Documentation for Budget Program

This program, c:\Users\Cheri\Excel\Budget\Budget.xlsm, provides up to date balances on checking and savings. It also provides information on each type of monthly expense incurred for each month in the year. These expenses are averaged and used in the WhatIf worksheet for decision making regarding proposed expenses.

It automates Excel leaving the user with user form prompts to navigate through the program.

The following documentation describes how to use the program and also how it functions.

1. **Choose Worksheet Form**

Before proceeding it is important to note that the *Choose Worksheet* form is the primary user interface to the program. It is automatically displayed when the workbook is opened.

The user has the option of exiting the Budget.xlsm workbook by selecting *Exit*. This will save the current iteration of the workbook to the D: drive (memory stick) and exit the workbook.

At the heart of the program is the use of Export in the USAA banking application. For our purposes the transactions are specified from January 1 to today’s date. Each month is separated by the program in the Dashboard worksheet where the current balance for checking and savings accounts are also displayed according to the dates data was downloaded.

The following documentation outlines the procedures for obtaining and maintaining these exports. This is the heart of the program and is vital for its operation and veracity.

* 1. **Importing Year to Date Checking Data**

All banking applications and the link to the *Downloads* file folder on Windows are available to the user from the *Choose Worksheet* user form shown in figure 1.2. Selecting *USAA, Upgrade, Download* takes the user to the hyperlinks user form figure 1.3

.

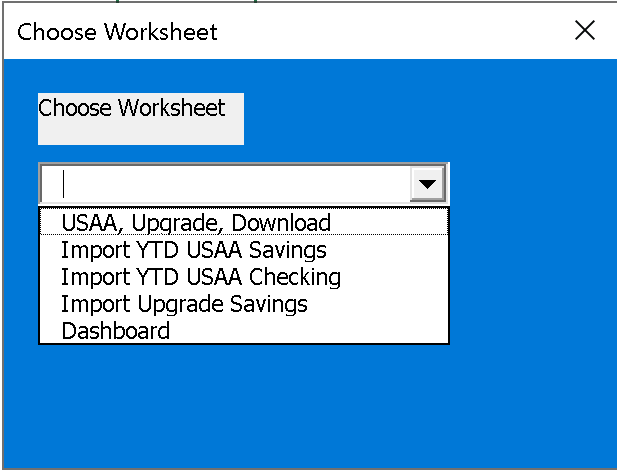


Figure 1.2

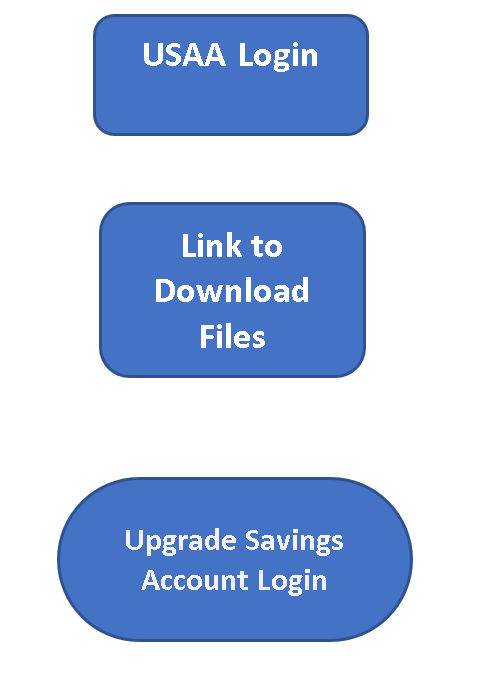


Figure 1.3

The user has three options: *USAA Login, Link to Download Files,* and *Upgrade Login*. To export YTD Checking data, select *USAA Login* and log into USAA bank.

Once in USAA the checking balance is displayed along with the savings balance. Select the checking balance for *USAA Classic Checking 7707* and choose *Export*. In the *Export* window click on *Select Transaction Range*, *Select Date Range* and input *From* and *To* date ranges.

Always select the *From* date as January 1 and the *To* date as the current day. This way we get the year to date information to process in our program. When the export is complete it will tell you it has created a new file prefaced with *bk*. For example: *bk\_download, bk\_download(1), bk\_download(2)* etc.

To see this file on Windows, select *Link to Download Files* as shown in Figure 1.2. This takes you to the file folders in Windows. Scroll up through the list until you see . Select this folder and the system will show you all of the download files.

First delete the existing *YTD Checking* file. Then rename the *bk\_download* file to *YTD Checking*, Exit Windows. This presents the *Choose Worksheet* user form shown in Figure 1.1.

Select the dropdown list and choose *YTD USAA Checking*. This action executes two macros (programs): *ImportYTDCheckingcsv* and *ExpenseType*. These programs are explained in detail later in this document.

Once the *ExpenseType* macro completes its execution, the user is automatically returned to the *Dashboard* worksheet shown in Figure 1.4.

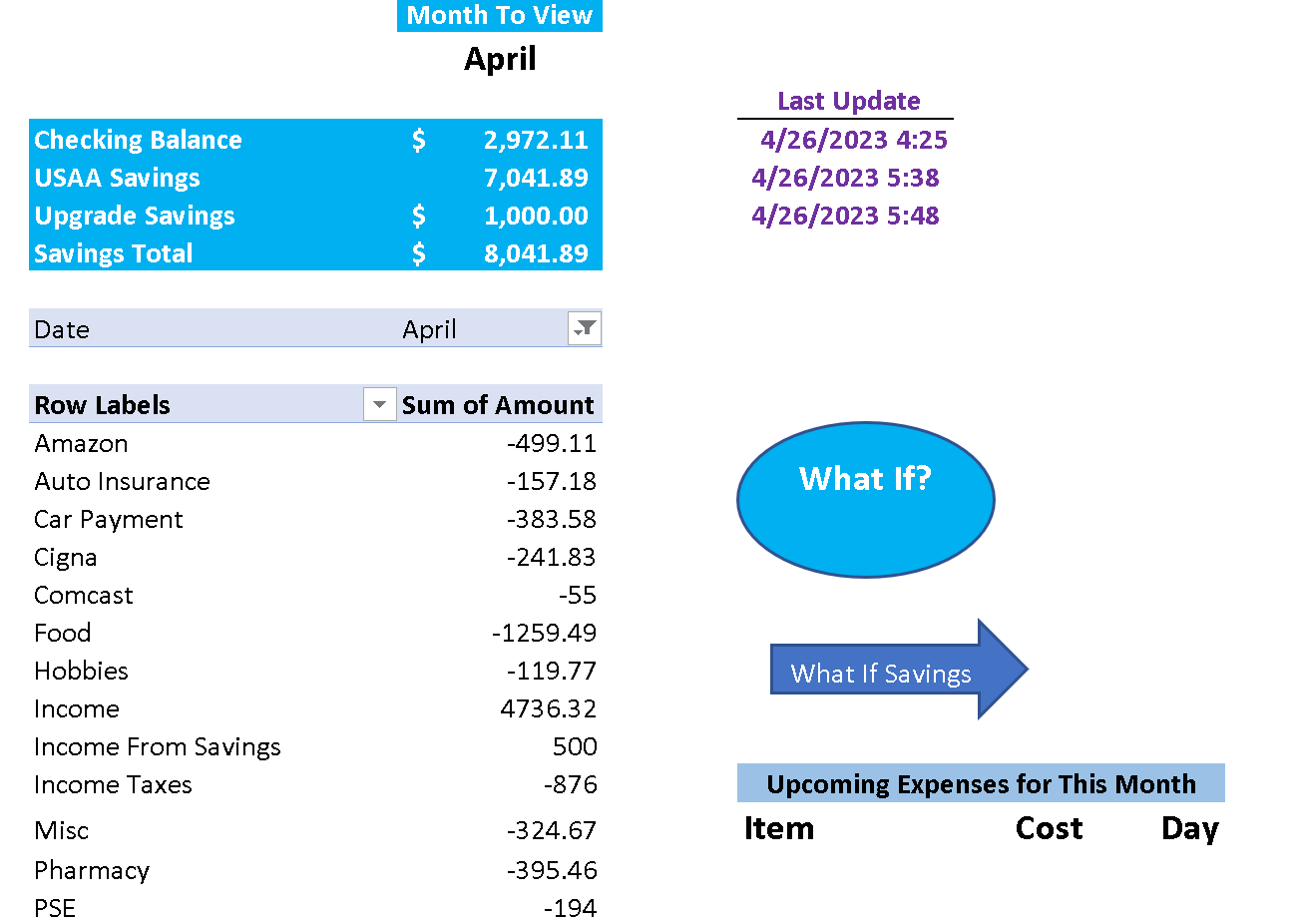


Figure 1.4

* 1. **Updating the Pivot Table**

Since the data source has changed (it is still YTD Checking but it has been updated) the pivot table source must be updated. To do this click on one of the items in the table (e.g., Amazon). This will display the *Pivot Table Analyze* option in the ribbon. Click on it. Figure 1.5 shows the ribbon after *Pivot Table Analyze* is selected.

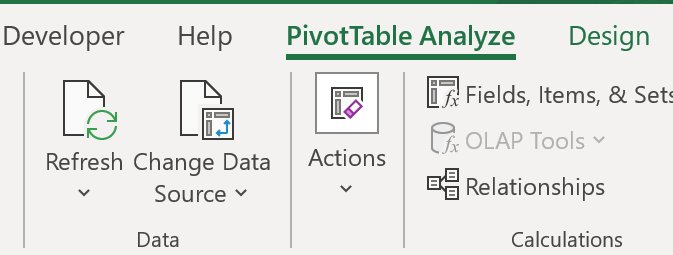


Figure 1.5

Select *Change Data Source*. After confirming you want to change the data source, the popup shown in Figure 1.6 appears.

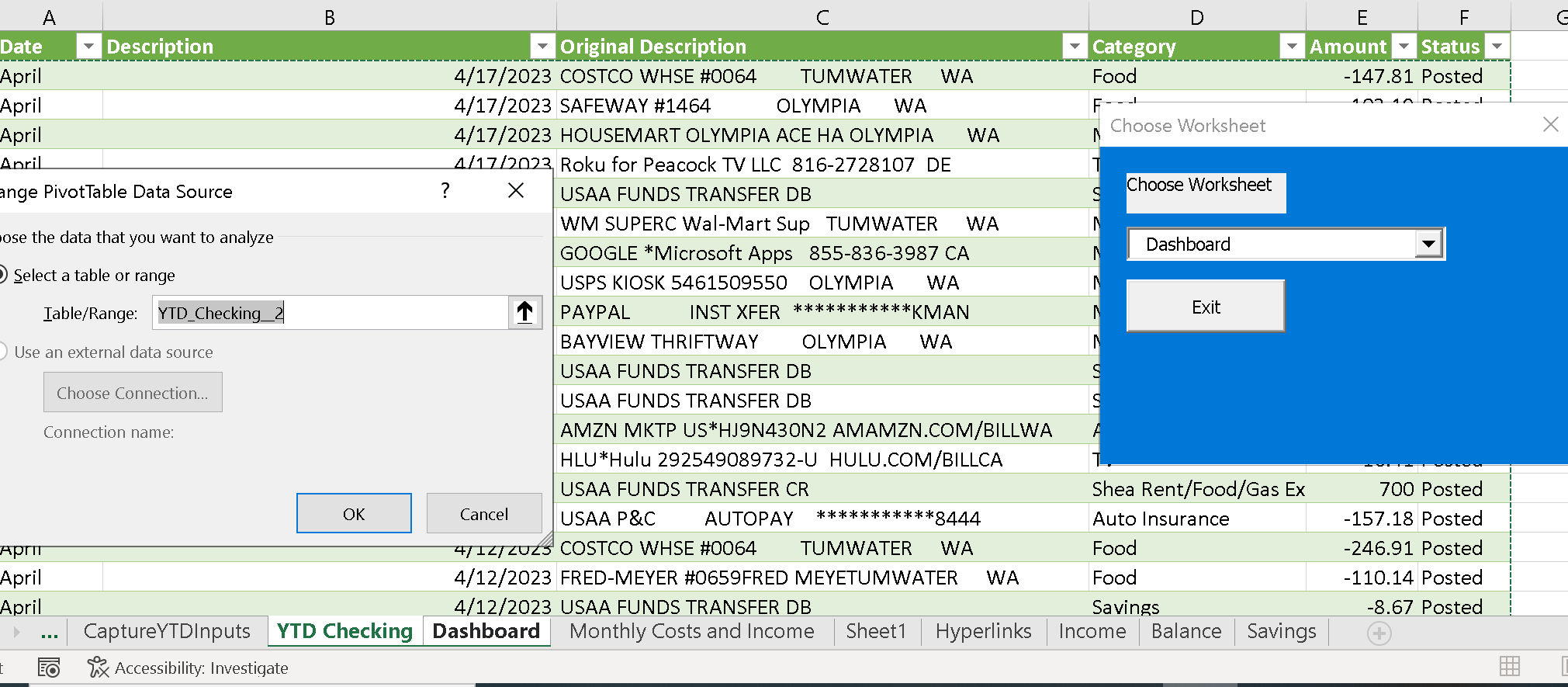


Figure 1.6

The *Table/Range:* is highlighted with the current source displayed and the table in the background. Select the *Date* column on the table and key in Ctrl + Shift + Down Arrow. Then press Ctrl + Shift + Right Arrow. This selects the entire table. The source for the pivot table is updated. Enter Return key and you’ll be back at the *Dashboard* worksheet.

Select *Refresh* from the ribbon and select *Refresh* from the drop down. Don’t select *Refresh All* as this will result in an error. The pivot table is now updated. Select a different month to view to see the updated table.

* 1. **Dashboard**

The *Dashboard* worksheet is shown in Figure 1.4. Under the *Month to View* heading is the current selected month. The user can select different months from the dropdown list by clicking on the month and selecting the month to view using the arrow to the right. Also note that the *Checking Balance* and savings balances are shown along with the date and time they were updated.

The *Item* and *Sum of Amount* columns provide insight into how much money was spent and where for the month selected. This is a pivot table and is explained thoroughly later in this document.

* + 1. **Upcoming Costs and Income for This Month**

The data for this section is provided by a macro *DashboardCosts*. It gets information about the upcoming costs from the *Monthly Costs and Income* worksheet which contains all of the costs for the current month that are fixed by amount and date.

To run this macro, choose the macro from the Quick Access Toolbar as shown in figure 1.7.

 *Dashboard Costs*

Figure 1.7

Selecting this icon clears all of the fields of the costs section and imports the costs greater than or equal to today’s date. If a cost has already occurred it will not be displayed.

It also displays the upcoming income for the month. If the income date has passed, it will not be displayed.

* + 1. **What If**

Selecting this option opens the *frmWhatIf* user form. This presents the user with a form for inputting the proposed expense and the day of the expense. The system then processes the information and returns with an approximate balance in the checking account based on estimated expenses and income between the proposed day and the current date.

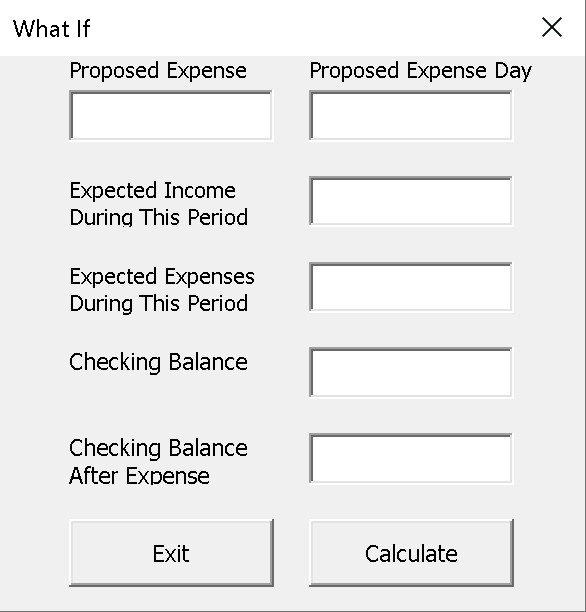


Figure 1.8

For a proposed expense, enter the amount in the *Proposed Expense* field, and the day of the proposed expense in the *Proposed Expense Day.* The entered day must be in the current month and be greater or equal to today’s date. For example if we want to put $500 in savings on the 20th, we enter 500 in the *Proposed Expense* field and 20 in the *Proposed Expense Day* field. Then select *Calculate* to process the request .

The system computes the expected income based on the *Expense Day* and today’s date. Further it computes the estimated expenses (based on daily averages), presents the current checking balance and shows the checking balance after the estimated expenses and proposed expense are added.

* + 1. **What If Savings**

Selecting this option runs the macro *SavingsForm* which displays the *frmSavings* form providing the user with the option to select either USAA or Upgrade banks. Once the choice is made, the worksheet *Savings* is displayed with the information relevant to which bank was chosen (interest rates vary).

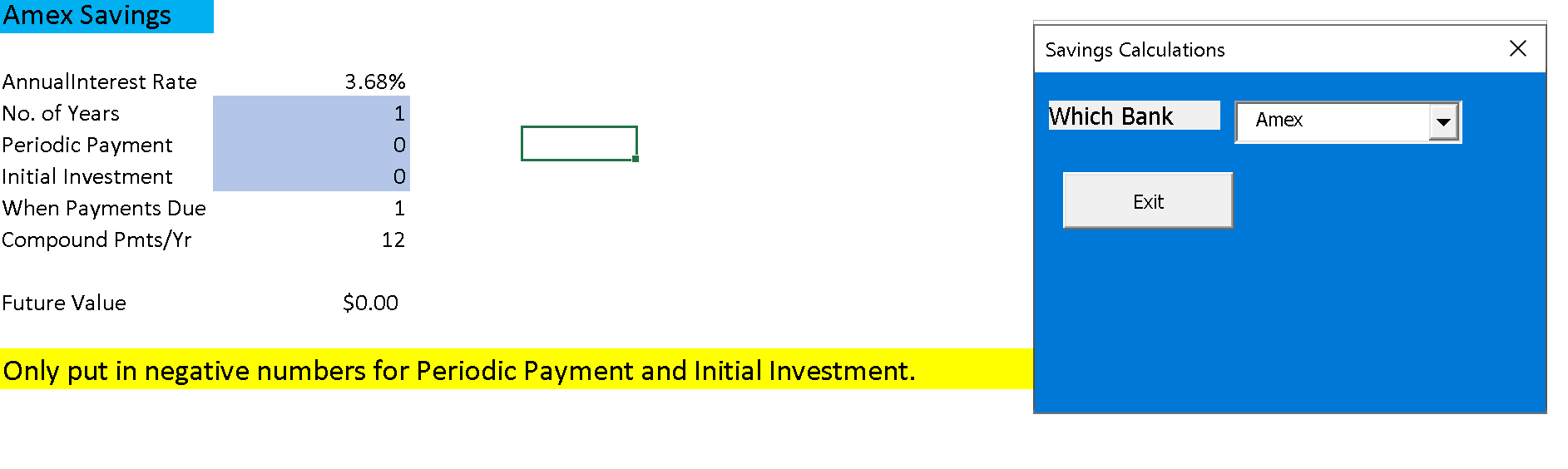


Figure 1.9

On this worksheet, the user may input the number of years, the periodic payment and the initial investment. The periodic payment and initial investment must be negative numbers. The result will show the future value based on the interest rate, initial investment, periodic payment, and number of payments per year. To do this the Excel function =FV is used.

* 1. **Import USAA Savings**

To start the process, choose *USAA, Upgrade, Download* as shown in Figure 1.1. This opens the user form shown in Figure 1.2 where the hypertext links are offered. Select the *USAA Login* hypertext link to open the URL for USAA.COM.

Login to USAA and select the Savings Balance. Choose *Export* and select the dates to download from January 1 to today’s date. Once the system completes the download, it will show which file it downloaded as *bk\_download*. This is a comma separated values (csv) file containing the banking information.

Exit USAA and return to the hypertext link page. Select *Link to Download Files* and delete *USAA Savings.* Then rename the *bk\_download* file to *USAA Savings*. Once this is done the next step is to import the csv savings file.

From the *Choose Worksheet* form (frmChooseWorksheet) choose *Import YTD Savings.* This opens the form *frmImportSavings* which is not a physical form, but it has code behind the form which imports the csv files. To do this it executes the *USAA\_Savings* macro. After this macro is complete, it returns control back to the code behind the *frmImportSavings* form. This code sums the credits and debits from the csv file and puts the result in the *Balance* worksheet where the initial savings amount as of January 1 is added to the result of the sum of the debits and credits. This puts the balance of the savings account on the *Balance*  worksheet. The worksheet is then switched to the *Dashboard* worksheet. The balance of the savings account is updated on the *Dashboard* worksheet from the *Balance* worksheet.

* 1. **Import Upgrade Savings**

There is no transaction history for Upgrade. Just get the balance and enter it into the Dashboard worksheet.

* 1. **The Code Behind the Forms, Worksheets and Macros**

Code may be stored in three distinct places: as a macro, in a worksheet, or in a form. All three have been utilized in making this program.

The first thing that happens when the workbook is opened is the code associated with opening the workbook is executed.

Private Sub Workbook\_Open()

Worksheets("Dashboard").Activate

End Sub

This displays the *Dashboard* worksheet which is the heart of the program. On this sheet is the user form *Choose Worksheet* (*frmChooseWorksheet*)providing the user with the means to navigate through the system. The form has two items of code that run in the background populating the form and responding to the user’s input.

* 1. **Private Sub UserForm\_Initialize**

Private Sub UserForm\_Initialize()

Me.cboChooseWorksheet.AddItem "USAA, Upgrade, Download"

Me.cboChooseWorksheet.AddItem "Import YTD USAA Savings"

Me.cboChooseWorksheet.AddItem "Import YTD USAA Checking"

Me.cboChooseWorksheet.AddItem "Dashboard”

End Sub

Each form has events associated with it. One of these is the *initialize event*. This code is executed upon initialization of the *Choose Worksheet* form. Its purpose is to set up the combo box (*cboChooseWorksheet)* is the name of the control on the form. Items are added for the drop down control (combo box) on the form. A second code is run to display the user form to the user.

* 1. **Private Sub cboChooseWorksheet\_Change()**

Private Sub cboChooseWorksheet\_Change()

If cboChooseWorksheet.Value = “Import YTD USAA Savings" Then

frmImport\_Savings.Show ‘If the user chose Import USAA Savings then open this form

ElseIf cboChooseWorksheet.Value = "Import YTD USAA Checking" Then

ImportYTDCheckingcsv

ElseIf cboChooseWorksheet.Value = "Dashboard" Then

Worksheets("Dashboard").Activate

ElseIf cboChooseWorksheet.Value = "USAA, Upgrade, Download" Then

Worksheets("Hyperlinks").Activate

End If

End Sub

The code in this portion of the form executes based on the change of any field associated with a change in the cboChooseWorksheet combo box. When the user makes a selection, the system registers that selection and, using If statements, it determines what happens next.

For the first option, the form *frmImport\_Savings* is executed. This form does not have a physical presence on the computer, but it has code behind the form which executes when the form is opened.

* 1. **frmImportSavings**

The following is the code behind the form that is executed when this form is selected.

Private Sub UserForm\_Activate()

Dim Answer As Integer

Dim YTDSavings As Long

Dim SheetName As String

Application.DisplayAlerts = False 'Turn off the alerts

USAA\_Savings 'run this macro it gets the csv file data

ActiveSheet.Select

SheetName = ActiveSheet.Name ‘get the worksheet name so we can come back to it

YTDSavings = Application.WorksheetFunction.Sum(Range("E:E")) 'sum column E debits and credits

Worksheets("Balance").Activate ‘go to the Balance ws and open it up

Range("$B$5").Value = YTDSavings 'update $B$5 in the Balance worksheet - YTD Debits and Credits from the USAA savings account

Worksheets(SheetName).Activate 'reactivate the worksheet containing the csv data

ActiveSheet.Delete ‘delete the current worksheet

Application.DisplayAlerts = True 'Turn on the alerts

Worksheets("Dashboard").Activate ‘go to the Dashboard ws

Range("E5").Value = Now() 'put in date of update on the Dashboard ws

End Sub

* + 1. **USAA\_Savings Macro**

Sub USAA\_Savings()

' This is all computer generated

' USAA\_Savings Macro

' Import the USAA Savings csv file and update Monthly Summary savings cell $N$19

'

'

'ActiveWorkbook.Queries.Delete = Name: = "USAA Savings"

On Error Resume Next

ActiveWorkbook.Queries("USAA Savings").Delete

ActiveWorkbook.Queries.Add Name:="USAA Savings", Formula:= \_

"let" & Chr(13) & "" & Chr(10) & " Source = Csv.Document(File.Contents(""C:\Users\cheri\Downloads\USAA Savings.csv""),[Delimiter="","", Columns=6, Encoding=1252, QuoteStyle=QuoteStyle.None])," & Chr(13) & "" & Chr(10) & " #""Promoted Headers"" = Table.PromoteHeaders(Source, [PromoteAllScalars=true])," & Chr(13) & "" & Chr(10) & " #""Changed Type"" = Table.TransformColumnTypes(#""Promoted Headers"",{{""Date"", type date}, {""Description"", t" & \_

"ype text}, {""Original Description"", type text}, {""Category"", type text}, {""Amount"", type number}, {""Status"", type text}})" & Chr(13) & "" & Chr(10) & "in" & Chr(13) & "" & Chr(10) & " #""Changed Type"""

ActiveWorkbook.Worksheets.Add

With ActiveSheet.ListObjects.Add(SourceType:=0, Source:= \_

"OLEDB;Provider=Microsoft.Mashup.OleDb.1;Data Source=$Workbook$;Location=""USAA Savings"";Extended Properties=""""" \_

, Destination:=Range("$A$1")).QueryTable

.CommandType = xlCmdSql

.CommandText = Array("SELECT \* FROM [USAA Savings]")

.RowNumbers = False

.FillAdjacentFormulas = False

.PreserveFormatting = True

.RefreshOnFileOpen = False

.BackgroundQuery = True

.RefreshStyle = xlInsertDeleteCells

.SavePassword = False

.SaveData = True

.AdjustColumnWidth = True

.RefreshPeriod = 0

.PreserveColumnInfo = True

.ListObject.DisplayName = "USAA\_Savings"

.Refresh BackgroundQuery:=False

End With

End Sub

This macro was recorded on the computer and it imports data from the *USAA Savings* csv file. When it finishes, it relinquishes control back to the calling program (the user form from the form *frmImportSavings*) which takes the user back to the *Dashboard* worksheet.

* 1. **ImportYTDCheckingcsv**

If Import Checking is chosen, this macro runs. This macro imports the csv file for the *USAA YTD* Checking file. Ultimately it creates a worksheet and calls the *ExpenseType* macro.

Sub ImportYTDCheckingcsv()

**'This is computer generated except for the part In blue.**

**'** ImportYTDCheckingcsv Macro

**'**

**'**

**Dim ws As String**

**Dim SumYTDCheckings As Currency**

**ws = ActiveSheet.Name**

**Application.ScreenUpdating = False ‘Turn off the screen while updates are occurring**

**On Error Resume Next**

**ActiveWorkbook.Queries("YTD Checking (2)").Delete ‘Delete the existing query to avoid conflicts**

ActiveWorkbook.Queries.Add Name:="YTD Checking (2)", Formula:= \_

"let" & Chr(13) & "" & Chr(10) & " Source = Csv.Document(File.Contents(""C:\Users\cheri\Downloads\YTD Checking.csv""),[Delimiter="","", Columns=6, Encoding=1252, QuoteStyle=QuoteStyle.None])," & Chr(13) & "" & Chr(10) & " #""Promoted Headers"" = Table.PromoteHeaders(Source, [PromoteAllScalars=true])," & Chr(13) & "" & Chr(10) & " #""Changed Type"" = Table.TransformColumnTypes(#""Promoted Headers"",{{""Date"", type date}, {""Description"", t" & \_

"ype text}, {""Original Description"", type text}, {""Category"", type text}, {""Amount"", type number}, {""Status"", type text}})" & Chr(13) & "" & Chr(10) & "in" & Chr(13) & "" & Chr(10) & " #""Changed Type"""

ActiveWorkbook.Worksheets.Add

ws = ActiveSheet.Name

With ActiveSheet.ListObjects.Add(SourceType:=0, Source:= \_

"OLEDB;Provider=Microsoft.Mashup.OleDb.1;Data Source=$Workbook$;Location=""YTD Checking (2)"";Extended Properties=""""" \_

, Destination:=Range("$A$1")).QueryTable

.CommandType = xlCmdSql

.CommandText = Array("SELECT \* FROM [YTD Checking (2)]")

.RowNumbers = False

.FillAdjacentFormulas = False

.PreserveFormatting = True

.RefreshOnFileOpen = False

.BackgroundQuery = True

.RefreshStyle = xlInsertDeleteCells

.SavePassword = False

.SaveData = True

.AdjustColumnWidth = True

.RefreshPeriod = 0

.PreserveColumnInfo = True

.ListObject.DisplayName = "YTD\_Checking\_\_2"

.Refresh BackgroundQuery:=False

Application.ScreenUpdating = True

SumYTDCheckings = Application.WorksheetFunction.Sum(Range("E:E")) 'sum column E'

Worksheets("Balance").Activate

Range("$B$4").Value = SumYTDCheckings 'update $B$4in the Balance worksheet - Yearly Checkings

End With

**Worksheets("Dashboard").Activate ‘Return to the Dashboard ws**

**Range("E4").Value = Now() 'put the date and time on the Dashboard ws to show when it was last updated**

**Worksheets(ws).Activate ‘Re-activate the csv worksheet so the program can process the data**

**ExpenseType 'call this macro**

End Sub

* 1. **ExpenseType Macro**

The expense type macro compares all of the descriptions of debits and credits from the checking account with known categories of expenses that we want to track. For instance, if the description contains “Innovative” this flags it as income from Innovative Care Management. Depending on the date of the payday, it is assigned the category of ICM MidMonth or ICM EndMonth. This helps us to average the mid-month and end-month values for ICM so we can predict what income is possible for the mid-month and end of month paydays. This is used in the WhatIf form to estimate income with dates that fall after the date of the proposed expense.

Public Sub ExpenseType()

Dim noData As Boolean

Dim strText As String

Dim userInput As String

Dim cellValue As String

Dim searchString As String

Dim SearchText As String

Dim strCategory As String

Dim strOriginDesc As String

Dim MyPos As Integer

Dim arrcategory(0 To 44) As Long

Dim i As Integer

Dim DDay As Date

Dim ICMDay As Integer

Dim TransactionDate As Date

Dim TransactionDay As Integer

Dim MnthName As String

Dim DDate As Date

Dim Mnth As Integer

Dim LastRow As Long

Dim ws As String

ws = ActiveSheet.Name

Application.ScreenUpdating = False ‘turn the screen off so the user can’t see changes

'Worksheets("YTD Checking").Activate ‘the csv file

LastRow = Cells(Rows.Count, 1).End(xlUp).Row ‘Find the last row in the worksheet (Always use xlUp).

‘The csv file has 6 columns shown as A – F. We are only interested in columns A, B, C, D, and E. These

Have the Date, Description, Category, and Amount.

‘In this next section, read the dates from column A in the csv file and change them to Month

For i = 2 To LastRow ‘The first row has table headers

On Error Resume Next ‘If there is an error, just go on to the next command

If ActiveCell.Row <> LastRow Then ActiveCell.Offset(1, 0).Select ‘If not on the last row move down one row. This ensures the program starts at row 2 as row 1 has header information for the csv table

DDate = ActiveCell.Value ‘For example 4/18/2023 this is the date of the transaction

MnthName = (Format(DDate, "mmmm")) ‘Get the full month name from the date in column 1. For 4/18/2023 this would equate to April. This data is used to filter the pivot table in the *Dashboard* work sheet to filter the results by month.

Range("B" & i) = DDate 'The original purpose of the B column was to hold a description of the transaction. We need it for the date dd/mm/yyyy which is used only for discerning the date of pay for the ICM deposits into the checking account to determine if they are mid-month or end month deposits. This is used in the WhatIf form to determine average pay over the year to date credits for ICM.

Range("A" & i) = MnthName ‘I starts at 2 so we don’t overwrite the heading with the month

Next I ‘increment the value of I by 1

noData = False

Worksheets(ActiveSheet.Name).Range("C2").Activate 'sets the active cell of the current worksheet to C2 the first entry in the Category column row 2 (first row has headers).

Do While noData = False ‘as long as there is data in the csv file, keep processing)

SearchText = "DDA”

strCategory = ActiveCell.Value ‘This gets the text from column C as a basis to search for the SeachText

MyPos = InStr(1, strCategory, SearchText, 1) ‘strCategory has the full description of the transaction. SearchText contains part of this description. Look for the text contained in SearchText in the strCategory text. If a match is found, put the positon where the first character is found into the variable MyPos. MyPos contains the position of the first letter of the text searched for in strCategory. If the bit of text wasn’t found, MPos = 0.

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate ‘The system is currently in column C so go to column D in the current row in the csv file which contains the original category.

ActiveCell.Value = "Debit" ‘Replace the current value in the D column with this new input

ActiveCell.Offset(0, -1).Select ‘go back to the C column in the same row which contains the description of the transaction

GoTo Line1: ‘Don’t go through the rest of the If statements. Increment the row number and start the search over.

End If

SearchText = "PETCO"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "PAYPAL"

'ActiveCell.Offset(0, -1).Activate

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "TUMWATER AUTOMOTIVE"

'ActiveCell.Offset(0, -1).Activate

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Car Maint"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "VERIZON"

'ActiveCell.Offset(0, -1).Activate

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Verizon"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "USPS"

'ActiveCell.Offset(0, -1).Activate

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "229"

'ActiveCell.Offset(0, -1).Activate

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Debit"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Express"

'ActiveCell.Offset(0, -1).Activate

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Pharmacy"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "DDA"

'ActiveCell.Offset(0, -1).Activate

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "DDA"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Amzn"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Amazon"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Micro"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Credit One Bank"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Credit Card"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Amazon"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Amazon"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "JOANN"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Hobbies"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "JO-ANN"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Hobbies"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Fuel"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Gas"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "DB"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Savings"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "IRS"

'ActiveCell.Offset(0, -1).Activate

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Income Taxes"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "CR"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 2).Select

If ActiveCell.Value = 700 Then

ActiveCell.Offset(0, -1).Activate

ActiveCell.Value = "Shea Rent/Food/Gas Expenses"

Else

ActiveCell.Offset(0, -1).Activate

ActiveCell.Value = "Income From Savings"

End If

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "HOUSEMART"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Benefit"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Income"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "COMCAST"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Comcast"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "KUOW"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Paid"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Income"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "USAA P&C"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Auto Insurance"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = ""

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

TransactionDate = ActiveCell.Offset(0, -2).Value

TransactionDay = DatePart("d", TransactionDate)

If TransactionDay = DatePart("d", TransactionDate) = 11 Then 'Insurance is always on the 11th of the month

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Auto Insurance"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

End If

SearchText = "USAA INSURANCE"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Auto Insurance"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "HAGGEN"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Food"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Kindle"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Amazon"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Sound CU"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Car Payment"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "HOME DEPOT"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Fred-Meyer"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Food"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Multicare"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Medical"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Puget Sound"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "PSE"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "REFUND"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Income"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Microsoft"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "UDEMY"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Hobbies"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "CHECK"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Utilities"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Hulu"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "TV"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Walgreens"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Pharmacy"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Costco"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Food"

ActiveCell.Offset(0, -1).Select

'msgBox ActiveCell.Address

GoTo Line1:

End If

SearchText = "SSA"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Income"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "PNNL"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Income"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Safeway"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Food"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Fred"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Food"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "TMobile"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "T-Mobile"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Roku"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "TV"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Innov"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, -1).Activate 'go to the mm/dd/yyyy date column

MsgBox ActiveCell.Value

DDate = ActiveCell.Value

ICMDay = Format(DDate, "dd") 'get the day of the month the payday occured on

If ICMDay <= 15 Then

ActiveCell.Offset(0, 2).Select

ActiveCell.Value = "ICM Mid-Month" 'track mid-month and end-month for monthly income expenses ws

Else

ActiveCell.Offset(0, 2).Select

ActiveCell.Value = "ICM End-Month"

End If

ActiveCell.Offset(0, -1).Select

'MsgBox ("Innovative Care ActiveCell.Address = " & ActiveCell.Address)

GoTo Line1:

End If

SearchText = "RE/MAX"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Rent"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "CIGNA-LOYAL"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Cigna"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "CR"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Savings"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "Verizon"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Verizon"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

SearchText = "JO"

strCategory = ActiveCell.Value

MyPos = InStr(1, strCategory, SearchText, 1)

If MyPos > 0 Then

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Hobbies"

ActiveCell.Offset(0, -1).Select

GoTo Line1:

End If

If MyPos = 0 Then

‘If the entire IF list has been gone through and no matches occurred, it is reasonable to conclude that the credit or debit is Misc. It is possible to ask the user to select the expense type from a drop down list but this proves to be tedious and unnecessary. The alternative is to have anything not in an If statement on this list to be Misc. Experience shows that most un-matched items are Misc anyway, so this option was chosen.

'ActiveCell.Offset(0, 1).Activate ‘if uncommented this and the next line of code provide the means to ask the user to input the category.

'frmExpenseType.Show 'Open the form frmExpenseType

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = "Misc"

ActiveCell.Offset(0, -1).Select

End If

'GoTo Line1:

Line1:

ActiveCell.Offset(1, 0).Select 'Continue through the rows of columns by incrementing the row until the last row has been processed.

If ActiveCell.Value = "" Then ‘If there is no data in the row after the last row then exit the loop.

noData = True

End If

Loop

‘After the contents of the csv file have been processed …

Application.ScreenUpdating = True

Worksheets(ws).Name = "YTD Checking" ‘rename the working worksheet to YTD Checking

Worksheets("Dashboard").Activate ‘Return the user to the *Dashboard* worksheet.

End Sub

* 1. **Dashboard**

This is the heart of the program and the cornerstone worksheet for the project. It is capable of showing a month by month listing of expenses, updating the checking and savings balance, showing what if scenarios for checking, and what if for savings for future planning.

There is code behind the worksheet for this worksheet.

Private Sub Worksheet\_Change(ByVal Target As Range)

This simple code behind the worksheet provides the means of selecting a specific month to view through the filtering process in Excel. I did not write it, I just used it from the internet. But it was my idea to have a single cell to use as a filter for the data displayed on the pivot table. A pivot table is a summary of information from a table that contains many items. Our pivot table identifies like items and sums their associated debits and credits presenting them in the table shown to the user.

Private Sub Worksheet\_Change(ByVal Target As Range)

Dim xPTable As PivotTable

Dim XPFile As PivotField

Dim xStr As String

If Target.Address = "$C$2" Then ‘check to see is cell C2 (month to display) is being changed. Changing any other cell will not run the pivot table macro. For example, when the macro *DashbordCosts* is run changing values in E26:G31 then the pivot table will be ignored.

On Error Resume Next

If Intersect(Target, Range("C2")) Is Nothing Then Exit Sub

Application.ScreenUpdating = False

Set xPTable = Worksheets("Dashboard").PivotTables("PTableYTDExpenses")

Set XPFile = xPTable.PivotFields("Date")

xStr = Target.Text

XPFile.ClearAllFilters

XPFile.CurrentPage = xStr

Application.ScreenUpdating = True

End If

End Sub

* + 1. **What If**

There are two parts to the What If process: the what if for *frmWhatIf* and the worksheet *Monthly Costs and Income*. The code behind the form works in conjunction with the worksheet to provide the data necessary to perform the calculations.

The following is the code behind the what if form:

Private Sub cboCalculate\_Click()

Dim DDate As Date

Dim ChkBalance As Integer

Dim DaysUntilIncome As Integer

Dim CurDay As Integer

Dim ExpenseDate As Integer

Dim SheaAmount As Integer

Dim SSAAmountVicki As Integer

Dim SSAAmountJerry As Integer

Dim PNNLAmount As Integer

Dim ICMAmount As Integer

Dim SumDailyExpenses As Integer

Dim SumDailyIncome As Integer

Dim SubTotal As Integer

Dim DaysToExpense As Integer

Dim TotalDailyExpense As Integer

Dim TotalPayDayIncome As Integer

Dim FDDate As Integer

Dim TotalDailyExpenses As Integer

Dim SubMoExpenses As Integer

Dim Year As Integer

Dim Mnth As Integer

Dim SecondWed As Integer

Dim FourthWed As Integer

Dim subDailyExpenses As Long

Dim SubUpcomingExpenses As Long

Dim TotalExpenses As Long

Dim PayDate As Integer

Dim j As Integer

Dim NextIf As Boolean

NextIf = True

PayAmount = 0

SubMoExpenses = 0

ExpenseDate = CInt(Me.txtWhatIfExpenseDate) ‘convert the text value to an integer

Worksheets("Dashboard").Activate

ChkBalance = Range("$C$4").Value 'get the checking balance from the Dashboard worksheet

Worksheets("Monthly Costs And Income").Activate

ExpenseDate = CInt(Me.txtWhatIfExpenseDate) ‘convert from text to integer

Range("B31").Value = ExpenseDate

VickiSSADate = Range("C23").Value

VickiSSAAmt = Range("B23")

VickiICMDate = Range("C24").Value

VickiICMAmt = CInt(Range("B24").Value)

SheaMidMoDate = 14

SheaMidMoAmt = 700

JerrySSADate = Range("C26").Value

JerrySSAAmt = Range("B26").Value

VickiICMDate2 = 28

VickiICMAmt2 = Range("C27").Value

SheaEndMoDate = 28

SheaEndMoAmt = 700

PNNLDate = 1

PNNLAmt = 834.8

'Calculate the 2nd and 4th Wednesday of the current year and month

DDate = Now()

Mnth = CInt(Format(DDate, "mm")) 'curent month

Year = CInt(Format(DDate, "yyyy")) 'current year

SecondWed = CInt(Format(Range("B19"), "dd")) 'second Wednesday of the month

FourthWed = CInt(Format(Range("B20"), "dd")) 'fourth Wednesday of the month

FDDate = CInt(Format(DDate, "dd")) 'Current day of the current date

Range("B34").Value = FDDate 'update the day in the month costs and income ws

ExpenseDate = CInt(Me.txtWhatIfExpenseDate)

If ExpenseDate >= PNNLDate And FDDate <= PNNLDate Then

TotalPayDayIncome = PNNLAmt

PayDate = PNNLDate

ElseIf ExpenseDate >= VickiSSADate And FDDate <= VickiSSADate Then

TotalPayDayIncome = VickiSSAAmt

PayDate = VickiSSADate

ElseIf ExpenseDate >= VickiICMDate And FDDate <= VickiICMDate Then

TotalPayDayIncome = VickiICMAmt + SheaMidMoAmt ‘add Vicki mid-month pay with Shea amount paid

ElseIf ExpenseDate >= JerrySSADate And FDDate <= JerrySSADate Then

TotalPayDayIncome = JerrySSAAmt

PayDate = JerrySSADate

End If

‘If end of month ICM payday, add Shea income

If ExpenseDate >= VickiICMDate2 And FDDate <= VickiICMDate2 Then

TotalPayDayIncome = TotalPayDayIncome + VickiICM2Amt + SheaEndMoAmt

PayDate = VickiICMDate2

End If

'Daily Expenses H29 through H36

Do Until NextIf = False

'If the fixed cost >= FDDate and <= Expense Date then include the expense cost in the subMoExpenses. Check for expenses around the FDDate. We must check all expenses for each of the initial dates to see if there is another expense on the same day or a day less than or equal to the value in B31 which is the date of the proposed expense. For example Cigna (first of month) and rent are both due on the 2nd of the month. Without these If statements, only one of the expenses would be found.

If Range("H29").Value >= FDDate And Range("H29").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G29").Value

If Range("H30").Value >= FDDate And Range("H30").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G30").Value

If Range("H31").Value >= FDDate And Range("H31").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G31").Value

If Range("H32").Value >= FDDate And Range("H32").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G32").Value

If Range("H33").Value >= FDDate And Range("H33").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G33").Value

If Range("H34").Value >= FDDate And Range("H34").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G34").Value

If Range("H35").Value >= FDDate And Range("H35").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G35").Value

If Range("H36").Value >= FDDate And Range("H36").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G36").Value

End If

End If

End If

End If

End If

End If

End If

NexIf = False

End If

If Range("H30").Value >= FDDate And Range("H30").Value <= Range("B31").Value And NextIf <> False Then

SubMoExpenses = SubMoExpenses + Range("G30").Value

If Range("H31").Value >= FDDate And Range("H31").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G31").Value

If Range("H32").Value >= FDDate And Range("H32").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G32").Value

If Range("H33").Value >= FDDate And Range("H33").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G33").Value

If Range("H34").Value >= FDDate And Range("H34").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G34").Value

If Range("H35").Value >= FDDate And Range("H35").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G35").Value

If Range("H36").Value >= FDDate And Range("H36").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G36").Value

End If

End If

End If

End If

End If

End If

NexIf = False

End If

If Range("H31").Value >= FDDate And Range("H31").Value <= Range("B31").Value And NextIf <> False Then

SubMoExpenses = SubMoExpenses + Range("G31").Value

If Range("H32").Value >= FDDate And Range("H32").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G32").Value

If Range("H33").Value >= FDDate And Range("H33").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G33").Value

If Range("H34").Value >= FDDate And Range("H34").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G34").Value

If Range("H35").Value >= FDDate And Range("H35").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G35").Value

If Range("H36").Value >= FDDate And Range("H36").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G36").Value

End If

End If

End If

End If

End If

NexIf = False

End If

If Range("H32").Value >= FDDate And Range("H32").Value <= Range("B31").Value And NextIf <> False Then

SubMoExpenses = SubMoExpenses + Range("G32").Value

If Range("H33").Value >= FDDate And Range("H33").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G33").Value

If Range("H34").Value >= FDDate And Range("H34").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G34").Value

If Range("H35").Value >= FDDate And Range("H35").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G35").Value

If Range("H36").Value >= FDDate And Range("H36").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G36").Value

End If

End If

End If

End If

NexIf = False

End If

If Range("H33").Value >= FDDate And Range("H33").Value <= Range("B31").Value And NextIf <> False Then

SubMoExpenses = SubMoExpenses + Range("G33").Value

If Range("H34").Value >= FDDate And Range("H34").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G34").Value

If Range("H35").Value >= FDDate And Range("H35").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G35").Value

If Range("H36").Value >= FDDate And Range("H36").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G36").Value

End If

End If

End If

NexIf = False

End If

If Range("H34").Value >= FDDate And Range("H34").Value <= Range("B31").Value And NextIf <> False Then

SubMoExpenses = SubMoExpenses + Range("G34").Value

If Range("H35").Value >= FDDate And Range("H35").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G35").Value

If Range("H36").Value >= FDDate And Range("H36").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G36").Value

End If

End If

NexIf = False

End If

If Range("H35").Value >= FDDate And Range("H35").Value <= Range("B31").Value And NextIf <> False Then

SubMoExpenses = SubMoExpenses + Range("G35").Value

If Range("H36").Value >= FDDate And Range("H36").Value <= Range("B31").Value Then

SubMoExpenses = SubMoExpenses + Range("G36").Value

End If

NexIf = False

End If

If Range("H36").Value >= FDDate And Range("H36").Value <= Range("B31").Value And NextIf <> False Then

SubMoExpenses = SubMoExpenses + Range("G36").Value

NexIf = False

End If

NextIf = False

Loop

subDailyExpenses = Range("J14").Value ‘get the average daily expenses from the Month Costs and Income ws

TotalExpenses = SubMoExpenses + subDailyExpenses

DaysToExpense = (CInt(Me.txtWhatIfExpenseDate) - CInt(FDDate))

Me.txtWhatIfExpenses = TotalExpenses

Me.txtWhatIfIncome = TotalPayDayIncome

Me.txtWhatIfCurrentBalance = ChkBalance

SumMoExpenses = TotalDailyExpenses + TotalPayDayIncome

Me.txtWhatIfEndingBalance = ChkBalance - Me.txtProposedCost + SumMoExpenses

Worksheets("Dashboard").Activate

End Sub

* + 1. **What If Savings**

This form is a physical presence on the system. As such it has an associated Initialize event where it puts the two bank names in the combo box for user selection.

Private Sub UserForm\_Initialize()

cboWhichBank.AddItem "USAA"

cboWhichBank.AddItem "Upgrade"

End Sub

When the combo box is displayed, it provides two options: one for USAA bank and one for Upgrade bank. Each choice provides interest rates associated with each bank. The code behind the form for the combo box is:

Private Sub cboWhichBank\_Change()

‘This code is activated by a change on the worksheet such as number of years, initial amount, or number of payments.

Dim USAABalance As Long

Dim UpgradeBalance As Long

If cboWhichBank.Value = "USAA" Then

Worksheets("Savings").Activate

Range("A1").Value = "USAA Savings"

Range("B3").Value = 0.0001

‘The following three lines initialize the number of years, number of payments, and initial amount.

Range("B4").Value = 0

Range("B5").Value = 0

Range("B6").Value = 0

ElseIf cboWhichBank.Value = "Upgrade" Then

Worksheets("Savings").Activate

Range("A1").Value = "Upgrade Savings"

Range("B4").Value = 0

Range("B3").Value = 0.0368

Range("B5").Value = 0

Range("B6").Value = 0

End If

End Sub

The future value is derived from the function =FV on the cell containing the Future Value.

* 1. Dashboard Month Costs

The macro *DashboardCosts* in conjunction with the macro *clear\_dasboard\_costs* presents the upcoming costs for the current month. If the current day is greater than any costs expected to be incurred, the table for upcoming costs remains clear. The following code is used.

Public Sub DashboardCosts()

Dim IDate As Integer

Dim FDDate As Integer

Dim DDate As Date

Dim Item As String

Dim Cost As Integer

Dim Day As String

DDate = Now()

FDDate = Format(DDate, "dd")

'Clear all data from costs table

clear\_dashboard\_costs ‘macro to clear all upcoming costs on the dashboaord it is very simple:

Public Sub clear\_dashboard\_costs()

Worksheets("Dashboard").Activate

Range("E26:G31").Value = ""

End Sub

Worksheets("Monthly Costs and Income").Activate

If Range("C2").Value >= FDDate Then ‘compare the date in cell C2 of the Monthly Costs and Income ws with the current day of the month. If it is >= to this date, proceed. Otherwise go to the next If statement and do the same check for cell C3 and each subsequent value in the monthly costs and income worksheet. Each row of the table in the monthly costs and income worksheet must be checked (C2:C9) because if C2 or C3 or C4 If statements don’t result in a True condition, subsequent values must be checked.

Also, each value in the table (C2:C9) must be checked for each of the If statements in case there are more than one matches for dates >= the current day. For example, rent is due on the 2nd of the month and Cigna is due on the 2nd. So each value in the table of Monthly Costs and Income must be checked.

Item = Range("A2").Value

Cost = Range("B2").Value

Day = Range("C2").Value

Worksheets("Dashboard").Activate

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

End If

'If the fixed cost >= FDDate and <= Expense Date then include the expense cost in the 'SubMoExpenses

If Range("C2").Value >= FDDate Then

Item = Range("A2").Value

Cost = Range("B2").Value

Day = Range("C2").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C3").Value >= FDDate Then

''SubMoExpenses = 'SubMoExpenses + Range("B3").Value

Item = Range("A3").Value

Cost = Range("B3").Value

Day = Range("C3").Value

''SubMoExpenses = 'SubMoExpenses + Range("B2").Value

Worksheets("Dashboard").Select

Range("E27").Value = Item

Range("F27").Value = Cost

Range("G27").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C4").Value >= FDDate Then

''SubMoExpenses = 'SubMoExpenses + Range("B4").Value

Item = Range("A4").Value

Cost = Range("B4").Value

Day = Range("C4").Value

Worksheets("Dashboard").Select

Range("E28").Value = Item

Range("F28").Value = Cost

Range("G28").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C5").Value >= FDDate Then

Item = Range("A5").Value

Cost = Range("B5").Value

Day = Range("C5").Value

Worksheets("Dashboard").Select

Range("E29").Value = Item

Range("F29").Value = Cost

Range("G29").Value = Day

Worksheets("Monthly Costs and Income").Select

If Range("C6").Value >= FDDate Then

Item = Range("A6").Value

Cost = Range("B6").Value

Day = Range("C6").Value

Worksheets("Dashboard").Select

Range("E30").Value = Item

Range("F30").Value = Cost

Range("G30").Value = Day

Worksheets("Monthly Costs and Income").Select

If Range("C7").Value >= FDDate Then

Item = Range("A7").Value

Cost = Range("B7").Value

Day = Range("C7").Value

Worksheets("Dashboard").Select

Range("E31").Value = Item

Range("F31").Value = Cost

Range("G31").Value = Day

Worksheets("Monthly Costs and Income").Select

If Range("C8").Value >= FDDate Then

Item = Range("A8").Value

Cost = Range("B8").Value

Day = Range("C8").Value

Worksheets("Dashboard").Select

Range("E32").Value = Item

Range("F32").Value = Cost

Range("G32").Value = Day

Worksheets("Monthly Costs and Income").Select

If Range("C9").Value >= FDDate Then

Item = Range("A9").Value

Cost = Range("B9").Value

Day = Range("C9").Value

Worksheets("Dashboard").Select

Range("E31").Value = Item

Range("F31").Value = Cost

Range("G31").Value = Day

Worksheets("Monthly Costs and Income").Select

End If

End If

End If

End If

End If

End If

End If

If Range("C3").Value >= FDDate Then

Item = Range("A3").Value

Cost = Range("B3").Value

Day = Range("C3").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C4").Value >= FDDate Then

Item = Range("A4").Value

Cost = Range("B4").Value

Day = Range("C4").Value

Worksheets("Dashboard").Select

Range("E27").Value = Item

Range("F27").Value = Cost

Range("G27").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C5").Value >= FDDate Then

Item = Range("A5").Value

Cost = Range("B5").Value

Day = Range("C5").Value

Worksheets("Dashboard").Select

Range("E28").Value = Item

Range("F28").Value = Cost

Range("G28").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C6").Value >= FDDate Then

Item = Range("A6").Value

Cost = Range("B6").Value

Day = Range("C6").Value

Worksheets("Dashboard").Select

Range("E29").Value = Item

Range("F29").Value = Cost

Range("G29").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C7").Value >= FDDate Then

Item = Range("A7").Value

Cost = Range("B7").Value

Day = Range("C7").Value

Worksheets("Dashboard").Select

Range("E30").Value = Item

Range("F31").Value = Cost

Range("G32").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C8").Value >= FDDate Then

Item = Range("A8").Value

Cost = Range("B8").Value

Day = Range("C8").Value

Worksheets("Dashboard").Select

Range("E33").Value = Item

Range("F34").Value = Cost

Range("G35").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C9").Value >= FDDate Then

Item = Range("A9").Value

Cost = Range("B9").Value

Day = Range("C9").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

Worksheets("Monthly Costs and Income").Activate

End If

End If

End If

End If

End If

End If

End If

If Range("C4").Value >= FDDate Then

Item = Range("A4").Value

Cost = Range("B4").Value

Day = Range("C4").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C5").Value >= FDDate Then

Item = Range("A5").Value

Cost = Range("B5").Value

Day = Range("C5").Value

Worksheets("Dashboard").Select

Range("E27").Value = Item

Range("F27").Value = Cost

Range("G27").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C6").Value >= FDDate Then

Item = Range("A6").Value

Cost = Range("B6").Value

Day = Range("C6").Value

Worksheets("Dashboard").Select

Range("E28").Value = Item

Range("F28").Value = Cost

Range("G28").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C7").Value >= FDDate Then

Item = Range("A7").Value

Cost = Range("B7").Value

Day = Range("C7").Value

Worksheets("Dashboard").Select

Range("E29").Value = Item

Range("F29").Value = Cost

Range("G29").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C8").Value >= FDDate Then

Item = Range("A8").Value

Cost = Range("B8").Value

Day = Range("C8").Value

Worksheets("Dashboard").Select

Range("E30").Value = Item

Range("F30").Value = Cost

Range("G30").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C9").Value >= FDDate Then

Item = Range("A9").Value

Cost = Range("B9").Value

Day = Range("C9").Value

Worksheets("Dashboard").Select

Range("E31").Value = Item

Range("F32").Value = Cost

Range("G33").Value = Day

Worksheets("Monthly Costs and Income").Activate

End If

End If

End If

End If

End If

End If

If Range("C5").Value >= FDDate Then

Item = Range("A5").Value

Cost = Range("B5").Value

Day = Range("C5").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

Worksheets("Monthly Costs and Income").Activate ''SubMoExpenses = 'SubMoExpenses + Range("B4").Value

If Range("C6").Value >= FDDate Then

Item = Range("A6").Value

Cost = Range("B6").Value

Day = Range("C6").Value

Worksheets("Dashboard").Select

Range("E27").Value = Item

Range("F27").Value = Cost

Range("G27").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C7").Value >= FDDate

Item = Range("A7").Value

Cost = Range("B7").Value

Day = Range("C7").Value

Worksheets("Dashboard").Select

Range("E28").Value = Item

Range("F28").Value = Cost

Range("G28").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C8").Value >= FDDate

Item = Range("A8").Value

Cost = Range("B8").Value

Day = Range("C8").Value

Worksheets("Dashboard").Select

Range("E29").Value = Item

Range("F29").Value = Cost

Range("G29").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C9").Value >= FDDate Then

Item = Range("A9").Value

Cost = Range("B9").Value

Day = Range("C9").Value

Worksheets("Dashboard").Select

Range("E30").Value = Item

Range("F30").Value = Cost

Range("G30").Value = Day

Worksheets("Monthly Costs and Income").Activate

End If

End If

End If

End If

End If

If Range("C6").Value >= FDDate Then

Item = Range("A6").Value

Cost = Range("B6").Value

Day = Range("C6").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C7").Value >= FDDate Then

Item = Range("A7").Value

Cost = Range("B7").Value

Day = Range("C7").Value

Worksheets("Dashboard").Select

Range("E27").Value = Item

Range("F27").Value = Cost

Range("G27").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C8").Value >= FDDate Then

Item = Range("A8").Value

Cost = Range("B8").Value

Day = Range("C8").Value

Worksheets("Dashboard").Select

Range("E28").Value = Item

Range("F28").Value = Cost

Range("G28").Value = Day

Worksheets("Monthly Costs and Income").Activate

''SubMoExpenses = 'SubMoExpenses + Range("B6").Value

If Range("C9").Value >= FDDate Then

Item = Range("A9").Value

Cost = Range("B9").Value

Day = Range("C9").Value

Worksheets("Dashboard").Select

Range("E29").Value = Item

Range("F29").Value = Cost

Range("G29").Value = Day

Worksheets("Monthly Costs and Income").Activate

End If

End If

End If

End If

If Range("C7").Value >= FDDate Then

Item = Range("A7").Value

Cost = Range("B7").Value

Day = Range("C7").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C8").Value >= FDDate Then 'And Range("C5").Value <= FDDate Then

''SubMoExpenses = 'SubMoExpenses + Range("B5").Value

Item = Range("A8").Value

Cost = Range("B8").Value

Day = Range("C8").Value

Worksheets("Dashboard").Select

Range("E27").Value = Item

Range("F27").Value = Cost

Range("G27").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C9").Value >= FDDate Then

Item = Range("A9").Value

Cost = Range("B9").Value

Day = Range("C9").Value

Worksheets("Dashboard").Select

Range("E28").Value = Item

Range("F28").Value = Cost

Range("G28").Value = Day

Worksheets("Monthly Costs and Income").Activate

End If

End If

End If

End If

If Range("C8").Value >= FDDate Then

Item = Range("A8").Value

Cost = Range("B8").Value

Day = Range("C8").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

Worksheets("Monthly Costs and Income").Activate

If Range("C9").Value >= FDDate Then 'And Range("C5").Value <= FDDate Then

Item = Range("A9").Value

Cost = Range("B9").Value

Day = Range("C9").Value

Worksheets("Dashboard").Select

Range("E27").Value = Item

Range("F27").Value = Cost

Range("G27").Value = Day

Worksheets("Monthly Costs and Income").Activate

End If

End If

If Range("C9").Value >= FDDate Then

Item = Range("A9").Value

Cost = Range("B9").Value

Day = Range("C9").Value

Worksheets("Dashboard").Select

Range("E26").Value = Item

Range("F26").Value = Cost

Range("G26").Value = Day

ElseIf Range("C2") < FDDate Then

Range("E26").Value = ""

Range("F26").Value = ""

Range("G26").Value = ""

End If

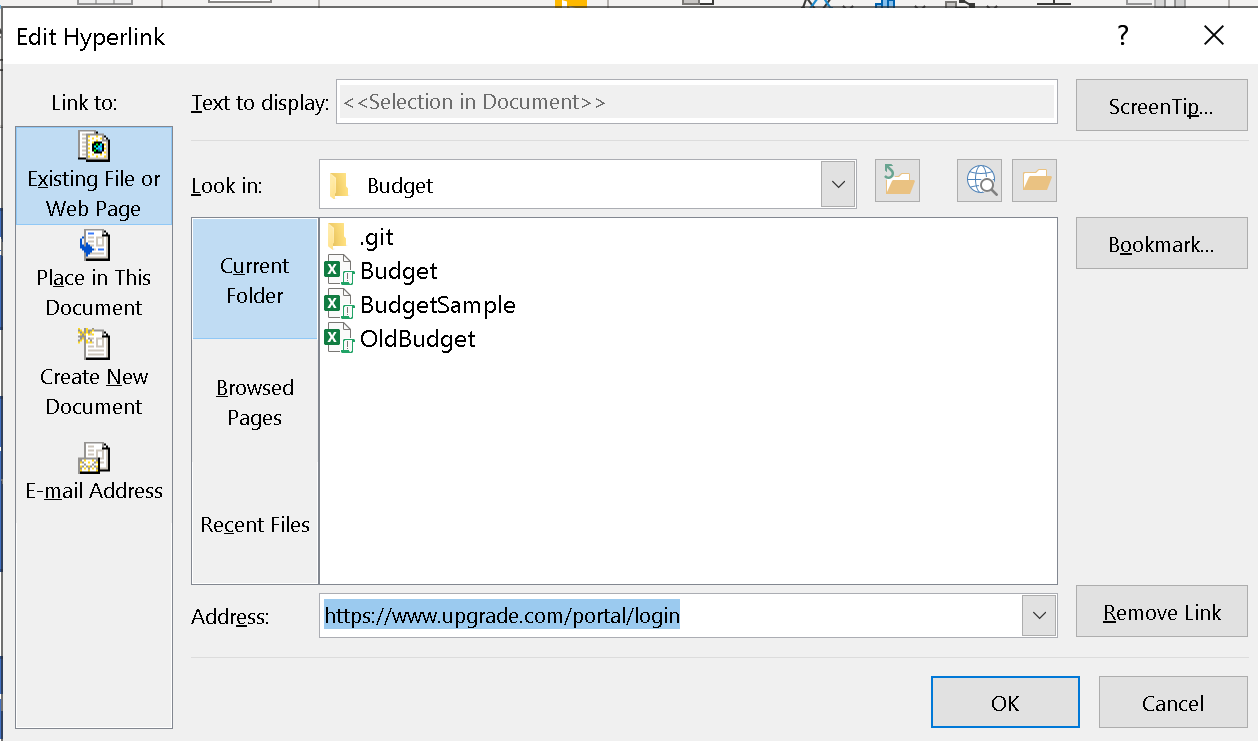
Worksheets("Dashboard").Activate ‘return to the Dashboard worksheet

End Sub

1. **Hyperlinks Worksheet**

This worksheet has four hyperlinks on it: ***USAA Login, Upgrade, and Link to Download Files****.* To add a hyperlink go to the worksheet and choose *Insert, Illustrations, Shapes* and choose the shape you want. Position the shape on the form, edit the text and link the hyperlink.

To link the hyperlink to a web address, right click on the image and choose *edit link*. This brings up a form to add the link as shown in the following figure.



Note that there is already a URL in the Address: field. This is where you paste the link to the web site. The user may also remove the link or add a screen tip.