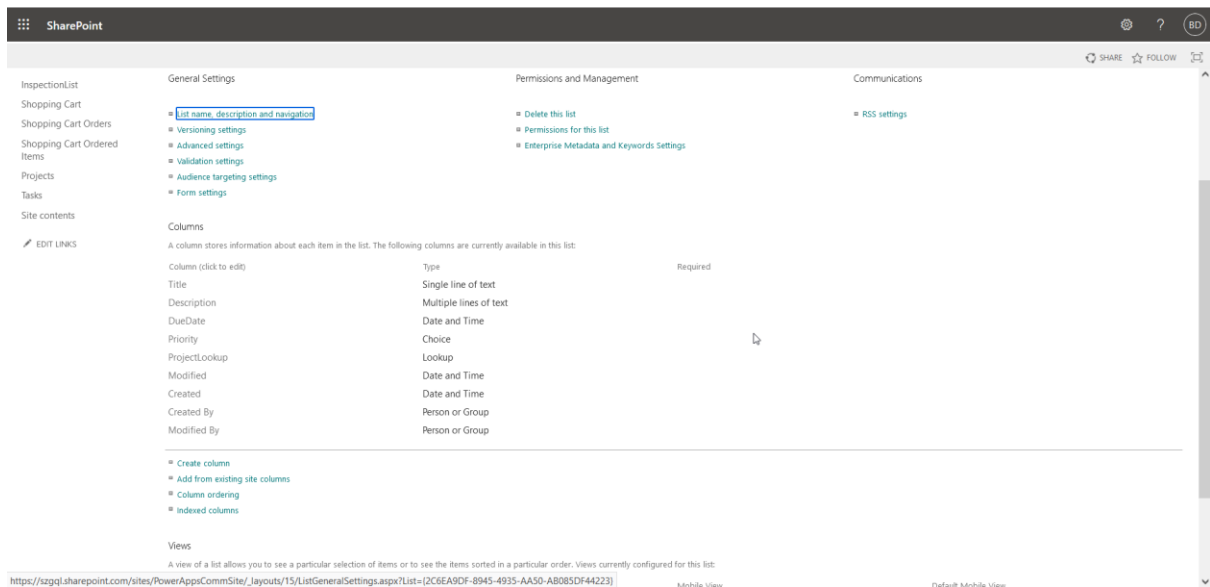
- First lets understand the schema of SharePoint list.
- Create a list, name it as projects (Main List).
- Add columns Title, Description, Status, StartDate, EndDate, Project Manager.
- Create another list, and name it as Tasks (Related List).
- Add columns Title, DueDate, Description, Priority.
- Create a relationship more like a connection is SharePoint between the two lists and to do that add a column of type lookup. This makes a column from one list become the choice of selection from another list (Column name, Project lookup, Type- lookup, select source list projects, select a column from the above list title, select more options, deletion from source list affect this list. And select deleting source items deleted value in this list.
- Create items in both the lists, In tasks list create item Project plan, provide description, priority and due date and in project lookup provide Project list's item title as lookup and save it.

Project List



Title	Description	Status	StartDate	EndDate	ProjectManager	+ Add column
NoteVed Adarsh	Educational project	New	3/11/2024	4/27/2024	Debasis Rout	
Info path	Migration task	In Progress	3/4/2024	3/30/2024	Bibhuti Dakua	

Task List



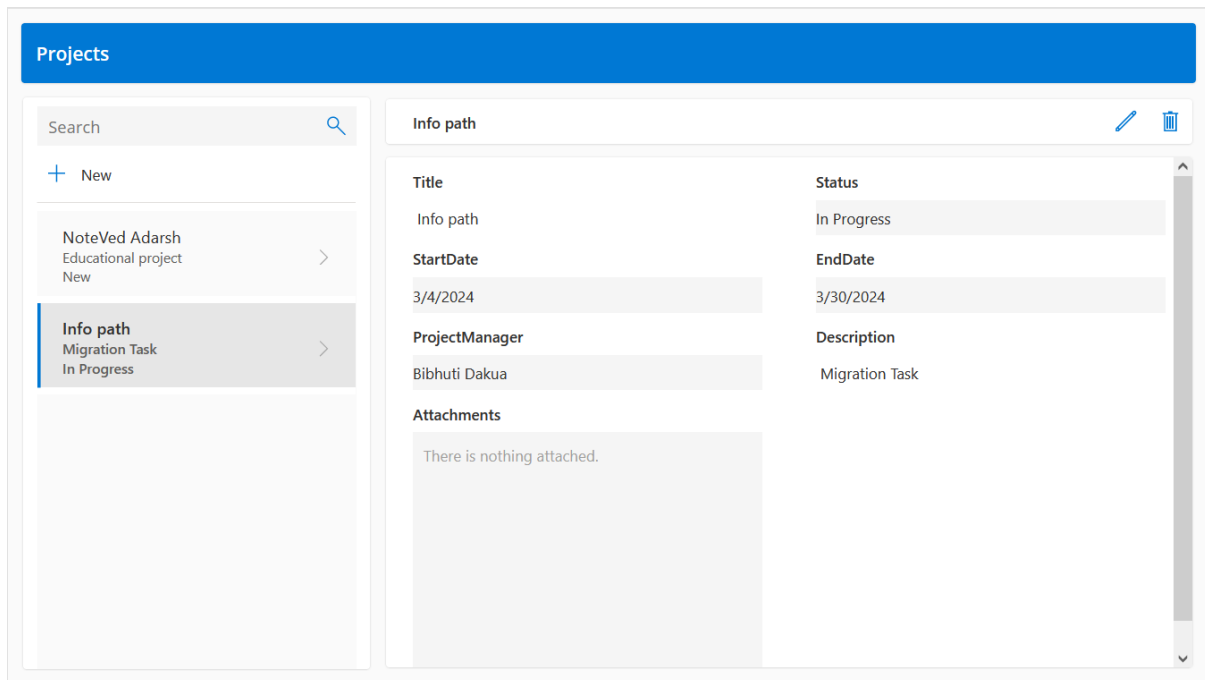
Column (click to edit)	Type	Required
Title	Single line of text	
Description	Multiple lines of text	
DueDate	Date and Time	
Priority	Choice	
ProjectLookup	Lookup	
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

To make Canvas App in SharePoint

- Go to make.powerapps.com
- Go to home -> start with a page design.
- Select gallery connected to external data.
- Choose a dataset to start from SharePoint site. Pick up SharePoint list, Click create

App.

- Now App is ready, The connected list is project list.
- You can add a new project by clicking on + button.



- To add task related to project and create a full grid like experience something similar to the Grid view experience in SharePoint itself allows for that first connect to task list, add data, SharePoint list and connect.
- To create a task screen, create a add screen, select layout as header and footer.
- Go to my newly created screen, go to header container, and add a header label and make the name as tasks.
- We need to create the grid in the main container of the screen.
- In new screen we have footer container, above the grid, If we like to give option to the user to go back, or create an item in the grid or save the grid.
- For that right click on footer container in tree view, move up, so it sets above the main container.
- Copy project screen header label and paste it on new screen header container.
- In text property provide "Tasks for "& RecordGallery1.Selected.Title . [RecordGallery1 is on the main screen]
- From project screen we need a option for the user to the task screen. We have a container named as selectedRecordheaderContainer1, go and insert a button control, add text for the button control as "Tasks".
- In OnSelect property of the button provide Navigate(Screen1).
- For the middle container, here we would like to give an option for the user to go back to the previous screen. Just select the container and add text as "Back".
- In OnSelect property of back button provide back(). [The back function in powerapps returns to the screen that was most recently displayed].
- To load all the tasks that are related to specific project from tasks screen and show them in a gallery, In OnVisible of task screen we need to create a collection

ClearCollect(colGridData, Filter(Tasks, 'Project lookup'.Value=RecordsGallery1.selected.Title))

- To load all the tasks that are related to specific project from tasks screen and show them in a gallery. In OnVisible of tasks screen we need to create a collection for that we need to create a collection.
ClearCollect(colGridData, Filter(Tasks, project looks.Value=RecordsGallery1.Selected.Title)
- Now to show this tasks in main container, we will insert a gallery by inserting vertical gallery and rename this gallery as Tasks gallery.
- The Items property for this gallery will be colGridData and we are creating editable grid like experience. In the gallery we will add controls specific to column type of tasks list.
- For description add inputText and in value property add ThisItem.description.
- For due date add a date picker in value property add ThisItem.DueDate.
- For priority add a choice column in value property add choices(Task.Priority), and add fields, for defaultselectedItems provide ThisItem.Priority.
- Add a button to middle container and name it as Save.
- Now when ever a change is made to any of the controls. We would like to store the updated data in another collection, For that in the tasks gallery we will add a button, name the button as btnUpdate, sets its visibility to false.
- Now go ahead and select all the controls which user can change on all of these, for that in Onchange property provide Select(btnUpdates).
- In that update button in OnSelect property provide If(IsBlank(Lookup(colGridUpdates, ID=ThisItem.ID)) collect(colGridUpdates, ThisItem)

[If the item is not exist in the collection only then add the item to the collection. After this update the data in new collection colGridUpdates, where ID is ThisItem.ID and here we will define columns that we are updating. For that UpdateIf(colGridUpdates, ID=ThisItem.ID, {Title:txtTitle.Value, Description: txtDescription.Value, Date: dpDueDate.Value, Priority: {Value: drpPriority.selected.value}})

```
If(
  IsBlank(
    Lookup(
      colGridUpdates,
      ID = ThisItem.ID
    )
  ),
  Collect(
    colGridUpdates,
    ThisItem
  )
);
UpdateIf(
  colGridUpdates,
  ID = ThisItem.ID,
  {
    Title: InputTitle.Value,
    Description: txtDescription.Value,
    DueDate: DatePickerCanvas4.SelectedDate,
    Priority: {Value: DropdownCanvas1.Selected.Value}
  }
)
```

- When the user clicks save button, In OnSelect we will Patch to Task List for that in OnSelect= Patch(Tasks, ShowColumns(colGridUpdates, "Title", "Description", "DueDate", "Project lookup", "ID"); Clear(colGridUpdates)
- To give user an option to create an item in the middle container we will add a button and name it as "Add", In OnSelect Patch(Tasks, Default(Tasks),{Title: "New Task", Priority: {Value: "medium"}, Project lookup: {Value:RecordsGallery1.Selected.Title, Id: RecordsGallery1.selected.ID}})

```
Patch(
    Tasks,
    Defaults(Tasks),
    {
        Title: "New Task",
        Priority: {Value: "Medium"},
        ProjectLookup: {
            Value: RecordsGallery1.Selected.Title,
            Id: RecordsGallery1.Selected.ID
        }
    }
)
```

- Add a button to the middle container and name it as btnLoadGrid and in select property provide ClearCollect(colGridData, Filter(Tasks, 'ProjectLookup'.Value = RecordsGallery1.Selected.Title)); Clear(colGridUpdates)
- In TaskScreen OnVisible change to Select(btnLoadGrid).
- In button Add's OnSelect property after Patch() call the same function Select(btnLoadGrid).
- When Saving after Save, we will call that button for this in save button's OnSelect property after Patch provide Select(btnLoadGrid).
- To remove white space between items in the gallery, makeTemplateSize property value to 40.
- The save button should only be enabled if user has made any updates to the Grid, So for the DisplayMode of the button If(CountRows(colGridUpdates)>0, DisplayMode.Edit, DisplayMode.Disabled)
- For DisplayMode property of Add button if(CountRows(colGridUpdates)=0, DisplayMode.Edit, DisplayMode.Disabled).
- To add headers we need to add a container and add text controls for controls and make adjustments, make width of the container as Gallery2.Width

Tasks for NoteVed Adarsh

[Back](#) [Save](#) [Add](#)

Title	Description	Due Date		Priority	
Project Plan	Define Project plan	3/29/2024		Medium	Delete
New Task	This Is a new Task	3/15/2024		Medium	Delete

This project was successfully completed by using the above-mentioned technology, and it is maintaining the required requirements.