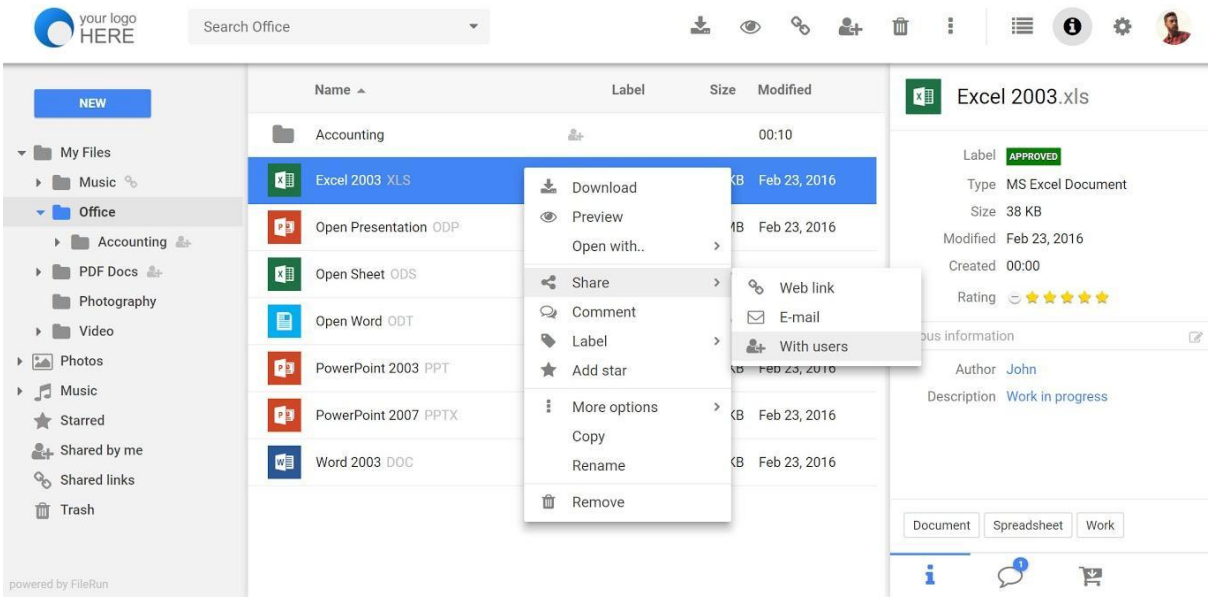
Introduction

In this project you will have to create a **system file explorer** that allows the **user** to **navigate**, **create directories** and **upload files** in the same way as he would in his usual **operating system.** The file explorer is a **tool** that allows you to directly **view and manipulate the files and directories** associated with a **path**, so you must take into account from which **path** the user starts and which path they can access.

In the following image you can see an **example** that can serve as a reference:



What are the main objectives in this project?

- Understand how the **file system works**
- Improve your knowledge in **PHP**
- Improve your knowledge in **HTML**, **CSS** & **Javascript**

- Improve your knowledge in **logic** and **programming**

1. General analysis

First of all, you will have to **analyze the project requirements**, with which you can get an idea of the **project objectives**, then **design the structure** of the application and analyze the **actions** that the user can do so that when **programming** you have clear which are the **keys points** and a previous idea of what the **user interface** will be like .

Step 1: Analyze the project requirements

In the first step you must **analyze** the **requirements** that the **project must meet** and what **actions** the **user** can do.

- **Create**, **modify** and **delete directories**
- **Browse through directories** from an **initial path**
    - The **initial path** will start from a **folder** inside the **project repository** whose name will be "**root**".
- **Search directories** and **files** by **name**
    - In the case of **searching** for **files by name**, you must also be able to specify **their extension as part of the name**.
- **Navigate through the initial path** established and all the folders created from that path. Therefore the **user will not be able** to see or **navigate** to the **parent folders of the “root” folder**.
- **Upload a file** to a directory
- **See** the following **information** of **files** and **directories**
    - Creation date
    - Last Modified Date

    - Extension (if it’s a file)
    - Size
        - If it is less than **1 MB** show **KB**, otherwise show **MB**
- **Show** the **icon** of the main **file extensions** such as:
    - doc
    - csv
    - jpg
    - png
    - txt
    - ppt
    - odt
    - pdf
    - zip
    - rar
    - exe
    - svg
    - mp3
    - mp4
- **View** the uploaded **images**.
- **Play** the uploaded **videos**.
- **Play** the uploaded **audios**.

Step 2: Design the project structure

Before starting to **develop the project** you should take into account and **analyze** the following points:

- How will the **interface** be
    - You will have to **design a wireframe** of your application taking into account the requirements.
- What **actions** can be executed by the user
    - You will have to **design a use case diagram**
- **Analyze** and **understand** what brings more value to the user
- **Analyze** how you will organize the project at the level of **directories** and **files**

Step 3: Start to develop the project

Once you have all the **designs** and **organization** of your **project** raised, you must **start developing it**.

Step 4: Extra functionalities

In addition to the **requirements** mentioned above, you can add the following **extra functionalities** so that the **user is able to**:

- **Show the information** of the uploaded **".csv"** files on the screen.
- **Move files** and **directories between folders**
- When **deleting** a **file** or **folder**, it will be moved to a **specific folder** called "**trash**", so if you want to **delete it completely** you must delete it from the mentioned folder.

2. Project organization

Next you will have to create a document where you can explain in detail how the current project is organized. It is important that it be updated throughout the life of the project. A PDF version is required within the project folder for the project documentation.

The document must include at least:

- Requirements documentation
- Wireframes
- Use case diagram
- Record of incidents that were detected during project execution
- Record of lessons learned

3. Development

Develop the **file system explorer** taking into account its **architecture** and respecting the **frontend** and **backend layers**. It is important that you keep in mind that the **organization** is very important as well as the **documentation** you provide to the project.

4. Requirements

- You **will not be able** to use **global variables** in **PHP**.
- You must use **GIT**

- You must use the **PHP > v8**
- Create a **clear** and **orderly directory structure**
- Both the **code** and the **comments** must be written in **English**
- Use the **camelCase** code style to define variables and functions
- In the case of using **HTML**, **never** use **inline styles**
- In the case of using different programming languages **always define the implementation in separate terms**
- Remember that it is important to **divide the tasks into several sub-tasks** so that in this way you can associate each particular step of the construction with a **specific commit**
- You should try as much as possible that the **commits** and the **planned tasks** are the same
- **Delete files** that are not used or are not necessary to evaluate the project

5. Deliverables

To evaluate the project you will need the following deliveries:

- Forked repository with code: https://github.com/assembler-institute/filesystem-explorer
- You must create a correctly documented README file in the root directory of the project (see guidelines in **Resources**)
- Project documentation in PDF format

- Presentation in PDF format explaining:
    - Comparison of the original design (Wireframe) with the final result of the project
    - Comparison of the use case diagram with the actions that the user can finally perform

    - Comparison of the original use case diagram with the final user actions
    - What lessons you've learned during this project
    - What problems have you encountered when developing this project?
    - How you have organized and distributed the tasks

6. Resources

- File system: https://es.wikipedia.org/wiki/Administrador_de_archivos
- PHP FileSystem W3C:
  https://www.w3schools.com/php/php_ref_filesystem.asp
- PHP FileSystem [Oficial]:
  https://www.php.net/manual/es/book.filesystem.php
- README Guidelines Example:
  https://gist.github.com/PurpleBooth/109311bb0361f32d87a2

6.1. Wireframes web tools

- Moqups: https://app.moqups.com/
- Draw.io: https://app.diagrams.net/
- Cacoo: https://cacoo.com/es/