

# FILE EXPLORER

## PYTHON

Harshita Balaji (22BAI1038)  
Venna Divya Reddy (22BAI1476)  
Sanjana Suresh (22BAI1439)  
Mirdulaa Balaji (22BAI1284)





# TABLE OF CONTENTS

01 ●

**INTRODUCTION**

02 ●

**Designing the  
User  
interface(UI)**

03 ●

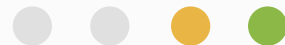
**File navigation**

04 ●

**Basic File  
Operations**

05 ●

**Error Handling**





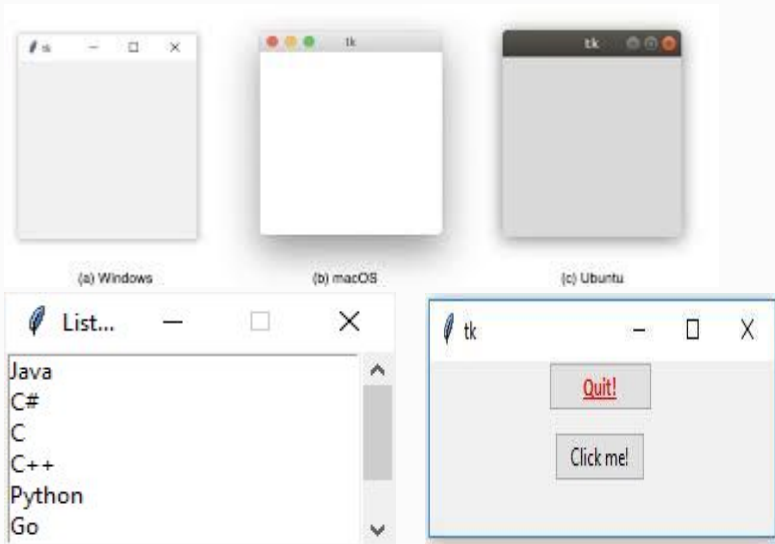
# INTRODUCTION

File explorer is like a digital map for organizing and managing computer files. Here, in this project we will be utilising the in built os module of python in order to access files of the system.

Tkinter , an inbuilt python library will serve as the graphical interface of the file explorer



# Designing the UI interface



The process begins by creating the main application window as the container for UI elements.

We design various widgets like labels, buttons, and listboxes by creating instances of Tkinter widget classes and configure their appearance and behavior




# FILE NAVIGATION

In this project, file navigation is achieved through a series of steps:

1. **List Directory Contents:** To display the files and folders in the current directory, a `list_directory` function is implemented. This function uses the `os.listdir` method to obtain a list of all items in the current directory. It filters out items that start with a dot (usually indicating hidden files and folders), and returns the result.
2. **User Interaction:** Users can interact with the file explorer's user interface, which includes a listbox that displays the files and folders. They can click on folders in the listbox to navigate into subdirectories.
3. **Folder Double-Click Event:** A key part of the navigation functionality is the double-click event handling. When a user double-clicks on a folder in the listbox, the `navigate_directory` function is called. This function retrieves the name of the folder that was double-clicked and creates a new path by joining the current path with the selected folder name.
4. **Updating Display:** The `refresh_display` function is responsible for updating the displayed file list with the contents of the new path. It sets the current path to the new path and refreshes the listbox to show the files and folders in the new directory.

The combination of these steps allows users to navigate through directories in the file explorer, exploring and interacting with files and folders as they go deeper into the directory structure





## BASIC FILE OPERATIONS

In this project, we are including three basic file operations:

**Copy:** Allowing users to create a copy of a selected file in the same directory with a modified name, e.g., appending "Copy\_" to the file name.

**Move:** Allowing users to move a selected file to the same directory with a modified name, e.g., appending "Move\_" to the file name.

**Delete:** Allowing users to delete a selected file.






## BASIC FILE OPERATIONS

**Journaling:** Maintaining an extensive log of operations done by the user.

**File Retrieval:** Allowing the user to retrieve files deleted or delete them permanently.

These operations provide essential file management capabilities and are commonly found in file explorer applications.



# ERROR HANDLING

Error handling in the provided code is primarily done through the use of try-except blocks and the messagebox module from

This error handling ensures that if there are issues with file operations (e.g., file not found, permission denied), the user is informed about the problem through an error message dialog.



**THANK YOU**