“GUI Calculator Using Tkinter”

A Project Submitted
In Partial Fulfillment of the Requirements
For the Degree of

Bachelor in Technology

In

Computer Science and Engineering

Submitted By:

KUNAL

Department of Computer Science and Engineering

University of Petroleum and Energy Studies

BIDHOLI, via, Prem Nagar, Dehradun, Uttarakhand - 248007

October, 2021

UNDERTAKING

I declare that the work presented in this project titled "*GUI Calculator Using Tkinter*", submitted to the Department of computer Science and Engineering, University of Petroleum and Energy Studies, BIDHOLI, via Prem Nagar, Dehradun, Uttarakhand for the award of the ***Bachelor of Technology*** degree in ***Computer Science and engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

October, 2021
Bhagalpur, Bihar

_____

Kunal

Acknowledgements

My first foremost thanks go to my Trainer Jayesh Gautam Sir (Project Trainer) for his guidance, help and constant encouragement throughout this project.

I am thankful to those who have always been with me generously in bringing this report on "*GUI Calculator Using Tkinter*" from the outset of the project.

I am grateful to their inspiration and valuable suggestions from time to time and providing all the facilities during project.

I would like to thanks also my friends for their support and cooperation time to time.

Above all, I am thankful to all mighty god and my parents for their at most moral support and blessings without which the project would not have been success.

Kunal

INDEX

CHAPTER 1-INTRODUCTION

PYTHON

Python Language Introduction

- Python may be a wide used general, high level programing language. It absolutely was at first designed by Guido van Rossum in 1991 and developed by Python computer code Foundation. It absolutely was chiefly developed for stress on code readability, and its syntax permits programmers to precise ideas in fewer lines of code.
- Python may be a programing language that allows you to work quickly and integrate systems a lot of expeditiously.
- Python may be a high-level, taken, interactive and object-oriented scripting language. Python is meant to be extremely decipherable. It uses English keywords often wherever as different languages use punctuation, and its fewer grammar constructions than different languages.
- Python is taken Python is processed at runtime by the interpreter. You are doing not have to be compelled to compile your program before execution it. This can be like PERL and PHP.
- Python is Interactive − you'll really sit at a Python prompt and act with the interpreter on to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented vogue or technique of programming that encapsulates code inside objects.
- Python may be a Beginner's Language − Python may be a nice language for the beginner-level programmers and supports the event of a large vary of applications from easy text process to web browsers to games.

PYTHON TKINTER GUI

Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-orientedinterface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table −

Sr.No.	Operator & Description
1	**Button** The Button widget is used to display buttons in your application.
2	**Canvas** The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	**Checkbutton** The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.

4	**Entry** The Entry widget is used to display a single-line text field for accepting values from a user.
5	**Frame** The Frame widget is used as a container widget to organize other widgets.
6	**Label** The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
7	**Listbox** The Listbox widget is used to provide a list of options to a user.
8	**Menubutton** The Menubutton widget is used to display menus in your application.
9	**Menu** The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	**Message** The Message widget is used to display multiline text fields for accepting values from a user.
11	**Radiobutton** The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
12	**Scale** The Scale widget is used to provide a slider widget.
13	**Scrollbar** The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
14	**Text** The Text widget is used to display text in multiple lines.

15	**Toplevel** The Toplevel widget is used to provide a separate window container.
16	**Spinbox** The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.
17	**PanedWindow** A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
18	**LabelFrame** A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
19	**tkMessageBox** This module is used to display message boxes in your applications.

Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

- The *pack()* Method − This geometry manager organizes widgets in blocks before placing them in the parent widget.
- The *grid()* Method − This geometry manager organizes widgets in a table-like structure in the parent widget.
- The *place()* Method − This geometry manager organizes widgets by placing them in a specific position in the parent widget.

BASIC FUNCTIONS

Addition

The addition (sum function) is used by clicking on the "+" button or using the keyboard. The function results in (a+b).

Subtraction

The subtraction (minus function) is used by clicking on the "-" button or using the keyboard. The function results in (a-b).

Multiplication

The multiplication (times function) is used by clicking on the "x" button or using the keyboard"*" key. The function results in a*b.

Division

The division (divide function) is used by clicking on the "/" button or using the keyboard "/"key. The function results in (a/b).

Sign

The sign key (negative key) is used by clicking on the "(-)" button. The function results in -1*x.

Square Root

The square root function is used by clicking "sqrt()". This function represents x^0.5 where the result squared is equal to x.

Logarithm

The logarithm (LOG) is used by clicking on the "log" button or type "log(".

Exponent

Numbers with exponents of 10 are displayed with an "e", for example 4.5e+100 or 4.5e-100.This function represents 10^x. Numbers are automatically displayed in the format when the number is too large or too small for the display. To enter a number in this format use the exponent key "exp". To do this enter the mantissa (the non-exponent part) then press "exp" and then enter the exponent.

Factorial

The Factorial function is used by clicking the "factorial" button or type "factorial".

CHAPTER 2 - IMPLEMENTATION

Technologies used - Python 3.9.7

- Python Tkinter GUI

Language used - Python

CODE OF PROJECT

```
from tkinter import *
from math import *


class calculate():

   def _init_(self):
      self.root = Tk()
      self.root.title("Calculator")
      self.root.geometry("500x610")
      self.root.config(bg="#607B8B")

      self.root.maxsize(500, 610)
      self.root.minsize(500, 610)

      self.resultwindow = Entry(self.root, borderwidth=15, width=41, relief=SUNKEN)
      self.resultwindow.grid(row=0, column=0, columnspan=10, pady=10)
      self.resultwindow.config(font=("Arial", 18))
      self.resultwindow.focus_set()

      # Buttons
      self.buttonlog10 = Button(self.root, text="log10", width=8, height=3,
command=lambda: self.btn('log10('),
                          relief=RAISED,
                          bg='light green')
      self.buttonlog10.grid(row=1, column=0, padx=5, pady=5)
      self.buttonlog10.config(font=("Arial", 18))

      self.buttonexp = Button(self.root, text="exp", width=8, height=3,
command=lambda: self.btn('exp('),
                        relief=RAISED,
                        bg='light green')
      self.buttonexp.grid(row=1, column=1, padx=3, pady=3)
      self.buttonexp.config(font=("Arial", 18))
```

```
    self.buttonsin = Button(self.root, text="sin", width=8, height=3,
command=lambda: self.btn('sin('),
                        relief=RAISED,
                        bg='light green')
    self.buttonsin.grid(row=1, column=2, padx=3, pady=3)
    self.buttonsin.config(font=("Arial", 18))

    self.buttoncos = Button(self.root, text="cos", width=8, height=3,
command=lambda: self.btn('cos('),
                        relief=RAISED,
                        bg='#FFEE58')
    self.buttoncos.grid(row=1, column=3, padx=3, pady=3)
    self.buttoncos.config(font=("Arial", 18))

    self.buttontan = Button(self.root, text="tan", width=8, height=3,
command=lambda: self.btn('tan('),
                        relief=RAISED,
                        bg='#FFEE58')
    self.buttontan.grid(row=1, column=4, padx=3, pady=3)
    self.buttontan.config(font=("Arial", 18))

    self.buttonlog2 = Button(self.root, text="log2", width=8, height=3,
command=lambda: self.btn('log2('),
                         relief=RAISED,
                         bg='light green')
    self.buttonlog2.grid(row=2, column=0, padx=5, pady=5)
    self.buttonlog2.config(font=("Arial", 18))

    self.buttonradians = Button(self.root, text="radians", width=8, height=3,
command=lambda: self.btn('radians('),
                           relief=RAISED,
                           bg='light green')
    self.buttonradians.grid(row=2, column=1, padx=3, pady=3)
    self.buttonradians.config(font=("Arial", 18))

    self.buttonasin = Button(self.root, text="asin", width=8, height=3,
command=lambda: self.btn('asin('),
                         relief=RAISED,
                         bg='light green')
    self.buttonasin.grid(row=2, column=2, padx=3, pady=3)
    self.buttonasin.config(font=("Arial", 18))

    self.buttonacos = Button(self.root, text="acos", width=8, height=3,
```

```
command=lambda: self.btn('acos('),
                        relief=RAISED,
                        bg='#FFEE58')
      self.buttonacos.grid(row=2, column=3, padx=3, pady=3)
      self.buttonacos.config(font=("Arial", 18))

      self.buttonatan = Button(self.root, text="atan", width=8, height=3,
command=lambda: self.btn('atan('),
                        relief=RAISED,
                        bg='#FFEE58')
      self.buttonatan.grid(row=2, column=4, padx=3, pady=3)
      self.buttonatan.config(font=("Arial", 18))

      self.buttonfactorial = Button(self.root, text="factorial", width=8, height=3,
                            command=lambda: self.btn('factorial('),
                            relief=RAISED,
                            bg='light green')
      self.buttonfactorial.grid(row=3, column=0, padx=5, pady=5)
      self.buttonfactorial.config(font=("Arial", 18))

      self.buttonpow = Button(self.root, text="pow", width=8, height=3,
command=lambda: self.btn('pow('),
                        relief=RAISED,
                        bg='light green')
      self.buttonpow.grid(row=3, column=1, padx=3, pady=3)
      self.buttonpow.config(font=("Arial", 18))

      self.buttonsqrt = Button(self.root, text="sqrt", width=8, height=3,
command=lambda: self.btn('sqrt('),
                        relief=RAISED,
                        bg='light green')
      self.buttonsqrt.grid(row=3, column=2, padx=3, pady=3)
      self.buttonsqrt.config(font=("Arial", 18))

      self.buttondiv = Button(self.root, text="/", width=8, height=3,
command=lambda: self.btn('/'),
                        relief=RAISED,
                        bg='#FFEE58')
      self.buttondiv.grid(row=3, column=3, padx=3, pady=3)
      self.buttondiv.config(font=("Arial", 18))

      self.buttonmod = Button(self.root, text="%", width=8, height=3,
command=lambda: self.btn('%'),
```

```
                    relief=RAISED,
                    bg='#FFEE58')
        self.buttonmod.grid(row=3, column=4, padx=3, pady=3)
        self.buttonmod.config(font=("Arial", 18))

        self.button1 = Button(self.root, text="1", width=8, height=3, command=lambda:
self.btn('1'), relief=RAISED,
                    bg='light green')
        self.button1.grid(row=4, column=0, padx=5, pady=5)
        self.button1.config(font=("Arial", 18))

        self.button2 = Button(self.root, text="2", width=8, height=3, command=lambda:
self.btn('2'), relief=RAISED,
                    bg='light green')
        self.button2.grid(row=4, column=1, padx=3, pady=3)
        self.button2.config(font=("Arial", 18))

        self.button3 = Button(self.root, text="3", width=8, height=3, command=lambda:
self.btn('3'), relief=RAISED,
                    bg='light green')
        self.button3.grid(row=4, column=2, padx=3, pady=3)
        self.button3.config(font=("Arial", 18))

        self.button4 = Button(self.root, text="4", width=8, height=3, command=lambda:
self.btn('4'), relief=RAISED,
                    bg='light green')
        self.button4.grid(row=5, column=0, padx=3, pady=3)
        self.button4.config(font=("Arial", 18))

        self.button5 = Button(self.root, text="5", width=8, height=3, command=lambda:
self.btn('5'), relief=RAISED,
                    bg='light green')
        self.button5.grid(row=5, column=1, padx=3, pady=3)
        self.button5.config(font=("Arial", 18))

        self.button6 = Button(self.root, text="6", width=8, height=3, command=lambda:
self.btn('6'), relief=RAISED,
                    bg='light green')
        self.button6.grid(row=5, column=2, padx=3, pady=3)
        self.button6.config(font=("Arial", 18))

        self.button7 = Button(self.root, text="7", width=8, height=3, command=lambda:
self.btn('7'), relief=RAISED,
                    bg='light green')
```

```
        self.button7.grid(row=6, column=0, padx=3, pady=3)
        self.button7.config(font=("Arial", 18))

        self.button8 = Button(self.root, text="8", width=8, height=3, command=lambda:
self.btn('8'), relief=RAISED,
                              bg='light green')
        self.button8.grid(row=6, column=1, padx=3, pady=3)
        self.button8.config(font=("Arial", 18))

        self.button9 = Button(self.root, text="9", width=8, height=3, command=lambda:
self.btn('9'), relief=RAISED,
                              bg='light green')
        self.button9.grid(row=6, column=2, padx=3, pady=3)
        self.button9.config(font=("Arial", 18))

        self.button0 = Button(self.root, text="0", width=8, height=3, command=lambda:
self.btn('0'), relief=RAISED,
                              bg='light green')
        self.button0.grid(row=7, column=0, padx=3, pady=3)
        self.button0.config(font=("Arial", 18))

        self.button_open = Button(self.root, text="(", width=8, height=3,
command=lambda: self.btn('('), relief=RAISED)
        self.button_open.grid(row=7, column=1, padx=3, pady=3)
        self.button_open.config(font=("Arial", 18))

        self.button_close = Button(self.root, text=")", width=8, height=3,
command=lambda: self.btn(')'), relief=RAISED)
        self.button_close.grid(row=7, column=2, padx=3, pady=3)
        self.button_close.config(font=("Arial", 18))

        # Operations Buttons

        self.buttonplus = Button(self.root, text="+", width=8, height=3,
command=lambda: self.btn('+'), relief=RAISED)
        self.buttonplus.grid(row=4, column=3, padx=3, pady=3)
        self.buttonplus.config(font=("Arial", 18))

        self.buttonminus = Button(self.root, text="-", width=8, height=3,
command=lambda: self.btn('-'), relief=RAISED)
        self.buttonminus.grid(row=4, column=4, padx=3, pady=3)
        self.buttonminus.config(font=("Arial", 18))

        self.buttondivide = Button(self.root, text="•", width=8, height=3,
```

```
command=lambda: self.btn('.'), relief=RAISED)
        self.buttondivide.grid(row=5, column=3, padx=3, pady=3)
        self.buttondivide.config(font=("Arial", 18))

        self.buttonmultiply = Button(self.root, text="", width=8, height=3,
command=lambda: self.btn(''),
                                     relief=RAISED)
        self.buttonmultiply.grid(row=5, column=4, padx=3, pady=3)
        self.buttonmultiply.config(font=("Arial", 18))

        self.buttoncancel = Button(self.root, text="AC", width=8, height=3,
command=lambda: self.cancel(),
                                   relief=RAISED,
                                   bg='#EF5350', fg='black')
        self.buttoncancel.grid(row=6, column=4, padx=3, pady=3)
        self.buttoncancel.config(font=("Arial", 18))

        self.buttondeleteall = Button(self.root, text="DEL", width=8, height=3,
command=lambda: self.delete_all(),
                                      relief=RAISED)
        self.buttondeleteall.grid(row=6, column=3, padx=3, pady=3)
        self.buttondeleteall.config(font=("Arial", 18))

        self.buttonresult = Button(self.root, text="=", width=17, height=3,
command=lambda: self.calculate(),
                                   relief=RAISED,
                                   bg='#FFEE58')
        self.buttonresult.grid(row=7, column=3, padx=3, pady=3, columnspan=2)
        self.buttonresult.config(font=("Arial", 18))

        self.root.mainloop()

    def btn(self, val):
        self.resultwindow.insert(END, val)

    def cancel(self):
        self.resultwindow.delete(0, 'end')

    def delete_all(self):
        x = self.resultwindow.get()
        self.resultwindow.delete(0, 'end')
        y = x[:-1]
        self.resultwindow.insert(0, y)
```

```
    def calculate(self):
        x = self.resultwindow.get()
        answer = eval(x)
        self.resultwindow.delete(0, 'end')
        self.resultwindow.insert(0, answer)


if _name_ == "_main_":
 calculate()
```

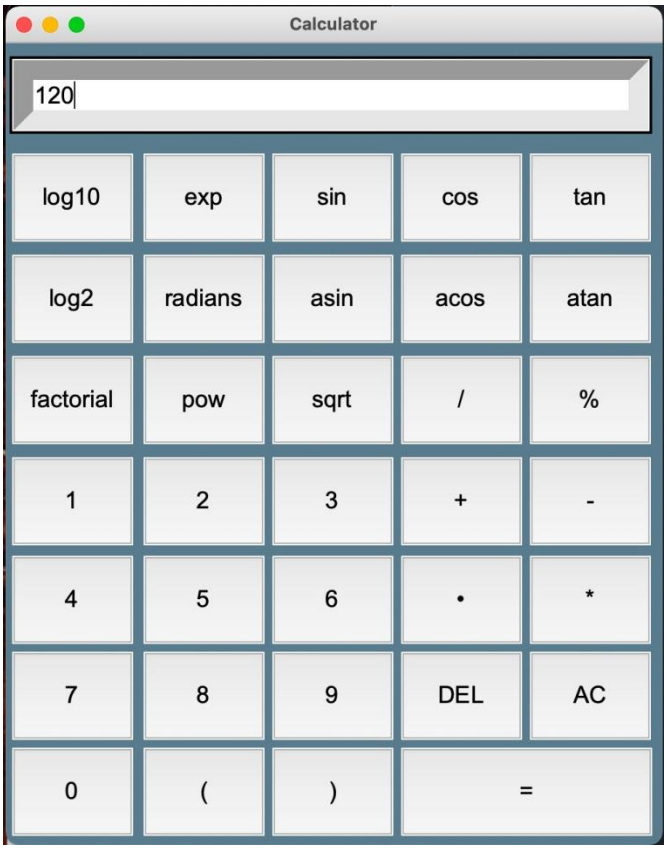
CHAPTER 3 - SCREENSHOTS

GUI – Main display window with name of The Calculator

Calculate some Operations and Answer

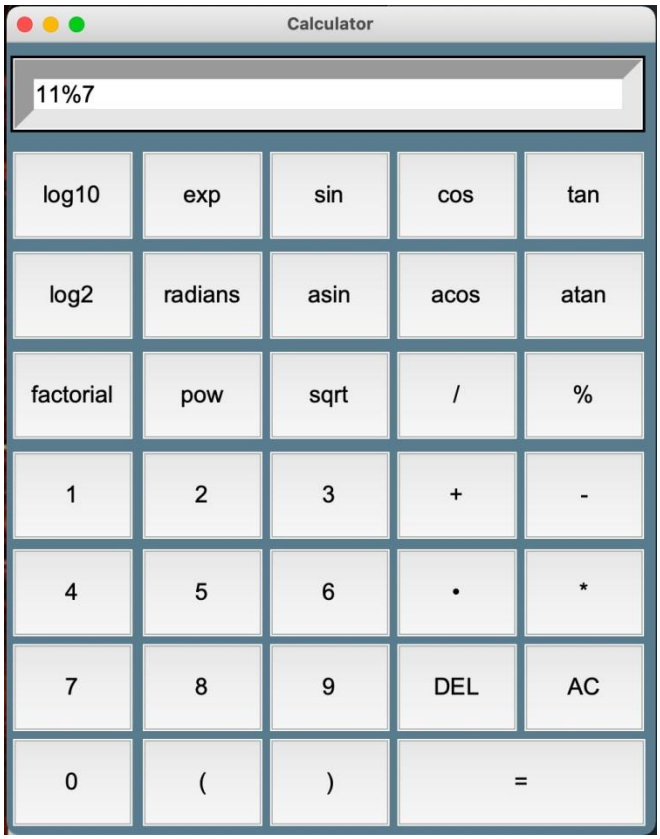
Log Function:

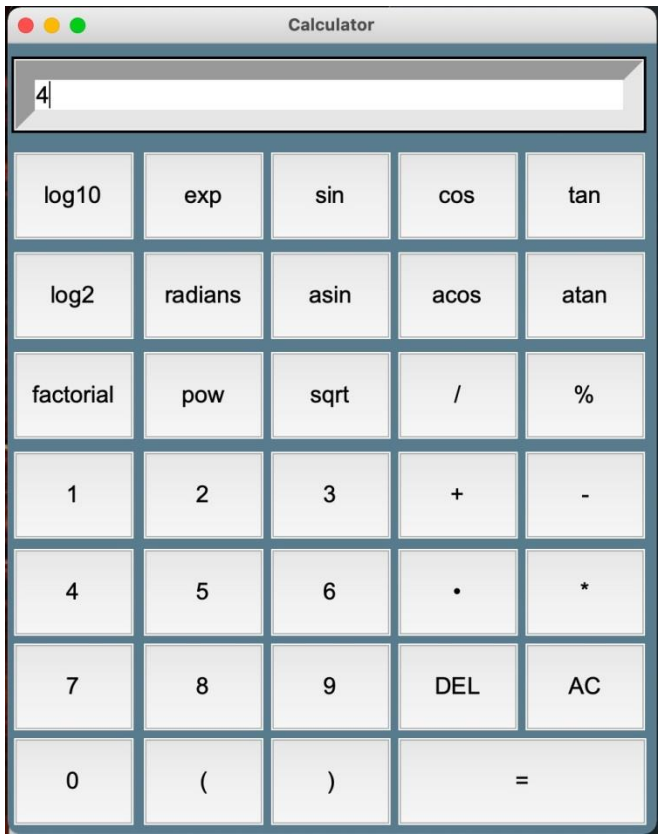
Factorial Function:



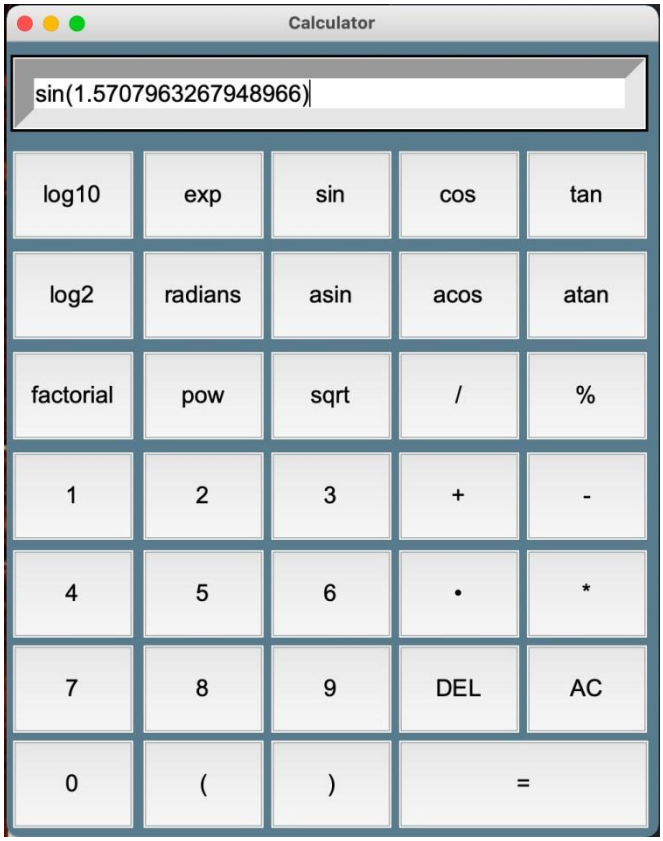
SQRT Function:

Modulus Function:





Sine Function:



CHAPTER 4 - CONCLUSION

We have successfully designed a python calculator with tkinter. This means that you have a better understanding of the way tkinter is used to build GUI applications.

This project has very been trustworthy and informative. It’s created us learn and perceive the various trivial ideas of Python Language. As we've used python Tkinter as an interface it provides varied controls, like buttons, labels, and text boxes to make an easy application.

The invasive use of the net confirms the nice future and scope of the planned project. Finally, it's tutored us a valuable womb-to-tomb lesson concerning the enhancements and dealing.