SpendMyCents.com in Excel

Executive Summary

About a year ago a friend and I created www.spendmycents.com, a reverse product search that searches amazon based on price. A picture of the website is below.

I struggled a bit about what to do for this final project. So I decided to recreate some of the functionality of this web app in Excel. There really isn't really a business need for this, but nevertheless it was an opportunity to display some of the things I learned in this class. To recreate some of the functionality of the web app I started with a user form to allow a user to search for products in different categories based on a desired price. When a user clicks search it takes the information they entered and queries our api, which talks to amazon and finds products that are available in that category and at that price. As the results are received, I

download the images associated with the products. Once the product information is all downloaded, a user form pops up that allows a user to navigate through each product, with the option to view it on amazon's we page, or save the information for later.

Implementation

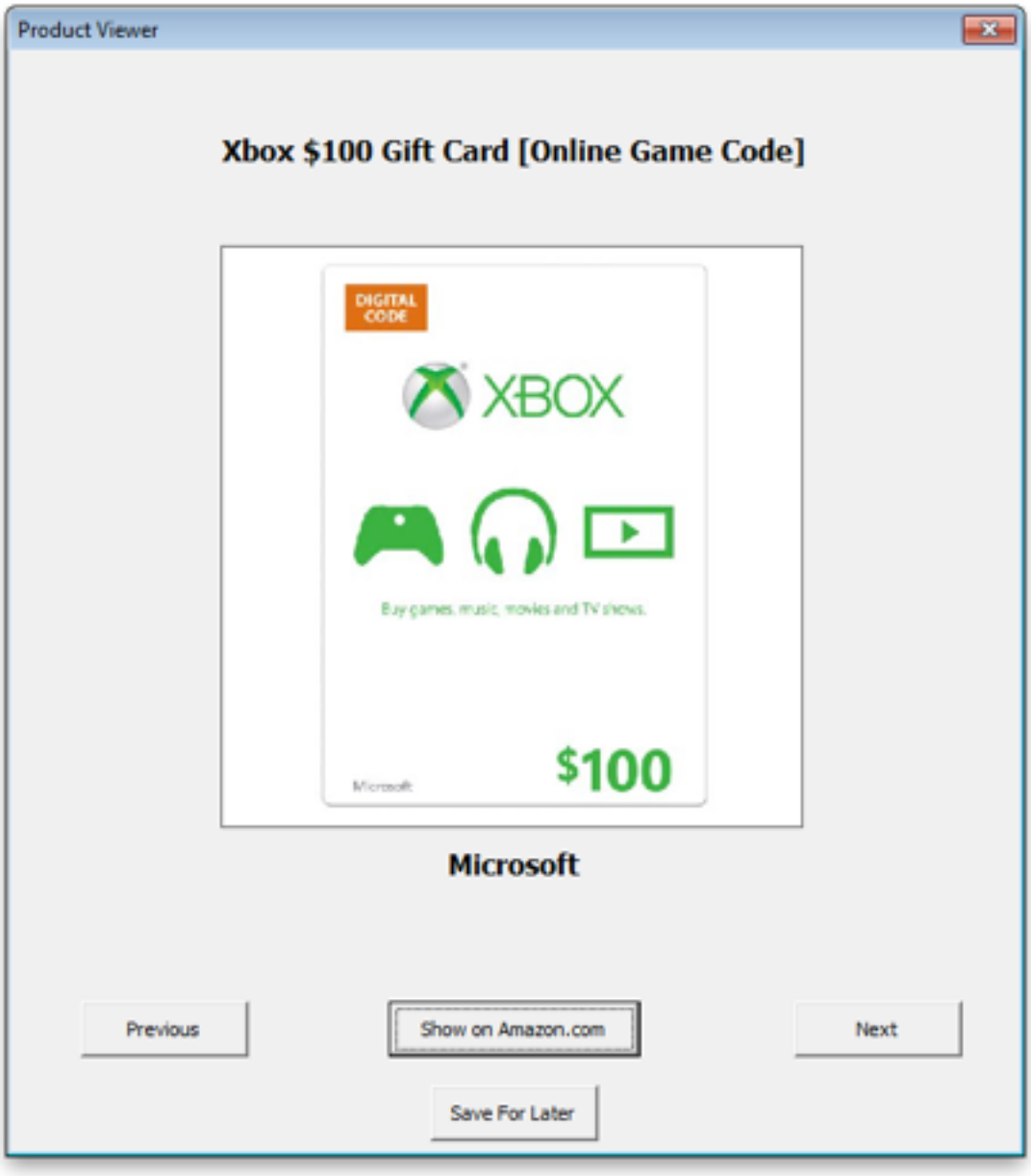
In order to accomplish this project I had to first learn how to connect to a web server over http and parse an XML response. Actually learning the http request was not too difficult, I used MSXML2.XMLHTTP60 to create and send my requests. I will note that it took a bit of googling and some reading to figure out. What took me a bit longer, however, was learning how to deal with XML in VBA. It turns out that it is not that difficult, but I was not able to find good documentation and there seems to be many ways to work with MSXML2 objects. I ended up writing some pretty complex code to get access to the XML text values, only to find out later that much of it was unnecessary and that there was an easier way to get that information.

After learning how to request and access the XML response from the web server, I focused on creating a user form that allowed a user to form a request. The user form I created is pictured below:

This form allows a user to chose one of 48 categories to search, the number of responses to show (10-50), and to enter a dollar amount. Once the user clicks "Search" the code does a few preliminary checks. First it deletes all of the previous image files, and then checks to see the number of desired responses, takes the category and dollar amount, and then makes one request to the server for each set of 10 responses the user desires. Once each request comes

in, the XML is received and added to the table in the spreadsheet, and the agent class is instructed to download the image for each result. This is done at the point where the product is added to the spreadsheet. The images are stored in a folder named “images” that resides in the same directory as the Excel file. References are relative so this code should work on any computer, provided the directory in which the file resides has a folder entitled “images”.

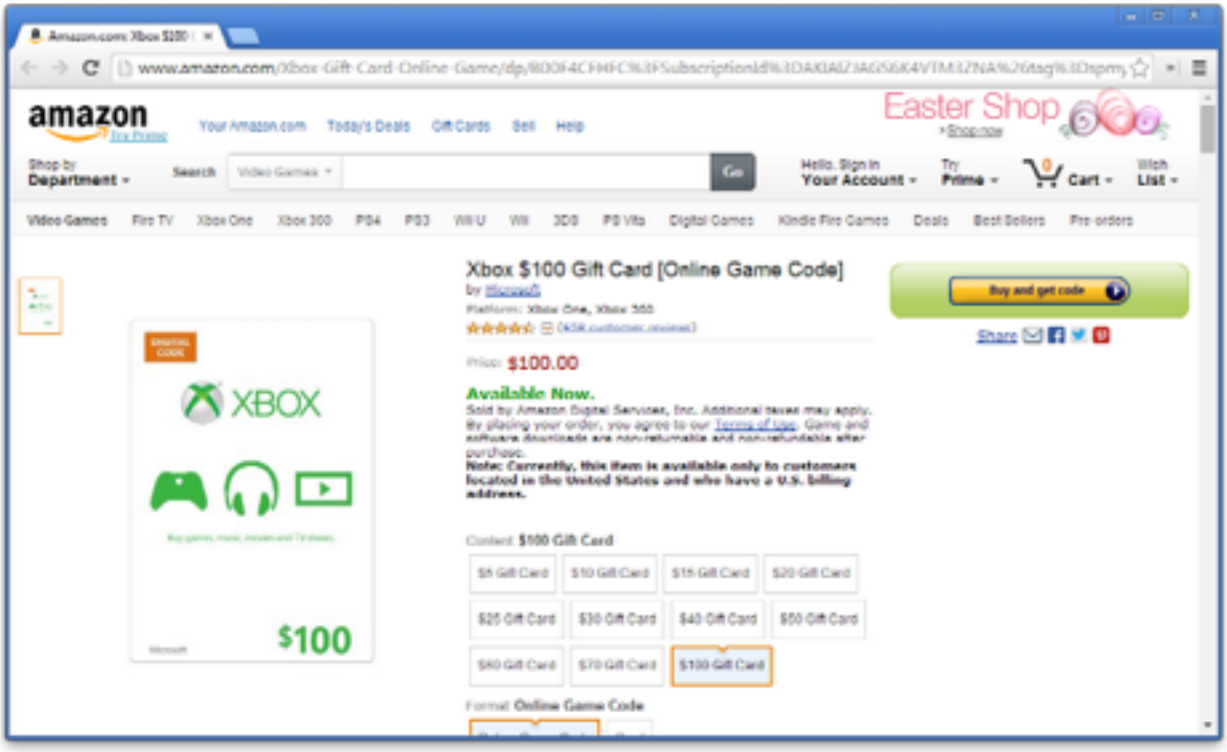
Once all of the images and product information is downloaded, a user form is brought up, and the information, and image, for each product is displayed. An example is provided below:



When this user form is initialized, it looks gets the first result on the spreadsheet and loads the information and image associated with it. To do so the image url is split to get the desired file

name, and the file is loaded from the images folder (note: the images are saved in the same manner, the image file name is derived from the url). As the data is loaded, the respective values are set on the user form. Next and previous buttons allow for continuous browsing through each product on the list.

Perhaps the coolest feature of this form is the button “Show on amazon.com”. When the form is loaded I store the amazon url associated with the product, and if the “Show on amazon” button is clicked, I run a handy script that opens the url in the OS’s default browser (in my case chrome).



This is my favorite feature because, one, its cool to use VBA to open other programs and send them commands, and two, it makes the process easy for the user. I thank stack overflow for showing me the script and commands to do this bit of awesome-ness.

The last useful feature of this project is the ability to save this information for later. When a user clicks the “Save for Later” button, it takes the information that is currently being displayed, and puts it on another sheet (“SavedProducts”) so that it can be found later on after other searches are performed and older results are gone. In addition the image associated with the product is saved in another folder called “saved_images”. Products can be viewed in the SavedProducts sheet by clicking the “Show Viewer” button.

Learning and Conceptual Difficulties

One of the greatest struggles for me on this project was learning how to utilize VBA to request and access XML. The code itself is not that complex, but learning how to get the data, and

especially to make sense of the response was a process of looking on many stack overflow posts, forum posts, and some nasty Microsoft help sites. If there is anything I learned through this project, it is that the Microsoft support for VBA is a bit convoluted, at least for those new to it.

Another struggle for me was trying to get images to download. I tried recording a macro in which I inserted a picture into a sheet from a url. It work fine when I recorded the macro, but the code would not run when I tried to run it later. It was a bit frustrating, and come to find out, I had to utilize the agent class to save the files, and then load them from the disk. I was trying to avoid this so that the process could be a bit faster, but it ended up working ok. One thing I did learn through this is that different versions of Excel will respond differently to the same code, and you need to keep that in mind when writing scripts.

Assistance

I asked Dr. Allen a couple questions about how to download pictures and he mentioned to use the agent. He also helped me realized that I was using the wrong url for my web server. I don' think either of these qualify as substantial assistance, most of the assistance that I received was on forum posts and stack overflow. Thank goodness for both.