

OS Project Final Report

- **Carol Michail 900194282**
- **Sara Naseef 900202369**
- **Ahmed El Zoghby 900204333**

Features and Use Cases :

- Move in and out of Directories
- Fetch the Size of Directories
- Visually display the contents of Directories as well as their respective sizes
- Enter Specific Directories for Analysis
- Delete Directories

Illustrations:

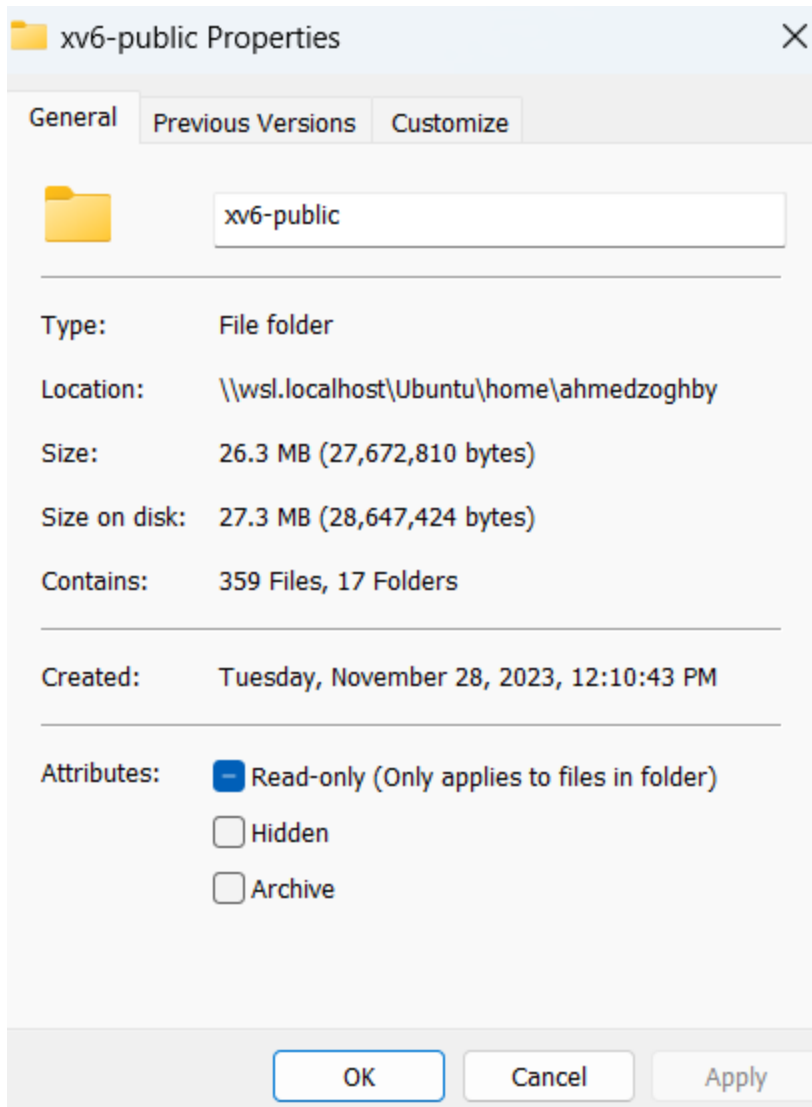
Enter Directory :

Using GTK a textbox is declared this text box can be manually filled or automatically by clicking on a visible Folder on the Screen.

Fetch Size :

The file scanners.rs provides the code to calculate the size of our current directory or file. It is imported to our main program where it is modelled to a button upon clicking on the button the size is returned. Below is an example where both the properties section, as well as the app, showcase the same folder size for xv6.

```
addDirectory.connect_clicked(move |_| {  
    let path = textClone.text().to_string();  
    let path2 = Path::new(&path);  
    if path2.is_dir() {  
        let size = scanner::display_total_size(&path2)  
        let label_text = format!("Size: {}", size);  
        directoryLabelClone.set_text(&label_text);  
    } else {  
        let size = scanner::displayFileSize(&path2);  
        let label_text = format!("Size: {}", size);  
        directoryLabelClone.set_text(&label_text);  
    }  
});
```



Enter Directory:
/home/ahmedzoghby/xv6-public
Size: 26.39 MB

Delete :

The file `delete_file.rs` has the functionality for deleting a file it is imported to our main program and modelled to a button upon clicking the button the file or folder available in our text box is deleted. An example is provided below where an Excel file "Delete Me" is deleted upon pressing the button Delete

```

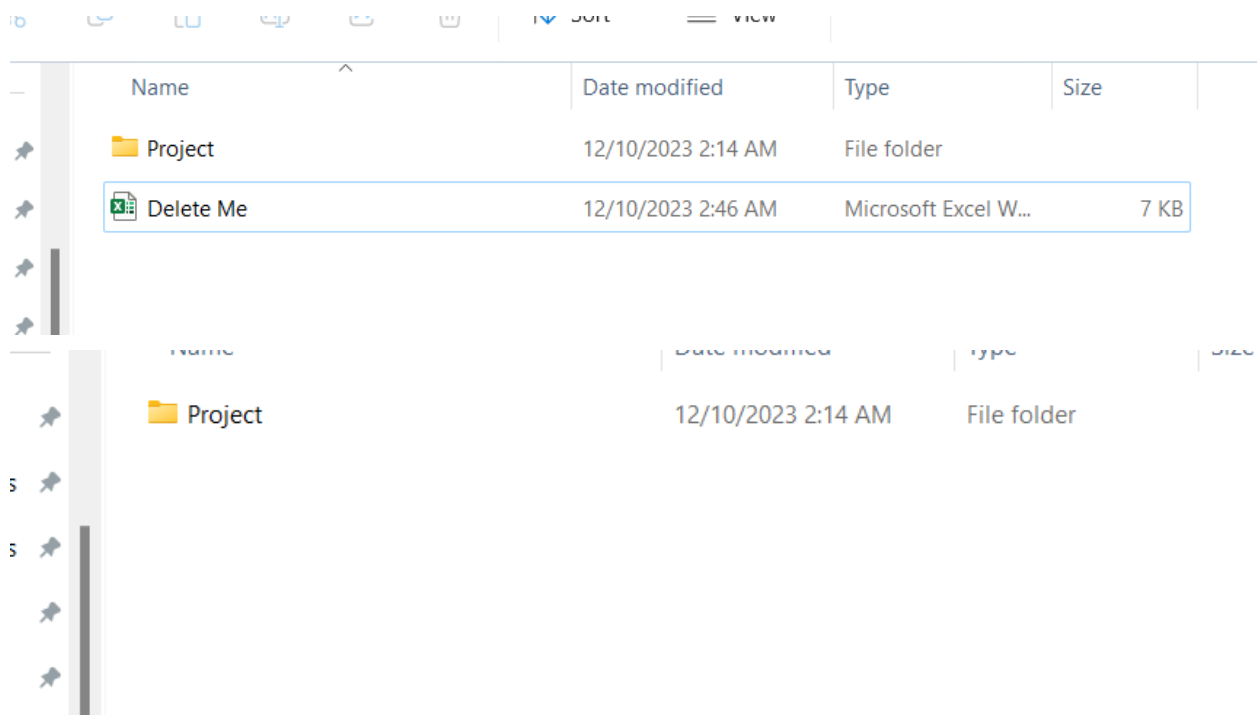
deleteButton.connect_clicked(move |_| {
    let path = textBoxCloneDelete.text().to_string();
    let path2 = Path::new(&path);
    if path2.is_dir() {
        println!("This is a folder");
        //delete_file::delete_folder(&path2);

    }
    else{
        println!("This is a file");

        delete_file::delete_file(&path2);

    }
});

```



The screenshot shows a file explorer window with a sidebar on the left containing several folder icons. The main pane displays a table of files and folders. The table has columns for Name, Date modified, Type, and Size. The first row shows a folder named 'Project' with a date of 12/10/2023 2:14 AM and type 'File folder'. The second row shows a file named 'Delete Me' with a date of 12/10/2023 2:46 AM, type 'Microsoft Excel W...', and size '7 KB'. The 'Delete Me' file is selected, indicated by a blue border around its row.

Name	Date modified	Type	Size
Project	12/10/2023 2:14 AM	File folder	
Delete Me	12/10/2023 2:46 AM	Microsoft Excel W...	7 KB

Display Contents :

Upon clicking this button 2 vectors of buttons are created one for the folders and another for the files. After iterating through the contents of the current folder which is

written in the text box. Buttons are created and displayed. Upon clicking on any of the newly formed buttons the text available in our text box changes allowing us to move into another directory or delete it. Also, a Pie chart is presented which showcases the sizes of the files/folders as a percentage of the total directory size.

```

display.connect_clicked(move |_| {
    let path = textClone2.text().to_string();
    let path2 = Path::new(&path);
    let data: Vec<(PathBuf, String, u64)> = scanner::display_file_sizes(&path2);

    let mut folder_buttons: Vec<Button> = Vec::new();
    let mut file_buttons: Vec<Button> = Vec::new();
    let mut totalSize = scanner::display_total_size_unformatted(&path2);

    let d2 = data.clone();
    let iClone1 = iClone.clone();

    for entry in data {
        if entry.1 == "Folder" {
            let new_button: Button = gtk::Button::builder()
                .label(&format!("Folder: {}", entry.0.file_name().unwrap().to_string_lossy()))
                .build();

            let s = sizeClone.clone();
            let t = textClone4.clone();
            let mainClone = mainBoxClone.clone();
            let l = directoryLabelClone1.clone();
            let textB = textClone4.clone();
            let addDir = addClone.clone();
            let delete = delClone.clone();
            let remove = removeButtonClone.clone();
            let dis = displayClone.clone();
            let iClone2 = iClone1.clone();

            new_button.connect_clicked(move |_| {
                t.set_text(&entry.0.display().to_string());
                unsafe {
                    while (mainClone.last_child().is_some()) {
                        mainClone.remove(&mainClone.last_child().unwrap());
                    }
                }
            })
        }
    }
}

```

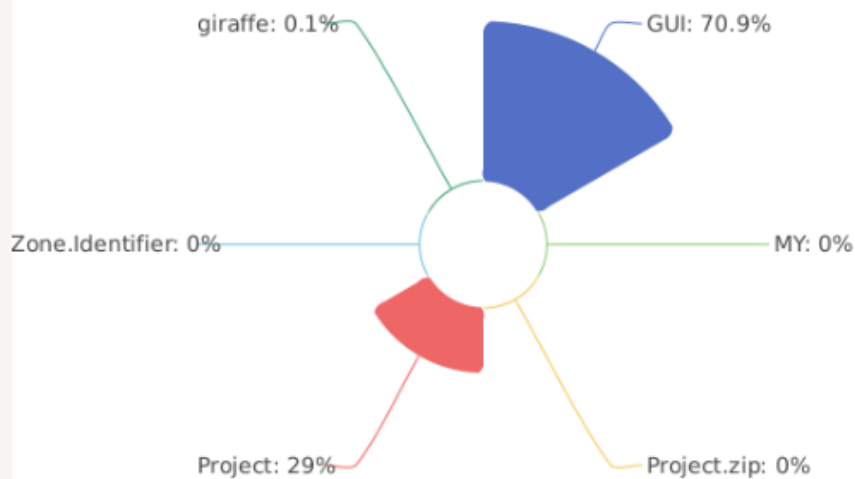
```

        folder_buttons.push(new_button);
    } else if entry.1 == "File" {
        let new_button: Button = gtk::Button::builder()
            .label(&format!("File: {}", entry.0.file_name().unwrap().to_string_lossy()))
            .build();
        let t = textClone4.clone();
        let mainClone = mainBoxClone.clone();
        let l = directoryLabelClone1.clone();
        let textB = textClone4.clone();
        let addDir = addClone.clone();
        let delete = delClone.clone();
        let remove = removeButtonClone.clone();
        let dis = displayClone.clone();
        let s = sizeClone.clone();
        let iClone2 = iClone1.clone();

        new_button.connect_clicked(move |_| {
            t.set_text(&entry.0.display().to_string());
            unsafe {
                while (mainClone.last_child().is_some()) {
                    mainClone.remove(&mainClone.last_child().unwrap());
                }
            }
        })
    }
}

```

Disk Analyzer



Enter Directory:
/home/ahmedzoghby/OSProject
Size: 26.39 MB
Fetch Size
Delete
Display Contents
Back
Folder: GUI
Folder: MY
Folder: Project
Folder: giraffe
File: Project.zip
File: Project.zip:Zone.Identifier

	Name	Date modified	Type	Size
	giraffe	12/9/2023 12:49 PM	File folder	
ds	GUI	10/29/2023 3:42 AM	File folder	
nts	MY	12/7/2023 7:35 PM	File folder	
	Project	12/10/2023 2:47 AM	File folder	
	Project	12/10/2023 2:08 AM	Compressed (zipp...	13 KB
	Project.zip· Zone.Identifier	12/10/2023 2:08 AM	IDENTIFIER File	1 KB

Back :

Upon clicking on the button back the section of the text in the textbox after the \ will be deleted. Ultimately allowing us to backtrack to previous directories.


```

removeButton.connect_clicked(move |_| {
    let text = textBoxCloneback.text().to_string();

    let t = textBoxClone3.clone();
    let l = directoryLabelClone2.clone();
    let textB = textBoxCloneback.clone();
    let addDir = addDirectoryClone.clone();
    let delete= deleteButtonClone.clone();
    let dis = displayCl.clone();
    let iClone = imageClone.clone();

    if let Some(index) = text.rfind('/') {
        let modified_text = &text[0..index];
        textBoxCloneback.set_text(modified_text);
    } else {
        textBoxCloneback.set_text("");
    }
}

```

Enter Directory:
/home/ahmedzoghby/OSProject/Project
Size: 26.39 MB
Fetch Size
Delete
Display Contents
Back

Enter Directory:
/home/ahmedzoghby/OSProject
Size: 26.39 MB
Fetch Size
Delete
Display Contents
Back

Justification Based on the Survey:

We tried to mix all the positive features we found during our survey period

1 - we opted for a Pie chart similar to the one on Baobab rather than the bar graph approach in QDirStat

2- We opted for a delete file feature available in ncd and QDirStat but not available in Baobab

3- Also all of the applications studied during our survey could move in and out of directories, display the files and folders inside a directory, and fetch sizes of directories so implementing such features was essential.

Needed Libraries :

```
[dependencies]
gtk4 = "0.7.3"
charts-rs = "0.2.0"
```

Gtk4 is used for the UI implementation

Charts-rs is used for the Pie chart

Compilation Procedure :

After entering the correct directory where our code is stored a cargo run command will initiate the program.

```
ahmedzoghby@LAPTOP-J0NC2K0C:~/OSProject/Project/Project$ ls
Cargo.lock      delete_file.rs      main.rs
Cargo.lock:Zone.Identifier  delete_file.rs:Zone.Identifier  main.rs:Zone.Identifier
Cargo.toml      directory_operations.rs  scanner.rs
Cargo.toml:Zone.Identifier  directory_operations.rs:Zone.Identifier  scanner.rs:Zone.Identifier
chart.svg       'main (1).rs:Zone.Identifier'  target
chart.svg:Zone.Identifier  'main (1).rs:Zone.Identifier:Zone.Identifier'
```

Contributions:

Sara Naseef : implemented the code for the following functionalities : Delete file , Scanning operations, and all the directory operations. Focus on Backend and provided support for the Frontend

Carol Michail: implemented the pie chart feature, as well as the UI of the Application. Focus on Frontend and provided support for Backend

Ahmed El Zoghby: implemented the moving out of directories feature, as well as the UI of the Application.