

Abhishek Pandit

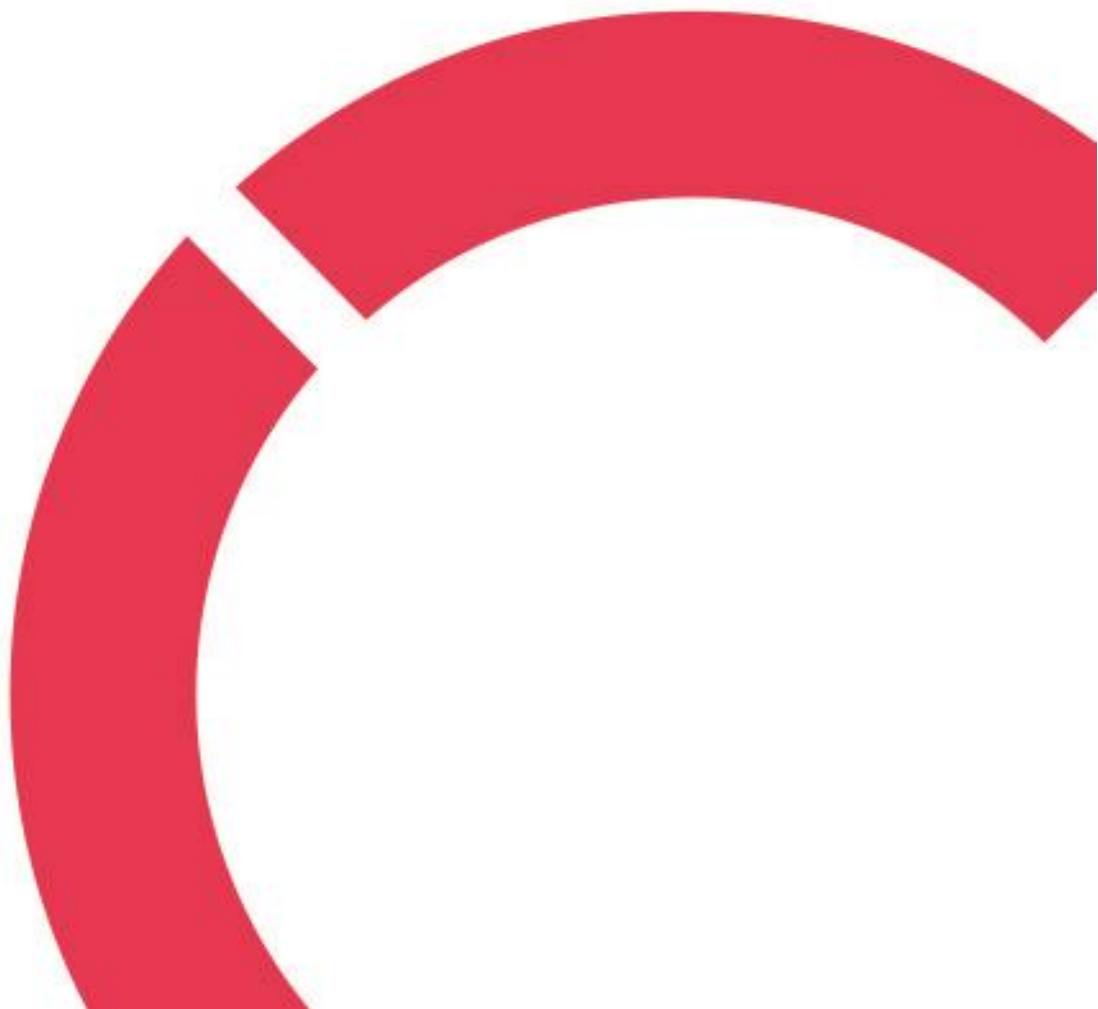
**EXPLORING LOW-CODE AND NO-CODE DEVELOPMENT WITH
POWERAPPS**

Thesis

CENTRIA UNIVERSITY OF APPLIED SCIENCES

Bachelor of Engineering, Information Technology

May 2024



ABSTRACT

Centria University of Applied Sciences	Date May 2024	Author Abhishek Pandit
Degree programme Bachelor of Engineering, Information Technology		
Name of thesis EXPLORING LOW-CODE AND NO-CODE DEVELOPMENT WITH POWERAPPS		
Centria supervisor Henry Paananen		Pages 30+5
<p>Traditional software development often demands extensive coding expertise, limiting its accessibility to a select group of individuals. Low-code and no-code development platforms (LCDPs) have emerged as game-changers, empowering anyone to create applications without requiring in-depth coding knowledge. PowerApps, a leading LCDP from Microsoft, stands out for its user-friendly drag-and-drop interface, pre-built connectors and templates, and seamless integration with various data sources. This thesis deeply explores the transformative power of PowerApps, exploring its versatility and applicability across a wide range of business scenarios. To conclude, the key takeaways from the exploration of PowerApps, emphasizing its ability to rapidly develop applications that enhance productivity, streamline operations, and drive innovation. LCDPs like PowerApps Empower businesses of all sizes to embrace digital transformation without being constrained by coding limitations.</p>		
Key words Business application, CRUD operation, Low-code and No-code, Microsoft, PowerApps		

CONCEPT DEFINITIONS

LCDPs

Low-code and No-code Development Platforms

RAD

Rapid Application Development

XP

Extreme Programming

ERP

Enterprise Resource Planning

CRUD

Create, Read, Update, Delete

SQL

Structured Query Language

ABSTRACT
CONCEPT DEFINITIONS
CONTENTS

1 INTRODUCTION.....	1
2 LOW-CODE AND NO-CODE DEVELOPMENT	3
2.1 Evolution of Development Approaches	4
2.2 Essentials of Low-code and No-code Development	5
3 POWERAPPS FUNDAMENTALS.....	6
3.1 Key Features	7
3.2 Application Types.....	8
3.2.1 Canvas Apps	9
3.2.2 Model-Driven Apps.....	10
3.2.3 Web Portals Apps	11
3.3 Merits and Demerits of Using PowerApps.....	12
4 LOW-CODE DEVELOPMENT WITH POWERAPPS	13
4.1 Requirement Analysis	13
4.2 Development Process	16
4.2.1 Creation of the Application.....	16
4.2.2 Connecting application to the Data Source	18
4.2.3 Developing the Application	21
4.3 Results	27
5 CONCLUSION	28
6 REFERENCES.....	29

FIGURES

Figure 1 Blank canvas app layout.	9
Figure 2 A dashboard in Model driven layout with different sections (Microsoft).	10
Figure 3 The landing page of the application that stores First Name, Last Name and Email as an input (Desktop version).	14
Figure 4 Two different responsive versions of the same app for Mobile and Tablet.	15
Figure 5 New application options to create.	16
Figure 6 Options for different types of applications.	17
Figure 7 Excel table where the data is stored.....	18
Figure 8 Selecting OneDrive for Business for data source	19
Figure 9 selecting the table from the excel file to store the data from PowerApps.	20
Figure 10 changing display setting to work with responsive designs.	21
Figure 11 The CONTINUE button is deactivated unless all the fields are correctly given.....	22
Figure 12 The CONTINUE button activates when all fields are in correct format.	22
Figure 13 Use of If statement to ensure the UI detects the screen size and adjust itself.	23
Figure 14 Code to check if all input fields have valid inputs.	23
Figure 15 Collection function with input taken from the first page to collect and store data in excel ...	24

Figure 16 Third page of the application ListPage.....25
Figure 17 Patch function with its three arguments to save the modified data26
Figure 18 Data is successfully stored in the Excel.....27

1 INTRODUCTION

Businesses are constantly looking for innovations and innovative solutions to grow their businesses effectively. Classical software development is often linked with complicated coding skills and heavily dependent on specialized expertise, which has been taken as a major obstacle for many businesses. To minimize this issue, low-code, and no-code development platforms (LCPDs) have been an effective solution, enabling developers or anyone with little to zero coding knowledge to design complex software in less time.

Low-code and no-code applications give people and businesses the ability to develop services and solutions for their business without the need for programming skills. The simplicity of use of these application services is a top priority nowadays. The first stage in developing an application is to build the services using questionnaires (Name of the app, different devices) in order to narrow down the services that the application may offer. The next step in the process is development using drag and drop choices for pages, data displays, and text buttons. The final steps in the construction process include prototype testing, connecting the application to a data source, and determining if the program can do the desired tasks. LCPDs are not limited to those without any prior coding knowledge. It all boils down to the purpose and the application of the intended application. Professional software developers might reduce their workload and produce prototypes more quickly with the help of such LCPDs.

One of the best LCPDs is PowerApps because it is a collection of applications, services, and connectors, as well as data platform, that enables quick development of custom designs as according to the company's needs and requirements. Also, with PowerApps, data can be stored either in the provided data platforms by Microsoft itself or in any other online or local data sources. Users can easily construct different app components, connect to various data sources, and personalize user interface without writing a single line of code. This strategy encourages a larger range of people, including non-experienced individuals, to engage in the application development process, boosting the cooperation and creativity throughout the business. PowerApps provides a large collection of pre-built templates that respond to several business demands while also making the app building process easy, simple, and effective. These templates provide as a strong base for developing customized apps, saving the time and workload required to carve a fully functional app. Users may simply adapt these templates to match their personal demands, resulting in a unique solution that meets their specific organizational goals. PowerApps has transformed the way corporations approach application creation by simplifying

app development and empowering non-technical individuals. This low-code platform has made accessible app development, removing obstacles and allowing businesses to swiftly innovate and adapt to changing business environments. PowerApps is prepared to shape the future of digital transformation by enabling organizations to achieve increased agility, efficiency, and creativity. The application designed by using PowerApps can be shared by a person or groups whoever needs it, without following traditional approach for the application development process with all the stages. This thesis unboxes the properties of PowerApps, showing its flexibility and applicability with the basic implementation.

The objective of the thesis was divided into two separate goals, first to understanding the core concept of LCDPs like PowerApps for building a business application which is responsive to all the devices and perform simple CRUD operation to the data and second, to implement all the knowledge and concept by building a simple prototype. To gain these two primaries, the thesis was divided into 5 different sections, beginning from the introduction. The first “Introduction” part addresses about the low-code and no-code development and the LCDPs. The second part “Low-code and no-code development” covers the historical background and the evolution of low code approach of application creation. The third section “PowerApps fundamentals” covers about the core concept of one of the modern LCDPs Microsoft PowerApps and its features, merits and demerits. The fourth section “Low-code Development with PowerApps” is the demonstration section of creating a simple business prototype that handles the data and apply basic CRUD operations to manipulate the data as necessary. The fifth section “Results” shows the outcome of the thesis and the project prototype and the comparison with the requirements. Lastly, the thesis ends with a conclusion.

2 LOW-CODE AND NO-CODE DEVELOPMENT

No-code and low-code development are innovative approaches that enable people with variety of technical skills to build software and web applications without following traditional programming approach. No-code platforms need little to zero coding knowledge, relying on visual interfaces and readymade modules or templates for program development instead. Low-code systems, on the other hand, require some coding experience, but at a more abstract level, allowing for faster creation through graphical user interface and pre-assembled modules. (Woo 2020.)

The evolution of development approaches has seen a shift from manual coding to higher degrees of abstraction. Classical coding required skills in languages like Java, JavaScript, Python, PHP and many other languages. The emergence of low-code and no-code techniques has made software production more accessible, allowing for more participation in the process from business users and non-developers. Both no-code and low-code conceptual frameworks focus around shortening the app development cycles, reducing direct dependency on specialist experts, and encouraging collaboration among technical and non-technical stakeholders. This strategic advancement attempts to improve efficiency, agility, and inclusion in the application development process. These frameworks are positioned to play a crucial role in creating the future landscape of software development as the technology advances. (Rokis & Kirikova 2023.)

2.1 Evolution of Development Approaches

Advances in innovation, changing advertising needs, and the needs to speed up the development process have all fueled the development areas of software development approaches. Early process, such as Waterfall Model, depended on a consistent, straightforward rule, with each stage must be wrapped up before moving on to the next one (Petersen, K., Wohlin, C., & Baca, D 2009). In any case, within the boundary of persistently changing necessities and advertising pattern, this strict system needed to be consistent. More iterative techniques, such as Rapid Application Development (RAD), replaced the waterfall model around the 1970s. In order to enable quick prototyping and adaptability to change requirements, RAD placed a strong impact on partner and engineer communication. Businesses seeking to shorten their advancement period, provided programs more quickly to demonstrate began to employ the strategy. (Chrismanto, Santoso, Wibowo, Delima, & Kristiawan 2019.)

In 1990s, the generation saw the trend of agile methodologies such as Scrum and Extreme Programming (XP), reducing the development process into smaller periods referred to as sprints, agile methodologies embraced flexibility and responsiveness in its workflow (Simplilearn 2023). This approach was suitable for the rapidly developing modern mode of software development because it provided constant feedback and improvements. With the concept of DevOps, a corporative model that pairs development application with IT operations, software development lifecycle is now shorter. It supports efficient organization and input rings by its communication, computerization, continuous delivery (Educative 2024). The degree of this problem has risen correspondingly with increasing complexity and trade and also the connectivity of the program.

Platforms for low-code and no-code creation have surfaced recently, expanding software development by enabling non-technical individuals to create web apps and software with little to no programming experience. With the use of these platforms' drag and drop interface, prebuilt parts, and visual programming tools, people and organizations may create applications without needing to have much programming skills. The rising need for quicker, more flexible software development that can accommodate the demands of a larger user base is reflected in the trend towards low-code and no-code development. These platforms allow companies of all sizes to develop and adjust to a constantly changing digital world by reducing difficulties and boosting productivity. (Böck & Frank 2021.)

2.2 Essentials of Low-code and No-code Development

The principle of low-code and no-code development involves choosing the right platform, mastering its use, utilizing visual tools, promoting reusability, and leveraging automation capabilities. These principles aim to streamline application development, increase productivity, and reduce maintenance efforts. Selecting the right low-code or no-code platform is crucial for efficient development. It involves considering features, intended scenarios, and organizational needs. Avoiding vendor lock-in is also essential to protect investments. Learning and mastering the chosen platform is vital for unleashing its full potential. While initial development may be straightforward, understanding platform features, integrations, and development processes ensures optimal outcomes. (Rokis & Kirikova 2023.)

Visual application development plays a key role, allowing developers to create applications without extensive manual coding. Predefined components, elements, and templates provided by the platform enhance efficiency and reduce development time. Reusability is emphasized to efficiently utilize previously created artifacts, leading to increased productivity and reduced maintenance efforts. This principle extends across various components, including data models, UI elements and testing. Automation capabilities integrated into low-code and no-code platforms facilitate tasks throughout the development lifecycle. From generating applications automatically to deploying and scaling, automation speeds up the application delivery process. Customization is encouraged but should be fair. While additional customization features allow for a higher degree of personalization, developers must balance the benefits with increased development time and complexity. Empowering citizen developers, individuals with minimal coding experience, in a collaborative fusion team with IT professionals enhances development speed and aligns better with business needs. Promoting IT-business collaboration ensures faster translation of business requirements into functional applications. Regular feedback and continuous refinement through quick iteration improve communication and project outcomes. (Rokis & Kirikova 2023.)

Establishing governance is also crucial to prevent security and IT governance concerns associated with fast application delivery. It ensures that proper measures are in place to uphold security and compliance standards. Following an iterative development lifecycle, embracing a test-and-learn culture for innovation, and supporting changing requirements are additional principles that contribute to the success of low-code and no-code development (Rokis & Kirikova 2023). These principles collectively drive agility, responsiveness, and innovation in the software development process.

3 POWERAPPS FUNDAMENTALS

A complete set of tools called PowerApps was invented to let developers create customized apps with little code needed to server both individual users and businesses. It operates in a low-code environment and connects to Microsoft Office services with ease. PowerApps works with typical web browsers and mobile platforms (iOS and Android). PowerApps effectively use integrated data sets and operate within the Office 365 domain. The development of applications can be divided into three distinct paradigms, the Model-driven paradigm, Canvas-driven paradigm, and Portals apps paradigm. The Model-driven paradigm, in which data is sourced from supported connectors to enable ordered and visually compelling data representation. The Canvas-driven paradigm, which allows users the freedom to create interfaces from scratch using simple drag and drop features. Whereas Portals apps allow the creation of canvas or model-driven apps to share internally with authenticated users. (Palmer 2020.)

An essential part of Office 365, PowerApps enables smooth data sharing between many Microsoft products, including Excel, Power BI, and SharePoint. PowerApps also promotes integration inside Microsoft's business ecosystems. With PowerApps, users can create, share, and execute programs on any device by utilizing a cloud-based architecture. It may be integrated with several different data sources, including on-premises systems and cloud services like Dynamics 365, Microsoft 365, and Azure. It is less simple to create apps that make use of current data and services because of this architecture's seamless data communication and integration capabilities. The compatibility of application deployment across several platforms is further increased with the release of the PowerApps container application. Like Microsoft Excel or Access, PowerApps improves workflow efficiency with a little coding needed. Knowing the software inside and out is crucial since it uses a variety of formulas and functions to carry out front-end and backend operations. As a result, PowerApps becomes a competent and intuitive tool that facilitates quick and effective application development process in both university and business settings. (Palmer 2020.)

3.1 Key Features

Microsoft PowerApps offers a range of key features that make it a powerful tool for app development. One of its main features is its low-code development platform, enabling businesses to create custom apps without extensive coding experience. The PowerApps platform offers drag and drop features which allows users to create applications without needing to write a single line of code which makes it more understandable to the users with different skillsets and coding knowledge. Along with providing flexibility and interaction with other products and services offered by Microsoft, PowerApps also shows its compatibility with Excel, OneDrive, and SharePoint (Microsoft 2022). Moreover, Common Data Service (CDS), which is a cloud-based data storing and managing service to record and distribute data across platforms, is another feature provided by PowerApps. Using this feature, organizations may get a more complete view of their data, helping business to discover patterns, make intelligent choices, and acquire competitive advantages. PowerApps offers responsive designs, allowing users to create designs which are compatible in various screen sizes across browsers and mobile devices. Another important feature is the AI Builder, an artificial intelligence platform that gives businesses the ability to develop intelligent AI models in addition to pre-built models that can be used for a variety of purposes (Singh 2023). These features, combined with its user-friendly interface and easy connectivity with different cloud services, make PowerApps a unique, versatile, and valuable platform for any business looking to create custom apps and improve productivity.

Low-code platforms, known as LCPs, typically contain elements, like tools for creating data structures and accessing data sources through APIs. User interfaces are user-friendly allowing for the design of interface using predefined widgets. Functional features cover decision making rules and standard operation along with support for calling functions through APIs (Böck & Frank 2021). Also, deployment methods vary from installation on web servers to applications on devices. All LCPs have components for roles and user permissions in their structure. Some additional features may include workflow modeling and engines that often use Business Process Model and Notation (BPMN) or custom structures. Certain platforms also offer coding features for programming languages such as Java and JavaScript. Leading service platforms provide units for business intelligence, intelligence, and robotic process automation. Less common features involve domain reference implementations. A few platforms provide libraries of reference data models and functions. While some systems offer collections of functions, they are usually generic or limited in scope. (Böck & Frank 2021.)

3.2 Application Types

There are mainly three different types of apps that can be designed with PowerApps. They are Canvas app, Model-driven app, and Web portal apps. Each app can be made to accommodate various project sizes, kinds, and development requirements. Their approaches to app's general design are where they diverge most. Canvas apps provide a blank canvas for flexible creation which involves creating a customized and intricate interface with a strong focus on usability. Like making a PowerPoint presentation or utilizing common prototype tools, developers can choose components from menu and drag and drop items into the respective place as required. Model-driven apps make use of pre-existing data structures for a more structured approach. These applications work best in situations where handling vast volumes of data is the primary focus of it and the user interface is much simpler. Lastly, web portal apps concentrate on offering external access to the application. One of the unique features of Portals is that users can access the sites using a variety of pre-installed identifiers such as login credentials, without necessarily being employees of the user company. With these alternatives, developers may select the most appropriate method of creation according to the needs of the project. (Microsoft 2023; Tollefson 2022.)

3.2.1 Canvas Apps

The most visually simple kind of PowerApps app development is called a Canvas app. From a blank canvas app, it implements creating a customized and intricate interface with a strong focus on usability. Similar to making a PowerPoint presentation or utilizing common prototype tools, developers can choose things from menu and drag and drop items into the respective place as per the design. Once ready, the interface may connect to a variety of data sources, and canvas enables the creation of particular logic pipelines for data using expressions similar to those found in Microsoft Excel (Microsoft 2023). However, it is essential to note that Canvas apps can be time consuming for data-oriented requirements. Canvas apps may lack responsiveness, with a fixed screen size, and creating multiple apps may be necessary for different screen sizes. Additionally, grids/views can be complex and may require some coding knowledge for sorting and searching functionalities. (Tollefson 2022.)

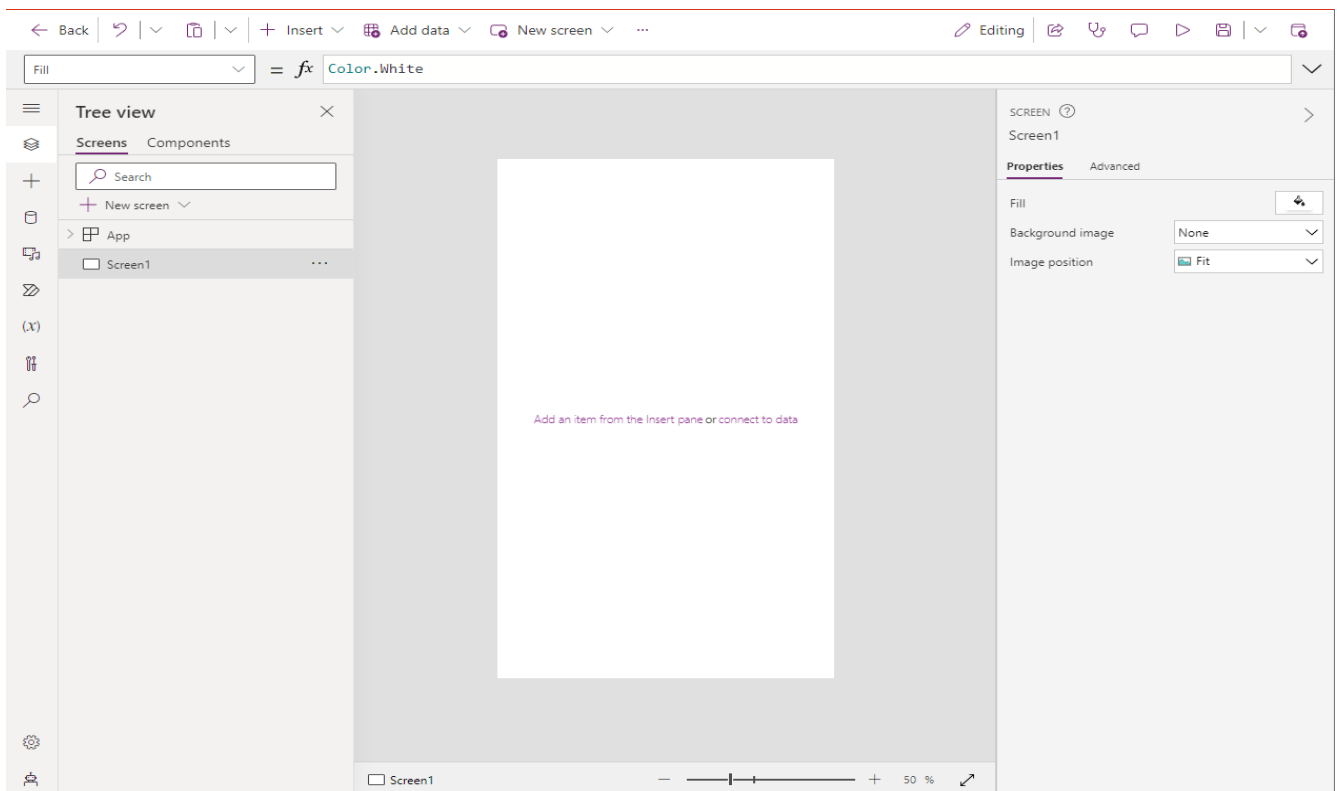


Figure 1 Blank canvas app layout.

3.2.2 Model-Driven Apps

Model-driven app development offers benefits including fast build process, a uniform appearance across different devices, and simple environment transition because a larger portion of the user experience is decided by the components which are added to the layout. Model-driven apps, as compared to the Canvas apps, start with a basic data model, which makes them ideal for tasks requiring a higher degree of data quality. These applications work best in situations where handling vast volumes of data is the primary focus and the user interface is much simpler. By using this component, the user can quickly develop a model-driven application that allows users to see, update and manage data through configurable dashboards, forms, and views. Users may also add a personalized touch to Model-driven layout by embedding Canvas app, even if the UI/UX may not be as customizable. (Microsoft 2023.)

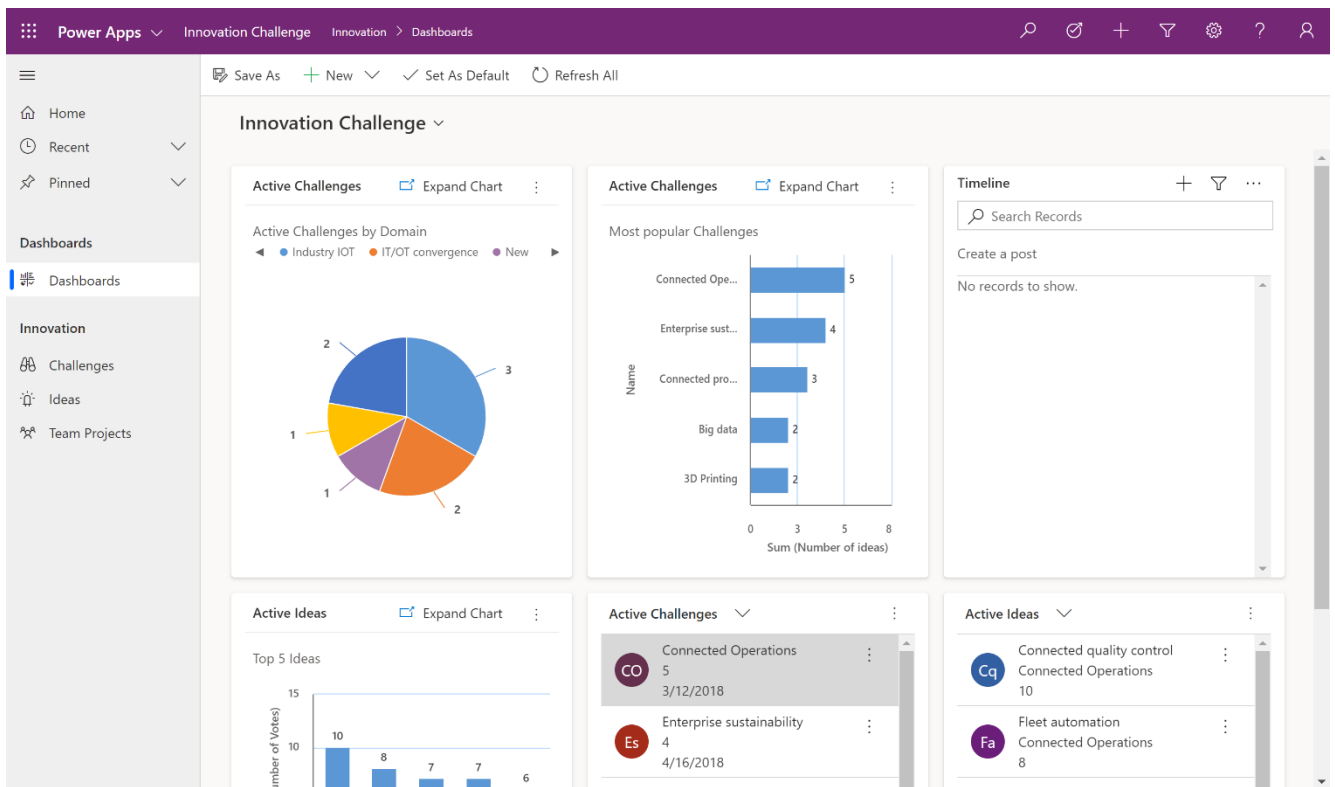


Figure 2 A dashboard in Model driven layout with different sections (Microsoft).

3.2.3 Web Portals Apps

The third application of PowerApps developed in 2019 offers a unique user experience by enabling the development of websites with an external facing page. One of the unique features of Portals is that users can access the sites using a variety of pre-installed identifiers, such as login credentials, without necessarily being employees of the user company. As a great breakthrough in the application development without necessitating advanced coding knowledge, online users can read and edit certain available data even in the absence of corporate credentials (Microsoft 2023). In other words, through Portals, users from outside the company may engage with Dataverse if they are external websites. Security roles can be provided to external users, allowing them access to important data while preventing any misuse with the data. Portals may now be set up to function independently with Dataverse, but they were originally connected to Dynamics 365 customer interaction apps. Additionally, Portals' responsive default user interface may also be customized to match corporate identity. (Tollefson 2022.)

3.3 Merits and Demerits of Using PowerApps

Application creation that is quick and simple is made possible by drag and drop feature of PowerApps, which requires little to no coding experience to create a unique application. This makes it possible for companies to create applications fast and effectively without needing to hire costly developers. Since PowerApps is a low-code platform, it takes less time and money to build an app than traditional coding approaches. When designing several applications, this may save firms a significant amount of time and capital. PowerApps and Microsoft Dynamics 365 are closely connected with each other. Microsoft Dynamics is a well-liked enterprise resource planning (ERP) solution. This facilitates the process of linking PowerApps applications to Dynamics 365 and workflows that already exist. PowerApps programs may be installed on a range of devices, including desktops, laptops, tablets, and smartphones, with smooth cross-platform connectivity. This makes it possible for businesses to provide their staff members with access to apps from any location at any time. As discussed about powerful customization possibilities in section 2, PowerApps provides several modification options, allowing companies to adapt their apps to the exact requirement. This includes the availability to bring in new data sources, connect to other services, and define custom controls. (Juuti 2021.)

On the other hand, PowerApps has limited scalability and performance. A more traditional development platform may be a better fit for projects that require a big number of users and data sets. PowerApps may not be secure as those created using standard coding languages. This is due to PowerApps' dependency on Azure Active Directory for login and permissions, which may be less secure than customized security solutions. While PowerApps is simple to learn for those who have no prior knowledge of coding or programming languages, companies may need some training to effectively develop and manage PowerApps applications. Model-driven applications have limited customization availability. Apps are expected to be as quick as possible and simple to construct and use, but it gives less flexibility than Canvas-driven applications. This means businesses may have to give up certain customizations possibilities to save time and development expenses. (Juuti 2021.)

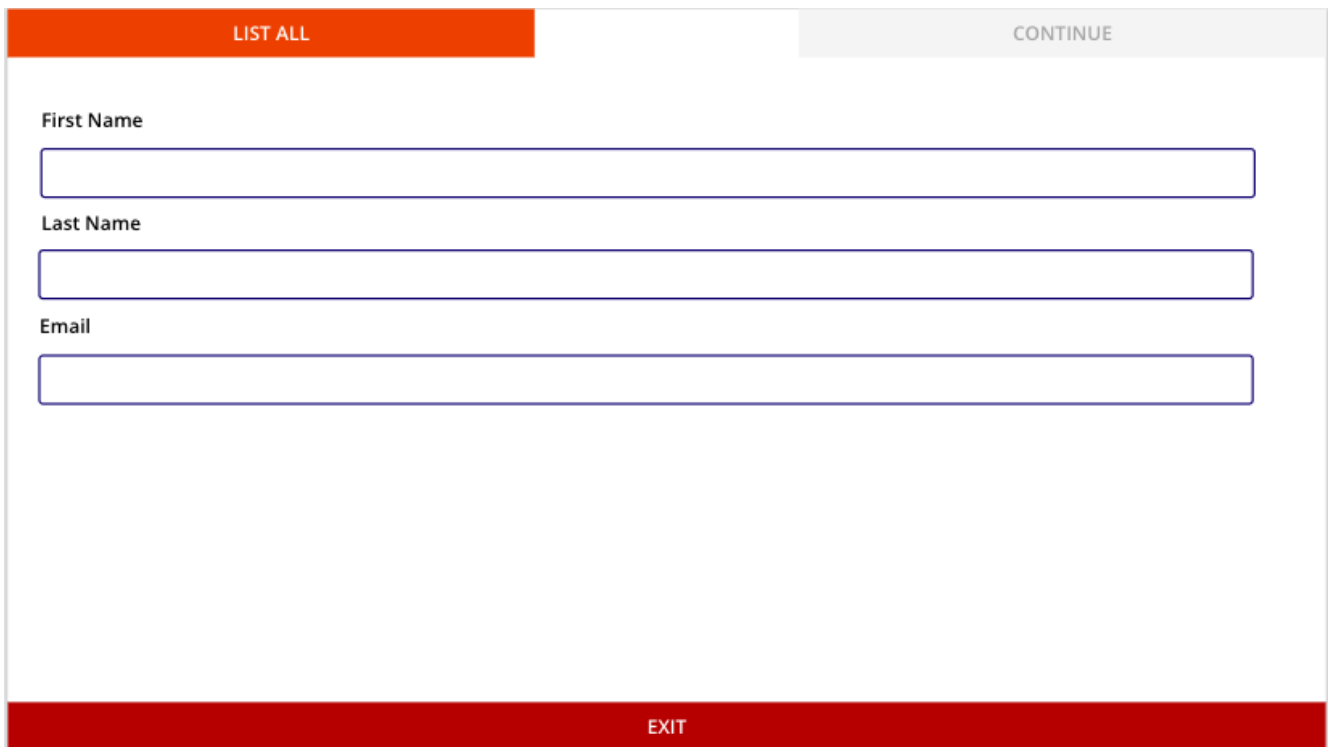
4 LOW-CODE DEVELOPMENT WITH POWERAPPS

This chapter will explore the PowerApps user-friendly characteristics as set out to simplify the process of making a business application. This section demonstrates the application's adaptability and ensures smooth operation on several devices, even for those with no prior coding knowledge. In addition to that, simplicity and efficiency have come together to create an app whose capabilities are enhanced by its iteration with Excel. The goal is to enable users to easily manage data within the application. With PowerApps, users may change records, delete entries, read old data, and add more new data. These important tasks are significantly easier to do because of its user-friendly UI. This bridges the gap between people with diverse technological backgrounds and demonstrates the viability of both low-code and no-code development approaches. The purpose of this research is to show how the application development process may be enhanced with PowerApps. Because of its user-centric design and integration option, it offers an interactive and visual environment that inspires creativity and is reachable to anyone without any prior coding skills. This chapter, which serves as a practical guide, addresses the aspects of developing a business application that best utilizes a low-code technique.

4.1 Requirement Analysis

The application designed in PowerApps has a landing page with three input sections as shown in Figure 3: "First Name," "Last Name," and "Email," as well as three buttons: "LIST ALL", "CONTINUE", and "EXIT". This layout illustrates a business application designed for basic CRUD (Create, Read, Update, Delete) operations on data saved in Microsoft Excel. PowerApps' versatility is demonstrated by its easy connection with diverse data sources in the Dataverse, providing dependable and simple to use data storage. With the help of different components of PowerApps like text input, gallery, and buttons. A user-friendly UI is developed with components like text input field, buttons to save the data, to continue to next or back to previous page and to exit the app. When users enter data into all three fields, the program prompts them to save the information on the next page. Users may then see, edit, change, or remove the input data shown as a gallery from the data source, which improves the application's usability and data management, and users may edit the data gallery later the way users required to display the data. This user-friendly interface, together with PowerApps' features to connect with a

variety of data sources, displays the platform's versatility in developing efficient and engaging business applications.



The image shows a desktop version of an application landing page. At the top, there is a navigation bar with two buttons: "LIST ALL" on the left, which is highlighted in orange, and "CONTINUE" on the right, which is in a light gray box. Below the navigation bar, there are three input fields stacked vertically. The first is labeled "First Name", the second "Last Name", and the third "Email". Each label is positioned to the left of its corresponding text input box. At the bottom of the page, there is a dark red footer bar containing the "EXIT" button in white text.

Figure 3 The landing page of the application that stores First Name, Last Name and Email as an input (Desktop version).

Another feature of the application is that the application must be responsive for all the digital devices available, such as for mobile phones, tablets, and bigger displays like desktops and laptops. The primary focus of modern developers is to make applications that are responsive for mobile phones. Mobile phones are widely used all over the world and any websites or any business applications must have their mobile friendly version available. Figure 4 below shows the compatibility of this software application with all possible digital devices like mobile phones and tablets.



Figure 4 Two different responsive versions of the same app for Mobile and Tablet.

To build a functional and responsive app, it is important to analyze the requirements of a responsive application. Upon analyzing the requirements of a business application, the app must be device and user-friendly, and have data storage for data manipulation. The data source for this application is Microsoft Excel in Microsoft OneDrive Business. All the data is stored in Microsoft Excel file. Any operation on data is done with in the Excel file which is directly connected to the PowerApps application and users can interact with the user-friendly UI design. The changes made to the data can also be saved and stored in the existing source.

4.2 Development Process

This section briefly explains the steps for creating an application in PowerApps. All the steps are explained in details with figures. The application's needs and purpose were first defined in a limited way. Regarding the target audience, goal, and intended needs for various features in this description. Using Microsoft's Dataverse, a pre-planned database description served as the basis for developing the real application. Relationships between the data were added for connections, and tables tailored for the application were developed in Dataverse. The program was designed to extract data about users in order to utilize PowerApps from the user database. This allowed the user to utilize the program while using a browser logged in with Microsoft account.

4.2.1 Creation of the Application

The application was created by selecting an empty Canvas-type application as the base of this application. This creates a new version of the application which is without any initial preset data connections to the application. Choose the Blank app to create the app (see Figure 5), then select Blank canvas app to continue with the naming the application (see Figure 6) and select Create. This will generate a fresh new canvas screen (see Figure 1).

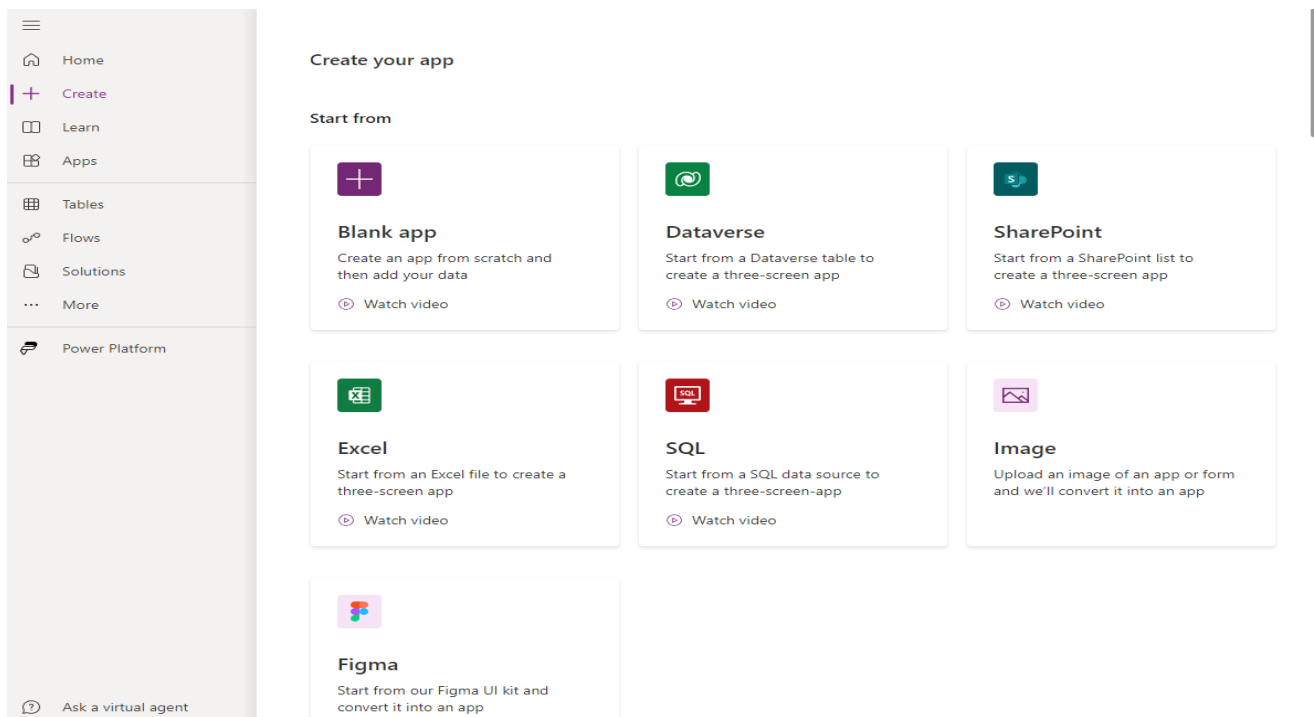


Figure 5 New application options to create.

In Figure 5, there are multiple choices for different purposes in app development. Blank app is used to create Canvas apps or website like applications. Dataverse is used for building business applications that allows developers to securely store and manage data used by PowerApps applications. SharePoint allows developers to store, organize, share, and access information from any devices, making it a useful tool for document management, and workflow automation. Excel is a popular spreadsheet allowing programmers to leverage Excel data within the application. SQL refers to the ability to connect and interact with SQL server databases within PowerApps application. Image option is used to create interface by uploading an image which will be converted to a working app. Figma is an application like PowerApps with drag and drop features to create UI designs. After selecting the first option, the page will be directed to the next page (see Figure 6).

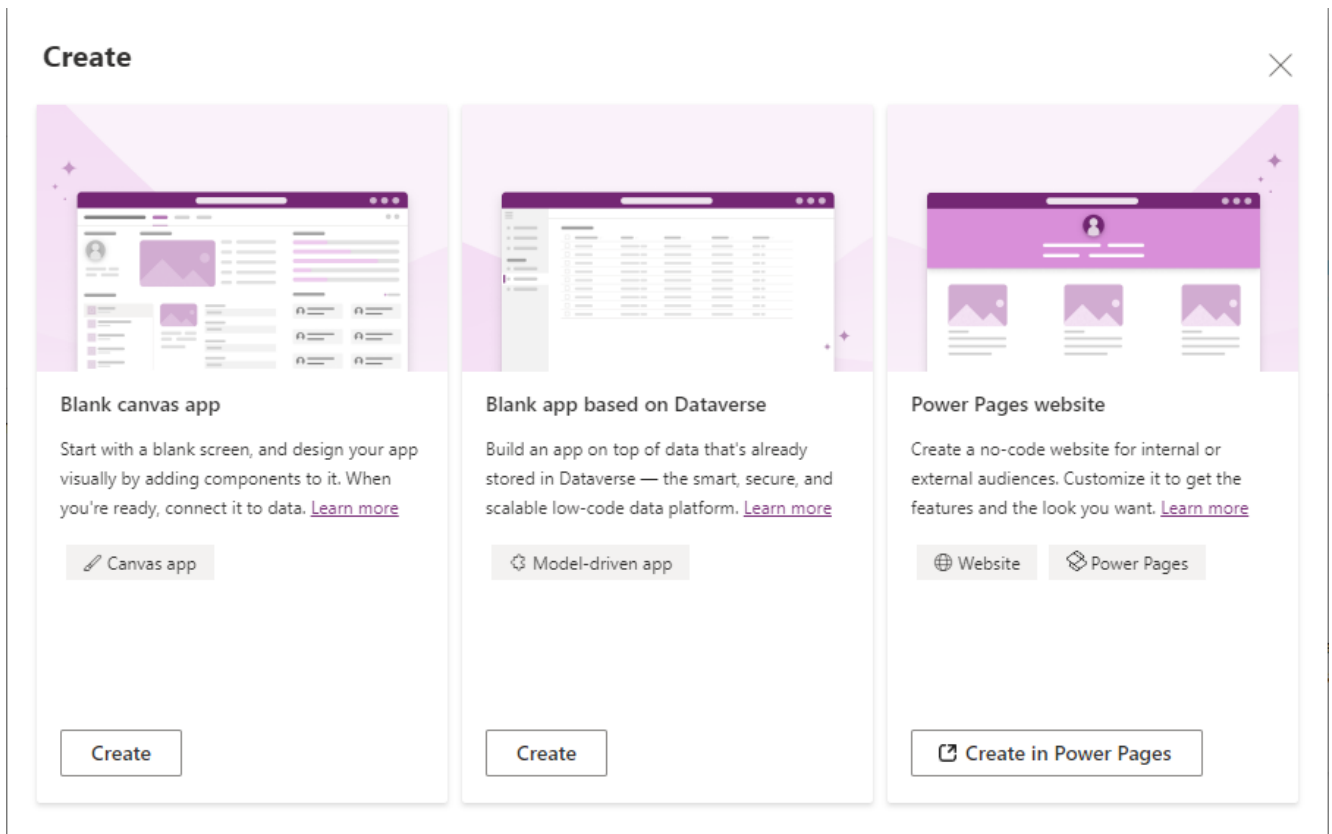
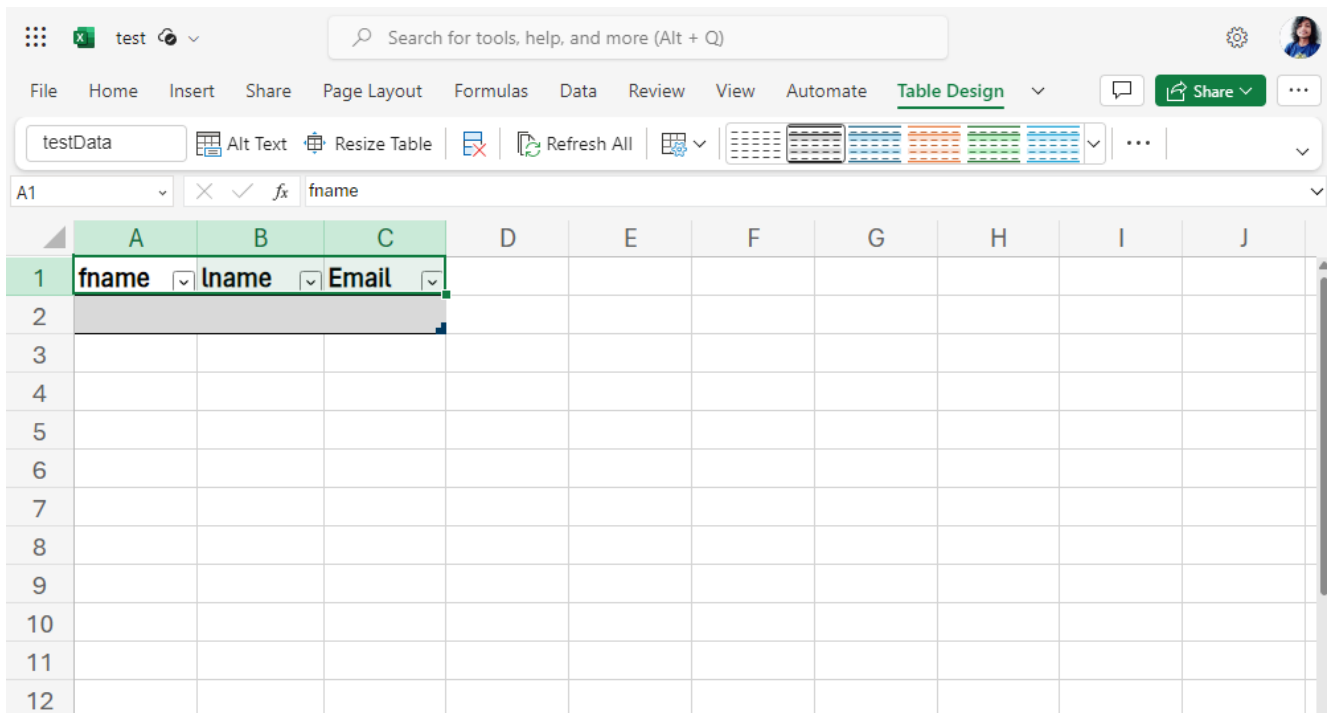


Figure 6 Options for different types of applications.

4.2.2 Connecting application to the Data Source

In this section, the application is required to be connected to a data source. In this case, one Excel file is created in OneDrive environment where all the data from the application is stored. In figure 7, the name “testData” is the name of the table where first name, last name and email address is stored directly from the application. Once the Excel file is created and table is generated as shown in the figure below. The excel file needs to be connected to PowerApps.



The image shows a screenshot of an Excel spreadsheet. The spreadsheet is titled 'test' and is in the 'Table Design' tab. The table is named 'testData' and is located in the range A1:C3. The table has three columns: 'fname', 'lname', and 'Email'. The first row (A1) contains the headers 'fname', 'lname', and 'Email'. The second row (A2) is empty. The third row (A3) is empty. The spreadsheet is displayed in a grid format with columns A through J and rows 1 through 12. The 'Table Design' ribbon is visible at the top, showing various options for formatting and managing the table.

	A	B	C	D	E	F	G	H	I	J
1	fname	lname	Email							
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										

Figure 7 Excel table where the data is stored

The file or the Google Chrome tab must be closed before connecting the Excel file to the application in PowerApps. On the left-hand side, from the hamburger menu, select the tab “Data” and click “Add data” to add the data source. OneDrive for Business is selected from the search as shown in figure 8 below. This allows to access the Excel file created to this application as the primary data source. There is no need to have separate storage for the data. Any data from the application can be directly online without worrying about the files being deleted or corrupted.

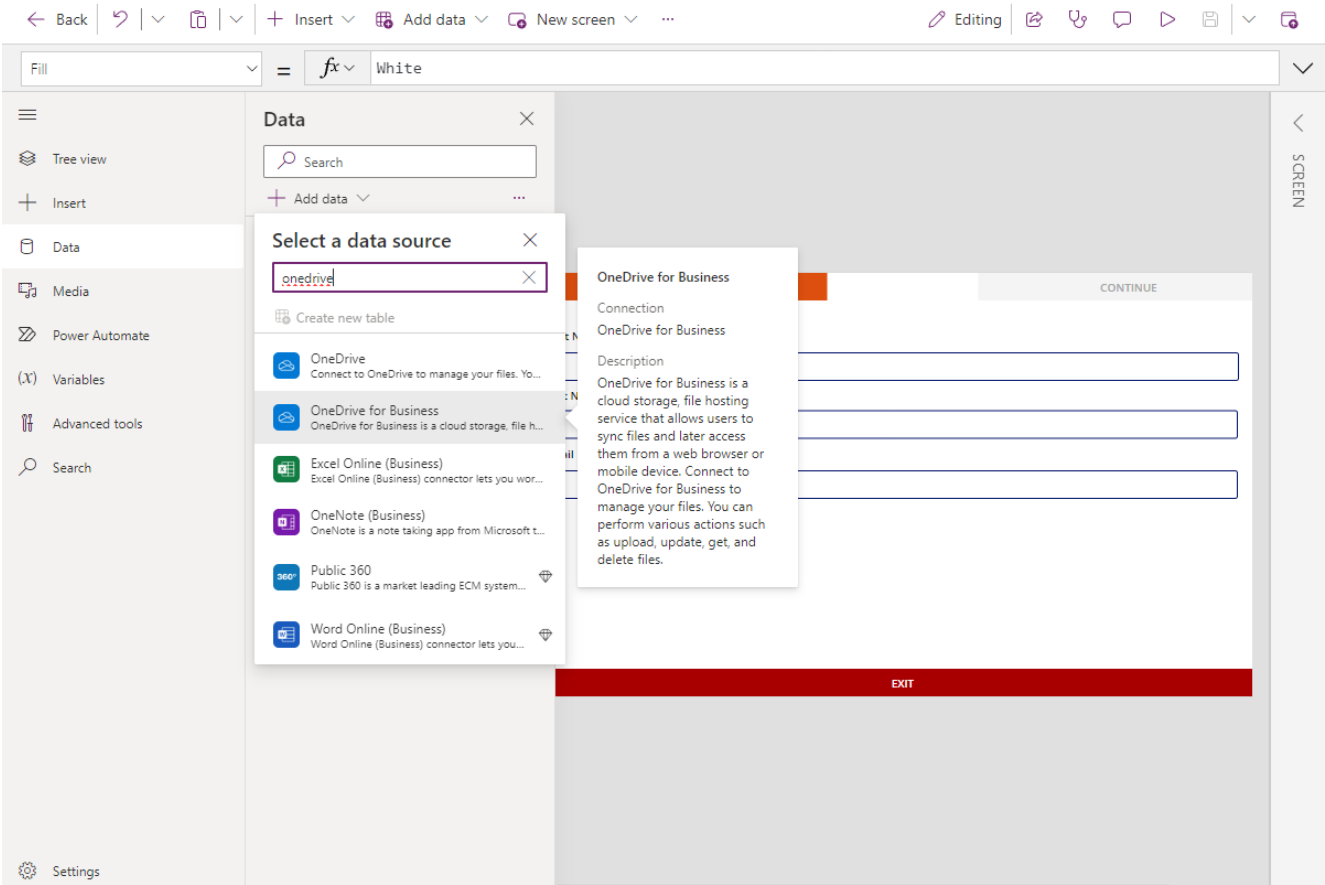


Figure 8 Selecting OneDrive for Business for data source

Once the OneDrive for Business is selected from the search, another tab is opened on the right-hand side of the screen. From there, the excel file is selected and press connect to connect the file to PowerApps. The table name “testData” must be shown when the Excel file is selected, which shows the PowerApps is successfully reading the data file from the OneDrive. In this way, the information received from the tables could be combined to create components and display information to the user from PowerApps. Once the data source is connected to PowerApps, the designing part of the application can be started.

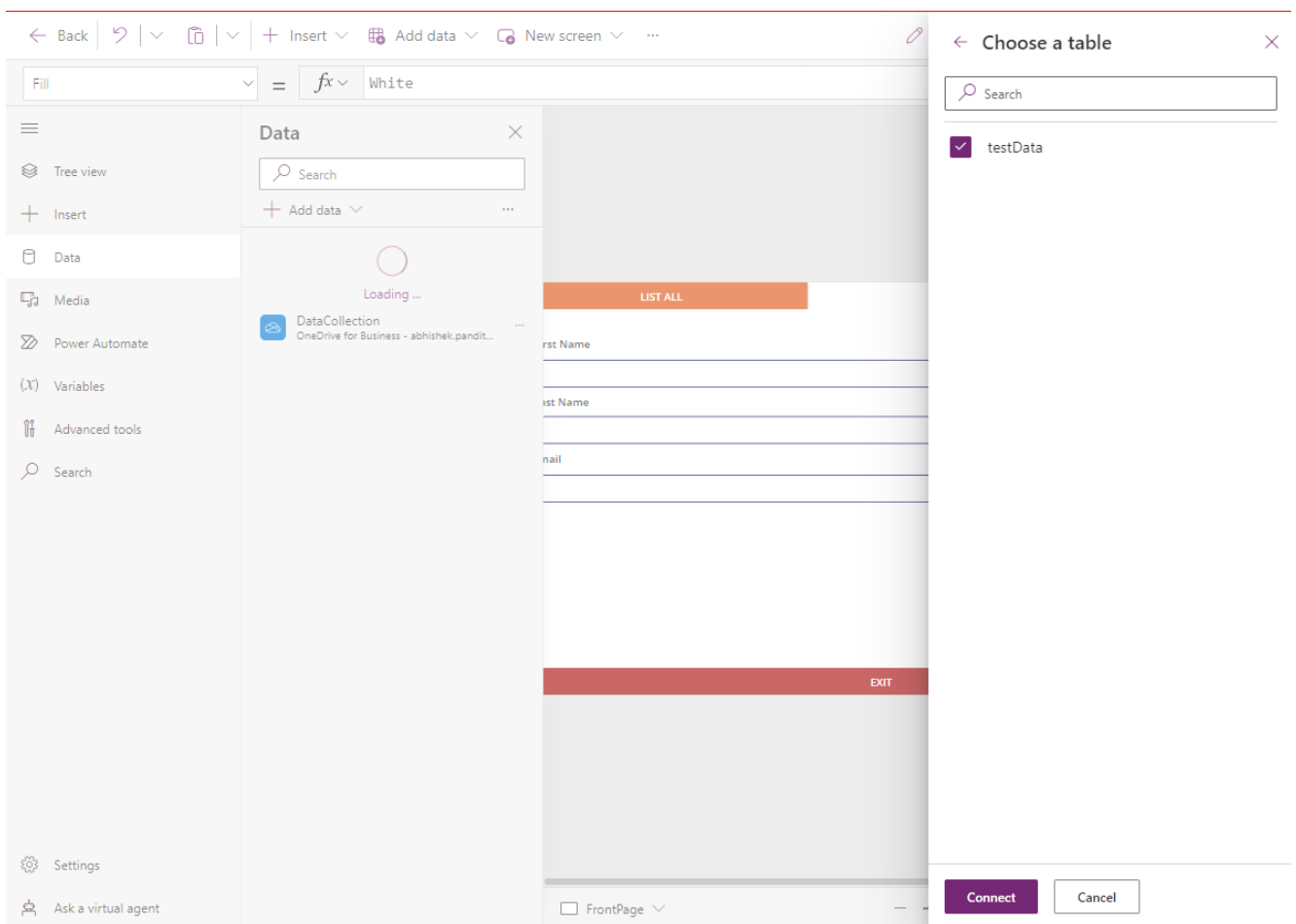


Figure 9 selecting the table from the excel file to store the data from PowerApps.

4.2.3 Developing the Application

Generally, there are 6 major components of PowerApps that can be used to create user-friendly interface. They are gallery, screen, card, control, property, and function. Each component is useful on their own way. Missing one or two components in the PowerApps can lead to bad UI designs and functionality of the application. The design should also be responsive for all the screens and to make the UI design responsive, a small change needed to be adjusted before adding components to the screen. As shown in figure 10, in display settings, scale to fit option should be turned off. Once it is turned off, the design can be made responsive by manually adjusting the height and width of the components on the screen which might require some little coding.

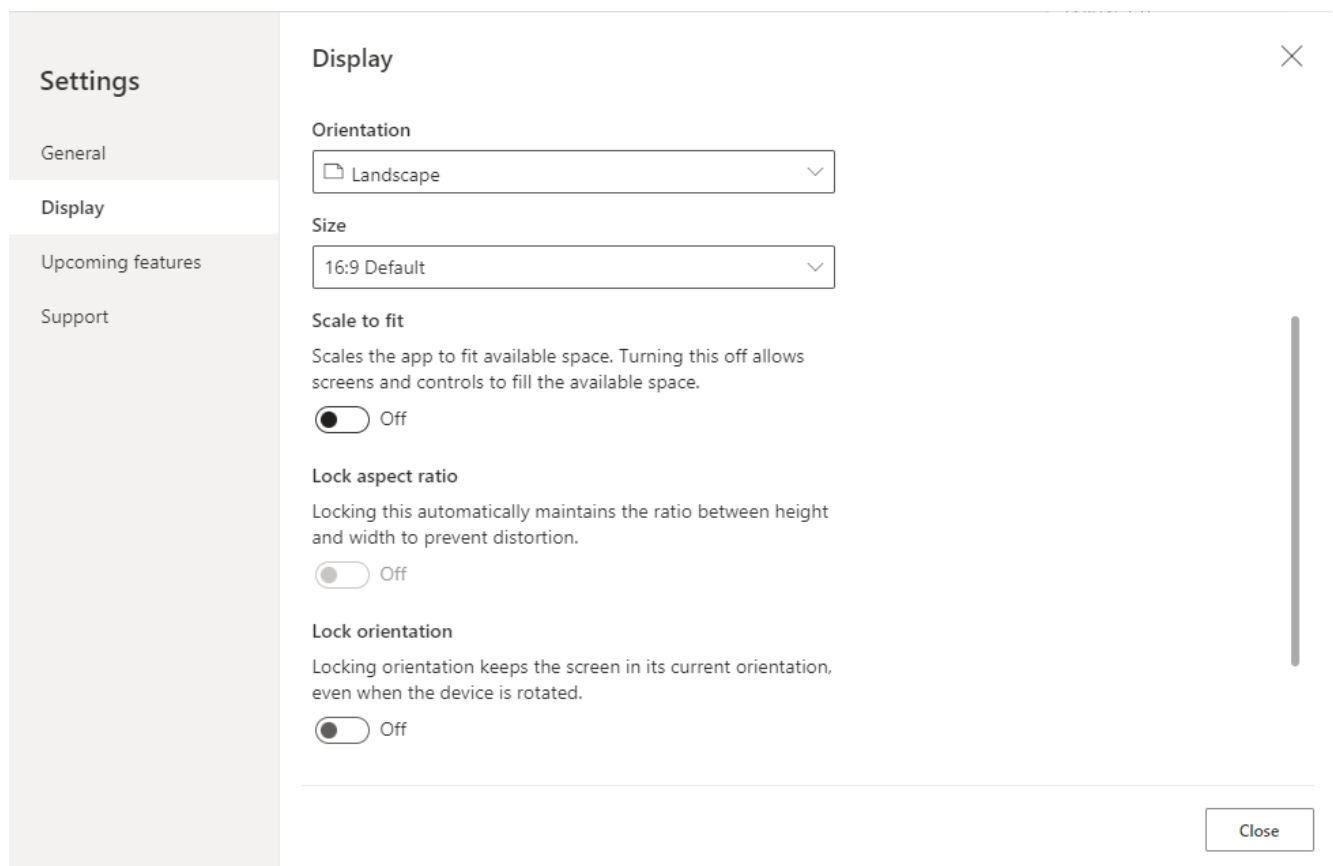


Figure 10 changing display setting to work with responsive designs.

The application consists of four different pages, front page, second page, last page, and detail page. The design of the front page of the application consists of text labels, input components and buttons. First, three buttons were added to the screen with the help of “Insert” dropdown menu from the tab. Two buttons “LIST ALL” and “CONTINUE” were placed on top of the screen on each side and “EX-

IT” button at bottom of the screen. Then text labels were added to the screen as “First Name”, “Last Name”, and “Email” and three text inputs were also added to the screen respectively. The form contains three inputs as the form which takes first name, last name, and an email address. An algorithm is used to ensure if the input email data is in valid email address format. Unless the user enters a valid email id format, the “CONTINUE” button will not be activated to get the user to the second page as shown in figure 11. On the other hand, even if the user fails to input all the fields, the button still does not activate (see Figure 12).

The figure shows a form with three input fields and two buttons. The buttons are 'LIST ALL' (orange) and 'CONTINUE' (greyed out). The input fields are labeled 'First Name', 'Last Name', and 'Email'. The values entered are 'hello', 'world', and 'hello.worldgmail.com' respectively.

Figure 11 The CONTINUE button is deactivated unless all the fields are correctly given.

The figure shows the same form as Figure 11, but with the 'CONTINUE' button now activated (orange). The email address entered is 'hello.world@gmail.com', which is a valid format.

Figure 12 The CONTINUE button activates when all fields are in correct format.

Another important thing to note is that the width and height of the form expands or contracts detecting different sizes of devices. The formula used to make itself adjustable according to the screen sizes. The formula states that if parent width is less or equal to 541 pixel screen size then the width of the input

field size is multiplied by 0.80 and if greater than 541 px then new width will be the multiplied value of current width by 0.92. This is how the UI fixes itself every time it detects screen size change.



Figure 13 Use of If statement to ensure the UI detects the screen size and adjust itself.

The code that was used to check the empty fields for the inputs and email validity, the code (see Figure 14) is used to ensure the functionality of the application works as expected. The following figure 14 shows a conditional if statement that checks for availability of first two inputs for first name and last name and checks the validity format of the email address. The code must be written in the “Display-Mode” property of the “CONTINUE” button.



Figure 14 Code to check if all input fields have valid inputs.

The second page is the confirmation page to save the data input from the first page. The layout of the second page is similar to the first page with a checkbox at the end of the data form. The user inputs the data, and the same data is shown to the second page when user navigates to the second page. The “SAVE” button in the top right also checks if the checkbox is clicked. Similar code is used to check the availability of the checked checkbox as in figure 15. The idea is to display the data which the user is about to submit, if the data information is correct then the user can check the checkbox and can navigate to the third page with the help of “SAVE” button. The formula used to store the data is with Collect() function two inputs, the name of the table in excel and array of input take from first page. This formula collects data and copies it to the excel table, which is later displayed in the third page.

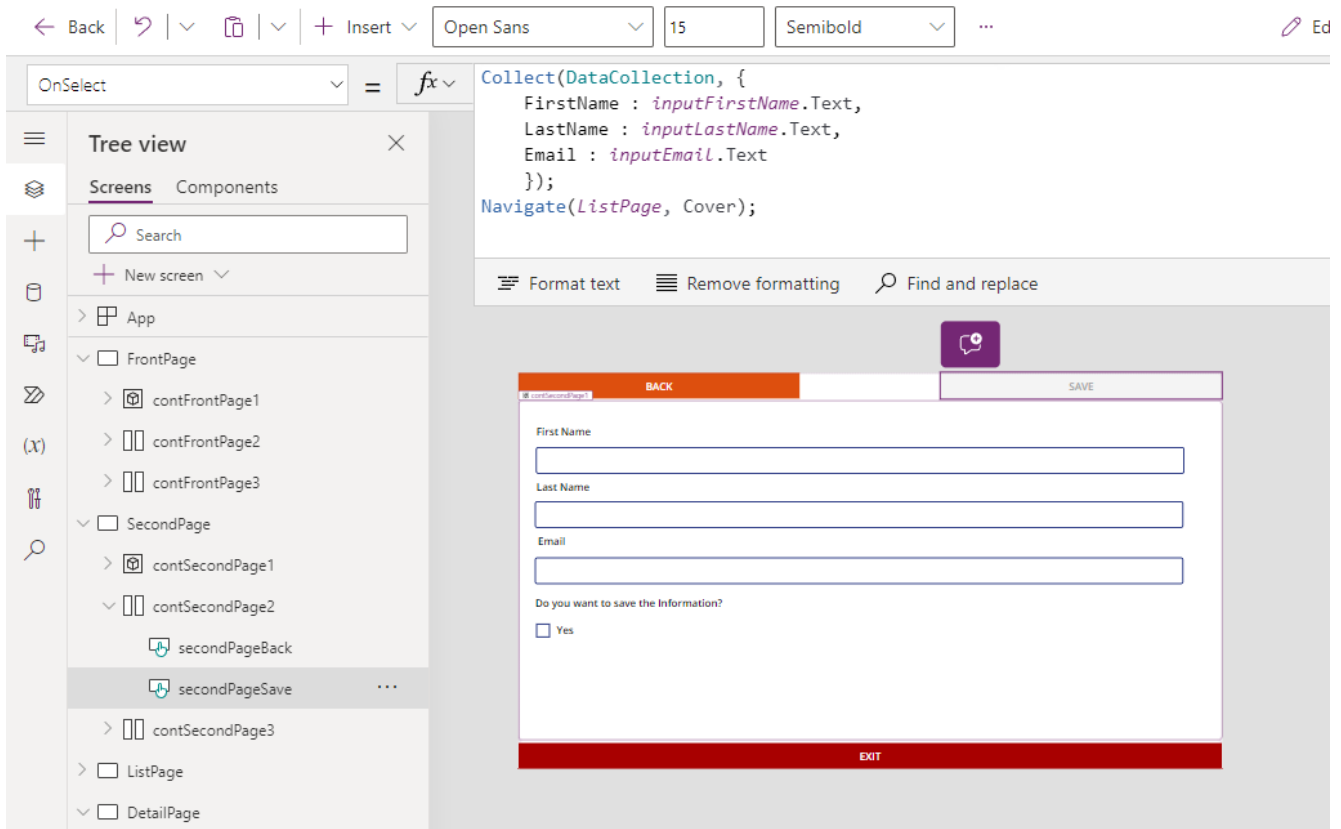


Figure 15 Collection function with input taken from the first page to collect and store data in excel

The third page of this application is the list page, where the user can view all the data stored. This page was designed with the help of gallery component of the PowerApps. To list all the contact information of the users, the gallery component was used, specifically Vertical gallery from the insert menu. Data is connected to the gallery through which the information from the excel can be retrieved and displayed in the third page. Data layout can be customized according to the requirements. The third page also has three buttons “BACK”, “ADD NEW”, and “EXIT”. The “ADD NEW” button takes the user to the first page where the user can refill all the information to add a new contact.

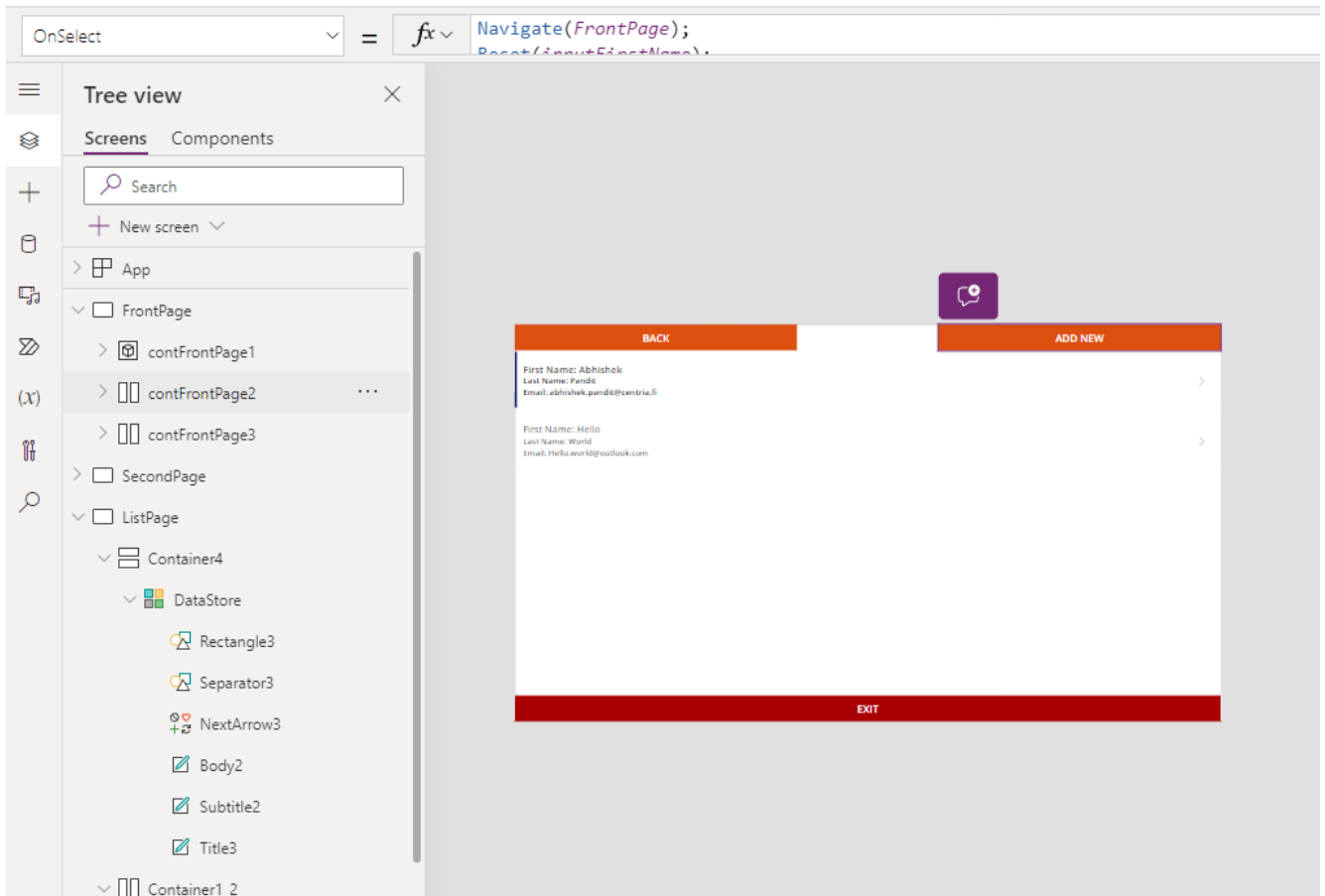


Figure 16 Third page of the application ListPage

The final page of this application is the detail page. From the third page, if the user wants to update the data, then by clicking on the data, the user is directed to the detail page where user can modify the data and save it. The data will also be modified to the excel table where all the data is stored. This page has “SAVE” button to the top right corner of the page. The formula used to edit and save the new data over the old data can be achieved using Patch() function. Patch() function takes three inputs Data-source, BaseRecord, ChangeRecord which is an array (Microsoft 2024). Here in the application the patch function takes DataCollection, varRecord, and list of arrays of contacts as shown in figure 17.

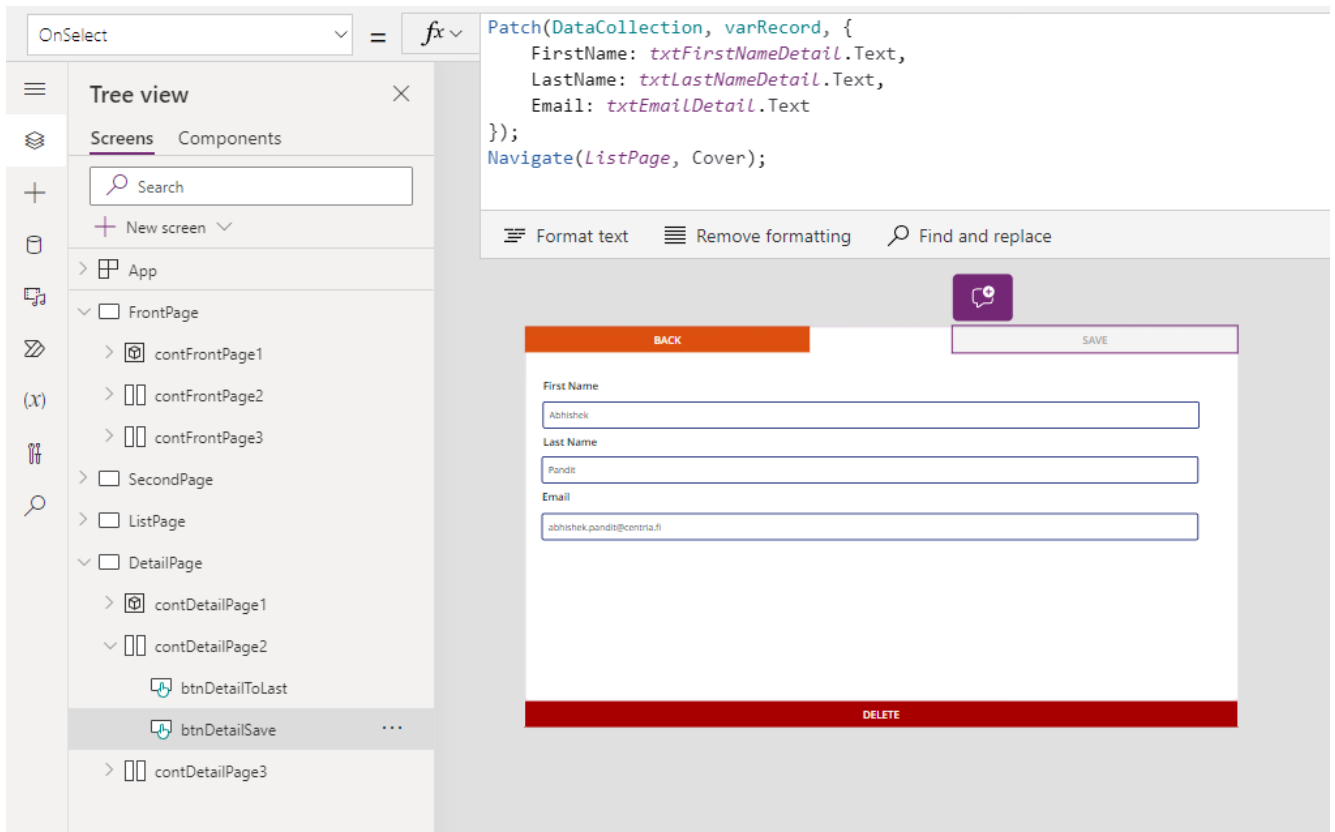


Figure 17 Patch function with its three arguments to save the modified data

4.3 Results

In the end result of the application, it was well suited for its purpose and the application was designed to specific requirements. The application worked in accordance with its intended purpose and works smoothly for its purpose as a simple tool for managing contacts. The design of the application was made interactive and user-friendly with different buttons, texts, input fields, and gallery. With all the functionality of performing create, read, update, and delete operation, the successful design of the application made it possible to manage and store multiple contact information in Microsoft Excel as shown in figure 19. The application was both responsive and interactive at the same time. Although the actual development work was straightforward and easy based on experience. It should be noted that experience is heavily influenced by prior knowledge and expertise in software development. For example, a citizen developer who does not necessarily have experience coding applications may still be able to meet up on the development but will require time and more familiarization. Despite this, the application's smooth development and execution shows how well PowerApps enable users to develop useful and interactive applications.

A	B	C	D
FirstName	LastName	Email	Power
Abhishek	Pandit	abhishek.pandit@centria.fi	5GIHtyLlnaM
Hello	World	Hello.world@outlook.com	em-1EivX_4g

Figure 18 Data is successfully stored in the Excel

5 CONCLUSION

The theory part of this thesis was initially driven by two goals. The initial objective was to understand the theory of a PowerApps application. Another goal was to put into practice the knowledge and develop a basic prototyping responsive application that can adjust its contents with the screen size and with basic crud operation for the data. With these focus in mind, the thesis began with theoretical subjects about different types of PowerApps application, various features, with the support of in-depth research from several articles.

Furthermore, PowerApps as a platform is a great solution, especially for companies majorly focused on the Microsoft ecosystem. Establishing easy connections to get information, for example, an Excel list, increases the ability of applications produced with Microsoft's own solution to offer versatile and agile solutions for different needs and requirements. The low-code implementation also enables product scalability and additional features without major challenges, and thus provides a good starting point for the further development of the project.

The thesis also highlights how important it is to choose the best platform, become familiar with it, make use of visual tools, encourage reusability, and take advantage of all features in order to accelerate application development and boost the output. Businesses may fully utilize LCDPs like PowerApps and enhance creativity in the app development lifecycle. This thesis essentially restates the main point, which is that companies may be enabled to innovate, adapt, and grow in the ever-changing digital ecosystem by adopting low-code and no-code development platforms like PowerApps. Business may gain more speed, efficiency, and become more creative by utilizing LCDPs' fundamental potential, which will ultimately shape the direction of digital transformation.

6 REFERENCES

- Böck, A. and Frank, U. 2021. Low-code platform. *Business & Information Systems Engineering*, 63(6), 733-740. Available at: <https://doi.org/10.1007/s12599-021-00726-8> Accessed 10th April 2024.
- Chrismanto, A. R., Santoso, H. B., Wibowo, A., Delima, R., & Kristiawan, R. A. 2019. Developing agriculture land mapping using rapid application development (rad): a case study from indonesia. *International Journal of Advanced Computer Science and Applications*, 10(10). Available at: <https://doi.org/10.14569/ijacsa.2019.0101033> Accessed 19th January 2024.
- Educative 2024. What is DevOps? A Complete Guide to DevOps Methodology, (Online) Available at: <https://www.educative.io/answers/what-exactly-is-devops>. Accessed 19th January 2024.
- Juuti, J. 2021. Powerappsilla toteutettu lainausjärjestelmä. <https://urn.fi/URN:NBN:fi:amk-2021052611310> Accessed 10th April 2024.
- Lehto, P. 2021. Power Apps:n käytön aloittaminen. Available at: <https://urn.fi/URN:NBN:fi:amk-2021082017071>. Accessed 19th January 2024.
- Microsoft. 2022. PowerApps and Features of PowerApps! TechCommunity. (Online) Available at: <https://techcommunity.microsoft.com/t5/nta-techies/powerapps-and-features-of-powerapps/ba-p/3672520> Accessed 19th January 2024.
- Microsoft. 2023. What are model-driven apps in PowerApps? Available at: <https://learn.microsoft.com/en-us/power-apps/maker/model-driven-apps/model-driven-app-overview> Accessed 19th January 2024.
- Palmer, T., 2020. Microsoft PowerApps as an Alternative Solution to Business Application Development. Available at: <https://urn.fi/URN:NBN:fi:amk-2020120325932>. Accessed 19th January 2024.
- Petersen, K., Wohlin, C., & Baca, D. 2009. The waterfall model in large-scale development. *Lecture Notes in Business Information Processing*, 386-400. Available at: https://doi.org/10.1007/978-3-642-02152-7_29 Accessed 24th January 2024.
- Rokis, K. and Kirikova, M. 2023. Exploring low-code development: a comprehensive literature review. *Complex Systems Informatics and Modeling Quarterly*, (36), 68-86. Available at: <https://doi.org/10.7250/csimq.2023-36.04> Accessed 19th January 2024.
- ScienceSoft, 2024. Microsoft Power Apps in Brief, (Online) Available at: <https://www.scensoft.com/microsoft-power-apps>. Accessed 19th January 2024.
- Simplilearn, 2023. Agile Development Methodologies, (Online) Available at: <https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-agile>. Accessed 19th January 2024.
- Singh, A. 2023. 5 key features of Microsoft Power Apps that can help your business grow. LinkedIn. (Online) <https://www.linkedin.com/pulse/5-key-features-microsoft-power-apps-can-help-your> Accessed 19th January 2024.

Tollefson, K. 2022. Power Platform: Model-Driven vs. Canvas Apps vs Portal – What To Use When. (Online) Available at: <https://www.withum.com/resources/power-platform-model-driven-vs-canvas-apps-vs-portal-what-to-use-when/> Accessed 19th January 2024.