**Group 5 - Programming Project Presentation**

Created by Olivia B, Benjamin L, Noah H1, Elspeth CJ

**Overview:**

* We were asked to create a program allowing a University Medical School to be able to download pharmaceutical trial data from a trusted scientific partner for medical research
* The files are stored on an FTP server, and there should be both command-line options and an interactive front end
* There should also be error checking and logging for common issues, for example missing columns, or empty files
* As a team we chose to use Python in our solution, and work on the assumption that the users are able to use the command line to download any missing Python modules

**Accessing the FTP server:**

ftpmodule.py and run.py

The user creates or specify a local directory, into which files are downloaded.

All the files are collected from the server, but only the ones matching the user’s inputted date are downloaded to their machine (Figure 1).

Text

Description automatically generated

Figure 1

Then the given file name is split up, so the program can collect the date.

This function is used in downloadFiles() to check available files for the desired one (Figure 2).

Text

Description automatically generated

FIGURE 2

This function allows the user to connect to the target server or throws an error if it is unable to (Figure 3).

Text

Description automatically generated

FIGURE 3

These functions are the core of the program. The user can use the command line to specify the server to be connected to, and the date of data to be fetched. It also validates the date format using validate\_date\_format() (Figure 4).

Text

Description automatically generated

FIGURE 4

The user can also save the files from the server (Figure 5).

Text

Description automatically generated

FIGURE 5

**File Validation:**

fileValidation.py

Once all the required files have been downloaded, the program ensures that they are all valid, checking against the criteria:

* The file isn’t empty
* All the column headings follow standard naming conventions and order
* Each batch in the file has a unique ID
* There are no missing columns
* The data entered is valid (0 < data <= 9.999)

If an error is found, a log is created and added to the log file (Figure 6).

Text

Description automatically generated

FIGURE 6

There are two interpretations of an empty file – entirely blank, or with the correct headers but no readings taken. This function checks for both possibilities (Figure 7).

A screenshot of a computer

Description automatically generated with medium confidence

FIGURE 7

Once we ensure the file has content and is not empty, we make sure that the headings have the correct names in the correct orders (Figure 8).

Text

Description automatically generated

FIGURE 8

Each file consists of multiple batches. We validate that there are no duplicated batch numbers in the file (Figure 9).

Text

Description automatically generated

FIGURE 9

Inside the file, each row must have the same number of columns. This function checks this is correct (Figure 10).

Text

Description automatically generated

FIGURE 10

Finally, we ensure each item of data in the file is valid within the correct range (Figure 11).

A screenshot of a computer

Description automatically generated with medium confidence

FIGURE 11

These functions are open the file and create the log file to store when any problems arise (Figure 12).

Text

Description automatically generated

FIGURE 12

**What have we learnt:**

This has been a new experience for us - working within a group with a big range in our individual skillsets. We had a variety of levels of programming, teamwork and project management skills however we managed to work well together so that everyone found a role benefiting their strengths.

It was also interesting using GitHub to manage the project. It took some time to learn when to push and pull for people not so used to version control systems, but by the end we had a good flow working together.

Trying to divide the project up into separate sections for each of us presented an interesting problem. We had to make sure that everyone had an equal share of work while also everyone played to their strengths. After having a group meeting we understood these factors and came to a conclusion of what role each person would be best suited to undertake.

On reflection if we repeated a project like this there would be a few things we would try to improve for next time. Firstly, we would meet more often to make sure that everyone is on schedule and working towards the same idea. Secondly, we would set better conventions and standards before starting the project. For example, before starting we did not choose a consistent variable name convention so some variable names are CamelCase whereas others are snake\_case.

Overall, we really enjoyed this project and have learnt lots of useful new skills for our futures.