 CERTIFICATE

This is to certify that the project entiled “STUDENT MANAGEMENT SYSTEM” is a record of bonafide work carried out by ” Nisha Yadav 12th M1,Ananya Srivastava 12th M1,Shreyanshi Singh 12th M3”.In partial fulfillment of the requirements in COMPUTER SCIENCE prescribed by CBSE for CBSE BOARD 2020-2021 in the school Sant Atulanand Residential Academy,Holapur ,Varanasi,Uttar Pradesh 221003.

DATE PRINCIPAL

INTERNAL EXTERNAL

EXAMINAR EXAMINARACKNOWLEDGMENT

We wish to express our sincere thanks to Mr.AVINASH PANDEY Principal, Sant Atulanand Residential Academy,Holapur ,Varanasi,Uttar Pradesh for guiding us to cause the successful outcome of this project work.

We wish to express our deep and profound sense of gratitude to our guide/teacher Mr.sunil sir ,For her expert help and valuable guidance,comments and suggestions.

We also place on record ,our sincere Gratitude to one and all who, Directly or Indirectly,Have Lent

their helping hand in this venture.

SYSTEM REQUIREMENT

Software:-

* MYSQL
* PYTHON
* COMMAND PROMPT

COMMAND IN COMMAND PROMPT:-

1)pip install pymysql

2)pip install pillow

3) pip install mysql.connector-python

Query:- CREATE DATABASE stm4;

USE stm4;

CREATE TABLE students

(

roll\_no INT(11) PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(150),

gender CHAR(30),

contact VARCHAR (12),

dob VARCHAR(30),

addr VARCHAR(250)

);

WINDOW FOR LOGINN

CODING:-

import os

from subprocess import call

import sys

from PIL import Image, ImageTk

try:

from Tkinter import \*

except ImportError:

from tkinter import \*

try:

import ttk

py3 = False

except ImportError:

import tkinter.ttk as ttk

py3 = True

def click\_list():

call(["python", "adminlogin.py"])

def click\_mnbv():

call(["python","studentlogin.py"])

class SCHOOL\_LOGIN:

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

root = Tk()

root.geometry("998x600+250+50")

root.title("SMS")

root.configure(background="#d9d9d9")

root.configure(highlightbackground="#d9d9d9")

root.configure(highlightcolor="black")

root.resizable(0,0)

self.menubar = Menu(root,font=("roman"),bg="gray",fg="black")

root.configure(menu = self.menubar)

############## background image ###################

self.bg\_pic1 = ImageTk.PhotoImage(file="quote.jpg")

bgl\_lbl=Label(root,image=self.bg\_pic1).pack()

################ image #######################

self.pic = ImageTk.PhotoImage(file="admin.jpg")

labelimage=Label(

root,image=self.pic,fg="green",font=("comic sans MS",30,"bold"),relief=RIDGE,background="#ffffff",bd=1)

labelimage.place(x=80,y=85,width=220,height=200)

self.pic1 = ImageTk.PhotoImage(file="stu.jpg")

labelimage=Label(

root,image=self.pic1,fg="green",font=("comic sans MS",30,"bold"),relief=RIDGE,background="#ffffff",bd=1)

labelimage.place(x=80,y=340,width=220,height=200)

################# buttons ########################

self.Message =Label(root,text="WELCOME",bg="black",fg="white",font=("times new roman",30,"bold")).place(x=82,y=10,width=300,height=60)

self.Button1 =Button(root,text="ADMIN LOGIN",width=556,bg="black",fg="white",font=("times new roman",20,"bold"),command=click\_list).place(x=320,y=150,width=240,height=50)

self.Button2 = Button(root,text="STUDENT LOGIN",width=556,bg="black",fg="white",font=("times new roman",20,"bold"),command=click\_mnbv).place(x=320,y=400,width=240,height=50)

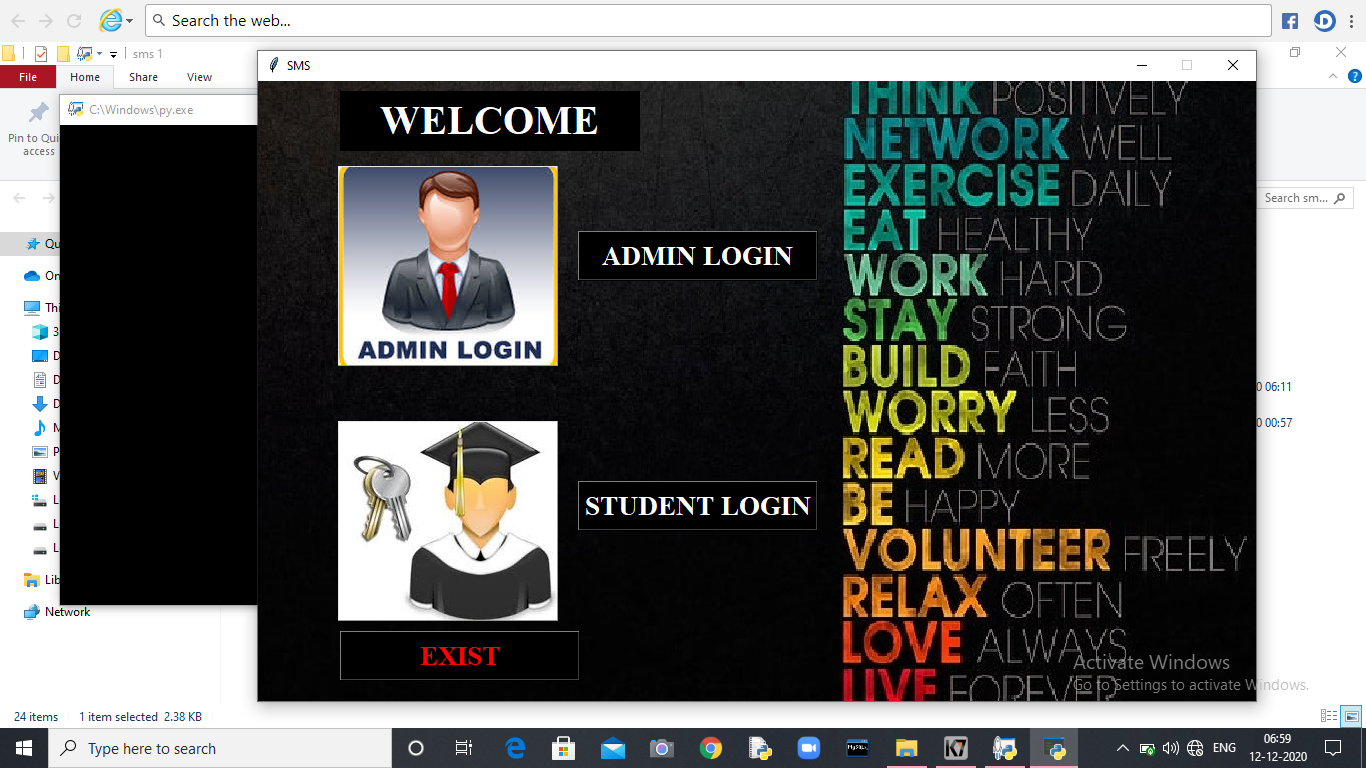
self.Button3 =Button(root,text="EXIST",width=556,bg="black",fg="red",font=("times new roman",20,"bold"),command=quit).place(x=82,y=550,width=240,height=50)

root.mainloop()

if \_\_name\_\_ == '\_\_main\_\_':

GUUEST=SCHOOL\_LOGIN()

OUTPUT:-



ADMINLOGINN:-

CODING;-

import os

from subprocess import call

import sys

from PIL import Image, ImageTk

try:

from Tkinter import \*

except ImportError:

from tkinter import \*

try:

import ttk

py3 = False

except ImportError:

import tkinter.ttk as ttk

py3 = True

def click\_list():

call(["python", "adminlogin.py"])

def click\_mnbv():

call(["python","studentlogin.py"])

class SCHOOL\_LOGIN:

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

root = Tk()

root.geometry("998x600+250+50")

root.title("SMS")

root.configure(background="#d9d9d9")

root.configure(highlightbackground="#d9d9d9")

root.configure(highlightcolor="black")

root.resizable(0,0)

self.menubar = Menu(root,font=("roman"),bg="gray",fg="black")

root.configure(menu = self.menubar)

############## background image ###################

self.bg\_pic1 = ImageTk.PhotoImage(file="quote.jpg")

bgl\_lbl=Label(root,image=self.bg\_pic1).pack()

################ image #######################

self.pic = ImageTk.PhotoImage(file="admin.jpg")

labelimage=Label(

root,image=self.pic,fg="green",font=("comic sans MS",30,"bold"),relief=RIDGE,background="#ffffff",bd=1)

labelimage.place(x=80,y=85,width=220,height=200)

self.pic1 = ImageTk.PhotoImage(file="stu.jpg")

labelimage=Label(

root,image=self.pic1,fg="green",font=("comic sans MS",30,"bold"),relief=RIDGE,background="#ffffff",bd=1)

labelimage.place(x=80,y=340,width=220,height=200)

################# buttons ########################

self.Message =Label(root,text="WELCOME",bg="black",fg="white",font=("times new roman",30,"bold")).place(x=82,y=10,width=300,height=60)

self.Button1 =Button(root,text="ADMIN LOGIN",width=556,bg="black",fg="white",font=("times new roman",20,"bold"),command=click\_list).place(x=320,y=150,width=240,height=50)

self.Button2 = Button(root,text="STUDENT LOGIN",width=556,bg="black",fg="white",font=("times new roman",20,"bold"),command=click\_mnbv).place(x=320,y=400,width=240,height=50)

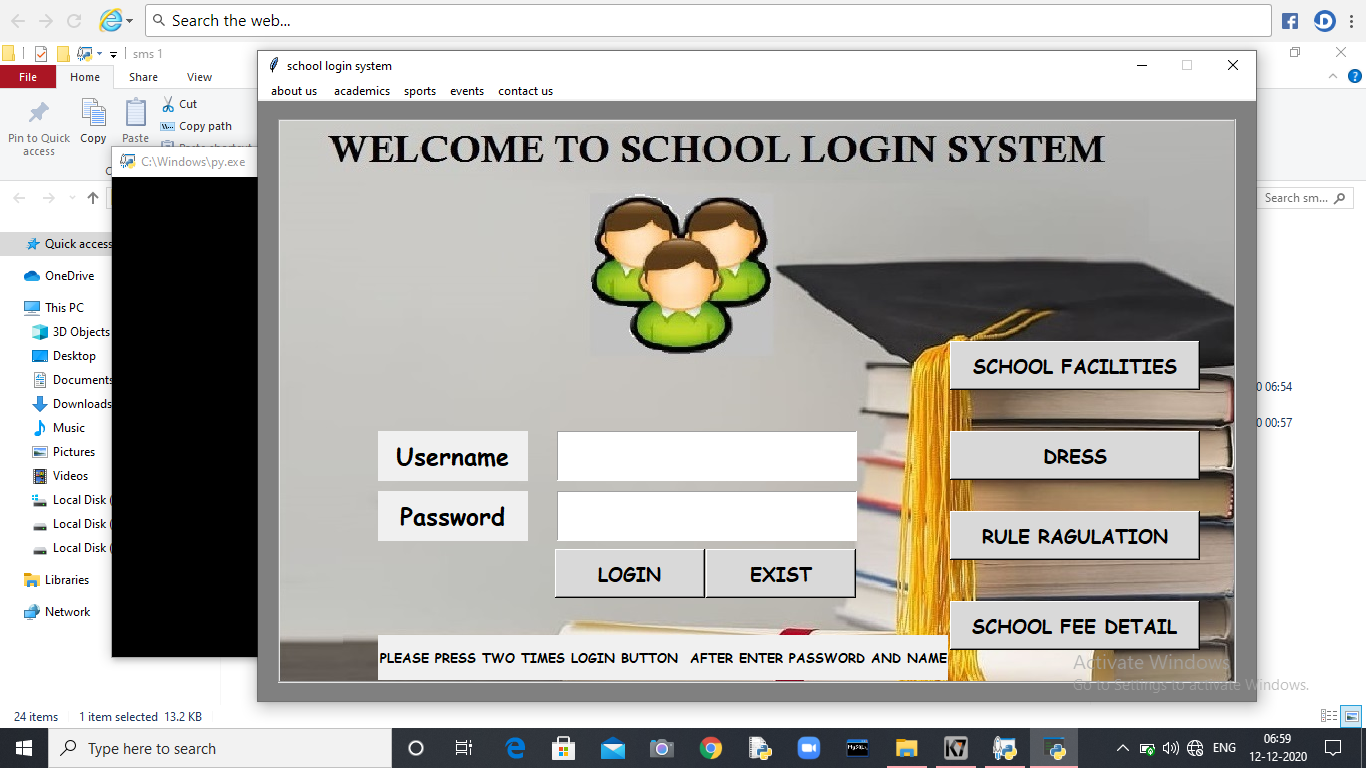
self.Button3 =Button(root,text="EXIST",width=556,bg="black",fg="red",font=("times new roman",20,"bold"),command=quit).place(x=82,y=550,width=240,height=50)

root.mainloop()

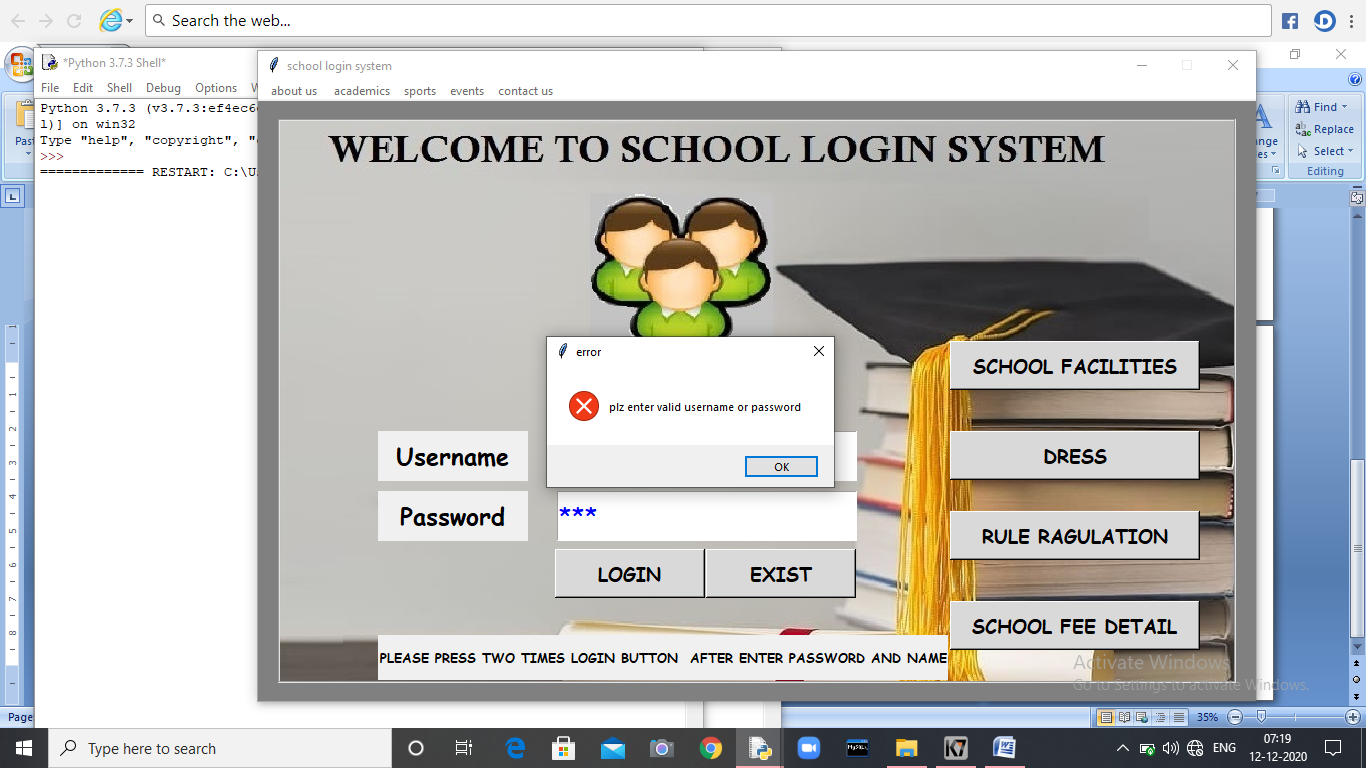
if \_\_name\_\_ == '\_\_main\_\_':

GUUEST=SCHOOL\_LOGIN()

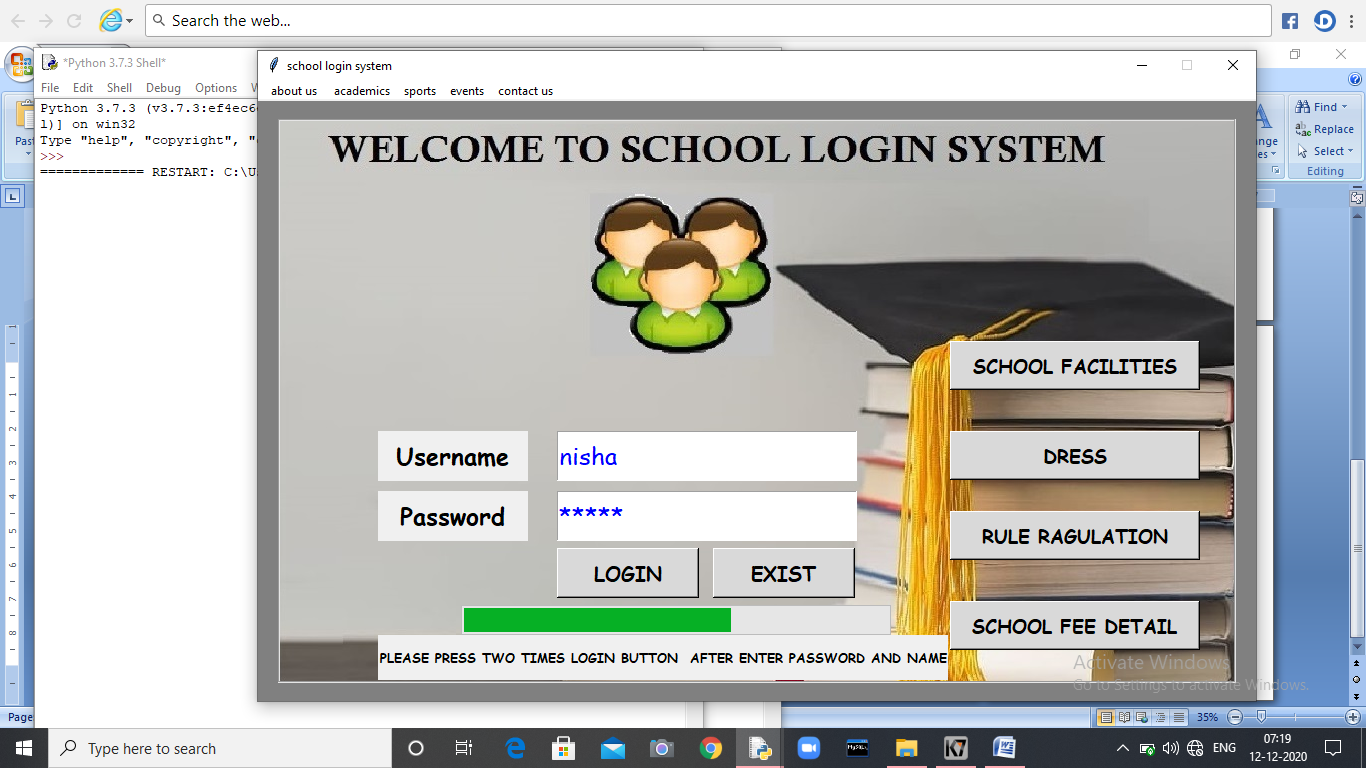
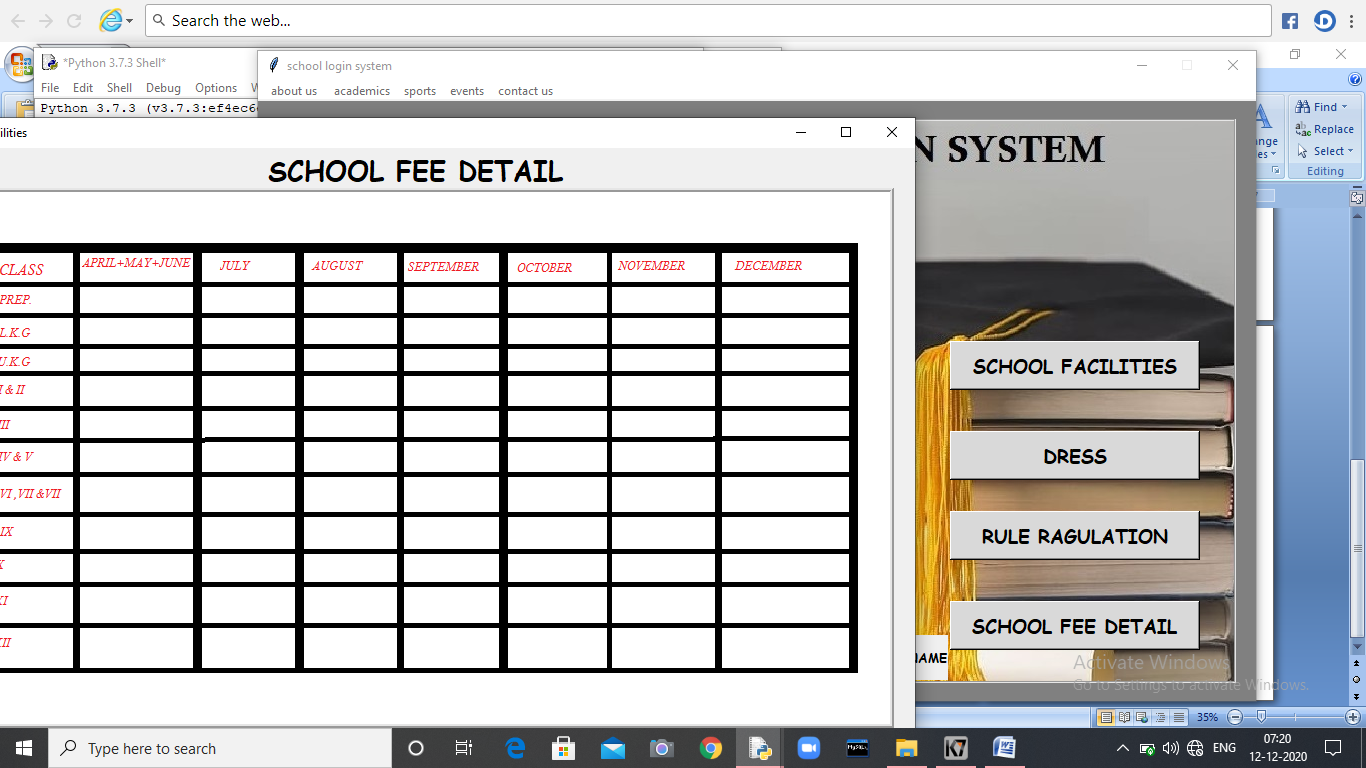
Output:-



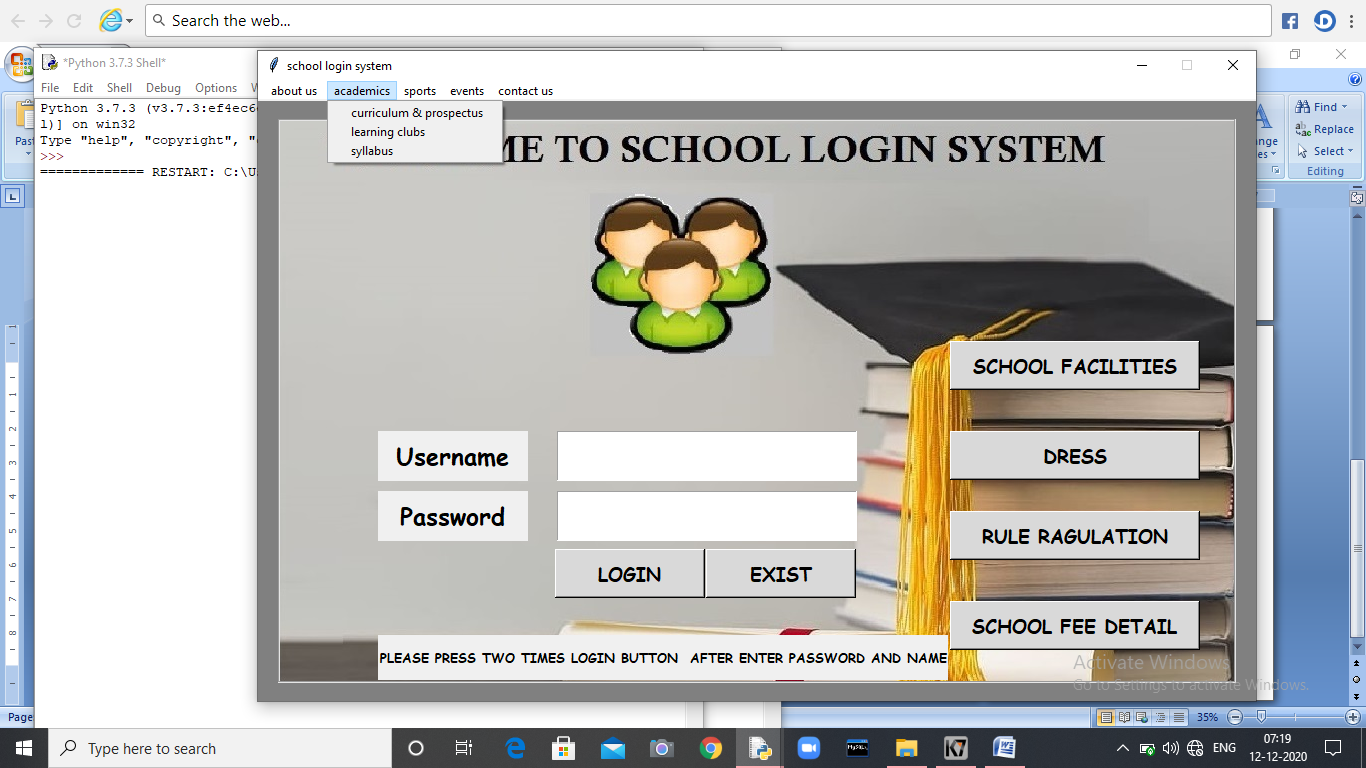
Password or username wrong:-



Password and username correct:-

 BUTTONS:- 

MENU:-



AFTER ADMINLOGIN:-

CODING:-

from tkinter import \*

from tkinter import ttk

import pymysql

import random

import os

from PIL import Image, ImageTk

import pickle

import sys

import os

from subprocess import call

import sys

try:

from Tkinter import \*

except ImportError:

from tkinter import \*

try:

import ttk

py3 = False

except ImportError:

import tkinter.ttk as ttk

py3 = True

class Student\_system:

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

self.root=root

self.root.title("WELCOME TO SCHOOL MANAGEMENT SYSTEM")

self.root.geometry("1350x700+0+0")

#To insert Menu in Window

main\_menu=Menu(self.root)

self.root.config(menu=main\_menu)

#Create File Menu

fileMenu=Menu(main\_menu,tearoff=0)

#add\_command is used to add the Menu items to the Menu

fileMenu.add\_command(label="Log out",command=self.login\_sucess)

#add menu to the menu.

main\_menu.add\_cascade(label=" File ",menu=fileMenu)

fileMenu.add\_separator()

title=Label(self.root,text="WELCOME TO SCHOOL MANAGEMENT SYSTEM",font=("algerian",32,"bold"),bg="sky blue",fg="black")

title.pack(side=TOP,fill=X)

self.bg\_pic1 = ImageTk.PhotoImage(file="121.jpg")

bgl\_lbl=Label(self.root,image=self.bg\_pic1).pack()

#\*\*\*\*\*\*\*\*\*\* All variable\*\*\*\*\*\*\*\*\*

self.roll\_no\_var=StringVar()

self.name\_var=StringVar()

self.email\_var=StringVar()

self.gender\_var=StringVar()

self.contact\_var=StringVar()

self.dob\_var=StringVar()

self.search\_by=StringVar()

self.search\_txt=StringVar()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*Manage Frame\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

lbl\_roll=Label(self.root,text="Roll",bg="pink",fg="black",font=("times new roman",15,"bold"))

lbl\_roll.place(x=30,y=100,width=100,height=40)

txt\_roll=Entry(self.root,textvariable=self.roll\_no\_var,font=("times new roman",10,"bold"),bd=5,relief=GROOVE)

txt\_roll.place(x=200,y=100,width=200,height=40)

lbl\_name=Label(self.root,text="Name",bg="pink",fg="black",font=("times new roman",15,"bold"))

lbl\_name.place(x=30,y=160,width=100,height=40)

txt\_name=Entry(self.root,textvariable=self.name\_var,font=("times new roman",10,"bold"),bd=5,relief=GROOVE)

txt\_name.place(x=200,y=160,width=200,height=40)

lbl\_email=Label(self.root,text="Class",bg="pink",fg="black",font=("times new roman",15,"bold"))

lbl\_email.place(x=30,y=220,width=100,height=40)

txt\_email=Entry(self.root,textvariable=self.email\_var,font=("times new roman",10,"bold"),bd=5,relief=GROOVE)

txt\_email.place(x=200,y=220,width=200,height=40)

lbl\_gender=Label(self.root,text="Gender",bg="pink",fg="black",font=("times new roman",15,"bold"))

lbl\_gender.place(x=30,y=280,width=100,height=40)

combo\_gender=ttk.Combobox(self.root,textvariable=self.gender\_var,font=("times new roman",9,"bold"),state="readonly")

combo\_gender['values']=("Male","Female","Other")

combo\_gender.place(x=200,y=280,width=200,height=40)

lbl\_contact=Label(self.root,text="Contact",bg="pink",fg="black",font=("times new roman",15,"bold"))

lbl\_contact.place(x=450,y=100,width=100,height=40)

txt\_contact=Entry(self.root,textvariable=self.contact\_var,font=("times new roman",10,"bold"),bd=5,relief=GROOVE)

txt\_contact.place(x=600,y=100,width=200,height=40)

lbl\_dob=Label(self.root,text="D.O.B.",bg="pink",fg="black",font=("times new roman",15,"bold"))

lbl\_dob.place(x=450,y=220,width=100,height=40)

txt\_dob=Entry(self.root,textvariable=self.dob\_var,font=("times new roman",10,"bold"),bd=5,relief=GROOVE)

txt\_dob.place(x=600,y=220,width=200,height=40)

lbl\_addr=Label(self.root,text="Address",bg="pink",fg="black",font=("times new roman",15,"bold"))

lbl\_addr.place(x=450,y=160,width=100,height=40)

self.txt\_addr=Text(self.root,width=19,height=3)

self.txt\_addr.place(x=600,y=160,width=200,height=40)

#\*\*\*\*\*\*\*button Frame\*\*\*\*\*\*\*

addbutton=Button(self.root,text="Add",width=8,command=self.add\_student).place(x=1200,y=100,width=100,height=40)

updatebutton=Button(self.root,text="Update",width=8,command=self.update\_data).place(x=1200,y=160,width=100,height=40)

deletebutton=Button(self.root,text="Delete",width=8,command=self.delete\_data).place(x=1200,y=220,width=100,height=40)

clearbutton=Button(self.root,text="Clear",width=8,command=self.clear).place(x=1200,y=280,width=100,height=40)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*Detail Frame\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

self.pic = ImageTk.PhotoImage(file="adminicon.png")#(for image in main(app) page)

labelimage=Label(

self.root,text="ALWAYS BE HAPPY \n AND DO YOUR BEST",image=self.pic,fg="green",font=("comic sans MS",30,"bold"),relief=RIDGE,background="#ffffff",bd=1)

labelimage.place(x=890,y=100,width=250,height=160)

lbl\_search=Label(self.root,text="Search By",bg="pink",fg="black",font=("times new roman",15,"bold"))

lbl\_search.place(x=450,y=280,width=100,height=40)

combo\_search=ttk.Combobox(self.root,width=10,textvariable=self.search\_by,font=("times new roman",10,"bold"),state="readonly")

combo\_search['values']=("roll\_no","name","contact")

combo\_search.place(x=600,y=280,width=100,height=40)

txt\_search=Entry(self.root,textvariable=self.search\_txt,font=("times new roman",10,"bold"),bd=4,relief=GROOVE)

txt\_search.place(x=750,y=280,width=130,height=40)

searchbtn=Button(self.root,text="Search",width=7,pady=3,command=self.search\_data).place(x=900,y=280,width=100,height=40)

showbtn=Button(self.root,text="Show All",width=8,pady=3,command=self.fetch\_data).place(x=1050,y=280,width=100,height=40)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*Table Frame\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Table\_frame=Frame(self.root,bd=4,relief=RIDGE,bg="pink")

Table\_frame.place(x=450,y=350,width=850,height=320)

scroll\_x=Scrollbar(Table\_frame,orient=HORIZONTAL)

scroll\_y=Scrollbar(Table\_frame,orient=VERTICAL)

self.Student\_table=ttk.Treeview(Table\_frame,columns=("roll","name","email","contact","gender","dob","address"),xscrollcommand=scroll\_x.set,yscrollcommand=scroll\_y.set)

scroll\_x.pack(side=BOTTOM,fill=X)

scroll\_y.pack(side=RIGHT,fill=Y)

scroll\_x.config(command=self.Student\_table.xview)

scroll\_y.config(command=self.Student\_table.yview)

self.Student\_table.heading("roll",text="Roll No")

self.Student\_table.heading("name",text="Name")

self.Student\_table.heading("email",text="Class")

self.Student\_table.heading("gender",text="Gender")

self.Student\_table.heading("contact",text="Contact")

self.Student\_table.heading("dob",text="D.O.B.")

self.Student\_table.heading("address",text="Address")

self.Student\_table['show']='headings'

self.Student\_table.column("roll",width=60)

self.Student\_table.column("name",width=110)

self.Student\_table.column("email",width=110)

self.Student\_table.column("gender",width=110)

self.Student\_table.column("contact",width=110)

self.Student\_table.column("dob",width=110)

self.Student\_table.column("address",width=110)

self.Student\_table.pack(fill=BOTH,expand=1)

self.Student\_table.bind("<ButtonRelease-1>",self.get\_cursor)

self.fetch\_data()

######################## def function#######################################

def add\_student(self):

con=pymysql.connect(host='localhost',user='root',password='nishayadav',database='stm4')

cur=con.cursor()

cur.execute("insert into students values(%s,%s,%s,%s,%s,%s,%s);",(

self.roll\_no\_var.get(),

self.name\_var.get(),

self.email\_var.get(),

self.gender\_var.get(),

self.contact\_var.get(),

self.dob\_var.get(),

self.txt\_addr.get('1.0',END)

))

con.commit()

self.fetch\_data()

con.close()

def fetch\_data(self):

con=pymysql.connect(host='localhost',user='root',password='nishayadav',database='stm4')

cur=con.cursor()

cur.execute("select \* from students")

rows=cur.fetchall()

if len(rows)!=0:

self.Student\_table.delete(\*self.Student\_table.get\_children())

for row in rows:

self.Student\_table.insert('',END,values=row)

con.commit()

con.close()

def clear(self):

self.roll\_no\_var.set("")

self.name\_var.set("")

self.email\_var.set("")

self.contact\_var.set("")

self.gender\_var.set("")

self.dob\_var.set("")

self.txt\_addr.delete("1.0",END)

def get\_cursor(self,ev):

cursr\_row=self.Student\_table.focus()

contents=self.Student\_table.item(cursr\_row)

row=contents['values']

self.roll\_no\_var.set(row[0])

self.name\_var.set(row[1])

self.email\_var.set(row[2])

self.contact\_var.set(row[3])

self.gender\_var.set(row[4])

self.dob\_var.set(row[5])

self.txt\_addr.delete("1.0",END)

self.txt\_addr.insert(END,row[6])

def update\_data(self):

con=pymysql.connect(host='localhost',user='root',password='nishayadav',database='stm4')

cur=con.cursor()

cur.execute("update students set name=%s,email=%s,gender=%s,contact=%s,dob=%s,addr=%s where roll\_no=%s ;",(

self.name\_var.get(),

self.email\_var.get(),

self.gender\_var.get(),

self.contact\_var.get(),

self.dob\_var.get(),

self.txt\_addr.get('1.0',END),

self.roll\_no\_var.get()

))

con.commit()

self.fetch\_data()

self.clear()

con.close()

def delete\_data(self):

con=pymysql.connect(host='localhost',user='root',password='nishayadav',database='stm4')

cur=con.cursor()

cur.execute("delete from students where roll\_no=%s;",self.roll\_no\_var.get())

con.commit()

con.close()

self.fetch\_data()

self.clear()

def login\_sucess(self):

global login\_success\_screen

login\_success\_screen = Toplevel(self.root)

login\_success\_screen.title("Success")

login\_success\_screen.geometry("400x150")

Label(login\_success\_screen, text="Logout Success",font=("times new roman",15,"bold")).pack()

Button(login\_success\_screen, text="OK",font=("times new roman",15,"bold"), command=quit).pack()

def search\_data(self):

con=pymysql.connect(host='localhost',user='root',password='nishayadav',database='stm4')

cur=con.cursor()

cur.execute("select \* from students where "+str(self.search\_by.get())+" LIKE '%"+str(self.search\_txt.get())+"%'")

rows=cur.fetchall()

if len(rows)!=0:

self.Student\_table.delete(\*self.Student\_table.get\_children())

for row in rows:

self.Student\_table.insert('',END,values=row)

con.commit()

con.close()

root=Tk()

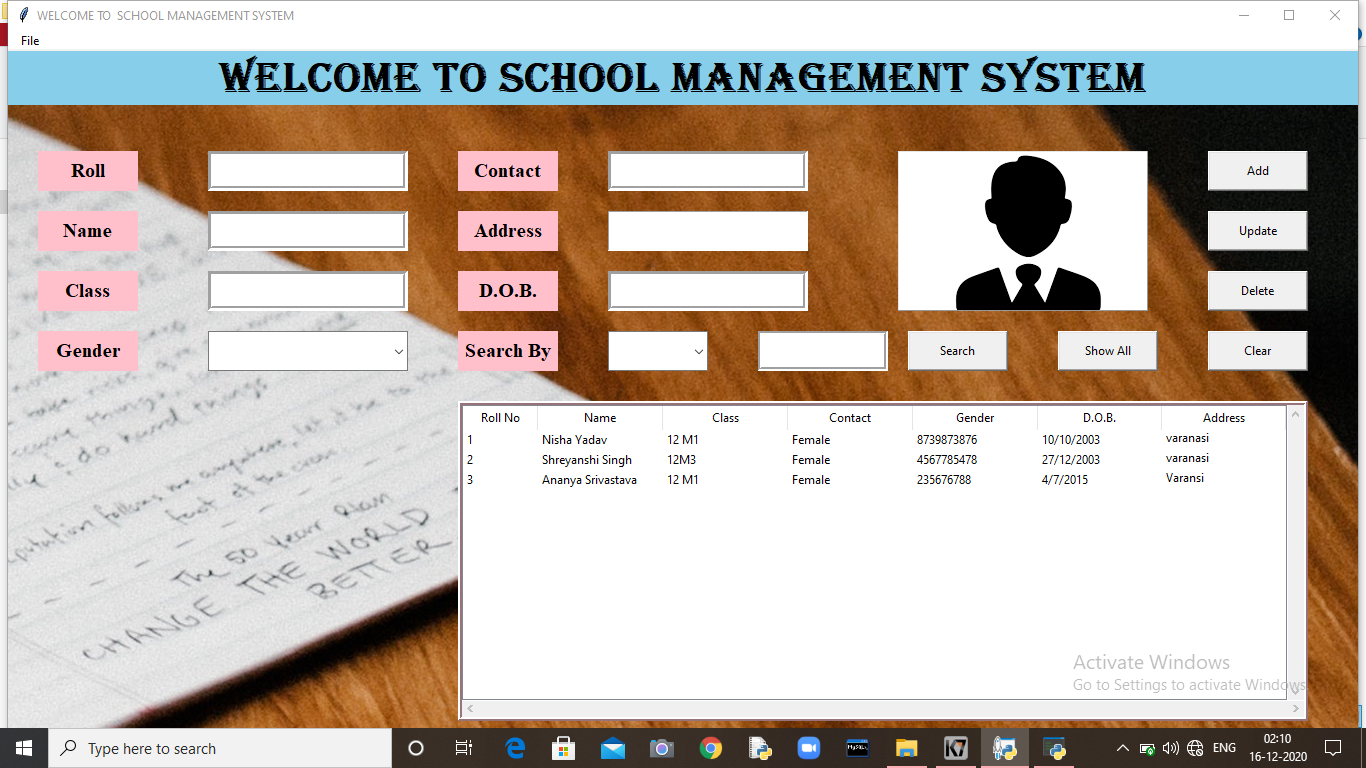
ob=Student\_system(root)

root.mainloop()

if \_\_name\_\_ == '\_\_main\_\_':

school=Student\_system()

OUTPUT:-



STUDENTLOGIN:-

CODING:-

from tkinter import \*

from tkinter import ttk

import pymysql

import random

import datetime

import os

from PIL import Image, ImageTk

from tkinter import messagebox

import pickle

import sys

import os

import mysql.connector as sqltor

from subprocess import call

import sys

try:

from Tkinter import \*

except ImportError:

from tkinter import \*

try:

import ttk

py3 = False

except ImportError:

import tkinter.ttk as ttk

py3 = True

class Student\_system:

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

self.root=root

self.root.title("WELCOME TO SCHOOL MANAGEMENT SYSTEM")

self.root.geometry("720x500+250+50")

"""#To insert Menu in Window

main menu=Menu(self.root)

self.root.config(menu=main\_menu)

#Create File Menu

fileMenu=Menu(main\_menu,tearoff=0)

#add\_command is used to add the Menu items to the Menu

fileMenu.add\_command(label="Log out",command=self.login\_sucess)

#add\_cascade is used to create a hierarchical Menu to the parent Menu by associating the given menu to the parent menu.

main\_menu.add\_cascade(label=" File ",menu=fileMenu)

fileMenu.add\_separator()

#It is used to add the separator line to the menu.

fileMenu.add\_command(label="Books Entry")

fileMenu.add\_separator()

#It is used to add the separator line to the menu.

fileMenu.add\_command(label="jnnj")"""

#\*\*\*\*\*\*\*\*\*\* All variable\*\*\*\*\*\*\*\*\*

self.roll\_no\_var=StringVar()

self.name\_var=StringVar()

self.email\_var=StringVar()

self.gender\_var=StringVar()

self.contact\_var=StringVar()

self.dob\_var=StringVar()

self.father\_var=StringVar()

self.mother\_var=StringVar()

self.search\_by=StringVar()

self.search\_txt=StringVar()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*Manage Frame\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

global x3

self.bg\_pic1 = ImageTk.PhotoImage(file="gh.png")

bgl\_lbl=Label(self.root,image=self.bg\_pic1).place(x=110,y=0,width=680,height=500)

Table\_frame=Frame(self.root,bd=4,relief=RIDGE,bg="white")

Table\_frame.place(x=50,y=100,width=330,height=350)

lb1=Label(self.root,text=" STUDENT ID",bg="white",fg="black",font=("times new roman",15,"bold"))

lb1.place(x=80,y=300,width=120,height=40)

x3=Entry(self.root,font=("times new roman",10,"bold"),bd=4,relief=GROOVE)

x3.place(x=220,y=300,width=130,height=40)

searchbtn=Button(self.root,text="Login",width=7,bg="white",font=("times new roman",15,"bold"),bd=0,pady=3,command=self.modify).place(x=80,y=380,width=100,height=40)

existbtn=Button(self.root,text="EXIST",width=7,bg="white",font=("times new roman",15,"bold"),bd=0,pady=3,command=quit).place(x=220,y=380,width=100,height=40)

fbtn=Button(self.root,text="Forget Password ?",width=7,bg="white",fg="red",font=("times new roman",15,"bold"),bd=0,pady=3,command=self.er).place(x=190,y=350,width=180,height=30)

lbl2=Button(self.root,text="STUDENT LOGIN",width=7,bg="white",fg="blue",font=("times new roman",15,"bold"),bd=0,pady=3).place(x=120,y=250,width=180,height=30)

self.pic = ImageTk.PhotoImage(file="icon.png")#(for image in main(app) page)

labelimage=Label(

self.root,text="ALWAYS BE HAPPY \n AND DO YOUR BEST",image=self.pic,fg="green",font=("comic sans MS",30,"bold"),relief=RIDGE,background="#ffffff",bd=0)

labelimage.place(x=130,y=120,width=170,height=120)

def modify(self):

global x3,x4

p1=x3.get()

con=pymysql.connect(host='localhost',user='root',password='nishayadav',database='stm4')

cur=con.cursor()

cur.execute('select \* from students where roll\_no=(%s)',(p1,))

dat=cur.fetchall()

a=[]

for i in dat:

a.append(i)

if len(a)==0:

messagebox.showerror("ERROR", "NO DATA FOUND!!")

else:

root6=Toplevel(self.root)

frame=Frame(root6,height=300,width=500,relief=RIDGE,bg="white")

frame.pack()

"""self.bg\_pic2 = ImageTk.PhotoImage(file="4.jpg")

bgl\_lbl=Label(frame,image=self.bg\_pic2,bd=1).place(x=180,y=30,width=130,height=90)"""

self.pic9 = ImageTk.PhotoImage(file="icon.png")#(for image in main(app) page)

labelimage=Label(

frame,image=self.pic9,fg="green",font=("comic sans MS",30,"bold"),relief=RIDGE,background="#ffffff",bd=3)

labelimage.place(x=320,y=70,width=140,height=120)

for i in dat:

lb1=Label(root6,text=" STUDENT DETAIL",bg="white",fg="black",font=("times new roman",25,"bold")).place(x=50,y=20)

lb1=Label(root6,text=" STUDENT PHOTO",bg="white",fg="black",font=("times new roman",15,"bold")).place(x=300,y=200)

name=Label(root6,bg="white",fg="red",text='NAME:-',font=("times new roman",15,"bold"))

name.place(x=50,y=60)

name1=Label(root6,bg="white",fg="blue”, text=i[1],font=("times new roman",15,"bold"))

name1.place(x=150,y=60)

age=Label(root6,bg="white",fg="red",text='CLASS:-',font=("times new roman",15,"bold"))

age.place(x=50,y=90)

age1=Label(root6,bg="white",fg="blue”, text=i[2],font=("times new roman",15,"bold"))

age1.place(x=150,y=90)

gen=Label(root6,bg="white",fg="red",text='CONTACT:-',font=("times new roman",15,"bold"))

gen.place(x=50,y=120)

gen1=Label(root6,bg="white",fg="blue”, text=i[3],font=("times new roman",15,"bold"))

gen1.place(x=170,y=120)

pho=Label(root6,bg="white",fg="red",text='GENDER:-',font=("times new roman",15,"bold"))

pho.place(x=50,y=150)

pho1=Label(root6,bg="white",fg="blue”, text=i[4],font=("times new roman",15,"bold"))

pho1.place(x=180,y=150)

bg=Label(root6,bg="white",fg="red",text='DOB:-',font=("times new roman",15,"bold"))

bg.place(x=50,y=180)

bg1=Label(root6,bg="white",fg="blue”, text=i[5],font=("times new roman",15,"bold"))

bg1.place(x=150,y=180)

dob=Label(root6,bg="white",fg="red",text='ADDRESS:-',font=("times new roman",15,"bold"))

dob.place(x=50,y=210)

dob1=Label(root6,bg="white",fg="blue”, text=i[6],font=("times new roman",15,"bold"))

dob1.place(x=150,y=210)

root6.resizable(False,False)

root6.mainloop()

def er(self):

messagebox.showerror("ERROR", "PLEASE CONTECT WITH ADMIN") def login\_sucess(self):

global login\_success\_screen

login\_success\_screen = Toplevel(self.root)

login\_success\_screen.title("Success")

login\_success\_screen.geometry("400x150")

Label(login\_success\_screen, text="Logout Success",font=("times new roman",15,"bold")).pack()

Button(login\_success\_screen, text="OK",font=("times new roman",15,"bold"), command=quit).pack()

root=Tk()

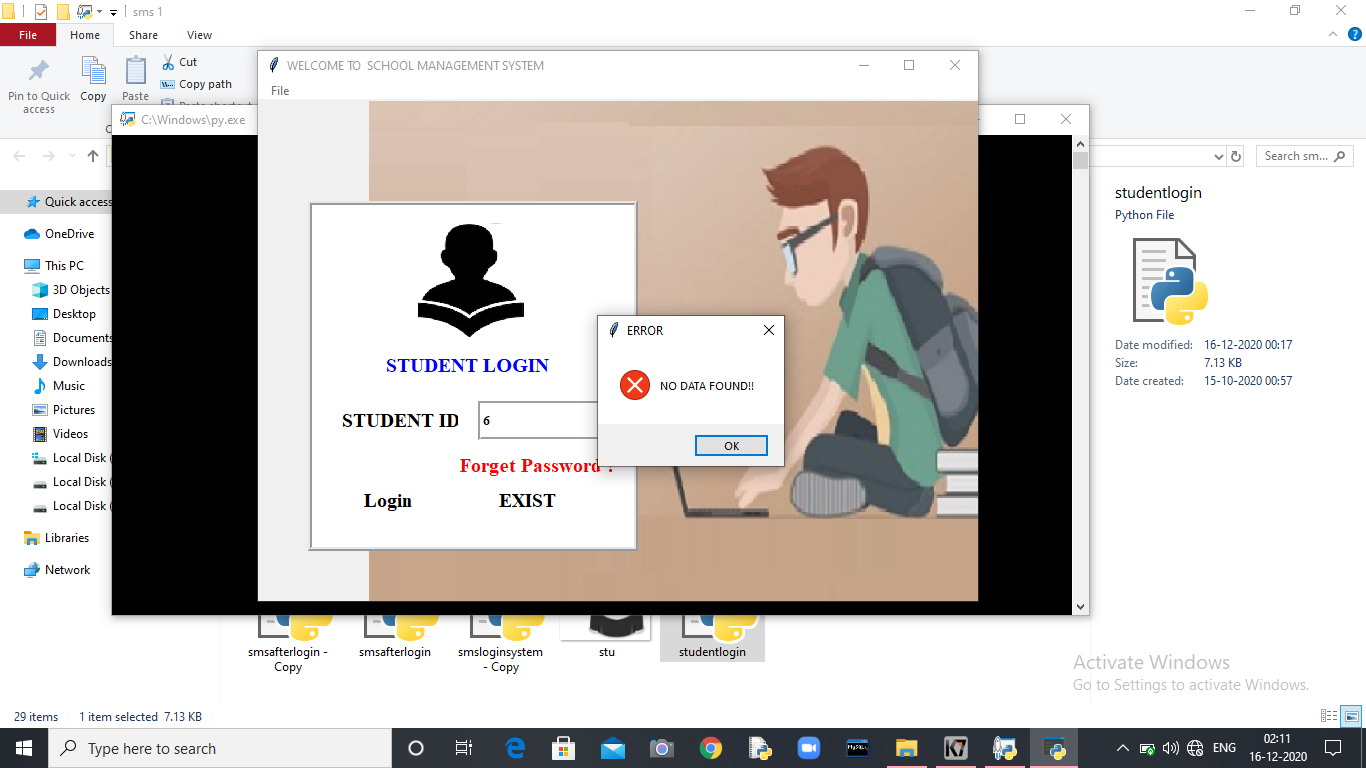
ob=Student\_system(root)

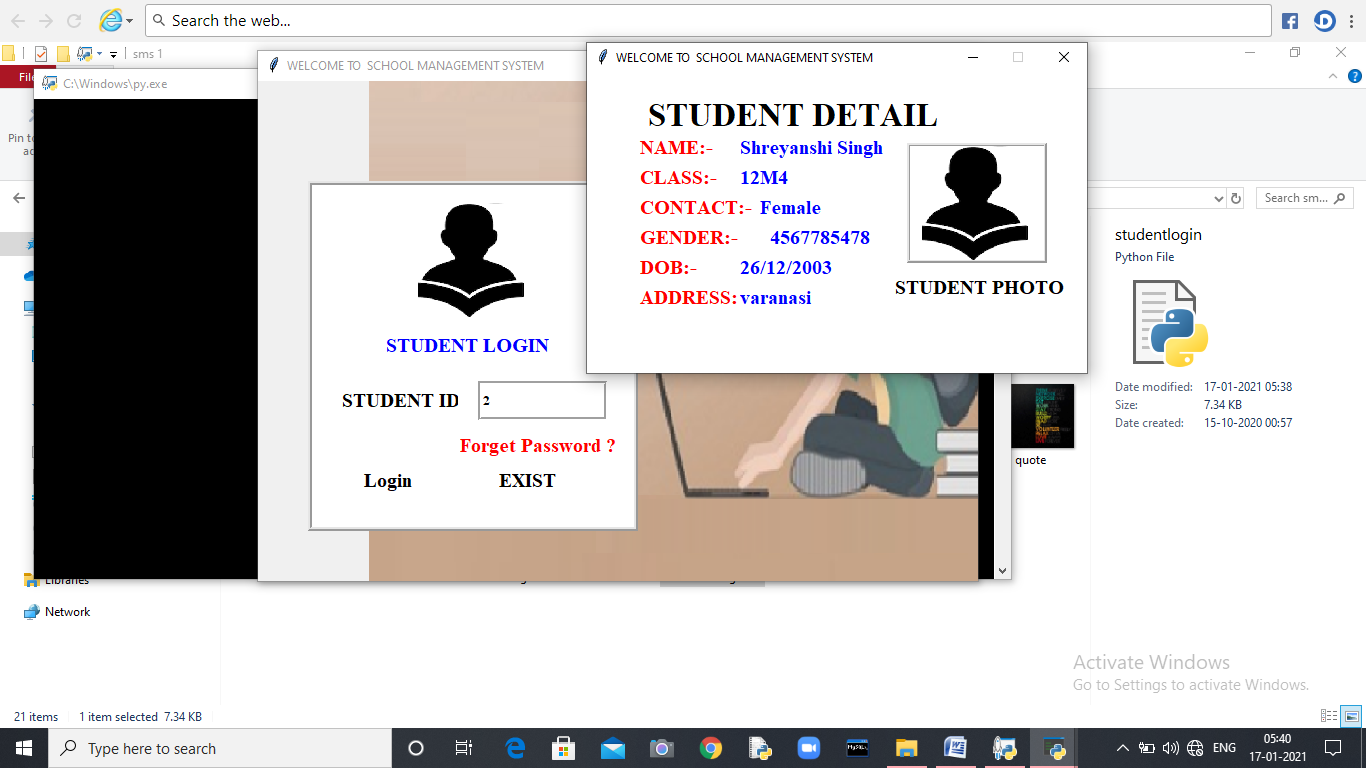
root.mainloop()

if \_\_name\_\_ == '\_\_main\_\_':

school=Student\_system()

OUTPUT:-





BIBLIOGRAPHY

HELP FROM INTERNET



INFORMATION FROM BOOK



HELP FROM TEACHERS

