Unit-4

Network Programming and GUI using Python

26. Get IP Address of the PC

```
1  # Python Program to Get IP Address
2  import socket
3  hostname = socket.gethostname()
4  IPAddr = socket.gethostbyname(hostname)
5
6  print("Your Computer Name is: " + hostname)
7  print("Your Computer IP Address is: " + IPAddr)
```

```
Output:
     Your Computer Name is: BCA-Laptop
     Your Computer IP Address is: 192.168.160.1
```

27. Download a Web Page from Internet

```
1  import bs4
2  import urllib.request
3
4  url="https://www.tutorialspoint.com/python/python_networking.htm"
5  webpage=str(urllib.request.urlopen(url).read())
6  soup = bs4.BeautifulSoup(webpage)
7
8  print(soup.get_text())
```

```
Output:
```

28. Download an Image from Internet

```
1  import requests # request img from web
2  import shutil # save img locally
3
4  url = input('Please enter an image URL (string):') #prompt
5  user for img url
6  file_name = input('Save image as (string):') #prompt user
7  for file_name
8
9  res = requests.get(url, stream = True)
10
11 if res.status_code == 200:
12     with open(file_name,'wb') as f:
13         shutil.copyfileobj(res.raw, f)
14     print('Image sucessfully Downloaded: ',file_name)
15 else:
16     print('Image Couldn\'t be retrieved')
```

```
Output:
```

29. Communication between Client/Server

```
Server.py
----------
# first of all import the socket library
import socket

# next create a socket object
s = socket.socket()
print ("Socket successfully created")

# reserve a port on your computer in our
# case it is 40674 but it can be anything
port = 40674

# Next bind to the port
# we have not typed any ip in the ip field
# instead we have inputted an empty string
# this makes the server listen to requests
# coming from other computers on the network
s.bind(('', port))
print ("socket binded to %s" %(port))

# put the socket into listening mode
s.listen(5)
print ("socket is listening")

# a forever loop until we interrupt it or
# an error occurs
while True:

     # Establish connection with client.
     c, addr = s.accept()
     print ('Got connection from', addr )

     # send a thank you message to the client.
     c.send(b'Thank you for connecting')

     # Close the connection with the client
     c.close()
```

```
Client.py
---------
# Import socket module
import socket

# Create a socket object
s = socket.socket()

# Define the port on which you want to connect
port = 40674

# connect to the server on local computer
s.connect(('127.0.0.1', port))
```

```
12
13 # receive data from the server
14 print(s.recv(1024))
15
16 # close the connection
17 s.close()
```

```
Output:
  Serer:
     Socket successfully created
     socket binded to 40674
     socket is listening
     Got connection from ('127.0.0.1', 4259)

  Client:
     b'Thank you for connecting'
```

30. GUI Pack() layout

```
1  # TkInter example
2
3  import tkinter
4  from functools import partial
5  from tkinter import messagebox
6
7  top = tkinter.Tk()
8  top.title("Hello")
9  top.geometry("300x200")
10
11
12 def helloCallBack(x):
13     messagebox.showinfo("MsgBox", x.get())
14     var.set(x.get())
15
16 var = tkinter.StringVar()
17 label = tkinter.Label(top, textvariable=var)
18 var.set("Welcome to GUI in Python")
19 label.pack()
20
21 #Add Entry
22 var2 = tkinter.StringVar()
23 E1 = tkinter.Entry(top, textvariable=var2, bd=5)
24 E1.pack()
25
26 #Add Button1
27 helloCallBack = partial(helloCallBack, var2)
28 B1       =       tkinter.Button(top,       text="Click",
   command=helloCallBack)
29 B1.pack()
30
31 #Add Button2
32 B2       =       tkinter.Button(top,       text="Close",
   command=top.destroy)
33 B2.pack()
```

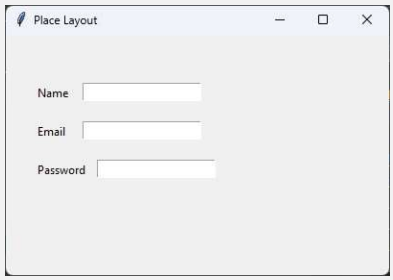
```
34
35 top.mainloop()
```

Output:

31. GUI Place() layout

```
1  from tkinter import *
2  top = Tk()
3  top.geometry("400x250")
4  name = Label(top, text = "Name").place(x = 30,y = 50)
5  email = Label(top, text = "Email").place(x = 30, y = 90)
6  password = Label(top, text = "Password").place(x = 30, y =
   130)
7  e1 = Entry(top).place(x = 80, y = 50)
8  e2 = Entry(top).place(x = 80, y = 90)
9  e3 = Entry(top).place(x = 95, y = 130)
10 top.mainloop()
```

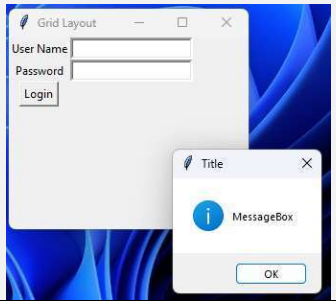
Output:



32. GUI Grid() layout

```
1  from tkinter import *
2  from tkinter import messagebox
3
4  def hello():
5      messagebox.showinfo("Title", "MessageBox")
6
7  top = Tk()
8  top.title("Grid Layout")
9  top.geometry("250x200")
10 L1 = Label(top, text="User Name").grid(row=1,column=1)
11 E1 = Entry(top, bd =3).grid(row=1,column=2)
12 L2 = Label(top, text="Password").grid(row=2,column=1)
13 E2 = Entry(top, bd =3).grid(row=2,column=2)
14 B    =    Button(top,    text    ="Login",    command    =
15 hello).grid(row=3,column=1)
16 top.mainloop()
```

Output:



Unit-5

Connecting with Database

33. Verify Database Connection with MySQL

```
import mariadb as MySQLdb

# Open database connection
db  =  MySQLdb.connect(  host="localhost",  user="root",
password="", database="college")
# prepare a cursor object using cursor() method
cursor = db.cursor()
# execute SQL query using execute() method.
cursor.execute("SELECT VERSION()")
# Fetch a single row using fetchone() method.
data = cursor.fetchone()
print("Database version : %s " % data)
print(db)
print(cursor)
# disconnect from server
db.close()
```

```
Output:
      Database version : 10.1.38-MariaDB
      <mariadb.connection connected to 'localhost' at 0000029CD62402F0>
      <mariadb.cursor at 0000029CD64B1FD0>
```

34. List Database

```
import mariadb as MySQLdb
myconn  =  MySQLdb.connect(host="localhost",  user="root",
passwd="")
cur = myconn.cursor()
dbs = cur.execute("show databases")
for x in cur:
    print(x)
myconn.close()
```

```
Output:
      ('college',)
      ('information_schema',)
      ('library',)
      ('mydatabase',)
      ('mysql',)
      ('performance_schema',)
      ('phpdemo',)
      ('phpmyadmin',)
      ('test',)
```

35. Create Database

```
import mariadb as MySQLdb

# establishing the connection
conn   =   MySQLdb.connect(host="localhost",   user="root",
password="")
# Creating a cursor object using the cursor() method
cursor = conn.cursor()
# Doping database MYDATABASE if already exists.
cursor.execute("DROP database IF EXISTS MyDatabase")
```

```
# Preparing query to create a database
sql = "CREATE database MYDATABASE"
# Creating a database
cursor.execute(sql)
# Retrieving the list of databases
print("List of databases: ")
cursor.execute("SHOW DATABASES")
print(cursor.fetchall())
# Closing the connection
conn.close()
```

```
Output:
      ('college',)
      ('information_schema',)
      ('library',)
      ('mydatabase',)
      ('mysql',)
      ('performance_schema',)
      ('phpdemo',)
      ('phpmyadmin',)
      ('test',)
```

36. Create Table

```
import mariadb as MySQLdb

db  =  MySQLdb.connect(  host="localhost",  user="root",
password="", database="mydatabase" )
cursor = db.cursor()
cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")
sql = """CREATE TABLE EMPLOYEE (
         FIRST_NAME  CHAR(20) NOT NULL,
         LAST_NAME  CHAR(20),
         AGE INT,
         INCOME FLOAT )"""
cursor.execute(sql)
db.close()
```

```
Output:
```

37. Insert Records into Table

```
import mariadb as MySQLdb

db   =   MySQLdb.connect(host="localhost",   user="root",
password="", database="mydatabase" )
cursor = db.cursor()
sql = """INSERT INTO EMPLOYEE(FIRST_NAME,
         LAST_NAME, AGE, INCOME)
         VALUES ('Malay', 'Dave', 42, 5000)"""
try:
    cursor.execute(sql)
    db.commit()
except:
    db.rollback()
```

```
db.close()
```

```
Output:
```

38. Display Records from Table

```
import mariadb as MySQLdb

myconn  =  MySQLdb.connect(host="localhost",  user="root",
passwd="", database="mydatabase")
cur = myconn.cursor()
try:
    cur.execute("select * from employee")
    result = cur.fetchall()
    for x in result:
        print(x)
except:
    myconn.rollback()
myconn.close()
```

```
Output:
     ('Mohit', 'Rank', 22, 4500.0)
     ('Malay', 'Dave', 42, 5000.0)
```

39. Update Records into Table

```
import mariadb as MySQLdb

# Open database connection
db   =   MySQLdb.connect(host="localhost",   user="root",
passwd="", database="mydatabase")

# prepare a cursor object using cursor() method
cursor = db.cursor()

# Prepare SQL query to UPDATE required records
sql = "UPDATE EMPLOYEE SET AGE = AGE + 1
                           WHERE SEX = '%c'" % ('M')
try:
   # Execute the SQL command
   cursor.execute(sql)
   # Commit your changes in the database
   db.commit()
except:
   # Rollback in case there is any error
   db.rollback()

# disconnect from server
db.close()
```

```
Output:
```

40. Delete Records from Table

```
import mariadb as MySQLdb

# Open database connection
db   =   MySQLdb.connect(host="localhost",   user="root",
passwd="", database="mydatabase")

# prepare a cursor object using cursor() method
cursor = db.cursor()

# Prepare SQL query to DELETE required records
sql = "DELETE FROM EMPLOYEE WHERE AGE > '%d'" % (20)
try:
   # Execute the SQL command
   cursor.execute(sql)
   # Commit your changes in the database
   db.commit()
except:
   # Rollback in case there is any error
   db.rollback()

# disconnect from server
db.close()
```

```
Output:
```