

Shopping Cart App

Overview:

The project involved developing a solution using PowerApps application with SharePoint list for developing Shopping cart app. This project enabled Shopping items and adding them to cart.

Technologies:

SharePoint, PowerApps (Canvas Apps),

Project Description:

The PowerApps Shopping Cart project is an application built using PowerApps, a low-code platform by Microsoft. The project aims to create a shopping cart experience for users to select and purchase items from a catalog. The app utilizes a SharePoint list as a data source to store product information, and a collection to manage the shopping cart.

Components:

PowerApps: The development platform used to build the application. PowerApps offers a user-friendly interface with drag-and-drop functionality, making it accessible even to those with limited coding experience.

SharePoint List: Acts as the product database. It stores information about each item, such as product name, description, price, image link, and unique identifier.

Collection: A temporary storage within the app. It functions as the virtual shopping cart, holding information about the products a user chooses to purchase. This includes the chosen product ID, quantity desired, and potentially the calculated subtotal for that item.

Sample and Design :

Create datasource

Steps

- Create a SharePoint list and name it as Shopping Cart.
- Create fields Product Description(multilinetext field), Price(Currency field), Color(choice field), Size(singlelinetext field), ImageLink(Single line text field).
- Add items to the list.

[EDIT LINKS](#)

Shopping Cart ▸ Settings

List Information

Name: Shopping Cart
Web Address: <https://szgql.sharepoint.com/sites/PowerAppsCommSite/Lists/Shopping Cart/AllItems.aspx>
Description:

General Settings

- [List name, description and navigation](#)
- [Versioning settings](#)
- [Advanced settings](#)
- [Validation settings](#)
- [Audience targeting settings](#)
- [Form settings](#)

Permissions and Management

- [Delete this list](#)
- [Permissions for this list](#)
- [Enterprise Metadata and Keywords Settings](#)

Communications

- [RSS settings](#)

Columns

A column stores information about each item in the list. The following columns are currently available in this list:

Column (click to edit)	Type	Required
Title	Single line of text	
Product Description	Multiple lines of text	
Price	Currency	
Color	Choice	
Size	Single line of text	
ImageLink	Single line of text	
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

- [Create column](#)
- [Add from existing site columns](#)

- Create another SharePoint list with name Shopping Cart Ordered Items with fields such as Product price(currency field), Product Id(Number), QTY(number), ItemSize(single line text), subtotal (Number), Ordered ID(number).

[EDIT LINKS](#)

Shopping Cart Ordered Items ▸ Settings

List Information

Name: Shopping Cart Ordered Items
Web Address: <https://szgql.sharepoint.com/sites/PowerAppsCommSite/Lists/Shopping Cart Ordered Items/AllItems.aspx>
Description:

General Settings

- [List name, description and navigation](#)
- [Versioning settings](#)
- [Advanced settings](#)
- [Validation settings](#)
- [Audience targeting settings](#)
- [Form settings](#)

Permissions and Management

- [Delete this list](#)
- [Permissions for this list](#)
- [Enterprise Metadata and Keywords Settings](#)

Communications

- [RSS settings](#)

Columns

A column stores information about each item in the list. The following columns are currently available in this list:

Column (click to edit)	Type	Required
Title	Single line of text	
Product Price	Currency	
Product ID	Number	
QTY	Number	
ItemSize	Single line of text	
SubTotal	Number	
Ordered ID	Number	
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

- [Create column](#)
- [Add from existing site columns](#)

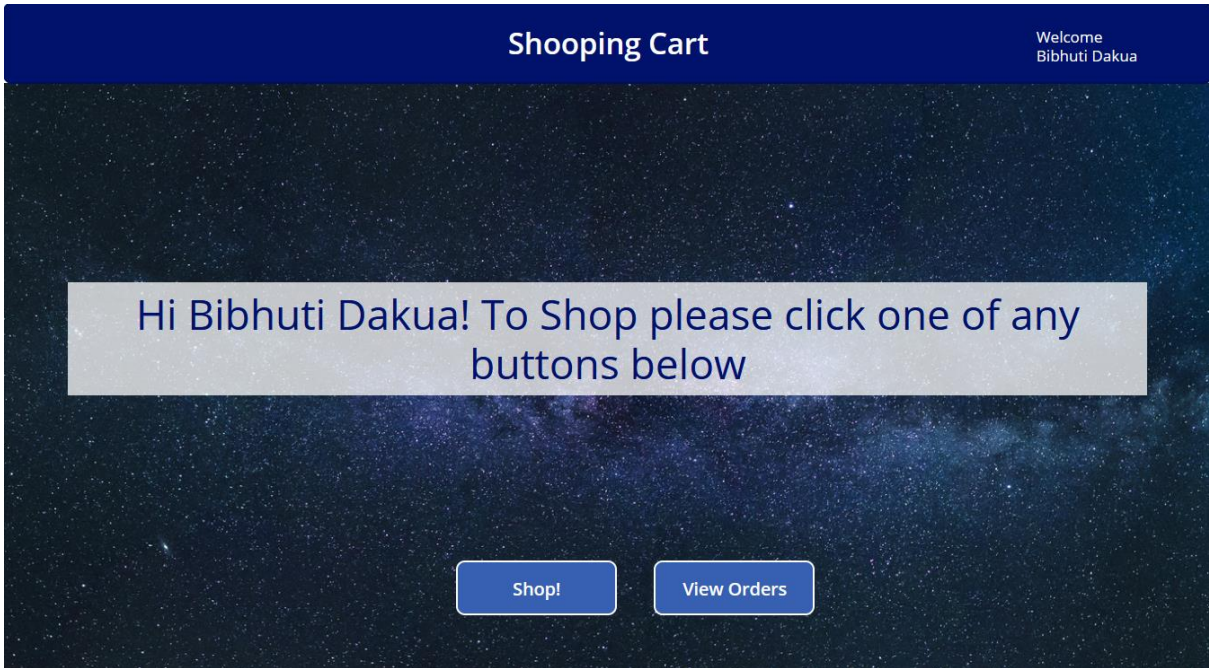
- Create a SharePoint list and name it as Shopping Cart orders. And add fields as OrderTotal(number field), OrderComments(multilinetext).

The screenshot shows the SharePoint 'Shopping Cart Orders' list settings page. The page is titled 'Shopping Cart Orders' and includes a navigation pane on the left with options like Home, Documents, Pages, Recent, InspectionList, Shopping Cart, Shopping Cart Orders, Shopping Cart Ordered Items, and Site contents. The main content area is divided into several sections:

- List Information:** Name: Shopping Cart Orders, Web Address: https://szgqj.sharepoint.com/sites/PowerAppsCommSite/Lists/Shopping Cart Orders/AllItems.aspx, Description: (empty).
- General Settings:** Includes options for List name, description and navigation; Versioning settings; Advanced settings; Validation settings; Audience targeting settings; and Form settings.
- Permissions and Management:** Includes options for Delete this list, Permissions for this list, and Enterprise Metadata and Keywords Settings.
- Columns:** A table showing the current columns in the list. The columns are: Title (Single line of text), Order Total (Number), Order Comments (Multiple lines of text), Modified (Date and Time), Created (Date and Time), Created By (Person or Group), and Modified By (Person or Group). Below the table are options to Create column, Add from existing site columns, Column ordering, and Indexed columns.

PowerApps Shopping Cart Home screen

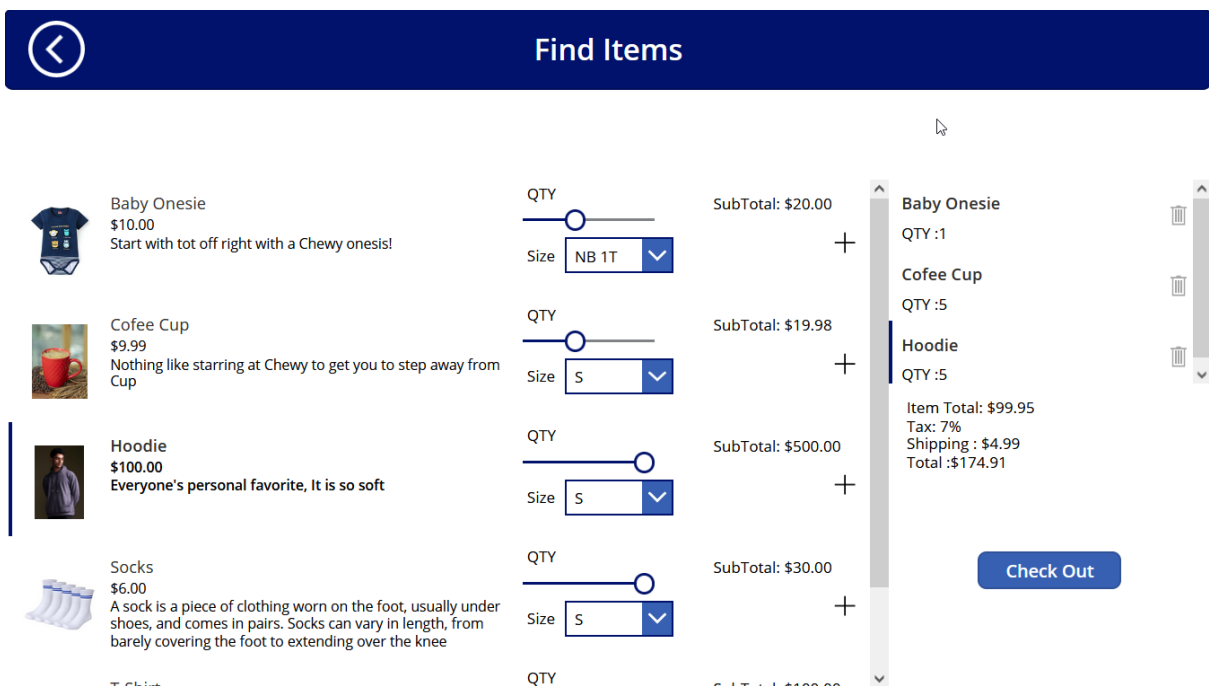
- Go to make.powerapps.
- Create blank app
- Create header by adding button and naming it with Shopping Cart.
- For welcome message add a label and in text property give "&User().FullName.
- Add a label provide fill property with RGBA(255, 255, 255, 0.75) and provide Text
- Add button and name it as Shop! and in OnSelect property give Navigate('FindItem').
- Add another button name it as View Orders and in OnSelect property give Navigate('Check out')



Use Case- Power App shopping cart app home screen is a welcome screen, when a user login to PowerApps he will get this screen, this screen will show welcome message, from this screen the user can navigate to Shopping item screen and also navigate to view order screen.

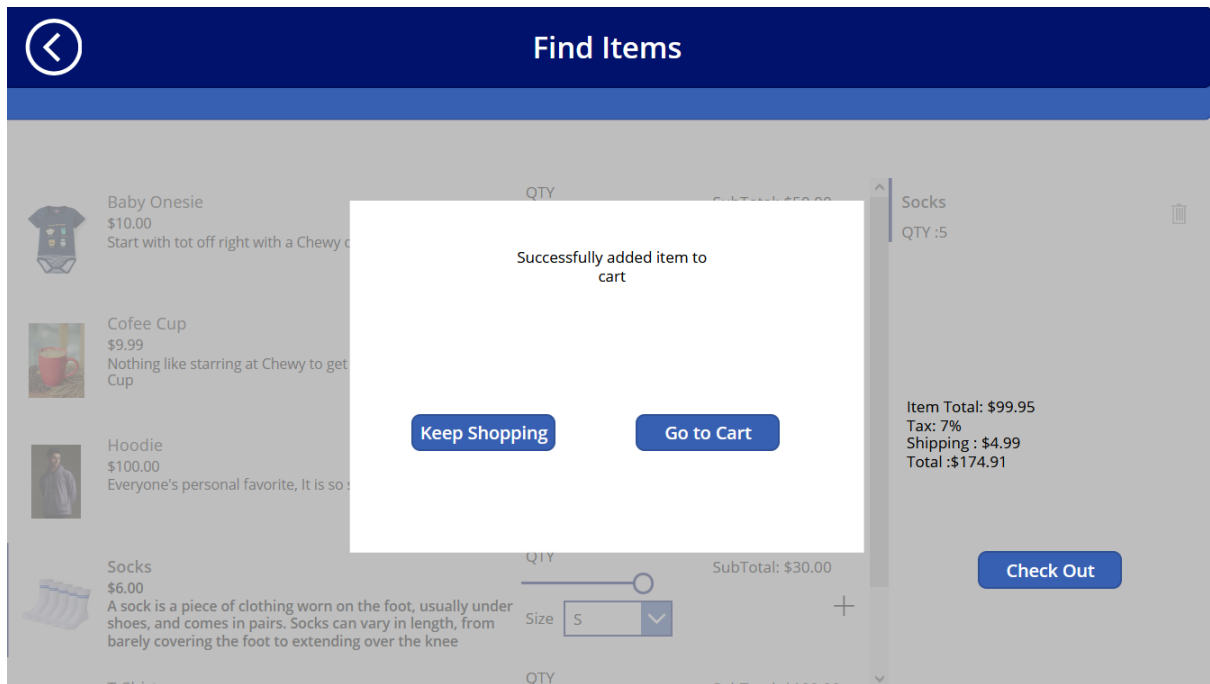
Find Items screen:

Use case- This screen shows items with cost, quantity, size, description of item subtotal cost based on selected number of items. Also show added items on cart.

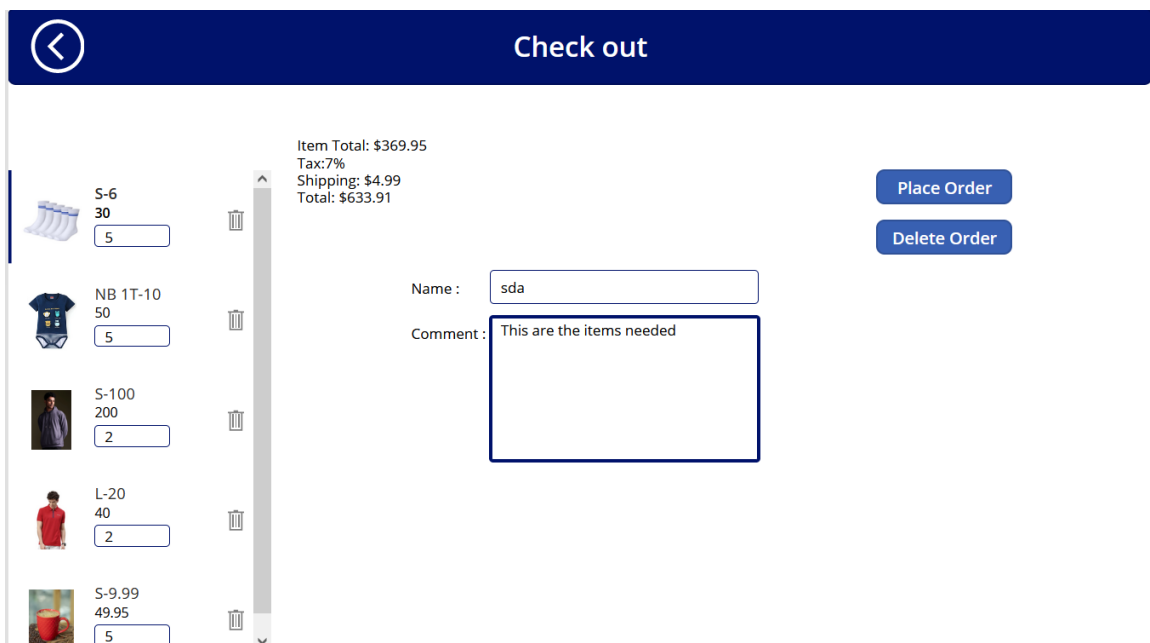


Steps

- Create new screen
- Add Header and name it with Find Items.
- Add a gallery with three created list.
- Change gallery layout to Image, Title, subtitle and body.
- In image property of gallery add ThisItem.ImageLink.
- Add a label in the gallery and name it as QTY.
- Add a slider and in Max property of the slider provide 5.
- Add a text label and in text property provide size.
- Add a dropdown to the gallery and in items property provide Split(ThisItem.Size,",").
- Add a label to the gallery and in text property provide "subtitle: " & Text(ThisItem.Price*Slider.Value,"\$#,##0.00".
- Change ">" icon to "+" icon and in DisplayMode provide If(Slider1.Value > 0,DisplayMode.Edit,DisplayMode.Disabled).
- In OnSelect property of +icon, provide Collect(colShoppingCart, {Title :ThisItem.Title, ProductPrice:ThisItem.Price,ProductID : ThisItem.ID, QTY :Slider1.Value, ItemSize :Dropdown1.selected.Value, SubTotal (Slider1.Value*ThisItem.Price)});Reset(Slider1);Reser(Dropdown1); Set(varShowScreen,true).
- Add another vertical gallery to the screen and add data source as colShoppingCart.
- Add layout of the gallery as Title.
- Add a label to the gallery and in text property provide "QTY : " & ThisItem.QTY.
- Change ">" icon to trans icon and in OnSelect property of trans provide Remove(colShoppingCart,ThisItem).
- Add a label and in text property provide With({ST:Sum(colShoopingCart,SubTotal)},"Item Total: " & Text(ST,"[\$-en-US]\$#,##0.00") & " Tax:7% Shipping: \$4.99 Total: " & Text(ST+ST*0.7+4.99,"[\$-en-US]\$#,##0.00").
- Add a button to screen name it as Check out. In DisplayMode of this button provide If(IsEmpty(colShoopingCart), DisplayMode.Disabled,DisplayMode.Edit).
- In OnSelect property of the button provide Navigate('Check out').



- For Success prompt message Add a rectangle and made Fill property as RGBA(169,169,169,0.75).
- Add another rectangle over here and make the fill property as Color.White.
- Add two buttons over the rectangle and name it as Keep Shopping and Add to Cart.
- In OnSelect property of Keep Shopping button provide Set(varShowSuccess, false).
- Add a label on the rectangle and in text property provide “Successfully added items to cart”.
- Group all these rectangles and buttons and in visible property provide varShowSuccess.
- In Go to Cart button OnSelect property provide Navigate(Checkout).



- In checkout screen add a vertical gallery and in datasource property add colShoppingCart collection.
- In gallery image property provide lookup('Shopping Cart Items', ID=ThisItem.ProductID).ImageLink.
- Add a text input to the gallery and in default property provide ThisItem.QTY.
- Make subtitle of the gallery to ThisItem.Subtotal.
- Make title as ThisItem.ItemSize & "-" & ThisItem.ProductPrice.
- In Input text's onChange property provide patch(colShoppingCart,ThisItem, {QTY: Value(self.text), subtotal: Value(self.text)*ThisItem.ProductPrice}).
- Add a text label to the checkout screen & make the text property as With({ST: Sum(colShoppingCart,subtotal)}, "Item Total: " & Text(ST, "[\${-en-US}]\$#,##0.00")& Tax:7% Shipping:\$4.99 Total :Text(ST+ST*0.7+4.99,"[\${-en-US}]\$#,##0.00").
- Add two label for name and comments.
- Add two input text fields, one for name and one for comment and make comments input text to multiline text.
- Add a button and name it as place order, In OnSelect property provide Set(varShowSpinner,true); Set(varRecordId, Patch('Shopping cart Orders', Default('Shopping Cart Orders'),{Order Total': With({ST:Sum(cplShoppingCart,Subtotal)}),ST+ST*0.7+4.99),'Order comments': Input Comments.Text,Title:InputName.Text}).ID); ClearCollect(colOrderedItems, AddColumns(colShoppingCart,"OrderID",varRecordID)); Collect('Shopping Cart Ordered Items',colorderedItems);Navigate(ShoppingCartScreen);Set(varShowSpinner,false);

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