Cameron Smith

Job Aging Commission Report

MBA 614 - Section 1

Cameron Smith

MBA 614

Job Aging Commission Report Write-up

2.1 Your write-up must be formatted as a PDF document.

2.2 Executive summary of the project. This should contain a description of the business along with an overview of the system you built

Executive Summary

The company I built this spreadsheet for is a physical therapy company. They provide staffing to physical therapy facilities around the country, mostly in nursing homes and things like them. I worked closely with David, who is the recruiting manager for the company. He had expanded on previous recruiting efforts, including hiring more recruiters and starting new facilities. With all the expansion, there were a lot of records to keep track of. Lots of jobs in progress, and lots of commissions to keep track of. In addition, he had just opened up a contingency recruiting business running within he organization to provide services to other companies. He wanted me to design a sheet to make record-keeping simple. I honestly don't know how he kept track of it all before, but apparently it was a mess.

I was in almost constant contact with David for a good chunk of the design phase. He was going to be the primary user, so it needed to make sense to him. There was a lot of data for each job to keep track of, and we worked together to make sure the display made sense. He wanted it to be robust enough to be able to add new recruiters or job markets without having to call me in again. Also, in case someone else in the company got their hands on it, he wanted it to have as few actual formulas in cells as possible.

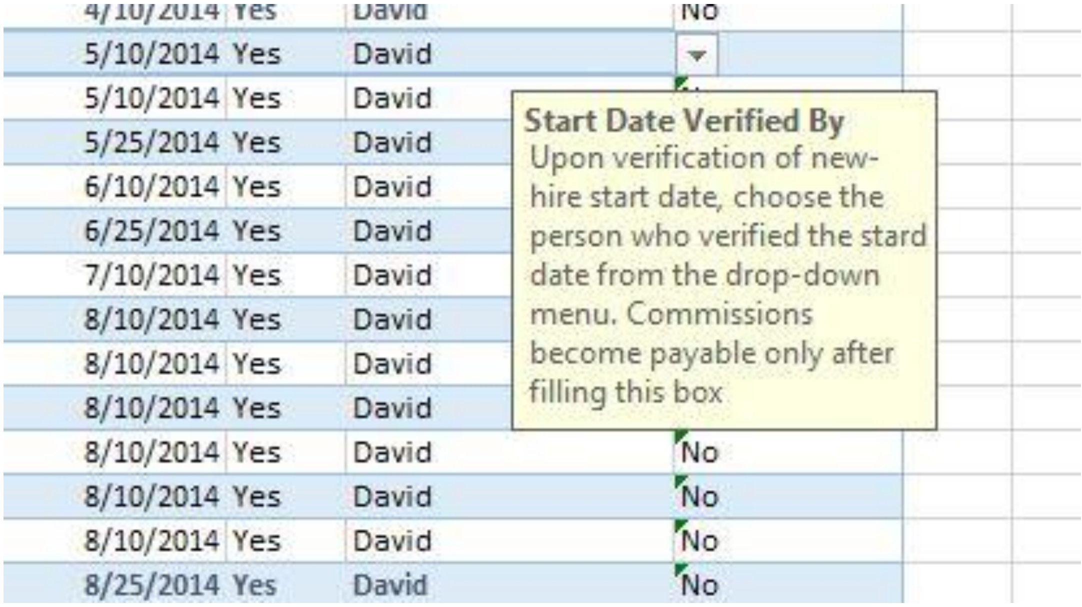
We came up with a good solution. The sheet automatically calculates all fields necessary through a combination of userforms and event activated procedures. There are no formulas in cells, and it keeps track of all jobs in an efficient manner, leaving room for additional growth in the company. David seems very happy with it, and says he uses it every day.

Implementation Documentation

David gave me Pay Scale sheet to work with, which contains all the wages to shoot for in each market the company deals in based on occupation and years of experience. I had to use that information later.

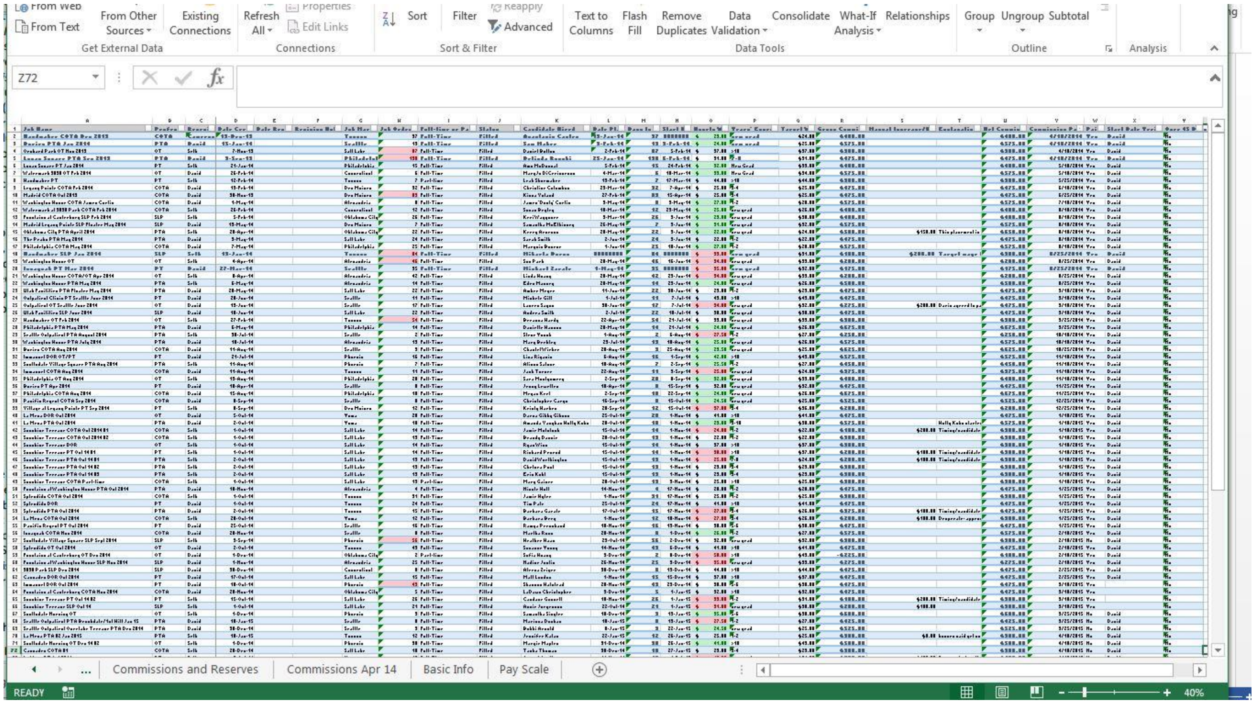
We put considerable time into setting things up. I created the Basic Info sheet to hold basic information. That's where David could go to add new recruiters or adjust base commissions, etc. I used dynamic named ranges to make things easier on myself later.

The Internal Jobs table is the part that David uses most often, so that's where we went next. We put together the fomat of the table, with drop-down boxes and data validation to provide guidance on usage. There ended up being 25 different pieces of information he wanted to store for each candidate.

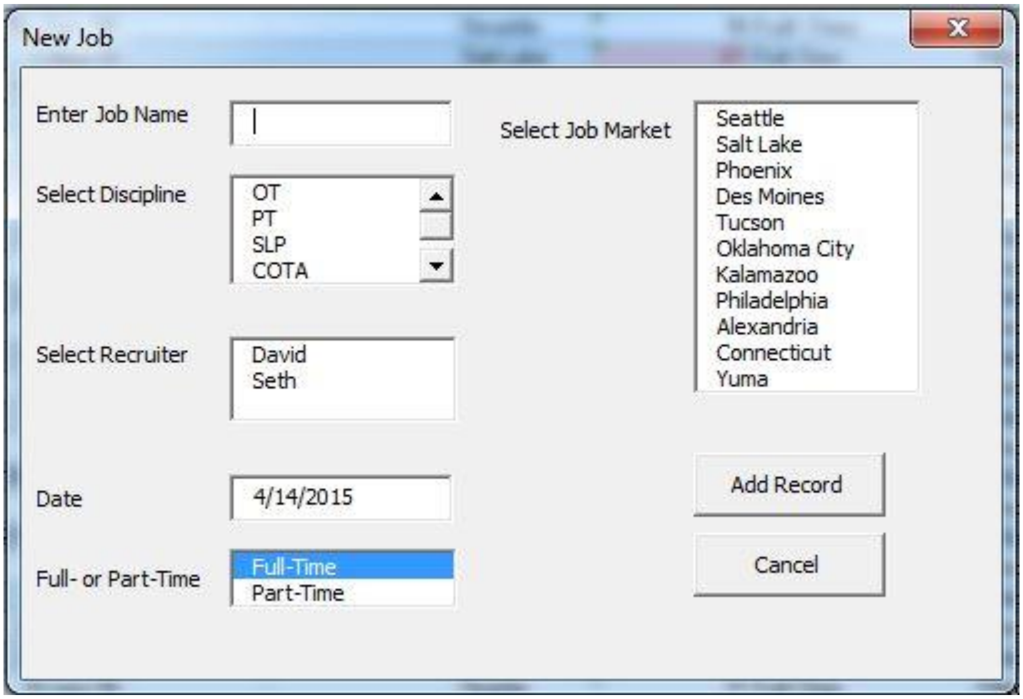


4/10/2014	Yes	David	No
5/10/2014	Yes	David	
5/10/2014	Yes	David	
5/25/2014	Yes	David	
6/10/2014	Yes	David	
6/25/2014	Yes	David	
7/10/2014	Yes	David	
8/10/2014	Yes	David	
8/10/2014	Yes	David	
8/10/2014	Yes	David	
8/10/2014	Yes	David	No
8/10/2014	Yes	David	No
8/10/2014	Yes	David	No
8/25/2014	Yes	David	No

Start Date Verified By
Upon verification of new-hire start date, choose the person who verified the stard date from the drop-down menu. Commissions become payable only after filling this box



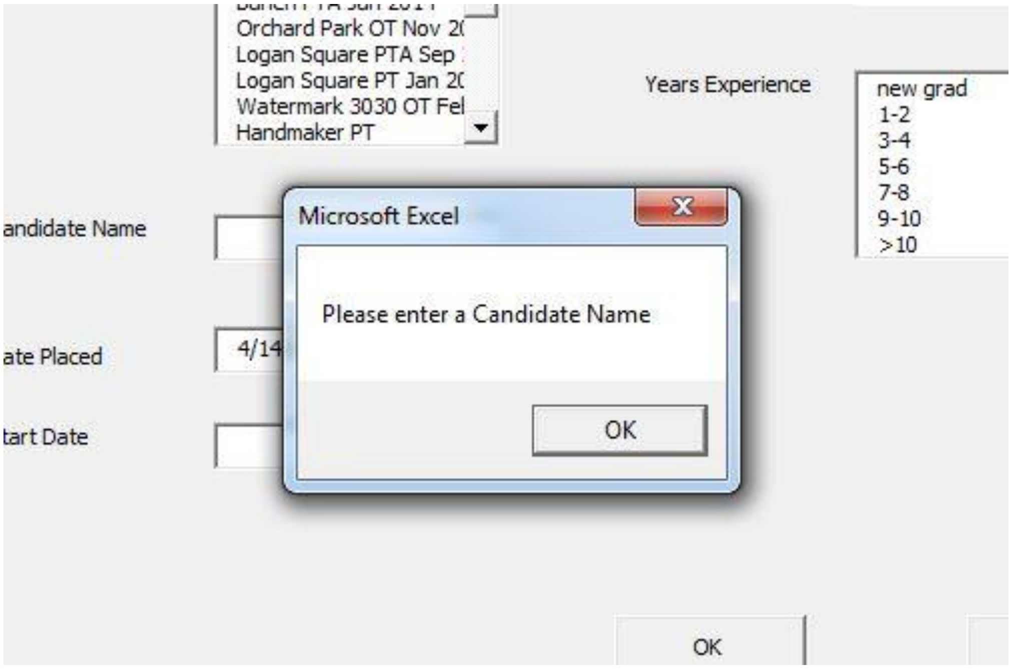
To make things as simple as possible, and to limit input errors, I made two userforms. The first creates a new job in the system when a position opens up somewhere.



Date 4/14/2015

Full- or Part-Time Full-Time

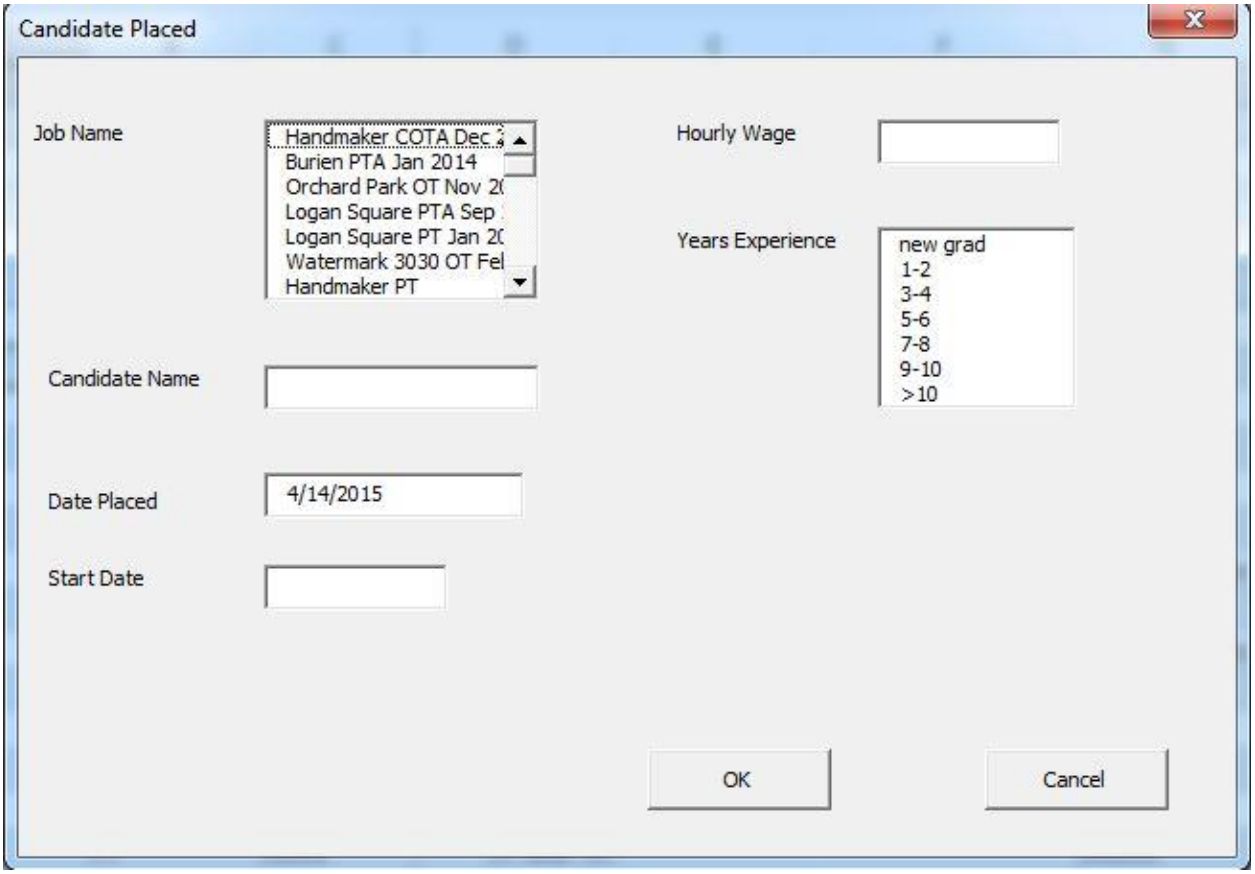
The fields fill automatically based on the Basic Info sheet. When Add Record is clicked, it checks the fields to make sure they've been filled. If fields are missing it lets the user know that they need to fill it.



Please enter a Candidate Name

Then it creates a new row on the table, and adds the info in the correct cells in that row.

The second form is used when a candidate has been placed in one of the open jobs.



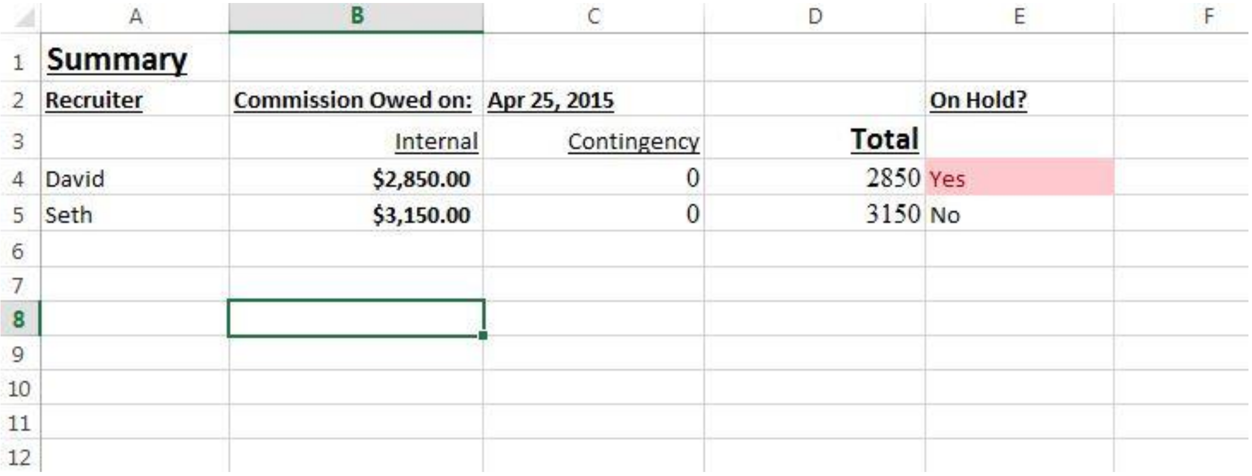
Date Placed 4/14/2015

This form populated a listbox with all the job names in the table. Then it calculates the ideal wage for that position, in that market, at that experience level based on the Pay Scale sheet. And of course it populates the appropriate cells in the table.

Originally, I used some formulas in several of the columns. David was worried about someone somehow accidentally overwriting those formulas and having to call me in again to fix it, so he asked if I would convert as much as possible to VBA. So now, there are no formulas. Several columns refresh automatically whenever you open that sheet, and others are refreshed whenever a new job is created or filled. There is also some conditional formatting included in those subs to highlight certain cells used for making decisions.

The Contingency Jobs table is very similar, but simpler than the Internal Jobs table, so I won't go into detail on that.

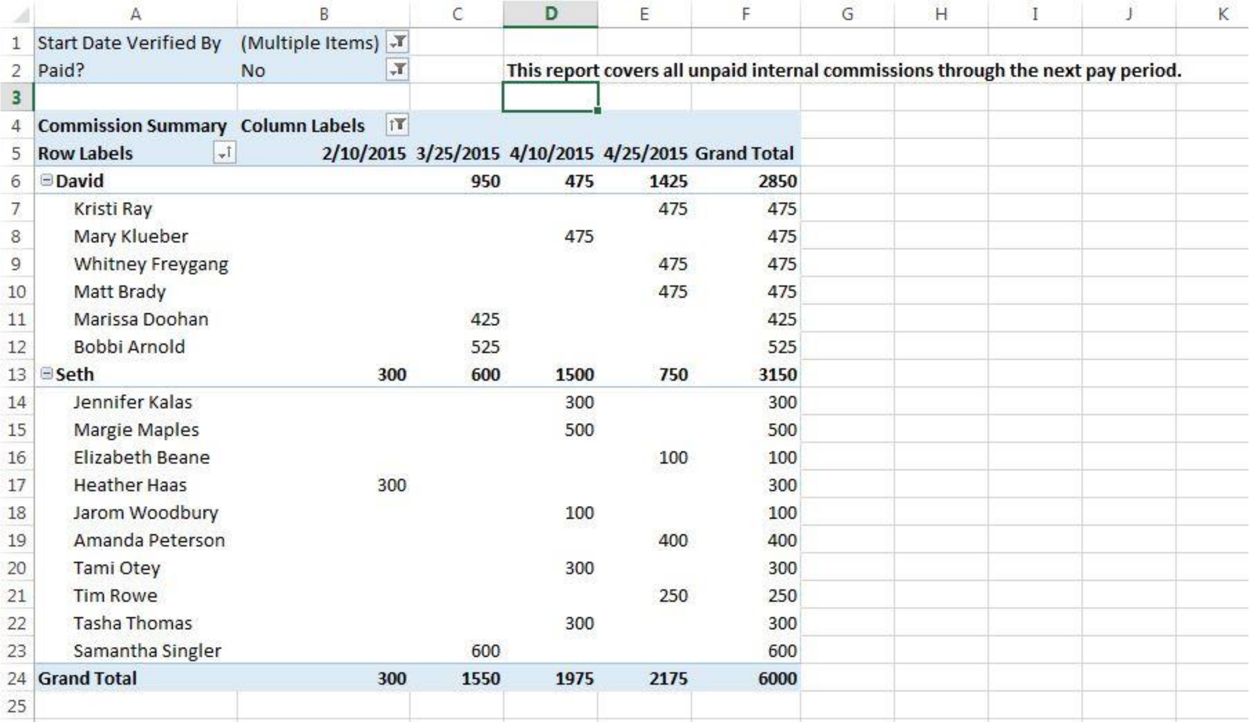
One of the most important uses of this spreadsheet is to keep track of commissions for recruiters, so David wanted to have a sheet or two in place for easy access and reporting. So I made two sheets. One is an at-a-glance summary of each recruiter's commission owed in the next pay period (paydays are on the 10th and 25th of each month).



	A	B	C	D	E	F
1	**Summary**					
2	Recruiter	Commission Owed on:	Apr 25, 2015		On Hold?	
3		Internal	Contingency	**Total**		
4	David	**$2,850.00**	0	2850	Yes	
5	Seth	**$3,150.00**	0	3150	No	
6						
7						
8						
9						
10						
11						
12						

Again, there are a couple of cells that are populated and formatted according to certain criteria, which eliminates the need for David to look through all of the table data, which saves time, which saves money.

I spent some time with David to teach him how to use pivot tables, which he seems to enjoy, but he still isn't super comfortable with it yet. So, with his input, I wrote a sub to create one that he would use often. The sub creates a new sheet named with today's date, and in that sheet it inserts and formats a pivot table that displays all of the unpaid commissions due the next payday expanded to show the dollar amounts for each recruiter and each candidate placed.



	A	B	C	D	E	F
1	Start Date Verified By	(Multiple Items)				
2	Paid?	No		**This report covers all unpaid internal commissions through the next pay period.**		
3						
4	**Commission Summary**	**Column Labels**				
5	**Row Labels**	**2/10/2015**	**3/25/2015**	**4/10/2015**	**4/25/2015**	**Grand Total**
6	**⊟David**		**950**	**475**	**1425**	**2850**
7	Kristi Ray				475	475
8	Mary Klueber			475		475
9	Whitney Freygang				475	475
10	Matt Brady				475	475
11	Marissa Doohan		425			425
12	Bobbi Arnold		525			525
13	**⊟Seth**	**300**	**600**	**1500**	**750**	**3150**
14	Jennifer Kalas			300		300
15	Margie Maples			500		500
16	Elizabeth Beane				100	100
17	Heather Haas	300				300
18	Jarom Woodbury			100		100
19	Amanda Peterson				400	400
20	Tami Otey			300		300
21	Tim Rowe				250	250
22	Tasha Thomas			300		300
23	Samantha Singler		600			600
24	**Grand Total**	**300**	**1550**	**1975**	**2175**	**6000**
25						

At the end of it, I added the new ribbon buttons to make it look more professional.

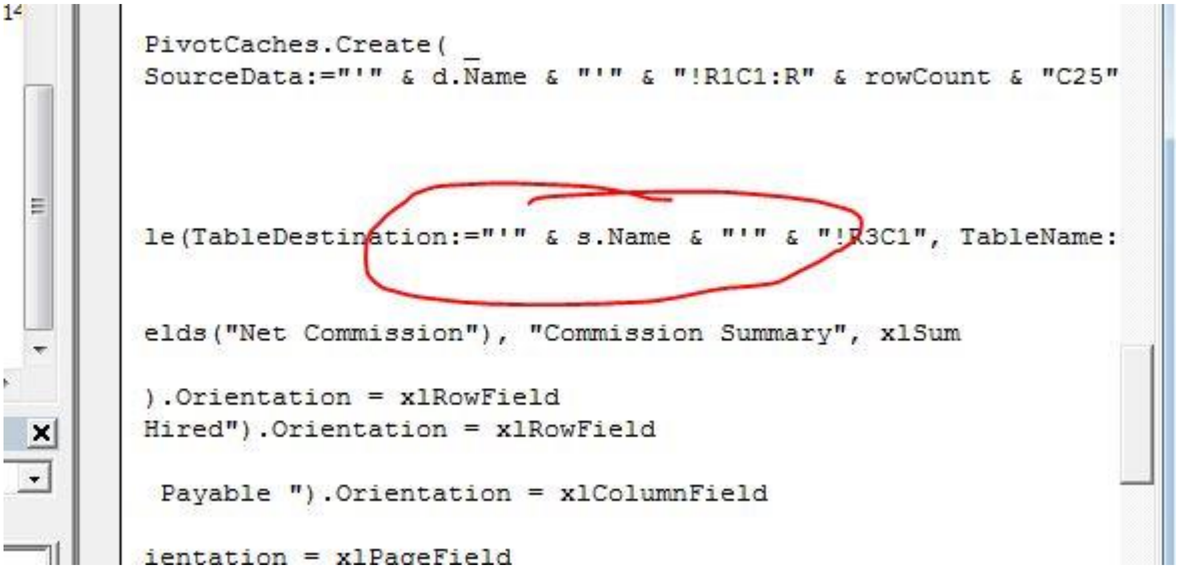
All of the code for this project is pretty robust. It will still work if you add or remove recruiters or markets, or if you move columns around on the tables.

Learning and Difficulties

I didn’t have to give up on anything I wanted to do, but there were quite a few things I found difficult on this project, so I’ll try to be brief with them.

Dynamic named ranges were tough. Converting big, hairy megaformulas into VBA code was a tedious challenge, but beneficial I think. Some of those formulas were more complicated than I expected them to be. For example, the Job Age formula seems like it should be pretty straightforward. Just today minus the day it was created. But it was actually a set of nested if statements based on other factors like whether or not a candidate had been placed or the job had been revised.

Perhaps the most challenging or time-consuming thing was working with tables. There is a whole new set of vocabulary to learn when you want to work directly with them. For example, I wanted to make the code good enough to be able to adjust the order of columns if needed, so I had to work on finding or filling info based on column headings rather than a basic offset. Another hard thing was the pivot table. I copied, pasted, and modified the code we used in class to do it, but it didn’t work for some reason. I spent a couple of hours on google troubleshooting before I discovered that I needed to add a pair of single quotes. That was really frustrating.



```
PivotCaches.Create( _
SourceData:="'" & d.Name & "'" & "!R1C1:R" & rowCount & "C25"



le(TableDestination:="'" & s.Name & "'" & "!R3C1", TableName:


elds("Net Commission"), "Commission Summary", xlSum

).Orientation = xlRowField
Hired").Orientation = xlRowField

 Payable ").Orientation = xlColumnField

ientation = xlPageField
```

Then there were the dates. Figuring out how to calculate the next payday was a pain, but ended up being one of the more useful parts of the project to David.

Assistance

I didn’t have any help from another person on this project. I spent a substantial amount of time on google learning syntax, and I took small snippets of code from forums online, but those almost never looked much like the original once I had modified them for my purposes.

In the Future

I’m happy with the spreadsheet as it is, but there is some functionality I hope to add in the future. Perhaps another userform to add a new market to further lessen the chance of user error in the sheet is one example. Luckily, this David guy I’ve been talking about is my dad, so I’m in pretty consistent contact with him about the whole thing and I can make edits to it as I feel like it.