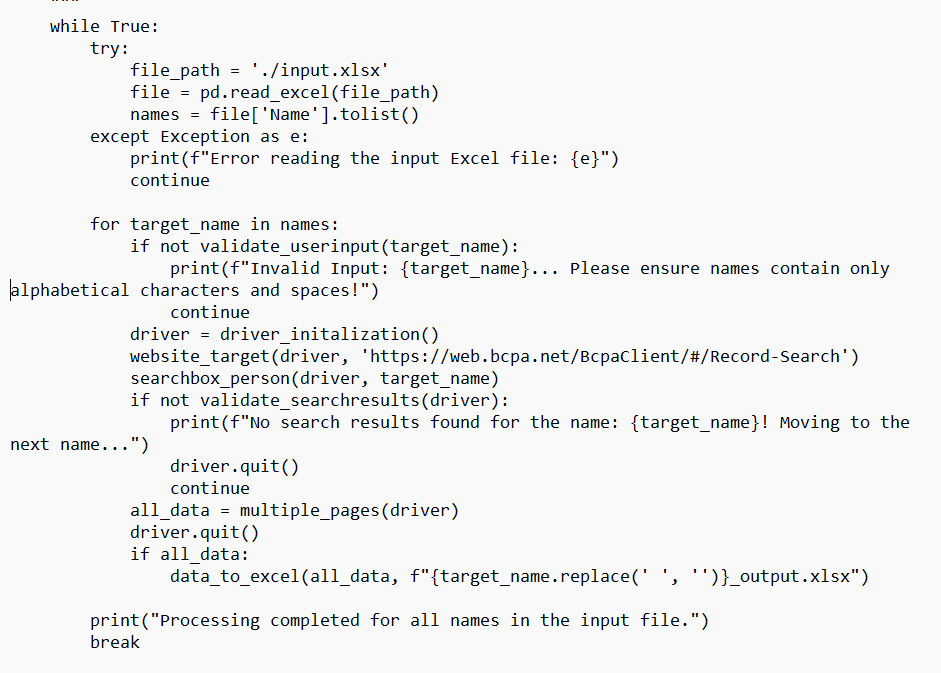
Guide to the back end of the county records inspector

Introduction

All functions of the back end are extremely similar to each other, so the first three code summaries are more in depth in regard to the common functions found among all the files. While the code summaries after the first here focus on the unique aspects of their respective functions such as the whisper library used only in collier.

Broward County

BR Part 1(Property Appraiser):

* Prereq: Install requirements.txt & input.xlsx
* Summary: All functions execute under the main function when called. The while loop is used to continually iterate through the list of names until no more names are available for ‘web scraping’. In the loop we do a try and except at the top where the try attempts to find/read the file and converts the column ‘Names’ into a list of names under the variable name. If there is an error reading the file then the ‘except Exception as e’ is raised, telling us there was an error.
* 
* If successful in finding a file, we continue to the next step which is a for loop where the names variable is iterated once for each name. Each name in the for loop is passed through each specific function under the for loop. Starting at the top of the for loop we pass it through ‘validate\_userinput’ where the name is checked to ensure it is only alphabets and spaces. In that function if it satisfies the requirement then we return True and continue with the other function below. If False, then we alert that the name is not valid and continue with the next name.
* A screen shot of a computer code

  Description automatically generatedAn example is if we pass ‘656345634’ will result in a raised alert. Ok assuming we got a valid name like ‘Smith John’ we continue down the for loop. By getting the driver and website url. Which will prompt us into the website. After that we execute the searchbox\_person function which will guide the text box to send the target name value into it. After that we execute the valdate\_searchresult function where we see if a results table appeared after sending the target name. If no results table appears then we return False and we know no results are available for that name. We quit the driver and continue with the next name or end process of no more names available. Ok assuming we returned True meaning a results table was found, so we continue to the next step which is storing all the results in the ‘all\_data’ variable where the variable executes ‘multiple\_pages’ function which includes a while loop that utilizes the ‘extract\_data’ function. Essentially in the while loop we have a try and except where the try extracts the results of that first page and clicks the next button if available. If no action occurs from that then we know there are no more pages of results.
* A computer screen shot of text

  Description automatically generated So, we break out the loop and return the ‘all\_data’ variable that contains all the results from the pages presented. After that the driver quits and ends and we have a if statement to see if there is data collected. If so then we convert the data into an excel by passing the data collected and output an excel file that contains the name of the target user. Now the while loop continues until all the names have been executed. After the last one we print out that processing has completed for all names and break out the while loop. Which ends the script for this part.
* More Comments on Specific Function/line in source code

BR Part 2(Clerk/Recorder)

* Prereq: Install requirements.txt & input.xlsx
* Summary: Now the main loop is similar to Part 1. The only difference is within each individual function executed. Due to it being a different website each website contains its own constraints such as navigating to a text box for the target name. In addition, some may require more steps to reach the textbox. As well the web elements are different for each website and you would have to identify which works best overall, etc. Here I will go more in depth on certain functions that require more explanation since other functions may be simple diagnosis of the web elements and seeing if the web element’s structure has changed on the website. If so then you would have to update the web elements or include more steps to reach the textbox. The function ‘conditions\_then\_searchboxperson ‘requires more steps than normally to reach the textbox so that we can send the target name value. Typically, we can pass the exact URL that leads directly to the prompt of the textbox but here it doesn’t allow that exactly. Each time we launch the driver we are prompted with a ‘Accept terms/conditions’ page where we must find the element of the text ‘Accept…’ then press click. After that we are then presented with the textbox and now, we can send the target name into it. These steps would be done every time a valid name is passed, and data needs to be scrapped. Another function that I can provide in more detail is ‘data\_to\_excel’ has 2 columns removed and the reason being that it's mainly empty data cells and we map the data frame of that data with ‘NULL’ values for those data cells that are empty.
* More Comments on Specific Function/line in source code

BR Part 3(Revenue Collection)

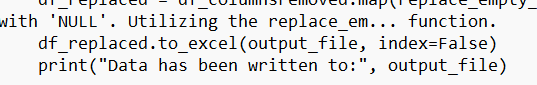
* Prereq: Install requirements.txt & input.xlsx
* Summary: As said before the main loop is similar but the key difference between this and the other 2 scripts for Broward this uses a ‘pagination element’ which is similar to a next button but with more requirements/steps. And in the main loop we have an addition of the ‘current\_page\_number’ variable that is set at 1 since the result page is 1 and if there are more pages then we add +1. And this variable is passed in multiple\_pages function as well rather than just the driver. In the multiple\_pages function we have a local global all\_data variable and a try and except that first attempt the try that extracts the data of that page and passes that data to the local global variable. From there we have the ‘if not navigate\_to…’ function that checks if there are more pages. If ‘True’ then we continue by adding +1 to the variable current\_page\_number. But False then we break out the while ‘True’ loop and return the all\_data variable. So, the ‘while True’ loop is for adding a +1 to the current\_page\_number until there are no more pages available and while adding it the all\_data variable for each new page. In the end, we return the data and after that we quit the driver and convert the data into excel. This is done for all names in the excel file. Another important note is the ‘extract\_data’ function requires more web elements since the results are not presented in a table as the previous 2 scripts. The data here are presented in ‘content cards’ which do not contain table rows and table cells as regular table data would. So, we must extract each ‘content card’ and strip the data into specific variables that are important such as owner/address, billing address, etc. Also, this process is done in a try and except for error handling purposes like if there was an error for extraction data then we raise the ‘except Exception as e’.
* More Comments on Specific Function/line in source code

Collier County

CO Part 1(Tax Collector)

* Prereq: Install requirements.txt & input.xlsx
* Summary: All functions execute under the main loop when called. With a ‘while True’ loop in the main() where it only breaks out of the loop when all the names from the excel files have been passed. At the top while loop for main() we have a try and except where the try attempts to read the input excel file and converts them into a list of names. Now if it couldn’t find the file then it would raise an ‘Exception’. Assuming it found the file, it would continue to the for loop where it would iterate through the ‘names’ list of names or ‘target\_name’ in this case. Where the for loop would send the target\_name to a set of functions such as ‘validate\_user\_input’ function that determines if the name is a valid string and not numbers or special characters. From there we initialize the driver and go to the URL. In this case the URL takes us directly to the search box page without pre-conditions. In the ‘searchbox’ function we pass the target name and driver and inside the function we find the textbox element through locators and then send the target\_name and press enter. Which would lead to the results pages and if an error occurred, we would raise an ‘Exception’. From there the validate\_searchresults function is to see whether results are available or not. If yes, then return True and if not then return False and print the reason why results weren't possible and quit the driver session. From here we have a current\_page\_number variable that sets the results page as page 1 and is passed to the function multiple\_pages() where it extracts data from the results page and the next pages if it exists. The while loop in the multiple\_pages ensures that all pages are extracted until the last page and through the use of ‘pagination’. After completion, what is returned is the all\_data variable that contains the data extracted. Next we quit the driver session and execute the data\_to\_excel() function that passes all\_data and appropriate output excel file with target\_name. Now the final print statement after this function is executed when all names have been passed then it breaks out of the while loop to the end of the execution of the script.
* More Comments on Specific Function/line in source code

CO Part 2(Property/Appraiser)

* Prereq: Install requirements.txt & input.xlsx
* Summary: The main loop is like part 1 of Collier. The major difference here is the ‘validate\_search\_results’ loop under the ‘try’ has two results tables to look for. But it would need to satisfy only one of the tables when looking for results after the target\_name has been passed. The reason behind this is the results can be presented in either a table of single results that displays the property summaries since there is only one result related to that target\_name and a table is not needed for that. Now the searchbox\_person() function requires switching to different frames wanting to reach the textbox field. Such as when on the website we must switch to the ‘r\_bottom’ frame to be able to find and click on the ‘Continue’ button. After that we must go back to the default frame and make our way to the ‘main’ frame to be able to access the side tab and click on the ‘search database’ button. After this has been executed, we switch to the ‘r\_bottom’ frame again to click the ‘I Accept’ button then we are finally led to the search box page of the website. Where we through a try and except we can click on the textbox field and the target\_name passed is broken into a last and first name variable to then reconstruct back to a single target\_name for the purpose of adding a comma between the target\_name when passed to the textbox field. The next function named ‘extract\_data’ gives an if and else statement for the purpose of seeing how many results are presented after passing the target\_name and clicking enter. If the results data is more than 0 then we execute the if statement since we know it’s a table of data and the else statement is executed since the results presented is just one and is presented in a property summary page rather than a table results of data. And in the execution of the if statement we go straight to converting the extracted data into an excel file and quitting the driver session because we know there are no more pages for extraction since there is only one piece of data available. While the if statement is being utilized multiple under the multiple\_pages function since there are more than one page of results that needs to be extracted until the last page which is determined when ‘ui-state-disabled’ is satisfied since it only activates when we are on the last page. Now what is returned is ‘all\_data’ which is passed to the ‘data\_to\_excel’ function which sends the collected data to an excel file and prints...
* More Comments on Specific Function/line in source code

CO Part 3(Clerk/Recorder)

* Prereq: Install requirements.txt & input.xlsx
* Summary: In the validate\_search\_results() function we are able to determine if results were yielded through the data\_total variable and return False if after the search the data\_total is set to 0 meaning that no results are available and we quit the session. While more than one result continues with the web scrape. Furthermore, in the searchbox\_person function we must meet the conditions beforehand like clicking on the dropbox button and checking the ‘Basic Property’ checkbox then clicking on the textbox field to send the target\_name using locators from the web elements. If an issue occurs under the web element, then we know that the locators have either been updated or other changes have been made to the website’s html source.
* More Comments on Specific Function/line in source code

Lee County

LE Part 1(Tax Collector)

* Prereq: Install requirements.txt & input.xlsx
* Summary: Like before All function executions are occurring in the call of the main loop. With the first steps being initializing the driver and reading in the input excel file into a list of names. Now the names list goes into a for loop where each name goes through data extraction. And the data variable under the for loop executes the ‘web scrape’ function where we send the name into the textbox field. And then extract data from the results and all the data extracted is stored in the ‘all\_data’ variable. And the while True statement only breaks when no more pages are available, letting us know that no more pages are to be extracted. Then we take the extracted data or ‘all\_data’ variable and pass it into the to\_excel function to convert the data into an excel file. This is done for each name from the names list.
* More Comments on Specific Function/line in source code

LE Part 2(Property Appraiser)

* Prereq: Install requirements.txt & input.xlsx
* Summary: All the execution of the functions occurs under the main function with the data extraction occurring in the webscrape() function which is stored under the data variable under the for loop in the main() function. Which is added to the all\_data variable that is later passed into the to\_excel function for converting the data into an excel file. The function that I will explain more in depth is the webscrape() function under the while True statement since we are comparing old data with new data in each page to determine if we should select the ‘naviagate\_to\_next\_page’ function. If the comparison is the same then we know no more data can be extracted because we are on the last page. On the other hand, if not true then we continue to select ‘Next Page’ until the last page of results. Once it is confirmed no more pages are available the extracted data is converted into an excel file for each name listed in the submitted excel file.
* More Comments on Specific Function/line in source code

LE Part 3(Clerk)

* Prereq: Install requirements.txt & input.xlsx
* Summary: The main difference in this script here when compared to the previous scripts is the use of the whisper library. Which is a model used to bypass the captcha found in the target website. Now if the script is later broken, the case may be that the model needs an update, or the captcha found on the website has been changed so we would need to redefine the human\_like function to an updated version. It would be helpful to utilize the whisper library documents for more information or new additions that can help you with bypassing the captcha. The function good\_results return a False when no records or changes of website structure have occurred. Going back to the captcha and whisper library, we will be selecting the audio option of the captcha where the model will translate the audio into text and that text will be passed into the textbox field. The process is better explained when executing the script of this part.
* More Comments on Specific Function/line in source code

GitHub Link

<https://github.com/Angelsantiago-lopez23/DHS-Project-ACDC>.