AVL AST Python Developer technical interview – TASK 1

Using **Python 3** and **PySide6** (Python binding for Qt framework) create the application with a main window featuring a folder name textbox, button marked with '>>' which opens a folder choosing dialog, button featuring a bar chart icon, and *read-only* multi-line textbox (Fig. 1.)

After successful folder selection, list of names of files in the selected folder (not full path, filename only), including the files in the corresponding subfolders, should be printed into a multi-line textbox. Files scanning should be limited to 1000 files, and if reached 1000 filenames should be printed, so that the application and the system are not clogged with scanning potentially huge (sub-)folders.

Bar chart button of the main window opens new Qt modal dialog which displays a bar graph presenting scanned filenames' first letter statistics. E.g. if there are 15 files with name starting with letter 'a', there should be a bar marked 'a' of height 15. Statistics should include both alphanumeric and all possible special characters scanned filenames start with, e.g. '-', '_'… Small and capital letters' statistics should be merged, i.e. there should be a single bar displaying a cumulative number of cases of both 'a' and 'A' letters. Only detected characters should be displayed, not those which are not found in any of the scanned files. Characters should be sorted according to some expect usual criteria, with digits and letters being grouped and ordered. Bar chart should be plotted using **matplotlib** library and the plot should be *embedded* into a modal Qt dialog featuring a 'Close' button at the bottom (Fig. 2.)

Python implementation should be done in a single file with a simple architecture. Optionally and advisably an elementary MVC separation can be performed. The application is simple so it should be the solution.

Colors, fonts, icons, specific dimensions or any other unspecified property or detail can be freely chosen, with a tendency toward a good design and pleasant user experience.

All the files necessary for the application execution, including the source code and accompanying graphical elements, should be packed into .zip archive and delivered via e-mail. Nor the task nor the solution is allowed to be publicly shared, distributed or communicated to other parties.
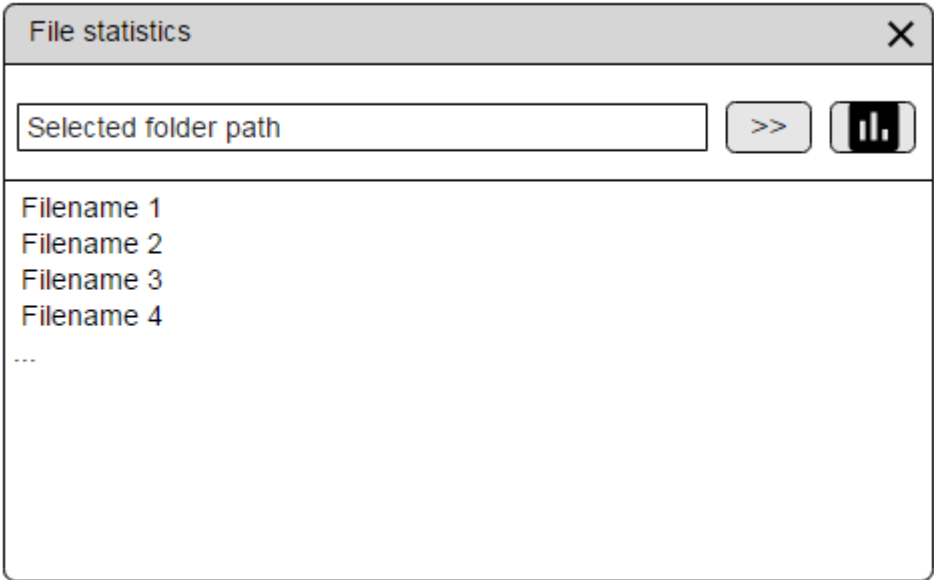
https://pypi.org/project/PySide6/

https://pypi.python.org/pypi/matplotlib/

Fig. 1

Fig. 2