

School of Computer Science and IT Department of Master of Computer Applications

Jain Knowledge Campus

#44/4 District Fund Road Jayanagar 9th Block Behind Big Bazaar, Bengaluru, Karnataka 560069

**Python Programming**

**Activity: 1**

**Python GUI application using tkinter**

**Secret Messaging Application**

**Name: YOGESH KUMAR**

**USN: 19MCAR0035**

**Semester: 4th Semester**

**Branch: MCA**

**Academic Year: 2020**

**Acknowledgement**

The successful completion of any task would be incomplete without thanking the people who made it possible whose constant guidance and encouragement motivated my effort with success.

It is my great pleasure to thank my guide “ Dr.Lakshmi JVN“for her excellent guidance, constant encouragement, support, constructive suggestions.

I, also thank all the faculties of the Computer Application Department for their suggestions enabled us to surpass many of the seemingly impossible hurdles.

Finally, I would like to show my gratitude to my family members and all friends for advice and kind co-operation for without which this project would have been just a dream.

Yogesh Kumar

**Declaration**

I hereby declare that the work which is being prepared in this activity report entitled “Python Programming “ for the award of the degree of Master of Computer Application, submitted in the Department of Master of Computer Application, Jain Deemed To Be University, Jayanagar 560068, Bengaluru, Karnataka, India is an authentic record of my own activity work carried out under the supervision of Dr.Lakshmi JVN.

Yogesh Kumar

USN Number: 19MCAR0035

Department of Master of Computer Application Jain Deemed To Be University, Jayanagar 560068, Bengaluru, Karnataka, India

Signature of the Evaluator Marks Obtained

**Python GUI application using tkinter:**

**Secret Messaging Application:**

#Step1: import tkinter module

from tkinter import \*

# import other necessery modules

import random

import time

import datetime

#Step2:- creating root object

root = Tk()

# defining size of window

root.geometry("1200x1000")

# setting up the title of window

root.title("Message Encoding and Decoding")

root.configure(bg="black")

#Step3:setting the frames

Tops = Frame(root, width = 1600, relief = GROOVE ,background="blue")

Tops.pack(side = TOP)

f1 = Frame(root, width = 800, height = 600, relief = GROOVE)

f1.pack(side = LEFT)

f1.configure(bg="black")

#Step4:setting Labels and their position

lblInfo = Label(Tops, font = ('helvetica', 40, 'bold'),

text = "SECRET MESSAGING \n Encoding | Decoding", background="black",

fg = "White", bd = 8, anchor='w')

lblInfo.grid(row = 0, column = 0)

#Step5: Creating 5 variable of StringVar type

rand = StringVar()

Msg = StringVar()

key = StringVar()

mode = StringVar()

Result = StringVar()

# exit function

def qExit():

root.destroy()

# Step6:-Function to reset the window

def Reset():

rand.set("")

Msg.set("")

key.set("")

mode.set("")

Result.set("")

#Step7:Setting reference

lblReference = Label(f1, font = ('arial', 14, 'bold'), background="black" ,fg="white",

text = "Name:", bd = 14, anchor = "w")

lblReference.grid(row = 0, column = 0)

#input Box

txtReference = Entry(f1, font = ('arial', 14, 'bold'),

textvariable = rand, bd = 8,

bg = "LemonChiffon", justify = 'right')

txtReference.grid(row = 0, column = 1)

# labels

lblMsg = Label(f1, font = ('arial', 14, 'bold'), bg="black" ,fg="white",

text = "MESSAGE", bd = 14, anchor = "w")

lblMsg.grid(row = 1, column = 0)

txtMsg = Entry(f1, font = ('arial', 14, 'bold'),

textvariable = Msg, bd = 8,

bg = "LemonChiffon", justify = 'right')

txtMsg.grid(row = 1, column = 1)

lblkey = Label(f1, font = ('arial', 14, 'bold'), bg="black" ,fg="white",

text = "KEY", bd = 14, anchor = "w")

lblkey.grid(row = 2, column = 0)

txtkey = Entry(f1, font = ('arial', 14, 'bold'),

textvariable = key, bd = 8, insertwidth = 3,

bg = "LemonChiffon", justify = 'right')

txtkey.grid(row = 2, column = 1)

lblmode = Label(f1, font = ('arial', 14, 'bold'), bg="black" ,fg="white",

text = "MODE(e for encrypt, d for decrypt)",

bd = 14, anchor = "w")

lblmode.grid(row = 3, column = 0)

txtmode = Entry(f1, font = ('arial', 14, 'bold'),

textvariable = mode, bd = 8, insertwidth = 3,

bg = "LemonChiffon", justify = 'right')

txtmode.grid(row = 3, column = 1)

lblService = Label(f1, font = ('arial', 14, 'bold'), bg="black" ,fg="white",

text = "The Result-", bd = 14, anchor = "w")

lblService.grid(row = 2, column = 2)

txtService = Entry(f1, font = ('arial', 14, 'bold'),

textvariable = Result, bd = 8, insertwidth = 4,

bg = "LemonChiffon", justify = 'right')

txtService.grid(row = 2, column = 3)

#Step8:Working of all the buttons defines here

# cipher

import base64

# Function to encode

def encode(key, clear):

enc = []

for i in range(len(clear)):

key\_c = key[i % len(key)]

enc\_c = chr((ord(clear[i]) +

ord(key\_c)) % 256)

enc.append(enc\_c)

return base64.urlsafe\_b64encode("".join(enc).encode()).decode()

# Function to decode

def decode(key, enc):

dec = []

#base64.urlsafe\_b64decode() is function in python library (base64)

enc = base64.urlsafe\_b64decode(enc).decode()

for i in range(len(enc)):

key\_c = key[i % len(key)]

dec\_c = chr((256 + ord(enc[i]) -

ord(key\_c)) % 256)

dec.append(dec\_c)

return "".join(dec)

#Step9:Fetching value from input boxes.

def Ref():

print("Message= ", (Msg.get()))

clear = Msg.get()

k = key.get()

m = mode.get()

if (m == 'e'):

Result.set(encode(k, clear))

else:

Result.set(decode(k, clear))

#Step10: Define Buttons

#Submit button

btnTotal = Button(f1, padx = 18, pady = 10, bd = 16, fg = "black",

font = ('arial', 16, 'bold'), width = 10,

text = "Submit", bg ="Gold",

command = Ref).grid(row = 12, column = 1)

# Reset button

btnReset = Button(f1, padx = 18, pady = 10, bd = 16,

fg = "black", font = ('arial', 16, 'bold'),

width = 10, text = "Reset", bg = "DarkSeagreen",

command = Reset).grid(row = 12, column = 2)

# Exit button

btnExit = Button(f1, padx = 18, pady = 10, bd = 16,

fg = "black", font = ('arial', 16, 'bold'),

width = 10, text = "Exit", bg = "IndianRed",

command = qExit).grid(row = 12, column = 3)

#Step11: keeps window alive

root.mainloop()

**Screenshots:**

