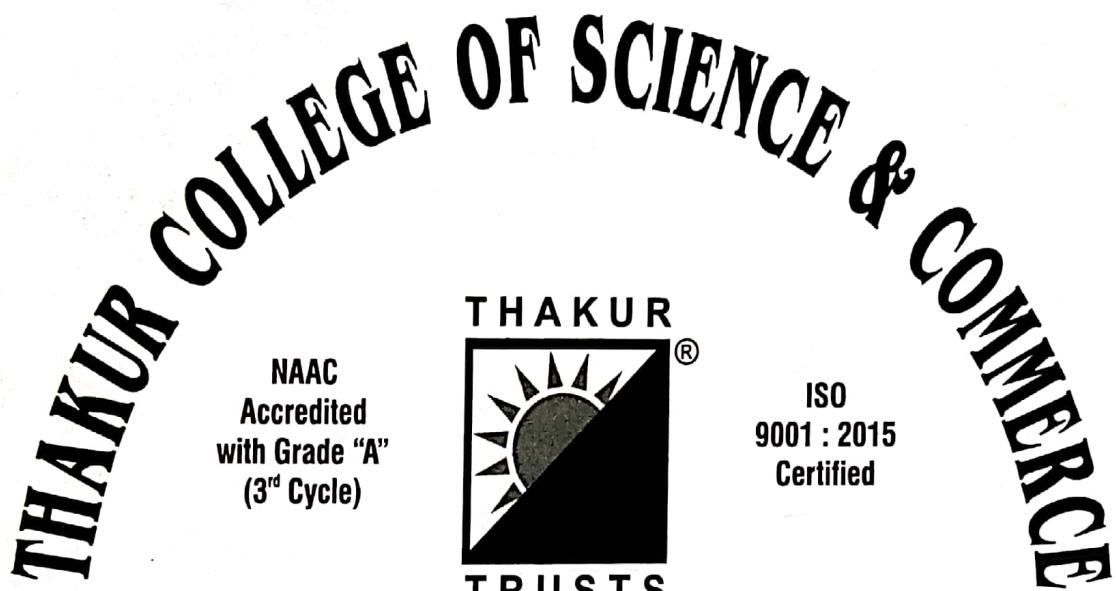


PERFORMANCE

Term	Remarks	Staff Member's Signature
I	<u>completed</u> <u>5/000</u>	 27/1919
II		

Exam Seat No. _____



NAAC
Accredited
with Grade "A"
(3rd Cycle)



ISO
9001 : 2015
Certified

Degree College
Computer Journal
CERTIFICATE

SEMESTER II UID No. _____

Class FYBSC CS Roll No. 1738 Year 2019 - 20

This is to certify that the work entered in this journal
is the work of Mst. / Ms. Anukai Suraj

Shivkumar
who has worked for the year 2019 - 20 in the Computer
Laboratory.

Teacher In-Charge

Head of Department

Date : _____

Examiner

★ ★ INDEX ★ ★

No.	Title	Page No.	Date	Staff Member's Signature
01	File handling using Python	23	26/11/19	Dr. [Signature]
02	Iterators	26	31/12/19	Dr. [Signature]
03	Exception	29	17/12/19	Latif [Signature]
04	Regular Expression	31	24/12/19	Dr. [Signature]
05	(a) GUI component (label) 5(A)	34	7/01/20	Dr. [Signature]
	(b) GUI component using radio, Frame and scrolling	36	14/01/20	Dr. [Signature]
5C	GUI (c)	40	21/01/20	Dr. [Signature]
5D	GUI component (D)	47	28/01/20	Dr. [Signature]
5E	GUI (E)	51	11/2/20	Dr. [Signature]
6	Database Connectivity	53	18/2/20	Dr. [Signature]
7	Project	58		

Source code :-

```
# Write Method :-  
fileobj = open("abc.txt", "w")  
fileobj.write("Subjects in "+ "\n Computer Science")  
fileobj.write("\n Python \n Dm \n Java \n C")  
fileobj.close()  
  
# Read Method :-  
fileobj = open("abc.txt", "r")  
Str1 = fileobj.read()  
print("The output of read method is : " + Str1)  
fileobj.close()  
'''The output of read method is : Subject in computer science  
Python  
Dm  
Java  
C  
  
# Read Line Method :-  
fileobj = open("abc.txt", "r")  
Str2 = fileobj.readline()  
print("The output of read is : " + Str2)  
fileobj.close()  
'''The output of read method is Subject in computer science  
  
# Readlines Method :-  
fileobj = open("abc.txt", "r")  
Str3 = fileobj.readlines()  
print("The output of readlines is : " + Str3)  
fileobj.close()  
'''The output of readlines method is : ['Subject in computer  
science\n', 'Python\n', 'Dm\n', 'Java\n', 'C\n']
```

26/11/19

23

PRACTICAL - 1

Aim :- To demonstrate the use of different file accessing modes, different attributes, read methods.

ALGORITHM :-

Step 01 :- Create a file object using open method and use the write access mode followed by writing some contents onto the file and then closing the file.

Step 02 :- Now, open the file in read mode and then use read(), readline() and readlines() and store the output in variable and finally display the contents of variable.

Step 03 :- Now, use the fileobject for finding the name of the file, the file mode in which its opened whether the file is still open or close and finally the output of the softspace attribute.

Step 04 :- Now, open the fileobj in write mode, write another content close subsequently. Then again open the file obj in "wt" mode that is the update mode and write contents.

18

Step 05 :- Open fileobj in read mode display the update written content and close. Open the fileobj in 'r+' mode with parameter passed and display the output subsequently.

Step 06 :- Now open file obj in append mode open write method write contents close the fileobj again open the fileobj in read mode and display the appending output.

Step 07 :- Open the fileobj in read mode, declare a variable and perform fileobject do "tellmethod" and store the output in variable.

Step 08 :- Use the Seek method with the arguments with opening the file object in read mode and closing subsequently.

```
# file attributes:  
a = fileobj.name  
b = fileobj.close()  
c = fileobj.mode  
d = fileobj.softspace  
print("Name of file : ", a)  
print("closed : ", b)  
print("filemode : ", c)  
print("softspace : ", d)  
>>> Name of file : abc.text  
>>> closed True  
>>> filemode : r  
>>> softspace 0
```

```
# w+  
fileobj = open("abc.text", "w+")  
fileobj.write(" Hey Siri")  
fileobj.close()
```

```
# r+  
fileobj = open("abc.text", "r+")  
a1 = fileobj.read(10)  
print("The output of r+ is : ", a1)  
>>> The output of r+ is : Hey Siri
```

```
# Tell Method:  
fileobj = open("abc.text", "r+")  
pos = fileobj.tell()  
print("Tell", pos)  
fileobj.close()
```

29

```
# Write Mode:  
fileobj = open("abc.text", "w")  
fileobj.write("Hello World")  
fileobj.close()  
  
# Read Mode:  
fileobj = open("abc.text", "r")  
a = fileobj.read()  
print("The output of read mode  
is : ", a)  
fileobj.close()  
>>> The output of read mode  
is : Hello World
```

```
# Append Method:  
fileobj = open("abc.text", "a")  
fileobj.write("\n Google")  
fileobj.close()  
  
fileobj = open("abc.text", "a")  
a2 = fileobj.read()  
print("The output of append  
mode is : ", a2)  
fileobj.close()  
>>> The output of append  
mode is : Hey Siri Google
```

Scanned by CamScanner

```

# Seek Method :- 
fileobj = open("abc.txt", "r")
s = fileobj.seek(0, 0)
print("seek(0,0)", s)
fileobj.close()
>>> seek(0,0) 0
fileobj = open("abc.txt", "r")
s1 = fileobj.seek(0, 1)
print("seek(0,1)", s1)
fileobj.close()
>>> seek(0,1) 0
fileobj = open("abc.txt", "r")
s2 = fileobj.seek(0, 2)
print("seek(0,2)", s2)
fileobj.close()
>>> seek(0,2) 16

# Length of different lines:
fileobj = open("abc.txt", "r")
stro = fileobj.readlines()
print("Length of different lines", stro)
for line in stro:
    print(len(line))
fileobj.close()

>>> length of different lines
['Hey Sir Google', 'Hey sir\n', 'Google']

```

25

Step 9 :- Open file obj with read mode also use the readlines method And Store the output consequently in stro and print the same for counting the length use the for conditional statement and display the length.

JY 3/17

3/17/19

PRACTICAL - 2

Aims- Iterators

Step ALGORITHM :-

Step 01 :- To create a tuple with elements first we need to iterate using the iter and then the next method. The number time we use the iter and the next method it will be displayed the next iterating element in the tuple.

Step 02 :- The similar output can be obtained by using for conditional statement. An iterable variable is to be declared in for loop which will iterate. iterate

Step 03 :- Define a function name square with a parameter which will obtain output of square value of the given number. In similar fashion declare value to get the raised by 3 and return the same

Step 04 :- To call the declared function use the function call

Source code :-

```
*iter() and next()
mytuple1 = ("Apple", "Banana", "Orange")
myiter1 = iter(mytuple1)
print(next(myiter1))
myiter2 = iter(mytuple1)
print(next(myiter2))
myiter3 = iter(iter(mytuple1))
print(next(myiter3))
>>> Apple
Banana
Orange
```

* For loop :-
mytuple1 = ("Apple", "Banana", "Orange")
for a in mytuple1 :
 print(a)
>>> Apple
Banana
Orange

* Square and cube

```
def square(x):
    y = x * x
    return y
def cube(x):
    z = x * x * x
    return z
func1 = [square, cube]
for i in range(6):
    value = list(map(lambda x: x(i), func1))
    print(value)
```

```
>>> [0, 0]
[1, 1]
[4, 8]
[9, 27]
[16, 64]
[25, 125]
```

```

at map()
listnum = [0, 4, 5, 7, 9, 11, 13, 15, 20, 19, 25]
listnum = list(map(lambda x: x%5, listnum))
Print(listnum)
def even(x):
    if x%2 == 0:
        return "EVEN"
    else:
        return "ODD"
list = map(even, listnum)
for Odd number
class odd:
    def __iter__(self):
        self.num = 1
        return self
    def next_(self):
        num = self.num
        self.num += 2
        return num
myObj = odd()
myIter = iter(myObj)
x = int(input("Enter a number:"))
for i in myIter:
    if i < x:
        print(i)
>>> Enter a number :25
      1
      3
      5
      7
      9
      11
      13
      17
      19
      21
      23

```

- 27
- Step 05 :- The use for conditional Statement Specifying the range, use the `map` typecasting with `map` method declare a 'lambda' i.e anonymous function and print the same.
- Step 06 :- Declare a `listnum` variable and some elements then use the `map` method with help of `lambda` function Give them two argument and display output
- Step 07 :- Define a function `even` with a parameter then using conditional Statement do check whether the number is even and odd and return respectively.
- Step 08 :- Define a class and within that define the `iter()` method which will initialise the first element within the container object
- Step 09 :- Use the `next()` and define the logic for displaying odd value.
- Step 10 :- Define an object of a class
- Step 11 :- Accept an number from the user till which we want to display the odd number.

B] Factorial of a number using the Map Function

Step 01 :- Define a function which will accept one argument.

Step 02 :- Use if conditional statement and use the else condition by declaring the factorial expression.

Step 03 :- Take input from the user and use the type casting method to integer.

Step 04 :- Create a blank list and append the input value to the list.

Step 05 :- Use the map method and map the Factorial and list.

Step 06 :- After that use print statement and print the factorial of the number.

88 # Exception :- IO Error :

```
try:  
    fo = open ("Object.text", "w")  
    fo.write ("Hello World")  
    fo = open ("Object.text", "r")  
    str1 = fo.read()  
    print (str1)  
except IOError:  
    print ("I/O Error found")  
else:  
    print ("Operation Successful")
```

Output :-
» Hello World
Operation Successful.

29

PRACTICAL - 03

AIM :- Exception

Algorithm :-

Step 01 :- Use the try block to define the normal course of action. For example define the file obj and open the file in the write mode and write some content on the file.

Step 02 :- Use the except block I/O as a environment error and convey the appropriate message to the user else display the message that the operation carried out is successful.

* Step 03 :- Accept the value from the user and if it is a valid value display the entered value and terminate the condition by using the break statement.

Step 04 :- Define the except block to the value error as a key error and display the appropriate message

Step 05: We can define the multiple exception using the except statement for finding the different category of errors.

Step 06: Use the try block to define the normal course. For example

#AT
#ARITHMETIC

```
try:
    a = int(input("enter a number 1:"))
    b = int(input("enter a number 2:"))
    print("Division", a/b)
except ArithmeticError:
    print("Arithmetic error found")
else:
    print("Operation successful")
```

Output:
 enter a number 1: 10
 enter a number 2: 5
 Division. 2.0
 Operation Successful

~~last~~
 Dr
 21/11/2022

30

```
*matchc()
import re
pattern = "FYCS"
sequence = "FYCS represents computer science stream"
if re.match(pattern, sequence):
    print("matched pattern found!")
else:
    print("not found!")
>>> matched pattern found!
```

*numerical values (separation)

```
import re
pattern = '\d+'
String = 'hello23, howdy798, 59how84'
output = re.findall(pattern, String)
print(output)
>>> ['23', '798', '59']
```

*splitc()

```
import re
pattern = '\d+'
String = 'hello23, howdy798, 59how84'
output = re.split(pattern, String)
print(output)
>>> ['hello', 'howdy', 'how84']
```

*no-space:

```
import re
String = 'abc def ghi'
pattern = ' \t'
re.replace = ''
UI = re.sub(pattern, replace, String)
print(UI)
>>> abcdefghi
```

31

PRACTICAL - 04

Aim:- Regular expression

Algorithm:-

Step 1: Import re module declare pattern and declare sequence use match method with declare arguments if arguments matched than 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 character use the split() and print the output

Step 4 :- Import re module declare string and accordingly declare pattern replace the blank space with no-space use sub() with 3 arguments and print the string without space

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 re module declare list with numbers use the conditional statement here we have used up the for condition statement use if condition for checking first number is either 8 or 9 and next number are in range of 0 to 9 and check whether the entered number are equal to 10 if criteria matches print cell number matches otherwise print failed

32

```
#group()
import re
sequence = 'Python is an interesting language'
v = re.search('Python', sequence)
print(v)
v1 = v.group()
print(v1)
>>> <sre.SRE_Match object at 0x02810F00>
#verifying the given set of phone numbers import re.
list1 = ['9975489821', '9145673210', '9565482196', '7210658981']
for value in list1:
    if re.match(r'[8-9]{1}[0-9]{9}'):
        value or len(value) == 10:
            print("criteria matched for cell number")
    else:
        print("criteria failed!")
>>> criteria matched for cell number
criteria matched for cell number
criteria failed
criteria matched for cell number
```

```
# vowels:
import re
str1 = 'Plant is life overall'
output = re.findall(r'\b[AEIOUAEIOU]\w+', str1)
print(output)
>>> ['is', 'overall']
```

§8

#host & domain

```
import re
seq='abc-tsc@edu.com, xyz@gmail.com'
pattern=r'([w]+)@([w]+\.)+[w]+'
output=re.findall(pattern, seq)
print(output)
>>>['abc.tsc', 'edu.com', 'xyz', 'gmail.com']
```

counting of first 2 letters:

```
import re
s='mr-a, ms-b, msc-c, ms-e'
p=re.findall(s)
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', 'ms']
('No. of males is : 1')
('No. of females is : 2')
```

§9

Step 7 :- Import re module declare a string use the module with findall() for finding the vowel in the string and declare the same

Step 8 :- Import re module declare the host and domain name declare pattern for separating the host and 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 findall() declare two variables with initial value as zero use for condition and subsequently use the if condition check whether condition satisfy add up the or else increment value and display the values subsequently.

J
2/1/19

1.8.

PRACTICAL - OS(H)

Aim:- The GUI Components Using Label

Algorithm:

Step 1:- Use the tkinter module to import the relevant feature of 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 about mention event.

Step 5:- Use the pack() along with the object from the text method and use the parameters.

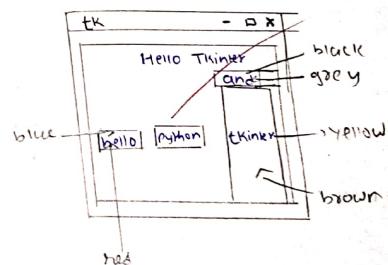
- 1] Side = LEFT , padx = 20
- 2] Side = LEFT , pady = 30
- 3] Side = TOP , ipadx = 40
- 4] Side = TOP , ipady = 50

Step 6:- Use the mainloop() for the triggering of the corresponding events

code:

```
from tkinter import *
root = Tk()
x = Label(root, text="Hello Tkinter")
x.pack()
x1 = Label(root, text="Hello", bg="blue", fg="red", font="10")
x1.pack(side=LEFT, padx=20)
x2 = Label(root, text="python", bg="white", fg="blue", font="30")
x2.pack(side=LEFT, pady=30)
x3 = Label(root, text="and", bg="black", fg="grey", font="20")
x3.pack(side=TOP, ipadx=40)
x4 = Label(root, text="Tkinter", bg="brown", fg="yellow", font="35")
x4.pack(side=TOP, ipady=50)
root.mainloop()
```

Output:



STEP 78- Now repeat the above steps with label

which takes the following arguments-

- 1] The name of Parent window
- 2] Text attribute which defines the string
- 3] The background color (bg) bg = " "
- 4] The foreground color(fg) fg = " " and
then use the pack() with relevant for
padding attributes.

Dr. MIA

Practical - 05(B)

Aim:- The GUI component using Radio, Frame and Scrolling

Algorithm:

Step01 :- Import the relevant method from the tkinter library.

Step02 :- Define the object corresponding to parent window and define the size of that window.

Step03 : Now define the Frame object from the method and place it on the parent window

Step04 :- Create another Frame object termed as the left frame and put it on the parent window on its left side.

Step05 :- Similarly define the Right frame and subsequently define the button object placed on to the given frame with the attribute as text, bg and fg.

Step06 :- Now use the pack method along with the side attribute

Code :- Frame

```
① from tkinter import *
root = Tk()
root.geometry("300x600")
frame = Frame(root)
frame.pack()

leftframe = Frame(root)
leftframe.pack(side="left")

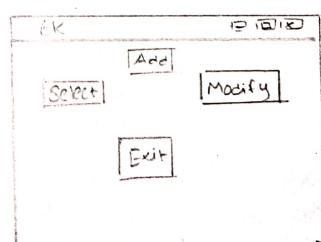
rightframe = Frame(root)
rightframe.pack(side="right")

B1 = Button(frame, text="Select", bg="blue", fg="white")
B1.pack(side="left", padx=20)

B2 = Button(frame, text="Modify", bg="black", fg="white")
B2.pack(side="right", padx=30)

B3 = Button(frame, text="Add", bg="white", fg="grey")
B3.pack(side="top", pady=40)

B4 = Button(frame, text="Exit", bg="blue", fg="purple")
B4.pack(side="bottom", pady=50)
```

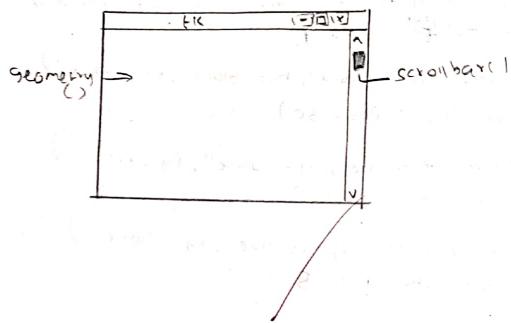
Output :-

38.

② Scrolling :-

```
from tkinter import *
root=Tk()
root.geometry("500x600")
Scrollbar()
S=Scrollbar()
S.pack(side="right", fill="y")
root.mainloop()
```

Output :-



37

Step 07:- Similarly create the button object corresponding to the modify operation and put it into the frame object with Side equal to Right attribute set.

Step 08:- Create another button object place on to the Right frame and label the button as a click.

Step 09:- Add another button and put it on to the Right frame object & turn exit.

Step 10:- Use the pack method simultaneously for all the objects and finally call the mainloop method.

Scrollbar

Step 1:- Import relevant method from the tkinter library

Step 2:- Create a parent object corresponding to the parent window.

Step 3:- Use the geometry() to define the size of parent window.

Step 4:- Create an object and use the scrollbar()

Step 5:- Use the pack() along with the scrollbar object with side and fill attribute.

Step 6:- Use the `mainloop` mainloop with the parent object

* Radio Button :-

Step 1:- Use the Tkinter module to import the relevant method.

Step 2:- Define a function which tells the user about the given selection made from the multiple option available.

Step 3:- Use the config method along with label object and call the variable as an argument within the method.

Step 4:- Now define the parent window and define option using control variable

Step 5:- Now create an object from the radio button method which will take the full argument

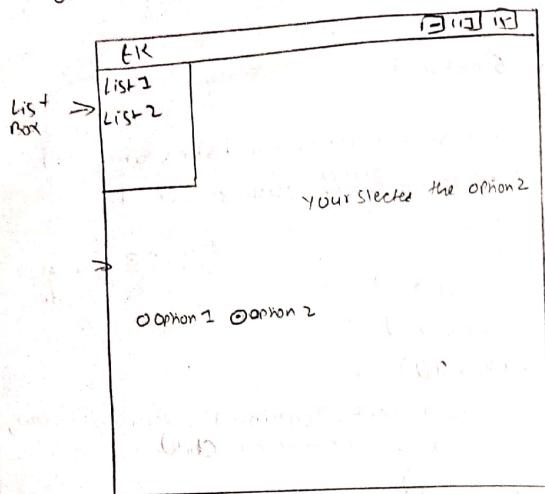
- 1) Positioning on parent window.
- 2) Defining text variable which will take value opt 1, 2, 3.
- 3) Define variable argument
- 4) The corresponding value
- 5) Trigger the given function

(3) Radio Button :-

```
from tkinter import *
root = Tk()
root.geometry("600x500")
def click():
    clicked = "You Selected the Option" + str(var.get())
    label.config(text=clicked, justify=LEFT)
var = StringVar()
l1 = Listbox()
l1.insert(1, "List 1")
l1.insert(2, "List 2")
l1.pack(anchor=S)
R1 = Radiobutton(root, text="Option 1", variable=var, value=1, command=click)
R1.pack(anchor=W)
R2 = Radiobutton(root, text="Option 2", variable=var, value=2, command=click)
R2.pack(anchor=W)
label = Label(root)
label.pack()
root.mainloop()
```

88

Output:



39

Step 6: Use the listbox() and insert option on the parent window along with pack() with specifying attribute.

Step 7: Now call the pack method for the corresponding radio obj so created and specify argument as an anchor attribute.

Step 8: Now define label object from the corresponding method and place it into the parents with window subsequently use the pack method for this widget and finally make use of mainloop method.

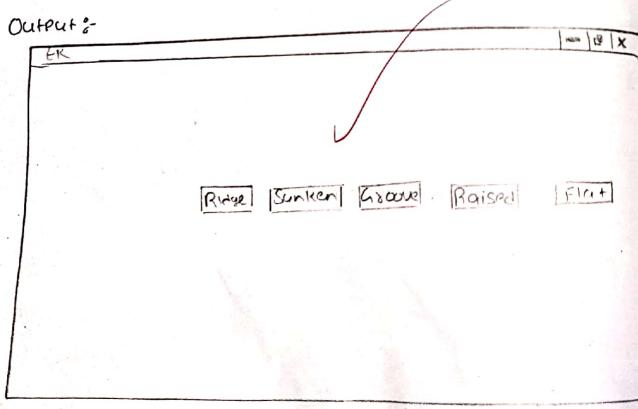
Dr. 211

Source Code :-

#Relief:

```
from tkinter import *
window = Tk()
B1 = Button(window, text="Flat", relief=FLAT)
B1.pack(side=RIGHT, padx=50)
B2 = Button(window, text="Raised", relief=Raised)
B2.pack(side=RIGHT, padx=50)
B3 = Button(window, text="Groove", relief=GROOVE)
B3.pack(side=RIGHT, padx=50)
B4 = Button(window, text="Sunken", relief=SUNKEN)
B4.pack(side=RIGHT, padx=50)
B5 = Button(window, text="Ridge", relief=RIDGE)
B5.pack(side=RIGHT, padx=50)
window.geometry("700x400")
window.mainloop()
```

Output :-



Practical - 5(c)

AIM:- GUI

#RELIEF

Algorithm:

Step 01:- Use the tkinter module to import the relevant method.

Step 02:- Use geometry to define the size of your parent window

Step 03:- After that use the button object with the following attributes, first the parent window, text attribute, relief.

Step 04:- Use the corresponding pack method for the respective button object and trigger the corresponding event.

Step 05:- Use the mainloop for triggering of the corresponding event.

MESSAGE BOX

Algorithm:-

Step01:- Import the relevant method from the tkinter library.

Step02:- Define a function and use the message box along with the different methods which contains one or more argument.

Step03:- The different option which are available in this method are Showinfo, Showwarning, Show error, Askyesno, Askquestion, Askokcancel.

Step04:- Create an object from the button method and place it on the parent window with the title of the button specified and corresponding event is called from triggering.

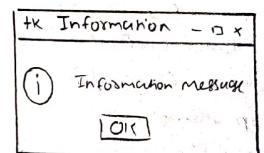
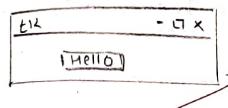
Step05:- Use the pack method to display the button widget and finally use the main loop method.

Step06:- Define object corresponding to parent window and use geometry method

Source code:

```
# Message box
import tkinter
root = Tk()
def function():
    tkMessageBox.showinfo("Information", "Information message")
    b1 = Button(root, text = "Hello", command = function)
    b2 = Pack()
    root.mainloop()
```

Output



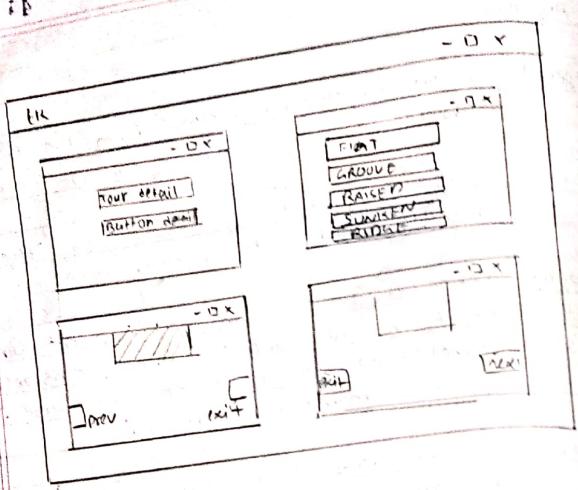
Source code:

```
from Tkinter import *
root = Tk()
root.minsize(500,500)
def main():
    top = Tk()
    top.config(bg="white")
    top.title("HOME")
    top.minsize(500,500)
    L = Label(top, text="SAN FRANCISCO\nPlaces of Interest,\nGolden Gate Bridge\nin Chinatown")
    L.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()
def second():
    top2 = Tk()
    top2.config(bg="orange")
    top2.title("About Us")
    top2.minsize(400,400)
    L = Label(top2, text="Created by: ABC\nFor more details contact to our official account")
    L.pack()
    b3 = Button(top2, text="prev", command=main)
    b3.pack(side=LEFT)
```

45

Algorithm:

- Step01: Import the relevant method from the `Tkinter` library along with parent window object declared.
- Step02: Use the parent window object along with the `minsize` function for window size.
- Step03: Define a function `main`, declare parent window object and use `config()`, `title()`, `minsize()`, `label()` as well as `button()` and use the `pack()`, `mainloop()` simultaneously.
- Step04: Similarly define the function `second` and use the attribute accordingly.
- Step05: Declare another function between along with parent object and declare button `b1` with attributes like `FLAT`, `RIDGE`, `GROOVE`, `Raised`, `Sunken` along with `relief` widget.
- Step06: Finally call the `mainloop()` for event driven programming.



```

46
b2 = Button (top2, text = "exit", command = terminade)
b2.pack (side = RIGHT)
top2.mainloop()

def button():
    top3 = Tk()
    top3.geometry ("500x500")
    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)
    b4.pack()
    b5 = Button (top3, text = "ridge button", relief = RIDGE)
    top3.mainloop()

def terminade():
    quit()

b5 = Button (root, text = "Tour DETAILS", command = main)
b5.pack()

b6 = Button (root, text = "Button DETAILS", command = button)
b6.pack()

root.mainloop()

```

Jan 2023

Source code :-

```
from tkinter import *
window = Tk()
window.title("THE IMAGE")
window.config()
leftframe = Frame(window, width=600, height=600)
leftframe.pack(side=LEFT)
rightframe = Frame(window, width=600, height=600)
rightframe.pack(side=RIGHT)
image = PhotoImage(file="xyz.gif")
scm = image.subsample(1, 2)
L1 = Label(rightframe, image=scm, bg="black")
L1.pack(side=LEFT)
def name():
    w1 = Tk()
def back():
    w1.withdraw()
    L2 = Label(w1, text="NAMES", bg="black", fg="white")
    L2.pack()
    l1 = Listbox(w1)
    l1.insert(1, "MESSI")
    l1.insert(2, "RONALDO")
    l1.insert(3, "PEPE")
    l1.pack()
    B1 = Button(w1, text="Back", command=back, bg="white", fg="black")
    B1.pack()
def click():
    w2 = Tk()
    def back():
        w2.withdraw()
```

47

PRACTICAL - 5(CD)

AIM :- GUI component

Algorithm :-

Step 01 :- Create an object corresponding to a parent window using the title method, config method, with a relevant library to be imported.

Step 02 :- Create a leftframe object and place it onto the parent window with height and width.

Step 03: Same create the rightframe object and placed it to parent window with width and height

Step 04: Now create image object from photo image method and import gif file and create one other image using subsample.

Step 05: Create label object placed onto the right frame and add the image object and use the pack method.

Step 06: Define function name Create A window again define the function for back and give label as ~~PLAYERS~~, NAME

Step 07: Use the listbox cmd insert values and create a button object for back command

Step 08:- Define a function age Create window again define function as back and use label object as many label object of your choice and use the button object for back operation

Step 09: Define Function gender Create window again define function as back and use label to display gender and use the button object for back command.

Step 10:- Define a function finish to terminate the program.

Step 11:- Create a button objects as PLAYERS, GENDER, AGE, and Exit use the sets commands Subsequently placing it onto corresponding

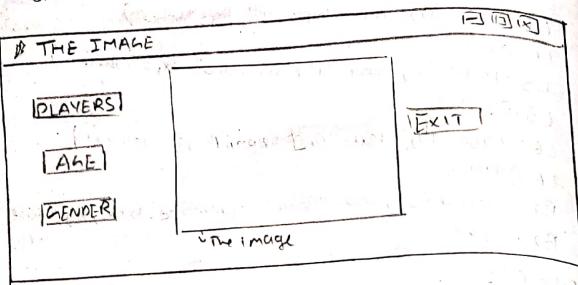
```

L3 = Label(w2, text = "AGE", bg="black", fg="Red")
L3.pack()
L4 = Label(w2, text = "MESS=35", bg="Yellow")
L4.pack()
L5 = Label(w2, text = "RONALDO=40", bg="white")
L5.pack()
L6 = Label(w2, text = "PEPE=45", bg="yellow")
L6.pack()
B2 = Button(w2, text = "Back", command = back, bg="white", fg="blue")
B2.pack()

def gender():
    w3 = Tk()
    def back():
        w3.withdraw()
    L7 = Label(w3, text = "GENDER", bg="black", fg="Red")
    L7.pack()
    L8 = Label(w3, text = "Male", bg="white")
    L8.pack()
    L9 = Label(w3, text = "Male", bg="white")
    L9.pack()
    B3 = Button(w3, text = "Back", command = back, bg="white", fg="blue")
    B3.pack()
    def finish():
        quit()
    B4 = Button(leftframe, text = "PLAYERS", command = name, bg="white")
    B4.pack(padx = "20", pady = "20")
    B5 = Button(leftframe, text = "GENDER", command = gender, bg="white", fg="black")
    B5.pack(padx = "20", pady = "20")
    B6 = Button(leftframe, text = "AGE", command = age, bg="white", fg="black")
    B6.pack(padx = "20", pady = "20")

```

```
B7=button (right-frame, text = "Exit", command = finish, bg = "white", fg = "black", font = "bold", height = 20, width = 50)
B7.pack (side = RIGHT)
window.mainloop()
```



TK - D X
NAMES
MESSI
RONALDO
PEPE
BACK

TK - D X
AGE
MESSI = 32
RONALDO = 40
PEPE = 45
BACK

TK - D X
GENDER
MALE
MALE
MALE
BACK

Step 12 8- Use the mainloop method with the Parent Object

JULY

PRACTICAL 5(E)

Aim:- GUI Components

Algorithm:

Step01:- Create an object from the Paned window method and use the pack method with fill and expand with orientation specified.

Step02:- Create an object from the label method and put it onto the paned window with the text attribute and use the add method to embed new objects.

Step03:- Now create another label object and place it onto the second panedwindow object, and add it onto second.

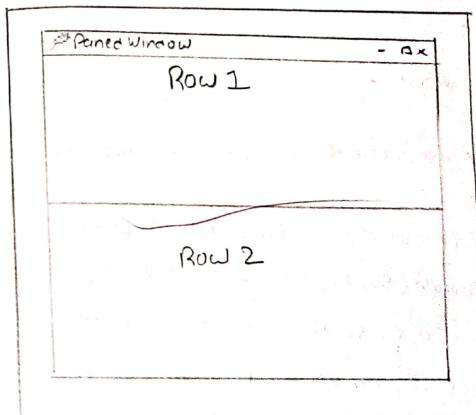
Step04:- Finally, use the trigger the mainloop method.

50

Source code:

```
from tkinter import *
window = Tk()
window.title ("Paned Window")
window.geometry ('100x100')
pane = PanedWindow(window, orient= VERTICAL)
pane.pack(fill=BOTH, expand=1)
row1 = Label(pane, text= "ROW 1", bg= "red")
pane.add(row1)
row2 = Label(pane, text= "ROW 2", bg= "blue")
pane.add(row2)
window.mainloop()
```

Output:

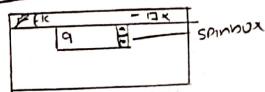


118.

Source code:

```
from tkinter import *
master = Tk()
s1 = Spinbox(master, from_=0, to=10)
s1.pack()
master.mainloop()
```

Output



Source code:

```
from tkinter import *
master = Tk()
w = Canvas(master, width=200, height=100, bg="blue")
w.pack()
w.create_oval(0, 100, 50, 80, fill="white", width=3)
w.create_rectangle(50, 20, 150, 80, fill="black", width=5)
w.create_oval(0, 0, 50, 20, fill="white", width=4)
w.create_line(150, 20, 200, 0, fill="red", width=10)
```

Algorithm :-

Step 01 :- Import a relevant method from the **tkinter** library

Step 02 : Create an object from the **tk** method and subsequently create an object from the **Spinbox** method

Step 03 :- Make the object so created on to the parent and trigger the corresponding

Step 04 : Use the **mainloop** method

Algorithm :-

Step 01 :- Import a relevant method from the **tkinter** library

Step 02 :- Create an object from the **Canvas** method and use the attribute, width, height, background color and the parent object.

Half
done

16

Step 03 :- Use the method to create circle, line, oval, rectangle along with the canvas object.

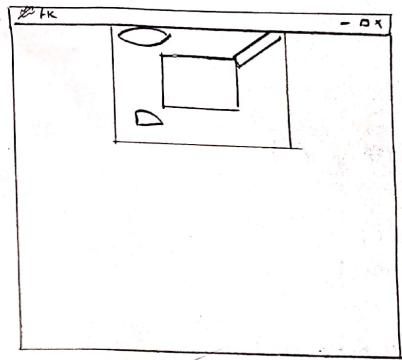
Step 04 :- Use the coordinate values to the every method.

Step 05 : Use the other method to create rectangle and call the pack method

Step 06 :- Use the mainloop method.

52

Output



PRACTICAL - 06

Aim:- Database Connectivity

Algorithm:- A]

Step 01 :- Import (DBM) library and use the open() for creating the database by specifying the name of the database along with the corresponding flag.

Step 02 :- Use the object so created for accessing the given website and corresponding regular name for website.

Step 03 :- Check whether given URL matches then Point Yes or else Point No if it does not match

Step 04 :- Use the close() method to terminate database library.

Source code :-

```
import sqlite3
import dbm
db=dbm.open("db1738", 'c')
db["abc"] = "def"
if db["abc"]!=None:
    print("Value Exists")
else:
    print("Value does not exists")
db.close()
```

Output:

Value exists

Algorithm:

Step01: Import corresponding library to make database connection, OS and SQL (sqlite3)

Step02: Now create connection object using sqlite3 library and connect for creating new database.

Step03: Now create cursor object using the cursor() from connection object created.

Step04: Now use execute() for creating the table with column name & data types

Step05: Now with cursor object use insert method for entering values corresponding fields with corresponding datatype

Step06: Use connection.commit() to complete the transaction using connection object.

Step07: Use execute statement algorithm cursor object for accessing values from the database using select from where clause

Source Code:-

```
import sqlite3
import dbm
connection = sqlite3.connect("employee.db")
cursor1 = connection.cursor()
cursor1.execute('create table Oracle, (empname char(20),
                           empid int(5))')
cursor1.execute('Insert into Oracle values ("Anakai", 01),
                ("Mane", 02), ("Prajanati", 03)')
connection.commit()
cursor1.execute("select empname from Oracle")
a = cursor1.fetchall()
print(a)
cursor1.close()
```

Output:-

[('Anakai'), ('Mane'), ('Prajanati')]

Dots

Python Project :-

#FLYBEE AIRLINES!

```

from Tkinter import *
import tkMessageBox
from ttk import *
import sqlite3
con=sqlite3.Connection('hrdb')
rootp=Tk()
rootp.title("FLYBEE AIRLINES")
rootp.geometry("600x500")
rootp.config("ivory2")
Label(rootp,text="FLYBEE..",font="Calibri 30").grid(row=1,column=2)
def fun9():
    root4=Tk()
    root4.title("Welcome,Search flights")
    root4.config()
    def back1():
        root4.withdraw()
        rootp.iconify()
        rootp.deiconify()
    Label(root4,text="Enter Boarding").grid(row=0,column=0,padx="10",pady="10")
    w1=Combobox(root4,height=5,width=15,values=["New York","Chicago","Dallas","San Francisco"])
    w1.grid(row=0,column=1)
    Label(root4,text="Select Destination").grid(row=1,column=0,padx="10",pady="10")
    w2=Combobox(root4,height=5,width=15,values=["New York","Chicago","Dallas","San Francisco"])
    w2.grid(row=1,column=1)

```

```

Label(root4,text="Choose Day Of Travel").grid(row=2,column=0,padx="10",pady="10")
w3=Combobox(root4,text="Choose Day",height=5,width=15,values=["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"])
w3.grid(row=2,column=1)

def fun10():
    a=w1.get()
    b=w2.get()
    c=w3.get()
    cur=con.cursor()
    if a==" " or b==" " or c==" ":
        tkMessageBox.showerror("Error","Cant leave any field empty")
    else:
        if a==b:
            cur.execute("create table eco(boarding char(20),destination char(20),day char(20),time number,class char(10),fare number)")
            cur.execute("insert into eco values('New York','Chicago','Sunday',1.00,'Economic',2500)")
            cur.execute("insert into eco values('New York','Dallas','Monday',1.00,'Common',4000)")
            cur.execute("insert into eco values('New York','San Francisco','Tuesday',1.00,'Economic',5500)")
            cur.execute("insert into eco values('Chicago','New York','Wensday',1.00,'Economic',3500)")
            cur.execute("insert into eco values('Chicago','New York','Wensday',7.00,'Common',2500)")
            cur.execute("select * from eco where boarding=? and destination=? and day=?",(a,b,c)) 
            con.commit()
            e=cur.fetchall()

```

```

tkMessageBox.showinfo("Info","Flights Are Available!")
else:
    tkMessageBox.showerror("Oops","Boarding And Destination Can't Be Same!")

Bs=Button(root4,text="Search",command=fun10).grid(row=3,column=0)
Button(root4,text="Back",command=back1).grid(row=3,column=1)
rootp.withdraw()
root4.mainloop()

def fun5():
    root=Tk()
    root.title('Flight Booking')
    def back2():
        root.withdraw()
        rootp.iconify()
        rootp.deiconify()
    Label(root,text="Enter Boarding").grid(row=1,column=0,padx="10",pady="10")
    w=Combobox(root,height=5,width=15,values=["Dallas","Washington","New York","San Francisco"])
    w.grid(row=1,column=1)
    Label(root,text="Enter Destination").grid(row=2,column=0,padx="10",pady="10")
    w1=Combobox(root,height=5,width=15,values=["Dallas","Washington","New York","San Francisco"])
    w1.grid(row=2,column=1)
    Label(root,text="Enter Last 4 Digit Of Your Citizenship Number").grid(row=3,column=0,padx="10",pady="10")
    e=Entry(root,width=20)
    e.grid(row=3,column=1)

```

```

57
w2=Combobox(root,text="BusinessClass",height=5,width=15,values=["BusinessClass","Economic"])
w2.grid(row=4,column=1,padx="10",pady="10")
Label(root,text="Choose Class").grid(row=4,column=0,padx="10",pady="10")
Label(root,text="Choose Day Of Travel").grid(row=5,column=0,padx="10",pady="10")
w3=Combobox(root,text="Choose Day",height=5,width=15,values=["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"])
w3.grid(row=5,column=1,padx="10",pady="10")
Label(root,text="Choose Time Of Your Flight").grid(row=6,column=0,padx="10",pady="10")
w4=Combobox(root,height=5,width=15,values=["1:00 AM","7:00 AM","1:00 PM","4:00 PM","9:00 PM"])
w4.grid(row=6,column=1,padx="10",pady="10")
def fun():
    root.withdraw()
    a=w.get()
    b=w1.get()
    c=e.get()
    d=w2.get()
    f=w3.get()
    g=w4.get()
    x={a,b,c,f,g}
    cur=con.cursor()

    if a=="" or b=="" or c=="" or d=="" or f=="" or g=="":
        tkMessageBox.showerror("OOPS","You Can't Leave Any Field Empty!")
    else:
        if d=='Economic':

```

```

58
if a==b:
    cur.execute("create table economic2(boarding char(20),destination char(20),adno
number,day char,time number)")
    cur.execute("Insert Into economic2 values(?, ?, ?, ?, ?)",x)
    tkMessageBox.showinfo("congrats","your seat has been reserved")
    con.commit()
    cur.execute("select * from economic2 where adno=(?)",c)
    tkMessageBox.showinfo("records",cur.fetchall())
    r1=Tk()
    r1.title("Booking Successful")
    r1.geometry("300x300")
    def ok1():
        r1.withdraw()
        rootp.iconify()
        rootp.deiconify()
    Label(r1,text="Booking Successfull").grid(row=1,column=2,padx="20",pady="20")
    Label(r1,text="-----").grid(row=2,column=1)
    Label(r1,text="-----").grid(row=2,column=2)
    Label(r1,text="-----").grid(row=2,column=3)
    cur.execute("select boarding,destination from economic2 where adno=(?)",c)
    Label(r1,text=w.get()+"=>"+w1.get()).grid(row=3,column=2,padx="20",pady="20")
    cur.execute("select adno from economic2 where adno=(?)",c)
    Label(r1,text="4 Digit Citizenship
No:"+e.get()).grid(row=4,column=2,padx="20",pady="20")
    cur.execute("select day,time from economic2 where adno=(?)",c)

```

```

Label(r1,text="On "+w3.get()+" "+w4.get()).grid(row=5,column=2,padx="20",pady="20")
Label(r1,text="-----").grid(row=6,column=1)
Label(r1,text="-----").grid(row=6,column=2)
Label(r1,text="-----").grid(row=6,column=3)
Button(r1,text="OK",command=ok1).grid(row=7,column=2)
r1.mainloop()

else:
    tkMessageBox.showerror("Error","You Can't Choose Same City")

if d=='BusinessClass':
    #cur.execute("create table common2(boarding char(20),destination char(20),adno
    number,day char,time number)")

    if a==b:
        cur.execute("insert into common2 values(?,?,?,?,?)",x)
        tkMessageBox.showinfo("Congratulations!","Your Seat Has Been Reserved!")

        con.commit()
        r2=Tk()
        r2.title("Booking Successful")
        r2.geometry("300x300")
        def ok2():
            r2.withdraw()
            rootp.iconify()
            rootp.deiconify()
        Label(r2,text="Booking Successful").grid(row=1,column=2,padx="20",pady="20")
        Label(r2,text="-----").grid(row=2,column=1)
        Label(r2,text="-----").grid(row=2,column=2)

    else :
        tkMessageBox.showerror("Error","You Can't Choose Same City")

```

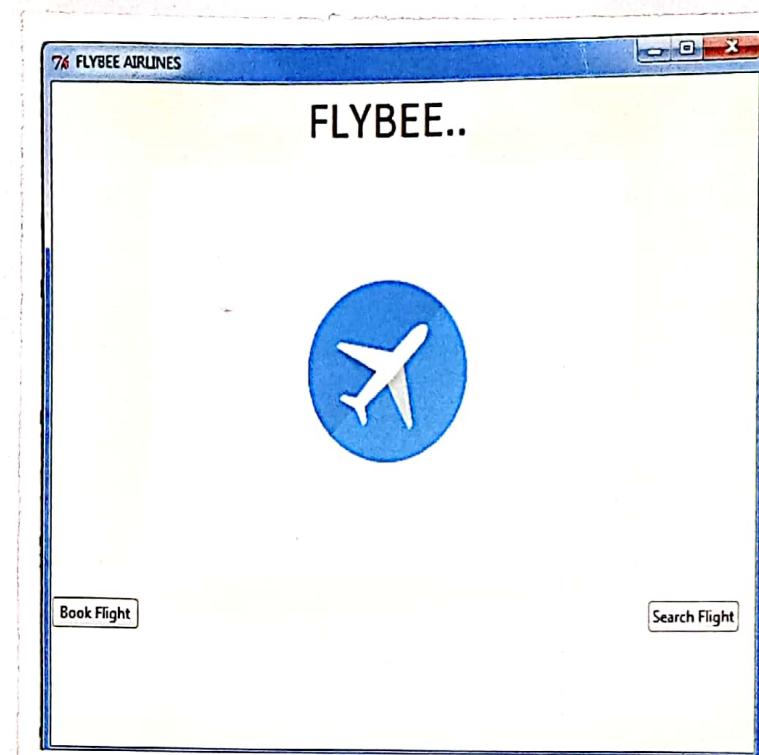
```

Label(r2,text="-----").grid(row=2,column=3)
cur.execute("select boarding,destination from common2 where adno=(?),(c,)")
Label(r2,text=w.get()+">"+w1.get()).grid(row=3,column=2,padx="20",pady="20")
cur.execute("select adno from common2 where adno=(?),(c,)")
Label(r2,text="4 Digit Citizenship
No:"+e.get()).grid(row=4,column=2,padx="20",pady="20")
cur.execute("select day,time from common2 where adno=(?),(c,)")
Label(r2,text="On "+w3.get()+" "+w4.get()).grid(row=5,column=2,padx="20",pady="20")
Label(r2,text="-----").grid(row=6,column=1)
Label(r2,text="-----").grid(row=6,column=2)
Label(r2,text="-----").grid(row=6,column=3)
Button(r2,text="OK",command=ok2).grid(row=7,column=2)
r2.mainloop()

else :
    tkMessageBox.showerror("Error","You Can't Choose Same City")
Button(root,text="Insert",command=fun).grid(row=7,column=1)
Button(root,text="Back",command=Lack2).grid(row=7,column=2)
rootp.withdraw()
root.mainloop()
img1=PhotoImage(file="flight.gif")
sub=img1.subsample(2,2)
Label(rootp,text="",image=sub).grid(row=2,column=2,padx="10",pady="10")
B2=Button(rootp,text="Search Flight",command=fun9).grid(row=3,column=3,padx="10",pady="10")
B3=Button(rootp,text="Book Flight",command=fun5).grid(row=3,column=1)
rootp.mainloop()

```

08
OUTPUT :-



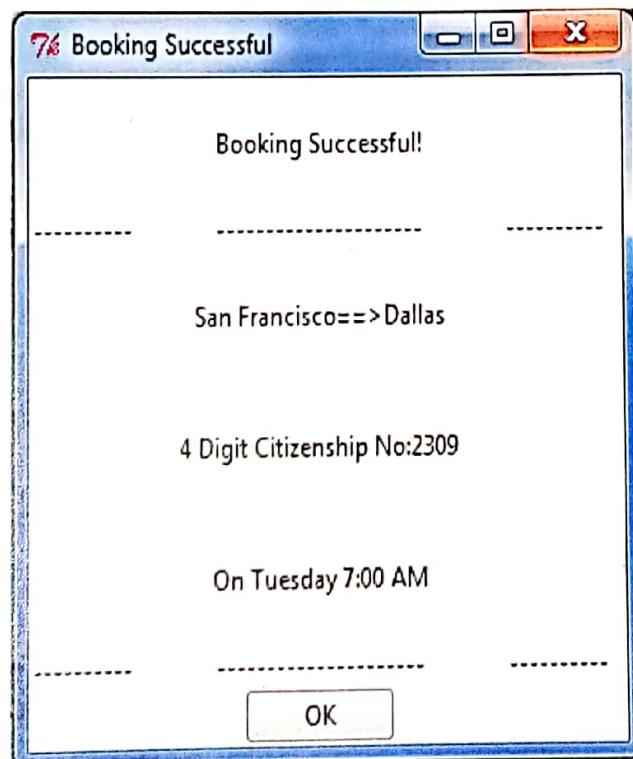
The screenshot shows a window titled "Flight Booking" with the following fields:

- Enter Boarding: A dropdown menu.
- Enter Destination: A dropdown menu.
- Enter Last 4 Digit Of Your Citizenship Number: An input field.
- Choose Class: A dropdown menu.
- Choose Day Of Travel: A dropdown menu.
- Choose Time Of Your Flight: A dropdown menu.

At the bottom right are "Insert" and "Back" buttons.

74 Flight Booking

Enter Boarding	San Francisco
Enter Destination	Dallas
Enter Last 4 Digit Of Your Citizenship Number	2309
Choose Class	BusinessClass
Choose Day Of Travel	Tuesday
Choose Time Of Your Flight	7:00 AM
<input type="button" value="Insert"/> <input type="button" value="Back"/>	



18



A blue-bordered window titled "76 Welcome, Search flights" in red. It has three input fields: "Enter Boarding" (with a dropdown arrow), "Select Destination" (with a dropdown arrow), and "Choose Day Of Travel" (with a dropdown arrow). At the bottom are two buttons: "Search" on the left and "Back" on the right.

