The “Who Can Help” document consolidates data from a variety of sources including the Student Information System (SIS) as well as school-based documents that track monitor teachers for students with individual education plans (IEPs), students connected with the Contact Program, as well as students that are identified as English Language Learners. The purpose of this document is to assist administrators, guidance councillors and teaching staff in quickly identifying the in-school human-resource supports available to students.

**Step 1: Gather Reports & Lists**

Before using this application, it is necessary to download the following reports from SIS:

* Master Timetable (report code: ??????????)
* IEP List (report code: ??????????)
* ELL List (report code: ??????????)

Monitor lists need to be generated by curriculum lead teachers. Below is a visual template to illustrate the standard structure of the document. There are four columns of data: column A is for the student number, column B is for the students’ last name, column C is for the assigned monitor teacher.

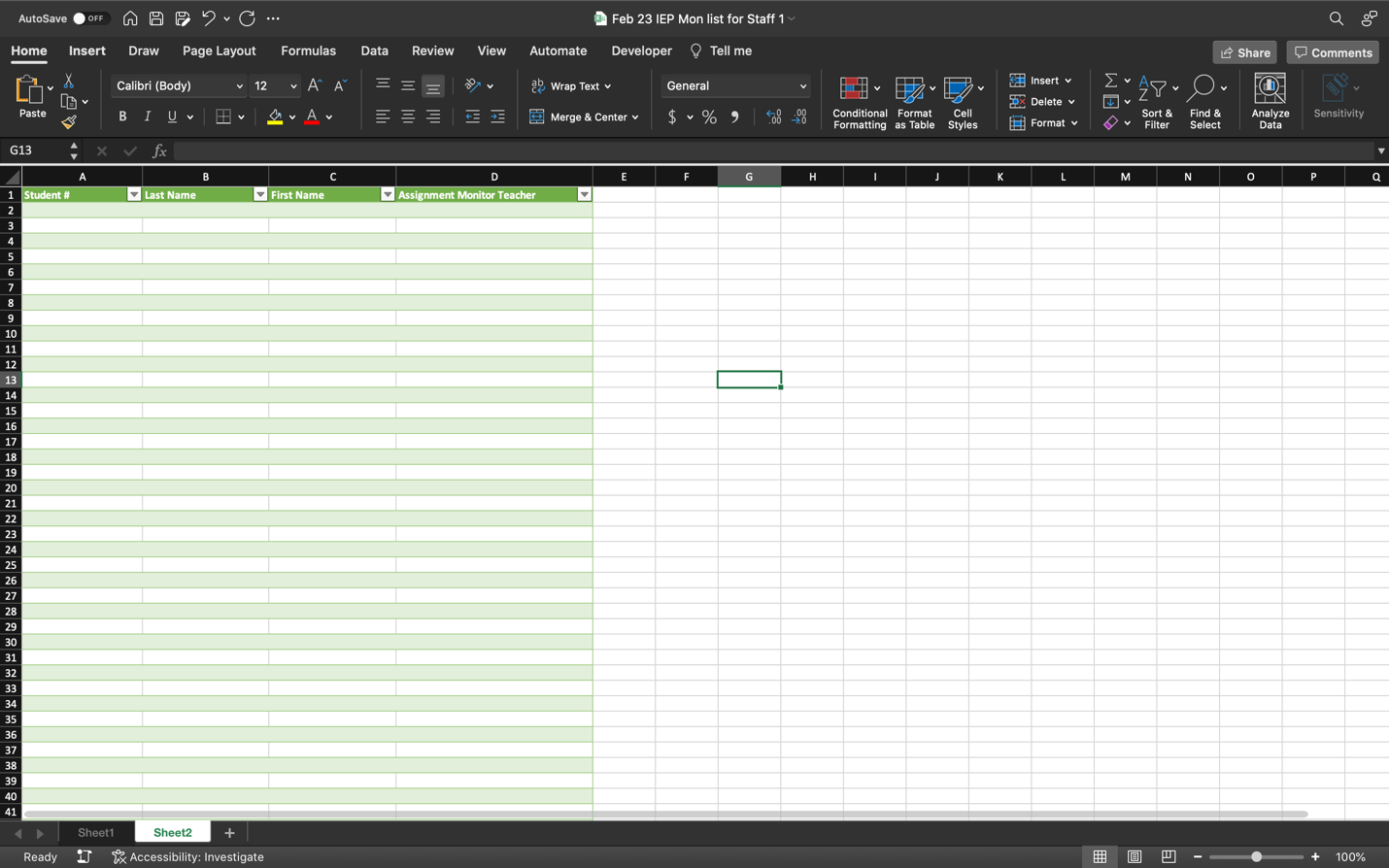


Figure 1: Sample Format for Monitoring Data

You will need monitoring data files for the following:

* Special Education monitor teacher assignments
* Contact monitor teacher assignments
* English Language Learner monitor teacher assignments

**Step 2: Adding Files to the Configuration File**

After the files have been gathered and renamed for easy identification, they should be placed in a folder so that it is easy to locate all the required files, as well as for archiving purposes. Currently the program is written with a command line structure (i.e., there is no graphic user interface), and it requires manual inclusion of the file paths into the configuration file. The location of each of the six required files needs to be entered into the config.yaml file. For more information on how YAML files are organized, you can reference [*YAML Tutorial: A Complete Language Guide with Examples*](https://spacelift.io/blog/yaml)(2022).

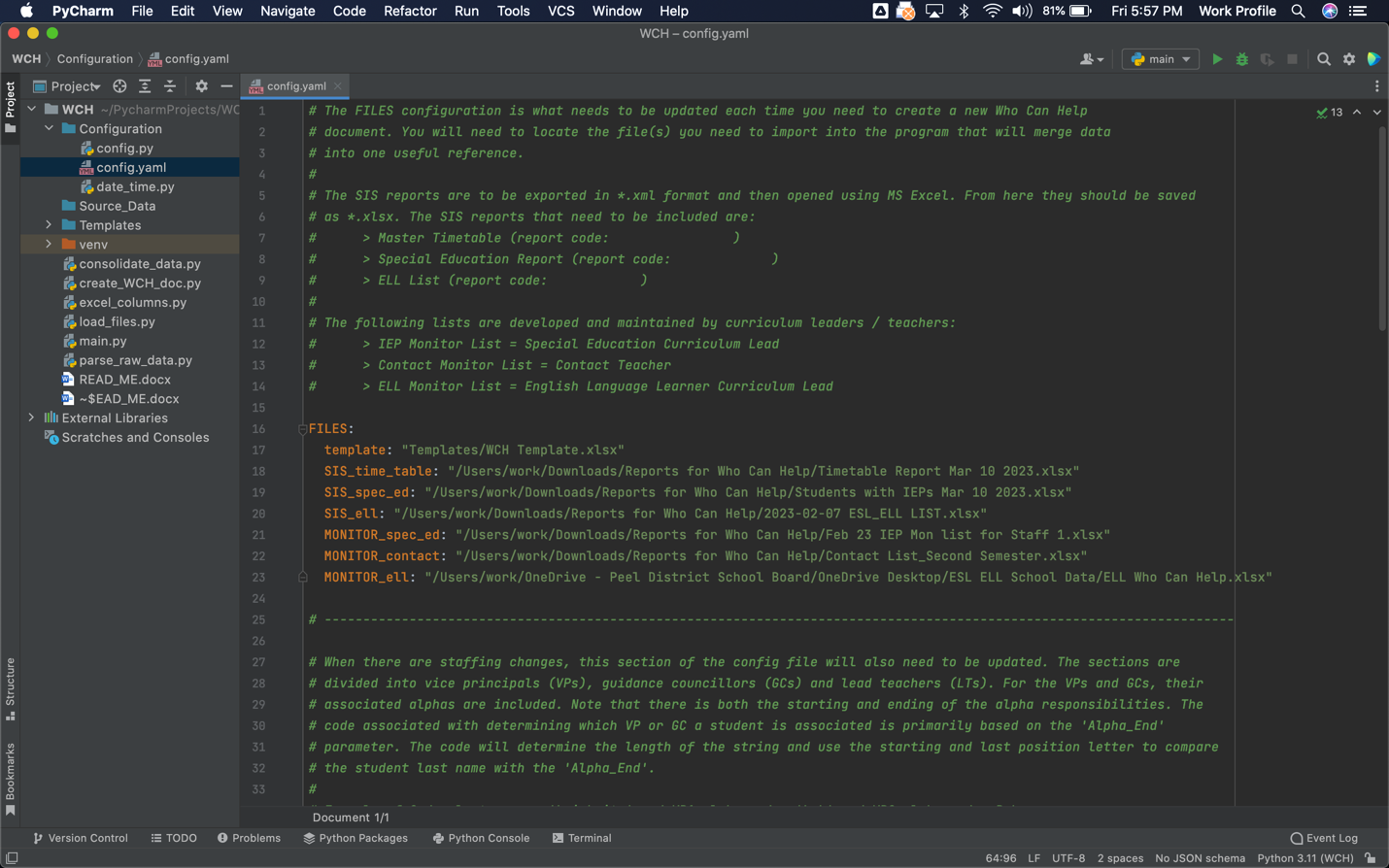


Figure 2: Image of the config.yaml file for inclusion of the file paths.

Within PyCharm, the left column is a file browser for all the items associated with a given project. The config.yaml needs to be opened. It is located under the main folder for the project:

* ….>WCH>Configuration>config.yaml

Double click on this file and the YAML file opens in the editor on the right side of the screen. This file has been documented through comments (lines beginning with #) to help provide some clarity about the needs and requirements within this file to allow the program to function properly.

In the orange text in Figure 2 you can see that there is a section called “FILES”, and subsequent keys “template”, “SIS\_time\_table”, “SIS\_spec\_ed”, “SIS\_ell”, “MONITOR\_spec\_ed”, “MONITOR\_contact” and “MONITOR\_ell”.

The template file should not be touched. This is set up for the “Who Can Help” document that will be published at the end of the running of this program. Each of the subsequent file paths need to be added to this document. You will need to locate the absolute path for each of the files.

On Windows, the absolute path can be found using Windows Explorer / File Explorer by right clicking on the desired file and selecting “Copy as path”.

Graphical user interface, text, application, email

Description automatically generated

Figure 3: Windows, copy absolute file path

When working in OSX+, set the view options for Finder need to set “Show the Path Bar”. From the menu bar, select View > Show Path Bar. At the bottom of the Finder window, you will see the absolute path for where the file you select is located within the system.

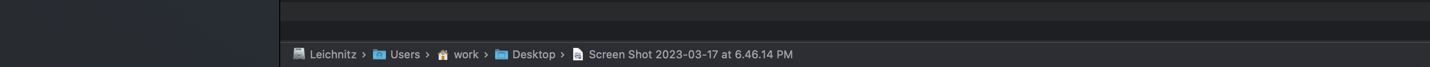


Figure 4: Bottom of Mac Finder window showing the absolute file path of selected file

Once you have the path visible, right click (or control click) on the file on the path bar, and a context menu will appear, and select the “Copy ‘FILE NAME’ as Path” option.

Now that the file path is on the clipboard of your system, go back to the config.yaml file and paste the path next to the appropriate key. When pasting, be sure that there is a space after the colon, and that the path is within quotation marks.

Text

Description automatically generated with low confidence

Figure 5: close-up of the config.yaml file for the absolute file paths

Repeat the process of locating, copying and pasting the \*.xlsx file for all six of the required files.

**Step 3: Confirming the Data Columns**

Each SIS file has its own data configuration. The data configuration is organized by columns and student data is entered into each row of the spreadsheet. It may be necessary to point the software to the correct columns if you get unexpected results when you run the program. Note: you may receive an ERROR if the student number data location changes from the pre-set in this version of the program. This is a clear indication that the data location data needs to be corrected in the config.py file.

Each file has its own primary key written in all caps within the configuration file, and the data to be extracted from the \*.xlsx file is identified in the form of a dictionary (key value pairs). The keys refer to the data to be extracted and the value references the column within the spreadsheet that the data is located (e.g., TIMETABLE: s1p4: “L” means that from the timetable MS Excel file the semester 1 period 4 data is in column “L”).

Note that the only data that needs to be located within the IEP or ESL/ELL lists from SIS are the student numbers. If the student number is present on this list, it indicates that the student has an IEP or is identified as being an English Language Learner.

Similarly, the monitor data is located within MS Excel files, and though there is more data in the file than is strictly required for creating the who can help, only two columns of data are required: the student number and the monitor teacher assigned to provide extra support. Ideally all the monitor lists should be formatted consistently, with the student number data located in column “A” and the monitor teacher name located in column “D”.

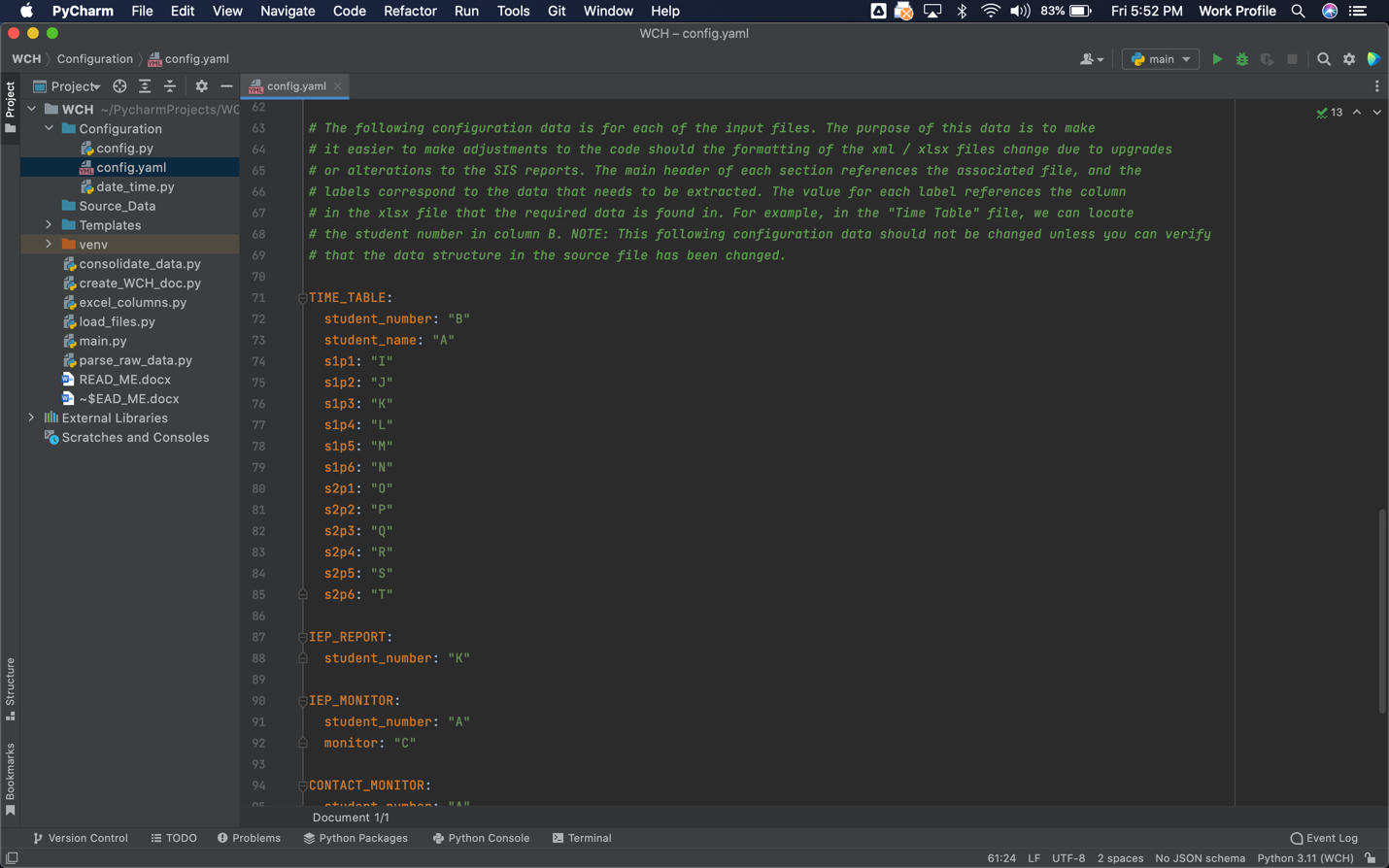


Figure 6: Screen shot of the config.py file for ensuring column identification of specific data is accurately tagged

**Step 4: Run the ‘main.py’ File**

Text

Description automatically generated

Figure 7: screen shot of PyCharm for running the program

There are a variety of ways in which the program can be run. Using PyCharm, double click on the ‘main.py’ file to open it in the editor. In the top of the editor window, there is a dropdown button that should say main in it. Next to this button, there is a green triangle (highlighted with red in Figure 7) that can be used to run the program. Alternatively, from the PyCharm menu bar you can select Run > Run ‘main’ or you can use keyboard short cut ^R.

Once the program is running, the first prompt for the user is to verify that the file paths are correct.

Graphical user interface, text

Description automatically generated

Figure 8: program has been started and asking for config data to be verified

The software will confirm that the configuration data is accurate and that files are in the correct file format (\*.xlsx). Next you need to identify whether the “Who Can Help” document is being built for semester 1 or semester 2.

Text

Description automatically generated

Figure 9: config data is verified and user is prompted to select semester

The software will confirm that the configuration data is accurate and that files are in the correct file format (\*.xlsx). Next you need to identify whether the “Who Can Help” document is being built for semester 1 or semester 2.

Graphical user interface, text

Description automatically generated

Figure 10: data from the source files are imported into the software, the data is parsed, sorted and filtered

After selecting the semester, the program will open each file in sequence, parse the data, sort the data based on student number and then filter out duplicate entries.

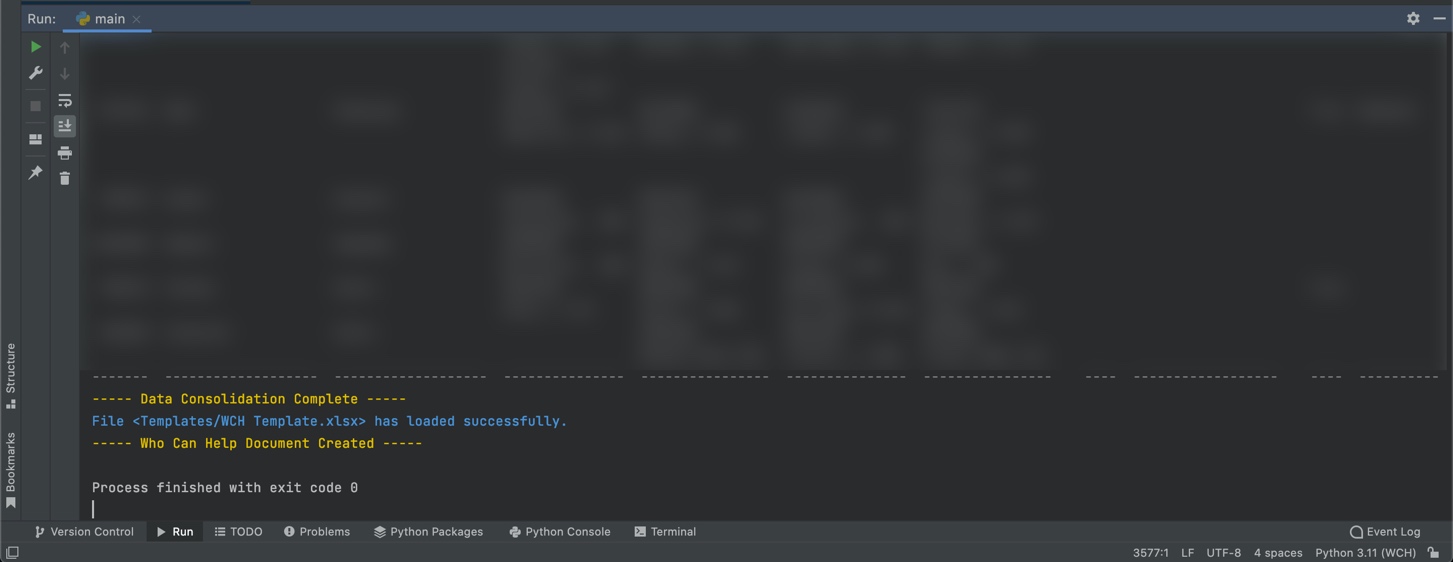


Figure 11: the program has been run and the who can help document has been created

Once all data has been parsed, the software consolidates the data into one array. As the program runs, you can preview the data in the terminal (blurred in figure 11 to maintain the privacy of student data). The final step for the program is to write the consolidated data into an \*.xlsx file. The program will create the title of the document as the date the program was run with “Who Can Help” (e.g. 2023-03-17 Who Can Help.xlsx). The file can be located in the file manager of PyCharm (left window) as seen by the red highlighted section in figure 12.

Text

Description automatically generated

Figure 12: the location of the created Who Can Help data