VPM's R.Z. SHAH COLLEGE OF ARTS, SCIENCE & COMMERCE
(Affiliated to University of Mumbai)
MULUND-MAHARASHTRA-400081

CERTIFICATE

This is to certify that the project entitled, "**USER REGISTERATION FORM "**, submitted by **TALHA USMAN ABDUL HAI KHAN** bearing Seat. No: **2153015** in partial fulfilment of B.Sc. CS (Semester- V) Examination had not been submitted for any other examination and does not form part of any other course undergone by the candidate. It is further certified that he has completed all required phases of the project.

Signature of Internal Guide **Signature of Coordinator**

Date:

Acknowledgement

After months of hard work, I am very happy to present my last year's project. But it wouldn't be right to do so without thanking those who have helped me in completion of my project. So I would like to take full advantage of this opportunity to thank each and every person who helped me throughout the completion of my project that is respected teachers, friends, and family.

I would like to thank my parents & family members who always support me greatly and encouraged me in each and every step. I express my sincere gratitude to towards our internalguide Prof. (**Prof. Yuvraz Wagh**) who gave me unconditional support from the starting point of the project.

I give my special thanks to my Respected **H.O.D Prof.Vilas Mahajan** for encouraging me to complete this project, guiding meand helping me out through all the obstacles in the project.

UNDERTAKING

This is to declare that the project entitled “USER REGISTRATION FORM” is an original work done by the undersigned,

in partial fulfilment of the requirement for the degree “Bachelor of Science in Computer Science” at CS Department, VPM’s R.Z. SHAH COLLEGE OF ARTS, SCIENCE & COMMERCE.

All the analysis design and System developed have been accomplished by the undersigned. Moreover, this project has not been submitted to any other College or University.

Mr. TALHA USMAN ABDUL HAI KHAN

INTRODUCTION

A **registered user** is a user of a website, program, or other system who has previously *registered.* Registered users normally provide some sort of credentials (such as a username or e-mail address, and a password) to the system in order to prove their identity: this is known as logging in. Systems intended for use by the general public often allow any user to register simply by selecting a *register* or *sign up* function and providing these credentials for the first time. Registered users may be granted privileges beyond those granted to unregistered users.

Advantage:-

1. Cheaper:

There is no need to print paper forms and waste money on postage stamps when you use an online form template. Free online forms are readily available and can reduce your costs further by meaning you won't have to allocate time and resources to enter results into a

database – responses are downloaded automatically and the results are accessible at any time for analysis.

2. **Faster:**

No time is wasted waiting for paper questionnaires to be delivered to you, information is gathered automatically. You can quickly create an online form by using a customisable survey template and obtain feedback almost instantly.

3. Easy data handling:

Internet surveys are gaining popularity because of the simplicity of data entry and ease of processing. The data entered by the user can be quickly evaluated and analysed. This significantly reduces the time gap between the collection of data and its evaluation.

4. **Easy follow up:**With online surveys, you can easily follow up with the respondent by just dropping them an email. This way it is convenient both for you and for the customer.

Disadvantage:-

1. Lack of trust with consumer:

People often don’t fully trust the company(website) to use their personal data in a correct manner. They don’t want a company to post

useless (company)information on their social media profile and are worried they will be spammed.

2. Social networks and login are sometimes blocked

3. Security issues.

4. Too many option

REQUIREMENT SPECIFICATION

A Software Requirement specification (SRS) is a description of a software system to be developed. It lays out functional and non functional requirements, and may include a set of use cases that describe the user interactions that the software must provide.

SOFTWARE REQUIREMNTS :-

Front End :- python

Back End :- Mysql workbeach

Server: Mysql server

Development Tool :- Visual studio code

HARDWARE REQUIREMENTS :-

Processor :- Intel i5/AMD

Hard Disk :- 5 GB

RAM :- 8 GB

NON-FUNCTIONAL REQUIREMENT:-

Performance Requirements :

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of the system, which will fit into required environment. It rests largely with the users of the system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stage so that the system can be designed according to those requirements. It is Very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use. The requirement Specification for any system can be broadly stated as given below :

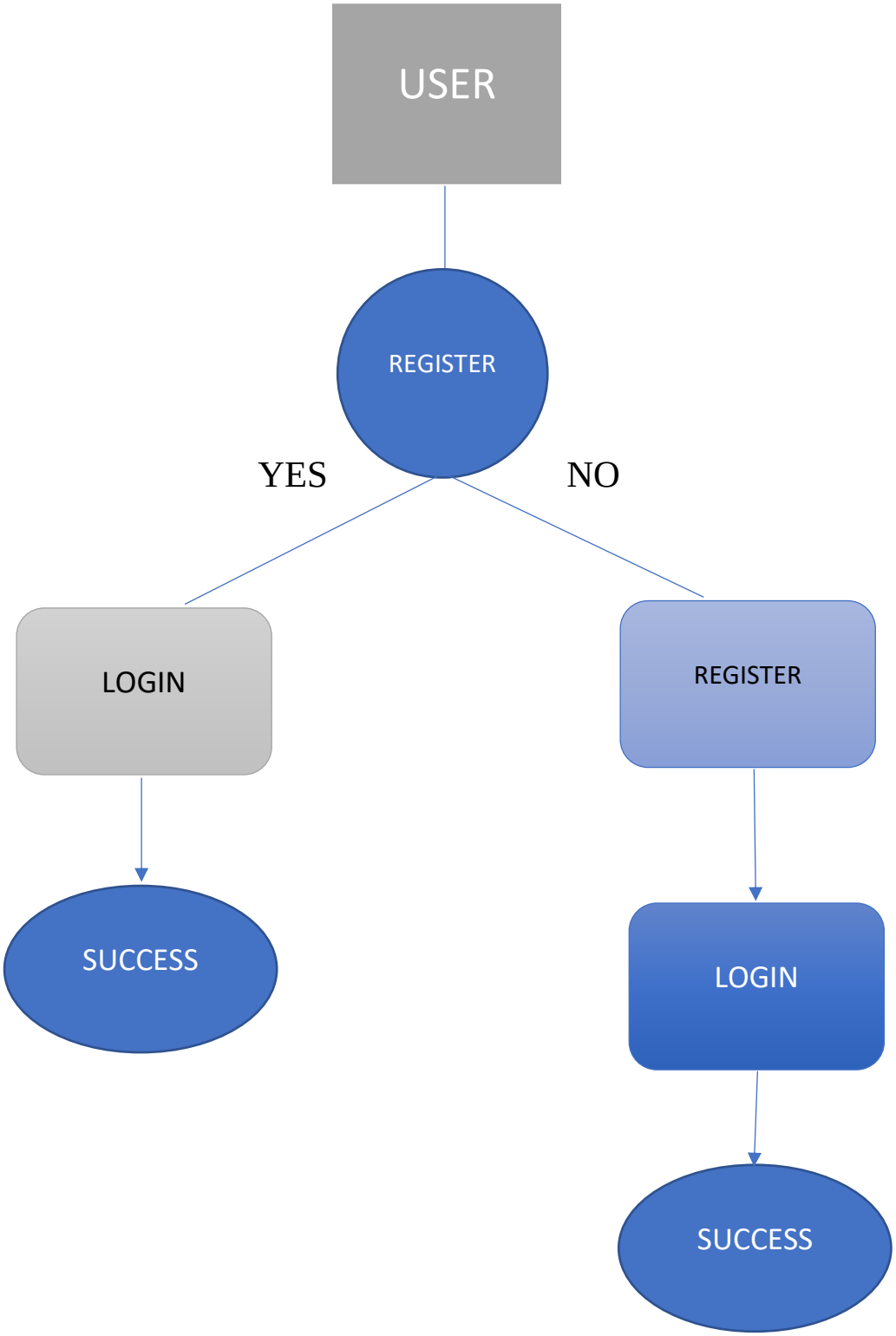
Security

- The Database should be protected from hacking, virus etc.

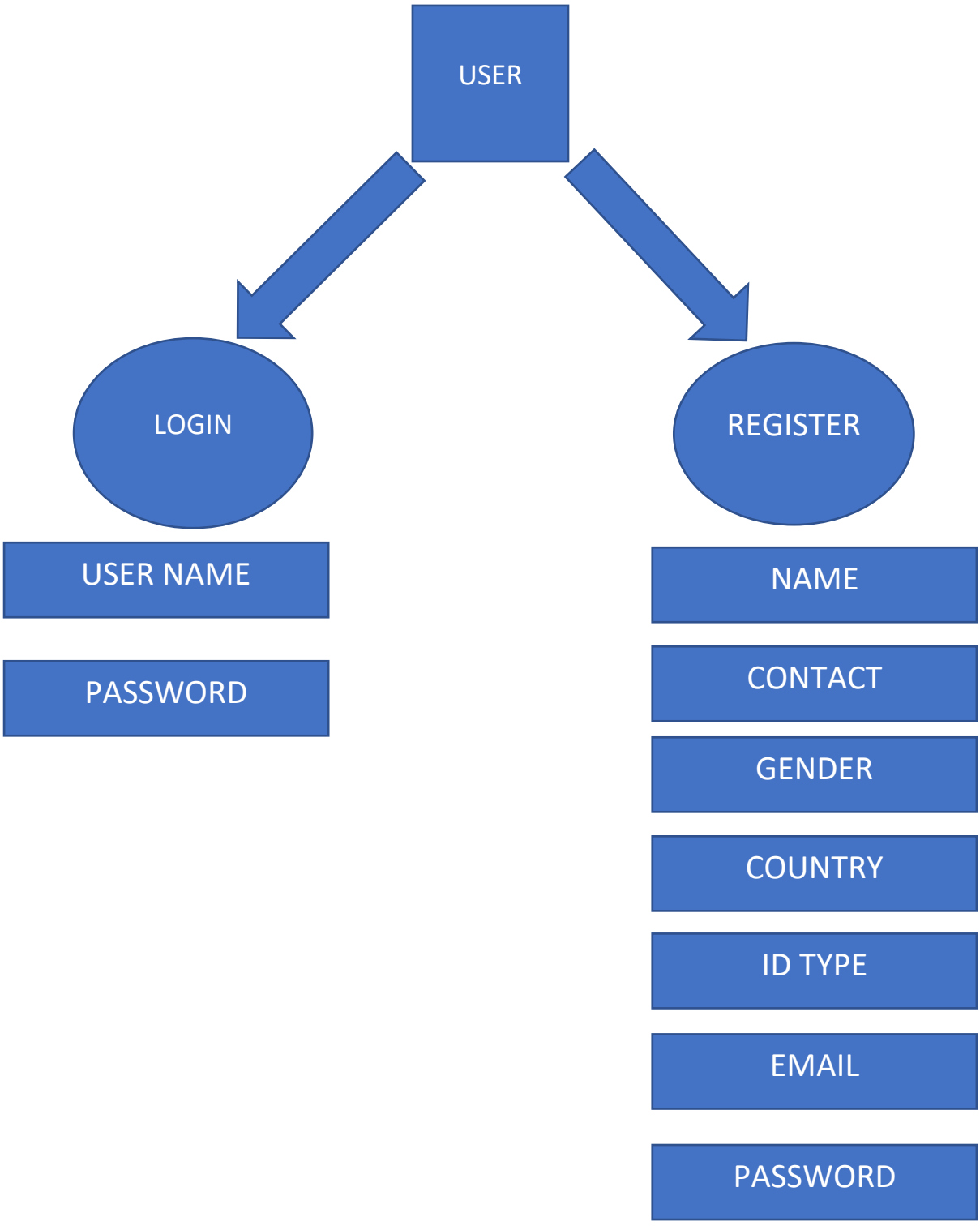
Availability

- This application will be available always

SYSTEM DESIGN



FLOW CHART


USER
LOGIN
REGISTER
USER NAME
PASSWORD
NAME
CONTACT
GENDER
COUNTRY
ID TYPE
EMAIL
PASSWORD

```
from tkinter import *
from tkinter import ttk
from typing import Match
from PIL import Image,ImageTk
from tkinter import messagebox
import re
import pyttsx3 as pt
import speech_recognition as sr
import mysql.connector

class Base:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1610x700+0+0")          #
        #self.root.minsize(1200,700)
        #self.root.maxsize(1000,700)
        self.root.title(" USER REGISTRAION FORM")
        self.Font=font=("monospace 20 ")
        self.root["bg"]= "black"
        self.loginpage()
```

```
def loginpage(self):
        self.v_user=StringVar()
        self.v_pass=StringVar()

        lgo_img=Image.open("img/logimg.png")
        lgo_img=lgo_img.resize((50,50),Image.ANTIALIAS)
        self.image_icon=ImageTk.PhotoImage(lgo_img)
        img=PhotoImage(file="img/loginimg.png")
        root.iconphoto(FALSE,img)

        tlt_frm=Frame(self.root,bd=1,relief=RIDGE)
        tlt_frm.place(x=330,y=180,width=550,height=70)

        tlt_lbl=Label(tlt_frm,image=self.image_icon,compound=LEFT,text="LOGIN FORM",
        font=("monospace 30 bold"),fg="black")
        tlt_lbl.place(x=105,y=5)

        log_frm=Frame(self.root,bd=1,relief=RIDGE)
        log_frm.place(x=330,y=252,width=550,height=295)

        user_lbl=Label(log_frm,text="Username",font=self.Font)
        user_lbl.grid(row=0,column=0,padx=15,pady=35  ,sticky=W)

        self.user=ttk.Entry(log_frm,font=self.Font,textvariable=self.v_user,width=20)
        self.user.grid(row=0,column=1,padx=0,pady=0,sticky=W)
```

```
pass_lbl=Label(log_frm,text="Password: ",font=self.Font)
pass_lbl.grid(row=1,column=0,padx=15,pady=0,sticky=W)

self.password=ttk.Entry(log_frm,font=self.Font,textvariable=self.v_pass  ,width=20,show="*")
self.password.grid(row=1,column=1,padx=0,pady=0,sticky=W)

btn_frm=ttk.Frame(log_frm)
btn_frm.place(x=170,y=175,width=290,height=50)

login_btn=Button(btn_frm,text="login",command=self.login,font=("monospace 15 bold"),width=7
,cursor="hand2",bg="green",fg="black",activebackground="black",activeforeground="white")
login_btn.grid(row=0,column=0,padx=30,pady=5,sticky=W)

reg_btn=Button(btn_frm,text="register",command=self.registerpage,font=("monospace 15 bold"),
width=7,cursor="hand2",bg="green",fg="black",activebackground="black",activeforeground="white")
reg_btn.grid(row=0,column=1,padx=10,pady=5,sticky=W)
```

```
def registerpage(self):
        self.v_name= StringVar()
        self.v_contact = StringVar()
        self.v_gender = StringVar()
        self.v_country = StringVar()
        self.v_id = StringVar()
        self.v_idno = StringVar()
        self.v_email = StringVar()
        self.v_password = StringVar()
        self.v_confirm = StringVar()
        self.v_check = IntVar()

        self.bg=ImageTk.PhotoImage(file="img/BG.png")
        bg_lbl=Label(self.root,image=self.bg,bd=2,relief=RAISED)
        bg_lbl.place(x=0,y=0,relwidth=1,relheight=1)

        img=PhotoImage(file="img/regicon.png")
        root.iconphoto(FALSE,img)

        logo_img=Image.open("img/regimg.png")
        logo_img=logo_img.resize((60,60),Image.ANTIALIAS)
        self.image_icon=ImageTk.PhotoImage(logo_img)

        title_frm=Frame(self.root,bd=1,relief=RIDGE)
        title_frm.place(x=450,y=15,width=550,height=70)
```

```
title_lbl=Label(title_frm,image=self.image_icon,compound=LEFT,text="REGISTRATION FORM",
font=("monospace 30 bold"),fg="black")
title_lbl.place(x=15,y=5)

info_frm=Frame(self.root,bd=1,relief=RIDGE)
info_frm.place(x=450,y=87,width=550,height=595)

name_lbl=Label(info_frm,text="Username: ",font=self.Font)
name_lbl.grid(row=0,column=0,padx=5,pady=5,sticky=W)

name_entry=ttk.Entry(info_frm,textvariable=self.v_name,font=self.Font,width=20)
name_entry.grid(row=0,column=1,padx=0,pady=0,sticky=W)

name_valid=self.root.register(self.validnanme)
name_entry.config(validate="key",validatecommand=(name_valid,"%P"))

cont_lbl=Label(info_frm,text="Contact: ",font=self.Font)
cont_lbl.grid(row=1,column=0,padx=5,pady=3,sticky=W)


cont_entry=ttk.Entry(info_frm,textvariable=self.v_contact,font=self.Font,width=20)
cont_entry.grid(row=1,column=1,padx=0,pady=5,sticky=W)
```

```
contact_valid=self.root.register(self.validcontact)
cont_entry.config(validate="key",validatecommand=(contact_valid,"%P"))
gen_lbl=Label(info_frm,text="Gender: ",font=self.Font)
gen_lbl.grid(row=2,column=0,padx=5,pady=3,sticky=W)

gen_frm=Frame(info_frm)
gen_frm.place(x=160,y=105,width=280,height=35)

rdo_btn=Radiobutton(gen_frm,variable=self.v_gender,value="Male",text="Male",font=("monospace 12 bold"))
rdo_btn.grid(row=0,column=0,padx=10,pady=0,stick=W)
self.v_gender.set("Male")

rdo_btn=Radiobutton(gen_frm,variable=self.v_gender,value="Female",text="Female",font=("monospace 12 bold"))
rdo_btn.grid(row=0,column=1,padx=10,pady=0,stick=W)

country_lbl=Label(info_frm,text="Country: ",font=self.Font)
country_lbl.grid(row=4,column=0,padx=3,pady=10,sticky=W)

countries=["INDIA", "USA" ,"UAE","UK","ENGLAND","PAKISTAN","AFGANISTAN"]
drp_lst=OptionMenu(info_frm,self.v_country, *countries)
drp_lst.config(width=21,font=self.Font,bg="white")
self.v_country.set("Select Your Country")
drp_lst.grid(row=4,column=1,padx=0,pady=10,sticky=W)
```

```
id_lbl=Label(info_frm,text="ID Type: ",font=self.Font)
id_lbl.grid(row=5,column=0,padx=3,pady=5,sticky=W)

self.combo_id=ttk.Combobox(info_frm,textvariable=self.v_id,font=self.Font,justify="center",
state="readonly",width=23)
self.combo_id["values"]=("Select Your ID","Adhar card","Passport","Driving Licence")
self.combo_id.grid(row=5,column=1,padx=0,pady=5)
self.combo_id.current(0)

id_lbl=Label(info_frm,text="ID number: ",font=self.Font)
id_lbl.grid(row=6,column=0,padx=3,pady=5,sticky=W)

name_entry=ttk.Entry(info_frm,textvariable=self.v_idno,font=self.Font,width=20)
name_entry.grid(row=6,column=1,padx=0,pady=10,sticky=W)

email_lbl=Label(info_frm,text="Email: ",font=self.Font)
email_lbl.grid(row=7,column=0,padx=3,pady=5,sticky=W)

email_entry=ttk.Entry(info_frm,textvariable=self.v_email,font=self.Font,width=20)
email_entry.grid(row=7,column=1,padx=0,pady=10,sticky=W)

dpass_lbl=Label(info_frm,text="Password: ",font=self.Font)
dpass_lbl.grid(row=8,column=0,padx=3,pady=5,sticky=W)
dpass_entry=ttk.Entry(info_frm,textvariable=self.v_password,show="*",font=self.Font,width=20)
dpass_entry.grid(row=8,column=1,padx=0,pady=10,sticky=W)
```

```
cpass_lbl=Label(info_frm,text="Confirm: ",font=self.Font)
cpass_lbl.grid(row=9,column=0,padx=3,pady=5,sticky=W)

cpass_entry=ttk.Entry(info_frm,textvariable=self.v_confirm,show="*",font=self.Font,width=20)
cpass_entry.grid(row=9,column=1,padx=0,pady=5,sticky=W)

chk_frm=Frame(info_frm)
chk_frm.place(x=20,y=480,width=350,height=55)

chk_btn=Checkbutton(chk_frm,variable=self.v_check,text="Agree Our terms & Condition",
font=("monospace 10 bold"),onvalue=1,offvalue=0)
chk_btn.grid(row=0,column=0,padx=10,pady=1,sticky=W)

self.chk_lbl=Label(chk_frm,text="",font=("monospace 10 bold"),fg="red")
self.chk_lbl.grid(row=1,column=0,padx=10,pady=0,sticky=W)

btn_frm=Frame(info_frm)
btn_frm.place(x=110,y=540,width=400,height=50)

save_btn=Button(btn_frm,text="Save",command=self.validation,font=("monospace 15 bold"),
width=7,cursor="hand2",bg="green",fg="blue",activebackground="black",activeforeground="white")
save_btn.grid(row=0,column=0,padx=10,pady=0,sticky=W)

verify_btn=Button(btn_frm,text="Verify",command=self.verifydata,font=("monospace 15 bold"),
width=7,cursor="hand2",bg="green",fg="blue",activebackground="black",activeforeground="white")
verify_btn.grid(row=0,column=1,padx=10,pady=0,sticky=W)
```

```
        clear_btn=Button(btn_frm,text="Clear",command=self.cleardata,font=("monospace 15 bold"),width=7,
        cursor="hand2",bg="green",fg="blue",activebackground="black",activeforeground="white")
        clear_btn.grid(row=0,column=2,padx=10,pady=0,sticky=W)

def validnanme(self,name):
        if name.isalnum():
            return True
        elif name=="":
            return True
        else:
                self.speach.say("This is Not Allowed")
                self.speach.runAndWait()
                messagebox.showerror("invalid","This is Not Allowed"+name[-1])


def validcontact(self,contact):
        if contact.isdigit():
                return True
        elif len(str(contact))==0:
                return True
        else:
                self.speach.say("Invalid Entry")
                self.speach.runAndWait()
                messagebox.showerror("invalid","Invalid entry")
                return False
```

```
def validemail(self,email):
    if len(email)>7:
        if re.match('^[a-z0-9]+[\._]?[a-z0-9]+[@]\w+[.]\w{2,3}$',email):
            return True
        else:
            self.speach.say("invalid email enter valid user email (demo@gamil.com)")
            self.speach.runAndWait()
            messagebox.showwarning("Alert","invalid email enter valid user email (demo@gamil.com)")
            return False
    else:
        self.speach.say("Email length is too small")
        self.speach.runAndWait()
        messagebox.showerror("invalid","Email length is too small")
        return False
```

```
def validpassword(self,password):
    if len(password)<=11:
        if re.match("^(?=.*[0-9])(?=.*[A-Z])(?=.*[^a-bA-B0-9])",password):
            return True
        else:
            self.speach.say("Enter valid a password")
            self.speach.runAndWait()
            messagebox.showinfo("invalid","Enter Valid Password (Xyz@123)")
            return False
    else:
        self.speach.say("Length try you exceed")
        self.speach.runAndWait()
        messagebox.showerror("invalid","Length try you exceed")
        return False
```

```
def validation(self):

    if self.v_name.get()=="":
            self.speach.say("Plese Enter You name")
            self.speach.runAndWait()
            messagebox.showerror("Error","Please Enter Your Name",parent=self.root)

    elif self.v_contact.get()=="" or len(self.v_contact.get())!=10:
            self.speach.say("Plese Enter Your Valid Contact number")
            self.speach.runAndWait()
            messagebox.showerror("Error","Please Enter Your Valid Contact Number",parent=self.root)

    elif self.v_gender.get()=="":
            self.speach.say("Plese Select Your Gender")
            self.speach.runAndWait()
            messagebox.showerror("Error","Please Select Your Gender",parent=self.root)

    elif self.v_country.get()=="" or (self.v_country.get())=="Select Your Country":
            self.speach.say("Plese Select Your Country Name")
            self.speach.runAndWait()
            messagebox.showerror("Error","Please Select Your Country Name",parent=self.root)

    elif self.v_id.get()=="Select Your ID":
            self.speach.say("Plese Swelect Your Td Type")
            self.speach.runAndWait()
            messagebox.showerror("Error","Please Select Your Id Type",parent=self.root)
```

```
elif self.v_idno=="":
        self.speach.say("Plese Enter Your Id Number")
        self.speach.runAndWait()
        messagebox.showerror("Error","Please Enter Your ID Number",parent=self.root)

elif len(self.v_idno.get())!=14:
        self.speach.say("Plese Enter Your 14 Digit Number")
        self.speach.runAndWait()
        messagebox.showerror("Error","Please Enter Your 14 digit",parent=self.root)

elif self.v_email.get()=="":
        self.speach.say("Plese Enter Your Email id")
        self.speach.runAndWait()
        messagebox.showerror("Error","Please Enter Your Email id",parent=self.root)


elif self.v_password.get()=="":
        self.speach.say("Plese Enter Your Password")
        self.speach.runAndWait()
        messagebox.showerror("Error","Please Enter Your Password",parent=self.root)

elif self.v_confirm.get()=="":
        self.speach.say("Plese Enter Your confirm Password")
        self.speach.runAndWait()
        messagebox.showerror("Error","Please Enter Your Confirm Password",parent=self.root)
```

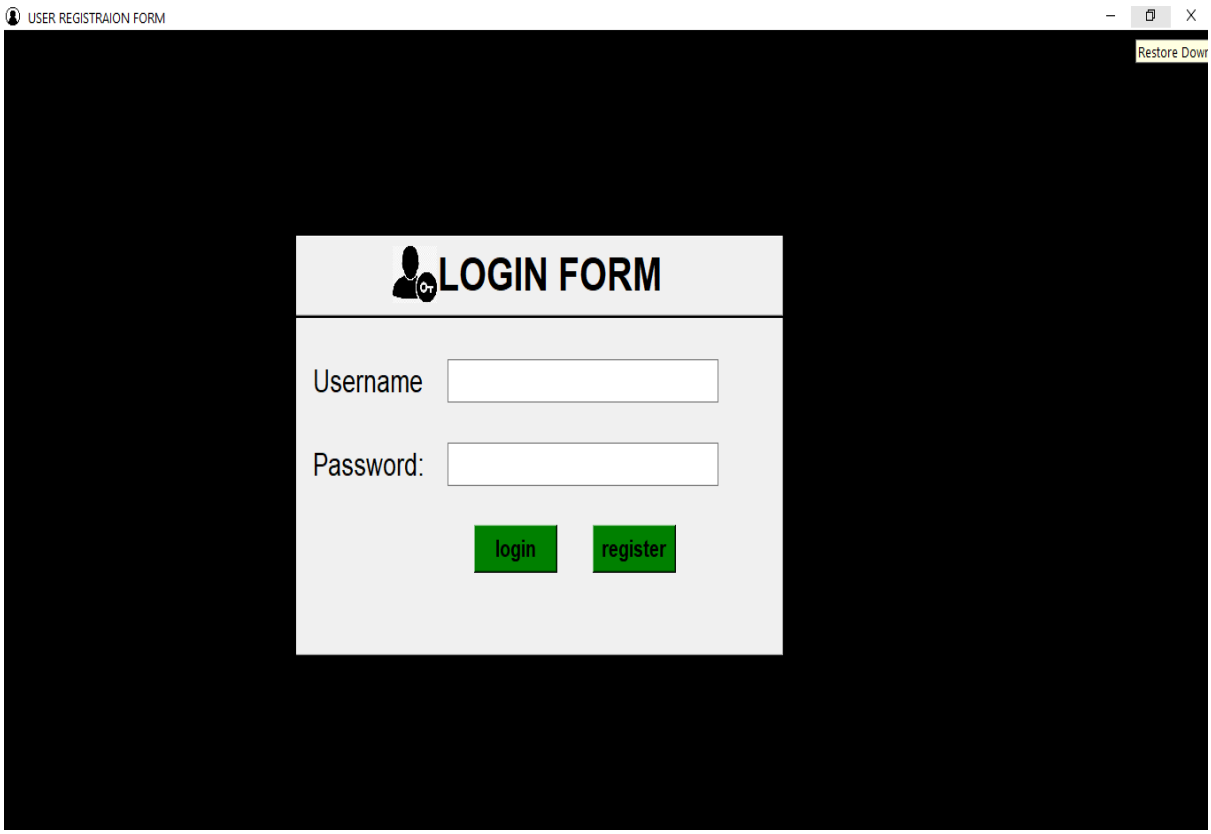
```
elif self.v_password.get()!= self.v_confirm.get():
        self.speach.say("Password & Confirm Passwod must be same")
        self.speach.runAndWait()
        messagebox.showerror("Error","Password & Confirm Passwod must be same",parent=self.root)

elif self.v_email.get()!= None and self.v_password.get()!= None:
        x =self.validemail(self.v_email.get())
        y =self.validpassword(self.v_password.get())

if (x == True) and (y == True):

        if self.v_check.get()==0:
                self.speach.say("Please Agree Our terms & Condition")
                self.speach.runAndWait()
                self.chk_lbl.config(text="Please Agree Our terms & Condition",fg="red")

        else:
                self.chk_lbl.config(text="checked",fg="green")
                try:
                        my_connection=mysql.connector.connect(host="localhost",username="root",
                        password="123456789",database="tkdb")
                        my_cursur=my_connection.cursor()
                        query=("select * from register where email=%s")
                        value=(self.v_email.get(),)
                        my_cursur.execute(query,value)
                        row=my_cursur.fetchone()
                        if row!=None:
```

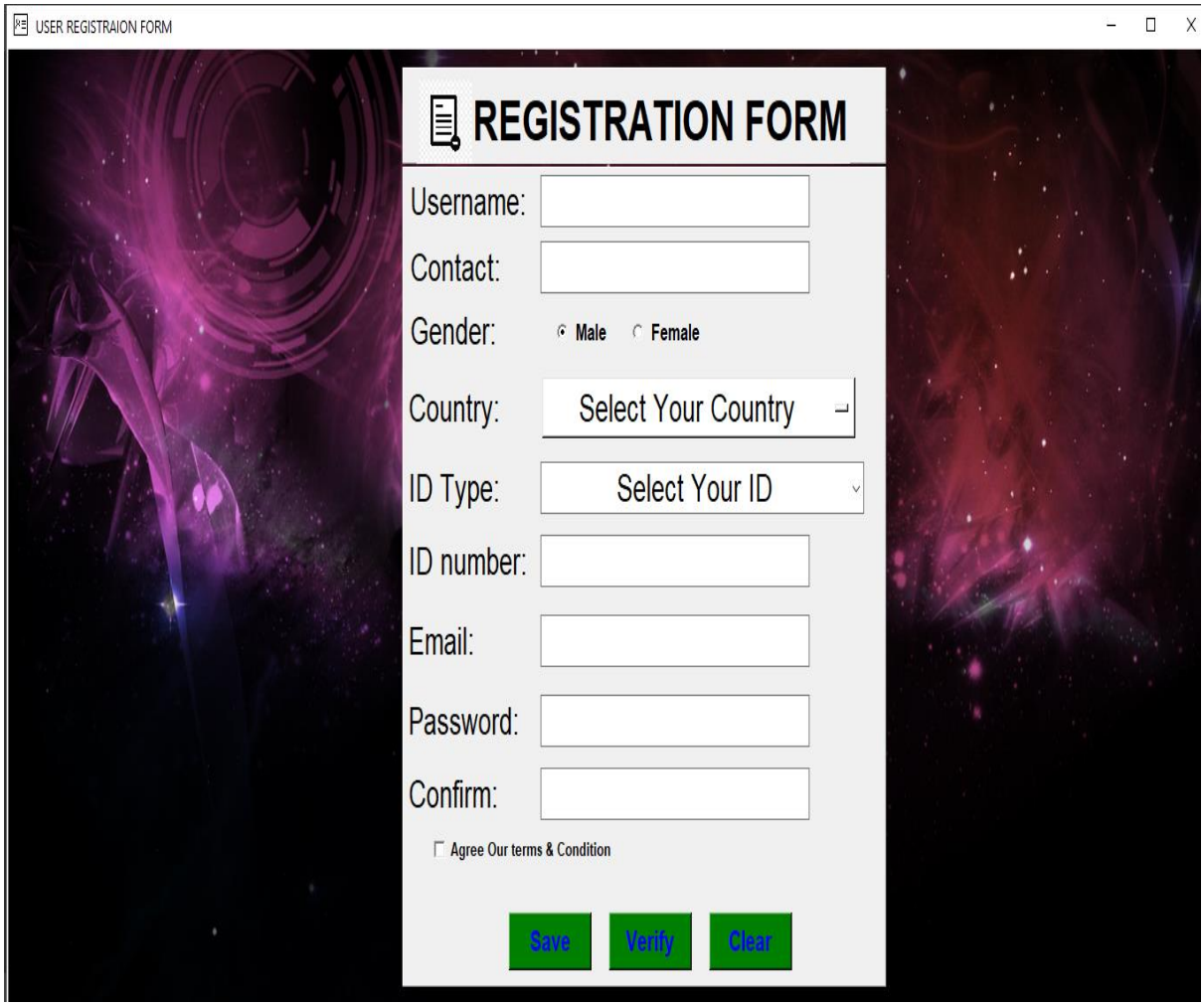
```
        if row!=None:
            messagebox.showerror("Error","user already exist plaese try another email")
        else:
            my_cursur.execute("insert into register values(%s,%s,%s,%s,%s,%s,%s,%s)",(
                                                    self.v_name.get(),
                                                    self.v_contact.get(),
                                                    self.v_gender.get(),
                                                    self.v_country.get(),
                                                    self.v_id.get(),
                                                    self.v_idno.get(),
                                                    self.v_email.get(),
                                                    self.v_password.get(),


            ))

            my_connection.commit()
            my_connection.close()


    except Exception as e:
            messagebox.showerror("Error",f"Due to:{str(e)}",parent=self.root)

    self.speach.say("Successfully")
    self.speach.runAndWait()
    messagebox.showinfo("Successfully",f"Your registration is completed your User name {self.v_name
```

```
def verifydata(self):
        data=(f"Name:  {self.v_name.get()}\n,Contact:  {self.v_contact.get()}\nGender:  {self.v_gender.get()}\nCountry:  {se
        messagebox.showinfo("Details",data)

def cleardata(self):
        self.v_name.set("")
        self.v_contact.set("")
        self.v_gender.set("Male")
        self.v_country.set("Select Your Country")
        self.v_id.set("Select Your Id")
        self.v_idno.set("")
        self.v_email.set("")
        self.v_password.set("")
        self.v_confirm.set("")
        self.v_check.set(0)

def login(self):
        if self.user.get()=="" or self.password.get()=="":
                messagebox.showerror("Error","All fields sre required",parent=self.root)

        elif self.user.get()=="talha" or self.password.get()=="Talha@123":
                messagebox.showwarning("succes","welcome bro your prgram is run")

        else:
                try:
                        my_connection=mysql.connector.connect(host="localhost",username="root",password="123456789",database
                        my_cursur=my_connection.cursor()
```

```
try:
        my_connection=mysql.connector.connect(host="localhost",username="root",
        password="123456789",database="mydata")
        my_cursur=my_connection.cursor()
        my_cursur.execute("insert into register values(%s,%s)",(
                                                        self.user.get(),
                                                        self.password.get()))

        row=my_cursur.fetchone()
        if row==None:
                messagebox.showerror("Error","Invalid user & password")
        else:
                open_main=messagebox.askyesno("YesNo","Access only Admin")
                if open_main>0:
                        print("your programm  in done")
                else:
                        if not open_main:
                                return
        my_connection.commit()
        my_connection.close()

except Exception as e:
        messagebox.showerror("Error",f"Due to:{str(e)}",parent=self.root)
```

```
if __name__=="__main__":
    root=Tk()
    obj=Base(root)
    root.mainloop()
```


USER REGISTRAION FORM
Restore Down
LOGIN FORM
Username
Password:
login
register


USER REGISTRAION FORM
REGISTRATION FORM
Username:
Contact:
Gender:
Male
Female
Country:
Select Your Country
ID Type:
Select Your ID
ID number:
Email:
Password:
Confirm:
Agree Our terms & Condition
Save
Verify
Clear

REFERENCE AND BIBLIOGRAPHY

https://www.w3schools.com/python/

https://www.youtube.com/

https://stackoverflow.com/