

Sales Invoice

Overview:

The project involved developing a solution using PowerApps application with SharePoint list and power automate for developing sales invoice app. This project enabled generating sales invoice and send email with attached invoice pdf.

Technologies:

SharePoint, PowerApps (Canvas Apps),Microsoft Power Automate Flow

Project Description:

The PowerApps Sales invoice project is an application built using PowerApps, a low-code platform by Microsoft. The project aims to leverage auto increment invoice number, usage of patch, updateIf(), left, right, concatenate functions in powerapps.

We can increment the invoice number by clicking on generate invoice number button. When we submit invoice item with details the data is stored in SharePoint list as list item, when an item is created in SharePoint list the automated flow will trigger and an email will be send to customer with invoice as pdf.

Sample and Design :

Steps:

- Go to make.powerapps.
- Create an blank app, make layout as mobile, and provide name to app.
- Add label for Invoice Number and add a button for generate invoice number.
- Add input text box for item name, cost and quantity.
- Add a label for amount and in text property add item1cost.text*QTY.text.
- Also add another label for secund items amount and in text property add item2cost.text*QTY2.text.
- Add a label to BillTo where you can provide for whome you are billing.
- Add a label for totalcost and in text property Item1Total.Text+Item2Total.Text.
- Add a button and name it as generate invoice.
- Add datasource to PowerApps.
- Once we click on generate invoice number, then Invoice Number should populate so OnSelect property of Generate Invoice Number provide `Set(InvoiceNum,First(Filter(TrackInvoiceNumber, ID=1)).Title);UpdateIf(TrackInvoiceNumber,ID=1, {Title: Concatenate("INV-", Text(Right(InvoiceNumber,4)+1))})`.

- So everytime if we click on Generate Invoice Number button a new invoice number will be generate.
- In Generate Invoice button we need to apply Patch function, for that in OnSelect property provide Patch(InvoiceList,Defaults(InvoiceList),{Title:'Invoice number'.Text,BillTo:BillToIP.Text,Item1:ItemName1.Text,Item1Cost:Cost1.Text,Item1Quantity:QTY1.Text,Item2:ItemName1_1.Text,Item2Cost:Cost1_1.Text,Item2Quantity:QTY1_1.Text,TotalCost:TotalCal.Text})

Sales Invoice

Invoice Number

Item Name	Cost	Quantity		
<input style="width: 100%;" type="text" value="Mobile cover"/>	<input style="width: 100%;" type="text" value="500"/>	<input style="width: 100%;" type="text" value="2"/>	1000	
<input style="width: 100%;" type="text" value="Charger"/>	<input style="width: 100%;" type="text" value="700"/>	<input style="width: 100%;" type="text" value="1"/>	700	
Total Cost			1700	

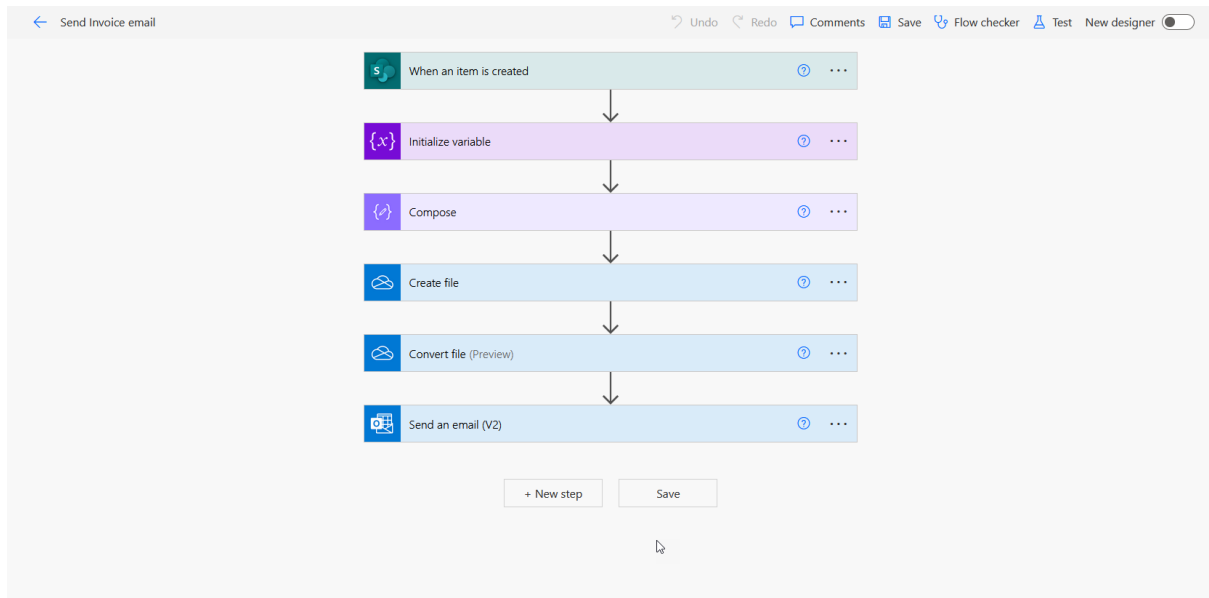
Billing To

- The invoice will be save on a SharePoint list.

The screenshot shows a SharePoint list titled 'InvoiceList' with the following data:

Title	BillTo	Item1	Item2	Item1Cost	Item2Cost	Item1Quantity	Item2Quantity	TotalCost
INV-1001	Surya Telecom	OnePlusNordCE2Lite	OnePlusNordCE2	19200	24000	2	2	86400
INV-1002	Surya Telecom	Realme9i	OnePlusNordCE2	28000	24000	2	2	104000
INV-1003	Surya Telecom	Realme9i	OnePlusNordCE2	28000	24000	2	2	104000
INV-1004	ABC battle shop	Coffee cups	Jar	40	50	1	5	290

To create flow for send invoice PDF email of generated invoice



- Go to make.powerautomate.com
- Create automated flow from blank and name it as send invoice email.
- Add a trigger when an item is created. Add site address as SharePoint site and in list name provide created invoice list.
- Add a new step as initial variable in name property provide ObjectList, type-object, value- body.
- Add another step Compose and in input provide html code as expression.

```
concat('<!DOCTYPE html>
      <html>
      <head>
      <style>
      table, th,td {
      Border:1px solid black;
      border-collapse:collapse;
      }
      th,td {
      padding:15px;
      text-align:left;
      }
      table#t01 {
      width:100%;
```

```

background-color: #f1f1c1;
}
</style>
</head>
<body>
<h2>Performa Invoice #: ', variables('objectList')['Title'],'</h2>
<table style="width:100%">
<tr><td>BillTo</td><td>', variables('objectList')['BillTo'],'</td></tr>
<tr><td>Item 1</td><td>',
variables('objectList')['Item1'],'</td></tr>
<tr><td>Item 2</td><td>',
variables('objectList')['Item2'],'</td></tr>
<tr><td>Total Cost</td><td>',
variables('objectList')['TotalCost'],'</td></tr>
</body></html>')

```

- Add a new step create onedrive file add folder path, file name as SalesInvoice.html, file content- output of compose.
- Add another step convert file(preview), In file provide /salesinvoice.html, target type PDF.
- Add a step send an email (v2), In to add email address, in subject- Invoice ready- Title from SharePoint list item.
- Attachment name- file name from convert file in body add body from file content.

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