PRODUCTIVITY MANAGER
MINI PROJECT REPORT

BATCH: B3

PROJECT CONTRIBUTORS:

UJJAWAL PATEL (16010121139)

YASKHIT POOJARY (16010121149)

AVNEESH PILLAI (16010121147)

PROBLEM STATEMENT:

To write a program where the user enters their timetable for the week/day and it sends a notification to the user about the task according to the day and time entered by the user. Also an extra functionality of the program is to notify the user about the extra assignments or submission at frequent intervals of time.

BRIEF OVERVIEW:

A basic use of the program is to set daily reminders or prompts for certain tasks at specified times this helps the user to maintain a productive day by not missing out on their work. Further we have added an extra edge to the program functionality by using an in-built module of python which is ‘random’ this is used to suggest the user about the extra assignments or the tasks which a not time bound at frequent intervals of time as we often forget to complete assignments which have a longer due date and we keep delaying such tasks. To overcome this issue we suggest the same tasks frequently so that the user can ensure they have their work ready well in advance and can utilize the rest of time for some other activities, all this leads to a better and productive schedule.

SYSTEM ARCHITECTURE:

In order to fulfil this problem statement, we used a few libraries like:

time and schedule- in order to schedule notifypy to send timely reminders.

random – to randomize the suggestion of the extra assignments.

pandas – to represent the timetable in a tabular form on the terminal.

notifypy – to create a notification window linked to the schedule module.

We prompt the user to provide their timetable to the system

Further we mould their input provided into a format that can be interpreted by our code and a suitable response can be provided to the user in our case the user is notified at the correct time about their task.

IMPLEMENTATION DETAILS:

CODE:

```
#Modules used for functionality
import time
import schedule
from notifypy import Notify
import random
import pandas as pd

#Made Lists and Dictionary to store date,time,tasks and number of tasks
days=[]
no_of_tasks=[]
rand_task=[]

SUNDAY={}
SUNDAY_task=[]
SUNDAY_time=[]

MONDAY={}
MONDAY_task=[]
MONDAY_time=[]

TUESDAY={}
TUESDAY_task=[]
TUESDAY_time=[]

WEDNESDAY={}
WEDNESDAY_task=[]
WEDNESDAY_time=[]

THURSDAY={}
THURSDAY_task=[]
THURSDAY_time=[]

FRIDAY={}
FRIDAY_task=[]
FRIDAY_time=[]

SATURDAY={}
SATURDAY_task=[]
SATURDAY_time=[]

#To input number of days,name of the day and number of tasks for that day.

j=0
while(j==0):
```

```
    n=input("Enter number of days from 1 to 7: ")
    if(n.isnumeric()==False):
        print("Invalid. Please do as stated above!!!")
        continue
    if(int(n)<0 or int(n)>7):
        print("Invalid. Please do as stated above!!!")
        continue

    for i in range(int(n)):
            day=input("Enter the day: ")
            day=day.upper()
            days.append(day)
            e=0
            while e==0:
                num = input("Enter the number of tasks on " +
days[i].upper()+": ")
                if (num.isnumeric()==False):
                    print("Please enter a valid number!!")
                    continue
                else:
                    no_of_tasks.append(int(num))
                e=1
            j=1


#To input the time and task
for i in range(len(days)):
    for j in range(no_of_tasks[i]):
        if days[i]=="MONDAY":
                t = input(f"Enter the time for task on {days[i].upper()} (in
24 hour format) : ")
                MONDAY_time.append(t)
                w = input("Enter the task: ")
                MONDAY_task.append(w)
                MONDAY["Time"]=MONDAY_time
                MONDAY["     Task"]=MONDAY_task

        if days[i]=="TUESDAY":
                t = input(f"Enter the time for task on {days[i].upper()} (in
24 hour format) : ")
                TUESDAY_time.append(t)
                w = input("Enter the task: ")
                TUESDAY_task.append(w)
                TUESDAY["Time"]=TUESDAY_time
                TUESDAY["     Task"]=TUESDAY_task

        if days[i]=="WEDNESDAY":
                t = input(f"Enter the time for task on {days[i].upper()} (in
24 hour format) : ")
```

```
                WEDNESDAY_time.append(t)
                w = input("Enter the task: ")
                WEDNESDAY_task.append(w)
                WEDNESDAY["Time"]=WEDNESDAY_time
                WEDNESDAY["     Task"]=WEDNESDAY_task

        if days[i]=="THURSDAY":
                t = input(f"Enter the time for task on {days[i].upper()} (in
24 hour format) : ")
                THURSDAY_time.append(t)
                w = input("Enter the task: ")
                THURSDAY_task.append(w)
                THURSDAY["Time"]=THURSDAY_time
                THURSDAY["     Task"]=THURSDAY_task

        if days[i]=="FRIDAY":
                t = input(f"Enter the time for task on {days[i].upper()} (in
24 hour format) : ")
                FRIDAY_time.append(t)
                w = input("Enter the task: ")
                FRIDAY_task.append(w)
                FRIDAY["Time"]=FRIDAY_time
                FRIDAY["     Task"]=FRIDAY_task

        if days[i]=="SATURDAY":
                t = input(f"Enter the time for task on {days[i].upper()} (in
24 hour format) : ")
                SATURDAY_time.append(t)
                w = input("Enter the task: ")
                SATURDAY_task.append(w)
                SATURDAY["Time"]=SATURDAY_time
                SATURDAY["     Task"]=SATURDAY_task

        if days[i]=="SUNDAY":
                t = input(f"Enter the time for task on {days[i].upper()} (in
24 hour format) : ")
                SUNDAY_time.append(t)
                w = input("Enter the task: ")
                SUNDAY_task.append(w)
                SUNDAY["Time"]=SUNDAY_time
                SUNDAY["     Task"]=SUNDAY_task

        if days[i]!="MONDAY" and days[i]!="TUESDAY" and days[i]!="WEDNESDAY"
and days[i]!="THURSDAY" and days[i]!="FRIDAY" and days[i]!="SATURDAY" and
days[i]!="SUNDAY":
            print(days[i]+ ": This day of the week does not exist!!!")


if MONDAY:
```

```
    dfmon=pd.DataFrame(MONDAY)
    print(f"\n\nYour time table for MONDAY is: \n{dfmon}")
if TUESDAY:
    dftue=pd.DataFrame(TUESDAY)
    print(f"\n\nYour time table for TUESDAY is: \n{dftue}")
if WEDNESDAY:
    dfwed=pd.DataFrame(WEDNESDAY)
    print(f"\n\nYour time table for WEDNESDAY is: \n{dfwed}")
if THURSDAY:
    dfthur=pd.DataFrame(THURSDAY)
    print(f"\n\nYour time table for THURSDAY is: \n{dfthur}")
if FRIDAY:
    dffri=pd.DataFrame(FRIDAY)
    print(f"\n\nYour time table for FRIDAY is: \n{dffri}")
if SATURDAY:
    dfsat=pd.DataFrame(SATURDAY)
    print(f"\n\nYour time table for SATURDAY is: \n{dfsat}")
if SUNDAY:
    dfsun=pd.DataFrame(SUNDAY)
    print(f"\n\nYour time table for SUNDAY is: \n{dfsun}")

# Input extra assignments to complete and randomly suggest using random
module.

z=0
while z==0:
    k=input("Do you have any other assignments to complete(Enter yes or no) :
")
    k=k.lower()
    if k=="yes":
        a=int(input("Enter number of assignments: "))
        for i in range(a):
            t=input("Enter tasks: ")
            rand_task.append(t)
        print(f"Your extra assignments are, {rand_task}")
        z=1
    if k=="no":
        rand_task.append(" ")
        z=1
    if k!="yes" and k!="no":
        print('''Please type "Yes" or "No" ''')
        continue
random_num=random.choice(rand_task)

#Function for notification and random task suggesting

def MONDAY_notification(MONDAY_task):
    notification = Notify()
```

```
    notification.title = "Productivity Manager"
    notification.icon="icon.ico"
    if k=="yes":
        notification.message = f"It is time for you to  {MONDAY_task} \nDo
remember to {random_num}"
    else:
        notification.message = "It is time for you to  "+ MONDAY_task
    notification.send()

def TUESDAY_notification(TUESDAY_task):
    notification = Notify()

    notification.title = "Productivity Manager"
    notification.icon="icon.ico"
    if k=="yes":
        notification.message = f"It is time for you to  {TUESDAY_task} \nDo
remember to {random_num}"
    else:
        notification.message = "It is time for you to  "+ TUESDAY_task
    notification.send()

def WEDNESDAY_notification(WEDNESDAY_task):
    notification = Notify()

    notification.title = "Productivity Manager"
    notification.icon="icon.ico"
    if k=="yes":
        notification.message = f"It is time for you to  {WEDNESDAY_task} \nDo
remember to {random_num}"
    else:
        notification.message = "It is time for you to  "+ WEDNESDAY_task
    notification.send()

def THURSDAY_notification(THURSDAY_task):
    notification = Notify()

    notification.title = "Productivity Manager"
    notification.icon="icon.ico"
    if k=="yes":
        notification.message = f"It is time for you to {THURSDAY_task} \nDo
remember to {random_num}"
    else:
        notification.message = "It is time for you to  "+ THURSDAY_task
    notification.send()

def FRIDAY_notification(FRIDAY_task):
    notification = Notify()
```

```
    notification.title = "Productivity Manager"
    notification.icon="icon.ico"
    if k=="yes":
        notification.message = f"It is time for you to  {FRIDAY_task} \nDo
remember to {random_num}"
    else:
        notification.message = "It is time for you to  "+ FRIDAY_task
    notification.send()

def SATURDAY_notification(SATURDAY_task):
    notification = Notify()

    notification.title = "Productivity Manager"
    notification.icon="icon.ico"
    if k=="yes":
        notification.message = f"It is time for you to  {SATURDAY_task} \nDo
remember to {random_num}"
    else:
        notification.message = "It is time for you to  "+ SATURDAY_task
    notification.send()

def SUNDAY_notification(SUNDAY_task):
    notification = Notify()

    notification.title = "Productivity Manager"
    notification.icon="icon.ico"
    if k=="yes":
        notification.message = f"It is time for you to  {SUNDAY_task} \nDo
remember to {random_num}"
    else:
        notification.message = "It is time for you to "+ SUNDAY_task
    notification.send()

#Scheduling notifications

if(MONDAY):
    i=0
    while i<len(MONDAY_time):
        schedule.every().monday.at(MONDAY_time[i]).do(MONDAY_notification,MOND
AY_task[i])
        i=i+1
if(TUESDAY):
    i=0
    while i<len(TUESDAY_time):
        schedule.every().tuesday.at(TUESDAY_time[i]).do(TUESDAY_notification,T
UESDAY_task[i])
        i=i+1
```

```
if(WEDNESDAY):
    i=0
    while i<len(WEDNESDAY_time):
        schedule.every().wednesday.at(WEDNESDAY_time[i]).do(WEDNESDAY_notifica
tion,WEDNESDAY_task[i])
        i=i+1
if(THURSDAY):
    i=0
    while i<len(THURSDAY_time):
        schedule.every().thursday.at(THURSDAY_time[i]).do(THURSDAY_notificatio
n,THURSDAY_task[i])
        i=i+1
if(FRIDAY):
    i=0
    while i<len(FRIDAY_time):
        schedule.every().friday.at(FRIDAY_time[i]).do(FRIDAY_notification,FRID
AY_task[i])
        i=i+1
if(SATURDAY):
    i=0
    while i<len(SATURDAY_time):
        schedule.every().saturday.at(SATURDAY_time[i]).do(SATURDAY_notificatio
n,SATURDAY_task[i])
        i=i+1
if(SUNDAY):
    i=0
    while i<len(SUNDAY_time):
        schedule.every().sunday.at(SUNDAY_time[i]).do(SUNDAY_notification,SUND
AY_task[i])
        i=i+1

while True:
    schedule.run_pending()
    time.sleep(1)
```

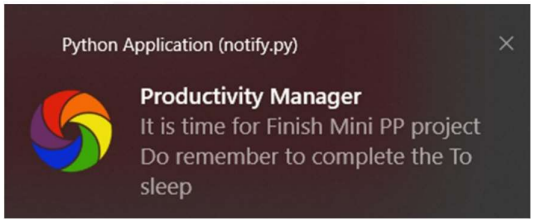
RESULT AND OUTPUT:

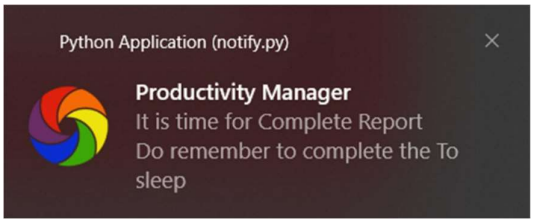
```
Enter number of days from 1 to 7: 3
Enter the day: Monday
Enter the number of tasks on MONDAY: 2
Enter the day: Tuesday
Enter the number of tasks on TUESDAY: 3
Enter the day: Sunday
Enter the number of tasks on SUNDAY: 2
Enter the time for task on MONDAY (in 24 hour format) : 13:00
Enter the task: Lunch
Enter the time for task on MONDAY (in 24 hour format) : 14:00
Enter the task: EM Lecture
Enter the time for task on TUESDAY (in 24 hour format) : 10:30
Enter the task: CS LECTURE
Enter the time for task on TUESDAY (in 24 hour format) : 15:00
Enter the task: EM LECTURE
Enter the time for task on TUESDAY (in 24 hour format) : 17:00
Enter the task: STUDY
Enter the time for task on SUNDAY (in 24 hour format) : 13:00
Enter the task: WAKE UP
Enter the time for task on SUNDAY (in 24 hour format) : 20:00
Enter the task: SLEEP
```

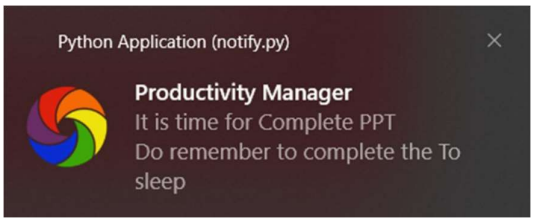
```
Your time table for MONDAY is:
    Time        Task
0  13:00       Lunch
1  14:00  EM Lecture


Your time table for TUESDAY is:
    Time        Task
0  10:30  CS LECTURE
1  15:00  EM LECTURE
2  17:00       STUDY


Your time table for SUNDAY is:
    Time      Task
0  13:00   WAKE UP
1  20:00     SLEEP
Do you have any other assignments to complete(Enter yes or no) : Yes
Enter number of assignments: 4
Enter tasks: Complete PP Mini Project
Enter tasks: Work on website
Enter tasks: Finish EM assignment
```







CONCLUSION:

With the help of this project we try to resolve a common problem of remembering tasks and to complete them on time.