

Lab 0: Validate lab environment

Scenario

Bellows College is an educational organization with multiple buildings on campus. Campus visitors are currently recorded in paper journals. The information is not captured consistently, and there are no means to collect and analyze data about the visits across the entire campus.

Campus administration would like to modernize their visitor registration system where access to the buildings is controlled by security personnel and all visits are required to be pre-registered and recorded by their hosts.

Throughout this course, you will build applications and perform automation to enable the Bellows College administration and security personnel to manage and control access to the buildings on campus.

In this Module 0 lab, you will acquire a Power Platform trial and access the Power Platform admin center. In the admin center, you will then create a **Practice** environment that you will perform the majority of your lab work in.

Exercise 1 – Setup

Task #1 - Acquire your Microsoft Power Platform trial tenant

1. Copy your **Microsoft 365 credentials** from the Authorized Lab Hoster.
2. Navigate to <https://powerapps.microsoft.com> and click **Start free**.
3. Under **Let's get you started**, enter the email address from your Microsoft 365 credentials in the text box that says **Enter your work email address** and click **Next**.
4. If you see a prompt that you have an existing account with Microsoft. Select **Sign in**.
5. Enter the password provided by the Authorized Lab Hoster and sign in.
6. Select **Yes** to stay signed in.

7. Do not change country.
8. For **Phone number**, enter 0123456789.
9. Complete your account info and select **Get started** to sign up for your Microsoft Power Platform trial.
10. At the confirmation screen, click **Get Started**.
11. If you see a prompt to enter contact details, click on **X** to close the popup window.

Task #2 – Create environment

1. Navigate to <https://admin.powerplatform.microsoft.com> and log in with your Microsoft 365 credentials if prompted.
2. If you see a Welcome popup, click **Get Started**.
3. Select **Environments** and click + **New**.
 - i. For **Name**, enter **[My Initials] Practice** (Example: AJ Practice).
 - ii. For **Type**, select **Trial** (do not select the Trial (subscription-based) option).
 - iii. Change the toggle on **Create a database for this environment?** to **Yes**.
 - iv. Leave all other selections as default and click **Next**.
 - v. On the next tab, leave all selections as default and click **Save**.
4. Your **Practice** environment should now show in the list of Environments.

Your environment may take a few minutes to provision. Refresh the page if needed.

Lab 1: Data Modeling

Scenario

Bellows College is an educational organization with multiple buildings on campus. Campus visits are currently recorded in paper journals. The information is not captured consistently, and there are no means to collect and analyze data about the visits across the entire campus.

Campus administration would like to modernize their visitor registration system where access to the buildings is controlled by security personnel and all visits are required to be pre-registered and recorded by their hosts.

Throughout this course, you will build applications and perform automation to enable the Bellows College administration and security personnel to manage and control access to the buildings on campus.

In this lab you will access your environment, create a Microsoft Dataverse database, and create a solution to track your changes. You will also create a data model to support the following requirements:

- R1 – Track information for scheduled campus visits
- R2 – Record basic information to identify and track the visitors
- R3 – Schedule, record, and manage visits

Finally, you will import sample data into Microsoft Dataverse.

High-level lab steps

To prepare your learning environments you will:

- Refer to the [data model document](#) for the metadata description (tables and relationships). You can hold CTRL+click or right click the link to open the data model document in a new window.
- create Visit table
- import Visit data using an Excel spreadsheet

Prerequisites

- Completion of **Module 0 Lab 0 - Validate lab environment**

Things to consider before you begin

- Naming conventions - type names carefully.

Exercise 1: Create New Table

Objective: In this exercise, you will create new custom table for Visits.

Task #1: Create Visit Table and Columns

The **Visit** table will contain information about the campus visits including the visitor, scheduled times and actual times of each visit.

We would like to assign each visit a unique number that can be easily entered and interpreted by a visitor when asked during the visit check-in process.

We use **Time zone independent** behavior to record date and time information, because time of a visit is always local to the location of the building and should not change when viewed from a different time zone.

1. Sign into <https://make.powerapps.com> (if you are not already signed in)
2. Select your **[my initials] Practice** environment at the top right if it is not already selected.
3. Using the navigation on the left, expand **Dataverse**, and select **Tables**.
4. Click + **New table**.
5. Enter **Visit** for **Display Name**.
6. Click **Save**.
7. Under the **Schema** section, select **Columns**.
8. Create Scheduled Start column
 - Select + **New column**.
 - Enter **Scheduled Start** for **Display name**.
 - Select **Date and Time** for **Data type**.
 - In **Required**, select **Business required**.
 - Expand **Advanced options**.
 - In **Time zone adjustment**, select **Time zone independent**.

- Click **Save**.

9. Create Scheduled End column

- Click + **New column**.
- Enter **Scheduled End** for **Display name**.
- Select **Date and Time** for **Data type**.
- In **Required**, select **Business required**.
- Expand **Advanced options**.
- In **Time zone adjustment**, select **Time zone independent**.
- Click **Save**.

10. Create Actual Start column

- Click + **New column**.
- Enter **Actual Start** for **Display name**.
- Select **Date and Time** for **Data type**.
- In **Required**, leave this as **Optional**.
- Expand **Advanced options**.
- In **Time zone adjustment**, select **Time zone independent**.
- Click **Save**.

11. Create Actual End column

- Click + **New column**.
- Enter **Actual End** for **Display name**.
- Select **Date and Time** for **Data type**.
- In **Required**, leave this as **Optional**.
- Expand **Advanced options**.
- In **Time zone adjustment**, select **Time zone independent**.
- Click **Save**.

12. Create Code column

- Click + **New column**.
- Enter **Code** for **Display name**.
- Select **Autonumber** for **Data type**.
- Select **Date prefixed number** for **Autonumber type**.
- Click **Save**.

13. Create Visitor lookup column

- Click + **New column**.
- Enter **Visitor** for **Display name**.
- Select **Lookup** for **Data type**.
- Select **Contact** for the **Related Table**.
- Expand **Advanced options**.
- Enter **visitor_id** for **Relationship name**.
- Click **Save**.

Exercise 2: Import Data

Objective: In this exercise you will import sample data into the Dataverse database.

Task #1.1: Load Excel file to OneDrive

1. You should have the **Visits.xlsx** file stored on your virtual machine in **C:/LabFiles**. Download [Visits.xlsx](#) if you do not.
2. If not already signed in, sign in to <https://make.powerapps.com>.
3. Select your **[my initials] Practice** environment at the top right if it is not already selected.
4. Click on the Waffle button in the upper left corner to change applications and select **OneDrive**. (It may take a moment for your OneDrive to be set up. Click Your OneDrive is ready when you see it on the screen.)
5. Click **Upload** from the top menu and select **Files**.

6. Locate and select the **Visits.xlsx** file and click **Open**.

Note: This file is located in the **All Files** folder on your machine.

Task #1.2: Create a dataflow

1. If not already signed in, sign in to <https://make.powerapps.com>.
2. Select your **[my initials] Practice** environment at the top right if it is not already selected.
3. Using the navigation on the left, expand **Dataverse**, and select **Tables**.
4. Locate and open the **Visit** table you created in the previous exercise.
5. Using the menu at the top, select the drop-down arrow next to **Import**, select the **Import data**.
6. In the **Choose data source** dialog, select **Excel workbook**.
7. Select **Link to File** option. Click **Browse OneDrive**. If prompted, sign in with your Microsoft 365 credentials.
8. Select the **Visits.xlsx** file which has been uploaded in OneDrive and click **Select**.
9. Click **Next**.
10. Under **Choose Data** check the box next to the **Visits** Excel workbook.
11. Click **Next**. Do not navigate away from this page.
12. Click **Next**.
13. On the **Map tables** section, select **Load to existing table** under the **Load settings**.
14. On the **Destination table** drop-down menu, Select the table name starts with **crXXX_visit** (where XXX is a random set of letters and numbers)
15. On the **Column Mapping**. Map the Columns to their corresponding destination columns.

Destination columns	Source values
----------------------------	----------------------

crxxx_ActualEnd	actual end
-----------------	------------

crxxx_ActualStart	actual start
-------------------	--------------

Destination columns	Source values
crxxx_Code	code
crxxx_Name	name
crxxx_ScheduledEnd	scheduled end
crxxx_ScheduledStart	scheduled start

16. Click **Next**.

17. Select **Refresh manually**.

18. Click **Publish**.

Note: It can take several minutes for your data to import into your table. Don't worry if you get a few errors, that is normal, and will not impact the rest of the course.

Task #3: Verify Data Import

1. After your data has been imported, use the navigation at the left of the screen to select the **Visit** table again.
2. Verify that you see the imported data under the **Visit columns and data** section.

Congratulations, you have successfully created a new table and imported data.

Lab 2: How to build a canvas app

Scenario

Bellows College is an educational organization with multiple buildings on campus. Campus visits are currently recorded in paper journals. The information is not captured consistently, and there are no means to collect and analyze data about the visits across the entire campus.

Currently, campus administration is leveraging an Excel spreadsheet to track visitor registration. They would like to modernize their visitor registration system where access to the buildings is controlled by security personnel and all visits are required to be pre-registered and recorded by their hosts.

Throughout this course, you will build applications and perform automation to enable the Bellows College administration and security personnel to manage and control access to the buildings on campus.

High-level lab steps

We will follow the below outline to design the canvas app:

- Create a canvas app from data in the Visit table
- Configure how the visits are shown on the browse screen
- Make some basic changes to the app
- Test the app functionality

Prerequisites

- Completion of **Module 0 Lab 0 - Validate lab environment**
- Completion of **Module 2 Lab 1 - Data Modeling**

Exercise 1: Create Visits Canvas app

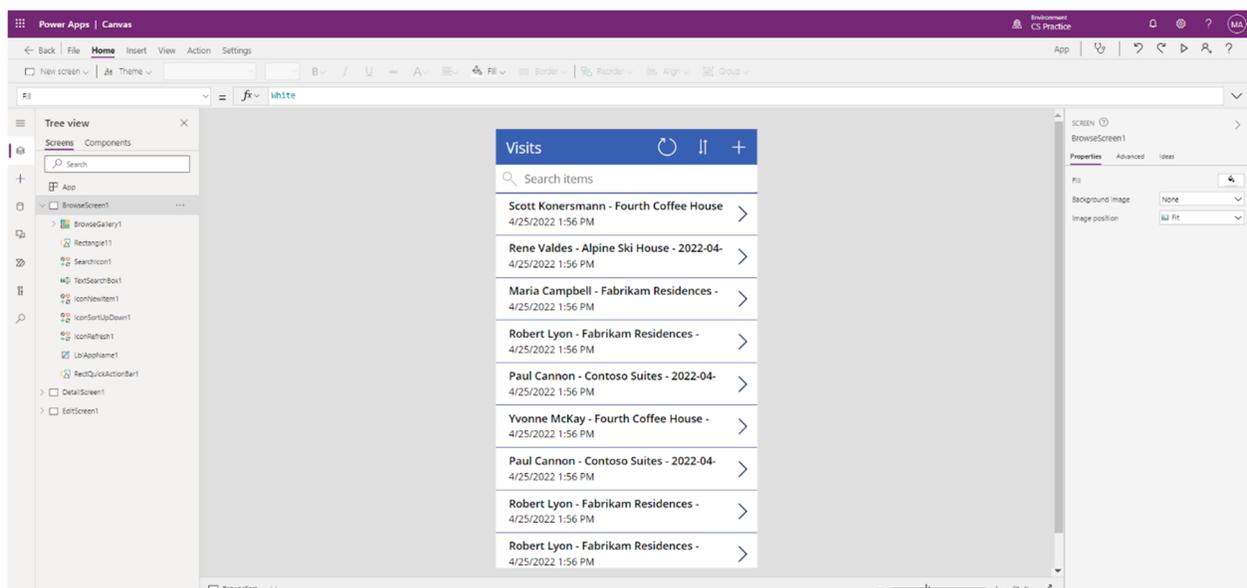
Objective: In this exercise, you will create a canvas app by connecting your Visits table you created earlier.

Task #1: Create the Visits app

1. Navigate to <https://make.powerapps.com>. You may need to reauthenticate - click **Sign in** and follow instructions if needed.
2. Select your **[my initials] Practice** environment at the top right if it is not already selected.
3. If necessary, click the **Home** icon on the left side of the screen. Under the **Start from** section, select **Dataverse**.
4. Select your Dataverse connection.

NOTE: *If a Dataverse connection does not exist:*

- Select **New Connection**
 - Locate **Microsoft Dataverse**
 - Click **Create**
5. Locate and select the **Visits** table you created in the previous lab.
 6. Select the **Connect** button in the bottom right corner.
 7. After your app is created, On the Welcome to Power Apps Studio screen, check the **Don't show me this again** box, and then select **Skip**.
 8. After creation has completed, it should look like the image below.



9. In the app designer, select the **preview the app** button (Play icon) on the command bar. *(You can also preview the app by pressing F5 on your keyboard.)* Take a look around and see how your app looks out-of-the-box.
10. Close the app preview by selecting the **X** in the upper right of the screen.

Congratulations, you have successfully created a Power App from a Dataverse table. The next step in the process is to tailor the app to match your college's branding. The next series of steps will walk you through providing some extra customization to the app.

Task #2: Modify and theme the newly created app

In this task, you will customize the header text on each of the three screens for your app (Browse, Detail, and Edit) and change the app theme.

1. You are on the Browse screen. Select the **Visits** label on the screen.
2. On the right side of the screen, under the Properties tab, update the **Text** control property to "**Bellows College Visits**".
3. In the properties, change the **Font size** to **24**.
4. Click in the blank background of the screen to see the updated text on your Browse screen.
5. Using the Tree view in the left-hand navigation, select **DetailScreen1**.
6. Select the **Visits** label on the screen.
7. On the right side of the screen, under the properties tab, update the **Text** control property to "**Visit Details**".
8. Click in the blank background of the screen to see the updated text on your Details screen.
9. Using the Tree view in the left-hand navigation, select **EditScreen1** (you may need to scroll down to see this on the Tree view).
10. Select the **Visits** label on the screen.
11. On the right side of the screen, under the properties tab, replace the text Table1 in the **Text** control property with "**Edit Details**".

12. Click in the blank background of the screen to see the updated text on your Edit screen.
13. Using the Tree view in the left-hand navigation, select the **BrowseScreen1**.
14. On the command toolbar, select the **Theme** button and from the list that appears select the **Red** theme color.

Task #3: Test your Visits app

In this task, you will test out your new app.

1. With your application open in the App Designer, select **Settings**, in the **General** section update the name of your app to **Visits App**, click the **X** to close the settings screen and then select **Save**.
2. Select the **back** arrow to return to your app.
3. Using the navigation on the left, select **BrowseScreen1**.
4. In the app designer, select the **preview the app** button (Play icon) on the command bar. *(You can also preview the app by pressing F5 on your keyboard.)*
5. Once the app opens, in the **Search Items** field, enter the text **Maria** *(Notice how the items in the gallery filter based on what is typed in the search field).*
6. Once the **Contoso Suites** record for **Maria Campbell** is displayed, click on the row to navigate to open the details for that visit. **(Note: If more than one Contoso Suites Maria Campbell record is displayed, select any of them)**
7. To edit the record, select the **Pencil Icon** in the upper right corner of the app.
8. You can edit the Visit Name here and click the Checkmark icon in the top right to save the change.
9. On the top right of the screen, click the **X** Icon to return to the canvas app editor.

Congratulations! You have created and configured your first canvas app.

Challenges

- Add the following columns to the forms in DetailScreen1 and EditScreen1: Actual Start, Actual End, Code, Scheduled Start, & Scheduled End

Lab 3: How to build a model-driven app

Scenario

Bellows College is an educational organization with multiple buildings on campus. Campus visitors are currently recorded in paper journals. The information is not captured consistently, and there are no means to collect and analyze data about the visits across the entire campus.

Campus administration would like to modernize their visitor registration system where access to the buildings is controlled by security personnel and all visits are required to be pre-registered and recorded by their hosts.

Throughout this course, you will build applications and perform automation to enable the Bellows College administration and security personnel to manage and control access to the buildings on campus.

In this lab, you will build a Power Apps model-driven app to allow the backoffice campus staff to manage visit records across the entire campus.

High-level lab steps

As part of creating the model-driven app, you will complete the following:

- Create a new model-driven app named Bellows Campus Management
- Edit the app navigation to reference the required tables
- Customize the forms and views of the required tables for the app

We will work with the following components:

- **Views:** Views allow the user to display the existing data in the form table.
- **Forms:** This is where the user creates/updates new rows in the tables.

Both will be integrated to the model-driven app for a better user-experience.

Prerequisites

- Completion of **Module 0 Lab 0 - Validate lab environment**
- Completion of **Module 2 Lab 1 - Data Modeling**

Things to consider before you begin

- What changes should we make to improve the user experience?
- What should we include in a model-driven app based on the data model we have built?
- What customizations can be made on the sitemap of a model-driven app?

Exercise 1: Customize Views and Forms

Objective: In this exercise, you will customize views and forms of the custom created tables that will be used in the model-driven app.

Task #1: Edit Visit Form

1. Sign in to <https://make.powerapps.com> if you are not already signed in.
2. Select your **[my initials] Practice** environment at the top right if it is not already selected.
3. Using the navigation on the left, expand **Dataverse**, select **Tables**, and click to open your **Visit** table.

If you do not see the Visit table, make sure you are in the correct environment (step 2).

4. Under the **Data experiences** section, select **Forms** and click to open the Information form with **Main** form type.

IMPORTANT: Since by default all forms are named Information, make sure to verify that the form you select has a Form Type of **Main** and not something else. By default, the form has two fields: Name and Owner.

5. On the right side of the screen on the Properties panel, select the **Display Name** field, and change it to **Main Information**.
6. Select the **Table Columns** from the left navigation pane and add the following fields below the **Owner** field by dragging columns to the form or simply clicking column names:

- i. **Visitor**
- ii. **Scheduled Start**
- iii. **Scheduled End**
- iv. **Actual Start**
- v. **Actual End**

7. Drag the **Code** column and drop it in the form header.

The header is the top right area of the form. You may need to collapse the Properties panel on the right side of the screen to see the field on the form.

8. With the **Code** field still selected, check the checkbox for **Read-only** in the Properties panel on the right side of the screen.
9. Select **Owner** field. In the Properties panel, change the **Label** to **Host**
10. Click **Save** at the top right and wait for the save to complete.
11. Click **Publish** at the top right and wait for the publishing to complete.
12. If the edit view opened in a new browser tab or window, close it. Otherwise, click **Back** at the top left of the screen. You should now be back to the Visit tables Forms.
13. Using the breadcrumbs in the upper left (Tables>Visit>Forms). Select **Visit** to return to the **Visit** table main screen.

Task #2: Edit Active Visits view

In this task, we will modify the default Active Visits view and create a new view for today's visits.

1. Under the **Data experiences** section, select **Views** and click to open your **Active Visits** view.
2. Add the following fields to the view by either clicking or dragging and dropping the fields:
 - i. **Code**

- ii. **Visitor**
 - iii. **Scheduled Start**
 - iv. **Scheduled End**
3. Click the **Created On** column and select **Remove**. Field **Created On** will now be removed from the view.
 4. Resize the individual column widths to fit the data.
 5. Click **Save** and wait until the changes are saved.
 6. Click **Publish** and wait for the publishing to complete.

Task #3: Create new view for today's visits

Now, we will clone the view to create a new view for today's visits.

IMPORTANT: Make sure that you do not close the Active Visits view, as we will be leveraging it to create the new today's visits view.

1. Click on the **dropdown arrow** by the Save button (be careful not to press the button itself) and select **Save As**.
2. Change the name to **Today's Visits** and press **Save**.
3. Click **Edit filters** link in the Properties panel.
4. Click **Add**, select **Add row**.
5. Select **Scheduled Start** as a field, then select **Today** as the condition in the dropdown.
6. Click the ... on the **Status** row and click **Delete** to delete that filter condition.
7. Press **Ok** to save the condition. The view is now filtered to show only records where the Scheduled Start date is today.
8. Add **Actual Start** and **Actual End** fields to the view.

Note: Since we no longer filter on the view status, we will get all today's visits including completed ones. These fields will help to differentiate completed visits and visits in progress.

9. Click **Save**.
10. Click **Publish** and wait for the publishing to complete.

Exercise 2: Create Model-driven app

Objective: In this exercise, you will create a model-driven app, customize the sitemap, and test the app.

For simplicity and time's sake, we will not be addressing some of the Visit columns in this lab.

Task #1: Create app

1. Sign into <https://make.powerapps.com> (if you are not already signed in).
2. Select your **[my initials] Practice** environment at the top right if it is not already selected.
3. If necessary, click the **Home** icon on the left side of the screen.
4. Create the Model-Driven Application:
 - i. Select **Blank app** in the **Start from** section of the Home screen.
 - ii. Under **Blank app based on Dataverse**, select **Create**.
 - iii. Enter **Bellows Campus Management** for Name and select **Create**.
5. After your new model-driven application loads, select the **+ Add Page** button.
6. On the Add Page screen, choose **Table based view and form**, and then select the **Next** button.
7. Add the following tables:
 - i. Visit
 - ii. Contact
8. After you have selected the 2 tables, select **Add**.
9. Using the navigation icons on the left side of the screen, select **Navigation**.

10. In the Navigation Pane, select **Group 1** below where it says Navigation bar. You may need to expand the menu on the left.
11. On the right side of the screen, in the **Display Options** section, change the **Title** property to **Security**.

Task #2: Edit your app

Now that we have all the necessary components added to your model-driven application, we will now organize items.

1. In the Navigation Pane, under the security group, select **SubArea1**.
2. Select the **Ellipsis**, and from the menu that appears, select **Remove SubArea1**.
3. Using the navigation on the left side of the screen, select **Pages**.
4. Locate and expand **Visit** on the Pages pane.
5. Select **Visit form**.
6. On the right side of the screen, select **Add form**.
7. Select the **Main Information** form.
8. Under **Visit** on the Pages pane, select **Visit view**.
9. On the right side of the screen, select **Add view**.
10. Select the **Today's Visits** view.
11. Select **Add view** again.
12. Select the **Active Visits** view.
13. Select **Save**.
14. Once the **Save** is complete, select the **Publish** button to publish your changes.

Task #3: Test Application

1. Start the application
 - i. Select **Play** to open your app in a new window.

2. Create new Contact

- i. The app should open to the **My Active Contacts** view. If it does not, select Contacts on the left-hand navigation.
- ii. Click **New** from the top menu.
- iii. Provide **First Name** as John and **Last Name** as Doe.
- iv. Provide your personal email as **Email**. This will be used in a future lab where you will receive an email.
- v. Click **Save & Close**.
- vi. You should now see the created contact on the **My Active Contacts** view.

3. Create new Visit

- i. Select **Visits** from the left-hand navigation of the sitemap.
 - ii. Click **+ New**.
 - iii. Enter the fields as following
 - a. **Name**: New test visit
 - b. **Visitor**: select John Doe
 - c. **Scheduled Start**: select tomorrow's date and 2:00 PM as start time
 - d. **Scheduled End**: select tomorrow's date and 3:30 PM as end time
 - Click **Save & Close**. This will create the Visit and you should be able to see it on the Active Visits View.
 - Change view to **Today's Visits** by using the drop down next to **Active Visits**. You should no longer see the new visit in the view, since it is scheduled for tomorrow.
4. You may add more test records.

Your running app should look approximately like the following:

Name	Code	Visitor	Scheduled Start	Scheduled End
Jim Glynn - Alpine Ski House - 2022-05-02	2022-15-04-1156		1/1/1753 12:00 AM	5/2/2022 3:00 PM
Jim Glynn - Contoso Suites - 2022-04-06	2022-15-04-1110		1/1/1753 12:00 AM	4/6/2022 10:30 AM
Jim Glynn - Fabrikam Residences - 2022-04-29	2022-15-04-1196		1/1/1753 12:00 AM	4/29/2022 4:00 PM
Jim Glynn - Fabrikam Residences - 2022-05-03	2022-15-04-1162		1/1/1753 12:00 AM	5/3/2022 2:30 PM
Jim Glynn - Fourth Coffee House - 2022-05-03	2022-15-04-1193		1/1/1753 12:00 AM	5/3/2022 6:00 PM
Jim Glynn - Northwind Traders Tower - 2022-04-08	2022-15-04-1091		1/1/1753 12:00 AM	4/8/2022 7:00 PM
Jim Glynn - Northwind Traders Tower - 2022-05-09	2022-15-04-1171		1/1/1753 12:00 AM	5/9/2022 9:00 AM
Maria Campbell - Alpine Ski House - 2022-04-18	2022-15-04-1211		1/1/1753 12:00 AM	4/18/2022 5:00 PM
Maria Campbell - Alpine Ski House - 2022-04-22	2022-15-04-1190		1/1/1753 12:00 AM	4/22/2022 4:30 PM
Maria Campbell - Contoso Suites - 2022-04-06	2022-15-04-1089		1/1/1753 12:00 AM	4/6/2022 4:30 PM
Maria Campbell - Contoso Suites - 2022-04-08	2022-15-04-1116		1/1/1753 12:00 AM	4/8/2022 1:00 PM
Maria Campbell - Contoso Suites - 2022-04-26	2022-15-04-1185		1/1/1753 12:00 AM	4/26/2022 10:30 AM
Maria Campbell - Contoso Suites - 2022-04-29	2022-15-04-1167		1/1/1753 12:00 AM	4/29/2022 6:00 PM
Maria Campbell - Fabrikam Residences - 2022-04-15	2022-15-04-1013		4/15/2022 7:13 PM	4/15/2022 8:45 PM
Maria Campbell - Fabrikam Residences - 2022-04-18	2022-15-04-1103		1/1/1753 12:00 AM	4/18/2022 9:00 AM
Maria Campbell - Northwind Traders Tower - 2022-04-13	2022-15-04-1145		1/1/1753 12:00 AM	4/13/2022 12:00 PM
Maria Campbell - Northwind Traders Tower - 2022-04-18	2022-15-04-1180		1/1/1753 12:00 AM	4/18/2022 3:00 PM
Maria Campbell - Northwind Traders Tower - 2022-05-06	2022-15-04-1159		1/1/1753 12:00 AM	5/6/2022 11:00 AM
Maria Campbell - Alpine Ski House - 2022-04-18	2022-15-04-1151		1/1/1753 12:00 AM	4/18/2022 4:00 PM

Congratulations! You have created and configured your first model-driven app.

Challenges

- Select specific views and forms for Contacts

Lab 4: How to build an automated solution

Scenario

Bellows College is an educational organization with multiple buildings on campus. Campus visitors are currently recorded in paper journals. The information is not captured consistently, and there are no means to collect and analyze data about the visits across the entire campus.

Campus administration would like to modernize their visitor registration system where access to the buildings is controlled by security personnel and all visits are required to be pre-registered and recorded by their hosts.

Throughout this course, you will build applications and perform automation to enable the Bellows College administration and security personnel to manage and control access to the buildings on campus.

In this lab, you will create a Power Automate flow to email a visitor when a visit is scheduled.

High-level lab steps

The following have been identified as requirements you must implement to complete the project:

- Contacts need to be notified via email when a visit is scheduled.

Prerequisites

- Completion of **Module 0 Lab 0 - Validate lab environment**
- Completion of **Module 2 Lab 1 - Data Modeling**
- Completion of **Module 2 Lab 3 - How to build a model-driven app**
- John Doe contact created with a personal email address populated

Exercise 1: Create Visit Notification flow

Objective: In this exercise, you will create a Power Automate flow that implements the requirement. The visitor should be sent an email that includes the unique code assigned to the visit when a visit is created.

Task #1: Create a flow

1. Navigate to <https://make.powerapps.com>. You may need to reauthenticate - click **Sign in** and follow instructions if needed.
2. Select your **[my initials] Practice** environment at the top right if it is not already selected.
3. In the left navigation, select **Flows**.
4. If prompted, select **Get started**.
5. Click **New flow** and select **Automated cloud flow**.
6. Enter "Visit Notification" for **Flow name**.
7. In **Choose your flow's trigger**, search for **Dataverse**.

8. Select the trigger **When a row is added, modified or deleted**, and then click **Create**.
9. Populate the trigger conditions for the flow:
 - i. Select **Added** for **Change type**
 - ii. Select **Visits** for **Table name**
 - iii. Select **Organization** for **Scope**
 - iv. On the trigger step, click the ellipsis (...) and click **Rename**. Rename this trigger "**When a visit is added**". This is a good practice, so you and other flow editors can understand the purpose of the step without having to dive into the details.

Task #2: Create a step to get the visitor row

1. Select + **New step**. This step is required to retrieve visitors information, including email address.
2. Search for **Dataverse**.
3. Select the **Get a row by ID** action.
4. Select **Contacts** as **Table name**
5. Select the **Row ID** field. Notice that a window pops up to select Dynamic content or Expressions.
6. In the **Row ID** field, select **Visitor (Value)** from the Dynamic content list. In this step, you are looking up the Contact for the Visit row that was created to trigger this flow. Since email address is part of the Contact table, you will need this information to send the email to the visitor.
7. On this action, click the ellipsis (...) and click **Rename**. Rename this action "**Get the Visitor**". This is a good practice, so you and other flow editors can understand the purpose of the step without having to dive into the details.

Task #3: Create a step to send an email to the visitor

1. Click + **New step**. This is the step that will send an email to the visitor.

2. Search for *mail*, select **Office 365 Outlook** connector and **Send an email (V2)** action.
3. If asked to Accept terms and conditions for using this action, click **Accept**.
4. Select **Add dynamic content** under the **To** field.
5. Select **Email** from the Dynamic content list. > Notice that it is beneath the **Get the visitor** header. This means you are selecting the Email that is related to the Visitor that you looked up in the previous step.
6. Enter **Your scheduled visit to Bellows College** in the **Subject** field.
7. Enter the following text in **Email Body**:

Dynamic content needs to be placed where fields are named in brackets. It is recommended to copy & paste all text first and then add dynamic content in the correct places.

Dear {First Name},

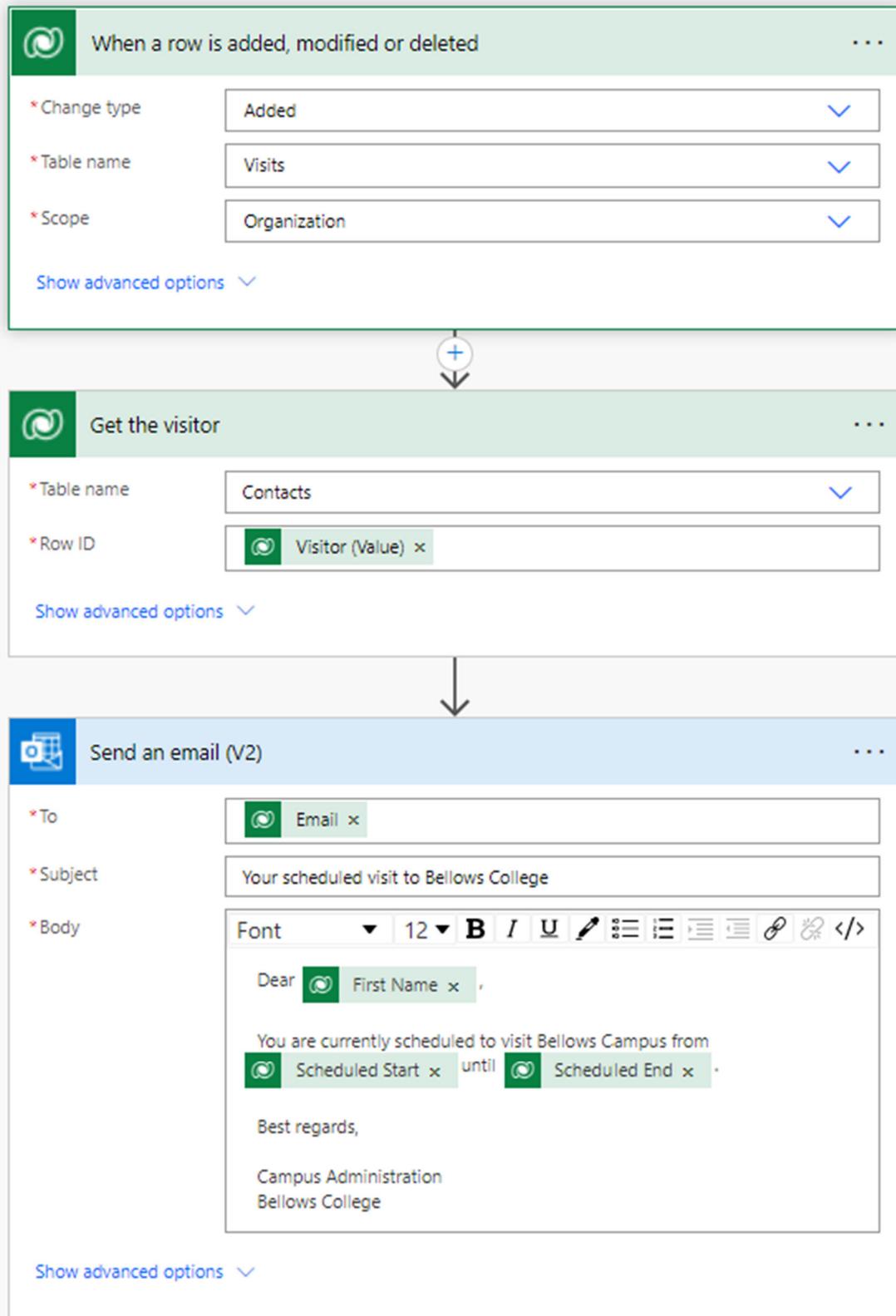
You are currently scheduled to visit Bellows Campus from {Scheduled Start} until {Scheduled End}.

Best regards,

Campus Administration
Bellows College

8. Highlight the **{First Name}** text. Replace it with the **First Name** field from the **Get the Visitor** step.
9. Highlight the **{Scheduled Start}** text. Replace it with the **Scheduled Start** field **When a visit is added** step.
10. Highlight the **{Scheduled End}** text. Replace it with the **Scheduled End** field from the **When a visit is added** step.
11. Click **Save**.

Leave this flow tab open for the next task. Your flow should look approximately like the following:



Task #4: Validate and test the flow

1. Open a new tab in your browser and navigate to <https://make.powerapps.com>.
2. Select your **[my initials] Practice** environment at the top right if it is not already selected.
3. Click **Apps** and select the **Bellows Campus Management** model-driven app you created earlier.
4. Leaving this browser tab open, navigate back to the previous tab with your flow.
5. On the command bar, click **Test**. Select **Manually** and then click **Test**.
6. Navigate to the browser tab with your model-driven app open.
7. Using the navigation on the left, select **Visits**
8. Press the + **New** button to add a new **Visit** record.
9. Complete the Visit record as follows:
 - o **Name:** Test Visit
 - o **Visitor:** John Doe
 - o **Scheduled Start:** Tomorrow at 8:00 AM
 - o **Scheduled End:** Tomorrow at 9:00 AM
10. Select the **Save and Close** button.
11. Navigate to the browser tab with your flow test running. After a short delay, you should see the flow running. This is where you can catch any issues in the flow or confirm that it ran successfully.

After a short delay, you should see an email in your inbox, since you populated John Doe's email as your personal email. Note that it may go to your Junk Email folder.

Challenges

- Play around with the formatting on the email. How can you make it more professional looking?

Lab 5: How to build a simple dashboard

Scenario

Bellows College is an educational organization with multiple buildings on campus. Campus visitors are currently recorded in paper journals. The information is not captured consistently, and there are no means to collect and analyze data about the visits across the entire campus.

Campus administration would like to modernize their visitor registration system where access to the buildings is controlled by security personnel and all visits are required to be pre-registered and recorded by their hosts.

Throughout this course, you will build applications and perform automation to enable the Bellows College administration and security personnel to manage and control access to the buildings on campus.

In this lab, you will build a Power BI report and dashboard that visualizes data about campus visits.

High-level lab steps

We will follow the below steps to design and create a Power BI dashboard:

- Create a report with various visualizations of the campus visits information
- Utilize a user natural language query to build additional visualizations

Prerequisites

- Completion of **Module 0 Lab 0 - Validate lab environment**
- Completion of **Module 2 Lab 1 - Data Modeling**

Things to consider before you begin

- Who is the target audience of the report?
- How will the audience consume the report? Typical device? Location?
- Do you have sufficient data to visualize?
- What are the possible characteristics you can use to analyze data about the visits?

Exercise 1: Create Power BI Report

Objective: In this exercise, you will create a Power BI report based on data the Excel spreadsheet we leveraged in a previous exercise.

Task #1: Prepare Power BI service

1. Download [visits.pbix](#) and save on your computer.
2. Navigate to <https://app.powerbi.com/> and sign in if needed.
3. In the lower Left corner of the screen, select **Get Data**
4. Select the **Get** button under **Files**, in the **Create new content** section.
5. Select **Local File**.
6. Locate and select **visits.pbix** file you've downloaded earlier.
7. Once data load is complete, expand **My workspace** and select **visits** report (notice that the Type is set to **Report**).
8. Click **Edit**. If **Edit** menu item is not visible click **...** and then select **Edit**.

You have now setup Power BI service to use for your labs.

Task #2: Create Chart and Time Visualizations

1. Press the **Pie chart** icon in the **Visualizations** panel to insert a chart.
2. Press the drop-down arrow beside **bc_Visit** in the Fields pane. Drag the **Building** field and drop it into **Legend** box.
3. Drag the **Visit** field and drop it into **Values** box.
4. Resize the pie chart using corner handles so that all chart components are visible.
5. Click on the report outside of the pie chart to deselect it and select stacked column chart in **Visualizations** pane.
6. Press the drop-down arrow beside **bc_Visit** in the Fields pane. Drag the **Visit** field and drop it into **Y-axis** target box.

7. Drag **Start** field and drop it into **X-axis** target box.
8. In the Visualizations pane, click **x** next to **Year** and **Quarter** to leave only **Month** and **Day** totals for the Axis.
9. Resize the chart as desired using the corner handles.
10. Test the report interactivity:
 - i. Click various building slices on the pie chart and observe changes on the time report.
 - ii. Click on the column chart. Press the down arrow to turn on **Drill down** mode (or right click on the chart and select **Drill down**), then click a column to drill down to the next level (days).
 - iii. Drill up and down and select various bars on the time column chart to observe changes on the pie report.
11. Save work in progress by pressing **Save**.

Exercise 2: Create Power BI Dashboard

Task #1: Create Power BI Dashboard

1. You should have the report open from the previous task.
2. Select **Pin to a dashboard** on the menu. Depending on the layout you may need to press ... to show additional menu items.
3. Select **New dashboard** on **Pin to dashboard** prompt.
4. Enter **Campus Management** as a **Dashboard name**, press **Pin live**.
5. A pop-up will prompt you that the dashboard has been created. Select **Go to dashboard**.
6. Test interactivity of the pie and bar charts displayed.

Task #2: Add Visualizations Using Natural Language

1. Within your **Campus Management** dashboard, select **Ask a question about your data** bar at the top.
2. Enter **buildings by number of visits** in Q&A area. A bar chart will be displayed.
3. Select **Pin visual**.
4. Select **Existing dashboard**, select your **Campus Management** dashboard, press **Pin**.
5. Click **Exit Q&A**.

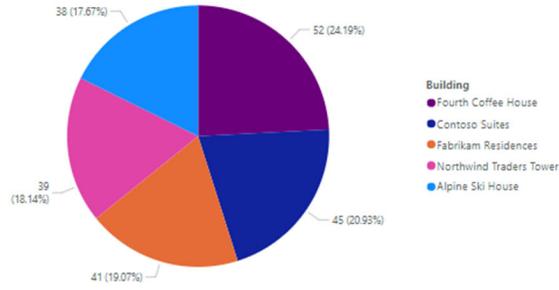
Your **Campus Management** dashboard should be displayed with three visuals on it. You may have to scroll down to see the new Q&A visual.

Your dashboard should look similar to the following:

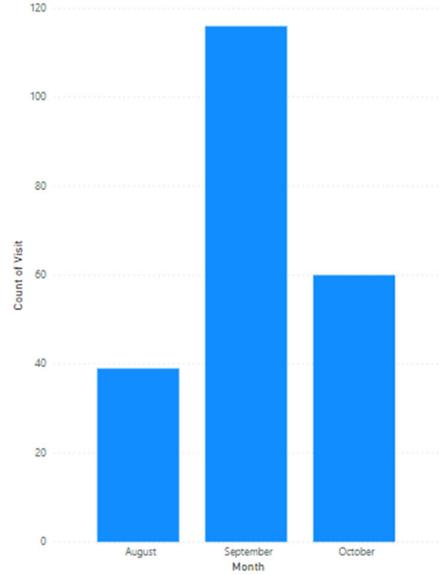
Ask a question about your data

Campus Visits PAGE 1

Count of Visit by Building



Count of Visit by Month



Count of Visit
BY BUILDING

