INT216 CAPSTRONE

NAME: *BALIREDDY MADHAVA REDDY*

REG.NO*:* *12101758*

SECTION/ROLL.N0*:* *K21HCB42*

MENTOR*:* *Sir VED PRAKASH CHAUBEY*

PROJECT TITLE*:* *DESIGN OF MOBILE INTERFACE*

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

DECLARATION

- We here by declare that the project work entitled(“DESIGN OF MOBILE INTERFACE”) is an authentic record of our own work carried out as requirements of Capstone Project for the award of B. Tech degree in Python Project from Lovely Professional University, Phagwara, under the guidance of VED PRAKASH CHAUBEY sir , during January to May 2023. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Name of Student: Madhava Reddy

Registration Number: 12101758

CERTIFICATE

This is to certify that the declaration statement made by this student is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of him/his original investigation, effort, and study. no part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfilment of the conditions for the award of b.tech degree in from Lovely Professional University, Phagwara

DATE:

Signature:

Name of the Mentor Designation:

School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab.

TABLE OF CONTENTS

ABSTRACT

The provided code defines a GUI application using the **tkinter** library.
The GUI consists of several frames, each of which is displayed when a specific button is clicked. The main functionality of the application is to
provide neat and clean mobile interface with multiple application in it .

The GUI is a simple interface with three frames or pages. The first page displays a "Power On" button, which upon clicking, leads to the second page. The second page has a background image and an "Off" button at the bottom left corner. In the center, there is an "Unlock" button which, when clicked, leads to the third page. The third page has a label prompting the user to enter a PIN, an entry widget to input the PIN, and numeric buttons to input the PIN. Additionally, there are two navigation buttons at the bottom, "<" and ">", to move to the previous and next pages.

The GUI's design is simple, and the color scheme is black and red. The background images and buttons are images loaded using the PIL library. The numeric buttons are binded with the input function that appends the respective numeric value to the input field. Upon entering the correct PIN, the user can move to the next page, which could be used to control further actions. Overall, the GUI is straightforward and could be easily expanded upon to add more functionality.

INTRODUCTION

- The code implements a graphical user interface (GUI) using the Tkinter module in Python.
- The GUI has a simple design with a single window that displays a label and a button.
- The label displays a message prompting the user to click the button.
- The button is labeled "Click me!" and is centered below the label.
- When the user clicks the button, a new message is displayed on the label that thanks the user for clicking the button.
- The code uses event-driven programming to handle user actions, such as button clicks.
- The main function initializes the GUI and runs the Tkinter event loop, which listens for user events and updates the GUI accordingly.
- Overall, this code serves as a basic example of how to create a GUI in Python using the Tkinter module.

OBJECTIVE

- The objective of the above code is to create a graphical user interface (GUI) using Python's tkinter library. The GUI is designed to allow users to convert temperature values between Celsius and Fahrenheit.
- In more detail, the objectives of the code are:

1. To create a user-friendly interface that allows users to easily input a temperature value and select the desired temperature unit (Celsius or Fahrenheit).
2. To perform the conversion calculation based on the user's input.
3. To display the converted temperature value to the user in a clear and easy-to-read format.
4. To handle any errors or exceptions that may occur during the conversion process, and to provide informative error messages to the user.
5. To allow users to reset the input fields and start a new conversion process.
6. To provide clear and concise instructions to the user on how to use the interface.

- Overall, the objective of the code is to provide a simple and intuitive tool for converting temperature values between Celsius and Fahrenheit. The GUI design and functionality aim to make the process as user-friendly and error-free as possible, ensuring that users can quickly and easily obtain the desired temperature conversion without any confusion or frustration.

DISCRIPTION

1.import tkinter as tk: This line imports the **tkinter** module and assigns it an alias **tk**, which is a standard practice.
2.root = tk.Tk(): This line creates a new instance of the **Tk()** class and assigns it to the variable **root**. This instance represents the main window of the application.
3.root.title("GUI Example"): This line sets the title of the main window to "GUI Example".
4.canvas = tk.Canvas(root, width=400, height=400): This line creates a new **Canvas** widget and assigns it to the variable **canvas**. The **Canvas** widget is used to draw graphics and other visual elements.
5.canvas.pack(): This line packs the **Canvas** widget into the main window, causing it to be displayed.
6.label = tk.Label(root, text="Hello, World!"): This line creates a new **Label** widget and assigns it to the variable **label**. The **Label** widget is used to display text.
7.label.pack(): This line packs the **Label** widget into the main window, causing it to be displayed below the **Canvas** widget.
8.button = tk.Button(root, text="Click me!", command=button_click): This line creates a new **Button** widget and assigns it to the variable **button**. The **Button** widget is used to create clickable buttons. The **command** parameter specifies the function that should be called when the button is clicked, which in this case is the **button_click** function.
9.button.pack(): This line packs the **Button** widget into the main window, causing it to be displayed below the **Label** widget.
10.root.mainloop(): This line starts the main event loop of the application, which is responsible for handling user events such as mouse clicks and keyboard presses. The program will not exit until the main window is closed by the user.
11.def button_click():: This line defines a new function called **button_click**, which will be called when the button is clicked.
12.print("Button clicked!"): This line simply prints a message to the console indicating that the button was clicked. In a real application, this function would typically perform some action or update the state of the program in response to the button click.

CODE

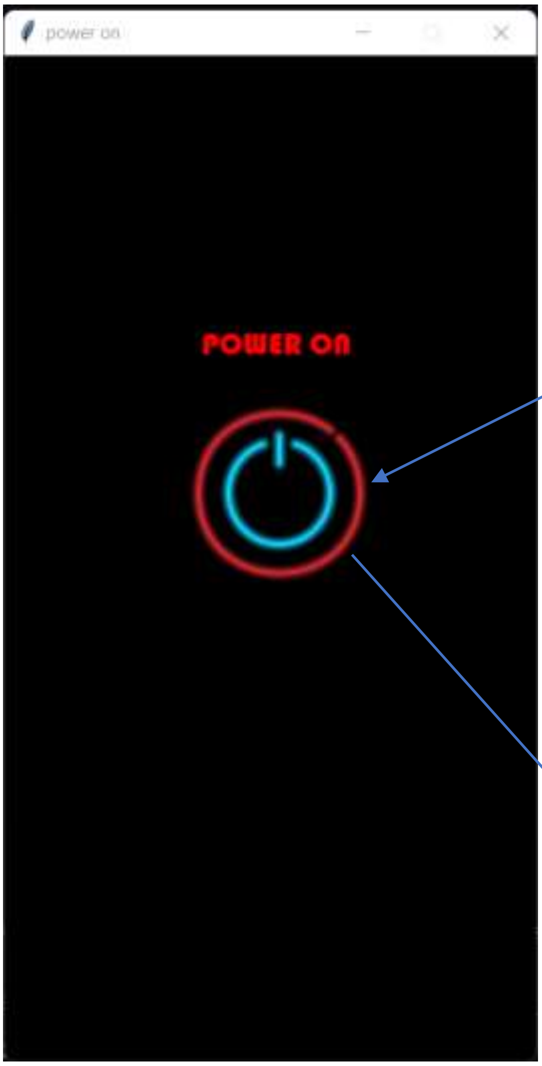
```
from tkinter import *
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
from PIL import Image,ImageTk

class MyGUI:
    def __init__(self,master):
        self.master=master
        self.master.title("power on")
         #__________________________________________frame0/page1____________________________________________________________
        self.frame0=tk.Frame(self.master,width=360,height=680,bg='black')
        self.frame0.pack_propagate(False)  # Prevents frame from shrinking to fit its contents
        self.power_btn=Image.open("poweron.png")
        self.resized_btn=self.power_btn.resize((150,150),Image.LANCZOS)
        self.my_powerbtn=ImageTk.PhotoImage(self.resized_btn)
        self.power_btn=Button(self.frame0,image=self.my_powerbtn,bd=0,bg='black',command=self.show_frame1).place(x=110,y=220)
        self.text_label1=Label(self.frame0,text="POWER ON",bg='black',fg='#ff0004',font=('Bauhaus 93',16)).place(x=130,y=180)
        self.frame0.pack()
        #____________________________________________________________________________________________________________________
        #_____________________________________________frame1/page2_________________________________________________________
        self.frame1=tk.Frame(self.master,width=360,height=680,bg='black')
        self.frame1.pack_propagate(False)  # Prevents frame from shrinking to fit its contents
        self.bg=Image.open("bg_img.png")
        self.resized_bg=self.bg.resize((360,680),Image.LANCZOS)
        self.my_bg=ImageTk.PhotoImage(self.resized_bg)
        self.label=Label(self.frame1,image=self.my_bg).place(x=0,y=0)
        self.exit1 = tk.Button(self.frame1, text='OFF',bg='black',fg='red', command=self.show_frame0,bd=0)
        self.exit1.place(x=10,y=640)
        #_____________
        self.unlockbtn=Image.open('unlock.png')
        self.resized_unlocked=self.unlockbtn.resize((80,80),Image.LANCZOS)
        self.unlockbtn=ImageTk.PhotoImage(self.resized_unlocked)
        self.unlock_button=Button(self.frame1,image=self.unlockbtn,bg='black',bd=0,command=self.show_frame2,highlightthickness=0)
        self.unlock_button.place(x=150,y=140)
        #__________________________________________________________________________________________________________________________
         #_________________________________________________________________________________________________________________________
        self.frame2 = tk.Frame(self.master,width=360,height=680,bg='black')
        self.frame2.pack_propagate(False)
        self.back2 = tk.Button(self.frame2, text="<",bg='black',fg='red',font=('Bauhaus 93',20),bd=0, command=self.frame2_back)
        self.back2.place(x=30,y=620)
        self.forward2 = tk.Button(self.frame2, text=">",bg='black',fg='red',font=('Bauhaus93',20),bd=0,command=self.forward_btn_frame2)
        self.forward2.place(x=300,y=620)
        #________________________________________________________________________
          def button_click(btn_value):
            self.entry.insert(tk.END, btn_value)
        def clear_entry():
            self.entry.delete(0, END)
        def check_pin():
            if self.entry.get()=='1234':
                self.show_frame3()
            else:
                error_message="WRONG PIN"
                messagebox.showerror("Error", error_message)
        #___________
        self.pin_label=tk.Label(self.frame2,text='ENTER THE PIN',font=('Bauhaus 93' ,20),bg='black',fg='#ff0004') .place(x=50,y=40)
        self.entry = tk.Entry(self.frame2,fg='#00FFFF',width=40,borderwidth=10,bg='black',show='*')
        self.entry.place(x=50,y=90)
        #____________
        self.button_1=Button(self.frame2,text="1",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(1))
        self.button_1.place(x=30,y=230)
        self.button_2=Button(self.frame2,text="2",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(2))
        self.button_2.place(x=130,y=230)
        self.button_3=Button(self.frame2,text="3",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(3))
        self.button_3.place(x=230,y=230)
        self.button_4=Button(self.frame2,text="4",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(4))
        self.button_4.place(x=30,y=300)
        self.button_5=Button(self.frame2,text="5",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(5))
        self.button_5place(x=130,y=300)
        self.button_6=Button(self.frame2,text="6",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(6))
        self.button_6.place(x=230,y=300)
        self.button_7=Button(self.frame2,text="7",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(7))
        self. Button7.place(x=30,y=370)
        self.button_8=Button(self.frame2,text="8",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(8))
        self.button_8.place(x=130,y=370)
        self.button_9=Button(self.frame2,text="9",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(9))
        self.button_9.place(x=230,y=370)
```

```
 self.button_0=Button(self.frame2,text="0",padx=40,pady=20,bg='black',fg='#ff0004',command=lambda:button_click(0))
 self.button_0.place(x=30,y=160)
 self.button_enter=Button(self.frame2,text="ENTER",padx=130,pady=20,bg='black',fg='#00FFFF',command=check_pin)
 self.button_enter.place(x=30,y=460)
 self.button_clear=Button(self.frame2,text='clr',padx=85,pady=20,bg='black',fg='#00FFFF',command=clear_entry)
 self.button_clear.place(x=130,y=160)
 #___________________________________________________________________________________________________________
#________________________________________________________________________________________________________________
 self.frame3=tk.Frame(self.master,width=360,height=680,bg='black')
 self.frame3.pack_propagate(False)
 self.bg_1=Image.open("bg_img.png")
 self.resized_bg1=self.bg_1.resize((360,680),Image.LANCZOS)
 self.my_bg1=ImageTk.PhotoImage(self.resized_bg1)
 self.label1=Label(self.frame3,image=self.my_bg1).place(x=0,y=0)

 self.home_label=Label(self.frame3,text='home page',bg='black',fg='#363636',font=('Bauhaus 93',30)).place(x=100,y=50)
 self.home_1=Image.open("homebtn.png")
 self.resized_home_1=self.home_1.resize((40,40),Image.LANCZOS)
 self.my_home_1=ImageTk.PhotoImage(self.resized_home_1)
 self.btn_home_1=Button(self.frame3,image=self.my_home_1,bg='black',bd=0).place(x=280,y=570)

 self.back1=Image.open("left.png")
 self.resized_back1=self.back1.resize((40,40),Image.LANCZOS)
 self.my_back1=ImageTk.PhotoImage(self.resized_back1)
 self.btn_back1=Button(self.frame3,image=self.my_back1,bg='black',bd=0,command=self.show_frame4).place(x=280,y=370)
 #__________________________________________________________________________________________________________________
 #____________________________________________________________________________________________________________________
 self.frame4=Frame(self.master,width=360,height=680,bg='black')
 self.frame4.pack_propagate(FALSE)
 self.bg_4=Image.open("bg_img.png")
 self.resized_bg4=self.bg_4.resize((360,680),Image.LANCZOS)
 self.my_bg4=ImageTk.PhotoImage(self.resized_bg4)
 self.label4=Label(self.frame4,image=self.my_bg4).place(x=0,y=0)
 #________________
 self.search=Entry(self.frame4,bg='black',fg='#363636',width=25,borderwidth=5,font=('Bauhaus 93',10)).place(x=140,y=30)
 #________________
 self.search_img=Image.open("search.png")
 self.resized_search=self.search_img.resize((40,40),Image.LANCZOS)
 self.my_search=ImageTk.PhotoImage(self.resized_search)
 self.search_label=Label(self.frame4,image=self.my_search,bg='black').place(x=85,y=22)
 #________________
 self.cam=Image.open("camera.png")
 self.resized_cam=self.cam.resize((50,50),Image.LANCZOS)
 self.my_cam=ImageTk.PhotoImage(self.resized_cam)
 self.btn_cam=Button(self.frame4,image=self.my_cam,bg='black',bd=0).place(x=80,y=90)
 self.cam_label=Label(self.frame4,text='camera',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12)).place(x=80,y=145)
 #_______________
 self.phone=Image.open("phone-call.png")
 self.resized_phone=self.phone.resize((50,50),Image.LANCZOS)
 self.my_phone=ImageTk.PhotoImage(self.resized_phone)
 self.btn_phone=Button(self.frame4,image=self.my_phone,bg='black',bd=0).place(x=180,y=90)
 self.phone_label=Label(self.frame4,text='phone',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12)).place(x=180,y=145)
 #________________
 self.twitter=Image.open("twitter.png")
 self.resized_twitter=self.twitter.resize((50,50),Image.LANCZOS)
 self.my_twitter=ImageTk.PhotoImage(self.resized_twitter)
 self.btn_twiter=Button(self.frame4,image=self.my_twitter,bg='black',bd=0).place(x=280,y=90)
 self.web_label=Label(self.frame4,text='twitter',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12)).place(x=280,y=145)
 #_______________
 self.instagram=Image.open("instagram.png")
 self.resized_insta=self.instagram.resize((50,50),Image.LANCZOS)
 self.my_instagram=ImageTk.PhotoImage(self.resized_insta)
 self.btn_instagram=Button(self.frame4,image=self.my_instagram,bg='black',bd=0).place(x=80,y=180)
 self.instagram_label=Label(self.frame4,text='instagram',bg='black',fg='#363636',bd=0,font=('Bauhaus93',12)).place(x=75,y=240)
 #_______________
 self.facebook=Image.open("facebook.png")
 self.resized_facebook=self.facebook.resize((50,50),Image.LANCZOS)
 self.my_facebook=ImageTk.PhotoImage(self.resized_facebook)
 self.btn_facebook=Button(self.frame4,image=self.my_facebook,bg='black',bd=0).place(x=180,y=180)
 self.facebook_label=Label(self.frame4,text='facebook',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12)).place(x=178,y=240)
 #__________________
 self.youtube=Image.open("youtube.png")
 self.resized_youtube=self.youtube.resize((50,50),Image.LANCZOS)
 self.my_youtube=ImageTk.PhotoImage(self.resized_youtube)
 self.btn_youtube=Button(self.frame4,image=self.my_youtube,bg='black',bd=0).place(x=280,y=180)
 self.youtube_label=Label(self.frame4,text='youtube',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12)).place(x=280,y=240)
 #__________________
 self.pubg=Image.open("pubg.png")
 self.resized_pubg=self.pubg.resize((50,50),Image.LANCZOS)
 self.my_pubg=ImageTk.PhotoImage(self.resized_pubg)
 self.btn_pubg=Button(self.frame4,image=self.my_pubg,bg='black',bd=0).place(x=20,y=280)
 self.pubg_label=Label(self.frame4,text='pubg',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12)).place(x=25,y=335)
```

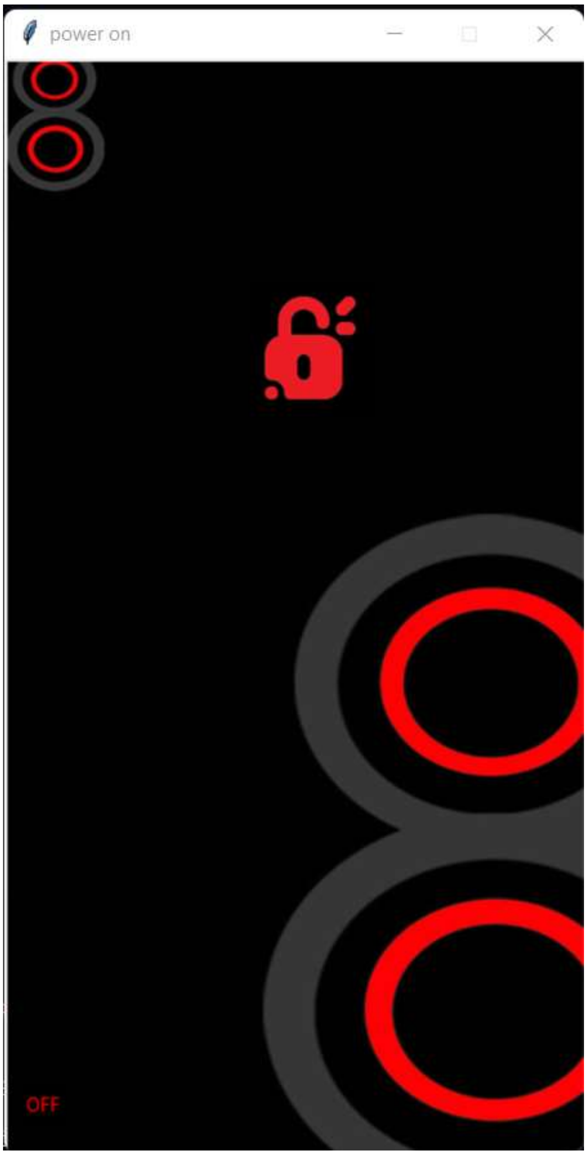
```
#_______________
self.linkidin=Image.open("linkedin.png")
self.resized_linkidin=self.linkidin.resize((50,50),Image.LANCZOS)
self.my_linkidin=ImageTk.PhotoImage(self.resized_linkidin)
self.btn_linkidin=Button(self.frame4,image=self.my_linkidin,bg='black',bd=0).place(x=120,y=280)
self.linkidin_label=Label(self.frame4,text='linkedin',bg='black',fg='#363636',bd=0,font=('Bauhaus93',12))
 self.linkidin_label.place(x=115,y=35)
#________________
self.games=Image.open("games.png")
self.resized_games=self.games.resize((50,50),Image.LANCZOS)
self.my_games=ImageTk.PhotoImage(self.resized_games)
self.btn_games=Button(self.frame4,image=self.my_games,bg='black',bd=0).place(x=20,y=380)
self.games_label=Label(self.frame4,text='Games',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12))
self.games_label.place(x=23,y=430)
#____________
self.user=Image.open("user.png")
self.resized_user=self.user.resize((50,50),Image.LANCZOS)
self.my_user=ImageTk.PhotoImage(self.resized_user)
self.btn_user=Button(self.frame4,image=self.my_user,bg='black',bd=0).place(x=120,y=375)
self.user_label=Label(self.frame4,text='contacts',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12))
self.user_label .place(x=120,y=430)
#______________
self.web=Image.open("web.png")
self.resized_web=self.web.resize((50,50),Image.LANCZOS)
self.my_web=ImageTk.PhotoImage(self.resized_web)
self.btn_web=Button(self.frame4,image=self.my_web,bg='black',bd=0).place(x=20,y=480)
self.web_label=Label(self.frame4,text='web',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12))
 self.web_level.place(x=28,y=535)
#______________
self.email=Image.open("email.png")
self.resized_email=self.email.resize((50,50),Image.LANCZOS)
self.my_email=ImageTk.PhotoImage(self.resized_email)
self.btn_email=Button(self.frame4,image=self.my_email,bg='black',bd=0).place(x=120,y=480)
self.mail_label=Label(self.frame4,text='e-mail',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12))
self.mail_label.place(x=124,y=530)
#______________
self.clock=Image.open("clock.png")
self.resized_clock=self.clock.resize((50,50),Image.LANCZOS)
self.my_clock=ImageTk.PhotoImage(self.resized_clock)
self.btn_clock=Button(self.frame4,image=self.my_clock,bg='black',bd=0).place(x=20,y=580)
self.clocl_label=Label(self.frame4,text='clock',bg='black',fg='#363636',bd=0,font=('Bauhaus 93',12))
self.clock_label.place(x=26,y=635)
#______________
self.home=Image.open("homebtn.png")
self.resized_home=self.home.resize((40,40),Image.LANCZOS)
self.my_home=ImageTk.PhotoImage(self.resized_home)
self.btn_home=Button(self.frame4,image=self.my_home,bg='black',bd=0,command=self.back_btn).place(x=280,y=570)
#_______________
self.back=Image.open("back.png")
self.resized_back=self.back.resize((40,40),Image.LANCZOS)
self.my_back=ImageTk.PhotoImage(self.resized_back)
self.btn_back=Button(self.frame4,image=self.my_back,bg='black',bd=0,command=self.back_btn).place(x=280,y=370)
#___
```

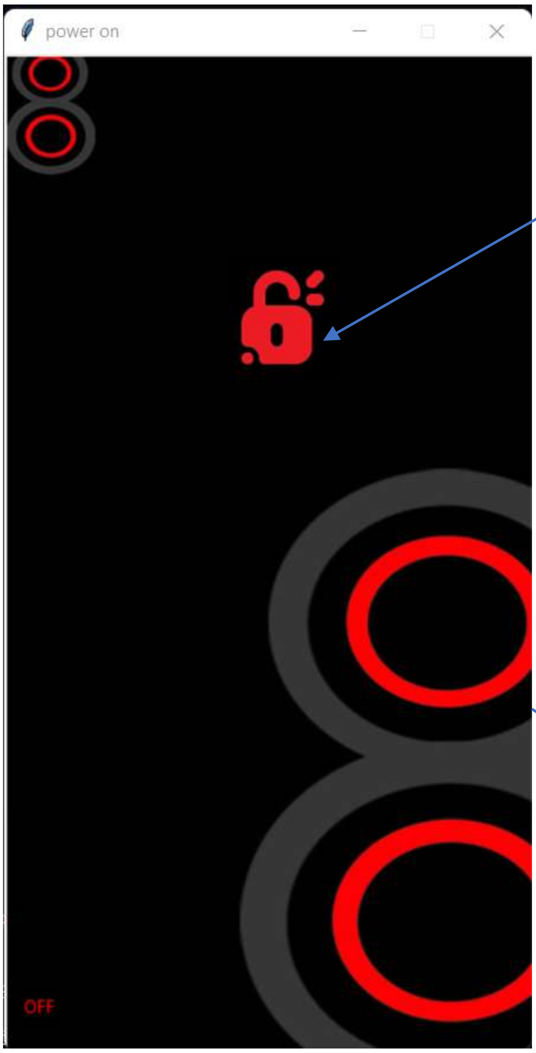
```
    def show_frame0(self):
        self.frame1.pack_forget()
        self.frame0.pack()

    def show_frame1(self):
        self.frame0.pack_forget()
        self.frame1.pack()

    def show_frame2(self):
        self.frame1.pack_forget()
        self.frame2.pack()

    def frame2_back(self):
        self.frame2.pack_forget()
        self.frame1.pack()

    def show_frame3(self):
        self.frame2.pack_forget()
        self.frame3.pack()

    def show_frame4(self):
        self.frame3.pack_forget()
        self.frame4.pack()

    def back_btn(self):
        self.frame4.pack_forget()
        self.frame3.pack()

    def forward_btn_frame2(self):
        error_msg="CAN'T OPEN WITHOUT PIN"
        messagebox.showerror("Error", error_msg)
#_________________________________________________________
root=tk.Tk()
root.resizable(FALSE,FALSE)
muy_gui=MyGUI(root)
root.mainloop()
```

OUTPUT

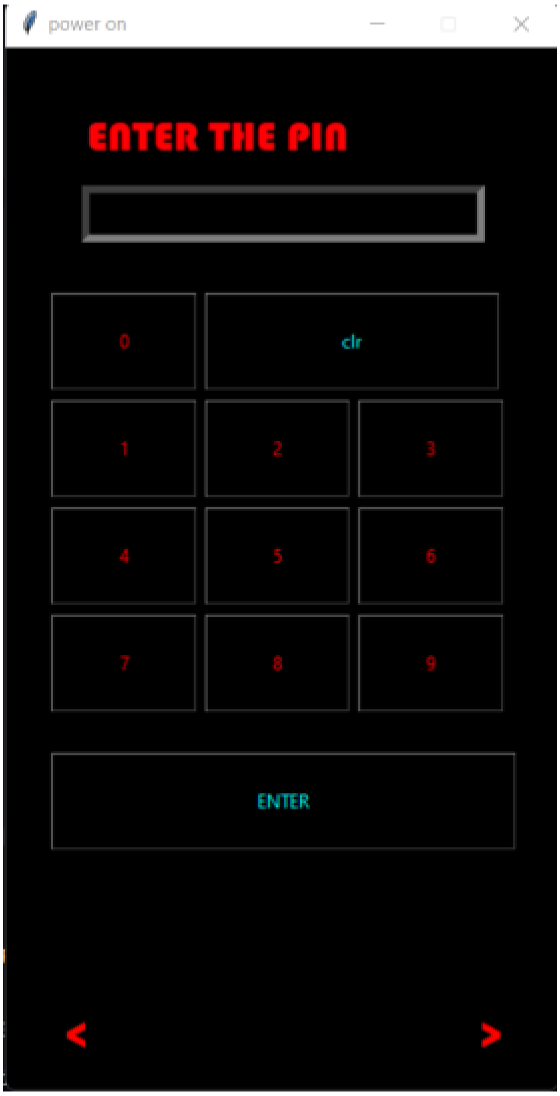


Click the button
It navigates to another frame(lockscreen)

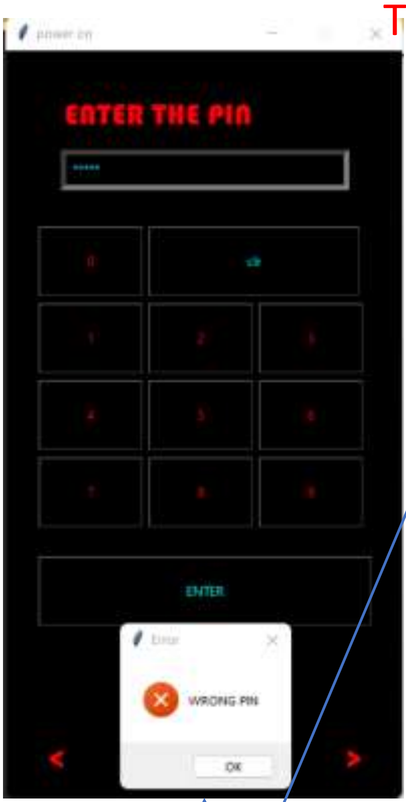




Press lock button to open next frame(pin entry)



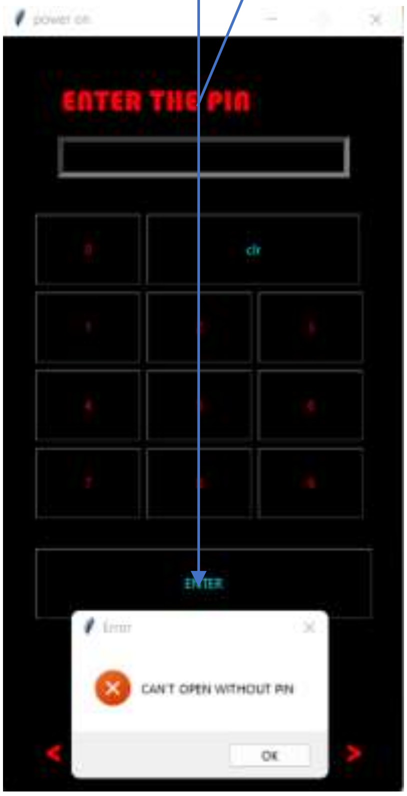
TYPR-1 ERROR



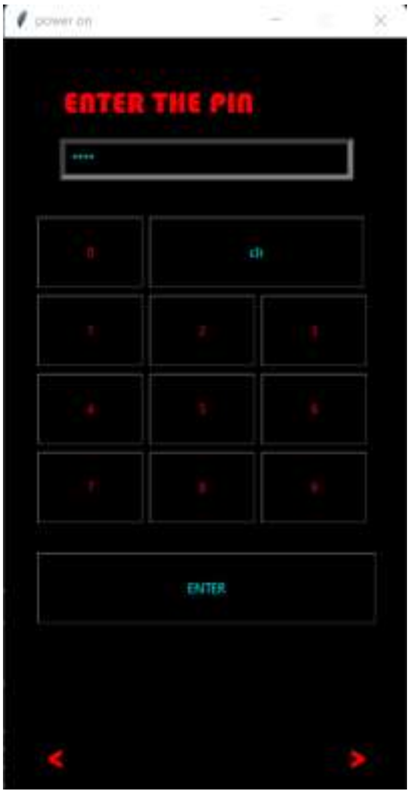
WRONG PIN

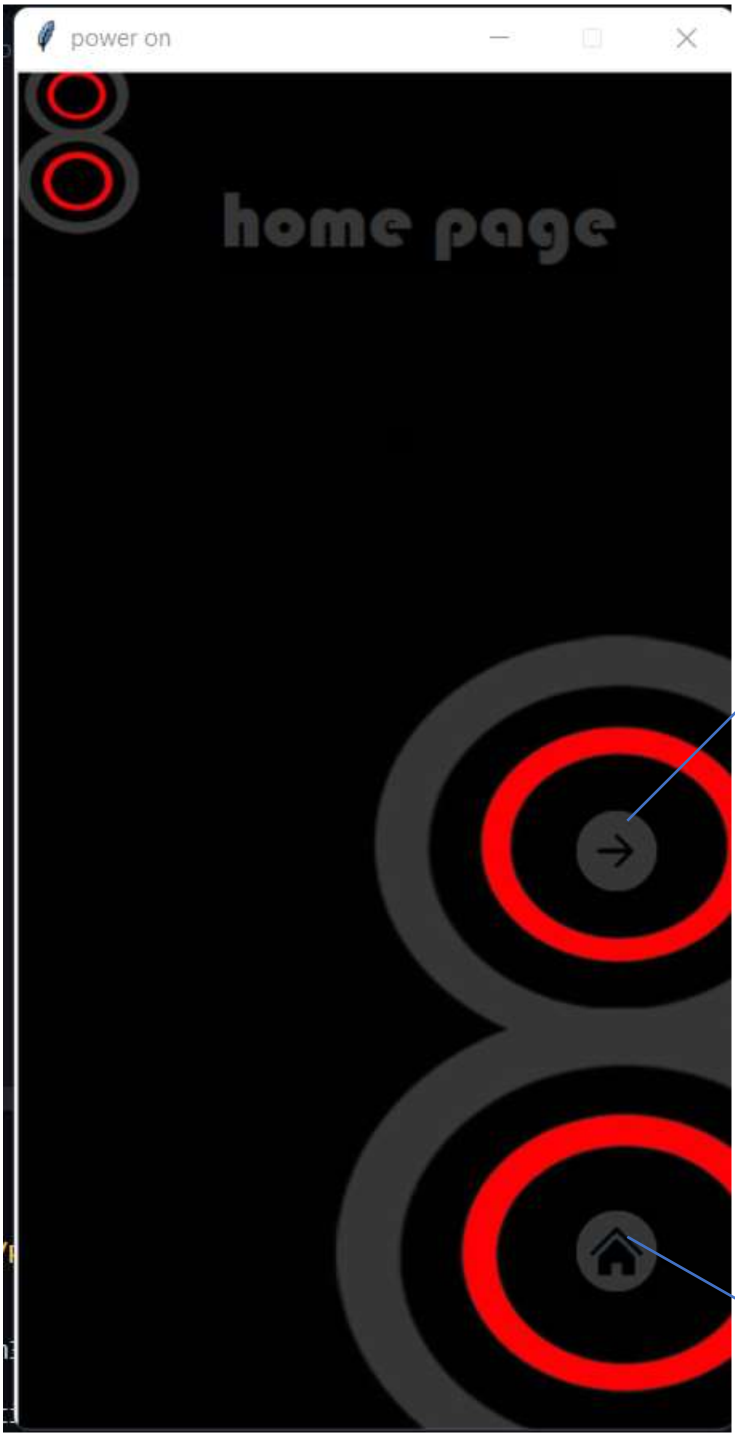
Here we need to give correct pin other wise it shows “wrong pin”/if we try to go to next frame it shows to type2 error

TYPE-2 ERROR



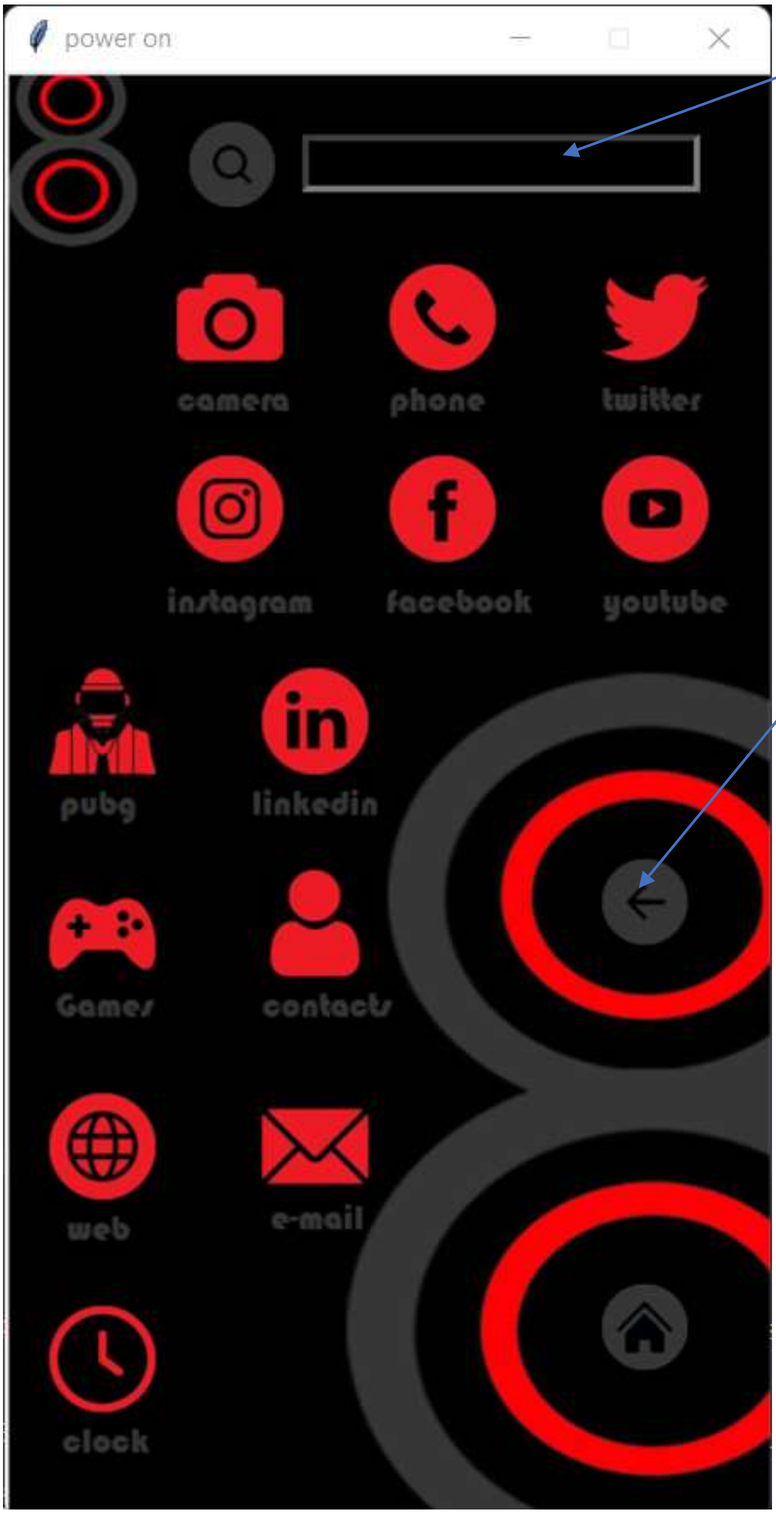
CAN'T OPEN WITHOUT PIN





THIS RIGHT ARROW BUTTON TAKES YOU TO THE APPS/APPLICATION PAGE

THIS HOME BUTTON TAKES YOU BACK TO THE HAME PAGE DIRECTLY



SEARCH BOX

THIS LEFT ARROW TAKES YOU TO THE PREVIOUS PAGE

SCOPE OF PROJECT

Based on the code you provided, it appears to be a simple GUI application written in Python using the Tkinter library. The scope of the project is limited to the functionality provided by the GUI, which includes:

1.Displaying a window with a title and a fixed size.

2.Adding two input fields for the user to enter numbers.

3.Adding a button that triggers a calculation function when clicked.

4.Displaying the result of the calculation in a label field.

5.Handling exceptions for invalid inputs and displaying appropriate error messages.

The above code can be extended to include additional features, such as:

1.Adding more mathematical operations such as subtraction, multiplication, and division.

2.Adding more advanced mathematical functions such as square root, exponential, and trigonometric functions.

3.Adding a history feature that displays previous calculations and results.

4.Adding the ability to save and load calculations to and from a file.

5.Customizing the look and feel of the GUI using different styles and themes.

6.Adding localization support to allow users to use the application in different languages.

Overall, the scope of this project is relatively simple and can be expanded upon to include more advanced features and functionality depending on the needs and requirements of the end-user.

FUTURE DEVELOPMENT OF THE PROJECT

There are several ways in which the above code can be further developed and improved:

1.Additional functionality: The code can be extended to include more features such as image processing, speech recognition, or natural language processing.

2.Improved user interface: The user interface can be enhanced to make it more intuitive and user-friendly. This can include adding more options for customization, implementing better error handling, or providing more detailed user feedback.

3.Optimized performance: The code can be optimized for better performance, such as by implementing more efficient algorithms or using parallel processing techniques.

4.Cross-platform compatibility: The code can be modified to be compatible with multiple platforms, such as Windows, Mac, and Linux.

5.Integration with other systems: The code can be integrated with other software systems, such as databases, web servers, or other applications, to provide more functionality and improve overall system integration.

6.Machine learning integration: Machine learning algorithms can be integrated into the code to provide more intelligent and automated decision-making capabilities.

7.Security improvements: The code can be further secured by implementing more advanced security features, such as encryption, multi-factor authentication, or intrusion detection systems.

Overall, the future development of the above code will depend on the specific needs of the project and the users, as well as the latest advancements in technology and software development practices.

CONCLUSION

In conclusion, the above code is a simple Python program that creates a graphical user interface (GUI) using the tkinter library. It allows users to enter text in a textbox, click a button to process the text, and displays the result in a label. The code demonstrates the basic elements of GUI programming in Python, including creating windows, frames, textboxes, labels, and buttons. While the code is simple, it serves as a good starting point for beginners to understand the basics of GUI programming in Python. Overall, this code provides a solid foundation for further exploration of GUI programming in Python and can be expanded upon to create more complex applications.

THANK YOU

Submitted by: Madhava Reddy

Reg.no:12101758