===

Exercise 1 – Filtering Data

In this task, you will filter the Survey to show only Active rows

that have Start Date in the past and End Date is in the future.

1. Navigate to https://make.powerapps.com.
2. Make sure you are in the correct environment.
3. Select **Solutions**.
4. Select **Wired Brain Coffee Solution**
5. Select the **HR Survey** Canvas application.
6. Click on the **Edit** button located on the command bar.
7. Wait for the app designer to load.
8. Select the **Survey List** gallery and click the **Advanced** tab in the Properties pane.
9. Set the **Items** Property to the snippet below. This snippet will filter the Survey rows and the list will show only the active rows

   with Start Date in the past and End Date in the future. *Note:* if you type

   this instead of pasting it, you will see how the editor helps you build

   expressions.

```
    Filter('Surveys', (Status = 'Status (Surveys)'.Active && 'Start
Date' <= Today() && 'End Date' >= Today()))
```

10. Click **File** and **Save** your changes.
11. Click on the **Back** button.
12. Do not close the app designer.

Task 2 – Get Current User

In this task, you will get the current User and save it in a global variable.

1. In the tree view, select the **App.**
2. Select the **OnStart** property and set it to snippet below. This snippet will create a global variable **UserPrimaryEmail** that will hold the

current user’s email.

```
    Set(UserPrimaryEmail, User().Email)
```

3. Add the snippet below to the **OnStart** property, after the snippet already there. This snippet will first terminate the first function with semicolon, get the current User and save it in a global variable name **CurrentUser**.

```
    ;Set(CurrentUser, LookUp(Users, 'Primary Email' =
UserPrimaryEmail))
```

4. Add the snippet below to the **OnStart** property, after the snippets already there. Add the following function to work around an existing bug that does not properly load the metadata for related properties. In the future this workaround will not be required.

```
    ;Set(FirstKABugWorkaround,First('Survey Test Results').'Survey')
```

5. Your **OnStart** property should now have the above three snippets. Click **File** and **Save**.
6. Select the **back** button to navigate back to the app designer.

Task 3 – Save Total Points

In this task, you will create a Patch that will create a Survey Test Result row. Using the Patch function allows us to on demand create a row with just specific properties being passed. In this case, we will be using the sum function to get a total of our points that are stored in our in-memory collection based on the answers the user provided.

1. Expand the **Take Survey Screen**.
2. Locate the button and rename it **Score Button**.
3. While you still have the **Score Button** selected, select the **Advanced**

tab of the Properties pane. Set the **OnSelect** property to snippet below. This snippet will create a new Survey Test Result row. **Note:** For long formulas, you can expand the **fx** bar to format your formula in a larger view.

```
    Patch('Survey Test Results',Defaults('Survey Test Results'),
{'Survey':'Survey List'.Selected,Name:'Survey List'.Selected.Title,
'Total Points': Sum(UserAnswers.Points,Points)});
```

Task 4 – Add Feedback Screen

In this task, you will add a new screen to the applications. This screen will let the user submit feedback. This tasks demonstrates how to use the EditForm to create a new row.

1. Click on the ellipses button of the **Take Survey Screen** and click **Duplicate Screen**. We are creating a duplicate screen because we want all our screens to look the same and we don’t want to recreate the header every time we add a new screen.
2. Rename the duplicate screen **Add Feedback Screen**.
3. Rename the Button on the **Add Feedback Screen** to **Submit Feedback Button**.
4. Change the **Text** property of the **Submit Feedback Button** to **"Submit Feedback"**.
5. Click on the ellipses button of the **Question List** inside the **Add Feedback Screen** and click **Delete**.
6. Select the **Add Feedback Screen**.
7. From the **Insert** tab, click **Forms** and then click **Edit.**
8. In the Property panel on the right, select **Feedback** for **Data Source**. (If you are already on the Advanced tab, you will need to switch to the Properties tab.)
9. Go to the tree view and rename the form **Feedback Form**.
10. Resize and reposition the **Feedback Form** to your liking.
11. From the tree view, expand the **Feedback Form**.
12. Select the **Title** column. In the **Advanced** pane, select **unlock** and

go to the **Data** section.

13.Set the **Default** property to the **Title** of the selected **Survey**.

```
'Survey List'.Selected.Title
```

14. With the **Title** data card still selected, select the **Properties** tab. Locate the **DisplayMode** property and set it to **View**.

```
DisplayMode.View
```

15. Rename the **Comments_DataCard** to **User Comments**. (You may need to click **Unlock** first.)

16. Expand the **User Comments** data card and rename the **DataCardValue** to **User Comments Text**.

17. Select the **Submit Feedback Button**.
18. Replace the **OnSelect** property value with the snippet below. This snippet will submit the form, reset the form, and navigate back to the previous

page.

```
SubmitForm('Feedback Form');ResetForm('Feedback
Form');Back(ScreenTransition.None)
```

19. With the **Submit Feedback Button** still selected, return to the **Properties** tab and select the **DisplayMode** property (click on the words **Display mode**).

20. Replace the **DisplayMode** property value with snippet below. This snippet will enable the button if the comments filed has value and disable it if the

comments field is blank.

```
If(IsBlank('User Comments Text'), DisplayMode.Disabled,
DisplayMode.Edit)
```

21. Select the **Take Survey Screen**.

22. Go to the **Insert** tab, navigate to **Icons** in the top menu, and select **Left.**

23. Resize and place the icon on the left side of the header.
24. Set the **Color** property of the icon to **White**.
25. Set the **OnSelect** property of the icon to the snippet below. This snippet will navigate back to the previous page.

```
    Back()
```

26. Select the **Take Survey Screen**.
27. Click **Insert**, navigate to **Icons** in the top menu, and select the **Emoji - Smile** icon.

28. Place the icon in the right side of the header and next to **username.**
29. Set the **Color** property of the icon to **White**.
30. Set the **OnSelect** property of the Emoji icon to the snippet below.

```
    NewForm('Feedback Form');Navigate('Add Feedback Screen',
ScreenTransition.None)
```

31. Click **File** and **Save** to save your application.

Exercise 2 – Customizing the User Experience

Task 1 – Show the Survey Result

In this task, you will show the survey result to the user. In this task you will use a local variable ShowResults to indicate if the results should show. It will be updated upon the user clicking the score button. Each item will then use an expression to highlight if the answer is right or wrong, only when ShowResults is true.

1. Navigate to https://make.powerapps.com.
2. Make sure you are in the correct environment.
3. Select **Solutions**.
4. Select **Wired Brain Coffee Solution**

5. Select the **HR Survey** Canvas application.
6. Click on the **Edit** button located on the command bar.
7. Wait for the app designer to load.
8. Select the **Take Survey Screen**
9. Select the **OnVisible** property of the **Take Survey Screen** and replace the value with the snippet below. This snippet will re-add the Clear

   function and add a new function that will initialize a variable

   ShowResults and set it to **false**.

```
Clear(UserAnswers);UpdateContext({ShowResults:false})
```

10. Select the **OnSelect** property of the **Score Button**.
11. Add the snippet below to the content you currently have. This snippet will add a function that will set the **ShowResults** value to **true**.

```
;UpdateContext({ShowResults:true})
```

12. Select **Answer3 Selected** checkbox of the **Survey Question List**.
13. Select the **Fill** property of **Answer3 Selected** and set it to the snippet below. This snippet will:

- Set the Fill Color to White if ShowResults is false.
- Set the Fill Color to Red id ShowResults is true, the checkbox is check, and the Points value is less than 0.

- Set the Fill Color to Green id ShowResults is true, the checkbox is check, and the Points value is more than 0.

```
If(ShowResults, If('Answer3 Selected'.Value = true &&
ThisItem.'Answer 3 Points' > 0, Green, If('Answer3 Selected'.Value =
false, White, Red)), White)
```

14. Repeat the previous setps for answer 2 and answer 1, adjusting the snippet as necessary.

Task 2 – Disable/Enable Button

In this task, you will disable the score button if there are no answers selected

and enable it when there is at least one answer selected.

1. Select the **Score Button**.
2. With the **Score Button** selected, set the **DisplayMode** property to the snippet below. This snippet will disable the button if there are no answers

   selected and enable it if there is at least one answer selected.

```
If(CountRows(UserAnswers) > 0, DisplayMode.Edit,
DisplayMode.Disabled)
```

Task 3 - Create the Results Screen

In this task, you will copy the Main Screen and rename it Results Screen.

1. Click on the ellipses button of Main Screen and select Duplicate Screen.
2. Click on the ellipses button of the new screen and click Rename.
3. Rename the new screen **Results Screen**.
4. Select the ** Survey List** under the **Results Screen**. Click on the ellipses button and click **Delete**.

5. Make sure the **Insert** tab is selected. Click **Gallery** and select **Horizontal**.

6. The **Data source** pop-out will come to view.
   Select ** Results** for data source.
7. Rename the gallery to **User Result List**.
8. Delete the image in **User Result List**. You will receive an error with a red X - Hover over the X and select **Edit in the formula bar**.

9. Delete the formula in the **fx** bar to remove the error.
10. Select the **User Results List** gallery.
11. In the **Advanced** tab, select the **Items** property and set it to the snippet below.

```
    Filter('Test Results',  Owner = CurrentUser)
```

11. Return to the **Properties** tab. Click **Edit** on the Fields property. For Title, using the dropdown, select the **createdon** value.

12. For the Subtitle element, using the dropdown, select crxxx_totalpoints.

(crxxx refers to your unique prefix.)

13. Select the **Results Screen** and navigate to the **Insert** tab. Add the **Left** icon to the upper left corner. Change the color to **white.**

14. Set the Left icon OnSelect to this snippet.

```
Back()
```

Task 4 – Add Button for Results Screen

In this task, you will add a button to the Main Screen. This button will

navigate to the Results page.

1. Select the **Main Screen**.
2. Go to the **Insert** tab and click **Icons**.
3. Select the **… More** Icon.
4. Place the icon in the left side of the header.
5. Change the **color** of the icon to **White**.
6. Select the **OnSelect** property of the icon and provide the function below. This function will run when the icon is clicked and navigate to the Results

   Screen.

```
Refresh(Users);Navigate('Results Screen', ScreenTransition.None)
```