Title/Problem Statement: Face Attendance System

RollNo: MCA033
Name: MUDALIYAR JAYASUDHA TAMILVANAN

RollNo: MCA036
Name: JIGNESH BHAVANBHAI NAKUM

Title/Problem description/explanation:

- Face Attendance System is basically created to make the attendance of people easier by just scanning their face. Only once we have to create an account after that we just need to login using our system by their registered face to make their attendance.

Expected outcomes:

1. For the first time we have to register our face in the system along with the name.
2. It may store the data of the user in our system so after that when user makes his/her attendance then he/she may have to just login in the system with their face.

3. If the face is detected and it is already saved in the system it will display "Welcome username" kind of message. Otherwise it won't allow the user to login in the system.
4. Each detected face is stored in the system along with the date and time in it for attendance purpose.

Tools:

1. User Interface Tools like **Tkinter** for Graphical user Interface.
2. **OpenCV (Open Source Computer Vision Library)** or **CV2** for face detection and face recognition.
3. **Dlib** for image processing and facial recognition.
4. **PIL (Python Imaging Library) ImageTk** supports different image file formats.
5. Hardware Requirement such as **webcams** for capturing images.

- **Code:**

➢ util.py:

```
import tkinter as tk
from tkinter import messagebox


def get_button(window, text, color, command, fg='white'):
        button = tk.Button(window,
                                    text=text,
                                    activebackground='black',
                                    activeforeground='white',
                                    fg=fg,
                                    bg=color,
                                    command=command,
                                    height=2,
                                    width=20,
                                    font=('Helvetica bold',20)
                                   )
        return button

def get_img_label(window):
        label=tk.Label(window)
        label.grid(row=0,column=0)
        return label

def get_text_label(window,text):
        label=tk.Label(window,text=text)
        label.config(font=("sans-serif",21),justify="left")
        return label
```

```
def get_entry_text(window):
        inputtxt=tk.Text(window,
                                height=2,
                                width=15,
                                font=("Arial",32))
        return inputtxt

def msg_box(title,description):
        messagebox.showinfo(title,description)
```

- main.py:

```
import tkinter as tk
import datetime
import subprocess
import os.path
import cv2
from PIL import Image, ImageTk
import util

class App:
   def __init__(self):
      self.main_window=tk.Tk()
      self.main_window.geometry("1200x520+370+120")

      self.login_button_main_window =
            util.get_button(self.main_window, 'login', 'green',
            self.login)
```

```
            self.login_button_main_window.place(x=750,y=300)

            self.register_new_user_button_main_window =
            util.get_button(self.main_window, 'register new user',
            'gray', self.register_new_user, fg='black')
            self.register_new_user_button_main_window.place(x=
            750,y=400)

            self.webcam_label =
             util.get_img_label(self.main_window)
            self.webcam_label.place(x=10, y=0, width=700,
            height=500)

            self.add_webcam(self.webcam_label)
            self.db_dir =
            'E:\MCA_1ST_YEAR\SEM2\PYTHON\Termwork\db'
            if not os.path.exists(self.db_dir):
                    os.mkdir(self.db_dir)

            self.log_path =
               'E:\MCA_1ST_YEAR\SEM2\PYTHON\Termwork\
                Attendance.xls'

    def add_webcam(self,label):
       if 'cap' not in self.__dict__:
          self.cap = cv2.VideoCapture(0)

       self._label = label
       self.process_webcam()
```

```
def process_webcam(self):
        ret, frame = self.cap.read()

        self.most_recent_capture_arr = frame

        img_ = cv2.cvtColor(self.most_recent_capture_arr,
         cv2.COLOR_BGR2RGB)

        self.most_recent_capture_arr_pil = Image.fromarray(img_)

        imgtk = ImageTk.PhotoImage(image
         =self.most_recent_capture_arr_pil)
        self._label.imgtk = imgtk
        self._label.configure(image=imgtk)

        self._label.after(20, self.process_webcam)

    def login(self):
        unknown_img_path = './.tmp.jpg'

        cv2.imwrite(unknown_img_path,
         self.most_recent_capture_arr)

        output= str(subprocess.check_output(['face_recognition',
         self.db_dir, unknown_img_path]))
         name = output.split(',')[1][:-5]

          if name in ['unknown_person','no_persons_found']:
                util.msg_box('Oops...','Unknown User. Please register
                new user or try again.')
```

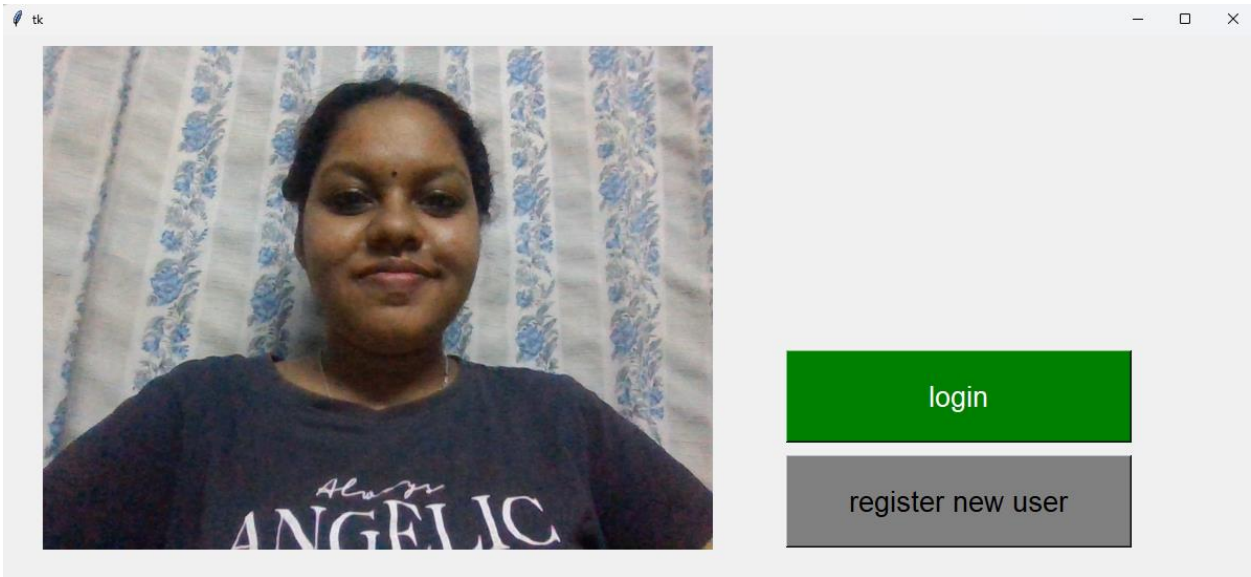
```
        else:
            util.msg_box('Welcome Back!','Welcome,
            {}.'.format(name))
            with open(self.log_path, 'a') as f:
                f.write('{},{}\n'.format(name,datetime.datetime.now()))
                f.close()

        os.remove(unknown_img_path)

    def register_new_user(self):
        self.register_new_user_window =
         tk.Toplevel(self.main_window)
        self.register_new_user_window.geometry("1200x520+370+
         120")

        self.accept_button_register_new_user_window =
         util.get_button(self.register_new_user_window, 'Accept',
         'green', self.accept_register_new_user)
        self.accept_button_register_new_user_window.place(x=750,y
         =300)

        self.try_again_button_register_new_user_window =
         util.get_button(self.register_new_user_window, 'Try Again',
         'red', self.try_again_register_new_user)
        self.try_again_button_register_new_user_window.place(
         x=750,y=400)

        self.capture_label =
         util.get_img_label(self.register_new_user_window)
```

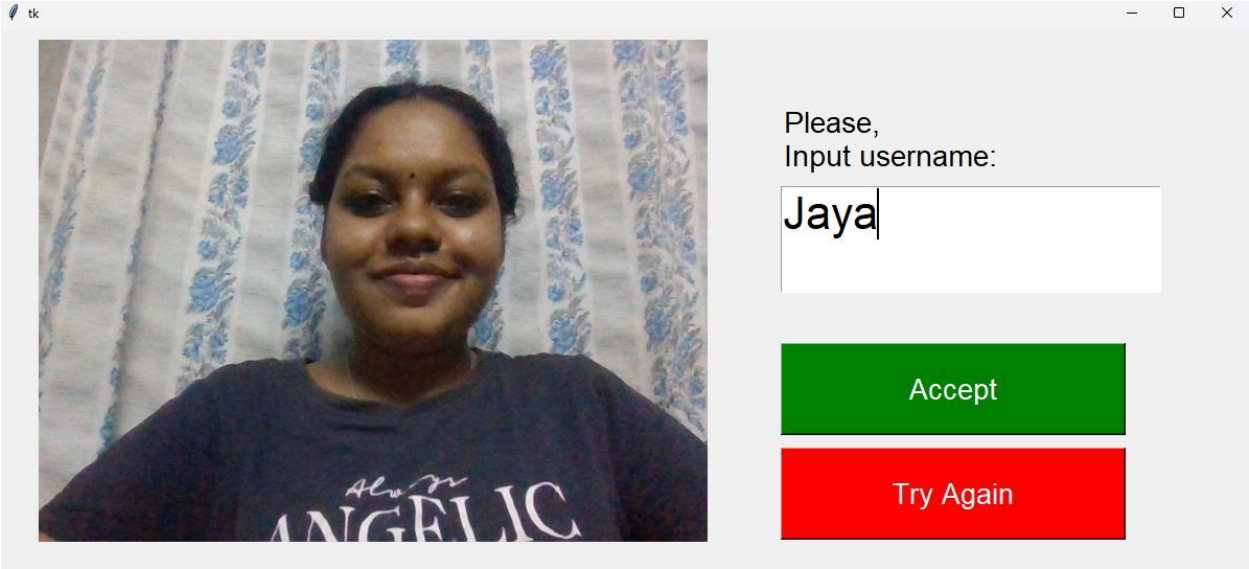
```
    self.capture_label.place(x=10, y=0, width=700, height=500)

    self.add_img_to_label(self.capture_label)

    self.entry_text_register_new_user =
     util.get_entry_text(self.register_new_user_window)
    self.entry_text_register_new_user.place(x=750, y=150)

    self.text_label_register_new_user =
     util.get_text_label(self.register_new_user_window,"Please,\n
     Input username:")
    self.text_label_register_new_user.place(x=750, y=70)

def try_again_register_new_user(self):
    self.register_new_user_window.destroy()

def add_img_to_label(self,label):
    imgtk = ImageTk.PhotoImage(image
     =self.most_recent_capture_arr_pil)
    label.imgtk = imgtk
    label.configure(image=imgtk)

    self.register_new_user_capture =
     self.most_recent_capture_arr.copy()

def start(self):
    self.main_window.mainloop()

def accept_register_new_user(self):
    name = self.entry_text_register_new_user.get(1.0, "end-1c")
```

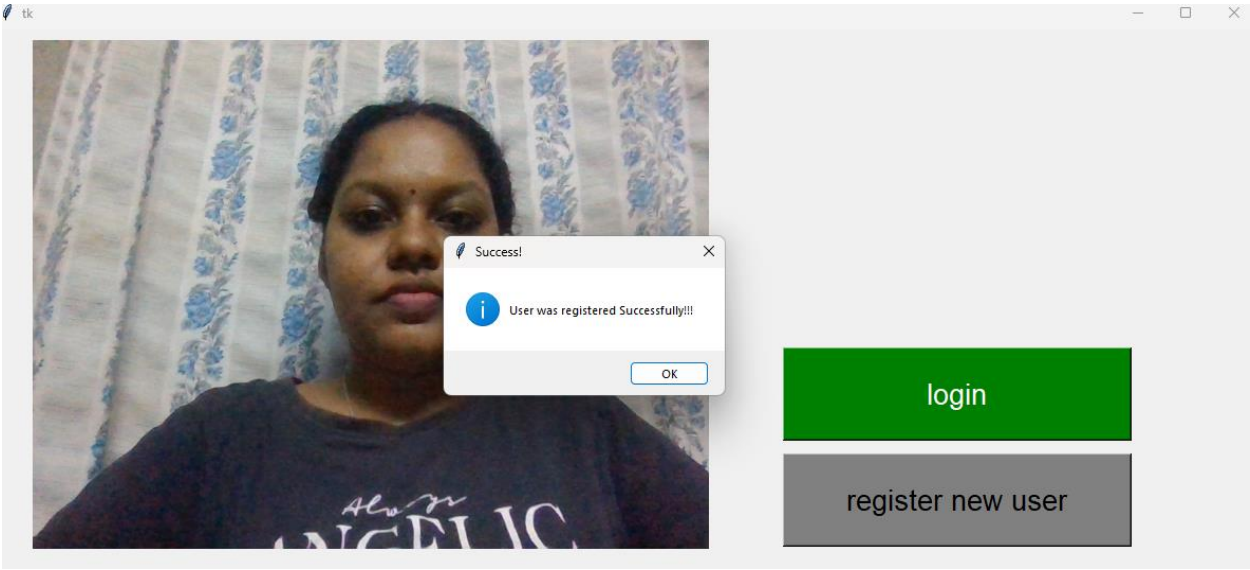
```
        cv2.imwrite(os.path.join(self.db_dir, '{}.jpg'.format(name)),
         self.register_new_user_capture)

        util.msg_box('Success!','User was registered Successfully!!!')

        self.register_new_user_window.destroy()

if __name__ == "__main__":
    app=App()
    app.start()
```

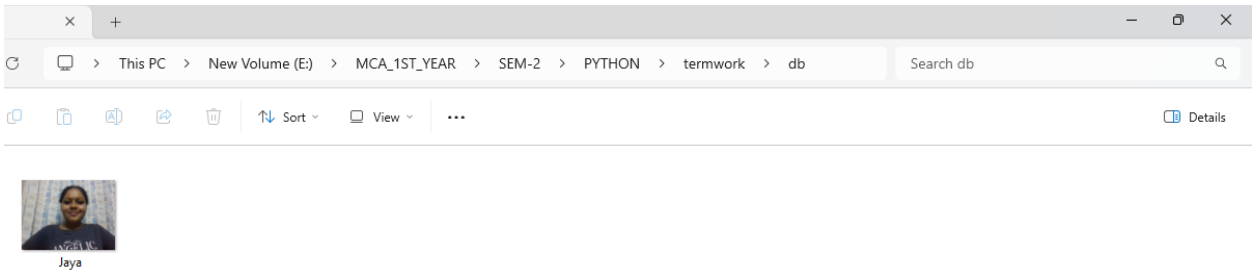
- **Output:**

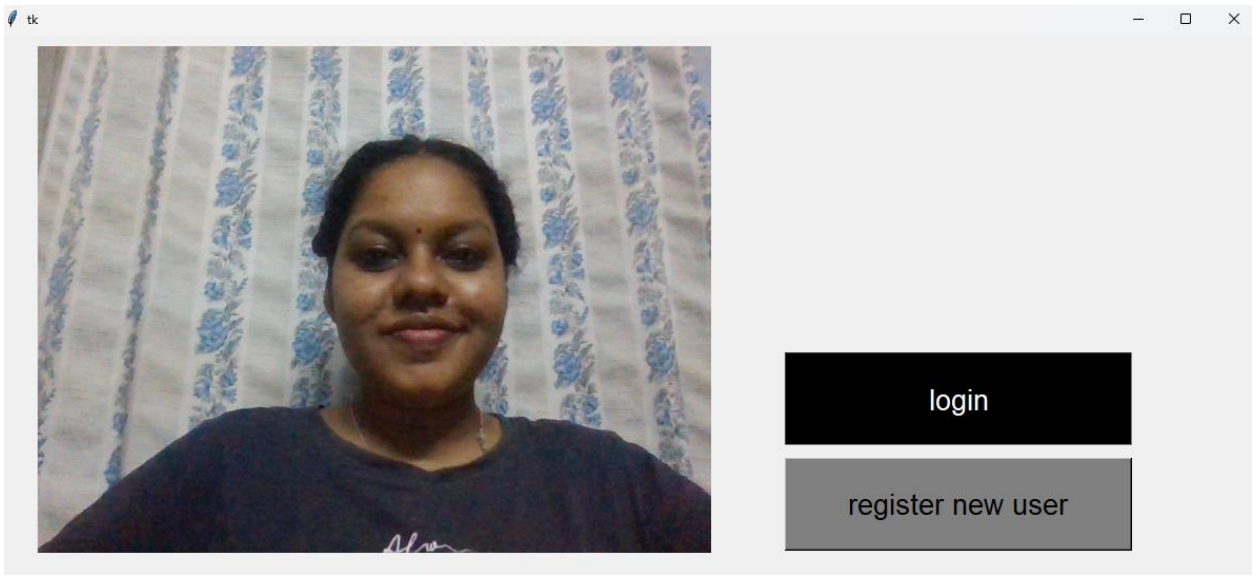
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    Python

PS C:\Users\USER> & C:/Users/USER/anaconda3/python.exe e:/MCA_1ST_YEAR/SEM-2/PYTHON/termwork/main.py
PS C:\Users\USER>
```

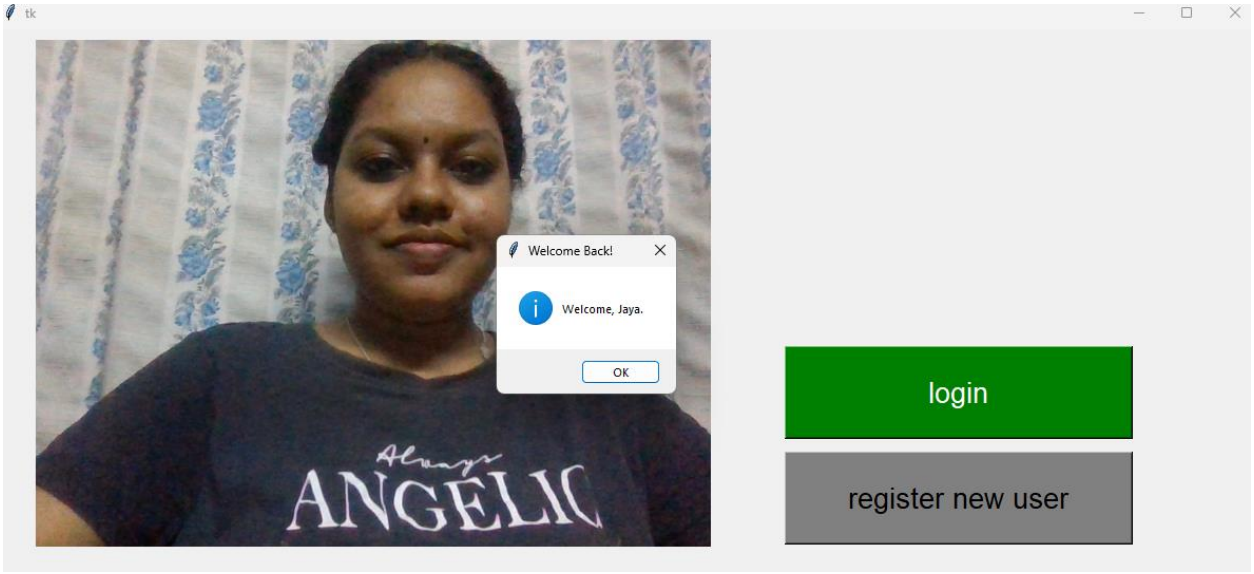


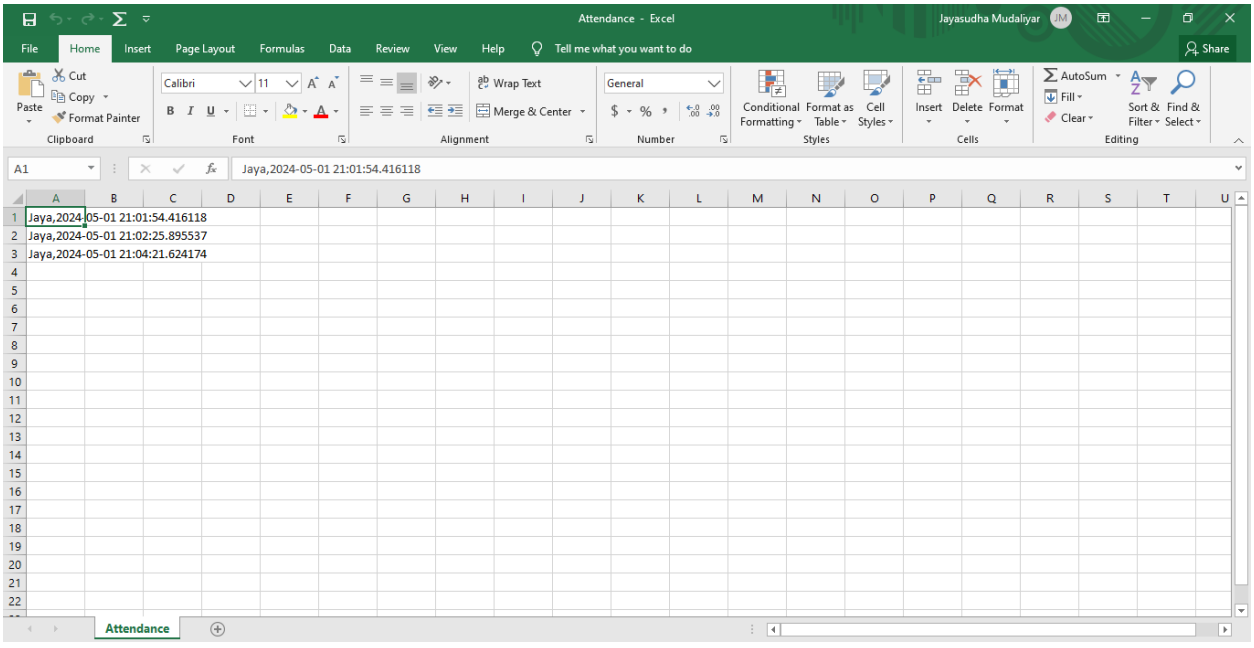

tk
Please,
Input username:
Jaya
Accept
Try Again


tk
Success!
User was registered Successfully!!!
OK
login
register new user


This PC
New Volume (E:)
MCA_1ST_YEAR
SEM-2
PYTHON
termwork
db
Search db
Sort
View
Details
Jaya


tk
login
register new user


tk
Welcome Back!
Welcome, Jaya.
OK
login
register new user


Attendance - Excel
Jayasudha Mudaliyar
A1
Jaya,2024-05-01 21:01:54.416118
Jaya,2024-05-01 21:02:25.895537
Jaya,2024-05-01 21:04:21.624174
Attendance