#\_\_\_\_\_\_\_\_\_\_INTRO PAGE\_\_\_\_\_\_\_\_\_\_\_\_#

from tkinter import \*

from PIL import ImageTk

root=Tk()

root.geometry("1590x800+0+0")

root.title("demo")

root.config(bg="#ea5455")

def lg\_in():

root.destroy()

import register

btn\_img=ImageTk.PhotoImage(file="button\_sign-u.png")

title=Label(text="Welcome to Password Manager",font=("Verdana 15underline",50,"bold"),relief="groove",bg="#ea5455",fg="white")

title.place(x=0,y=150,relwidth=1)

des=Label(text="It gives you a easy access to store, generate , and manage\n your passwords for local applications and online services.",font=("helvetica",25,"italic"),bg="#ea5455",fg='white')

des.place(x=330,y=350)

sign\_up\_btn=Button(image=btn\_img,borderwidth=0,bd=0,command=lg\_in,bg="#ea5455")

sign\_up\_btn.place(x=650,y=550)

root.mainloop()

**File:**#\_\_\_\_\_\_\_\_\_\_\_\_\_\_signuppage\_\_\_\_\_\_\_\_\_\_\_\_\_\_#

from tkinter import \*

from tkinter import messagebox

import mysql.connector

root=Tk()

root.geometry("1590x800+0+0")

root.title("demo")

root.config(bg="#ea5455")

img\_reg=PhoyoImage(file="images/button\_sign\_up.png")

frame1=Frame(bg="white",bd=0)

frame1.place(x=550,y=30,width=500,height=740)

lbl\_sign\_up=Label(text="SIGN UP",font("times new roman",30,"bold"),bg="white",fg="black")

lbl\_sign\_up.place(x=690,y=70)

username=Label(text="enter your email",font=("helvetica",15),bg="white")

username.place(x=600,y=200)

user\_mail=Entry(text="hell",font=("helvetica",15),bg="white",relief="ridge",borderwidth=1)

user\_mail.place(x=600,y=260,width=300)

user\_mail.focus()

paswd=Label(text="New Password",font=("helvetica",15),bg="white")

paswd.place(x=600,y=330)

new\_pswd=Entry(font=("helvetica",15),show="\*",relief="ridge",borderwidth=1)

new\_pswd.place(x=600,y=380,width=300)

cfrm\_paswd=Label(text="confirm password",font=("helvetica",15),bh="white")

cfrm\_paswd.place(x=600,y=440,height=35)

cnfrm\_pswd=Entry(font=("helvetica",15),show="\*",relief="ridge",borderwidth=1)

cnfrm\_pswd.place(x=600,y=490,width=300)

def regs():

if new\_pswd.get()=="" or cnfrm\_pswd.get()=="" or user\_mail.get()=="" :

messagebox.showerror("Error","All Fields are required")

elif new\_pswd.get()!=cnfrm\_pswd.get():

messagebox.showerror("Error","Password and confirm password doen't match")

else:

try:

db=mysql.connector.connect(host='localhost',user='root',passwd='root',database="password\_manager")

cur=db.cursor()

cur.execute("Insert into useraccounts(mail\_id,password)values(%s, %s)",(user\_mail.get(),new\_pswd.get()))

db.commit()

db.close()

root.destroy()

import log\_in

except EXCEPTION as e:

messagebox.showerror("Error",f "Error Due to {e}")

frame1=Frame(bg="white",bd=0)

frame1.place(x=550,y=30,width=500,height=740)

btn\_login=PhotoImage(file="images/button\_log-in.png")

title=Label(text="Login In",bg="white",font=("Californian FB",35,"bold"))

title.place(x=710,y=80)

label=Label(text="username",compound=LEFT,font=("helvetica",18),bg="white")

label.place(x=600,y=200)

email\_txt=Entry(text="hell",font=("helvetica",18),bg="white",relief="sunken",bd=1)

email\_txt.place(x=600,y=260,width=300)

email\_txt.focus()

pass\_lbl=Label(text="Password",font=("helvetica",18),bg="white")

pass\_lbl.place(x=600,y=320)

paswd\_txt=Entry(font=("helvetica",15),show="\*",relief="sunken",bd=1)

paswd\_txt.place(x=600,y=380,width=300)

sign\_in=Button(image=btn\_login,bd=0,bg="white",command=login)

sign\_in.place(x=610,y=500)

root.mainloop()

**file:** #\_\_\_\_\_\_\_\_\_\_\_password manager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#

from tkinter import \*

from PIL import ImageTk

from random import \*

import pyperclip

from tkinter import messagebox

import mysql.connector

from tkinter import ttk

import string

def clear():

website\_entry.delete(0,END)

password\_entry.delete(0,END)

mail\_entry.delete(0,END)

username\_entry.delete(0,END)

##\_\_\_\_\_PASSWORD SAVER\_\_\_\_\_\_\_\_\_\_\_\_\_###

def save():

website=website\_entry.get()

username=username\_entry.get()

password=password\_entry.get()

mail\_id=mail\_entry.get()

if website=="" or username=="" or password=="" or mail\_id=="" :

messagebox.showerror(title="oops",message="please don't leave any fields empty")

else:

is\_ok=messagebox.askyesno(title="Website",message="do you confirm to add the details entered")

if is\_ok:

try:

db=mysql.connector.connect(host='localhost',user='root',passwd='root',database="password\_manager")

cur=db.cursor()

cur.execute("INSERT INTO UserData(Website,Username,Password,mail\_id)VALUES(%s,%s,%s,%s)",(website,username,password,mail\_id))

db.commit()

db.close()

fetch\_data()

clear()

except EXCEPTION as e:

messagebox.showerror("Error",f"Error Due to {e}")

def update():

website=website\_entry.get()

username=username\_entry.get()

password=password\_entry.get()

mail\_id=mail\_entry.get()

try:

db=mysql.connector.connect(host='localhost',user='root',passwd='root',database="password\_manager")

cur=db.cursor()

cur.execute("UPDATE UserData SET Username=%s,Password=%s,mail\_id=%s WHERE Website=%s",(username,password,mail\_id,website))

db.commit()

db.close()

fetch\_data()

clear()

except EXCEPTION as e:

messagebox.showerror("Error",f"Error due to {e}")

def delete():

website=website\_entry.get()

db=mysql.connector.connect(host='localhost',user='root',passwd='root',database="password\_manager")

cur=db.cursor()

cur.execute("DELETE FROM UserData WHERE Website=%s",(website))

db.commit()

db.close()

fetch\_data()

clear()

window=Tk()

window.geometry("1590x800+0+0")

window.title("PASSWORD MANAGER")

window.config(bg="#ea5455")

frame1=Frame(width=1482,height=740,bg="white")

frame1.place(x=20,y=30)

frame2=Frame(frame1,width=600,height=570,bg="white",borderwidth=1,relief="solid")

frame2.place(x=30,y=150)

frame3=LabelFrame(frame1,text="password generator",font=("helvetica",18),width=800,height=260,bg="white",borderwidth=1,relief="solid")

frame3.place(x=650,y=140)

def check():

upper=upper\_var.get()

lower=lower\_var.get()

number=number\_var.get()

symbol=symbol\_var.get()

ps=""

if upper=='on':

ps+=string.ascii\_uppercase

if lower=='on':

ps+=string.ascii\_lowercase

if number=='on':

ps+=string.digits

if symbol=='on':

ps+=string.punctuation

else:

generate\_entry.delete(0,END)

generate\_entry.insert(0,"please check any one option")

return ps

def generate(ev):

pswrd=check()

password=''

for i in range(0,slider.get()):

password=password+choice(pswrd)

generate\_entry.delete(0,END)

password\_entry.delete(0,END)

generate\_entry.insert(0,password)

password\_entry.insert(0,password)

pyperclip.copy(password)

upper\_var=StringVar()

lower\_var=StringVar()

number\_var=StringVar()

symbol\_var=StringVar()

label=Label(frame3,text="customize your password",font=("helvetica",18,"unerline"),bg='white')

label.place(x=100,y=80)

slider=Scale(frame3,label="Password length",font=("helvetica",12),from\_=0,to=50,length=250,resolution=1,orient=HORIZONTAL,troughcolor='white',relief='groove',activebackground="#ea5455",bg="#ffcccb",bd=0,command=generate)

slider.place(x=100,y=140)

upper\_check=CheckButton(frame3,text="Uppercase",variable=upper\_var,font("helvetica",13),bg='white',onvalue='on',offvalue='off')

upper\_check.place(x=380,y=130)

upper\_check.select()

lower\_check = Checkbutton(frame3, text='Lowercase', variable=lower\_var,font=("helvetica", 13), bg='white',onvalue='on', offvalue='off')

lower\_check.place(x=380, y=180)

lower\_check.select()

number\_check = Checkbutton(frame3, text='Numbers', variable=number\_var,font=("helvetica", 13), bg='white',onvalue='on', offvalue='off')

number\_check.place(x=525, y=130)

number\_check.deselect()

symbol\_check = Checkbutton(frame3, text='Symbols', variable=symbol\_var,font=("helvetica", 13), bg='white',onvalue='on', offvalue='off')

symbol\_check.place(x=525, y=180)

symbol\_check.deselect()

generate\_entry = Entry(frame3, width=40, font=("helvetica", 15),relief="ridge", borderwidth=1)

generate\_entry.place(x=150, y=20, height=35)

add\_image = ImageTk.PhotoImage(file="button\_add.png")

update\_img = ImageTk.PhotoImage(file="button\_update.png")

delete\_img = ImageTk.PhotoImage(file="button\_delete.png")

title = Label(text="Password Manager", font=("Oswald", 30, "bold"), bg="white")

title.place(x=590, y=60)

web\_label = Label(frame2, text="Website or App Name", font=("helvetica",20), bg="white")

web\_label.place(x=40, y=50)

username\_label = Label(frame2, text="Username", font=("helvetica", 20), bg="white")

username\_label.place(x=40, y=150)

password\_label = Label(frame2, text="Password", font=("helvetica", 20,), bg="white")

password\_label.place(x=40, y=250)

mail\_label = Label(frame2, text="Email Id", font=("helvetica", 20), bg="white")

mail\_label.place(x=40, y=350)

website\_entry = Entry(frame2, width=40, font=("helvetica", 15), relief="ridge", borderwidth=1)

website\_entry.focus()

website\_entry.place(x=40, y=100, height=25)

username\_entry = Entry(frame2, width=40, font=("helvetica", 15),relief="ridge", borderwidth=1)

username\_entry.place(x=40, y=200, height=25)

password\_entry = Entry(frame2, width=40, font=("helvetica", 15),relief="ridge", borderwidth=1)

password\_entry.place(x=40, y=300, height=25)

mail\_entry = Entry(frame2, width=40, font=("helvetica", 15), relief="ridge", borderwidth=1)

mail\_entry.place(x=40, y=400, height=25)

add\_btn = Button(frame2, image=add\_image, borderwidth=0, bg="white",command=save)

add\_btn.place(x=15, y=480)

update\_btn = Button(frame2, image=update\_img, bg="white", borderwidth=0,command=update)

update\_btn.place(x=210, y=480)

delete\_btn = Button(frame2, image=delete\_img, bg="white", borderwidth=0,command=delete)

delete\_btn.place(x=410, y=480)

# ==================== Treeview================

def get\_cursor(event):

cursor\_row = password\_tree.focus()

contents = password\_tree.item(cursor\_row)

row = contents['values']

clear()

website\_entry.insert(0, row[0])

username\_entry.insert(0, row[1])

password\_entry.insert(0, row[2])

mail\_entry.insert(0, row[3])

frame4 = Frame(frame1, width=800, height=300, bg="white", borderwidth=1,relief="solid")

frame4.place(x=650, y=420)

style = ttk.Style()

style.configure("Treeview", rowheight=28)

style.map("Treeview", background=[('selected', '#ea5455')])

password\_tree = ttk.Treeview(frame4, columns=("website", "username","password", "mail\_id"))

password\_tree.tag\_configure('oddrow', background='white')

password\_tree.tag\_configure('evenrow', background='#ffcccb')

password\_tree.column("website", width=140)

password\_tree.heading("username", text="Username", anchor="w")

password\_tree.heading("website", text="Website", anchor="w")

password\_tree.heading("password", text="Password", anchor="w")

password\_tree.heading("mail\_id", text="Email", anchor="w")

password\_tree['show'] = 'headings'

password\_tree.pack(expand=True, fill=BOTH)

password\_tree.bind("<ButtonRelease-1>", get\_cursor)

def fetch\_data():

count = 0

db = mysql.connector.connect(host='localhost',user='root', passwd='root',database="password\_manager")

cur = db.cursor()cur.execute("SELECT \* FROM UserData ")

rows = cur.fetchall()

if len(rows) != 0:

password\_tree.delete(\*password\_tree.get\_children())

for row in rows:

if count % 2 == 0:

password\_tree.insert(parent='', index=END, values=row, tags='evenrow')

db.commit()

else:

password\_tree.insert(parent='', index=END, values=row, tags='oddrow')

db.commit()

count += 1

db.close()

fetch\_data()

window.mainloop()

**File:** #\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_datacreation&connection\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#

# ---------------Database Creation------------

import mysql.connect

db = mysql.connector.connect(host='localhost',user='root',passwd='root')

cur=db.cursor()

cur.execute("CREATE DATABASE password\_manager")

#----------Table Creation----------

import mysql.connect

db = mysql.connector.connect(host='localhost',user='root', passwd='root',database="password\_manager")

cur = db.cursor()

cur.execte(“CREATE TABLE useraccounts (userID int PRIMARY KEYAUTO\_INCREMENT, Username VARCHAR(30), Password VARCHAR(30)”)

cur.execte(“CREATE TABLE userdata (Website VARCHAR(50), UsernameVARCHAR(30), Password VARCHAR(30), Email\_id VARCHAR(30)” )

db.commit()