Introduction to Operating Systems

CMP 310

Final Project

Aadhith Shankarnarayanan - b00089801

Vibha Bhavikatti - g00089208

Koushal Parupudi - b00087520

22 November 2023

Fall 2023

American University of Sharjah

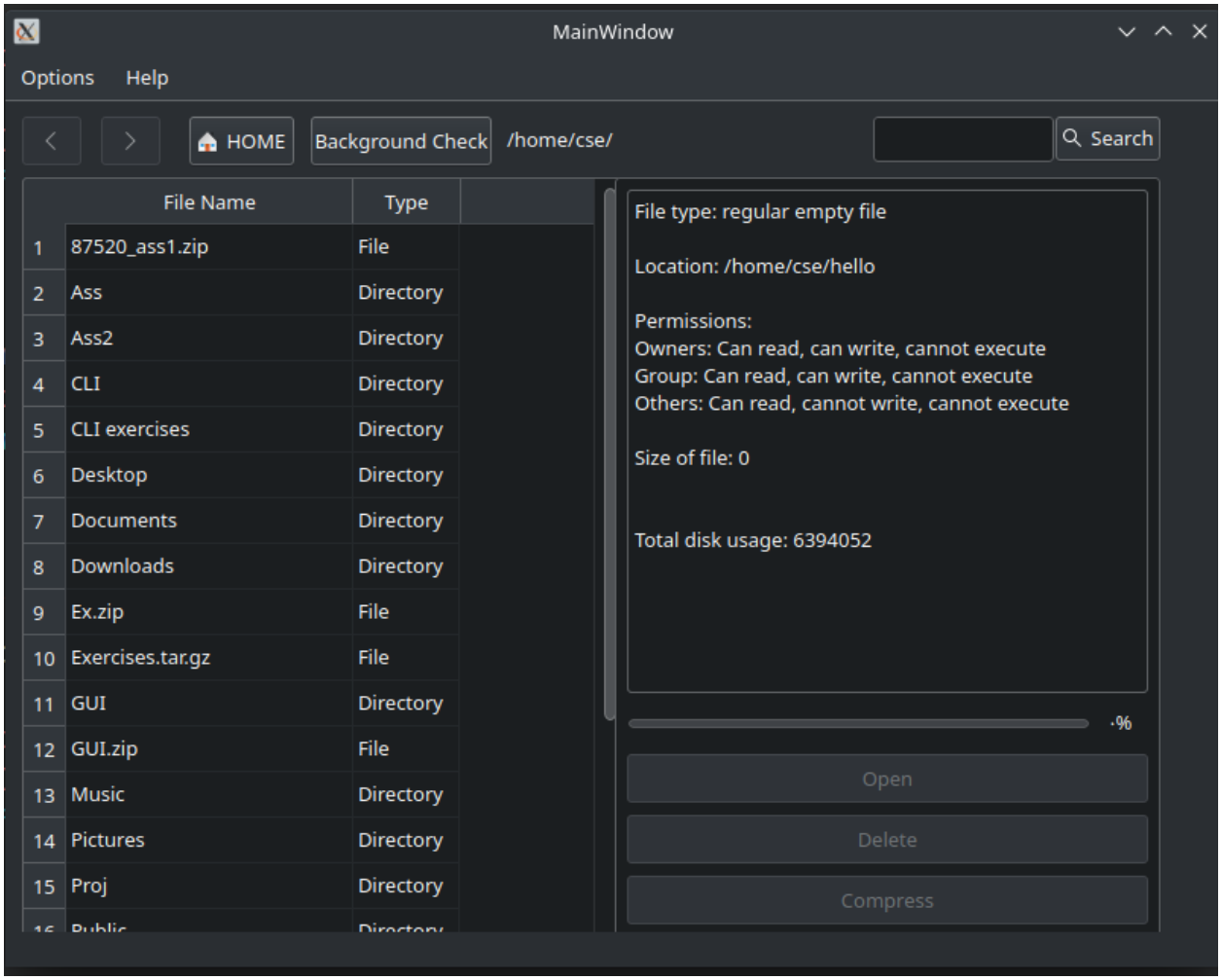
Dr. Gerassimos Barlas

Filesystem Utility Tool

The goal of this project is to develop a File System Utility application on the Linux platform using C/C++. The primary objective of the application would be to create a tool for viewing the file system along with its properties and additional features. Additionally for large files, users may be prompted to implement compression techniques to reduce the size of the file, and thus save disk space. To ensure user-friendliness and accessibility, the application aims to have a GUI to perform these tasks with ease.

As per the proposal of the project, the following tasks have been implemented in our application:

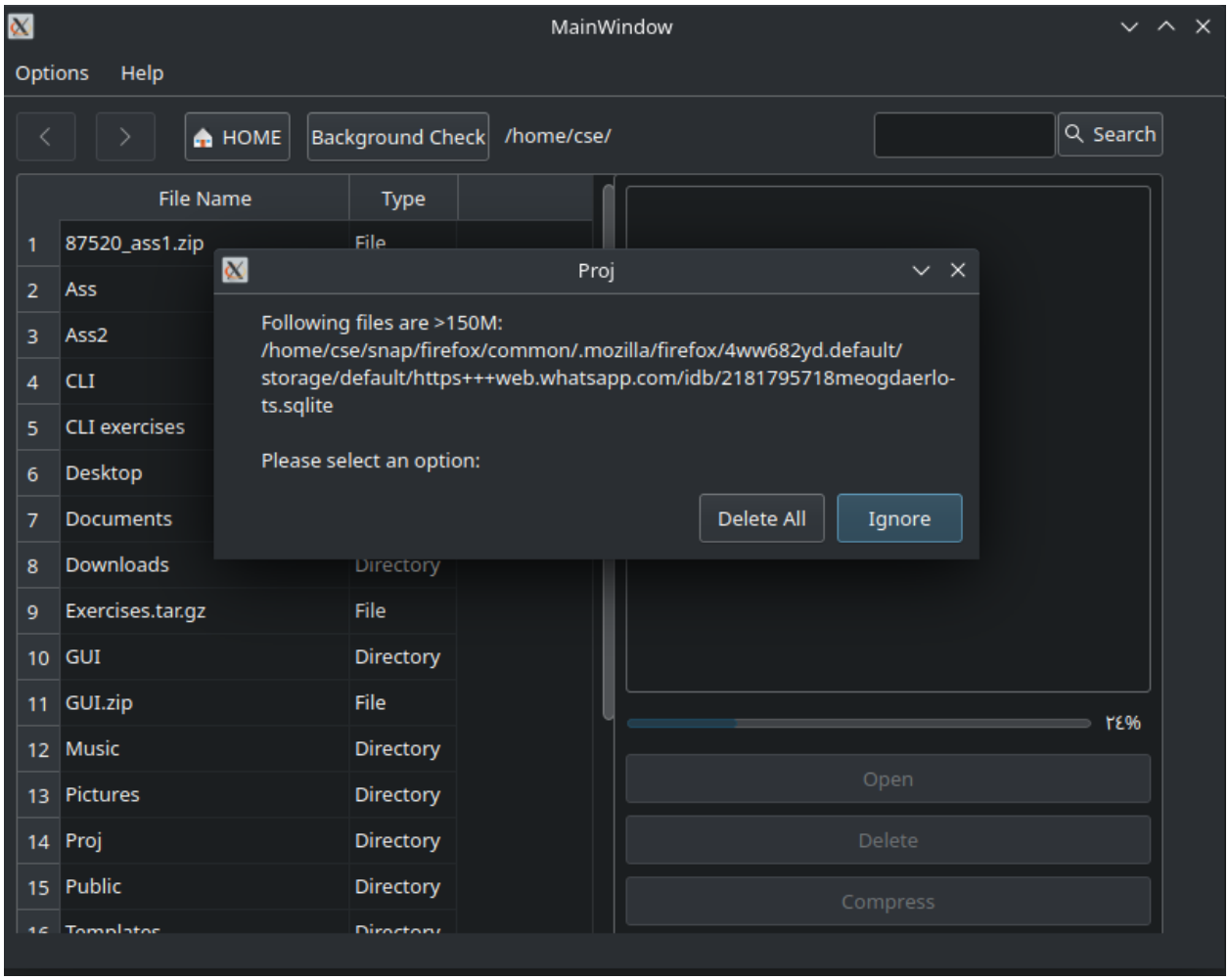
1. **File and Directory Listing:**
   The application will implement a user-friendly GUI that lists all the files and directories within a specified directory. Options to recursively list and explore sub-directories will also be provided to the user. Attributes like the size, modified date, and permissions for the files and attributes will also be displayed effectively.



This GUI displays the home screen of our application, where the list of files and directories are displayed. The files when clicked on show the metadata of each file, which displays all the file properties such as type of the file, location, the permissions and the size of the file. It also displays the total disk usage.

There are additional features added to enhance our application which will be listed below.

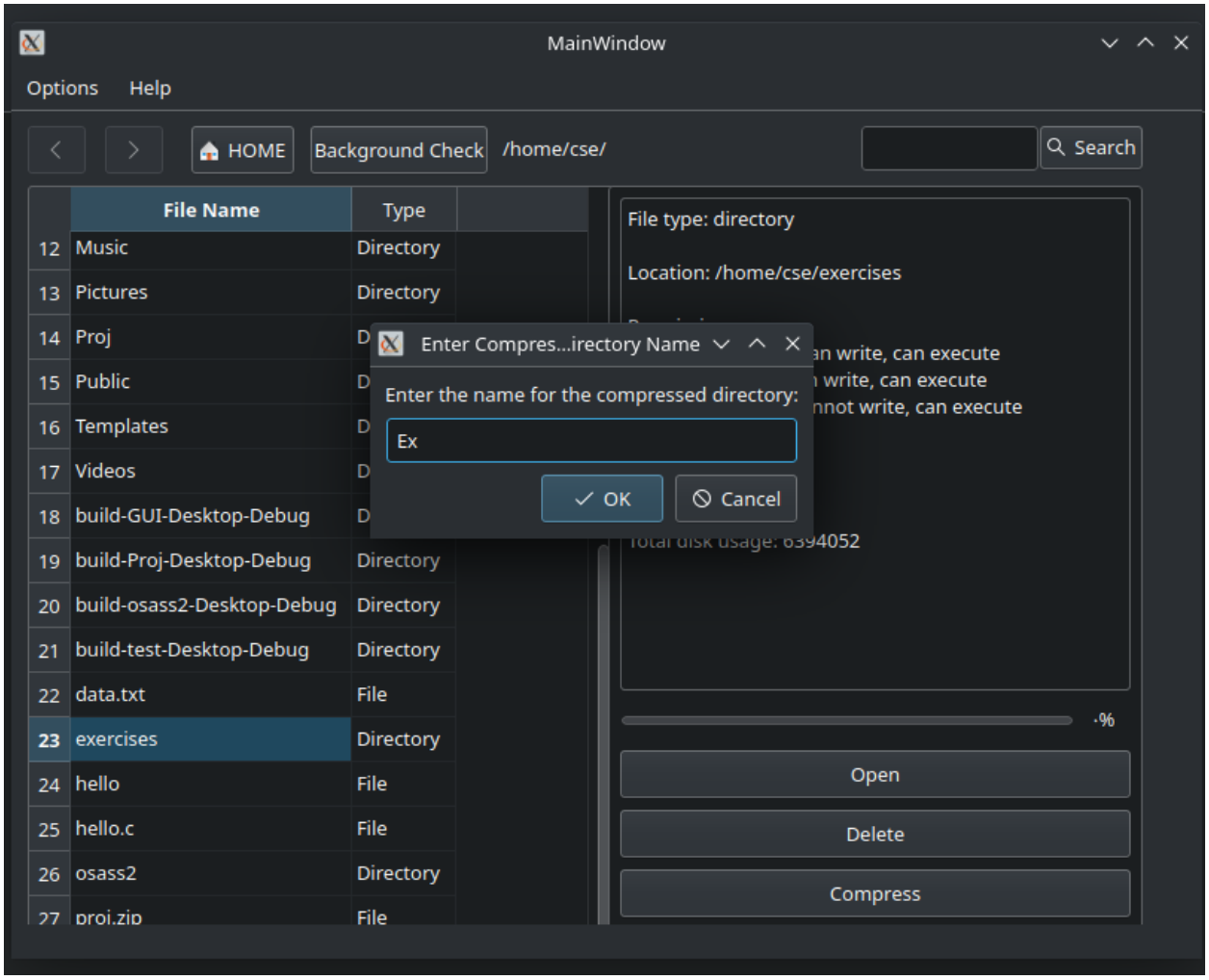
2. **Large File Detection:**
   The application will implement a feature that identifies and lists all the large files on the GUI that exceed a certain threshold, depending on the user's choice. It should also allow the user to be able to view and sort the list of large files based on their size or other attributes.

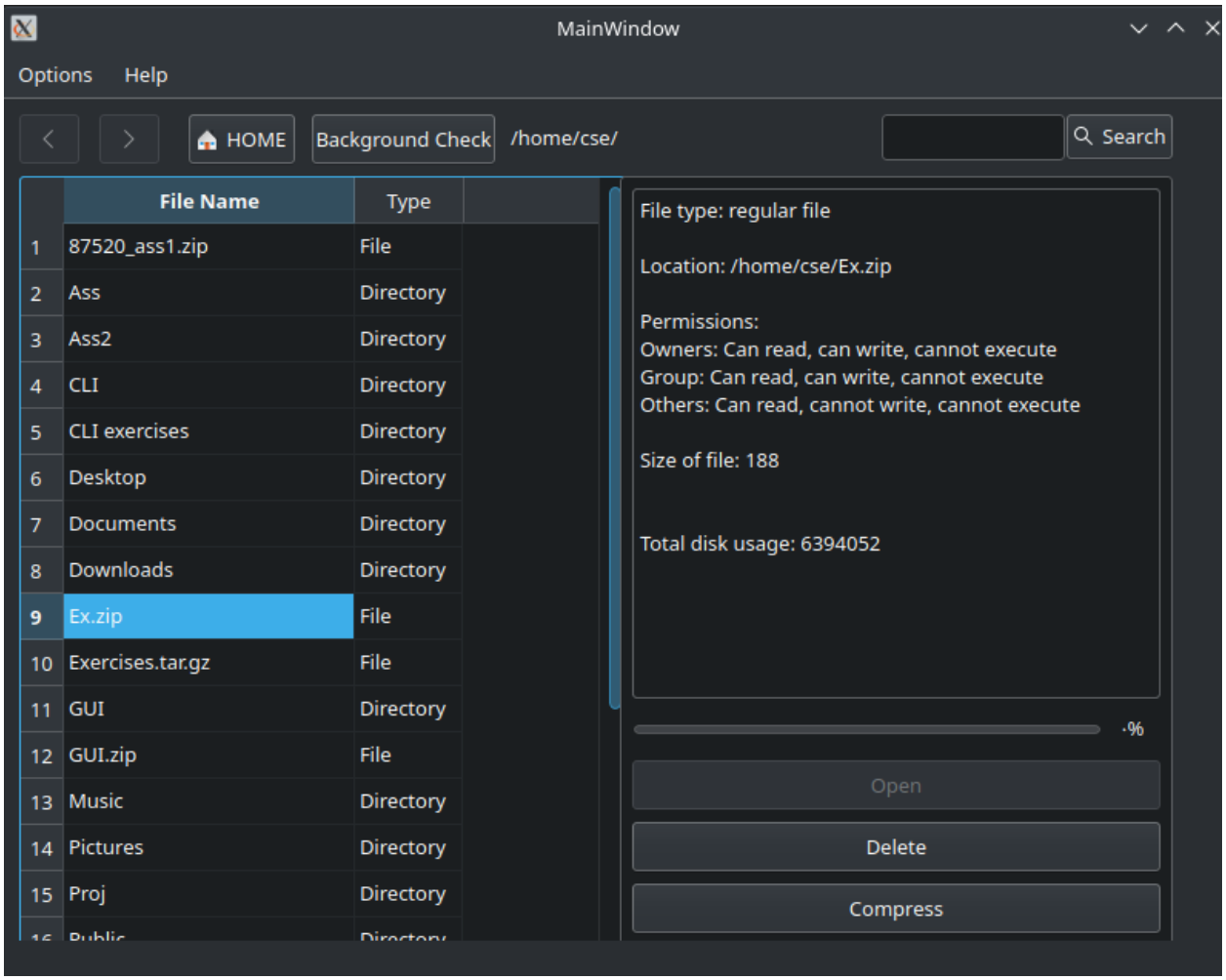


To detect and display the large files, a button is placed where a background thread runs to check all the large files, based upon a threshold size. This threshold size can be changed by the user if

needed. Once clicked on this button, it will display all the large files and ask the user if they want to compress these, delete or just exit from the dialog box which shows all the files.

3. **File Compression:**
   Upon identifying the large files, the application will provide the user with the ability to compress the large files, in order to save disk space. The application would also be able to decompress these files when needed by the user.

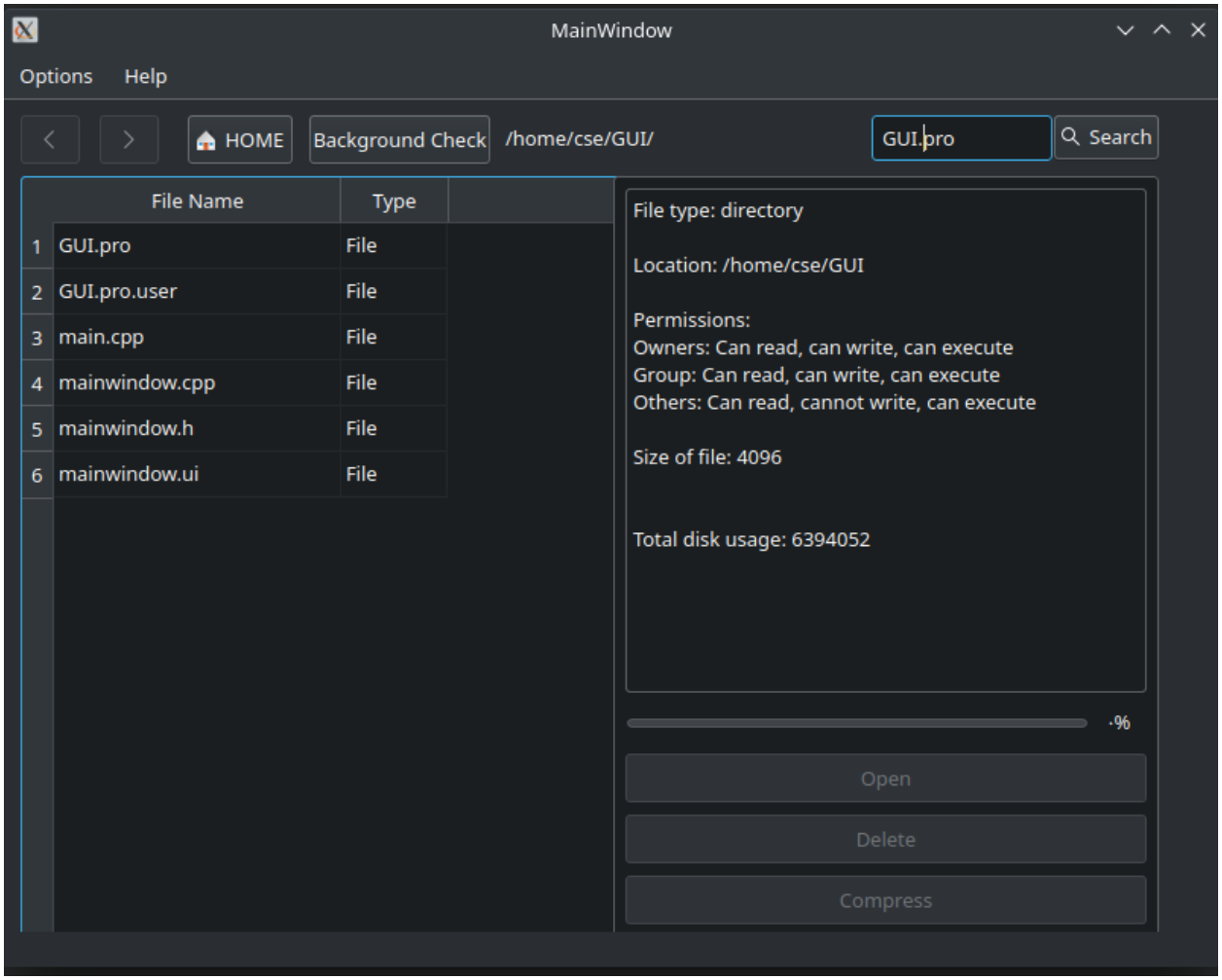




The file compression can be done using the ‘Compress’ button when a file or directory is clicked. Once the button is enabled if a file can be compressed, clicking it can help compress the file if the user wants to.

There is also an option to check all the large files as mentioned in the point above. Clicking this can compress all the files which are larger than the threshold size.

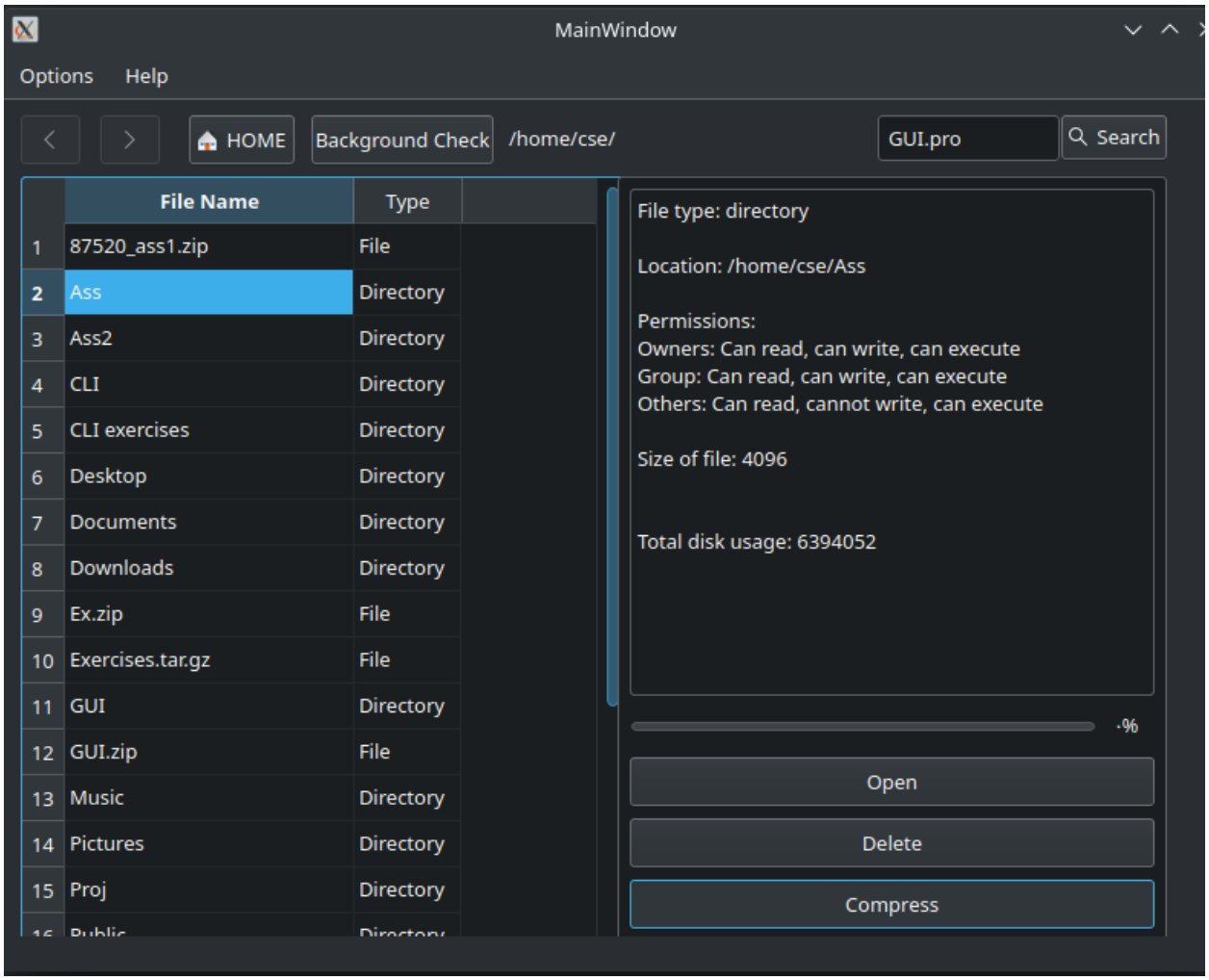
4. **Search Functionality:**
   The application will implement a feature that allows users to search for files or directories by name or by content. Search filters like the data-range, or file-type may also be provided by the user.



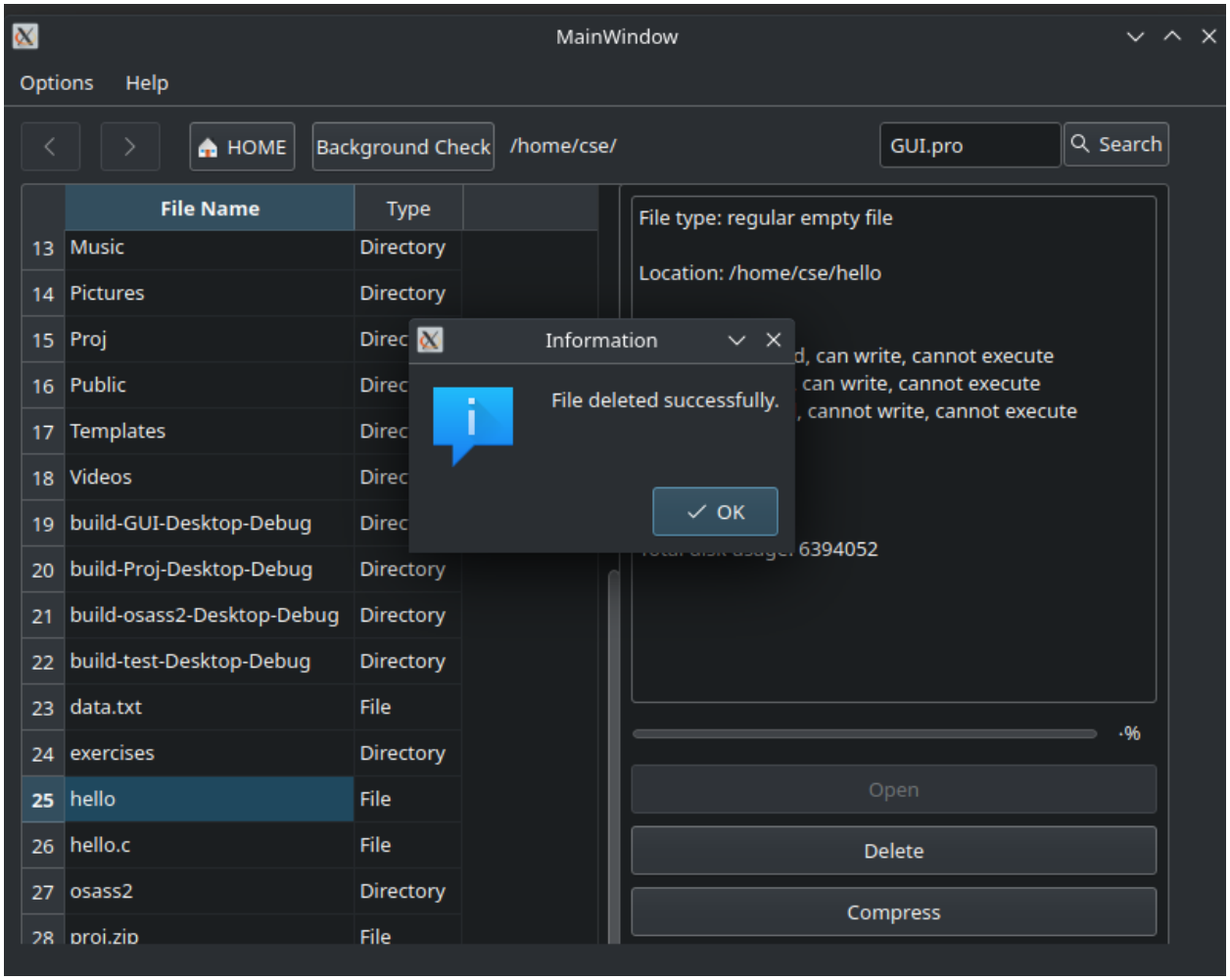
To help the user navigate through the application, an added functionality to search for whichever file they want to is located on the top right corner of the GUI. This helps the user search when needed.

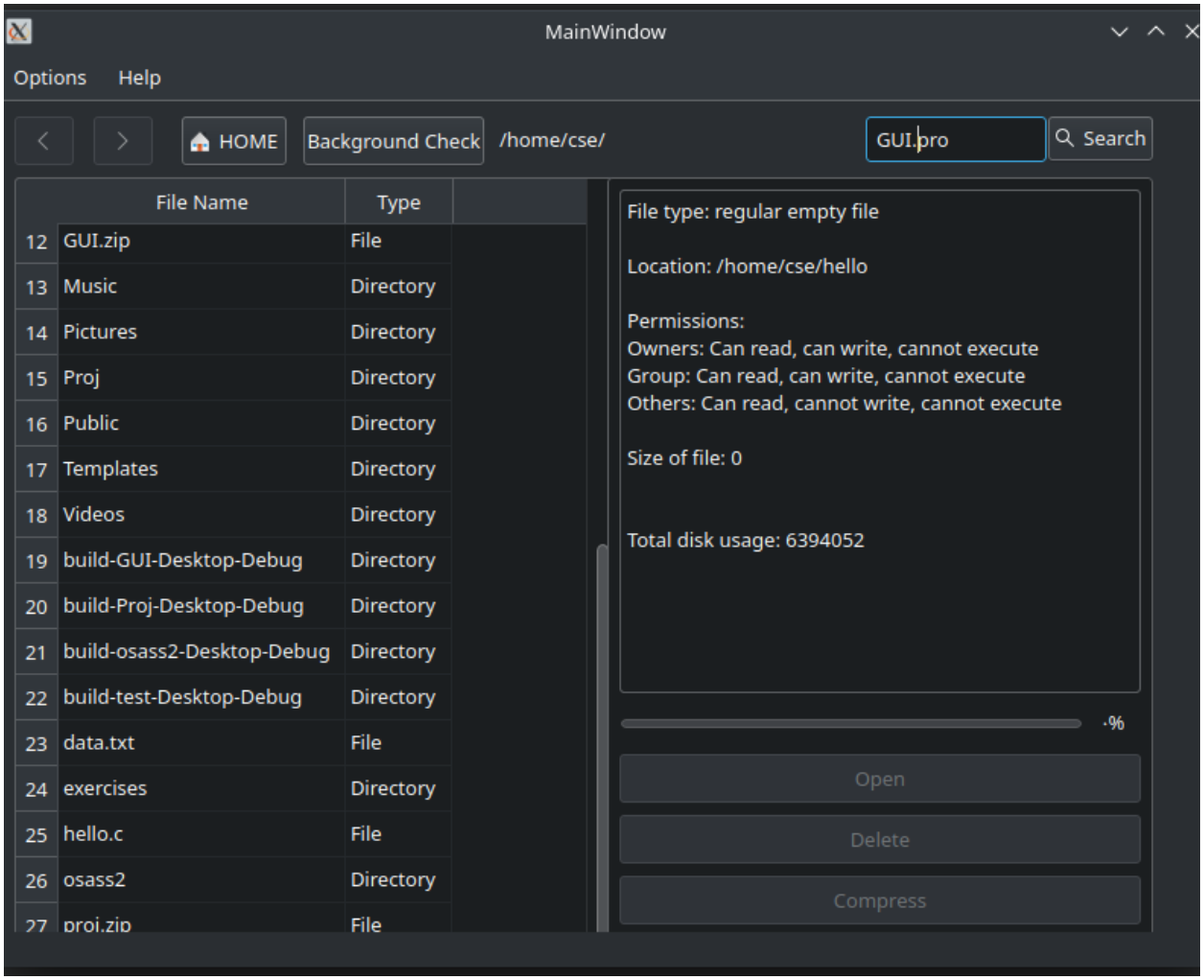
5. **Reporting Filesystem Statistics:**
   The GUI would implement the feature to generate reports that summarize filesystem statistics. This would include statistics such as the total number of files, disk usage, size of files and more.

This was implemented by finding the total disk usage and the size of the file that is clicked on as well. The comparison of the size of the file against the total disk usage was shown on a progress bar. This was accomplished using the linux commands ‘stat’ and ‘du’, where ‘du’ shows the total disk usage of the current directory, and ‘stat’ reports the properties of a file.



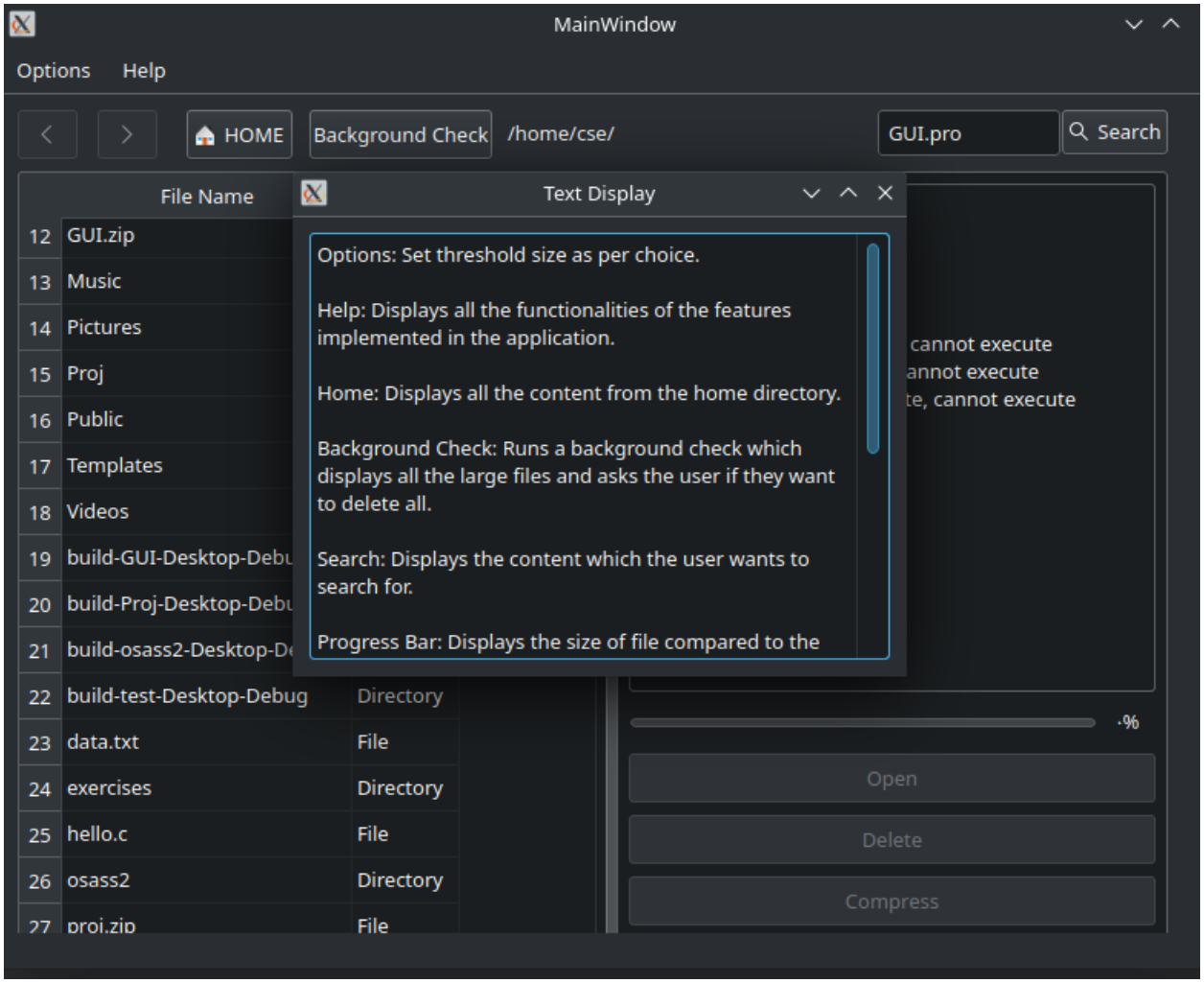
6. **Error Handling and Verification:**
   The application will implement strict user-checking to deal with file conflicts or permission issues. Moreover, the user would have to verify the user authenticity to confirm certain actions done by the user, such as when deleting files/directories.





A confirmation message box is sent when the user tries to delete a file. When they want to compress a file, it asks the user to enter a name for the compressed file, so if the user does not want to compress it, they can simply just exit the box.

7. **Documentation and Help:**
   The application will include extensive documentation and help options to help the users navigate through the application effectively and guide them on the usage of the features implemented in the application.



A help button is added to the application to help them figure out what each button does. It displays the following:

Options: Set threshold size as per choice.

Help: Displays all the functionalities of the features implemented in the application.

Home: Displays all the content from the home directory.

Background Check: Runs a background check which displays all the large files and asks the user if they want to delete all.

Search: Displays the content which the user wants to search for.

Progress Bar: Displays the size of file compared to the total disk usage.

Path: Displays the current location of the file as the file path.

The files when clicked on display the file properties such as the file type, location, file permissions and the size of file. It also displays the total disk usage of the current directory. Files when clicked on can be opened if its a directory, can be compressed or deleted upon clicking the respective buttons.