

CSC111 Project Proposal: UK VAT invoice generation using Python

Ayesha Nasir

Tuesday, March 16, 2021

Problem Description and Research Question

In UK and throughout the world, small businesses and contractors need to generate invoices that they send to their clients/vendors/companies/employers which outline their services rendered and is used for income tax purposes. While calculating the invoice amount, contractors/businesses also need to calculate the VAT rate. The Value-added tax (VAT) is a type of consumption tax that is levied as a percentage of the total cost of goods bought or services rendered. This tax is prevalent in European countries but not in US. Canada currently uses a mix of VAT at provincial and federal level - namely GST/HST and sales tax.

A lot of smaller businesses and contractors outsource their invoice generation to their accountants. They forward their daily timestamps from work to the accountant and the accountant has to calculate all the hours worked throughout the week/ quantity of goods sold, add any optional surcharge/overtime bonus, and then add the vat and generate the invoice, forward it to the client/vendor/employer on behalf of the business/contractor. The rate of VAT depends on the type of services provided or goods sold and is frequently changed according to the UK tax laws. Some of the accounting companies outsource this manual task of generating invoices to third world workforce. My dad happens to belong to this workforce, and he works for UK accounting firms with mainly data entry tasks that have added accounting aspects to it. This includes bank reconciliations and invoice generation. I would like to automate some of his tasks, taking the work burden off of him as there is some unnecessary typing involved, like inputting the client/vendor/employer information multiple times as most of the invoices for a specific person/business are redirected to the same vendors/clients/employers. Even the business/person's own information has to be inputted separately! For this reason, I decided to create a program that can help **automate UK VAT invoice generation using Python with autocomplete functionality**.

Computational Overview and Data

Project description and computation involved:

To start with, I generated some mock company and vendor data using mockaroo.com. I created csv files for payers(or the company/person the invoice is forwarded to) with 500 rows and a separate csv file for vendor (the business/contractor providing the service) with around 15 rows. I used this csv and dataclasses that I had written in `data_classes.py` to create lists of vendor and payer objects. The property for vendor and payer objects were mostly similar, like name, address, contact information and the vendor object had a separate property for the vendor's service type and the given vat rate.

These vendor and payer objects were passed into the AutocompleteTree object for each, which created a tree structure containing all the vendor/payer objects as leafs and the subtrees leading to the leaf object were breaking down into characters of the object's name, as this would help in creating the autocomplete functionality.

The `get_suggestions` function for the autocomplete tree returns a list of payer/node objects that correspond to a given list of characters. (they include these characters at the beginning of the name. The returned value from this function is used to give autocomplete suggestions in the suggestions listbox right below the entry widget for the name of the payer/vendor. In the GUI, you can manually click a name from the suggestions box for the vendor/payer name, or start typing in the entryboxes and see the suggestions box update according to the characters inputted. Clicking on a name would set the vendor/payer as the object represented by that name. Enter the correct data types in the remaining entry boxes and click the generate invoice button for the invoice to be generated and open up automatically in chrome.

Instructions

To obtain the dataset which generates the autocomplete tree, please download the folder 'data' in markus containing payer_data2.csv and vendor_data.csv and place the folder data in the same directory as the other python files. Create a empty invoices folder in the same directory. You can also download the project zip file folder I uploaded on markus and just work from there instead of downloading each file separately.

Install reportlab and tkinter.py, and open the main.py file and run it. In the resulting gui window, select the payer and vendor from either using the suggestions provided in the suggestion box (you may have to double click) or start typing in the entry box and select specific payers/vendors. Input any str in services rendered box, and type numerical values for Quantity and Amount. Add a string as a due date. Once you click generate invoice, the gui window should close and a invoice pdf should be generated in the empty invoice folder and also open up automatically in the web browser.

The autocomplete tree also has a display function which will display all the subtree and leafs with proper indentations (the last subtree will be linked to the name of the payer object it contains). After running the main.py file, you can print the respective autocomplete tree for payers with display function.

Discussion

I ended up having to forego a lot of the features I initially wanted to implement, including adding a new payer/vendor and using scrapy to get up-to-date and actual vat rates from the gov.uk website. The program proved to be quite complex since learning tkinter and creating the autocomplete functionality were pretty tough, and I was doing the project on my own. I still consider the autocomplete function a victory. There are many improvements that can still be made but those are minor additions. I did manage to generate an invoice and use autocomplete function using the Tree structure to shave time off manually inputting the information of payers and vendors everytime I would need to create an invoice, so the project's goal can be considered attained.

Next steps that would improve this program would be adding more entryboxes in the GUI, implementing the editing function by allowing to save the invoice as csv and recreating the proper invoice structure in reportlab.

Libraries/Modules used:

- **tkinter.py:** tkinter.py is a graphic user interface (GUI) library that will be used to get input from the user for invoice generation and give the option for user to download as a pdf or csv. CSV file can be further edited by the user in case something needs to be changed.
- **reportlab:** reportlab will be used to generate the final pdf invoice that the user can download.

References

- VAT rates on different goods and services. <https://www.gov.uk/guidance/rates-of-vat-on-different-goods-and-services>
- Value Added Tax Examples. (2020, December 29). Retrieved March 16, 2021, from <https://www.investopedia.com/ask/answers/are-some-examples-value-added-tax.asp>
- tkinter (n.d). <https://docs.python.org/3/library/tkinter.html#a-very-quick-look-at-tcl-tk>
- reportlab (n.d). <https://www.reportlab.com/opensource/>
- GUI programming with python (n.d). https://www.python-course.eu/tkinter_entry_widgets.php
- Mockaroo.com Random Data generator. (n.d). <https://www.mockaroo.com/>
- Basic Search and Autofill Tutorial, Youtube.com. (14 Jan 2021). <https://www.youtube.com/watch?v=0CXQ3bbBLVkt=190>