**INTERNATIONAL INSTITUTE OF PROFFESIONAL STUDIES**

**Devi Ahilya Vishwavidyalaya**

**Indore, M.P**

Session: July-Dec, 2020

A

Project Report on

File Sorting Application

Submitted By:

Aakanksha Solanki (IC-2K16-01)

Under the guidance of:

Mr. Jugendra Dongre Sir

### **Project report on:**

### **File Sorting Application**

### **Dissertation Approval Sheet**

The dissertation entitled ***File Sorting Application*** being Submitted by **Aakanksha Solanki (IC-2K16-01)** in partial fulfillment of the requirement for the award of IX Semester in Master of Computer Application (6 yrs), to IIPS, DAVV is satisfactory and approved.

­­­­­­­­­­­­­­

External Examiner

­­­­­­­­­­­­­­

Internal Examiner

### **Declaration**

I hereby declare that the Project entitled “**File Sorting Application**” which is submitted by me for the partial fulfillment of the requirement for the award of IX Semester in Master of Computer Application (6 yrs), International Institute Of Professional Studies, Devi Ahilya Vishwavidyalaya, Indore. The work has not been submitted anywhere else and comprises of our own workand due acknowledgement has been made to in text to all other material used.

**Name Of Student** : Aakanksha Solanki

**Roll Number** : IC-2K16-01

**Date** : 23-02-2021

### **Table of Content**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Topic** | **Page No.** |
| 01 | Introduction | 6-7 |
| 02 | Background Analysis | 8 |
| 03 | Process Description | 9-11 |
| 04 | About Python | 12-17 |
| 05 | System Designing and Testing | 18-24 |
| 06 | Coding | 25-38 |
| 07 | Output Screenshots | 39-42 |
| 08 | Further Improvements | 43 |
| 09 | Conclusion | 44 |
| 10 | Bibliography | 45 |

**INTRODUCTION**

This file sorting application is made with Python Technology. This application is used to sort the folder of your computer which contains many files like images, audios, videos, etc having different extensions. This application is based on real life problem which is faced by many users working with different files and folders.

Sometimes if we forget name of any file like image audio or video then it becomes a bit difficult to search it in any folder which contains many files with different extensions. Also searching it becomes time consuming. So this file sorting application helps user to sort all the files according to their extensions and place them into newly created folders containing files of only one type of extension.

For example a folder A contains thousands of file with different extensions then that application will create folders like image audio video and document inside the folder A and will please all the files with having extension JPG, PNG, JPEG and rest of the extensions of image into image folder the MP3 files into audio folder the MP4 files into video folder and all the documents like PDF text files and other files into document folder and extra files like zip file for Rar file into the folder named extra.

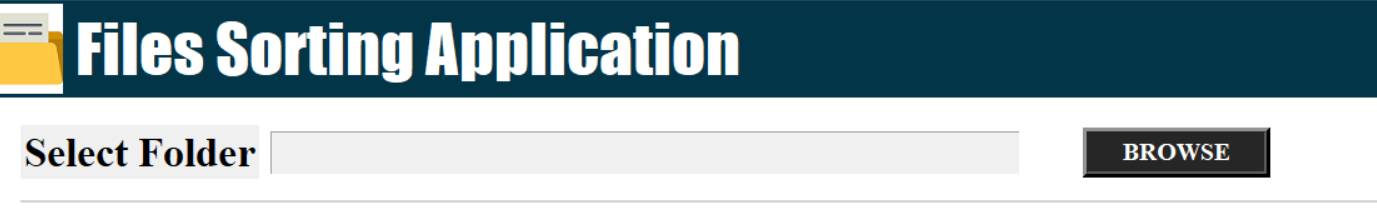
This application does not take much time to run and it sorts your folder in couple of seconds. This saves time of the user and makes it easy for any user to deal with files of different extensions. Also any user did not need to do it manually for sorting any folder. It has an easy user interface with which any user can work easily.

**Background Analysis**

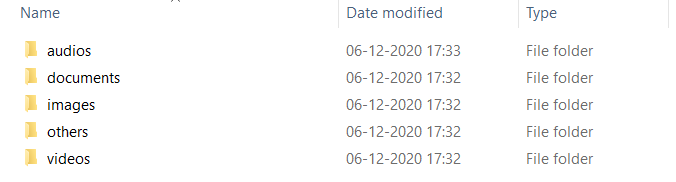
The File Sorting Application has been built up after seeing the real life problem which is faced by many users who are always dealing with different types of file with different extensions. It sometimes takes a lot of time for user to sort all the images, audios or videos of any folder. So this will allow us something that we were waiting for so long. The relief will be huge and so will be the benefit. Hopefully this will give us the benefit that we are expecting. The world is moving to internet so this is the right time to think about this. If we compare the benefits and the satisfaction from every point of view then we cannot conclude things beneficial for all. If a system does not provide help for every user group then it cannot be perfect system. I came out a long way with maximizing benefit for all. Still there are a lot of things that can be added but at this point this is the most I can do for making up things easy and less time consuming.

**Process Description:**

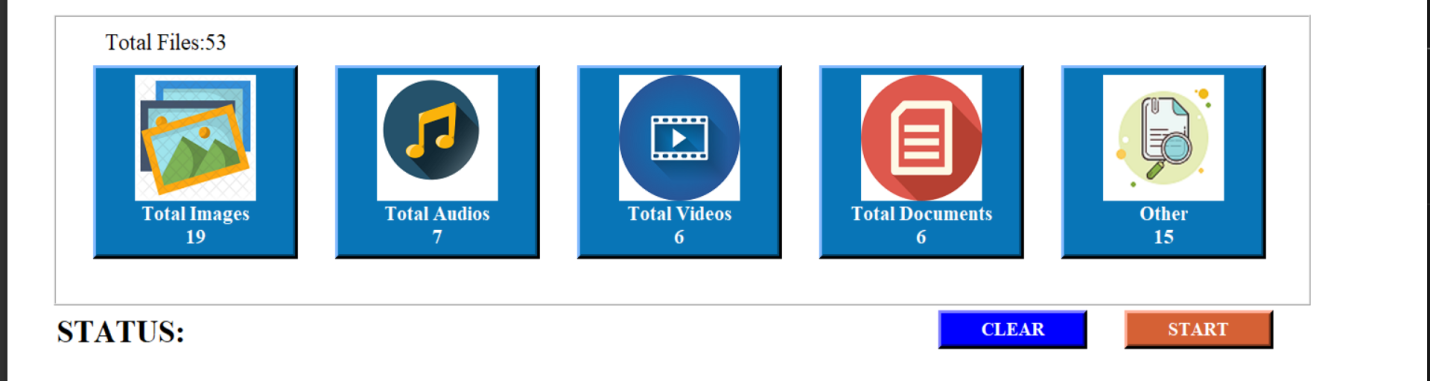
1. As soon as the user runs the application, that application will welcome the user by greeting him or her.
2. It firstly ask user for the path of the folder which he/she wants to sort, the user can select the path of the folder by clicking on the browse button.



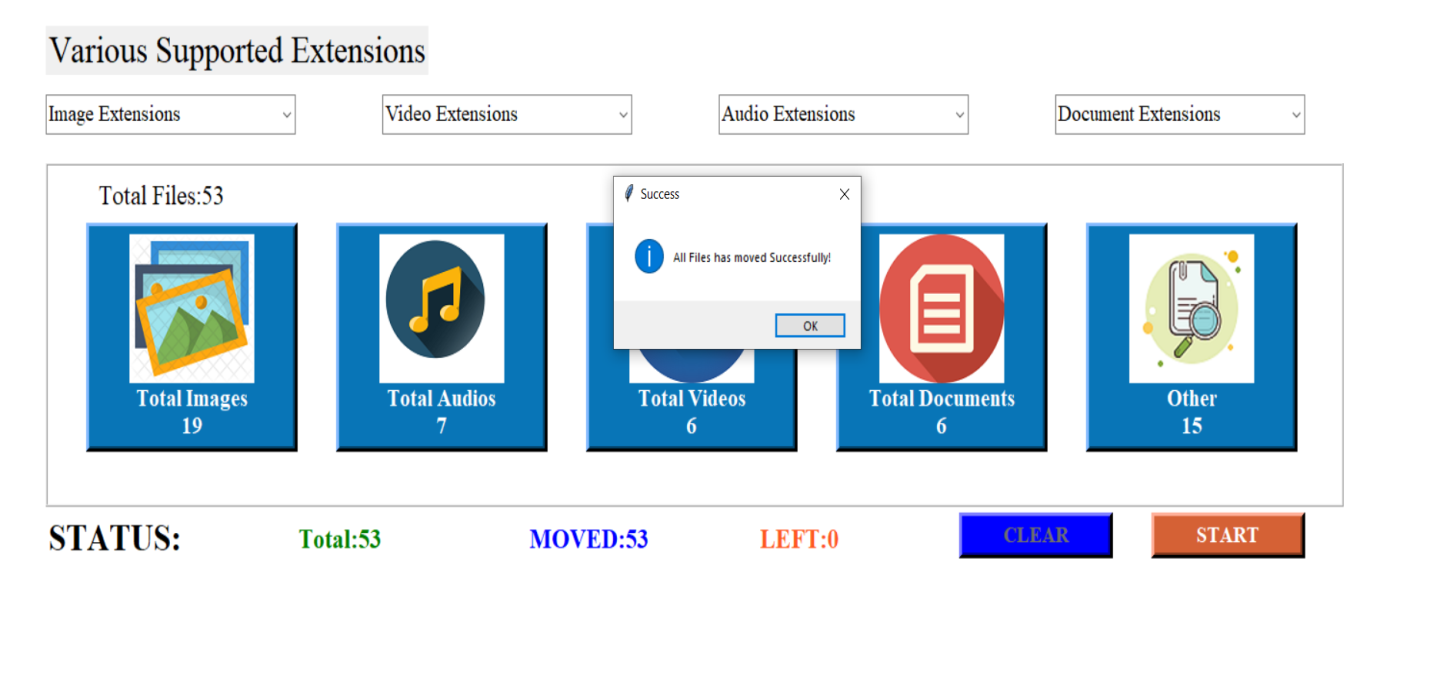
1. After the start button is clicked by user, the application will create some new folders like Images, Audios, Videos, Documents, etc into the given folder.



1. After creating different folders, it will calculate and tell the user about the total number of files and total number of images, audios, videos and documents that were present in the given folder.



1. Now user needs to click on the start button which will enable only after the user gives the path. After the user clicks on the start button the files moves into their respective folders according to their extensions. After all the files have moved successfully a popup success message appears on the screen and application tells us the status about total files moved files and left files.

****

### **Programming Language Used:**

### **Python**

**Why Python:**

Python is a high level language which is easy to understand and is portable. Its large and robust standard library makes Python score over other programming languages. It is a great general-purpose language. The standard library allows you to choose from a wide range of modules according to your precise needs. Each module further enables you to add functionality to the Python application without writing additional code. Python programmer wastes no time declaring the types of arguments or variables, and Python's powerful polymorphic list and dictionary types, for which rich syntactic support is built straight into the language, find a use in almost every Python program. Python has dynamic typing and binding, and everything in Python is an object. However, Python distinguishes built-in object types from user-defined classes.

**Python over Java:**

Python programs are generally expected to run slower than Java programs, but they also take much less time to develop. Python programs are typically 3-5 times shorter than equivalent Java programs. This difference can be attributed to Python's built-in high-level data types and its dynamic typing. For example, a Python programmer wastes no time declaring the types of arguments or variables, and Python's powerful polymorphic list and dictionary types, for which rich syntactic support is built straight into the language, find a use in almost every Python program. Because of the run-time typing, Python's run time must work harder than Java's.

**Python over C++:**

Almost everything said for Java also applies for C++, just more so: where Python code is typically 3-5 times shorter than equivalent Java code, it is often 5-10 times shorter than equivalent C++ code! Anecdotal evidence suggests that one Python programmer can finish in two months what two C++ programmers can't complete in a year. Python shines as a glue language, used to combine components written in C++.

**Modules Used:**

A Module refers to a file containing Python statements and definitions. A file containing Python code, for example: example.py is called a module, and its module name would be example. We use modules to break down large programs into small manageable and organized files. Furthermore, modules provide reusability of code. We can define our most used functions in a module and import it, instead of copying their definitions into different programs.

1. **Tkinter** : Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Itis a set of wrappers that implement the Tk widgets as Python classes.
2. [**Ttk**](https://docs.python.org/3/library/tkinter.ttk.html#module-tkinter.ttk): The [tkinter.ttk](https://docs.python.org/3/library/tkinter.ttk.html#module-tkinter.ttk) module provides access to the Tk themed widget set, introduced in Tk 8.5. If Python has not been compiled against Tk 8.5, this module can still be accessed if Tile has been installed.
3. **FileDialog** : tkFileDialog dialog allows a user to select a file from the filesystem.
4. **Messagebox**: Message boxes are convenient dialogs that provide messages to the user of the application. The message consists of text and image data. Message boxes in Tkinter are located in the tkMessageBox module.
5. **OS** **module**: The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. The \***os**\* and \***os**. path\* modules include many functions to interact with the file system.
6. **Shutil** **module**: The [shutil](https://docs.python.org/3/library/shutil.html#module-shutil) module offers a number of high-level operations on files and collections of files. In particular, functions are provided which support file copying and removal. For operations on individual files, see also the [os](https://docs.python.org/3/library/os.html#module-os) module.
7. **pyttsx3**: pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.You need to install in by using #pip install pyttsx3.
8. **Python Speech Recognition module**: Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. You need to install in by using #pip install speechrecognition.
9. **Datetime module**  **:**Datetimemodule supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

**System Design of File Sorting Application:**

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client’s requirements into a logically working system. Normally, design is performed in the following two steps:

**1. Primary Design Phase:**

In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks.

**2. Secondary Design Phase:**

In the secondary phase the detailed design of every block is performed.

**USER INTERFACE DESIGN**

User Design Interface is concerned with the dialogue between a user and the application. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue

**THE FOLLOWING ARE VARIOUS GUIDELINES FOR UI DESIGN:**

1. The screen should be formatted so that various types of information and messages always appear in the same general display screen.

2. Default values for fields and answers to be entered by user should be specified.

3. A user should not be allowed to proceed without correcting an error.

**Security testing of the project**

Testing is vital for the success of any software no system design is ever perfect testing is also carried into phases. First phase is during the software engineering that is during the module creation Second phase is after the completion of software this is system testing which verifies that the whole set of programs are handed together.

1. **White box testing**: in this technique the close examination of the logical parts through the software attested by cases that exercise species set of conditions or loops. Illogical parts of the software can do one error that can be corrected using this technique or typographical errors logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. When the box testing best on the independent bad within a month ecological decision on the new and the false side exercised.
2. **Black Box Testing**: This method enables the software engineer to device that of input technique that fully exercise all functional requirements for a program black box testing test the input the output and the external data it checks whether the input data is correct and whether we are getting the desired output or not.
3. **Alpha Testing**: Acceptance testing is also sometimes called Alpha testing. Alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirement.
4. **Beta Testing**: On the other hand when a system is to be marked as a software product another process cal beta testing is conducted during beta testing a system is delivered among a number of potential users who agreed to use it the customers that report problem to the developers thus provided the product for real use and detect error which may not have been anticipated by the system developer.
5. **Unit Testing**: Each module is considered independently it focuses on each unit of software as implemented in the source code it is white box testing.
6. **Integration Testing**: Integration testing AIMS at constructing the program structure while at the same construct and test to uncover error associated with interface in the module are integrated by using the top down approach.
7. **Validation Testing**: Validation testing was performed to ensure that all the functional and professional requirements are met.
8. **System Testing**: Executive programs to check logical changes made in it with intention of finding errors a system is tested for online response volume of transactions recovery from failure except system testing is done to ensure that the system satisfies all the user requirements.

**The steps involved during system testing are as follows:**

1. Integration of all the modulus/forms in the system.
2. Preparation of the test cases.
3. Preparation of the possible test data with all the validation checks.
4. Actual testing done manually.
5. Recording of all the reproduced errors.
6. Modifications done for the errors found during testing.
7. Prepared the test result script after rectification of the errors.

**Implementation and software specification tastings:**

**Detailed design of implementation:** This phase of the system development life cycle refined hardware and software specifications establishment programming plans, trains users and implement extensive testing procedures, to evaluate decide Operating specifications and/or provide the basis for the modifications.

**Technical design:** The activated builds upon specifications produced during new system design, and detailed technical specifications and documentation.

**Best specifications and planning:** This activity prepares detailed test specifications for entries for all models and programs, jobs streams, subsystems, and for the system as a whole.

**Programming and testing:** This activity and composes actual development, writing, and testing of the program units or modulus.

**User training:** Activity in composes writing use a procedure manual preparation of user training materials conducting training programs, and testing procedures.

**Acceptance test:** A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

**Installation phase:** Industries the new computerized system is installed the conversion to the new procedure is fully implemented, and the potential of the new system it explored.

**System installation:** The process of starting the actual use of system and training use of personnel in this operation.

**Review phase:** This place value it's the successes and failures during a system development project and to measure the results of the new computer is to transfer system and terms of benefits and saving projected at the start of the project.

**Development recap:** Review of a project immediately after completion to find successes and potential problems and future work.

**Post implementation review:** Are you conducted after a new system has been in operation for some time to evaluate actual system performance against original Expectations and projections for cost-benefit improvements also identifies maintenance projects to enhance or improve the system

**Coding**

from tkinter import\*

from tkinter import ttk,filedialog,messagebox

import os,shutil

import pyttsx3 #pip install pyttsx3

import speech\_recognition as sr #pip install speechRecognition

import datetime

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

# print(voices[1].id)

engine.setProperty('voice', voices[1].id)

def speak(audio):

engine.say(audio)

engine.runAndWait()

def wishMe():

hour = int(datetime.datetime.now().hour)

if hour>=0 and hour<12:

speak("Good Morning!")

elif hour>=12 and hour<18:

speak("Good Afternoon!")

else:

speak("Good Evening!")

speak("Welcome to Files Sorting Application")

class Sorting\_App:

def \_\_init\_\_(self,root):

self.root=root

self.root.title("Sorting Application")

self.root.geometry("1525x790+0+0")

self.root.config(bg="white")

self.logo\_icon=PhotoImage(file="F:\\aakanksha\\chinchu\\project\\images\\folder.png")

title=Label(self.root,text="Files Sorting Application",padx=10,image=self. Logo\_icon,compound=LEFT,font=("impact",40),bg="#023548",fg="white" ,anchor="w").place(x=0,y=0,relwidth=1)

#-------------Section one----------------#

self.var\_foldername=StringVar()

lbl\_select\_folder=Label (self.root,text="Select Folder",font=("times new roman", 25,"bold")).place(x=50,y=100)

txt\_folder\_name=Entry(self.root,textvariable=self.var\_foldername,font=("times new roman",15),state='readonly',bg="lightyellow").place(x=250,y=105, height=35,width=600)

btn\_browse=Button(self.root,command=self.browse\_function,text="BROWSE",bd=5,font=("times new roman",15,"bold"),bg="#262626",fg="white", activebackground="#262626",cursor="hand2",activeforeground="white").place(x=900,y=102,height=40,width=150)

hr=Label(self.root,bg="lightgrey").place(x=50,y=160,height=2,width=1350)

#-------------Section two----------------#

self.image\_extensions=["Image Extensions",".png",".jpg",".jpeg"]

self.audio\_extensions=["Audio Extensions",".wav",".mp3"]

self.video\_extensions=["Video Extensions",".mp4"]

self.doc\_extensions=["Document Extensions",".doc",".xls",".txt",".pdf", ".zip",".rar"]

self.folders={

'videos':self.video\_extensions,

'audios':self.audio\_extensions,

'images':self.image\_extensions,

'documents':self.doc\_extensions,

}

lbl\_support\_ext=Label(self.root,text="Various Supported Extensions", font= ("times new roman",25)).place(x=50,y=200)

self.image\_box=ttk.Combobox(self.root,values=self.image\_extensions,font=("times new roman",15),state='readonly')

self.image\_box.place(x=50,y=260,width=260,height=35)

self.image\_box.current(0)

self.video\_box=ttk.Combobox(self.root,values=self.video\_extensions,font=("times new roman",15),state='readonly')

self.video\_box.place(x=400,y=260,width=260,height=35)

self.video\_box.current(0)

self.audio\_box=ttk.Combobox(self.root,values=self.audio\_extensions,font=("times new roman",15),state='readonly')

self.audio\_box.place(x=750,y=260,width=260,height=35)

self.audio\_box.current(0)

self.doc\_box=ttk.Combobox(self.root,values=self.doc\_extensions,font=("times new roman",15),state='readonly')

self.doc\_box.place(x=1100,y=260,width=260,height=35)

self.doc\_box.current(0)

#-------------Section three----------------#

self.image\_icon=PhotoImage(file="F:\\aakanksha\\chinchu\\project\\images\\image.png")

self.audio\_icon=PhotoImage(file="F:\\aakanksha\\chinchu\\project\\images\\music.png")

self.video\_icon=PhotoImage(file="F:\\aakanksha\\chinchu\\project\\images\\video.png")

self.document\_icon=PhotoImage(file="F:\\aakanksha\\chinchu\\project\\images\\docu.png")

self.other\_icon=PhotoImage(file="F:\\aakanksha\\chinchu\\project\\images\\other.png")

Frame1=Frame(self.root,bd=2,relief=RIDGE,bg="White")

Frame1.place(x=50,y=320,width=1350,height=300)

self.lbl\_total\_files=Label(Frame1,text="Total Files:",font=("times new roman",18),bg="white")

self.lbl\_total\_files.place(x=50,y=10)

self.lbl\_total\_image=Label(Frame1,bd=5,relief=RAISED,image=self.image\_icon,compound=TOP,font=("times new roman",16,"bold"),bg="#0875B7",fg="white") self.lbl\_total\_image.place(x=40,y=50,width=220,height=200)

self.lbl\_total\_audio=Label(Frame1,bd=5,relief=RAISED,image=self.audio\_icon,compound=TOP,font=("times new roman",16,"bold"),bg="#0875B7",fg="white")

self.lbl\_total\_audio.place(x=300,y=50,width=220,height=200)

self.lbl\_total\_video=Label(Frame1,bd=5,relief=RAISED,image=self.video\_icon,compound=TOP,font=("times new roman",16,"bold"),bg="#0875B7",fg="white")

self.lbl\_total\_video.place(x=560,y=50,width=220,height=200)

self.lbl\_total\_document=Label(Frame1,bd=5,relief=RAISED,image=self.document\_icon,compound=TOP,font=("times new roman",16,"bold"),bg="#0875B7",fg="white")

self.lbl\_total\_document.place(x=820,y=50,width=220,height=200)

self.lbl\_total\_other=Label(Frame1,bd=5,relief=RAISED,image=self.other\_icon,compound=TOP,font=("times new roman",16,"bold"),bg="#0875B7",fg="white")

self.lbl\_total\_other.place(x=1080,y=50,width=220,height=200)

#-------------Section four----------------#

lbl\_status=Label(self.root,text="STATUS:",font=("times new roman",25,"bold") ,bg="white").place(x=50,y=625)

self.lbl\_st\_total=Label(self.root,text="",font=("times new roman",18,"bold") ,bg="white",fg="green")

self.lbl\_st\_total.place(x=310,y=632)

self.lbl\_st\_moved=Label(self.root,text="",font=("times new roman",18,"bold") ,bg="white",fg="blue")

self.lbl\_st\_moved.place(x=550,y=632)

self.lbl\_st\_left=Label(self.root,text="",font=("times new roman",18,"bold") ,bg="white",fg="#ff5722")

self.lbl\_st\_left.place(x=790,y=632)

#------------BUTTONS-----------------#

self.btn\_clear=Button(self.root,text="CLEAR",command=self.clear,bd=5,relief=RAISED,font=("times new roman",15,"bold"),bg="blue",fg="white", activebackground="#607d8b",cursor="hand2",activeforeground="white")

self.btn\_clear.place(x=1000,y=625,height=40,width=160)

self.btn\_start=Button(self.root,state=DISABLED,command=self.start\_function,text="START",bd=5,relief=RAISED,font=("times new roman",15,"bold"), bg="#D56135",fg="white",activebackground="#D56135",cursor="hand2",activeforeground="black")

self.btn\_start.place(x=1200,y=625,height=40,width=160)

def Total\_count(self):

images=0

audios=0

videos=0

documents=0

others=0

self.count=0

combine\_list=[]

for i in self.all\_files:

if os.path.isfile(os.path.join(self.directory,i))==True:

self.count+=1

ext="."+i.split(".")[-1]

for folder\_name in self.folders.items():

for x in folder\_name[1]:

combine\_list.append(x)

if ext.lower() in folder\_name[1] and folder\_name[0]=="images":

images=images+1

if ext.lower() in folder\_name[1] and folder\_name[0]=="audios":

audios=audios+1

if ext.lower() in folder\_name[1] and folder\_name[0]=="videos":

videos=videos+1

if ext.lower() in folder\_name[1] and folder\_name[0]=="documents":

documents=documents+1

for i in self.all\_files:

if os.path.isfile(os.path.join(self.directory,i))==True:

ext="."+i.split(".")[-1]

if ext.lower() not in combine\_list:

others=others+1

self.lbl\_total\_image.config(text="Total Images\n"+str(images))

self.lbl\_total\_audio.config(text="Total Audios\n"+str(audios))

self.lbl\_total\_video.config(text="Total Videos\n"+str(videos))

self.lbl\_total\_document.config(text="Total Documents\n"+str(documents))

self.lbl\_total\_other.config(text="Other\n"+str(others))

self.lbl\_total\_files.config(text="Total Files:"+str(self.count))

def browse\_function(self):

op=filedialog.askdirectory(title="SELECT FOLDER FOR SORTING")

if op!=None:

self.var\_foldername.set(str(op))

self.directory = self.var\_foldername.get()

self.other\_name="others"

self.rename\_folder()

self.all\_files=os.listdir(self.directory)

length=len(self.all\_files)

count=1

self.Total\_count()

self.btn\_start.config(state=NORMAL)

def start\_function(self):

if self.var\_foldername.get()!="":

self.btn\_clear.config(state=DISABLED)

c=0

for i in self.all\_files:

if os.path.isfile(os.path.join(self.directory,i))==True:

c=c+1

self.create\_move(i.split(".")[-1],i)

self.lbl\_st\_total.config(text="Total:"+str(self.count))

self.lbl\_st\_moved.config(text="MOVED:"+str(c))

self.lbl\_st\_left.config(text="LEFT:"+str(self.count-c))

self.lbl\_st\_total.update()

self.lbl\_st\_moved.update()

self.lbl\_st\_left.update()

messagebox.showinfo("Success","All Files has moved Successfully!")

self.btn\_start.config(state=DISABLED)

self.btn\_start.config(state=NORMAL)

else:

messagebox.showinfo("Error","PLease select folder!")

def clear(self):

self.btn\_start.config(state=DISABLED)

self.var\_foldername.set("")

self.lbl\_st\_total.config(text="")

self.lbl\_st\_moved.config(text="")

self.lbl\_st\_left.config(text="")

self.lbl\_total\_image.config(text="")

self.lbl\_total\_audio.config(text="")

self.lbl\_total\_video.config(text="")

self.lbl\_total\_document.config(text="")

self.lbl\_total\_other.config(text="")

self.lbl\_total\_files.config(text="Total Files:")

def rename\_folder(self):

for folder in os.listdir(self.directory):

if os.path.isdir(os.path.join(self.directory,folder))==True:

os.rename(os.path.join(self.directory,folder),os.path.join(self.directory,folder.lower()))

def create\_move(self,ext,file\_name):

find=False

for folder\_name in self.folders:

if "."+ext in self.folders[folder\_name]:

if folder\_name not in os.listdir(self.directory):

os.mkdir(os.path.join(self.directory,folder\_name))

shutil.move(os.path.join(self.directory,file\_name),os.path.join(self.directory,folder\_name))

find=True

break

if find!=True:

if self.other\_name not in os.listdir(self.directory):

os.mkdir(os.path.join(self.directory,self.other\_name))

shutil.move(os.path.join(self.directory,file\_name),os.path.join(self.directory,self.other\_name))

if \_\_name\_\_ == "\_\_main\_\_":

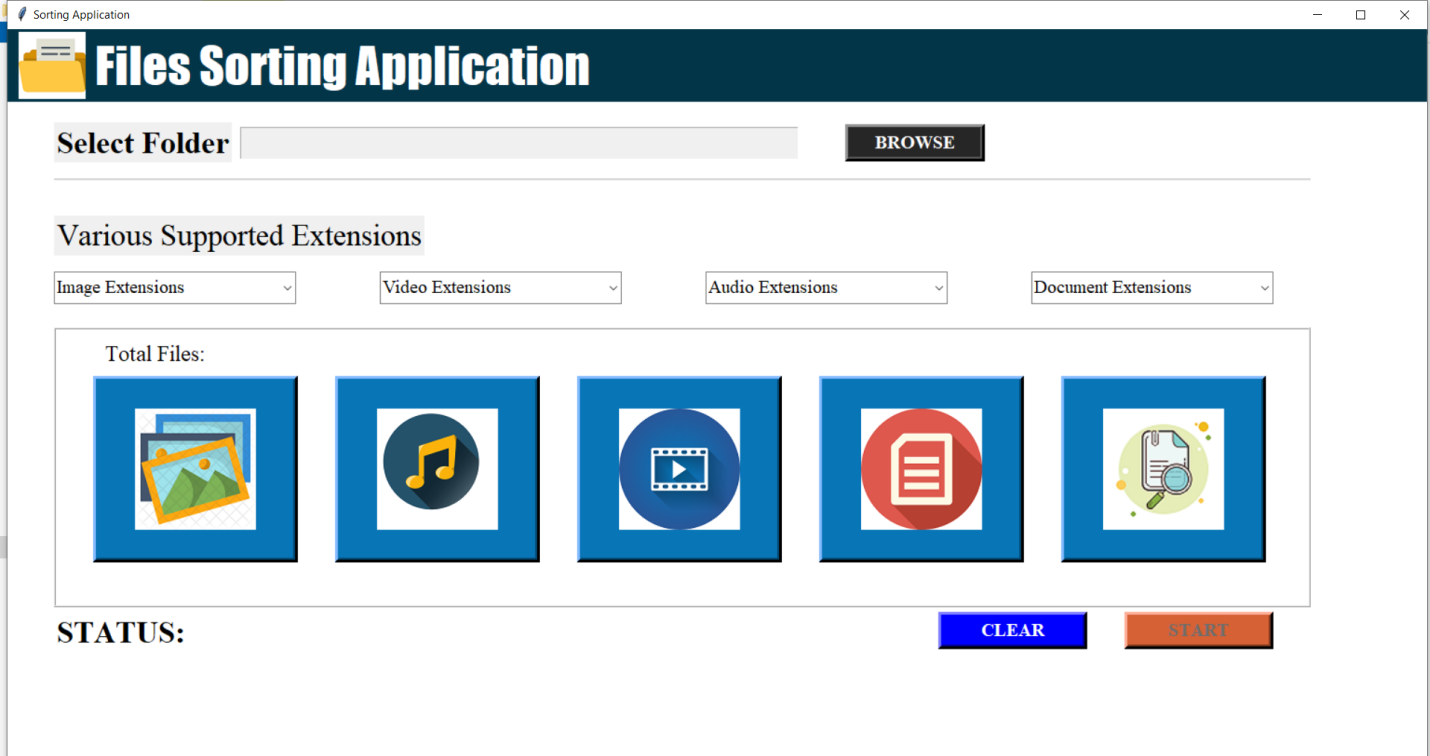
wishMe()

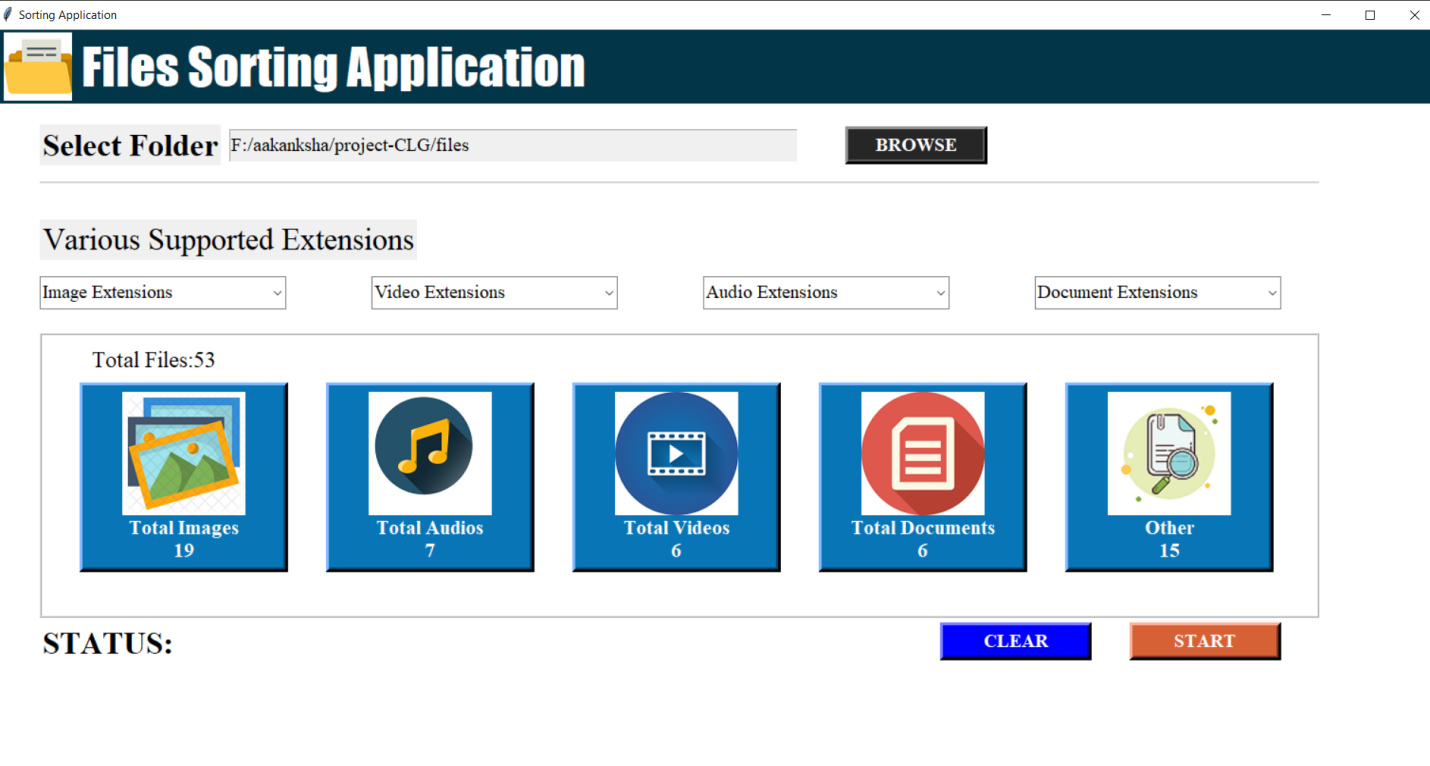
root=Tk()

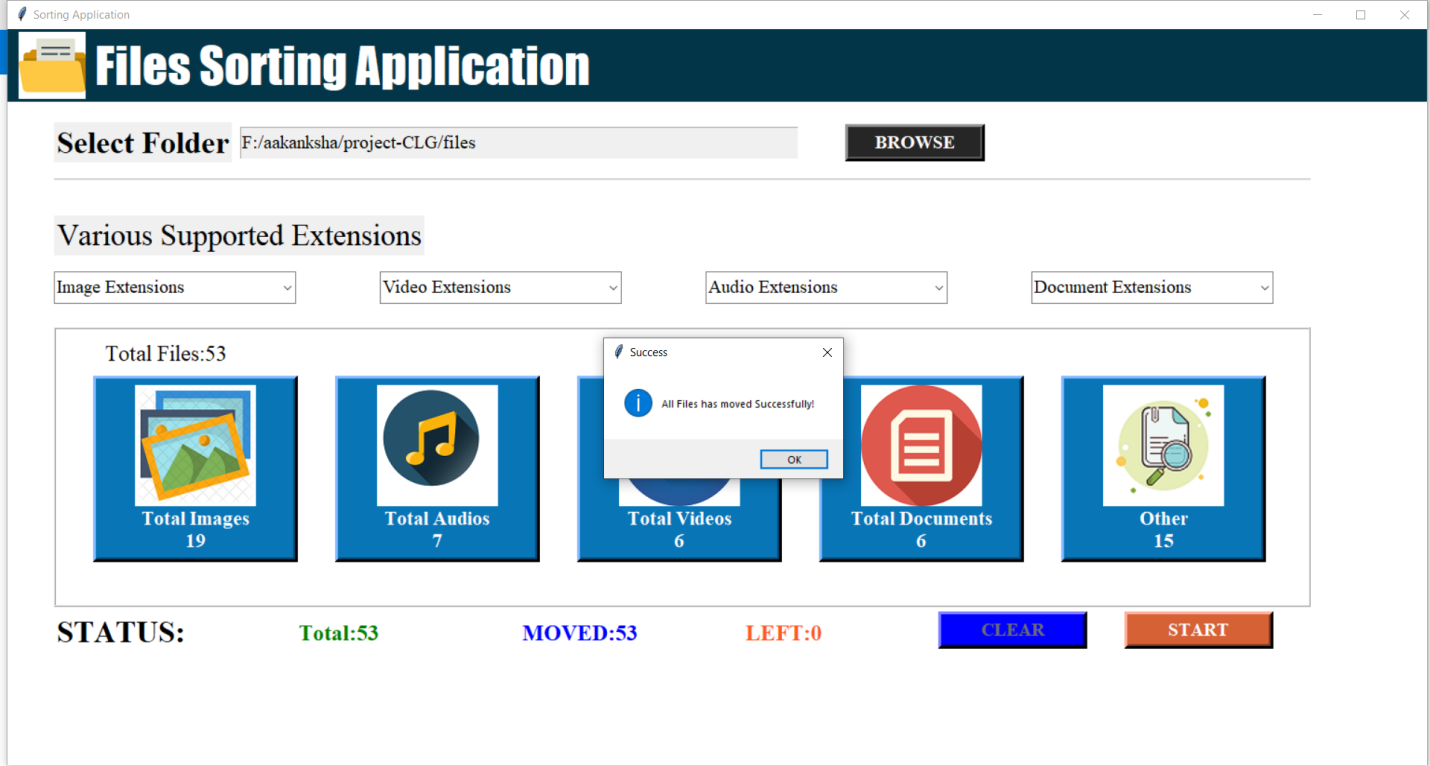
obj = Sorting\_App(root)

root.mainloop()

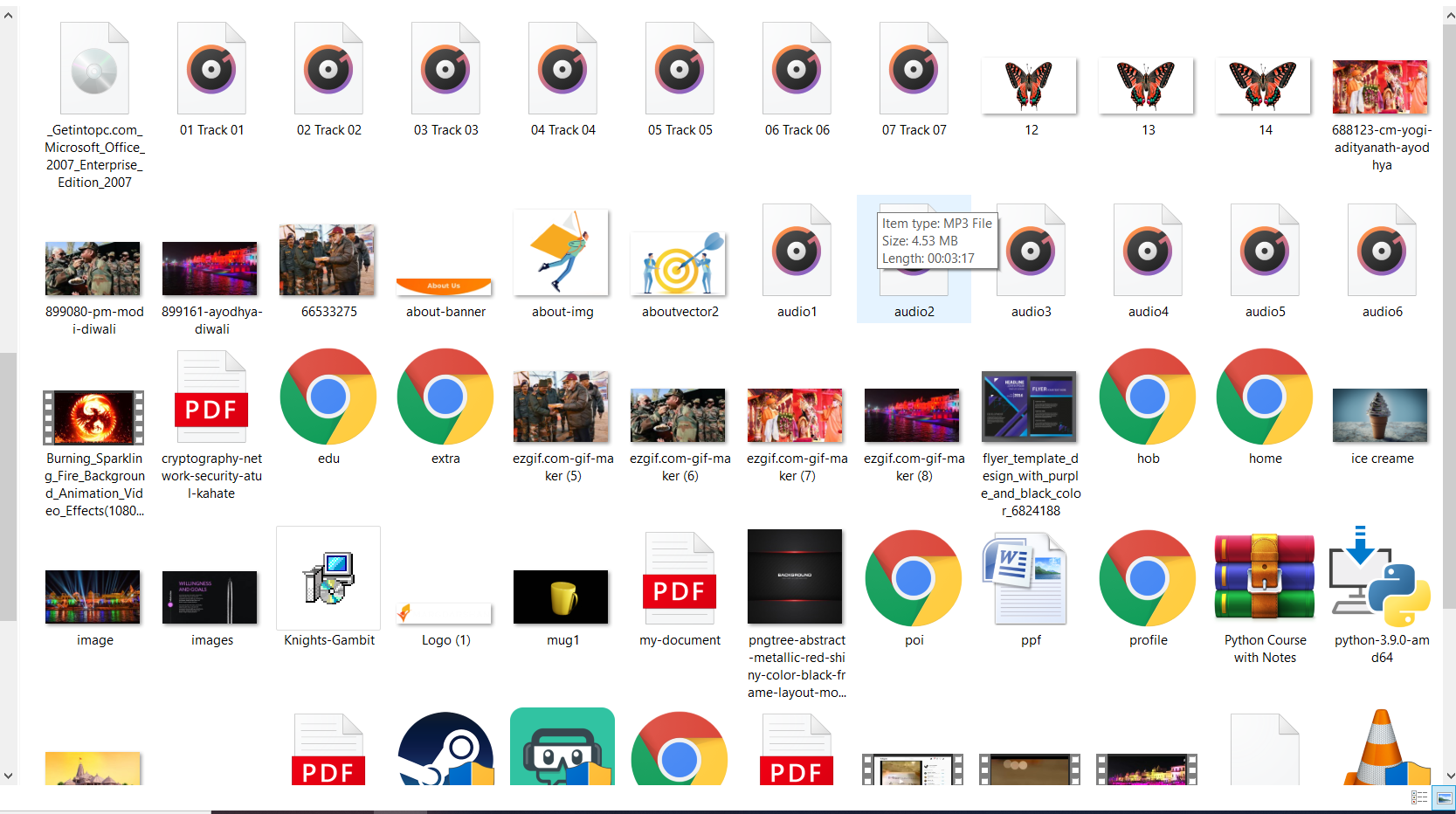
**Screenshots:**



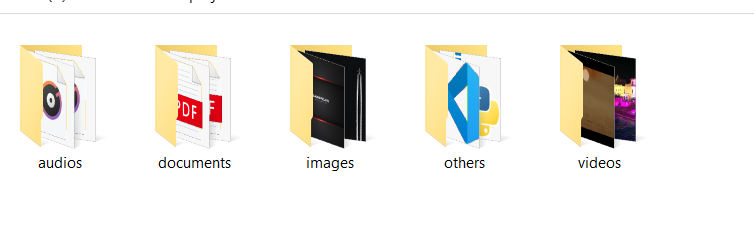




**Before:**



**After:**

****

**Further Improvements:**

I have tried my best to make this project but further some more Improvements can be done in this project to make is more efficient and useful for users. Improvements like:

* 1. We can add a search option for which will search for a required file for user once after the sorting is done.
  2. Also as this project sorts file according to extensions of different files, we can provide user with an option to decide extensions for a particular type of folder, i.e. user can edit the extension option.

**Conclusion:**

This was an excellent project to work on and I have learnt a lot of things to complete this project. A lot of new ideas and several problems that may help happen to a system like this are quite clear for me now. I am not going to claim that this program of 100% functionality, but it has excellent features why work in I have forgot about how we can make this more beneficial for us. I have enjoyed this project and I want to keep working on it to give it a better shape. I believe that I can do make this different in a way that people the love the final output. This project also helped me to learn about different extensions of images audios videos and other documents.

While working with this project I have learned a new programming language that is python. About its features, its module and why people in the world prefer Python over other are programming languages. I had a great experience and have fun while learning python.

**Bibliography:**

* 1. <https://www.python.org/>
  2. <https://docs.python.org/3/tutorial/>
  3. <https://www.w3schools.com/python/python_modules.asp>
  4. <https://code.visualstudio.com/>
  5. <https://pypi.org/project/pyttsx3/>

**Thank You**