Two approaches: App-controls with Desktop-Printing -OR- Digital-Template with Automate

# A screenshot of a computer Description automatically generated with low confidencePowerApps - Print Screen approach

***\*\* This approach may NOT be suitable for multiple-page printout forms – unless printable screens are chained and programmatically invoked – but user experience would be such that each page/screen would have to be printed individually – hence, suitable for 1-page scenarios.***

Consider legacy web pages where CSS for Print changes the Layouts/UI/DOM elements on the same page to format for print – while allowing normal or even complex interfaces while Editing Forms/UI/DOM elements.

1. From Canvas App Design time create a New Screen of type “Portrait print” or “Landscape print”
2. Either design the screen/page within the boundaries or place the scanned A4 or Landscape page from real-world as **IMAGE** and overlay UI/Controls such that desired input requirements can be achieved – and Low-code approaches to manage complex input and presentation can be used.
3. On the Print button press – handle/program by setting context/global variables as needed to further trigger screen-embedded controls to suit the print-out before **Print() function** is executed. [\*\**Alternatively, invoke Power Automate to possibly delegate printing to other formats like HTML or Word-Business-Online or SharePoint list item with office document properties and remove Print() function that uses Windows Desktop printing capabilities..*](#_Word_Office_embedded)
4. Consider programming a Reset button if user wants to Cancel the print and get back to editing.

# Word Office embedded properties

Using MS-Word Developer Tab and Document Properties – This methodology requires having pre-existing or-then preparing-new Word/Office/Pdf templates.

* With SharePoint list columns and embedded document properties in MS-Word etc.
* With Word-Online-Business premium connector (e.g. – where variable list of tabular records is to be listed from data source and where for example text/html-composite single string approach may not be suitable for presentation in the office/word document.
* Option A: If using only Model-driven App (DataVerse)
  + Needs, a Business Process flow Or Custom action menu button etc..
    - Ideally for many file/prints scenarios – Use Custom Activity on Related Parent record and build BPF on the custom activity – In such a scenario generated file/printout can be saved back to a custom File column or SharePoint integration folder/library associated with such custom activity-entity type.
  + Then, call Power Automate from a/any Stage as Button and in some cases allow user to provide/select inputs directly as flow instantiation parameters. Ensure flow is designed to update the BPF and Model-driven App (check documentation/search-Google-YouTube on how to)
* Option B: Using Canvas App
  + Provide UI elements and pass needed parameters or record reference to lookup data source from Power Automate to then Generate such document with SharePoint connector actions etc.

# Third-party connectors

1. Adobe Document Generator – JSON friendly
2. Others - Pdf or proprietary document-format forms based third-party connectors.