ABSTRACT

The project titled ‘EXPRESSIOUE’ is made in ‘Python’.

Have you ever done something extremely stupid or funny that you wanted to share with the world but couldn't because that would make you like a fool. Well, now there's a solution for that. We have launched EXPRESSIOUE, an app that allows its users to confess anything and everything and that too anonymously.

EXPRESSIOUE, acts like a platform where anyone and everyone can share their funny, dark secrets, get over the guilt of some unpleasant event, ease one’s regrets, rejoice some silly moments and much more. The application has been very thoughtfully created and has the potential to keep you hooked for days and nights.

The module used in the project are Tkinter , MySQL

TABLE OF CONTENTS

INTRODUCTION

Our project is a Tkinter program written in Python. Our objective for this project is to create a user friendly interface. It contains three dropdown boxes i.e., Register and Login dropdown box, viewing post dropdown box and posting your own post dropdown box. The user will move to tab respectively.

The app comes packed with a lot of interesting features which makes expressing yourself even more interesting and fun. The app even allows you to go through other's post like the ones posted by your friends, local posts (from people around your town or city) or the most popular ones on the platform. To make things more interesting, fun and spicy, the platform provides you to post your point of views with other users on the platform and talk about crazy incidents like crushes, pranks etc. that too without anyone's feeling get hurt.

Expressioue will be like a blessing in today's modern world where we often end up doing fun, stupid, crazy things which we don't want people associated to us to ever know.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. There are various widgets like button, canvas, checkbutton, entry, etc. that are used to build the python GUI applications. We used Tkinter to make different modes in buttons, and message box.

Python Tkinter – MessageBox Widget is used to display the message boxes in the python applications. This module is used to display a message using provides a number of functions.

The MySQL module in Python Connect to the database. Create an object for your database. Execute the SQL query. Fetch records from the result. Informing the Database if you make any changes in the table.

Pathlimb module in Python provides various classes representing file system paths with semantics appropriate for different operating systems. This module comes under Python's standard utility modules.

IMPLEMENTATION

Login/Register :-

```
class LR :
    def __init__(self) -> None:
        def encrypt(data) :
            return "".join(dict.fromkeys("".join([str(ord(i)) for i in data])))

        def login() :
            global LU
            cur.execute("select * from Userdb;")
            L  = {j:k for i,j,k in cur.fetchall()}
            cur.execute("select * from Userdb;")
            data  = [(i,j) for i,j,k in cur.fetchall()]
            uname = list(L.keys())
            psswd = list(L.values())
            email = loginemail.get()
            pwd = loginpsswd.get()
            if email in uname :
                print(psswd[uname.index(email)],encrypt(pwd),pwd)
                if psswd[uname.index(email)] == encrypt(pwd) :
                    print(data)
                    LU=data[uname.index(email)]
                    Index(self.window)

                else :
                    msg.showerror(message="Wrong Password")
            else :
                msg.showerror(message="User not found")

        def register() :
            cur.execute("select * from Userdb;")
            L  = {j:k for i,j,k in cur.fetchall()}
            uname = list(L.keys())
            psswd = list(L.values())
            name = regname.get()
            email = regemail.get()
```

```
            pwd = regpwd.get()
            if email not in uname :
                cur.execute("INSERT INTO Userdb
VALUES(%s,%s,%s)",(name,email,encrypt(pwd)))
                conn.commit()
                msg.showinfo(message="Register Successful")
            else :
                msg.showerror(message="User already exists!")


        self.window = Tk()
        self.window.title("Login | Register ~ Expressioue")
        self.window.geometry("1027x620")
        self.window.configure(bg = "#FFFFFF")


        canvas = Canvas(
            self.window,
            bg = "#FFFFFF",
            height = 620,
            width = 1027,
            bd = 0,
            highlightthickness = 0,
            relief = "ridge"
        )

        canvas.place(x = 0, y = 0)
        image_image_1 = PhotoImage(
            file=relative_to_assets(self.__class__,"image_1.png"))
        image_1 = canvas.create_image(
            513.0,
            310.0,
            image=image_image_1
        )

        image_image_2 = PhotoImage(
            file=relative_to_assets(self.__class__,"image_2.png"))
        image_2 = canvas.create_image(
            513.0,
            310.0,
            image=image_image_2
        )

        image_image_3 = PhotoImage(
            file=relative_to_assets(self.__class__,"image_3.png"))
```

```
        image_3 = canvas.create_image(
            215.0,
            341.0,
            image=image_image_3
        )

        image_image_4 = PhotoImage(
            file=relative_to_assets(self.__class__,"image_4.png"))
        image_4 = canvas.create_image(
            814.0,
            341.0,
            image=image_image_4
        )

        image_image_5 = PhotoImage(
            file=relative_to_assets(self.__class__,"image_5.png"))
        image_5 = canvas.create_image(
            513.0,
            84.0,
            image=image_image_5
        )

        loginbtn_image = PhotoImage(
            file=relative_to_assets(self.__class__,"loginbtn.png"))
        loginbtn = Button(
            image=loginbtn_image,
            borderwidth=0,
            highlightthickness=0,
            command=login,
            relief="flat"
        )
        loginbtn.place(
            x=69.0,
            y=440.3875427246094,
            width=292.91522216796875,
            height=51.612464904785156
        )

        signupbtn_image = PhotoImage(
            file=relative_to_assets(self.__class__,"signupbtn.png"))
        signupbtn = Button(
            image=signupbtn_image,
            borderwidth=0,
            highlightthickness=0,
            command=register,
```

```
        relief="flat"
    )
    signupbtn.place(
        x=668.0,
        y=440.3875427246094,
        width=292.91522216796875,
        height=51.612464904785156
    )

    entry_image_1 = PhotoImage(
        file=relative_to_assets(self.__class__,"entry_1.png"))
    entry_bg_1 = canvas.create_image(
        216.0,
        284.5,
        image=entry_image_1
    )
    loginemail = Entry(
        bd=0,
        bg="#9B7B9A",
        highlightthickness=0
    )
    loginemail.place(
        x=112.0,
        y=264.0,
        width=208.0,
        height=39.0
    )
    entry_image_4 = PhotoImage(
        file=relative_to_assets(self.__class__,"entry_4.png"))
    entry_bg_4 = canvas.create_image(
        216.0,
        372.5,
        image=entry_image_4
    )
    loginpsswd = Entry(
        bd=0,
        bg="#9B7B9A",
        highlightthickness=0,show="*"
    )
    loginpsswd.place(
        x=112.0,
        y=352.0,
        width=208.0,
        height=39.0
    )
```

```
        entry_image_2 = PhotoImage(
            file=relative_to_assets(self.__class__,"entry_2.png"))
        entry_bg_2 = canvas.create_image(
            815.0,
            259.0,
            image=entry_image_2
        )
        regname = Entry(
            bd=0,
            bg="#7F7F7F",
            highlightthickness=0
        )
        regname.place(
            x=709.0,
            y=241.0,
            width=212.0,
            height=34.0
        )

        entry_image_3 = PhotoImage(
            file=relative_to_assets(self.__class__,"entry_3.png"))
        entry_bg_3 = canvas.create_image(
            814.0,
            331.0,
            image=entry_image_3
        )
        regemail = Entry(
            bd=0,
            bg="#7F7F7F",
            highlightthickness=0
        )
        regemail.place(
            x=708.0,
            y=313.0,
            width=212.0,
            height=34.0
        )



        entry_image_5 = PhotoImage(
            file=relative_to_assets(self.__class__,"entry_5.png"))
        entry_bg_5 = canvas.create_image(
            814.0,
            403.0,
```

```
        image=entry_image_5
    )
    regpwd = Entry(
        bd=0,
        bg="#7F7F7F",
        highlightthickness=0,show="*"
    )
    regpwd.place(
        x=708.0,
        y=385.0,
        width=212.0,
        height=34.0
    )

    image_image_6 = PhotoImage(
        file=relative_to_assets(self.__class__,"image_6.png"))
    image_6 = canvas.create_image(
        139.0,
        246.0,
        image=image_image_6
    )

    image_image_7 = PhotoImage(
        file=relative_to_assets(self.__class__,"image_7.png"))
    image_7 = canvas.create_image(
        749.0,
        221.0,
        image=image_image_7
    )

    image_image_8 = PhotoImage(
        file=relative_to_assets(self.__class__,"image_8.png"))
    image_8 = canvas.create_image(
        742.0,
        295.0,
        image=image_image_8
    )

    image_image_9 = PhotoImage(
        file=relative_to_assets(self.__class__,"image_9.png"))
    image_9 = canvas.create_image(
        146.0,
        334.0,
        image=image_image_9
    )
```

```
        image_image_10 = PhotoImage(
            file=relative_to_assets(self.__class__,"image_10.png"))
        image_10 = canvas.create_image(
            749.0,
            366.0,
            image=image_image_10
        )

        image_image_11 = PhotoImage(
            file=relative_to_assets(self.__class__,"image_11.png"))
        image_11 = canvas.create_image(
            514.9999694824219,
            359.0,
            image=image_image_11
        )
        self.window.resizable(False, False)
        self.window.mainloop()
```

Landing Page :-

```
class Index :
    def __init__(self,parent) -> None:
        parent.withdraw()

        global data,N
        cur.execute("select * from msgdata")
        data = [(j,k) for i,j,k in cur.fetchall()]
        N = len(data)
        def reload() :
            global data
            global N
            conn.commit()
            cur.execute("select * from msgdata")
            data = [(j,k) for i,j,k in cur.fetchall()]
            print(data)
            N = len(data)
            lastmsg = data[-1]
            lastmsg_name_text = lastmsg[0].capitalize() + " says ..."
```

```
            messagebox.delete('1.0',END)
            messagebox.insert(END,lastmsg[1])
            mylabel.configure(text=lastmsg_name_text)
        def left() :
            try :
                global N
                info = data[N-2]
                print(data)
                messagebox.delete('1.0',END)
                messagebox.insert(END,info[1])
                mylabel.configure(text=info[0].capitalize() + " says ...")
                a = N
                N = a - 1
            except Exception as e:
                msg.showinfo("End","End of the content")
                print(data)
        def right() :
            try :
                global N
                info = data[N]
                print(N-1,info)
                messagebox.delete('1.0',END)
                messagebox.insert(END,info[1])
                mylabel.configure(text=info[0].capitalize() + " says ...")
                a = N
                N = a + 1
            except Exception as e:
                msg.showinfo("End","End of the content")
                print(data)
        def logout() :
            parent.deiconify()
            self.window.destroy()
        self.window = Toplevel(parent)
        self.window.title("{} ~ Expressioue".format(LU[1]))
        self.window.geometry("1027x620")
        self.window.configure(bg = "#FFFFFF")
        canvas = Canvas(
            self.window,
            bg = "#FFFFFF",
            height = 620,
            width = 1027,
            bd = 0,
            highlightthickness = 0,
            relief = "ridge"
        )
```

```
        canvas.place(x = 0, y = 0)
        image_image_1 = PhotoImage(
            file=relative_to_assets(self.__class__,"bglp.png"))
        image_1 = canvas.create_image(
            513.0,
            310.0,
            image=image_image_1
        )

        logout_button_image_1 = PhotoImage(
            file=relative_to_assets(self.__class__,"logoutbtn.png"))
        logoutbtn = Button(self.window,
            image=logout_button_image_1,
            borderwidth=0,
            highlightthickness=0,
            command=logout,
            relief="flat"
        )
        logoutbtn.place(
            x=0.0,
            y=396.0,
            width=61.0,
            height=172.0
        )

        post_button_image_2 = PhotoImage(
            file=relative_to_assets(self.__class__,"postbtn.png"))
        postbtn = Button(self.window,
            image=post_button_image_2,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: Post(parent),
            relief="flat"
        )
        postbtn.place(
            x=0.0,
            y=223.0,
            width=61.0,
            height=172.0
        )

        reload_button_image = PhotoImage(
            file=relative_to_assets(self.__class__,"reloadbtn.png"))
        reloadbtn = Button(self.window,
```

```
        image=reload_button_image,
        borderwidth=0,
        highlightthickness=0,
        command=reload,
        relief="flat"
    )
    reloadbtn.place(
        x=0.0,
        y=51.0,
        width=61.0,
        height=172.0
    )

    image_image_2 = PhotoImage(
        file=relative_to_assets(self.__class__,"image_2.png"))
    image_2 = canvas.create_image(
        538.0,
        348.0,
        image=image_image_2
    )

    image_image_3 = PhotoImage(
        file=relative_to_assets(self.__class__,"image_3.png"))
    image_3 = canvas.create_image(
        537.0,
        65.0,
        image=image_image_3
    )

    left_button_image_4 = PhotoImage(
        file=relative_to_assets(self.__class__,"prevbtn.png"))
    prevbtn = Button(self.window,
        image=left_button_image_4,
        borderwidth=0,
        highlightthickness=0,
        command=left,
        relief="flat"
    )
    prevbtn.place(
        x=125.0,
        y=223.114501953125,
        width=39.0,
        height=278.77099609375
    )
```

```
        right_button_image_5 = PhotoImage(
            file=relative_to_assets(self.__class__,"nextbtn.png"))
        nextbtn = Button(self.window,
            image=right_button_image_5,
            borderwidth=0,
            highlightthickness=0,
            command=right,
            relief="flat"
        )
        nextbtn.place(
            x=912.0,
            y=223.114501953125,
            width=39.0,
            height=278.77099609375
        )

        image_image_4 = PhotoImage(
            file=relative_to_assets(self.__class__,"image_4.png"))
        image_4 = canvas.create_image(
            537.0,
            244.0,
            image=image_image_4
        )



        labelarea = Canvas(canvas,bg="#244E6C",highlightthickness=0)
        labelarea.place(x=198.0,y=150.0,width=680.0,height=70.0)
        lastmsg = data[-1]
        lastmsg_name_text = lastmsg[0].capitalize() + " says ..."
        mylabel = Label(labelarea,text=lastmsg_name_text,font=("Lily Script One", 38 * -
1),foreground="#FFF",background="#244E6C",anchor=CENTER)
        mylabel.place(relx=0.0,y=0)
        messagebox = Text(canvas,height=12,width=80,font=("Comic Sans
MS",10),bg="#25445A",fg="#FFF",relief=FLAT,padx=20,pady=20)
        messagebox.place(x=197.0,y=278.0)
        messagebox.bind("<Key>", lambda a: "break")
        messagebox.insert(END,lastmsg[1])
        self.window.resizable(False, False)
        self.window.mainloop()
```

Posting :-

```
class Post :
    def __init__(self,parent) -> None:
        def post() :
            x = messagebox.get("1.0",END)
            cur.execute("INSERT INTO msgdata VALUES(%s,%s,%s)",(LU[1],LU[0],x))
            conn.commit()
            msg.showinfo("Success","Your post is online")
            self.window.destroy()
        self.window = Toplevel(parent)
        self.window.title("Post ~ Expressioue")
        self.window.geometry("583x526")
        self.window.configure(bg = "#FFFFFF")
        canvas = Canvas(
            self.window,
            bg = "#FFFFFF",
            height = 526,
            width = 583,
            bd = 0,
            highlightthickness = 0,
            relief = "ridge"
        )

        canvas.place(x = 0, y = 0)
        image_bg = PhotoImage(
            file=relative_to_assets(self.__class__,"bg.png"))
        bg = canvas.create_image(
            291.0,
            263.0,
            image=image_bg
        )

        image_frame = PhotoImage(
            file=relative_to_assets(self.__class__,"frame.png"))
        frame = canvas.create_image(
            292.0,
            300.0,
            image=image_frame
        )
```

```
        button_frame = PhotoImage(
            file=relative_to_assets(self.__class__,"button_1.png"))
        button_1 = Button(self.window,
            image=button_frame,
            borderwidth=0,
            highlightthickness=0,
            command=post,
            relief="flat"
        )
        button_1.place(
            x=209.0,
            y=423.0,
            width=166.0,
            height=39.0
        )

        image_container = PhotoImage(
            file=relative_to_assets(self.__class__,"container.png"))
        container = canvas.create_image(
            291.0,
            282.0,
            image=image_container
        )

        messagebox = Text(canvas,height=10,width=40,font=("Comic Sans
MS",10),bg="#484848",fg="#FFF",relief=FLAT,padx=20,pady=20)
        messagebox.place(x=130.0,y=170.0)


        image_logo = PhotoImage(
            file=relative_to_assets(self.__class__,"logo.png"))
        logo = canvas.create_image(
            292.0,
            60.0,
            image=image_logo
        )

        image_prompt = PhotoImage(
            file=relative_to_assets(self.__class__,"prompt.png"))

        prompt = canvas.create_image(
            108.0,
            182.0,
            image=image_prompt
```

```
        )
        self.window.resizable(False, False)
        self.window.mainloop()
```

Global Frame :-

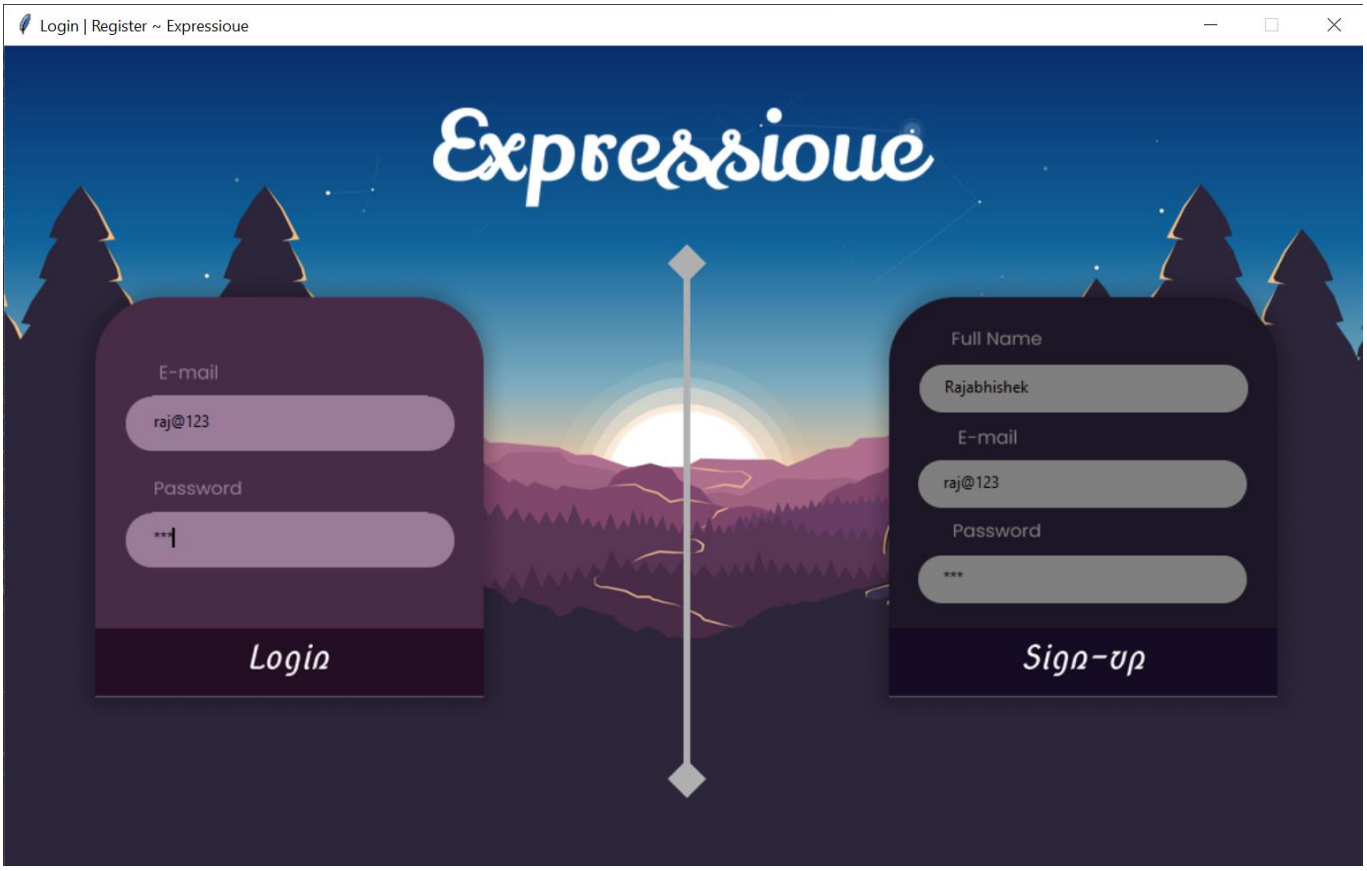
```
I = LR()
```

TESTING

We tested our program and the overall structure of the Expressioue by traversing each interface. We also tested the posting of message from different regions, internet and even the word limit (of 250 words). We also implemented the signup method by email and phone, we can login with either. The user can post any thing which will be broadcast to everyone. Also, the server was working perfectly and the post were online after posting them. Finally, we finished the integration of app with buttons using tkinter and also mysql was used for database management system.

RESULT

Followings are the screenshots of our project interfaces.



Login/Register Page

Landing Page

Posting Page

Landing Page After Refreshing

ANALYSIS

Everything in our project has been performed as planned and our team meets the goals outlined in the description. We designed an app “EXPRESSIOUE” based on talking community (like Tumbler) written in Python. We completed the development of our app within the time given and for demo purposes we encouraged our classmates, friends and even parents to have a look on it which gave everyone a lot of fun.

CONCLUSION

Our final project was planned, developed, and demonstrated as expected. We designed an app “EXPRESSIOUE” written in Python with the help of integrated modes using Tkinter and MySQL. A very user-friendly interface was implemented keeping in mind the varied audience of “EXPRESSIOUE” app user. We also implemented three different optional settings to keep the audience entertained along with display of beautiful backgrounds.

FUTURE ENHANCEMENTS

This is for sure that our app will get better with time with more upgrades in future.

- But for now we are looking to give a feature of sharing images, videos and links.
- We are also working to make a better and faster server.
- We are also looking forward to add the emoji's.

REFERENCES

- Python reference
- Tkinter tutorials
- Tkinter documents