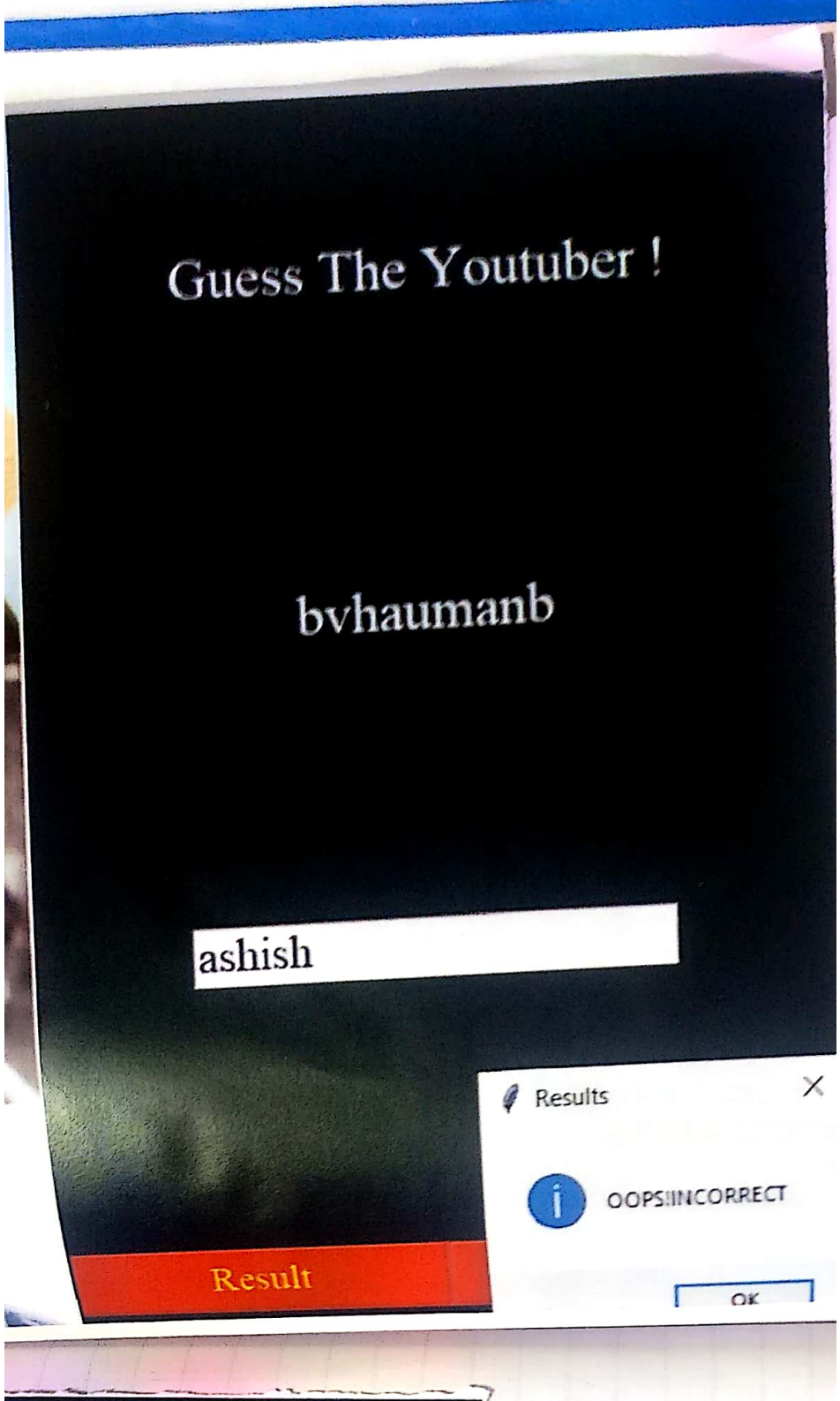


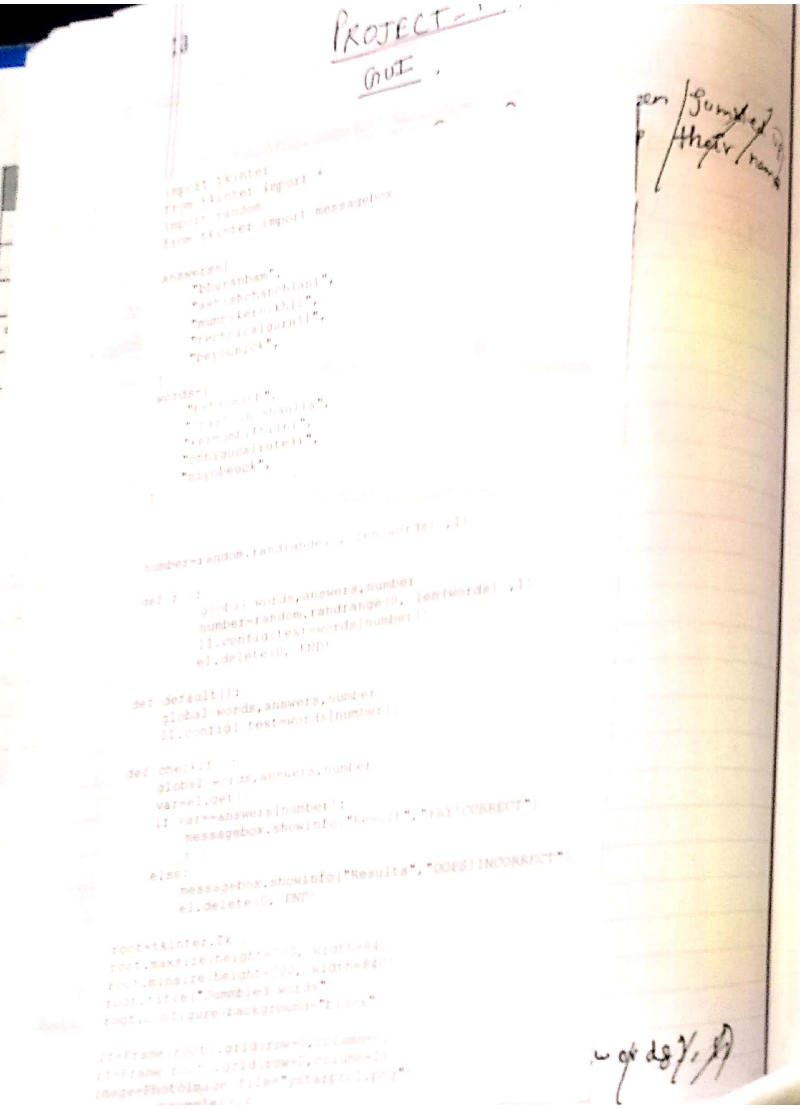
Result

Reset

PROJECT-1
GUI.

PRACTICAL - 7
(PROJECT-I).
(GUI).

Idea :- 5 youtubers (famous) have been jumbled up. If you are their fan, guess their names correctly.



Step 8: Finally use the fetchall method for display the value for the table using the cursor objects.

Step 9: Use the execute method and the drop table syntax for terminating the database finally. Use the close method.

Scanned with CamScanner

Step 1: Import the corresponding library for database connection.

Step 2: Now create connection objects using the library and connecting method for create the new database.

Step 3: Now create the cursor objects using cursor method from the connection objects created in steps 1 & 2.

Step 4: Now use the executing method in creating the table with the column name and respective data type.

Step 5: Now use the cursor objects use insert statement for entering the values coronding into the different field conditioning the data types.

Step 6: Use the commit method to complete the operation. Use the connection objects.

Step 7: Use the execute statement along with the cursor objects for accessing the value from data-base using selecting from where clause.

Code:-

```
import os, sqlite3
connection = sqlite3.connect("student.db")
C1 = connection.cursor()
C1.execute('create table student (Name, RNO, Pos)')
C1.execute("insert into student values ('Rajesh', 1848, 22-05-2000)")
C1.execute("insert into student values ('[illegible]', 1841, 25-04-1996)")
C1.execute("insert into student values ('[illegible]', 1842, 02-02-2000)")
connection.commit()
C1.execute('select * from student')
C1.fetchall()
C1.execute("drop table student")
```

32

CODE :-

```
import dbm
db = dbm.open("database", "c")
if db["www"] == None
        print("good")
else
        print("not good")
```

OUTPUT :-

good

PRACTICAL-6

Q.

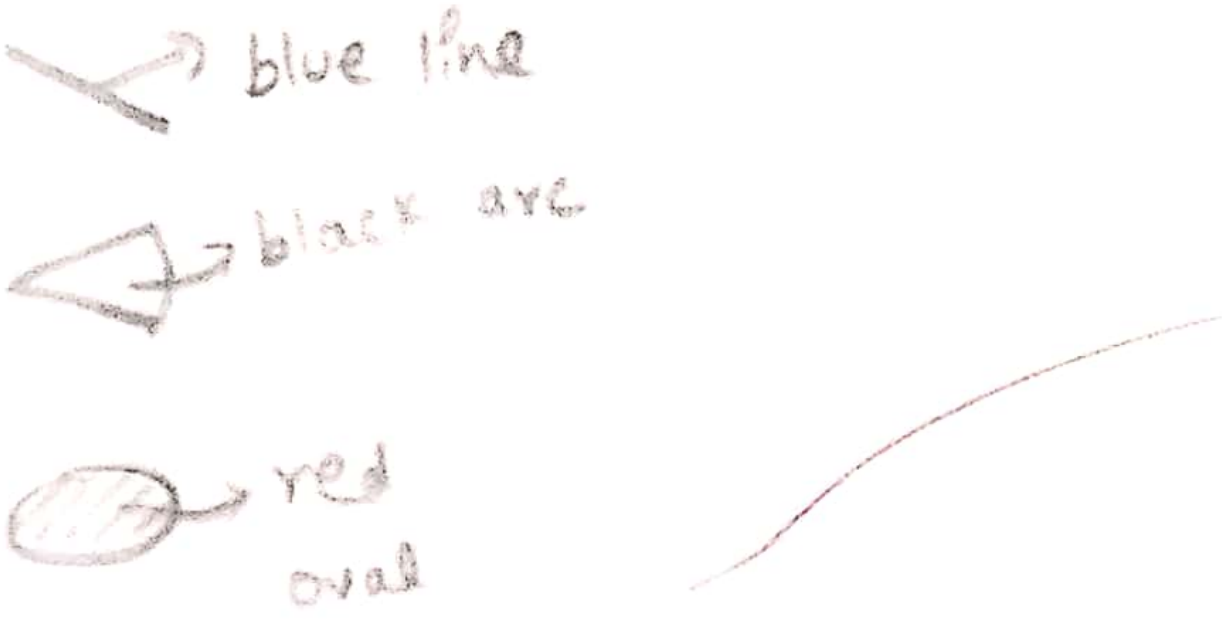
Aim: Database connectivity

Step 1 :- Import the library and use the open method for creating the data page by specifying name of the database along with the corresponding flag

Step 2 :- Use the objects for accessing to given size and the corresponding regular for the web size.

Step 3 :- Check whether the given URL address with the regular of the page is not equal to None than display the message from URL address else Not formed

Scanned with CamScanner



PRACTICAL - 5 (E)

Aim: GUI components.

#1. SPINBOX:

step 1:- Import tkinter.
step 2:- Add the spinbox syntax and use pack method.
step 3:- Use mainloop to terminate.

#2. Panned window:

step 1:- Import tkinter.
step 2:- Insert Panned window and use the pack method.
step 3:- use label method and add text along with add method.
step 4:- terminate by mainloop.

Step 8:- Now create 4 different button objects along with grid method attributes.

Step 9:- Use label method for right frame

Step 10:- Terminate the program using mainloop

PRACTICAL - 5 (CE):

Aim: Labels definition, frames.

ALGORITHM :-

Step 1:- Create parent window after import tkinter module and specify config, title, max size.

Step 2:- define four different functions :-
(i). name. (ii) address.
(iii). dob (iv) lm(latest movies).

Step 3:- Create the leftframe using leftframe method and give the specifications using .grid method.

Step 4:- Similarly, create the right frame and give specifications using .grid method.

Step 5:- Create 2 labels, one referring to the right frame with relief attribute raised and the other one also of rightframe. Both along with .grid attributes.

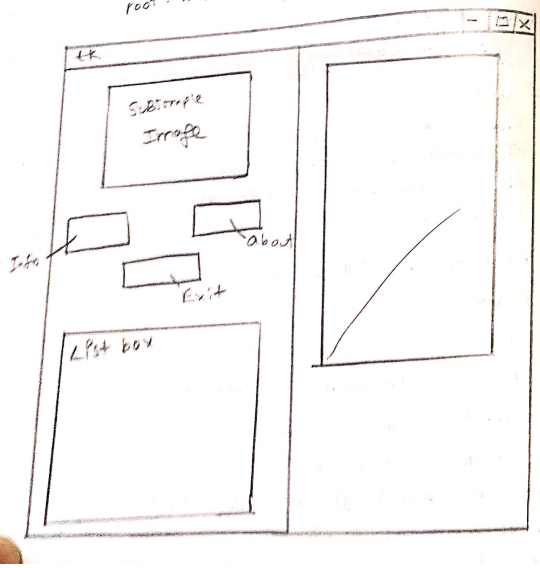
Step 6:- Create and object for using PhotoImage method and for inserting the image file (.png)

Step 7:- Create another object with (.Sample) method and giving attributes which are required

CODE :-

```
from tkinter import *
root = Tk()
root.config(bg="red")
root.title("Python")
root.maxsize(height=800, width=900)

def name():
    text1="Varun Dhawan"
    label1.config(text=text1)

def address():
    text2="VD villa, Bandra(w)"
    label2.config(text=text2)

def lm():
    text3="Street Dancer in Kalank"
    label.config(text=text3)

def dob():
    text4="20/March/1996"
    label.config(text=text4)

lf= Frame(root, height=500, width=400, bg="yellow").grid(row=0, column=0)
rf= Frame(root, height=500, width=500, bg="green").grid(row=0, column=1)

l1= Label(rf, text="Name", relief=RAISED).grid(row=1, column=0)
l2 = Label(rf).grid(row=0, column=1)
image = photoImage(file="vd.png")
```

l2.grid(padx=25, pady=10)
b1=Button(f1, text="Information", relief=SUNKEN,
command=info)
b1.grid(row=1, column=0)
b2=Button(f1, text="About", relief=SUNKEN,
command=about)
b2.grid(row=1, column=2, padx=5)
b3=Button(f1, text="Exit", relief=RAISED,
command=finish)
b3.grid(row=2, column=1, ipadx=15)
root.mainloop()



height and width specified. and subsequently use the grid() method with row & column attribute specified.

Step 8: Similarly create another frame object as declared by step 7.

Step 9: Create another object & use the subsample (3,4)

Step 10: Use value widget along with the frame object, relief attribute and subsequently use the grid() method.

Step 11: Now create button object dealing with different section of frame.

PRACTICAL - 5(A)

GUI Components

Import relevant methods from the tkinter library.

: Create parent window object and use the config method along with bg color attribute specified

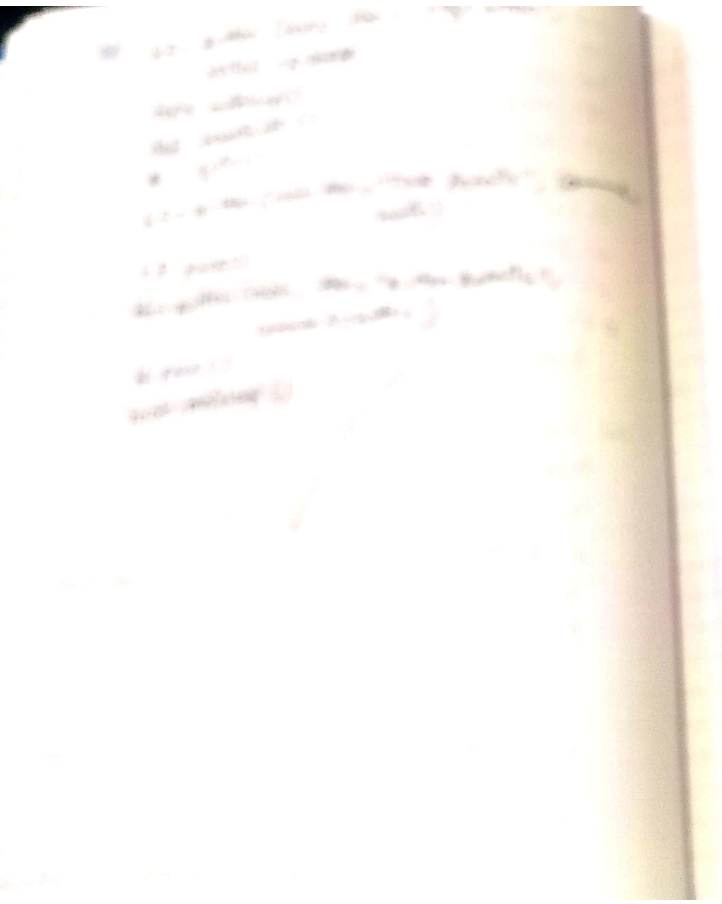
: Define a function finish with the messagebox widget which will display a message i.e. a warning message and subsequently terminate the program.

Step 4: Define a function info use a listbox widget along with the object of the same. Use the listbox object along with insert method and insert the same and finally use the grid() with ipadx attribute

Step 5: Define a function about us with value widget and text attribute and subsequently use the grid() method.

Step 6: Use PhotoImage widget with file and filename with .gif attribute

```
from tkinter import *
root = TK()
root.config(bg="green")

def finish():
    messagebox.askokcancel("warning", "This will end
                                the program")
    quit()

def info():
    list1 = Listbox()
    list1.insert(1, "Co.Name: Apple")
    list01.insert(2, "Products: Iwatch")
    list 1.insert(3, "O.S: IOS")
    list 1.grid(ipadx=70)

def about():
    list2 = Label(text="Info")
    list 2 0.grid(ipadx=30)
    list 3 = Label(text="Steve Jobs")
    list 3.grid()

p1 = PhotoImage(file="d abc.gif")
f1 = Frame(root, height=35, width=5)
f1.grid(row=1, column=0)
f2 = Frame(root, height=250, width=500)
f2.grid(row=1, column=1)
p2 = p1.Ø sab sample(5,4)
l1 = Label(fi, image=p2, relief=FLAT)
l1.grid(row=1, column=0)
```



```
def second():
    top2 = Tk()
    top2.config(bg="orange")
    top2.title("About us!!")
    top2.minsize(300, 300)
    L = Label(top2, text="created by: Vedant Sharma!")
    L.pack()
    b3 = Button(top2, text="prev", command=main)
    b3.pack(side=LEFT)
    b2 = Button(top2, text="exit", command=terminate)
    b2.pack(side=RIGHT)
    top2.mainloop()

def button():
    top3 = Tk()
    top3.geometry(300x300)
    b1 = Button(top3, text="flat button", relief=FLAT)
    b1.pack()
    b2 = Button(top3, text="groove button", relief=
                GROOVE)
    b2.pack()
    b3 = Button(top3, text="raised button", relief
                = RAISED)
    b3.pack()
    b4 = Button(top3, text="sunken button.", relief
                = SUNKEN)
```

```
# multiple window
# different button (relief())
from tkinter import *
root = TK()
root.minsize(300,300)
def main():
    top = TK()
    top.config(bg="black")
    top.title("Home")
    top.minsize(300,300)
    l1 = Label(top, text = "SPOT Photos"
"In Place of Interest : In JAPAN")
    l1.pack()
    b1 = Button(top, text = "next", command
            = second)
    b1.pack(side = RIGHT)
    b2 = Button(top, text = "exit", command
        = terminate)
    b2.pack(side = LEFT)
    top.mainloop()
```


Step 1 : Import the relevant methods from the tkinter library along with parent window object declared.

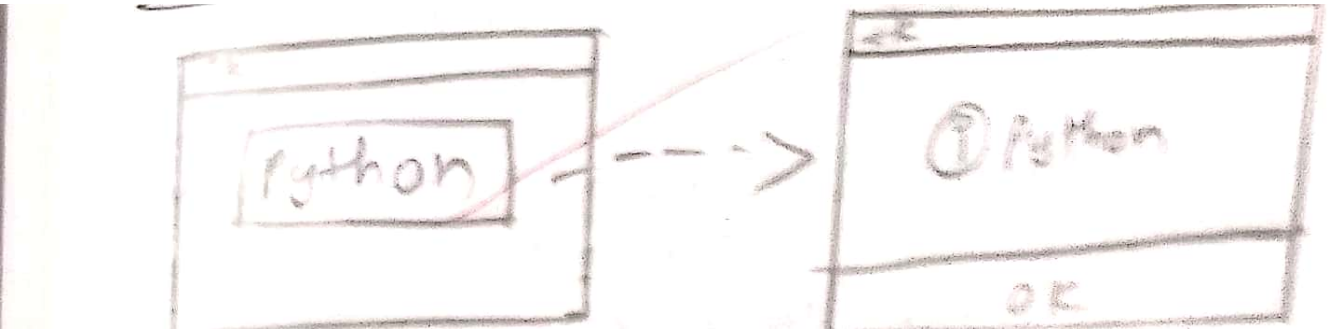
Step 2 : Use parent window object along with minsize function for window size.

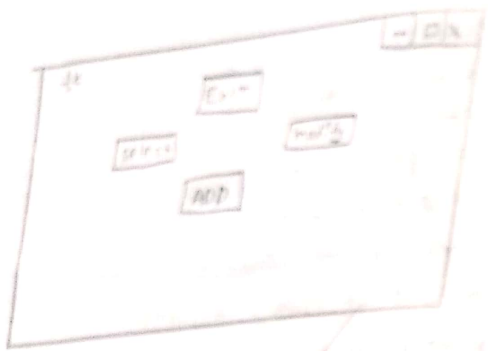
Step 3 : Define a function main, declare parent window object and use config(), title(), minsize(), Label() as well as button() and use pack() & mainloop() simultaneously.

Step 4 : Similarly define the function second and use the attributes accordingly.

Step 5 : Declare another function button along with parent object and declare button with attributes like FLAT, RIDGE, GROOVE, RAISED, SUNKEN along with the relief widget

Step 6 : Finally call the mainloop() for event driven programming.





step 8 → create another button object and place it in the RIGHT frame & label the button as ADD.

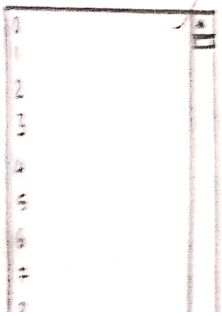
step 9 → Add another button & puts it on the left frame and label it as Exit.

step 10 → use the pack() simultaneously for all the objects & finally use the mainloop().

#4°

```
from tkinter import *
window = TK()
window.geometry("680x500")
frame = Frame(window)
frame.pack()
leftframe = frame(window)
leftframe.pack(side="left")
rightframe = Frame(window)
rightframe.pack(side="RIGHT")
b1 = Button(frame, text="Select", activebackground=
     "red", fg="blue")
b2 = Button(frame, text="modify", activebackground=
     "yellow", fg="black")
b3 = Button(frame, text="ADD", activebackground=
     "blue", fg="red")
b4 = Button(frame, text="EXIT", activebackground=
     "red", fg="green")
b1.pack(side="LEFT", padx=20)
b2.pack(side="right", padx=30)
b3.pack(side="bottom", pady=20)
b4.pack(side="top").
```

```
Sidget
import *
TK()
ometry ("680x500")
low, text = "numbers:"). pack ()
me (window)
K()
eg = Listbox (frame, width = 20, height = 20,
    font = ("Times New Roman", 10))
pack (side = "left", fill = "y")
= scrollbar (frame, orient = "vertical")
config (command = list no dog. yview)
.pack (side = "right", fill = "y")
n range (100):
odes. insert (END, str(k))
mainloop ()
```



PTO ▷

#4:

Step 1 → Import relevant methods from tkinter library.

Step 2 → Define the object corresponding to parent window and define the size of parent window and define the size of the parent window in terms of no. of pixels.

Step 3 → Now define the frame object from the method and place it on to the parent window.

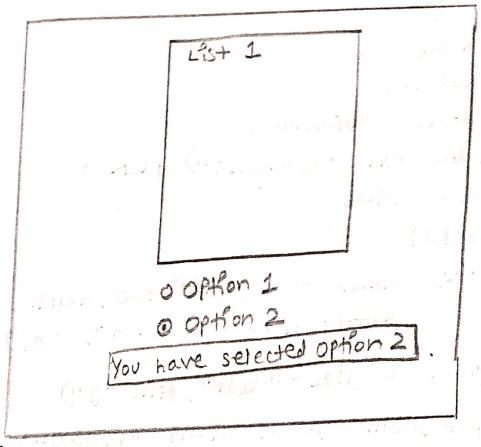
Step 4 → Create another frame object termed as the left frame and put it on the parent window on its LEFT side.

Step 5 → Similarly define the RIGHT frame and subsequently define the button object placed onto the given frame with the attribute as text, active background and foreground.

Step 6 → Now use the pack() along with the side attribute.

Step 7 → Similarly create the button object corresponding to the MODIFY operation put it into

Output:



#2:

```
Scrollbar()
from tkinter import *
root = TK()
root.geometry("500x500")
S = Scrollbar()
S.pack(side="right", fill="y")
root.mainloop()
```

#3:

Step 1 → Import the relevant libraries from the tkinter method.

Step 2 → Create an corresponding object of the parent window.

Step 3 → Use the geometry manager with pixel size (680 x 500) or any other suitable pixel value.

Step 4 → Use the label widget along with the parent object created and subsequently use the pack method.

Step 5 → Use the frame widget along with the parent object created and use the pack method.

Step 6 → Use the list box method along with the attributes like width, height font ~~bo~~ ~~weight~~
Do create a listbox method's object use pack() for the same.

Step 7 → Use the scrollbar() with an object use the attribute of vertical. then configure the same with object created from the scrollbar() and use pack().

Step 8 → Trigger, the events using mainloop.

Step 6 → trigger the function declared.

Step 7 → Now call the pack() for radio object is created and specify the requirement using anchor attribute.

Step 8 → Finally make use of the mainloop along with parent object.

#2.

Step 1 → Import relevant methods from the tkinter library.

Step 2 → Create a parent object corresponding to the parent window.

Step 3 → Use the geomentry() for laying of the window.

Step 4 → Create an object and use the scrollbars

Step 5 → Use the pack() along with the ~~structure~~ scrollbar with side and fill attributes

Step 6 → use the mainloop with the parent object.

#1
Radio button

```
from tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    selection = "you just selected" + str(var.get())
    l1 = Label(text = selection, bg = "white",
        fg = "green")
    l1.pack(side = TOP)

var = StringVar()
l1 = Listbox()
l1.insert(1, "List 1")
l2.insert(2, "List 2")
l1.pack(anchor = N)
r11 = RadioButton(root, text = "Option 1", variable
    = var, value = "option 1", command = select)
r11.pack(anchor = N)
r12 = RadioButton(root, text = "Option 2", variable =
    var, value = "option 2", command = select)
r12 = pack(anchor = N)
root.mainloop().
```

PRACTICAL - 5(B)

Aim: GUI components.

#1:

Step 1 → Import the relevant methods from the tkinter library. Create an object with the parent window.

Step 2 → Use the parent window object along with the geometry() declaring specific pixel size of the parent window.

Step 3 → Now define a function which tells the user about the given selection made from the multiple option available

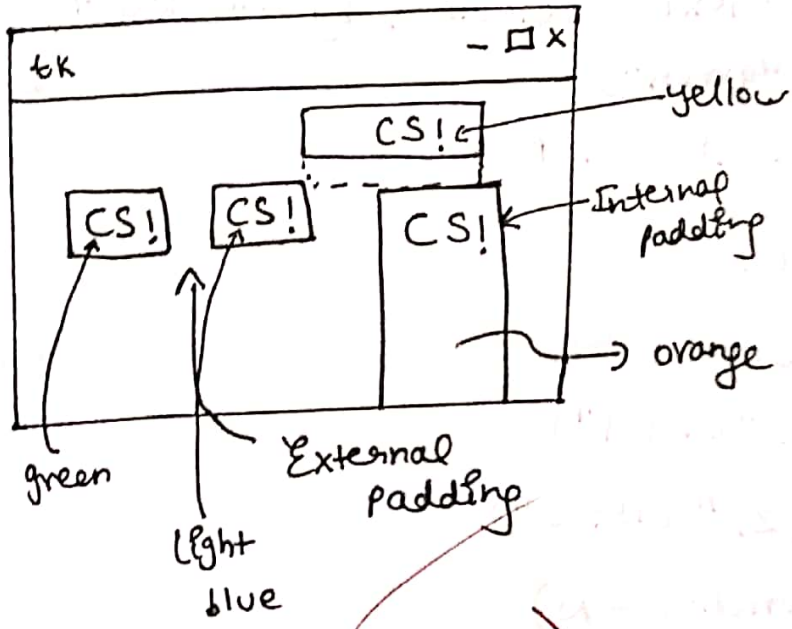
Step 4 → Now define the parent window and define the option with control variable.

Step 5 → Use the listbox() and insert options on the parent window along with the pack() with specifying anchor attribute.

Step 6 → Create an object from radiobutton which will take following arguements parentwindow object(), text variable which will take the values option no 1, 2, 3. ... variable arguments corresponding value.

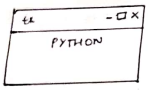
```
l4= Label (root, text = "CS!", bg = "orange",
          fg = "black", font = "10")
l4. pack (side = TOP, ipady = 50)
root. mainloop()
```

OUTPUT :-



```
## # Creation of parent window from TKinter import #
root = tk()
l = Label (root, text = "Python")
l.pack()
root.mainloop()
```

OUTPUT :-



#2

```
# from tKinter import #
root TK()
l = label (root, text="Python")
l. pack()
l1 = Label (root, text="CS!", bg="green", fg="black"
      font="10")
l1. pack (side =LEFT, padx = 20)
l2 = Label (root, text = "CS!", bg = "light blue"
      fg = "black", font = "20")
l2. pack (side = LEFT, pady = "30")
l3 = Label (root, text="CS!", bg = "yellow",
      fg = "black", font = "10")
l3.pack (side=TOP, ipadx = 40)
```


PRACTICAL - 5

* Topic :- GUI components.

Step 1 :- Use the tkinter library for importing the font features of the text widget.

Step 2 :- Create an object using the tk().

Step 3 :- Create a variable using the widget Label and use the text method

Step 4 :- Use the mainloop() for triggering of the corresponding above mention events.

#2

Step 1 :- Use the tkinter library for importing the features of the tex widget.

Step 2 :- Create a variable from the text method and position it on the parent window.

Step 7: import re module declare a string use the findall() with finding the vowels in the string and declare the same

Step 8: import re module declare the host and domain name declare pattern for separating the host & domain name use the findall() and print the output respectively

Step 9: import re module enter a string use pattern to display only two elements of the particular string use the findall() declare variables with initial value of zero use for conditional statement and subsequently use the if condition check whether condition satisfy add up the or else statement value and display the values subsequently

38

```
>>> criteria matched for all no
    criteria matched for all no
    criteria failed
    criteria matched for all no

# values
import re
str1 = 'plant is life overall'
output = re.findall(r'[aeiou]', str1, re.I)
print(output)
>>> ['a', 'i', 'e', 'a', 'o', 'e', 'a', 'u']

# counting of first 2 letters
import re
s = 'mr. a, @ ms.b, mrs.c, mr. d'
p = r'(?:ms|mr)'
o = re.findall(p, s)
print(o)
m = 0
f = 0
for v in o:
    if (v == 'ms'):
        f = f + 1
    else:
        m = m + 1
print("no of males is =", m)
print("no of females is =", f)
>>> ['mr', 'ms', 'ms', 'mr']
no of males is = 2
```

```
# ppn - space:
import re
string = ' abc def ghi'
pattern = r'\s+'
replace = ''
V1 = re.sub(pattern, replace, string)
print(V1)
>>> abcdefghi
# group()
import re
sequence = 'python is an interesting language'
V = re.search('\A python', sequence)
print(V)
V1 = V.group()
print(V1)
>>> <_sre.SRE_match object at 0x0281bf00>
    python.
# verifying the given set of phone numbers.
import re
list1=['8004567891', '9145673210', '7865432981',
       '9876543201']
for value in list1:
if re.match(r'[8-9]{1}[0-9]{9}',
value or len(value) == 10):
    print ("criteria matched for all numbers")
else:
    print ("Criteria failed!")
```


Step 4: Import the module declare string and accordingly declare pattern replace the black space with no-space. Use sub() with 3 arguements and print the string without spaces.

Step 5: Import re module declare a sequence use search method for finding subsequently use the group() with dot operator as search() gives memory location using group() it will show up the matched string.

Step 6: Import the module declare list with nos. Use the conditional statement here we have used up the for conditon statement. use if condition for checking first no. is either 8 or 9 and next 9 no. are in range of 0 to 9 and check whether the entered nos. are equal to 10. If criteria matches print cell numbers matched otherwise print failed.

PRACTICAL - 4

* Topic : Regular expression.

Step 1 : Import are module declare pattern and declare sequence use match method with declare arguements if arguements matched then print the same otherwise print pattern. NOT found!

Step 2 : Import re module declare pattern with literal and meta character Declare string value... use the findall() with arguments and print the same.

Step 3 : Import re module declare pattern with meta char. use the split() and print the output.

```
# match()
import re
pattern = r"PYCS"
sequence = "PYCS represents comp. sci. stream"
if re.match(pattern, sequence):
    print("matched pattern found!")
else:
    print("NOT FOUND!")
>>> matched pattern found!

# numerical values (segregation)
import re
pattern = r'\d+'
string = 'hello 123, howdy 789, 45 howru'
output = re.findall(pattern, string)
print(output)
>>> ['123', '789', '45']

# split()
import re
pattern = r'\d+'
string = 'hello 123, howdy 789, 45 howru'
output = re.split(pattern, string)
print(output)
>>> ['hello', 'howdy', 'howru']
```

using except keyword

```
try:
    a = open("abc.txt", "w")
    a.write("Python")
except IOError:
    print("Error !!")
def x():
    l = []
    print(len(l))
def y():
    l = [2, 4, 6, 1]
    print(len(l))
print(x())
print(y())
```

Output :
successful
0
None
4
None

raise keyword :

```
try:
    a = int(input("enter a number :"))
    raise ValueError
except ValueError:
```

Step 12 : use try block open a file in write mode and subsequently enter values in the file

Step 13 : use the IOError and display appr message

Step 14 : define a function with empty list and calculate the length of the list

Step 15 : define another function y initialize or delete some elements in list and calculate the length of the same and display the same

Step 16 : In try block accept input from the user and if the user enters character value raise an error that is saying up enter the integer values

Step 8: Accept an integer value from the user.
To do it by use the division method.

Step 9: For the exception to be raised use the except keyword (cmd) i.e. TypeError prints Incompatible values.

Step 10: Use except with error as zero division Error and print the message according that if the entered number is zero not able to perform operation!

Step 10: Declare first variable and values.

Step 11: For multiline exception use the error types by seperating them with a comma.


```
# type error, zero-division error
a= 1
b = input ("enter:")
try:
    print (a/b)
except TypeError:
    print ("Incompatible value")
except ZeroDivisionError:
    print ("denominator is 0 so operation cannot
           be performed")
```

```
>>> enter = t
    Incompatible value
>>> enter = 2
      0.5
>>> enter : 0
    denominator is 0. so operation cannot be
      performed.
```

```
# multiline exception
a,b = 1,0
try:
    print (1/0)  → 1
    print ('10'+10)  → '10'

except (TypeError, ZeroDivisionError):
      print("Invalid Input!")
-->  1.0
```

IOError:

```
try:
    fo = open("abc.txt", "w")
    fo.write("Python is an intendent language")
except IOError:
    print("Enter opp. mode for file operation")
else:
    print("Operation is successful!")
```

>>> Operation is successful!

A output:

>>> enter appropriate mode for file operation

Value Error:

```
try:
    x = int(input("Enter a statement:"))
except ValueError:
    print("Arithmetic Error!")
else:
    print("Operation is successful!")
```

>>> Enter a statement: abc
Arithmetic Error!

>>> Enter a statement: 123
Operation is successful!

PRACTICAL-3

* Topic: Exceptions.

Step 1: In the try block open a file in the open mode with write mode. write some contents in file

Step 2: In except (IOError) block use the error in IOError, use the appropriate message to display.

Step 3: Else display the operation is successful!

Step 4: In try block accept an input from the user

Step 5: In except block use valueerror and print the some message.

Step 6: Else display the operation is successful!