

VBA Project – KSL Home Scraper

Ryan Durrant
IS 520

Executive Summary

The Business Overview

Durrant Property Investments (DPI) is my father's company and has been in existence for approximately 22 years. DPI's core business model is to invest in real estate in the Provo, UT area, specifically in single family homes and apartment complexes. These properties are then rented out to BYU and UVU students. DPI has employed students for years to help manage the properties, collect rent, perform cleaning checks, and upkeep the landscaping of the homes and complexes.

For years DPI has found deals through building relationships with other real estate professionals in the community. However, a wide variety of past deals have come from scouring the newspaper, searching for "For Sale" signs, and using the internet. This last option is the one that is the most efficient, has the greatest amount of information, and is the best way to scan a lot of potential investments anytime and anywhere. The problem is that it is both time consuming and difficult to first narrow down the options to those homes that meet your requirements and then to open each link to see if the homes meet your criteria. It is also annoying because many of these properties have been posted multiple times. It is like trying to find a needle in the haystack.

The System Overview

My project solves these problems directly, providing a fast, organized, and painless way to rip all of the home data off KSL and then analyze all of the listed properties to find the best

deals. Using a UserForm, that takes the 15 different parameters to create the right URL to find all of the homes that meet the criteria. The web scraper then opens every link of all the property listings on every single page. For each of the specific properties, my tool scrapes off the address, price, property type, seller type, bedrooms, bathrooms, year built, square footage, acreage, garage information, basement information, and the description!

The system then pastes all of the information into the “Home Data” tab with hyperlinks to all of the KSL listings. After pasting all of the information for the homes that passed the initial filter, they are then sorted and analyzed on the “Dashboard” tab. Using conditional formatting the top five cheapest homes by price per square foot are highlighted in green. These top 5 homes are also ranked by price per acre and the conditional formatting of a green checkmark, yellow exclamation point, or red x are shown next to the price/acre, indicating the best deals. Lastly, if the home is being sold by the owner, the last cell in each of the top rows is highlighted in blue to indicate that there is potential to negotiate a lower price due to the fact that there are no agent commissions for the seller.

Implementation Documentation

To implement this solution, I began by creating a sub procedure entitled “getHouseData” which was used to declare variables, name worksheets, assign labels to different columns on the “Home Data” tab, and ultimately open every single page URL with the necessary variables inserted. Then the individual home URL’s are opened, ripping all of the needed data, pasting the values into the cells, and finally sorting everything. At the beginning

of this sub I call the UserForm "frmKSL" to initiate. Upon initiation, the form pulls in parts of the data that is held on a hidden sheet

and that sheet is then updated each time the form's "OK" button is pushed.

Zip code, bedrooms, bathrooms, home type, and square feet are the entries

that are pulled in and these entries were chosen due to the fact that they

are the most common parameters to search for houses on. I purposely left

the other parameters to be filled out each time because they are the ones that my Dad and I change as we explore different options.

The screenshot shows a window titled "KSL Search Parameters" with a close button (X) in the top right corner. The form contains the following elements:

- Zip Code: Text box containing "84604"
- Bedrooms: Dropdown menu with "3" selected
- Bathrooms: Dropdown menu with "2+" selected
- Home Type: Dropdown menu with "Homes For Sale" selected
- Square Feet: Dropdown menu with "1000+" selected
- Acres: Dropdown menu (empty)
- Price Range: Two dropdown menus labeled "From" and "To" (empty)
- Year Built: Two dropdown menus labeled "From" and "To" (empty)
- Seller Type: Two radio buttons, "By Owner" and "By Agent" (both unselected)
- Zipcode: Two radio buttons, "Exact" and "Within 10 miles" (both unselected)
- Buttons: "Cancel" and "OK" buttons at the bottom right.

That form has different data validation "If Statements" to ensure that there are 5-digits in the place for the zip code and that the upper limit price is not "<\$50,000." There are combo boxes for most of the parameters along with four option buttons and a text box. After the user inputs their chosen filters, the form then converts their entries to be readable by KSL's servers.

It was extremely time consuming and difficult to alter and input the variables into the URL depending on the users' selections. It required testing almost all of the possible options on

KSL's site and then comparing KSL's URL to what my form was producing and altering the variables as needed. Lastly, the form writes the user's input onto the hidden sheet to be

accessed next time the form is opened and for the variable assignments in the URL. That sub

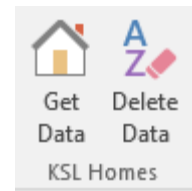
ends with two calls to “boldHeadings,” which bolds the labels on the data tab, and the “dashboard” sub.

The “dashboard” sub begins by assigning the first row’s values to be the labels for the columns and also places borders around those labels. The sub then inputs data onto the dashboard from the “Home Data” tab using a Do Loop that runs until there are no more addresses to analyze. It begins by pulling in the same hyperlinks from the first sheet to allow the user to click the link, see pictures of the home, and then contact the seller through KSL’s messaging. Next, it calculates the price per square foot and price per acre for each home which can then be compared with the data set’s averages of those ratios. The sub also shows the bedroom-to-bathroom ratio and whether an agent or the owner is selling the property. Before the sub calls the “FormattingDash” sub procedure it sorts the properties from the lowest to highest price per square foot.

After the data is sorted, the “FormattingDash” sub centers it, makes the headings bold, increases the font size, and wraps the text in the address column to allow users to read the entire address with a fairly small sized column. After formatting the entire dashboard, the sub calls “formatTop5” which highlights and shows the top 5 best properties on the market that meet the user’s requirements. This sub highlights the top properties by price per square foot, those that are being sold by the owner, the top ones by price per acre, and finally adds some spacing around the block of data and AutoFits the columns to make the dashboard easier to read.

Address	Price	Price/Ft	Average Price/Ft	Price/Acre	Average Price/Acre	Bedroom to Bathroom Ratio	For Sale By Owner
616 S. 400 W. Provo, UT 84601	\$117,900	\$79	\$98	✔ \$421,071	\$819,534	3 to 2	Not specified
1934 W 500 N Provo, UT 84601	\$128,800	\$86	\$98	✘ \$1,431,111	\$819,534	3 to 2	No
GRANT \$\$ AVAILABLE! 1934 W 500 N - NO MONEY DOWN! Utah Grants - Down Payment Assistance Provo, UT 84601	\$128,800	\$86	\$98	✘ \$1,431,111	\$819,534	3 to 2	No
1472 E 520 S Provo, UT 84606	\$184,900	\$97	\$98	⚠ \$1,027,222	\$819,534	3 to 2	No
1403 S 570 W	\$175,000	\$129	\$98	✔ \$388.889	\$819,534	3 to 2	No

The tool operates through two buttons and a new Excel tab that I created on the top ribbon. The tab is titled, “KSL Data” and it is located just after the “Developer” tab. In that tab there are two buttons that I designed called “Get Data” and “Delete Data.” “Get Data” runs the “getHouseData” macro that opens the UserForm and begins all the subsequent subs. The “Delete Data” button runs a simple macro that wipes the first two sheets clean to then add in the most up to date data from the web.



Learning and Conceptual Difficulties

I learned so much through my battle with this project, and I say battle because it felt like each time I would fix a bug I would find two more, and in fixing those I would accidentally create another. After well over 20 hours of work, I finally perfected my solution. I was able to accomplish this project with the help of online VBA documentation, examples, and a lot of critical thinking. One of the most difficult concepts I applied was web scraping the data and ensuring that my searches would work on all of the different pages, properties, and listings per page. Another difficult concept I applied was that of a complex UserForm with hundreds of

different combinations, the string manipulations to the user's selections, and the changes to the URL that accompanied them.

The web scraper tool I created was initially difficult because I had to open one page with all of the listings for that page and then I had to open another page for each of the twelve listings. The opening search page obviously had very different source code that I needed to parse through, as did the individual property pages. I used an array to store the different URL endings for each property which I then used to open all of those property URL's to rip data from. Once I had the data, I wrote it to the first sheet with counters and Do Loops. When one page was done, I would open up the next and repeat the process. Because of the different amounts of pages and properties with each and every search and every change in time and date, I researched and used On Error GoTo methods to account for every possible situation of page counts.

After I got the initial program to work and to pull all of the data I wanted, I went and spoke to Dr. Allen, who suggested I create a UserForm to allow the user more freedom to select the exact data they wanted. After battling with how many options I should give the user, I ended up creating 15 different parameters the user could choose from. As stated earlier, when the form is initialized it inserts the most common parameters and leaves the others blank and this is done purposely to allow easy manipulation of the more commonly altered options. It was difficult to convert the users' answers into strings that were understood when combined with the URL. There were many small changes that needed to be taken into consideration. For example, when the user selects "<.10" in the acre's combo box the URL needs to read, "lt_10,"

whereas if it is “.10,” then it reads, “gte_10.” There were many string manipulations like this that needed to be tracked and updated for each new search performed.

The entire project was very fulfilling due to its challenging nature and for the small victories along the way that led up to this point of having a completed product. I used past projects, class examples, and a lot of Google to achieve success. My goal with this project is to put it up online on a site that I will create for myself to begin selling the tool to the people of Utah. I see a lot of value being delivered through my creation and I believe that people would be willing to pay for it.

Assistance

I did not receive assistance from anyone else. I used the internet’s resources, past projects, and class examples to figure out the many problems I ran into.