Your name

Project name

Course #

Semester

Instructor name

Table of Contents

[Introduction 2](#_Toc102595239)

[Technologies 3](#_Toc102595240)

[Python Language 3](#_Toc102595241)

[Software 3](#_Toc102595242)

[Library 3](#_Toc102595243)

[Tkinter 3](#_Toc102595244)

[Cv2 3](#_Toc102595245)

[Os 4](#_Toc102595246)

[Pillow 4](#_Toc102595247)

[Goals 4](#_Toc102595248)

[Design 4](#_Toc102595249)

[Requirement 4](#_Toc102595250)

[Design Issues 4](#_Toc102595251)

[Implementation 5](#_Toc102595252)

[User Interface Screenshot 5](#_Toc102595253)

[Conclusion 6](#_Toc102595254)

[Appendix B 6](#_Toc102595255)

[**CODE** 8](#_Toc102595256)

[References 12](#_Toc102595257)

# Introduction

group is working on behalf of Forestview to create a “mock” application. This application simulates a Bartender/ Server’s User Interface. In this case, the product is called “BarMaster,”  see below image.   This is part of Forestview’s upcoming SafeZone© proposals to key customers and make app bar master that show a drinks in his bar and when the user say for a drink they give the drimk

## Technologies

### Python Language

Python is a PC programming language frequently used to fabricate sites and programming, robotize errands, and lead information investigation. Python is a universally useful language, meaning it very well may be utilized to make a wide range of projects and isn't particular for a particular issues

### Software

IDE PyCharm Community Edition 2020.2

PyCharm is a devoted Python Integrated Development Environment (IDE) giving a wide scope of fundamental instruments for Python engineers, firmly coordinated to establish an advantageous climate for useful Python, web, and information science advancement

## Library

### Tkinter

Python offers numerous choices for creating GUI (Graphical User Interface). Out of all the GUI techniques, Tkinter is the most regularly utilized strategy. It is a standard Python connection point to the Tk GUI tool compartment delivered with Python. Python with Tkinter is the quickest and least demanding method for making the GUI applications. Making a GUI utilizing Tkinter is a simple undertaking

### Cv2

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

Python is a general purpose programming language started by **Guido van Rossum** that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability.

Compared to languages like C/C++, Python is slower. That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

OpenCV-Python makes use of **Numpy**, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

### Os

The OS module in Python furnishes capacities for cooperating with the working framework. Operating system goes under Python's standard utility modules. This module gives a convenient approach to utilizing working framework subordinate usefulness. The \*os\* and \*os.path\* modules incorporate many capacities to interface with the record framework.

Taking care of the Current Working Directory

Think about Current Working Directory (CWD) as an organizer, where the Python is working. Whenever the documents are called out to simply by them, Python accepts that it begins in the CWD which implies that name-possibly reference will find lasting success provided that the record is in the Python's CWD.

Note: The organizer where the Python script is running is known as the Current Directory. Here the Python script is found

### Pillow

Python Imaging Library (development of PIL) is the accepted picture handling bundle for Python language. It consolidates lightweight picture handling devices that guides in altering, making and saving pictures. Support for Python Imaging Library got ended in 2011, however a task named cushion forked the first PIL project and added Python3.x backing to it. The cushion was declared as a swap for PIL for future use. Cushion upholds countless picture record designs including BMP, PNG, JPEG, and TIFF. The library energizes adding support for more current organizations in the library by making new document decoders.

This module isn't preloaded with Python

## Goals

* Making a Bar master application which is do like bar tender

# Design

## Requirement

The main objective is to collect data through questioning

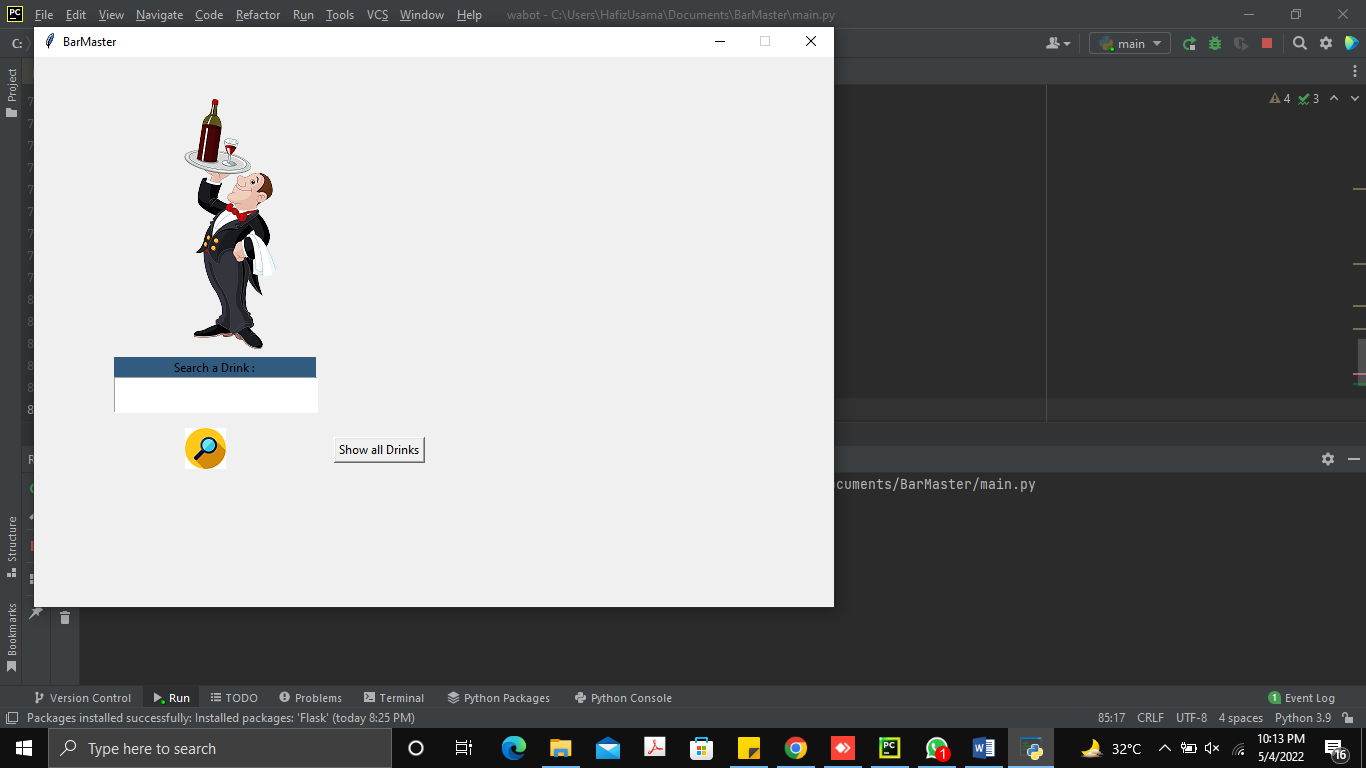
* What assistance is needed?
* What will be the outcome of this project?

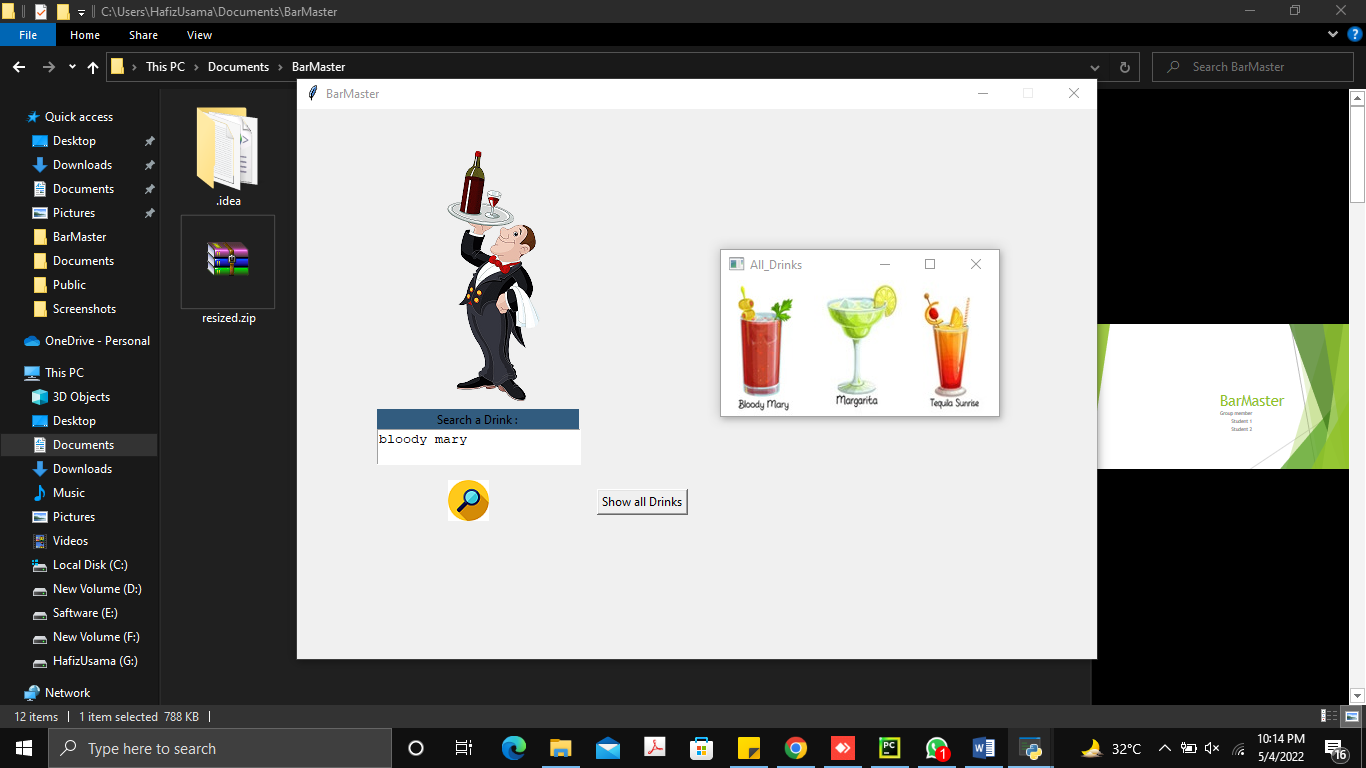
## Design Issues

The problem with this design is the user interface, which is the most difficult to solve. With the help of our method, we were able to make the easily find a drink . Users must be able to understand the design. The main objective is to make GUI a user-friendly application. And to keep an eye on the interactive user interface and other required elements.

# Implementation

## User Interface Screenshot



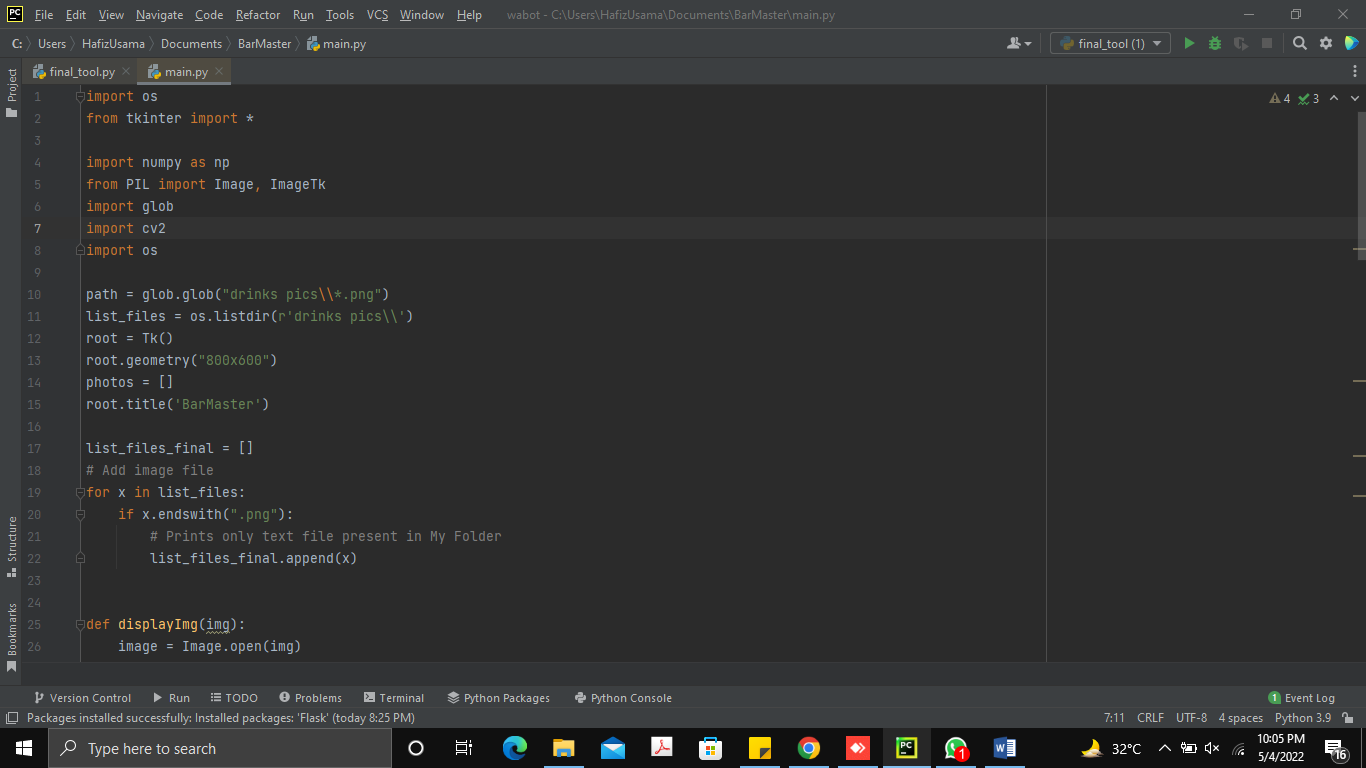


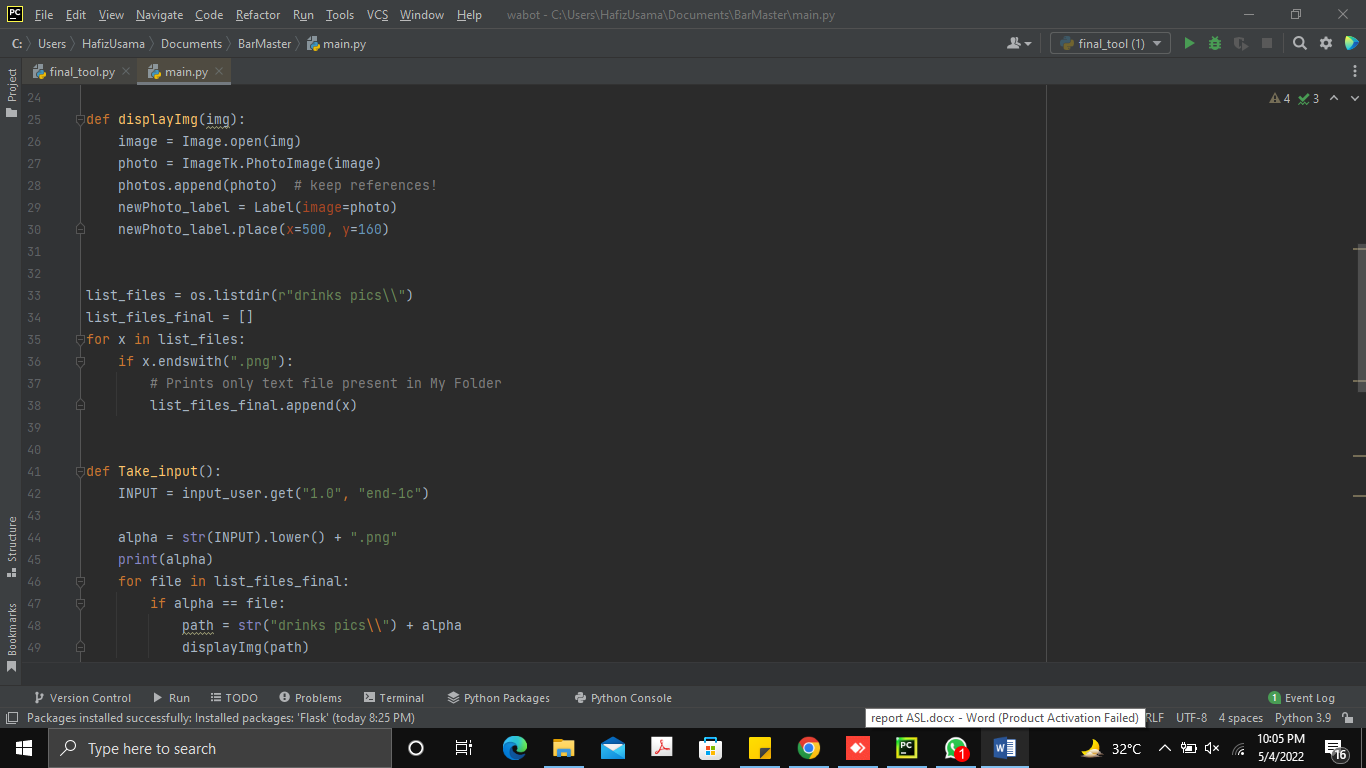
## 

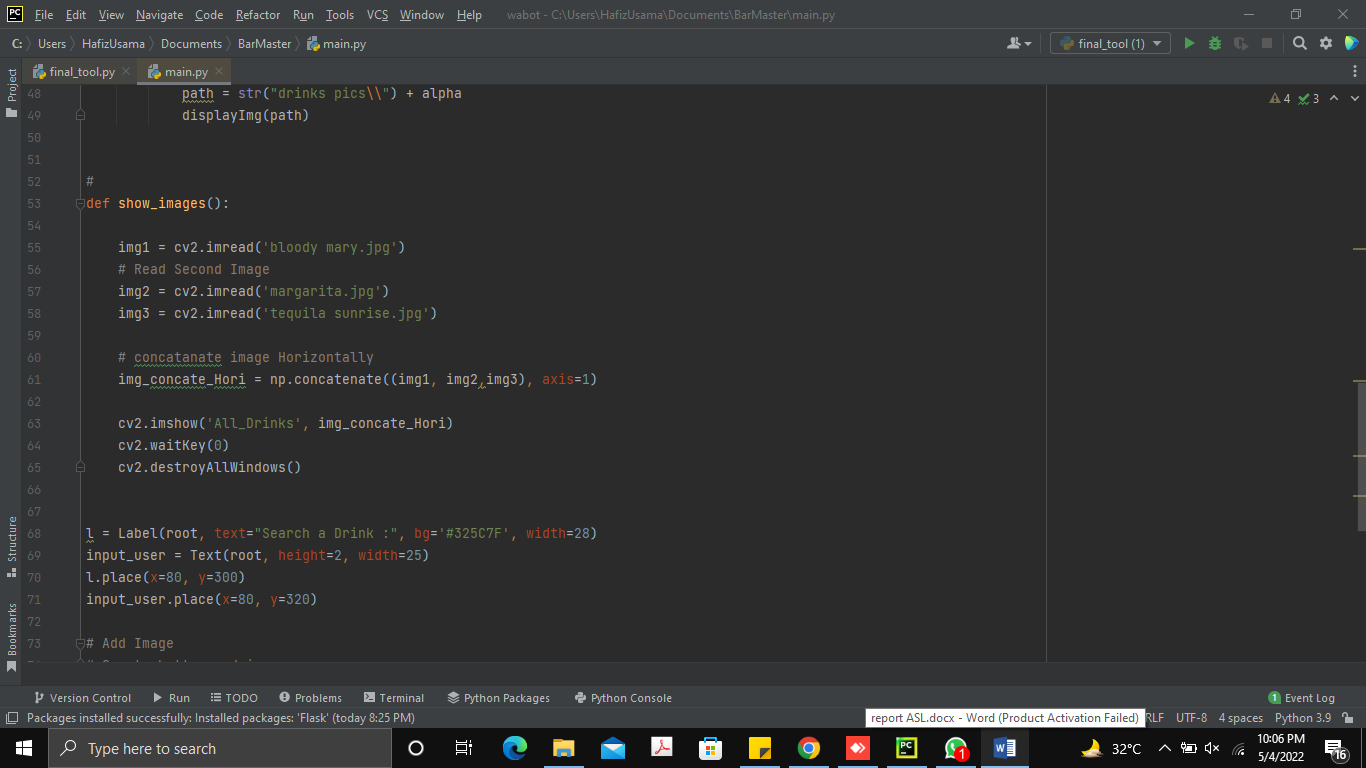
# Conclusion

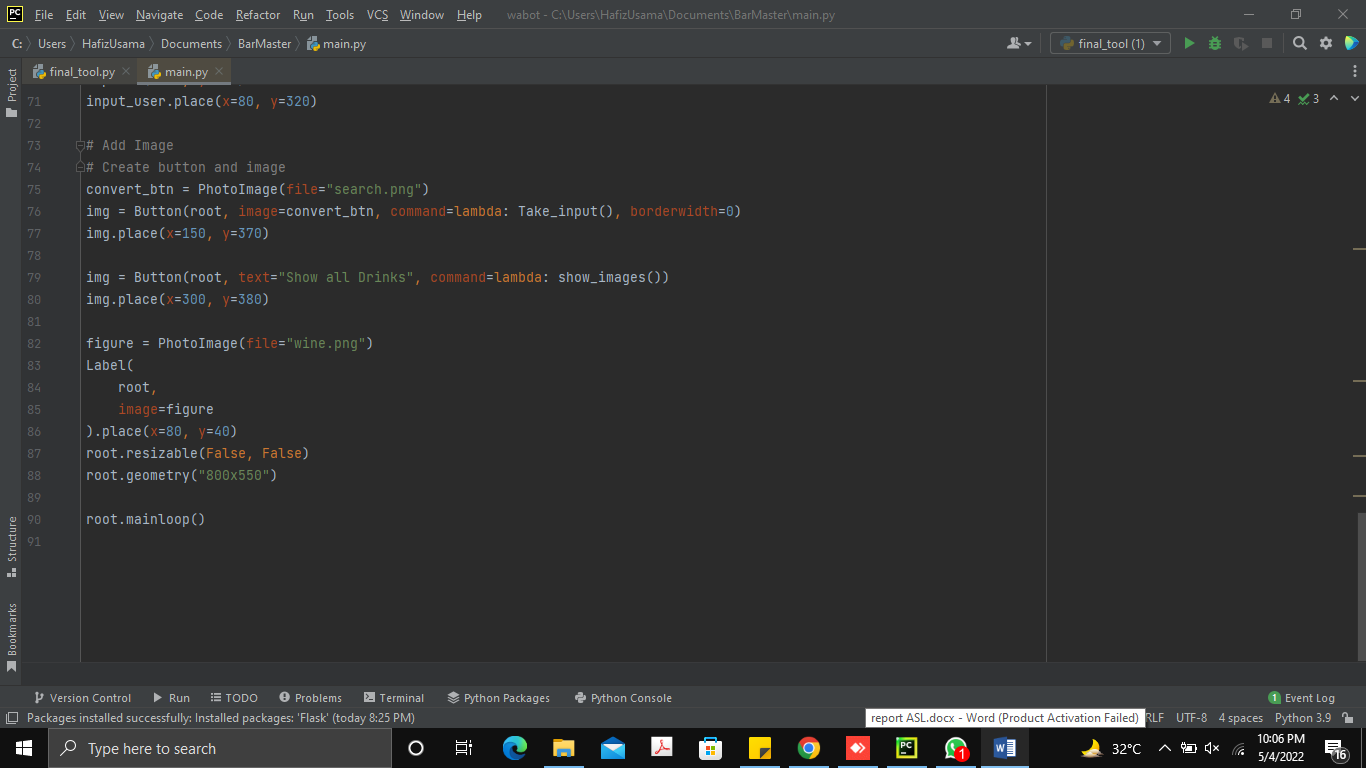
We have developed the system and it will help to minimize the work load this is bra master app make in python GUI based application

# Appendix B









### **CODE**

import os

from tkinter import \*

import numpy as np

from PIL import Image, ImageTk

import glob

import cv2

import os

path = glob.glob("drinks pics\\\*.png")

list\_files = os.listdir(r'drinks pics\\')

root = Tk()

root.geometry("800x600")

photos = []

root.title('BarMaster')

list\_files\_final = []

# Add image file

for x in list\_files:

if x.endswith(".png"):

# Prints only text file present in My Folder

list\_files\_final.append(x)

def displayImg(img):

image = Image.open(img)

photo = ImageTk.PhotoImage(image)

photos.append(photo) # keep references!

newPhoto\_label = Label(image=photo)

newPhoto\_label.place(x=500, y=160)

list\_files = os.listdir(r"drinks pics\\")

list\_files\_final = []

for x in list\_files:

if x.endswith(".png"):

# Prints only text file present in My Folder

list\_files\_final.append(x)

def Take\_input():

INPUT = input\_user.get("1.0", "end-1c")

alpha = str(INPUT).lower() + ".png"

print(alpha)

for file in list\_files\_final:

if alpha == file:

path = str("drinks pics\\") + alpha

displayImg(path)

#

def show\_images():

img1 = cv2.imread('bloody mary.jpg')

# Read Second Image

img2 = cv2.imread('margarita.jpg')

img3 = cv2.imread('tequila sunrise.jpg')

# concatanate image Horizontally

img\_concate\_Hori = np.concatenate((img1, img2,img3), axis=1)

cv2.imshow('All\_Drinks', img\_concate\_Hori)

cv2.waitKey(0)

cv2.destroyAllWindows()

l = Label(root, text="Search a Drink :", bg='#325C7F', width=28)

input\_user = Text(root, height=2, width=25)

l.place(x=80, y=300)

input\_user.place(x=80, y=320)

# Add Image

# Create button and image

convert\_btn = PhotoImage(file="search.png")

img = Button(root, image=convert\_btn, command=lambda: Take\_input(), borderwidth=0)

img.place(x=150, y=370)

img = Button(root, text="Show all Drinks", command=lambda: show\_images())

img.place(x=300, y=380)

figure = PhotoImage(file="wine.png")

Label(

root,

image=figure

).place(x=80, y=40)

root.resizable(False, False)

root.geometry("800x550")

root.mainloop()

# References

Dr Diane Bell. (2021). An interactive visual database for American Sign Language. *An interactive visual database for American Sign Language*, 10.

Woolf, B. P. (2009). Machine Learning. *Machine Learning*, 12.

B. Doner."Hand Shape Identification and tracking for Sign Language Interpretation". Looking at People Workshop, Chambery, France, 1993.

B. Bauer and H. Heinz, “Relevant Features for Video-Based Continuous Sign Language Recognition”, Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 64– 75, 2000.

.9002 ,"قواعدلغةالاشارةالعربيةالقطريةالموحدة S. Samreen and M. Benali,"

F-S. Chen, C-M.Fu and C-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models", Image and Vision Computing, No. 21, pp. 745–758, 2003.

M.F. Tolba, A.S.Elons Scientific Computing Department Faculty of Computer and information Sciences Ain Shams University Cairo, Egypt, "Recent Developments in Sign Language Recognition Systems ",2015..

David S., August, Ludwig H., 2014. Concept in a student competition Future Lions in Cannes Ryan H., Alexandr Opalka created Motionsavvy in 2014 Enable Talk presented In

Microsoft Imagine Cup (2012)