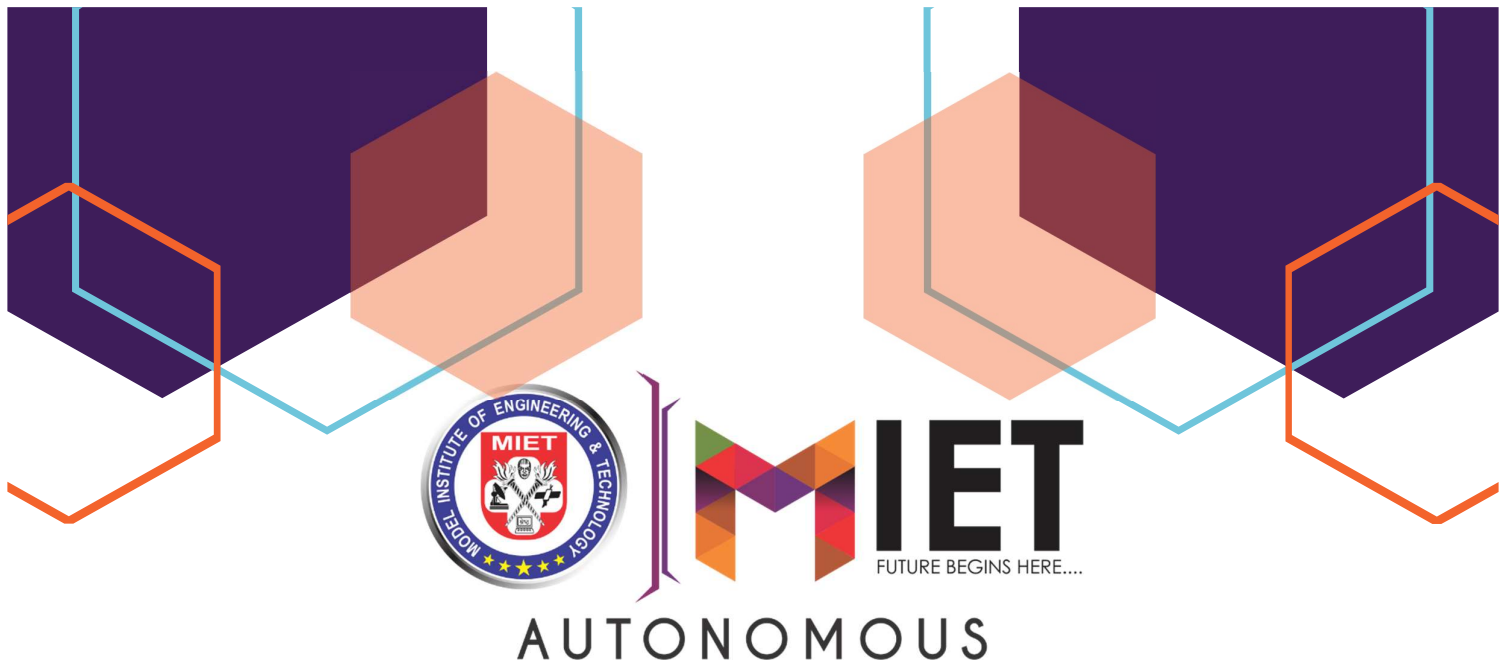


PYTHON LAB PROJECT REPORT
ON
BUILDING A CAMERA APP USING PYTHON

SUBMITTED BY

Khushboo Raina(2020A1r004)
Sparsh Sudan(2020A1R007)
Aryan Verma(2020A1r031)
Dhruv Gupta(2020A1r032)
Muskan Raina(2020A1r033)



SUBMITTED TO
CSE DEPARTMENT
Model Institute of Engineering and Technology (Autonomous)
Jammu, India
2022

CONTENTS

S.NO	TITLE	PAGE NO.
1.	INTRODUCTION TO PROJECT	3
2.	INTRODUCTION TO IDE USED	4
3.	TECH-STACK USED	5
4.	PROJECT IMPLEMENTATION	6
5.	REFERENCES	10

INTRODUCTION TO PROJECT

How can we access camera in PC's or in those laptops whose webcam is damaged?

The answer to this question is this camera app.

Through this camera app a user can access camera in his/her desktop or laptop.

This app operates in 3 modes:

1. Whether user wants to use WEBCAM as a camera.
2. Whether user wants to use his/her PHONE camera.
3. Whether user wants to BROWSE images.

Whenever user will run this app he/she will be given a choice whether to access camera of phone or he/she wants to use this app.

This app also provides user to save the captured images in required destination.

Through this app user can also browse the saved images in his/her desktop.

This app also provides user to capture or browse images in two filter i.e. one in rgb that is colored filter and second one in gray filter.

INTRODUCTION TO IDE USED

What an IDE is?

An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger.

IDE Used: - VS Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

TECH-STACK USED

For building this camera app we have used:

❖ Basic Python

❖ IP Webcam mobile application

- An internet protocol (IP) camera lets you monitor your home or business using software that connects it directly to the internet. Unlike a webcam, it doesn't need a computer to transmit video online.

❖ Python Libraries and Modules

➤ CV2

- Open CV is a cross-platform library using which we can develop real-time computer vision application.

➤ Tkinter

- Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

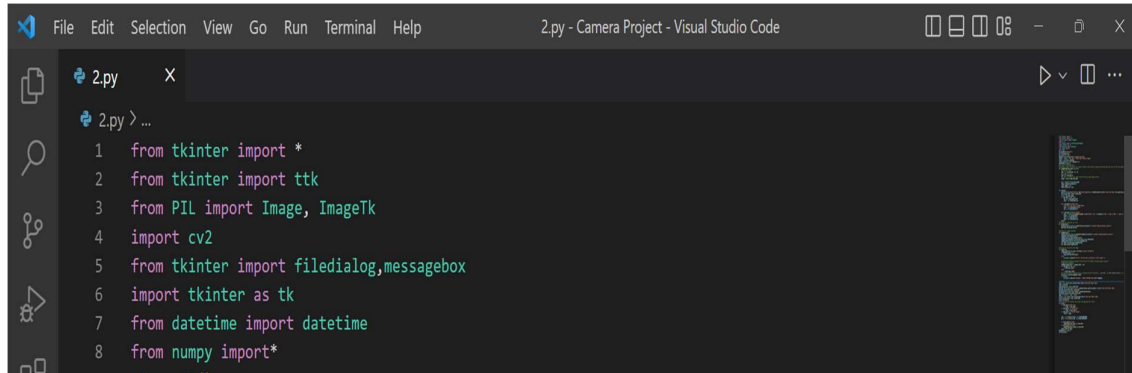
➤ PIL

- Python Imaging Library support opening, manipulating, and saving many different image file formats.

➤ Date Time

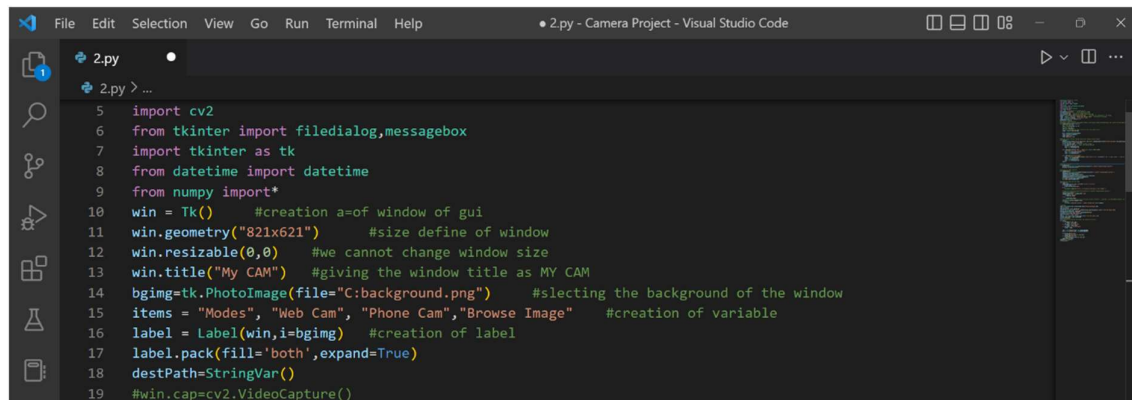
- This library supplies classes for manipulating dates and times.

PROJECT IMPLEMENTATION



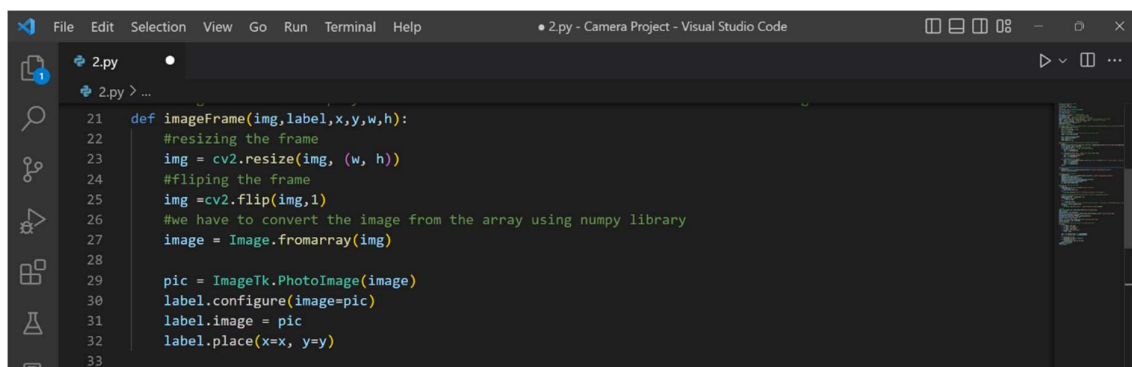
```
1 from tkinter import *
2 from tkinter import ttk
3 from PIL import Image, ImageTk
4 import cv2
5 from tkinter import filedialog,messagebox
6 import tkinter as tk
7 from datetime import datetime
8 from numpy import*
```

In lines 1 - 8 we are just importing the libraries.



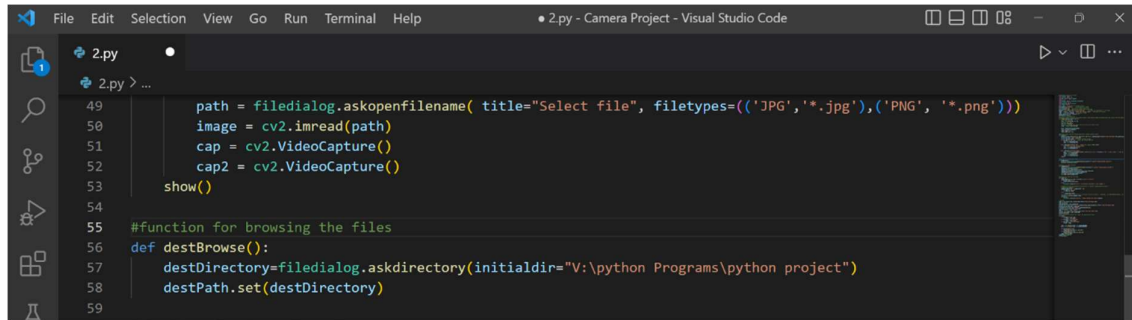
```
5 import cv2
6 from tkinter import filedialog,messagebox
7 import tkinter as tk
8 from datetime import datetime
9 from numpy import*
10 win = Tk() #creation a=of window of gui
11 win.geometry("821x621") #size define of window
12 win.resizable(0,0) #we cannot change window size
13 win.title("My CAM") #giving the window title as MY CAM
14 bgimg=tk.PhotoImage(file="C:background.png") #slecting the background of the window
15 items = "Modes", "Web Cam", "Phone Cam","Browse Image" #creation of variable
16 label = Label(win,i=bgimg) #creation of label
17 label.pack(fill='both',expand=True)
18 destPath=StringVar()
19 #win.cap=cv2.VideoCapture()
```

From lines 10 – 18 we are making a tkinter window and assigning its dimensions and title to it.



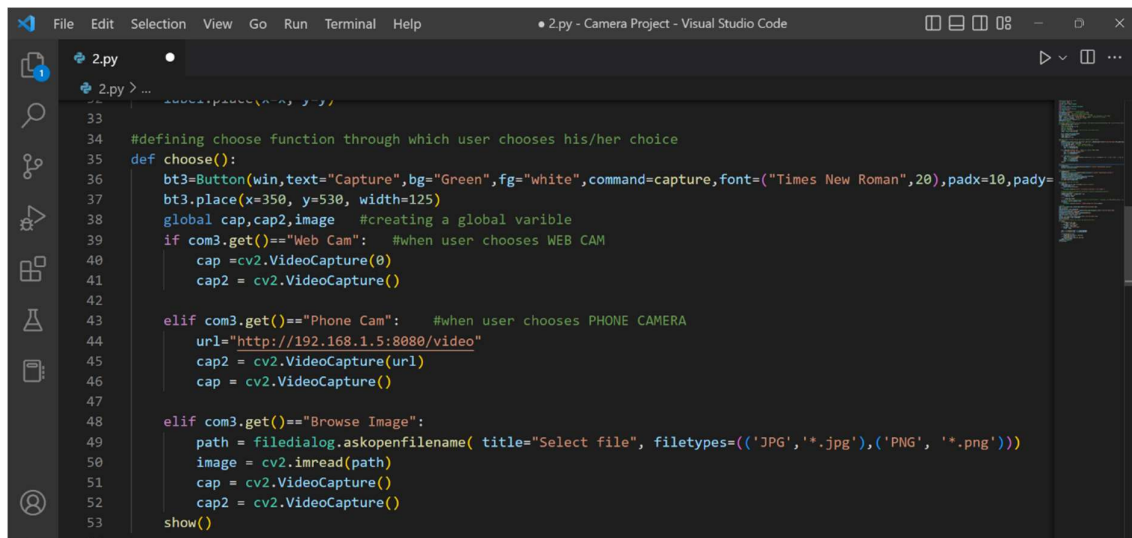
```
21 def imageFrame(img,label,x,y,w,h):
22     #resizing the frame
23     img = cv2.resize(img, (w, h))
24     #flipping the frame
25     img =cv2.flip(img,1)
26     #we have to convert the image from the array using numpy library
27     image = Image.fromarray(img)
28
29     pic = ImageTk.PhotoImage(image)
30     label.configure(image=pic)
31     label.image = pic
32     label.place(x=x, y=y)
33
```

From line 21 – 32 we are defining the frame size of image.



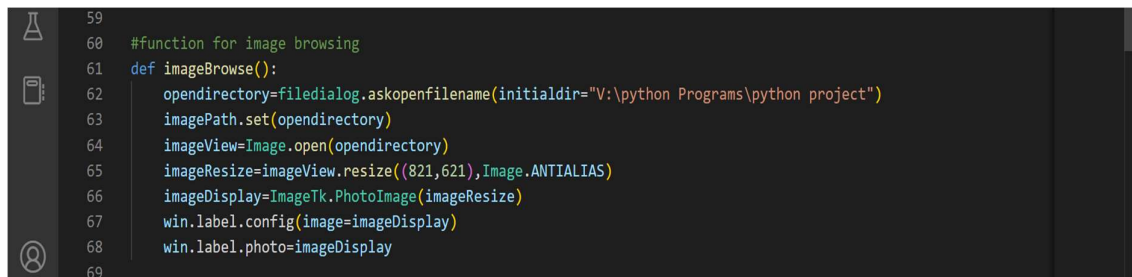
```
49 path = filedialog.askopenfilename( title="Select file", filetypes=(('JPG', '*.jpg'),('PNG', '*.png')))
50 image = cv2.imread(path)
51 cap = cv2.VideoCapture()
52 cap2 = cv2.VideoCapture()
53 show()
54
55 #function for browsing the files
56 def destBrowse():
57     destDirectory=filedialog.askdirectory(initialdir="V:\\python Programs\\python project")
58     destPath.set(destDirectory)
59
```

From lines 56 – 58 we are defining destination browse function through which user can choose the required destination of saving image.



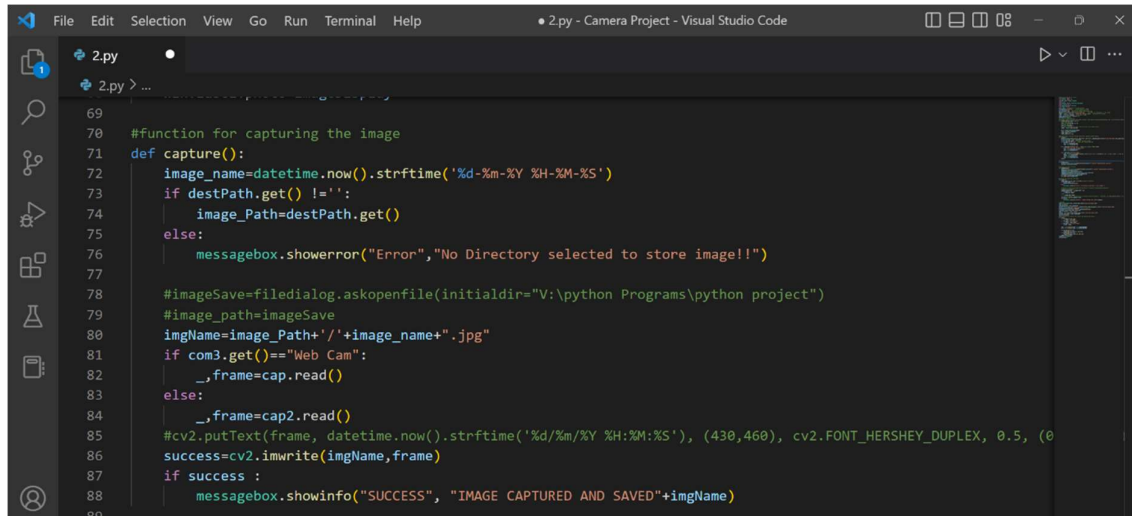
```
33
34 #defining choose function through which user chooses his/her choice
35 def choose():
36     bt3=Button(win,text="Capture",bg="Green",fg="white",command=capture,font=("Times New Roman",20),padx=10,pady=
37     bt3.place(x=350, y=530, width=125)
38     global cap,cap2,image #creating a global variable
39     if com3.get()=="Web Cam": #when user chooses WEB CAM
40         cap =cv2.VideoCapture(0)
41         cap2 = cv2.VideoCapture()
42
43     elif com3.get()=="Phone Cam": #when user chooses PHONE CAMERA
44         url="http://192.168.1.5:8080/video"
45         cap2 = cv2.VideoCapture(url)
46         cap = cv2.VideoCapture()
47
48     elif com3.get()=="Browse Image":
49         path = filedialog.askopenfilename( title="Select file", filetypes=(('JPG', '*.jpg'),('PNG', '*.png')))
50         image = cv2.imread(path)
51         cap = cv2.VideoCapture()
52         cap2 = cv2.VideoCapture()
53     show()
54
```

From lines 35 – 53 we are defining choose function by which user can switch between the options available to him/her.



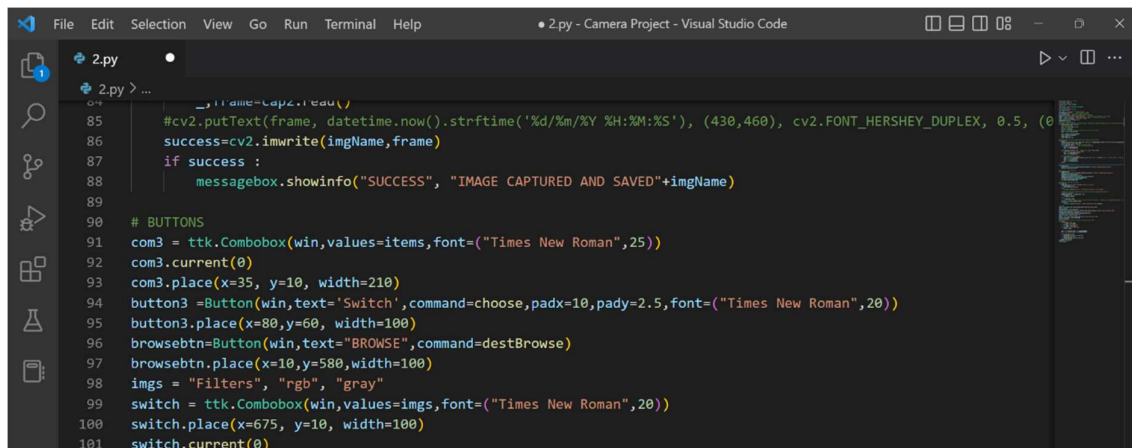
```
59
60 #function for image browsing
61 def imageBrowse():
62     opendirirectory=filedialog.askopenfilename(initialdir="V:\\python Programs\\python project")
63     imagePath.set(opendirdirectory)
64     imageView=Image.open(opendirdirectory)
65     imageResize=imageView.resize((821,621),Image.ANTIALIAS)
66     imageDisplay=ImageTk.PhotoImage(imageResize)
67     win.label.config(image=imageDisplay)
68     win.label.photo=imageDisplay
69
```

From lines 61 – 68 we are defining the image browse function by which user can browse the images in his pc or laptop.

A screenshot of the Visual Studio Code editor window titled "2.py - Camera Project - Visual Studio Code". The editor shows a Python file named "2.py" with the following code:

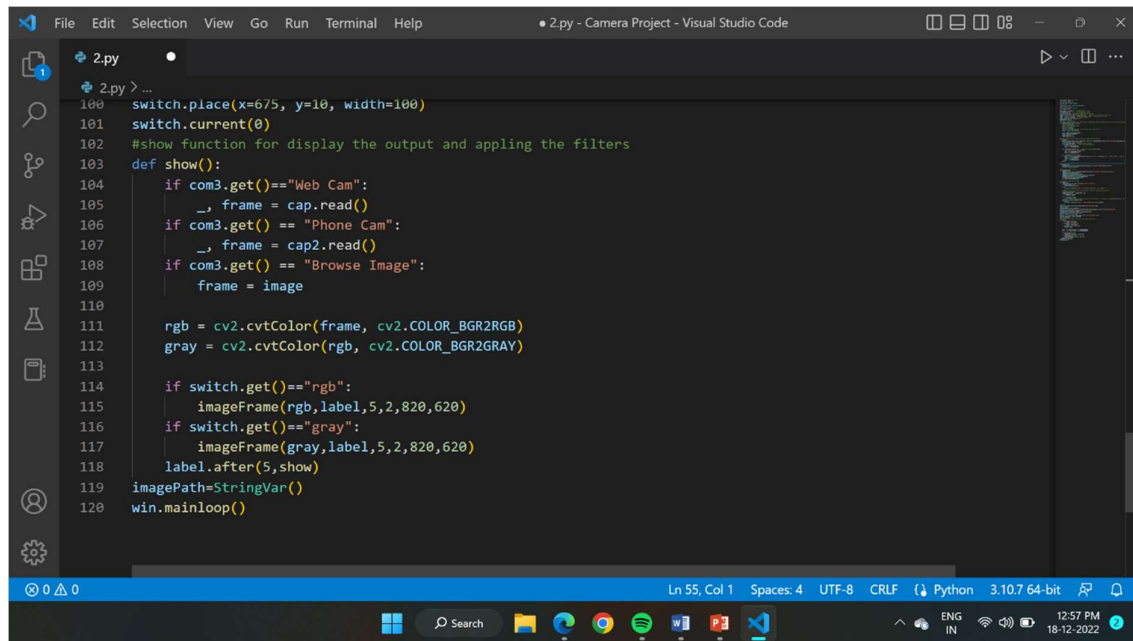
```
69
70 #Function for capturing the image
71 def capture():
72     image_name=datetime.now().strftime('%d-%m-%Y %H-%M-%S')
73     if destPath.get() !='':
74         image_Path=destPath.get()
75     else:
76         messagebox.showerror("Error","No Directory selected to store image!!")
77
78     #imageSave=filedialog.askopenfile(initialdir="V:\python Programs\python project")
79     #image_path=imageSave
80     imgName=image_Path+'/'+image_name+".jpg"
81     if com3.get()=="Web Cam":
82         _,frame=cap.read()
83     else:
84         _,frame=cap2.read()
85     #cv2.putText(frame, datetime.now().strftime('%d/%m/%Y %H:%M:%S'), (430,460), cv2.FONT_HERSHEY_DUPLEX, 0.5, (0
86     success=cv2.imwrite(imgName,frame)
87     if success :
88         messagebox.showinfo("SUCCESS", "IMAGE CAPTURED AND SAVED"+imgName)
89
```

From lines 71 - 88 we are defining the capture function through which user can capture images.

A screenshot of the Visual Studio Code editor window titled "2.py - Camera Project - Visual Studio Code". The editor shows the continuation of the Python file "2.py" with the following code:

```
85     _,frame=cap2.read()
86     #cv2.putText(frame, datetime.now().strftime('%d/%m/%Y %H:%M:%S'), (430,460), cv2.FONT_HERSHEY_DUPLEX, 0.5, (0
87     success=cv2.imwrite(imgName,frame)
88     if success :
89         messagebox.showinfo("SUCCESS", "IMAGE CAPTURED AND SAVED"+imgName)
90
91 # BUTTONS
92 com3 = ttk.Combobox(win,values=items,font=("Times New Roman",25))
93 com3.current(0)
94 com3.place(x=35, y=10, width=210)
95 button3 =Button(win,text='Switch',command=choose,padx=10,pady=2.5,font=("Times New Roman",20))
96 browsebtn=Button(win,text="BROWSE",command=destBrowse)
97 browsebtn.place(x=10,y=580,width=100)
98 imgs = "Filters", "rgb", "gray"
99 switch = ttk.Combobox(win,values=imgs,font=("Times New Roman",20))
100 switch.place(x=10, y=10, width=100)
101 switch.current(0)
```

From lines 91 – 101 we have created button for Tkinter window by which user access different functions.



```
File Edit Selection View Go Run Terminal Help • 2.py - Camera Project - Visual Studio Code
2.py > ...
100 switch.place(x=675, y=10, width=100)
101 switch.current(0)
102 #show function for display the output and applying the filters
103 def show():
104     if com3.get()=="Web Cam":
105         _, frame = cap.read()
106     if com3.get() == "Phone Cam":
107         _, frame = cap2.read()
108     if com3.get() == "Browse Image":
109         frame = image
110
111     rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
112     gray = cv2.cvtColor(rgb, cv2.COLOR_BGR2GRAY)
113
114     if switch.get()=="rgb":
115         imageFrame(rgb,label,5,2,820,620)
116     if switch.get()=="gray":
117         imageFrame(gray,label,5,2,820,620)
118     label.after(5,show)
119 imagePath=StringVar()
120 win.mainloop()
```

From lines 103 – 118 we are defining the show function to display outputs and also applying filters in output.

REFERENCES

1. <https://www.youtube.com/watch?v=BoIJhenSSSU>
2. <https://github.com/pythonpool/small-projects/blob/main/camera.py>
3. <https://thecleverprogrammer.com/2021/01/05/use-phone-camera-with-python/>