# Stage 1: Development

As there are well over seven thousand lines of code in my software solution, I encourage you to scan through the Python backend and JavaScript files in my GitHub repository. There are file and method header comments in just about every file which explain their purpose and explain what sorts of parameters they take, the type of, and what the different response codes mean. There is a developer log ‘dev log’ at the bottom of *main.py* annotating my development process throughout the month, if you are interested.

## Code Repository

[Serpyne/StorageSaver: SAT Project for VCE Software Development 3/4. (github.com)](https://github.com/Serpyne/StorageSaver)

* [files/upload.js](https://github.com/Serpyne/StorageSaver/blob/master/project/static/js/files/upload.js)
* [files.py](https://github.com/Serpyne/StorageSaver/blob/master/project/files.py)

However, I will only include screenshots of the necessary methods for the program to run and process as efficiently as it does. Furthermore, some of these methods are crucial to improve the application’s usability and ease of file management.

Contents

[Stage 1: Development 1](#_Toc173696880)

[Code Repository 1](#_Toc173696881)

[Linear Search Algorithm 3](#_Toc173696882)

[Password Check 4](#_Toc173696883)

[Hash Function 5](#_Toc173696884)

[Encryption and Decryption 6](#_Toc173696885)

[Binary Search Algorithm 7](#_Toc173696886)

[Reverse an Array 8](#_Toc173696887)

[Quick Sort 9](#_Toc173696888)

[Sorted by Name, Ascending 11](#_Toc173696889)

[Sorted by Type, Ascending 11](#_Toc173696890)

[Sorted by Size, Descending 11](#_Toc173696891)

[Checking Filenames (Maintainability) 12](#_Toc173696892)

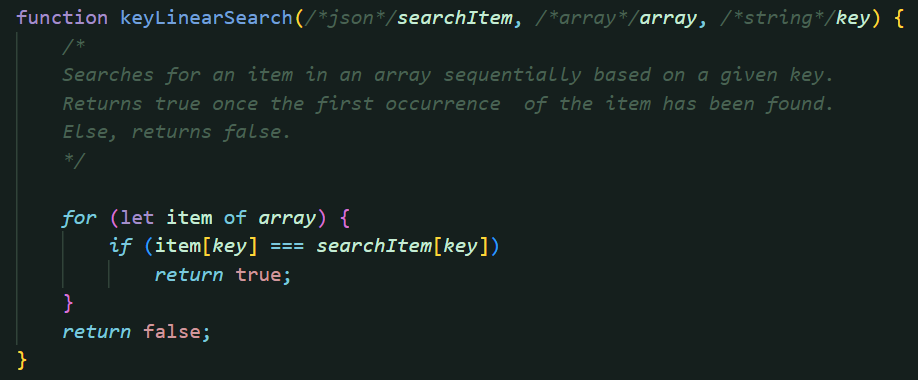
[Formatting File Size, Date, and Time 13](#_Toc173696893)

## Linear Search Algorithm

A computer screen shot of a program

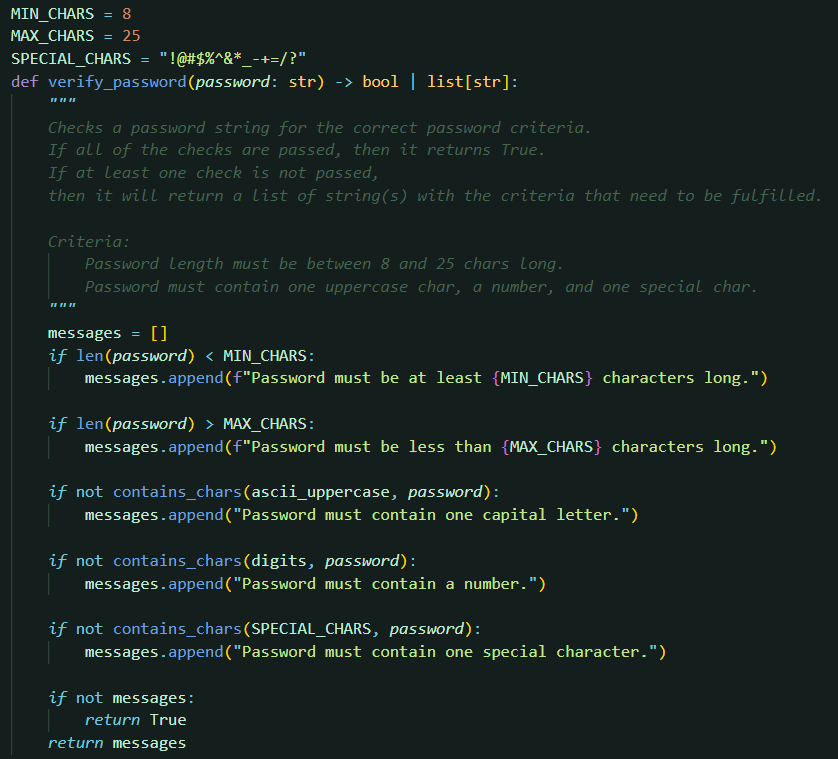
Description automatically generated

This algorithm searches a string for any number of given characters. It will return true once the first occurrence of any character is found. However, if it goes through all of the characters and none of the characters match, the method will finally return false.



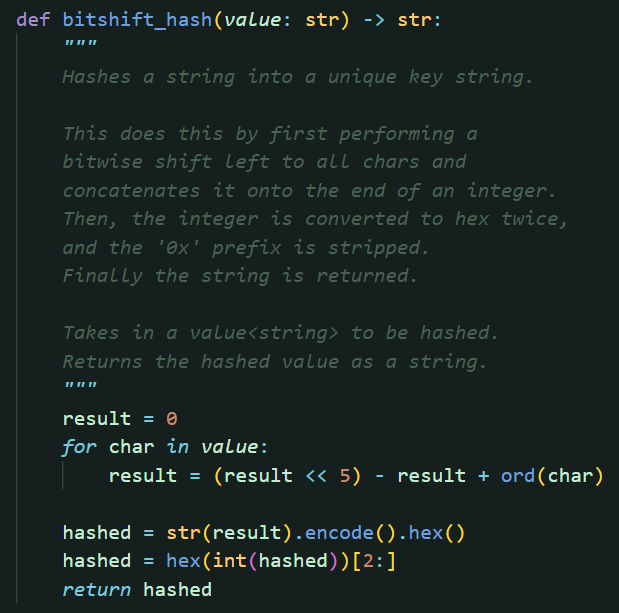
\*The same implementation in JavaScript, but to search an array of objects for an object, based on a given key/property.

## Password Check



This method is called when an account signup request is made. It makes use of the previous method by checking if the password contains both numerical and special characters. The password must meet these criteria otherwise the check will fail.

## Hash Function



The hash function is called once the password check returns true. It is used to hash the password to be stored into the user database. After the hash function has been applied, it is very difficult to be inverted. Therefore, this method is very useful, as the backend never has to unhash a password during comparison, it just has to apply the hash function the given password and compare it with hashed passwords in the database.

It makes use of a hash code computation algorithm that I found at [Hash Tables · Data Structures (maximal.io)](https://datastructures.maximal.io/hash-tables/), but modified slightly to be converted to a hexadecimal string at the end of the hash function.

The hashed password looks like this:



## Encryption and Decryption

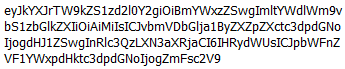
A screen shot of a computer code

Description automatically generated

These methods are complements of each other. They are called during the accessing and modification of user settings. Encryption differs from the hash function previously mentioned, as the encrypted string can then be decrypted using its inverse function.

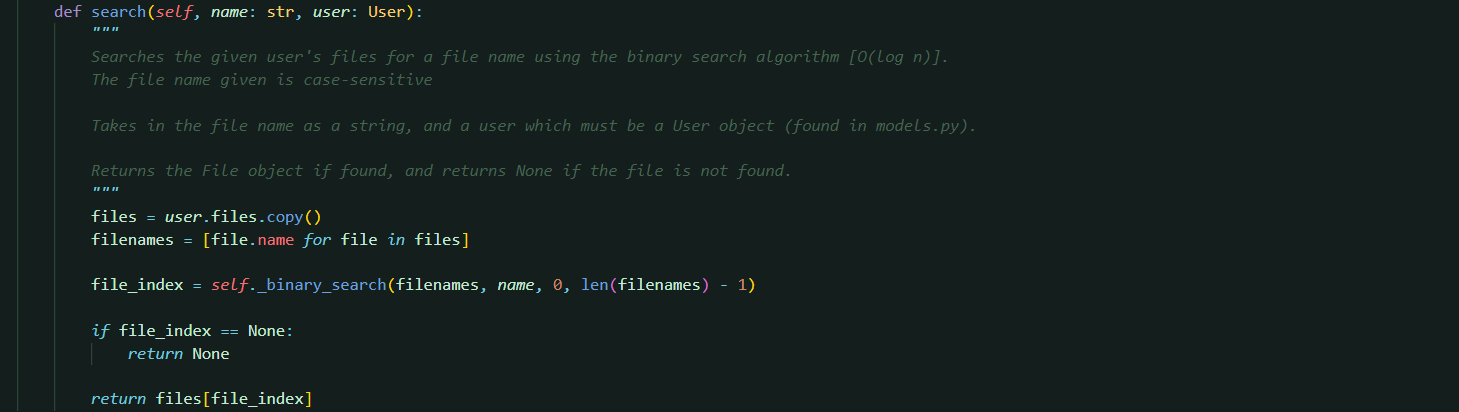
Therefore, the JSON version of a user’s settings is converted to a string and then encrypted, to be stored in the database. In future, It would be a better practice to use more ambiguous cryptography functions (and also to not store it on a public repository) such that user settings are much harder to access.

The encrypted user settings look like this:



## Binary Search Algorithm

A computer screen shot of text

Description automatically generated 

The search method is called whenever the backend queries individual files by their file name, such as getting, copying, renaming, uploading, and checking for duplicate files.

Note that whenever new files are added, they are always placed in alphabetical order according to the file’s name (This is automatically done by the SQL database). Hence, the binary search algorithm is able to work as it requires the list to be sorted.

Since the files are stored as File objects, a separate ‘filenames’ list is created. Then, the binary search is performed on the list of file names to search for the given name. If it is found, it will return the index of the file name, which can then be used to index the original files list. However, if the file name is not found, then it will return none.

Strings in Python can be compared with the greater or less than operators. This is why the file names are case sensitive as making all strings lowercase will mean that storing the files in sorted order is redundant.

## Reverse an Array

A computer screen shot of a program code

Description automatically generatedA computer code with text

Description automatically generated with medium confidence

The swap function makes use of JavaScript’s array properties. Traditionally, a temporary variable is needed to do a swap. In this case, we are able to skip the additional step.

To reverse an array, an index is set to the first element of the array. A swap is performed for the indexed element and its mirror (the length of the array minus the index). This is looped until the index reaches the middle of the array. Finally, we have reversed an array in place.

## Quick Sort

A screen shot of a computer code

Description automatically generated A computer screen shot of a computer screen

Description automatically generated

This function differs from the typical quick sort algorithm as we are working with a different format of array. I have stored my files as an array of objects. Therefore, to sort by a specific property, the property must be accessed from each object and then placed in its own array (in the same order).

As the swap function performs a swap on an array in place, there must then be two swaps performed for every swap originally, to accommodate for the two arrays. The quick sort function still recursively calls itself until all subarrays are sorted.

A screen shot of a computer program

Description automatically generatedA black background with white text

Description automatically generated with medium confidence

This is my implementation of sorting by individual properties of the files. On the next page, you will see a demonstration.

### Sorted by Name, Ascending

A screenshot of a computer

Description automatically generated

Sorted by Type, AscendingA screenshot of a computer

Description automatically generated

### Sorted by Size, Descending

A screenshot of a computer

Description automatically generated

## Checking Filenames (Maintainability)

A screen shot of a computer program

Description automatically generated

Despite the listed criteria, each if statement is actually a negation of the original criterion! This is done to remove nested if statements, which can be very unreadable. Instead of checking all criteria and returning true, If a single criteria is not met, it will single false.

Readability ensures greater maintainability.

## Formatting File Size, Date, and Time

A screen shot of a computer program

Description automatically generated

To enhance user experience, the file sizes are formatted to represent one of the standard units of file size (bytes, KB, MB, GB), instead of just showing the number of bytes.

The date would usually be shown as the date and time joined together, but as many have said, less is more. So only the date is shown, and if the user hovers over the date, then a tooltip will show its time as well.