Further Research

Research about useful python libraries for excel:

As this project is going to be relying on a lot of data to work properly, I need a library that can help me with the database handling. As I’m going to be using excel to host the databases, I need a library that can work with .xlsx files and that allows me to read and write data to those files.

Microsoft Excel is widely used in almost every industry. Its intuitive interface and ease of use for organising data, performing calculations, and analysis of data sets has led to it being commonly used in countless different fields globally. Python is an extremely powerful language with an extensive ecosystem of 3rd party libraries. Leveraging Python in Excel spreadsheets can be a fantastic way to enhance productivity and remove the need for importing and exporting data into and out of Excel.

For some tasks I may need to read or write an Excel file directly. The following packages allow you to read and write Excel files directly without needing to use Excel:

- OpenPyXL:

For working with Excel 2010 onwards, OpenPyXL is a great all-round choice. Using OpenPyXL you can read and write xlsx, xlsm, xltx and xltm files. OpenPyXL covers more advanced features of Excel such as charts, styles, number formatting and conditional formatting. It even includes a tokenizer for parsing Excel formulas. If you need to read Excel files to extract data, then OpenPyXL can do that too. The Excel file types are incredibly complicated and openpyxl does an amazing job of reading them into a form that’s easy to access in Python. There are some things that openpyxl can’t load though, such as charts and images, so if you open a file and save it with the same name then some elements may be lost.

- XlsxWriter:

XlsxWriter is a Python module that can be used to write text, numbers, formulas and hyperlinks to multiple worksheets in an Excel 2007+ XLSX file.

If you only need to write Excel workbooks and not read them then XlsxWriter is an easy to use package to use that works well. If you are working with large files or are particularly concerned about speed, then you may find XlsxWriter a better choice than OpenPyXL.

On the other hand, this library is not that useful to me as one of the main requirements of my program is to have the ability to also write data on the databases, which this library doesn’t seem to support.

- xlrd/xlwt:

Xlrd and xlwt read and write the old Excel .xls files respectively. These two libraries are now really only used when you are forced to deal with the legacy xls file format. They are both extremely mature packages that are very capable and stable, but xlwt will never be extended to support the newer xlsx/xlsm file formats therefore for new code dealing with modern Excel file formats they are no longer the best choice.

By doing this research I found that OpenPyXL is the best choice for my type of program and it supports the file extension that I need and also allows to both read and write data into excel files. Furthermore, this library offers a great documentation which will be very helpful when programming the bot and has a lot of support content created by the community in forums and on YouTube.

Research about useful python libraries for Telegram:

This chatbot will be hosted in Telegram, and for that reason I need to find a library that will allow my program to connect with Telegram via their API. There are many different types of libraries available, some of them created by the community as well, which means that they receive a lot of support. These are some of the libraries I can work with:

- Telebot:

The Bot API is an HTTP-based interface created for developers keen on building bots for Telegram. This library provides a pure Python interface for the Telegram Bot API. It's compatible with Python versions 2.7, 3.3+ and PyPy. In addition to the pure API implementation, this library features a number of high-level classes to make the development of bots easy and straightforward.

- Telepot:

Traditional version works on Python 2.7 and Python 3. It uses urllib3 to make HTTP requests and uses threads to achieve delegation by default. Async version works on Python 3.5 or above. It is based on asyncio, uses aiohttp to make asynchronous HTTP requests, and uses asyncio tasks to achieve delegation.