TO-DO LIST USING PYTHON AND SQLITE3

A Mini Project report submitted in partial requirements for Second Year in Computer Engineering

by

69 Sarthak Tukan (171061070)
72 Akash Yadav (171041049)
74 Sapana Pawar(171041035)

Under the guidance of
Mrs. S. P. Bansu.

Department of Computer Engineering

A. C. Patil College of Engineering, Kharghar, Navi Mumbai
University of Mumbai
2018-2019

TABLE OF CONTENTS

Index

1.ABSTRACT :

To-Do List is software developed for daily working human like school & college students, company employee, etc. It facilitates to access daily work schedule of an individual. The information can be fed up and he/she can view their schedules on a particular dates as well. This app will also help in check marking those tasks which are completed by that individual.

It facilitates the individual adding new task or without entering previous tasks, check mark task directly. It also facilitates viewing whole schedule of any person.

It also facilitates deletion of task from list by an individual by any reason. If an individual wants to see the number of tasks, that function is also available here.

2.INTRODUCTION :

Today every other person want to make sure that they do their jobs/tasks on time. In the present application, all work is done on computer itself. The whole day tasks are stored in register and at each end of the task the person can checkmark it. At the end of all jobs the person can insert or plan new day schedule.

To make sure you plan your schedule properly, so we have created python application based on GUI and Database.

- To-do List is used to create reminders or update daily routine work or chores.
- It helps us to stay organized and keep track of our assigned work
- To-do list can be implemented using any of the known programming languages.

3.OBJECTIVE:

The objective is to be able to see and stay updated in our day to day life. Also enter daily routine work.

a.Problem Definition

In this program we can add tasks, check-off the tasks which we have completed and also delete the tasks which have been completed or which are going to be cancelled.

b.System Requirements

Hardware Configuration

Processor: i5

RAM: 4GB

Modem:Broadband

Software Configuration

Operating System: Windows

Web Server:127.0.0.1:5000

Front end: tkinter

Scripts: SQL

Language: Python

Back End: Database

4.Technologies:

To-do list using python:

- To-do list is implemented in Python 3.7.2.
- To-do List is a GUI Application and uses Tkinter for the implementation of the GUI.
- Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python.

5.Implementation

System Overview

- Sqlite3 is used for the database connectivity.
- Sqlite3 is a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.
- SQLiteStudio is used to view,edit and create the database table.

Program:

```
from tkinter import *
from tkcalendar import *
import sqlite3
conn=sqlite3.connect("todo1.db")
c=conn.cursor()
tasks = []
v1=""
def example1():
    def print_sel():
        global v1
        v1=cal.selection_get()
        top.destroy()

    #root=Tk()
    top = Toplevel(root)

    cal = Calendar(top, font="Arial 14", selectmode='day', locale='en_US',
                   cursor="hand1", year=2019, month=4, day=10)

    cal.pack(fill="both", expand=True)
    Button(top, text="ok", command=print_sel).pack()
    top.mainloop()
    #root.mainloop()

def theme():
   def vio():
      b1.configure(bg="#8153E9")
      b2.configure(bg="#8153E9")
      b3.configure(bg="#8153E9")
      b4.configure(bg="#8153E9")
      b5.configure(bg="#8153E9")
      b6.configure(bg="#8153E9")
      b7.configure(bg="#8153E9")
      b9.configure(bg="#8153E9")
      b1.configure(fg="white")
      b2.configure(fg="white")
      b3.configure(fg="white")
      b4.configure(fg="white")
      b5.configure(fg="white")
      b6.configure(fg="white")
      b7.configure(fg="white")
      b9.configure(fg="white")
      top2.destroy()
   def yel():
      b1.configure(bg="#FFC547")
      b2.configure(bg="#FFC547")
      b3.configure(bg="#FFC547")
      b4.configure(bg="#FFC547")
      b5.configure(bg="#FFC547")
      b6.configure(bg="#FFC547")
      b7.configure(bg="#FFC547")
      b9.configure(bg="#FFC547")
      b1.configure(fg="black")
      b2.configure(fg="black")
      b3.configure(fg="black")
      b4.configure(fg="black")
      b5.configure(fg="black")
      b6.configure(fg="black")
      b7.configure(fg="black")
      b9.configure(fg="black")
      top2.destroy()
   def org():
      b1.configure(bg="#FF9447")
      b2.configure(bg="#FF9447")
      b3.configure(bg="#FF9447")
      b4.configure(bg="#FF9447")
      b5.configure(bg="#FF9447")
      b6.configure(bg="#FF9447")
      b7.configure(bg="#FF9447")
      b9.configure(bg="#FF9447")
      b1.configure(fg="black")
      b2.configure(fg="black")
      b3.configure(fg="black")
      b4.configure(fg="black")
      b5.configure(fg="black")
      b6.configure(fg="black")
      b7.configure(fg="black")
      b9.configure(fg="black")
      top2.destroy()
   def rd():
      b1.configure(bg="#FF4947")
      b2.configure(bg="#FF4947")
      b3.configure(bg="#FF4947")
      b4.configure(bg="#FF4947")
      b5.configure(bg="#FF4947")
      b6.configure(bg="#FF4947")
      b7.configure(bg="#FF4947")
      b9.configure(bg="#FF4947")
      b1.configure(fg="white")
      b2.configure(fg="white")
      b3.configure(fg="white")
      b4.configure(fg="white")
      b5.configure(fg="white")
      b6.configure(fg="white")
      b7.configure(fg="white")
      b9.configure(fg="white")
      top2.destroy()
   def grn():
      b1.configure(bg="#25E173")
      b2.configure(bg="#25E173")
      b3.configure(bg="#25E173")
      b4.configure(bg="#25E173")
      b5.configure(bg="#25E173")
      b6.configure(bg="#25E173")
      b7.configure(bg="#25E173")
      b9.configure(bg="#25E173")
      b1.configure(fg="black")
      b2.configure(fg="black")
      b3.configure(fg="black")
      b4.configure(fg="black")
      b5.configure(fg="black")
      b6.configure(fg="black")
      b7.configure(fg="black")
      b9.configure(fg="black")
      top2.destroy()
   def blu():
      b1.configure(bg="#1E90FF")
      b2.configure(bg="#1E90FF")
      b3.configure(bg="#1E90FF")
      b4.configure(bg="#1E90FF")
      b5.configure(bg="#1E90FF")
      b6.configure(bg="#1E90FF")
      b7.configure(bg="#1E90FF")
      b9.configure(bg="#1E90FF")
      b1.configure(fg="black")
      b2.configure(fg="black")
      b3.configure(fg="black")
      b4.configure(fg="black")
      b5.configure(fg="black")
      b6.configure(fg="black")
      b7.configure(fg="black")
      b9.configure(fg="black")
      top2.destroy()
   def bur():
      b1.configure(bg="#DB086B")
      b2.configure(bg="#DB086B")
      b3.configure(bg="#DB086B")
      b4.configure(bg="#DB086B")
      b5.configure(bg="#DB086B")
      b6.configure(bg="#DB086B")
      b7.configure(bg="#DB086B")
      b9.configure(bg="#DB086B")
      b1.configure(fg="white")
      b2.configure(fg="white")
      b3.configure(fg="white")
      b4.configure(fg="white")
      b5.configure(fg="white")
      b6.configure(fg="white")
      b7.configure(fg="white")
      b9.configure(fg="white")
      top2.destroy()
```

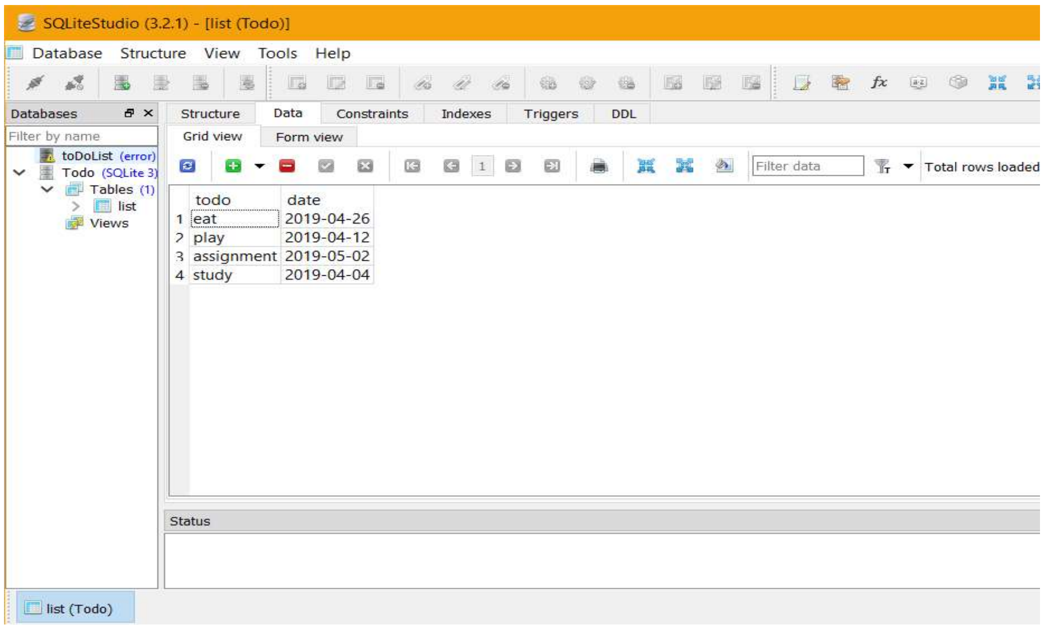
```
def blk():
      b1.configure(bg="black")
      b2.configure(bg="black")
      b3.configure(bg="black")
      b4.configure(bg="black")
      b5.configure(bg="black")
      b6.configure(bg="black")
      b7.configure(bg="black")
      b9.configure(bg="black")
      b1.configure(fg="white")
      b2.configure(fg="white")
      b3.configure(fg="white")
      b4.configure(fg="white")
      b5.configure(fg="white")
      b6.configure(fg="white")
      b7.configure(fg="white")
      b9.configure(fg="white")
      top2.destroy()
   def wht():
      b1.configure(bg="white")
      b2.configure(bg="white")
      b3.configure(bg="white")
      b4.configure(bg="white")
      b5.configure(bg="white")
      b6.configure(bg="white")
      b7.configure(bg="white")
      b9.configure(bg="white")
      b1.configure(fg="black")
      b2.configure(fg="black")
      b3.configure(fg="black")
      b4.configure(fg="black")
      b5.configure(fg="black")
      b6.configure(fg="black")
      b7.configure(fg="black")
      b9.configure(fg="black")
   top2.destroy()
   top2 = Toplevel(root)
   top2.geometry("150x150")
       b11=Button(top2,bg="#8153E9",command=vio)
       b12=Button(top2,bg="#FFC547",command=yel)
       b13=Button(top2,bg="#FF9447",command=org)
       b21=Button(top2,bg="#FF4947",command=rd)
       b22=Button(top2,bg="#25E173",command=grn)
       b23=Button(top2,bg="#1E90FF",command=blu)
       b31=Button(top2,bg="#DB086B",command=bur)
       b32=Button(top2,bg="black",command=blk)
       b33=Button(top2,bg="white",command=wht)
       b11.place(x=0,y=0,height=50,width=50)
       b12.place(x=0,y=50,height=50,width=50)
       b13.place(x=0,y=100,height=50,width=50)
       b21.place(x=50,y=0,height=50,width=50)
       b22.place(x=50,y=50,height=50,width=50)
```

```
  b23.place(x=50,y=100,height=50,width=50)
  b31.place(x=100,y=0,height=50,width=50)
  b32.place(x=100,y=50,height=50,width=50)
  b33.place(x=100,y=100,height=50,width=50)
  top2.mainloop()

def update():
  clearlb()
  tasksel=c.execute("SELECT todo from list")
  for task in tasksel:
    lb.insert("end",task)
  #conn.commit()

def inrecord():

  txt=t1.get()
  if txt !="":
    tasks.append(txt)
    c.execute("insert into list values (?,?)",(t1.get(),v1,))
    update()
  else:
    l2["text"]="Please Enter a Task"

  #c.execute("insert into list values (?)",(t1.get(),))
  t1.delete(0,"end")
  #c.execute("insert into list values('sleep')")
  conn.commit()

def clearlb():
  lb.delete(0,"end")

def del_one():
  txt = lb.get("active")
  #print(txt[0])
  c.execute("delete from list where todo = (?)",(txt[0],))
  conn.commit()
  update()

def clear():
  global tasks
  tasks=[]
  c.execute("delete from list")
  conn.commit()
  update()

def sort_asc():
  clearlb()
  tasksel=c.execute("SELECT * from list order by date")
```

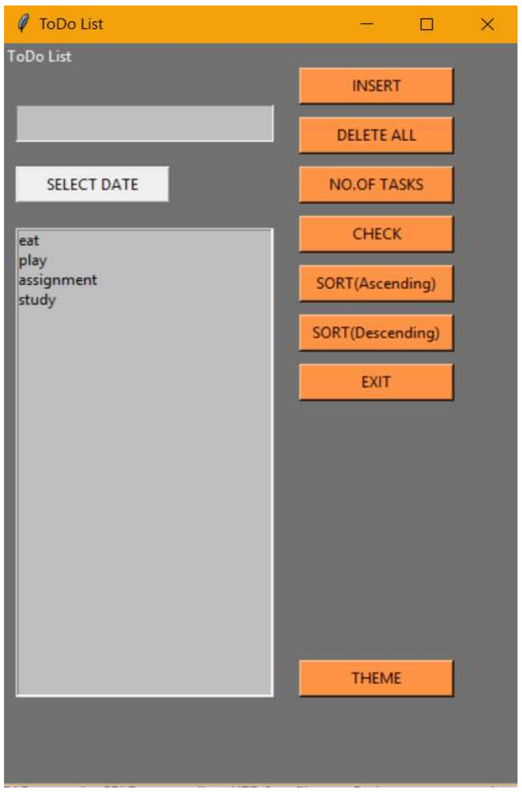
```
#tasksel=c.execute("SELECT todo from list")
  for task in tasksel:
    lb.insert("end",task)

def sort_desc():
  clearlb()
  tasksel=c.execute("SELECT * from list order by date desc")
  #=c.execute("SELECT todo from list")
  for task in tasksel:
    lb.insert("end",task)

def notasks():
  v=c.execute("SELECT COUNT( *) from list")
  for i in v:
    va=i
  l2["text"]=("Number of tasks:",va)

  #no_of_tasks=len(tasks)
  #msg="Number of Tasks: %s" %no_of_tasks
  #l2["text"]=msg


root = Tk()
root.configure(bg="#717171")
root.geometry("400x600")
root.title("ToDo List")
lb=Listbox(root)
tasksel=c.execute("SELECT todo from list")
for task in tasksel:
  lb.insert("end",task)
l1=Label(root,text="ToDo List")
l2=Label(root,text="")
t1=Entry(root)
t2=Entry(root)
b1=Button(root,text="EXIT",command=root.destroy)
b2=Button(root,text="INSERT",command=inrecord)
b3=Button(root,text="DELETE ALL",command=clear)
b4=Button(root,text="CHECK",command=del_one)
b5=Button(root,text="SORT(Ascending)",command=sort_asc)
b6=Button(root,text="SORT(Descending)",command=sort_desc)
b7=Button(root,text="NO.OF TASKS",command=notasks)
b8=Button(root,text="SELECT DATE",command=example1)
b9=Button(root,text="THEME",command=theme)
l1.grid(row=1,column=1)
l2.grid(row=2,column=1)
t1.place(x=10,y=50,height=30,width=200)
b8.place(x=10,y=100,height=30,width=120)
lb.place(x=10,y=150,height=380,width=200)
b2.place(x=230,y=20,height=30,width=120)
b7.place(x=230,y=100,height=30,width=120)
b3.place(x=230,y=60,height=30,width=120)
```

```
b4.place(x=230,y=140,height=30,width=120)
b5.place(x=230,y=180,height=30,width=120)
b6.place(x=230,y=220,height=30,width=120)
b1.place(x=230,y=260,height=30,width=120)
b9.place(x=230,y=500,height=30,width=120)
b1.configure(bg="#8153E9")
b2.configure(bg="#8153E9")
b3.configure(bg="#8153E9")
b4.configure(bg="#8153E9")
b5.configure(bg="#8153E9")
b6.configure(bg="#8153E9")
b7.configure(bg="#8153E9")
b9.configure(bg="#8153E9")
b1.configure(fg="white")
b2.configure(fg="white")
b3.configure(fg="white")
b4.configure(fg="white")
b5.configure(fg="white")
b6.configure(fg="white")
b7.configure(fg="white")
b9.configure(fg="white")
l1.configure(bg="#717171")
l1.configure(fg="white")
l2.configure(bg="#717171")
l2.configure(fg="white")
lb.configure(bg="#BFBFBF")
t1.configure(bg="#BFBFBF")
root.mainloop()
```

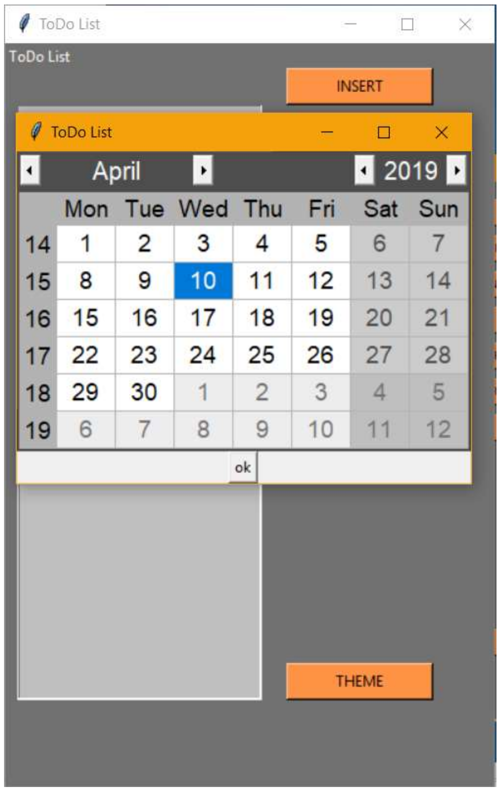
Screenshots of the project:



	todo	date
1	eat	2019-04-26
2	play	2019-04-12
3	assignment	2019-05-02
4	study	2019-04-04

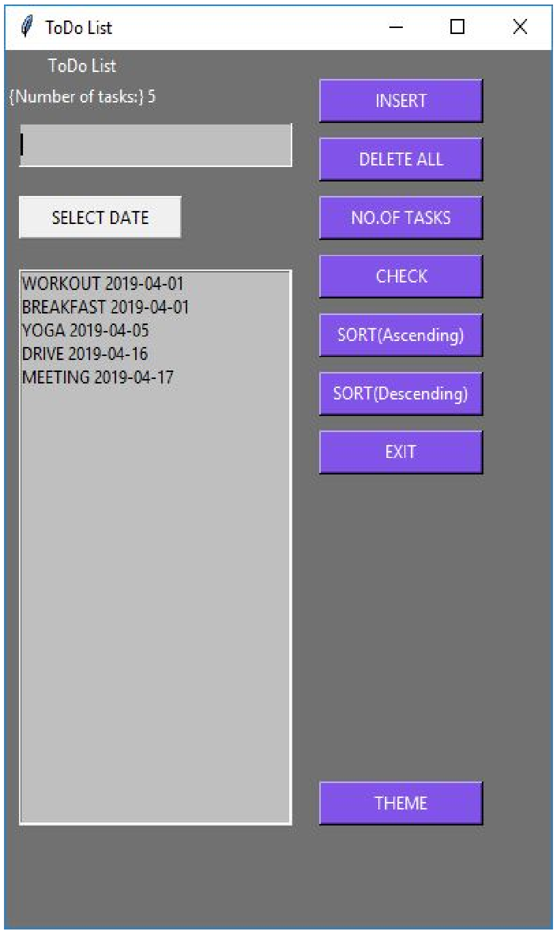


eat
play
assignment
study



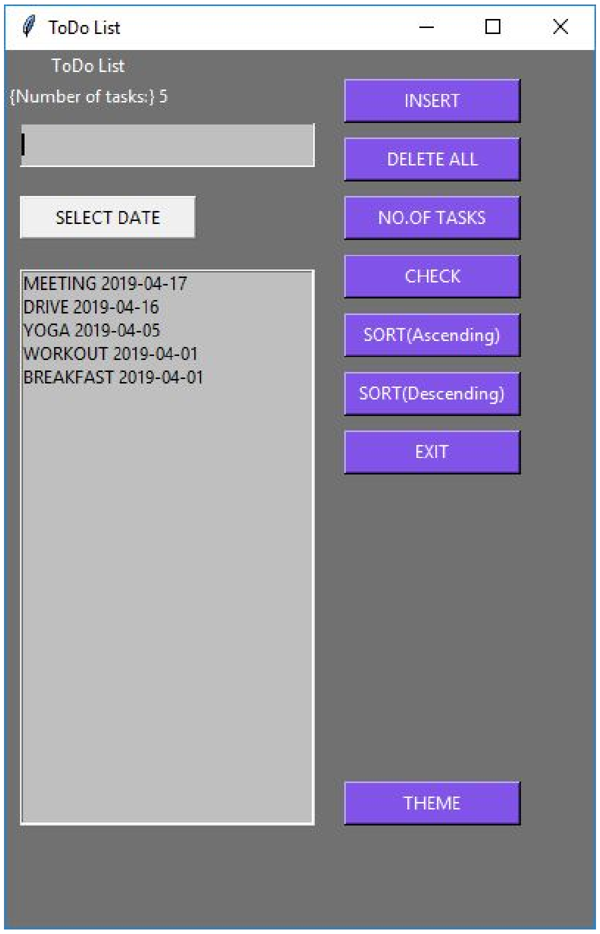
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
14	1	2	3	4	5	6	7
15	8	9	10	11	12	13	14
16	15	16	17	18	19	20	21
17	22	23	24	25	26	27	28
18	29	30	1	2	3	4	5
19	6	7	8	9	10	11	12

Fig. 4.2.1: Homepage



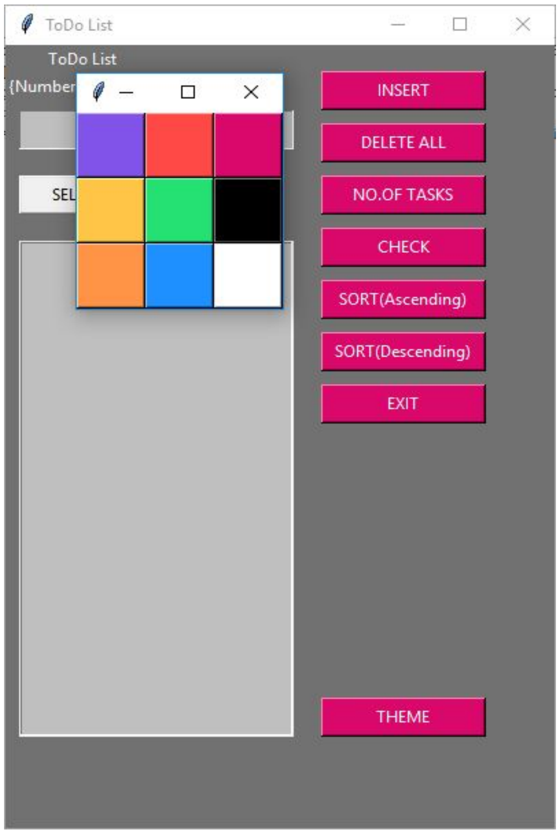
```
WORKOUT 2019-04-01
BREAKFAST 2019-04-01
YOGA 2019-04-05
DRIVE 2019-04-16
MEETING 2019-04-17
```

B1.NO. OF TASKS

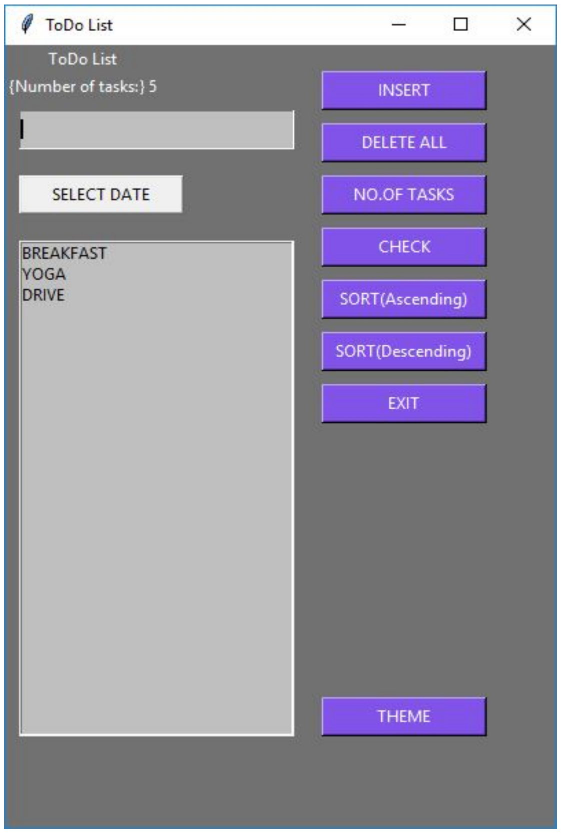


```
MEETING 2019-04-17
DRIVE 2019-04-16
YOGA 2019-04-05
WORKOUT 2019-04-01
BREAKFAST 2019-04-01
```

B2.SORT(Descending)



B3.THEME



```
BREAKFAST
YOGA
DRIVE
```

B4.CHECK

The home page of the web site contains buttons for entering various tasks. It also has other options to delete tasks or to check whether the task has been performed or not. It will also help to count the number of operations to be followed. we can update it according to time and date of our choice i.e Ascending or Descending.

6.Conclusion

a)Summary:

- ▶ We understood that using Python we can implement this program which will help us to in many ways. It Works as daily App. It helps us to stay organized and keep track of our assigned work
- ▶ To-do list can be implemented using any of the known programming languages.

b)Future Scope:

- ▶ This project has wide scope in future because this will automatically keep you organized.
- ▶ You can add various features in this program which will make it more attractive and useful for everyone.
- ▶ Like we can add priorities to every task and implement them accordingly.

7.References:

Websites:

- https://stackoverflow.com
- Python - Lists - Tutorials Point