

Executive Summary

BYU Arts Production

This project was done for the Brigham Young University Arts Production Department. BYU Arts Production is a segment of BYU which handles all of the various elements of shows and productions put on around campus. These shows come from different categories such as music, theatre, dance, touring groups, and outside performers. The business office for the BYU Arts Production Department will be using this report the most. This office handles all of the revenue and expenses associated with putting on these performances.

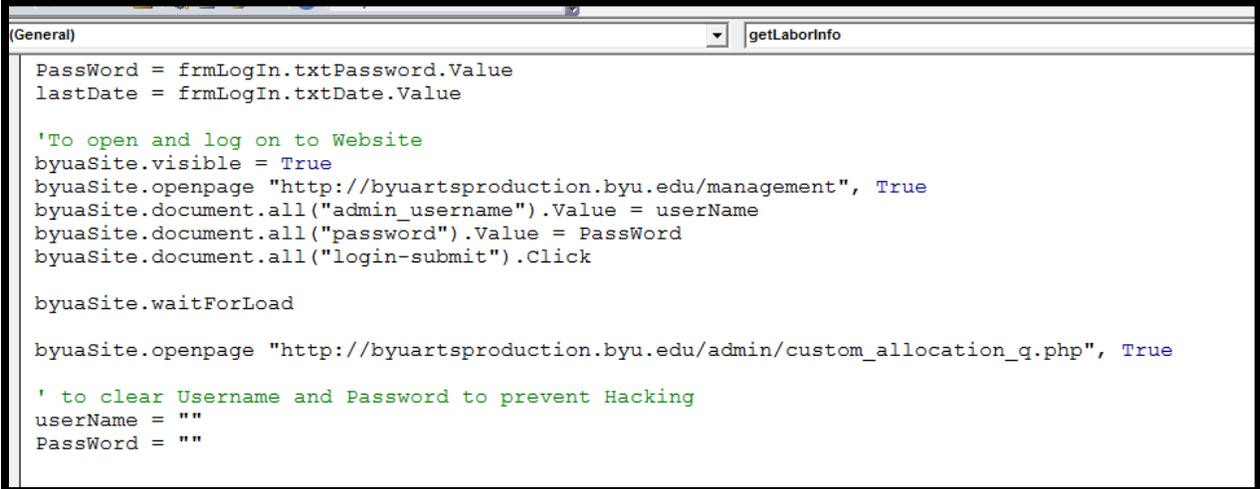
The Report

This report pulls information from the BYU Arts Production website, byuartsproduction.byu.edu, and formats it to provide useful information to department supervisors and decision makers. In particular, this report summarizes the total amount spent, total hours worked, number of students, and number of shows per semester for student labor. It then sorts this data into the various labor areas under BYU Arts Production.

Implementation

The first step in this project was to connect to the BYU Arts website and extract the necessary data. In order to do this I made a user form that captured the user name and password necessary to get onto the site. This user form also captures a start date for the data, designated by the user. Because this report is only to be run for one semester I give the user guidance as to what date they should enter on the form.

Next the macro makes a connection to the website using the New Agent data type. The code navigates through the website as seen below. It selects the appropriate values in a query and then exports the data to a worksheet named "LaborData".



```
(General) getLaborInfo
PassWord = frmLogIn.txtPassword.Value
lastDate = frmLogIn.txtDate.Value

'To open and log on to Website
byuaSite.visible = True
byuaSite.openpage "http://byuartsproduction.byu.edu/management", True
byuaSite.document.all("admin_username").Value = userName
byuaSite.document.all("password").Value = PassWord
byuaSite.document.all("login-submit").Click

byuaSite.waitForLoad

byuaSite.openpage "http://byuartsproduction.byu.edu/admin/custom_allocation_q.php", True

' to clear Username and Password to prevent Hacking
userName = ""
PassWord = ""
```

Once the data has been imported to the excel workbook a message box alerts the user that it is ready for the next step. There is another button that the user pushes to format the data and build the report. This step starts by deleting rows and adding columns and totals to the raw data. It then determines which semester it is looking at by evaluating the dates worked. This helps it know where to publish the data in the end.

The next part is to split the data onto separate tabs based on which labor area the student worked for. This was the most tricky part for me because I needed to make the code able to look at each for and determine which worksheet it should be added to. It also determined if it needed to add a worksheet with the new title if it did not already exist. This part of the code slows down the macro the most because it is normally sorting through over 15,000 lines of data. Below is a screenshot of some of the code needed to execute this section of the macro.

```

For r = 2 To endrow
Application.ScreenUpdating = False
area = sh.Cells(r, 3).Value
For Each ws In ThisWorkbook.Sheets
If StrComp(ws.Name, area, vbTextCompare) = 0 Then
sheetExist = True
Exit For
End If
Next
If sheetExist Then
CopyRow r, sh, ws
Else
Set ws = Worksheets.Add
ws.Name = area
CopyRow r, sh, ws
End If
Reset sheetExist

```

This code shows that it first looks to see if a worksheet with the area title already exists. If it does it simply copies the line over to that workbook. If it does not already exist, it makes the new workbook and gives it the appropriate title and then copies the line over. It loops through each line in the raw data until it comes to the end row.

The next part totals all of the different elements desired are reports them to the appropriate spot on the summary worksheet. It loops through each different labor area and uses a dictionary object to count the distinct values of show name and employee name. Below is a screenshot of the output that it produces. The line produced from this macro is found on the "Winter 2014" row. This is reproduced for each labor area. Over time the report will keep filling in the subsequent lines which will give supervisors an idea of the trends in their student labor.

Area	All		Winter	YYYY-01-01		Build Report		Labor Area	Arts Assistant
			Sp/Su	YYYY-05-01					28
			Fall	YYYY-09-01					
Semester	Total Cost	Total Hours	Employees	Shows	Avg Hrs/Employee	Avg wage rate/hour		Semester	To
Sp/Su 2010		313.74	62	20	5.06			Sp/Su 2010	
Fall 2010		18,292.76	133	86	137.54			Fall 2010	
Winter 2011		17,752.40	126	108	140.89			Winter 2011	
Sp/Su 2011		14,615.53	114	63	128.21			Sp/Su 2011	
Fall 2011		20,333.78	149	101	136.47			Fall 2011	
Winter 2012		22,468.41	143	131	157.12			Winter 2012	
Sp/Su 2012		14,160.99	122	73	116.07			Sp/Su 2012	
Fall 2012		17,780.62	136	95	130.74			Fall 2012	
Winter 2013		16,707.11	125	108	133.66			Winter 2013	
Sp/Su 2013		10,437.41	95	76	109.87			Sp/Su 2013	
Fall 2013		14,932.84	103	104	144.98			Fall 2013	
Winter 2014	\$129,566.23	14,351.04	104	110	137.99	\$9.03		Winter 2014	\$-
Sp/Su 2014								Sp/Su 2014	
Fall 2014								Fall 2014	

Finally, the macro ends by deleting the extra tabs created during the process in order to keep the workbook running quickly and ready to go for the next semester. It does this by finding any worksheet with a name less than three characters long and deleting it. In the end, the user is notified by another message box that the report has been successfully built!

Learning Concepts

I felt that I learned a great deal about programming in excel through this project. Before I began, I felt I had a pretty good understanding of most concepts. While that was mostly true, I found there were many weak spots I was able to overcome. One of those weak spots was with connecting to a website in order to obtain data. As I worked through the issues I learned it was not as difficult as I had imagined in the beginning. One of my biggest take-aways from this project was the use of loops in almost all of my coding. Loops are probably one of the most useful tools programming has because it allows you to write fewer lines for a lot of output. I especially found this helpful in the segment where my code calculates the values for each of the different departments. There were not any problems I was unable to resolve after speaking with the professor or consulting the internet.

Assistance

The biggest source of assistance I received was from the professor. He helped me work through code and find errors after I had made a concentrated effort. I also received some help in connecting to the website from another project that had been done for my business office before, although I made significant changes to the other macro.