# **LANCER’S CONVENT SCHOOL**

**PRASHANT VIHAR, DELHI-110085**



**A PROJECT REPORT ON:**

**OPERATING SYSTEM**

# **LANCER’S CONVENT SCHOOL**

SUBMITTED TO MS. SAKSHI PATODI P.G.T. (COMP. SC)

SUBMITTED BY:

SHIVAY MENDITRATTA

11 – B

12050



## This is to certify that SHIVAY MENDIRATTA of class XII A/B of LANCER’S CONVENT SCHOOL has done his/her project on STATISTICS under my supervision. He / She has taken interest and has shown at most sincerity in completion of this project.

## I certify this Project up to my expectation & as per guidelines issued by CBSE, NEW DELHI.

*Internal Examiner External Examiner*

*Principal*



It is with pleasure that I acknowledge my sincere gratitude to our teacher, ***MS. SAKSHI PATODI*** who taught and undertook the responsibility of teaching the subject computer science. I have been greatly benefited from her classes.

I am especially indebted to our Principal *<<Pri“nci“pa/* &ame>> who has always been a source of encouragement and support and without whose inspiration this project would not have been a successful.

Finally, I would like to express my sincere appreciation for all the other students for my batch their friendship & the fine times that we all shared together.

**HARDWARE AND SOFTWARE REQUIREMENT**

* HARDWARE REQUIREMENT: -

1. OPERATING SYSTEM : Windows 7 and Above
2. PROCESSOR : ANY (recommended – intel

core-i series)

1. RAM : 1GB+
2. KEYBOARD AND MOUSE

**🡺** SOFTWARE REQUIREMENT: -

1. WINDOWS OS
2. PYTHON

**DESCRIPTION OF THE PROJECT**

1. **FEATURES AND USES/PURPOSE: -**

This project is a semi-CUI (Character User Interface) and semi-GUI (Graphic User Interface) based operating system. This also contains encrypted username and password which makes is safer. In home page will find the following options.: -

[1] To Open Google

[3] To Open File Explorer

[4] To Configure and Open Bios’

[5] To Open Terminal

[6] To Open Calculator

[7] To Open Digital Clock

[8] To Close OS Safely

Here Calculator, digital clock and text editor are totally based on GUI whereas you will find others to be idle.

1. **INPUT AND OUTPUT: -**

In this project you have to give input by using numbers like by entering number 6 a calculator will open or by entering number 3 your file explorer will open.

While talking separately:

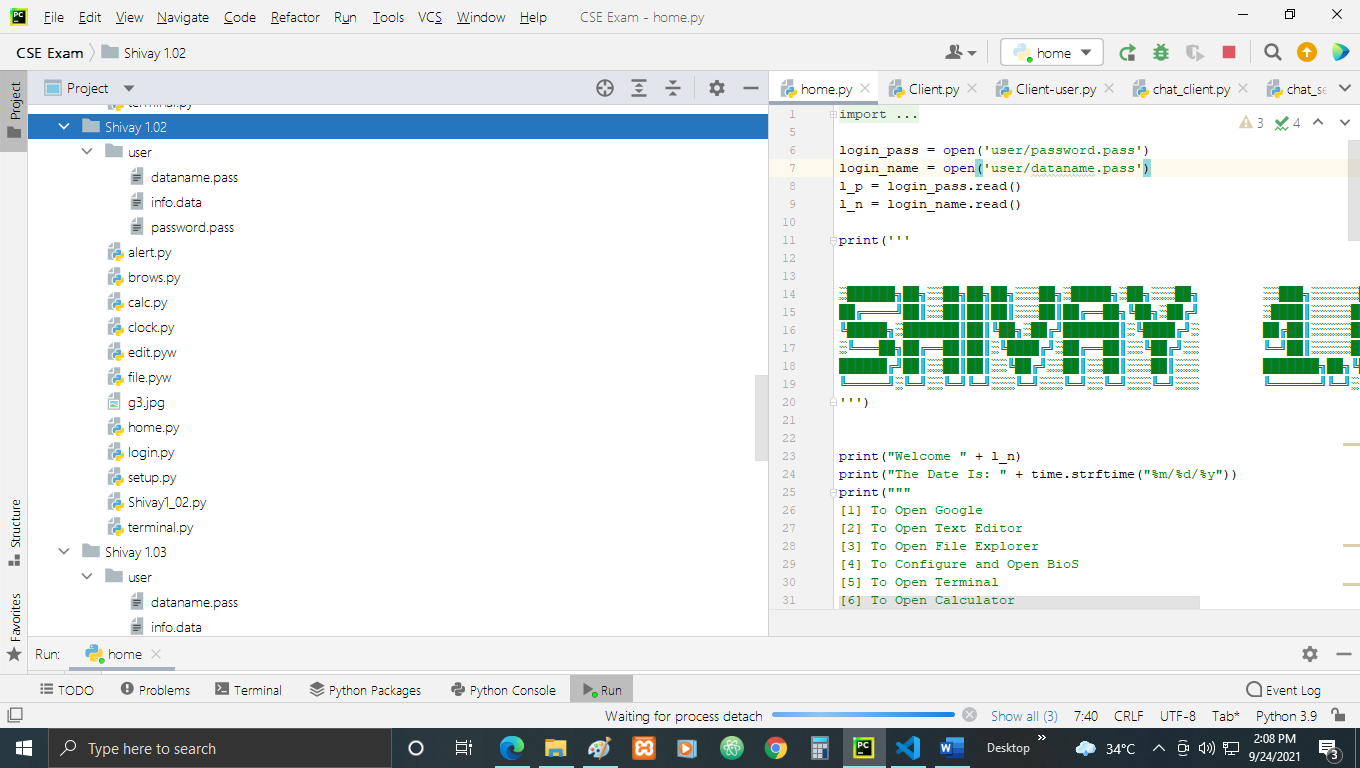
1. Google – You have to enter the URL of the page you want to open
2. File Explorer – This feature just opens your file explorer where you can move your file or open any file.
3. Configure and Bios’ – This feature is very helpful in changing password and username. At first you will be asked your current username and password which is encrypted in another file as mentioned above.
4. Terminal – This is my own made command prompt which have all those features in it.
5. Calculator – This is a totally GUI based calculator which allows you to add, multiply, subtract, divide, powers, roots and much more.
6. Digital Clock – This is a GUI based clock which tells you the current time.
7. Close OS Safely – This exits the OS.

**FLOW CHART / WORKING / PROGRAM LOGIC: -**

1. **WHO ARE THE USERS: -**

All the people who want to use a mixed OS which is not only based on CUI but also on GUI. This feature makes it the best in market.





1. Dataname.pass: -

Shivay

1. Info.data: -

1

1. Password.pass: -

1234

1. alert.py: -

from tkinter import messagebox, Tk  
  
  
def alert(title, message, kind='info', hidemain=True):  
 if kind not in ('error', 'warning', 'info'):  
 raise ValueError('Unsupported alert kind.')  
  
 show\_method = getattr(messagebox, 'show{}'.format(kind))  
 show\_method(title, message)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 Tk().withdraw()  
 alert('Hello', 'Hello World')  
 alert('Hello Again', 'Hello World 2', kind='warning')

1. brows.py: -

import webbrowser

print('''  
  
░██████╗██╗░░██╗██╗██╗░░░██╗░█████╗░██╗░░░██╗ ░░███╗░░░░░░█████╗░██████╗░  
██╔════╝██║░░██║██║██║░░░██║██╔══██╗╚██╗░██╔╝ ░████║░░░░░██╔══██╗╚════██╗  
╚█████╗░███████║██║╚██╗░██╔╝███████║░╚████╔╝░ ██╔██║░░░░░██║░░██║░░███╔═╝  
░╚═══██╗██╔══██║██║░╚████╔╝░██╔══██║░░╚██╔╝░░ ╚═╝██║░░░░░██║░░██║██╔══╝░░  
██████╔╝██║░░██║██║░░╚██╔╝░░██║░░██║░░░██║░░░ ███████╗██╗╚█████╔╝███████╗  
╚═════╝░╚═╝░░╚═╝╚═╝░░░╚═╝░░░╚═╝░░╚═╝░░░╚═╝░░░ ╚══════╝╚═╝░╚════╝░╚══════╝  
  
░██████╗███████╗░█████╗░██████╗░░█████╗░██╗░░██╗  
██╔════╝██╔════╝██╔══██╗██╔══██╗██╔══██╗██║░░██║  
╚█████╗░█████╗░░███████║██████╔╝██║░░╚═╝███████║  
░╚═══██╗██╔══╝░░██╔══██║██╔══██╗██║░░██╗██╔══██║  
██████╔╝███████╗██║░░██║██║░░██║╚█████╔╝██║░░██║  
╚═════╝░╚══════╝╚═╝░░╚═╝╚═╝░░╚═╝░╚════╝░╚═╝░░╚═╝   
███████╗███╗░░██╗░██████╗░██╗███╗░░██╗███████╗  
██╔════╝████╗░██║██╔════╝░██║████╗░██║██╔════╝  
█████╗░░██╔██╗██║██║░░██╗░██║██╔██╗██║█████╗░░  
██╔══╝░░██║╚████║██║░░╚██╗██║██║╚████║██╔══╝░░  
███████╗██║░╚███║╚██████╔╝██║██║░╚███║███████╗  
╚══════╝╚═╝░░╚══╝░╚═════╝░╚═╝╚═╝░░╚══╝╚══════╝  
''')  
  
  
while True:  
 browser = input("Serch Browser or Type URL: ")  
 webbrowser.open(browser)

1. calc.py: -

import tkinter as tk  
  
LARGE\_FONT\_STYLE = ("Arial", 40, "bold")  
SMALL\_FONT\_STYLE = ("Arial", 16)  
DIGITS\_FONT\_STYLE = ("Arial", 24, "bold")  
DEFAULT\_FONT\_STYLE = ("Arial", 20)  
  
OFF\_WHITE = "#F8FAFF"  
WHITE = "#FFFFFF"  
LIGHT\_BLUE = "#CCEDFF"  
LIGHT\_GRAY = "#F5F5F5"  
LABEL\_COLOR = "#25265E"  
  
  
class Calculator:  
 def \_\_init\_\_(self):  
 self.window = tk.Tk()  
 self.window.geometry("375x667")  
 self.window.resizable(0, 0)  
 self.window.title("Shivay 1.02 Calculator")  
  
 self.total\_expression = ""  
 self.current\_expression = ""  
 self.display\_frame = self.create\_display\_frame()  
  
 self.total\_label, self.label = self.create\_display\_labels()  
  
 self.digits = {  
 7: (1, 1), 8: (1, 2), 9: (1, 3),  
 4: (2, 1), 5: (2, 2), 6: (2, 3),  
 1: (3, 1), 2: (3, 2), 3: (3, 3),  
 0: (4, 2), '.': (4, 1)  
 }  
 self.operations = {"/": "\u00F7", "\*": "\u00D7", "-": "-", "+": "+"}  
 self.buttons\_frame = self.create\_buttons\_frame()  
  
 self.buttons\_frame.rowconfigure(0, weight=1)  
 for x in range(1, 5):  
 self.buttons\_frame.rowconfigure(x, weight=1)  
 self.buttons\_frame.columnconfigure(x, weight=1)  
 self.create\_digit\_buttons()  
 self.create\_operator\_buttons()  
 self.create\_special\_buttons()  
 self.bind\_keys()  
  
 def bind\_keys(self):  
 self.window.bind("<Return>", lambda event: self.evaluate())  
 for key in self.digits:  
 self.window.bind(str(key), lambda event, digit=key: self.add\_to\_expression(digit))  
  
 for key in self.operations:  
 self.window.bind(key, lambda event, operator=key: self.append\_operator(operator))  
  
 def create\_special\_buttons(self):  
 self.create\_clear\_button()  
 self.create\_equals\_button()  
 self.create\_square\_button()  
 self.create\_sqrt\_button()  
  
 def create\_display\_labels(self):  
 total\_label = tk.Label(self.display\_frame, text=self.total\_expression, anchor=tk.E, bg=LIGHT\_GRAY,  
 fg=LABEL\_COLOR, padx=24, font=SMALL\_FONT\_STYLE)  
 total\_label.pack(expand=True, fill='both')  
  
 label = tk.Label(self.display\_frame, text=self.current\_expression, anchor=tk.E, bg=LIGHT\_GRAY,  
 fg=LABEL\_COLOR, padx=24, font=LARGE\_FONT\_STYLE)  
 label.pack(expand=True, fill='both')  
  
 return total\_label, label  
  
 def create\_display\_frame(self):  
 frame = tk.Frame(self.window, height=221, bg=LIGHT\_GRAY)  
 frame.pack(expand=True, fill="both")  
 return frame  
  
 def add\_to\_expression(self, value):  
 self.current\_expression += str(value)  
 self.update\_label()  
  
 def create\_digit\_buttons(self):  
 for digit, grid\_value in self.digits.items():  
 button = tk.Button(self.buttons\_frame, text=str(digit), bg=WHITE, fg=LABEL\_COLOR, font=DIGITS\_FONT\_STYLE,  
 borderwidth=0, command=lambda x=digit: self.add\_to\_expression(x))  
 button.grid(row=grid\_value[0], column=grid\_value[1], sticky=tk.NSEW)  
  
 def append\_operator(self, operator):  
 self.current\_expression += operator  
 self.total\_expression += self.current\_expression  
 self.current\_expression = ""  
 self.update\_total\_label()  
 self.update\_label()  
  
 def create\_operator\_buttons(self):  
 i = 0  
 for operator, symbol in self.operations.items():  
 button = tk.Button(self.buttons\_frame, text=symbol, bg=OFF\_WHITE, fg=LABEL\_COLOR, font=DEFAULT\_FONT\_STYLE,  
 borderwidth=0, command=lambda x=operator: self.append\_operator(x))  
 button.grid(row=i, column=4, sticky=tk.NSEW)  
 i += 1  
  
 def clear(self):  
 self.current\_expression = ""  
 self.total\_expression = ""  
 self.update\_label()  
 self.update\_total\_label()  
  
 def create\_clear\_button(self):  
 button = tk.Button(self.buttons\_frame, text="C", bg=OFF\_WHITE, fg=LABEL\_COLOR, font=DEFAULT\_FONT\_STYLE,  
 borderwidth=0, command=self.clear)  
 button.grid(row=0, column=1, sticky=tk.NSEW)  
  
 def square(self):  
 self.current\_expression = str(eval(f"{self.current\_expression}\*\*2"))  
 self.update\_label()  
  
 def create\_square\_button(self):  
 button = tk.Button(self.buttons\_frame, text="x\u00b2", bg=OFF\_WHITE, fg=LABEL\_COLOR, font=DEFAULT\_FONT\_STYLE,  
 borderwidth=0, command=self.square)  
 button.grid(row=0, column=2, sticky=tk.NSEW)  
  
 def sqrt(self):  
 self.current\_expression = str(eval(f"{self.current\_expression}\*\*0.5"))  
 self.update\_label()  
  
 def create\_sqrt\_button(self):  
 button = tk.Button(self.buttons\_frame, text="\u221ax", bg=OFF\_WHITE, fg=LABEL\_COLOR, font=DEFAULT\_FONT\_STYLE,  
 borderwidth=0, command=self.sqrt)  
 button.grid(row=0, column=3, sticky=tk.NSEW)  
  
 def evaluate(self):  
 self.total\_expression += self.current\_expression  
 self.update\_total\_label()  
 try:  
 self.current\_expression = str(eval(self.total\_expression))  
  
 self.total\_expression = ""  
 except Exception as e:  
 self.current\_expression = "Error"  
 finally:  
 self.update\_label()  
  
 def create\_equals\_button(self):  
 button = tk.Button(self.buttons\_frame, text="=", bg=LIGHT\_BLUE, fg=LABEL\_COLOR, font=DEFAULT\_FONT\_STYLE,  
 borderwidth=0, command=self.evaluate)  
 button.grid(row=4, column=3, columnspan=2, sticky=tk.NSEW)  
  
 def create\_buttons\_frame(self):  
 frame = tk.Frame(self.window)  
 frame.pack(expand=True, fill="both")  
 return frame  
  
 def update\_total\_label(self):  
 expression = self.total\_expression  
 for operator, symbol in self.operations.items():  
 expression = expression.replace(operator, f' {symbol} ')  
 self.total\_label.config(text=expression)  
  
 def update\_label(self):  
 self.label.config(text=self.current\_expression[:11])  
  
 def run(self):  
 self.window.mainloop()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 calc = Calculator()  
 calc.run()

1. clock.py: -

from tkinter import \*  
from datetime import datetime  
from time import strftime  
from PIL import ImageTk,Image  
  
w=Tk()  
w.geometry('1280x720')  
w.minsize(750,200)  
w.title("Digital Clock")  
  
  
*#Extracting day*a=datetime.today().strftime('%A')  
b=(a.upper())  
c=(b[0:2])   
  
*#setting background*img1=Image.open("g3.jpg")  
img2= ImageTk.PhotoImage(img1)  
Label(w,image=img2).place(x=-2,y=0)  
  
  
f1=Frame(w,width=750, height=200,bg='#0e1013')  
f1.pack(expand=True)  
  
*#Mechenism*def time():  
 a=strftime('%H : %M : %S') *#%H %M %S* l1.config(text=a)  
 l1.after(1000,time)  
  
l1=Label(f1, font=('Century Gothic',60),  
 bg='#0e1013',  
 foreground='#d3d3d3')  
  
l1.place(x=275,y=35)  
time()  
  
l2=Label(f1, font=('Century Gothic',60),  
 bg='#0e1013',  
 foreground='#d3d3d3')  
l2.config(text=c+" |")  
l2.place(x=75,y=35)  
  
*#Required labels*def labels():  
 l3=Label(f1, font=('Century Gothic',8),bg='#0e1013',fg='#7f7f7f',text='DAY')  
 l3.place(x=122,y=130)  
  
 l4=Label(f1, font=('Century Gothic',8),bg='#0e1013',fg='#7f7f7f',text='HOURS')  
 l4.place(x=305,y=130)  
  
 l5=Label(f1, font=('Century Gothic',8),bg='#0e1013',fg='#7f7f7f',text='MINUTES')  
 l5.place(x=445,y=130)  
  
 l3=Label(f1, font=('Century Gothic',8),bg='#0e1013',fg='#7f7f7f',text='SECONDS')  
 l3.place(x=445+145+5,y=130)  
  
labels()  
  
w.mainloop()

1. file.pyw: -

from tkinter import \*  
from tkinter import filedialog  
import os  
  
def openFile():  
 filepath = filedialog.askopenfilename(initialdir="/",title="Shivay 1.02 Open",filetypes= (("text files","\*.txt"),("all files","\*.\*")))  
 os.startfile(filepath)  
  
window = Tk()  
button = Button(text="Open",command=openFile)  
button.pack()  
window.mainloop()

1. g3.jpg: -



1. home.py: -

import time  
import os  
import socket  
from tkinter import messagebox, Tk  
  
login\_pass = open('user/password.pass')  
login\_name = open('user/dataname.pass')  
l\_p = login\_pass.read()  
l\_n = login\_name.read()  
  
print('''  
  
  
░██████╗██╗░░██╗██╗██╗░░░██╗░█████╗░██╗░░░██╗ ░░███╗░░░░░░█████╗░██████╗░  
██╔════╝██║░░██║██║██║░░░██║██╔══██╗╚██╗░██╔╝ ░████║░░░░░██╔══██╗╚════██╗  
╚█████╗░███████║██║╚██╗░██╔╝███████║░╚████╔╝░ ██╔██║░░░░░██║░░██║░░███╔═╝  
░╚═══██╗██╔══██║██║░╚████╔╝░██╔══██║░░╚██╔╝░░ ╚═╝██║░░░░░██║░░██║██╔══╝░░  
██████╔╝██║░░██║██║░░╚██╔╝░░██║░░██║░░░██║░░░ ███████╗██╗╚█████╔╝███████╗  
╚═════╝░╚═╝░░╚═╝╚═╝░░░╚═╝░░░╚═╝░░╚═╝░░░╚═╝░░░ ╚══════╝╚═╝░╚════╝░╚══════╝  
''')  
  
  
print("Welcome " + l\_n)  
print("The Date Is: " + time.strftime("%m/%d/%y"))  
print("""  
[1] To Open Google   
[3] To Open File Explorer  
[4] To Configure and Open BioS  
[5] To Open Terminal  
[6] To Open Calculator  
[7] To Open Digital Clock  
[8] To Close OS Safely  
""")  
  
select = input("[?]: ")  
  
if select == '1':  
 os.startfile('home.py')  
 os.startfile('brows.py')

elif select == '3':  
 os.startfile('home.py')  
 os.startfile('file.pyw')  
  
elif select == '4':  
 while True:  
 b\_login = input(str("Please Enter The Password " + l\_n + " To Open BioS: "))  
 if b\_login == l\_p:  
 print("Opening BioS")  
 host\_name = socket.gethostname()  
 host\_ip = socket.gethostbyname(host\_name)  
 print("[1] USER NAME: " + l\_n)  
 print("[2] PASSWORD: " + l\_p)  
 print("Hostname:", host\_name)  
 print("LOCAL IPS: " + host\_ip)  
 print()  
 print()  
 print("Enter [3] to exit BIOS")  
 edit\_b = input("Enter [?] to change setting: ")  
 if edit\_b == '1':  
 edit\_n = input("Enter New Username: ")  
 with open('user/dataname.pass', 'w') as f:  
 f.writelines(edit\_n)  
 print("Username Changed To " + edit\_n)  
 input("Press Enter To Restart: ")  
 os.startfile('home.py')  
 os.system('exit')  
  
 elif edit\_b == '2':  
 edit\_p = input("Enter New Password: ")  
 with open('user/password.pass', 'w') as f:  
 f.writelines(edit\_p)  
  
 print("Password Changed To " + edit\_p)  
 input("Press Enter To Restart: ")  
 os.startfile('home.py')  
 os.system('exit')  
  
 elif edit\_b == '3':  
 os.startfile('home.py')  
 os.system('exit')  
 else:  
 def alert(title, message, kind='info', hidemain=True):  
 if kind not in ('error', 'warning', 'info'):  
 raise ValueError('Unsupported alert kind.')  
  
 show\_method = getattr(messagebox, 'show{}'.format(kind))  
 show\_method(title, message)  
  
  
 if \_\_name\_\_ == '\_\_main\_\_':  
 Tk().withdraw()  
 alert('Alert', 'Wrong Password!', kind='warning')  
  
elif select == '5':  
 os.startfile('home.py')  
 os.startfile('terminal.py')  
  
elif select == '6':  
 os.startfile('home.py')  
 os.startfile('calc.py')  
  
elif select == '7':  
 os.startfile('home.py')  
 os.startfile('clock.py')  
  
elif select == '8':  
 os.system('exit')  
  
else:  
 def alert(title, message, kind='info', hidemain=True):  
 if kind not in ('error', 'warning', 'info'):  
 raise ValueError('Unsupported alert kind.')  
  
 show\_method = getattr(messagebox, 'show{}'.format(kind))  
 show\_method(title, message)  
  
  
 if \_\_name\_\_ == '\_\_main\_\_':  
 Tk().withdraw()  
 alert('Alert', 'Please enter an input lying between 1 to 8')  
 os.startfile('home.py')

1. login.py: -

import os  
import time  
from tkinter import messagebox, Tk  
  
login\_pass = open('user/password.pass')  
login\_name = open('user/dataname.pass')  
l\_p = login\_pass.read()  
l\_n = login\_name.read()  
  
  
print('''  
░██████╗██╗░░██╗██╗██╗░░░██╗░█████╗░██╗░░░██╗ ░░███╗░░░░░░█████╗░██████╗░  
██╔════╝██║░░██║██║██║░░░██║██╔══██╗╚██╗░██╔╝ ░████║░░░░░██╔══██╗╚════██╗  
╚█████╗░███████║██║╚██╗░██╔╝███████║░╚████╔╝░ ██╔██║░░░░░██║░░██║░░███╔═╝  
░╚═══██╗██╔══██║██║░╚████╔╝░██╔══██║░░╚██╔╝░░ ╚═╝██║░░░░░██║░░██║██╔══╝░░  
██████╔╝██║░░██║██║░░╚██╔╝░░██║░░██║░░░██║░░░ ███████╗██╗╚█████╔╝███████╗  
╚═════╝░╚═╝░░╚═╝╚═╝░░░╚═╝░░░╚═╝░░╚═╝░░░╚═╝░░░ ╚══════╝╚═╝░╚════╝░╚══════╝  
  
██╗░░░░░░█████╗░░██████╗░██╗███╗░░██╗  
██║░░░░░██╔══██╗██╔════╝░██║████╗░██║  
██║░░░░░██║░░██║██║░░██╗░██║██╔██╗██║  
██║░░░░░██║░░██║██║░░╚██╗██║██║╚████║  
███████╗╚█████╔╝╚██████╔╝██║██║░╚███║  
╚══════╝░╚════╝░░╚═════╝░╚═╝╚═╝░░╚══╝  
  
  
  
With Alert box, Digital Clock and Improved Calculator!!!  
''')  
  
  
while True:  
 log = input("Enter Password To " + l\_n + " To Login: ")  
  
 if log == l\_p:  
 print("Opening Home Page...")  
 time.sleep(0.5)  
 os.startfile("home.py")  
 break  
  
 else:  
 def alert(title, message, kind='info', hidemain=True):  
 if kind not in ('error', 'warning', 'info'):  
 raise ValueError('Unsupported alert kind.')  
  
 show\_method = getattr(messagebox, 'show{}'.format(kind))  
 show\_method(title, message)  
  
  
 if \_\_name\_\_ == '\_\_main\_\_':  
 Tk().withdraw()  
 alert('Alert', 'Wrong Password!', kind='warning')

1. setup.py: -

import os  
import time  
  
with open('user/info.data', 'w') as f:  
 f.writelines("1")  
  
  
print('''  
░██████╗██╗░░██╗██╗██╗░░░██╗░█████╗░██╗░░░██╗ ░░███╗░░░░░░█████╗░██████╗░  
██╔════╝██║░░██║██║██║░░░██║██╔══██╗╚██╗░██╔╝ ░████║░░░░░██╔══██╗╚════██╗  
╚█████╗░███████║██║╚██╗░██╔╝███████║░╚████╔╝░ ██╔██║░░░░░██║░░██║░░███╔═╝  
░╚═══██╗██╔══██║██║░╚████╔╝░██╔══██║░░╚██╔╝░░ ╚═╝██║░░░░░██║░░██║██╔══╝░░  
██████╔╝██║░░██║██║░░╚██╔╝░░██║░░██║░░░██║░░░ ███████╗██╗╚█████╔╝███████╗  
╚═════╝░╚═╝░░╚═╝╚═╝░░░╚═╝░░░╚═╝░░╚═╝░░░╚═╝░░░ ╚══════╝╚═╝░╚════╝░╚══════╝  
  
░██████╗███████╗████████╗██╗░░░██╗██████╗░  
██╔════╝██╔════╝╚══██╔══╝██║░░░██║██╔══██╗  
╚█████╗░█████╗░░░░░██║░░░██║░░░██║██████╔╝  
░╚═══██╗██╔══╝░░░░░██║░░░██║░░░██║██╔═══╝░  
██████╔╝███████╗░░░██║░░░╚██████╔╝██║░░░░░  
╚═════╝░╚══════╝░░░╚═╝░░░░╚═════╝░╚═╝░░░░░  
''')  
  
  
name = input("Please Enter Your Username To Be Displayed: ")  
pas = input("Please Enter Your Password To Login: ")  
  
with open('user/dataname.pass', 'w') as f:  
 f.writelines(name)  
  
with open('user/password.pass', 'w') as f:  
 f.writelines(pas)  
  
print("Setup Complete!!!")  
print("Opening Login Page...")  
time.sleep(0.5)  
os.startfile('login.py')

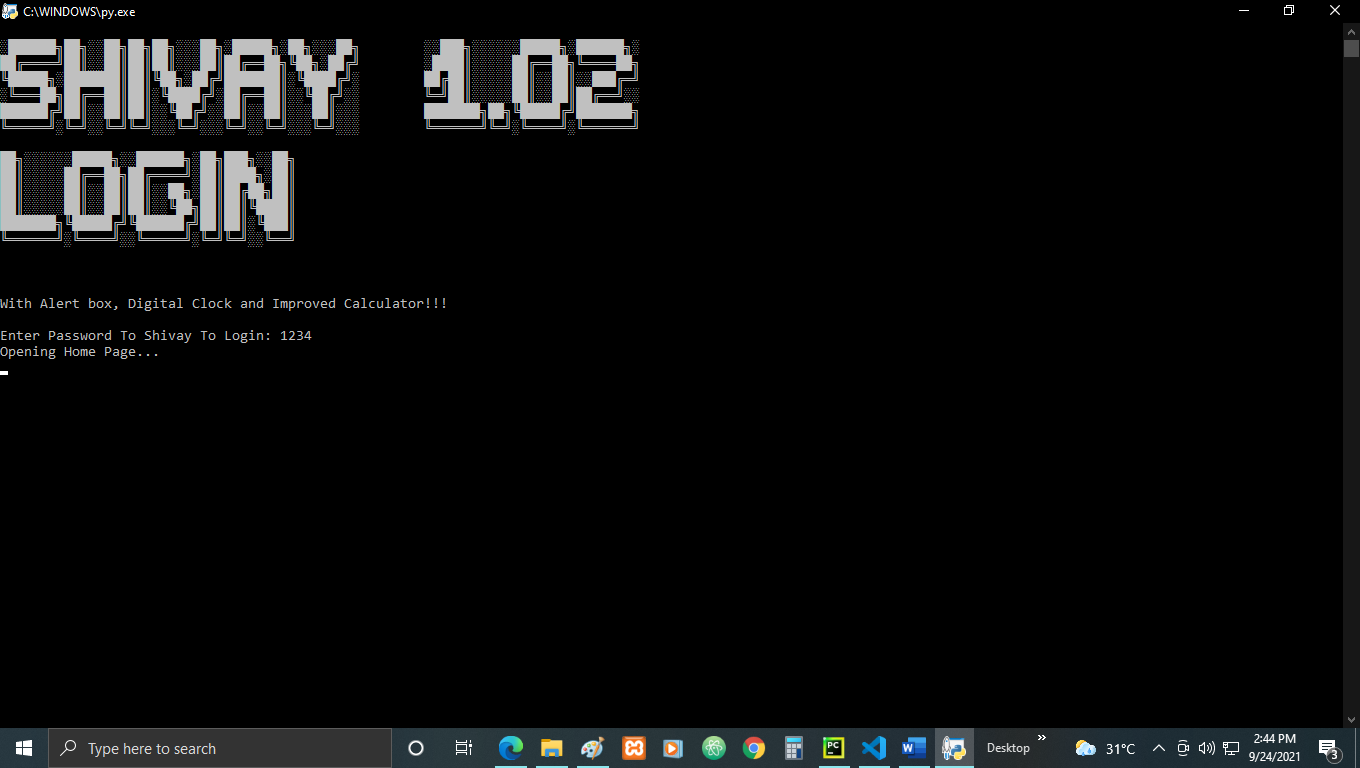
1. Shivay1\_02.py: -

import os  
  
data\_info = open('user/info.data')  
data = data\_info.read()  
  
if data == '0':  
 print("Opening Register Page...")  
 os.startfile("setup.py")  
  
if data == '1':  
 print("Opening Login Page...")  
 os.startfile("login.py")

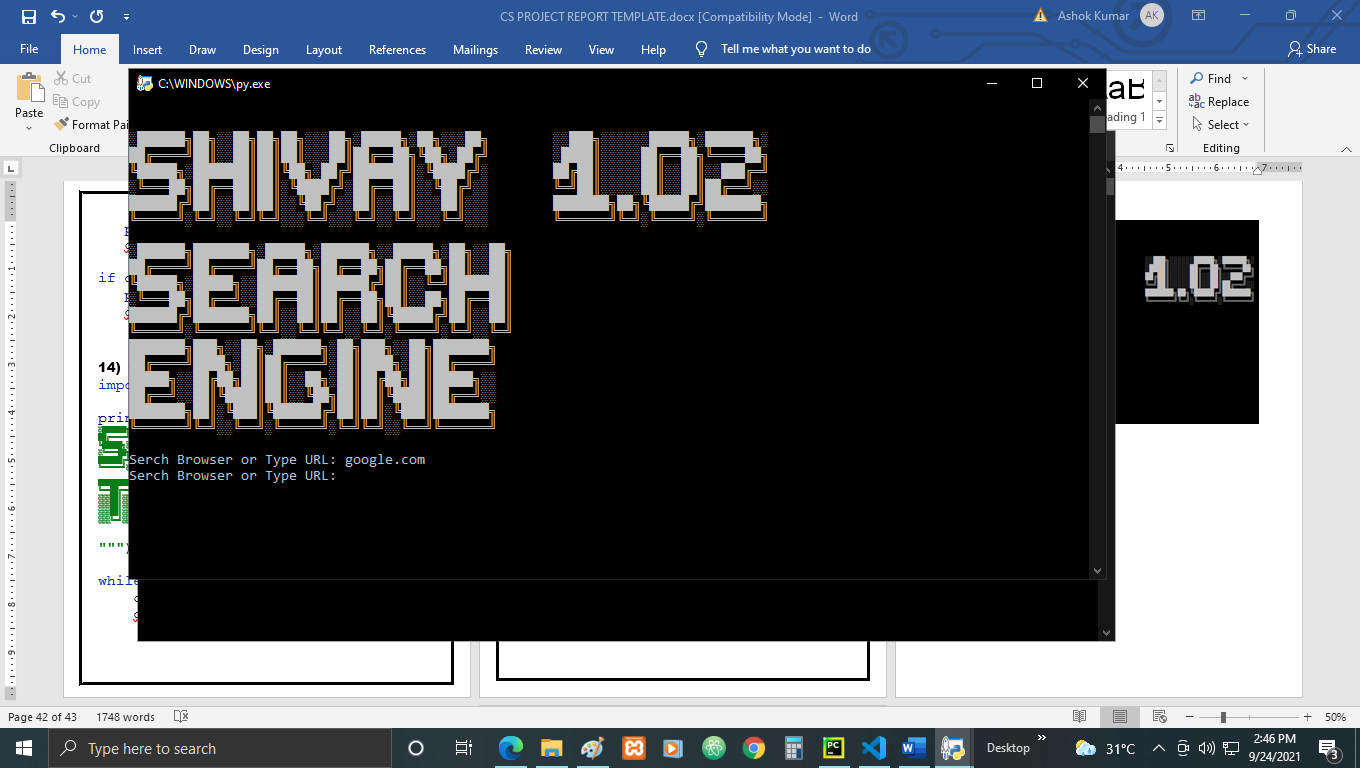
1. terminal.py: -

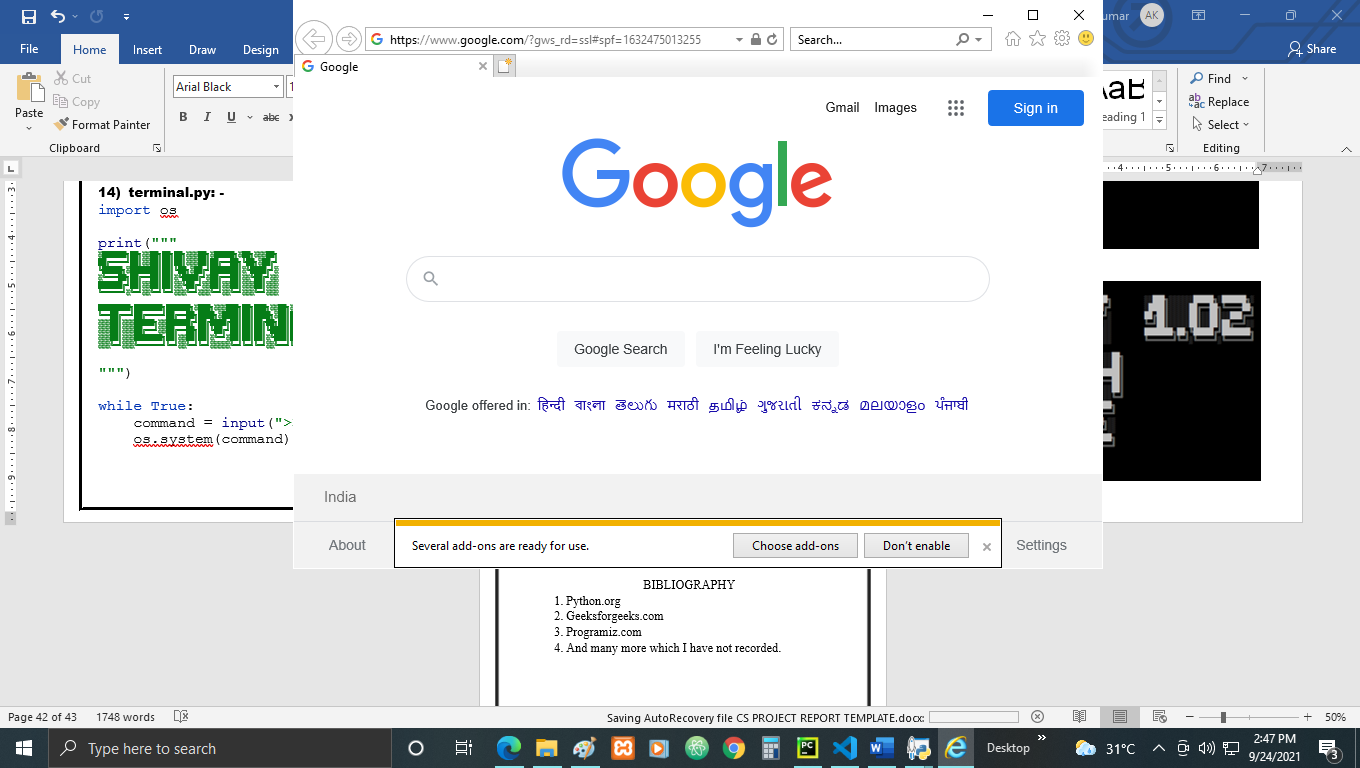
import os  
  
print("""  
░██████╗██╗░░██╗██╗██╗░░░██╗░█████╗░██╗░░░██╗ ░░███╗░░░░░░█████╗░██████╗░  
██╔════╝██║░░██║██║██║░░░██║██╔══██╗╚██╗░██╔╝ ░████║░░░░░██╔══██╗╚════██╗  
╚█████╗░███████║██║╚██╗░██╔╝███████║░╚████╔╝░ ██╔██║░░░░░██║░░██║░░███╔═╝  
░╚═══██╗██╔══██║██║░╚████╔╝░██╔══██║░░╚██╔╝░░ ╚═╝██║░░░░░██║░░██║██╔══╝░░  
██████╔╝██║░░██║██║░░╚██╔╝░░██║░░██║░░░██║░░░ ███████╗██╗╚█████╔╝███████╗  
╚═════╝░╚═╝░░╚═╝╚═╝░░░╚═╝░░░╚═╝░░╚═╝░░░╚═╝░░░ ╚══════╝╚═╝░╚════╝░╚══════╝  
  
████████╗███████╗██████╗░███╗░░░███╗██╗███╗░░██╗░█████╗░██╗░░░░░  
╚══██╔══╝██╔════╝██╔══██╗████╗░████║██║████╗░██║██╔══██╗██║░░░░░  
░░░██║░░░█████╗░░██████╔╝██╔████╔██║██║██╔██╗██║███████║██║░░░░░  
░░░██║░░░██╔══╝░░██╔══██╗██║╚██╔╝██║██║██║╚████║██╔══██║██║░░░░░  
░░░██║░░░███████╗██║░░██║██║░╚═╝░██║██║██║░╚███║██║░░██║███████╗  
░░░╚═╝░░░╚══════╝╚═╝░░╚═╝╚═╝░░░░░╚═╝╚═╝╚═╝░░╚══╝╚═╝░░╚═╝╚══════╝  
  
   
""")  
  
while True:  
 command = input(">>> ")  
 os.system(command)

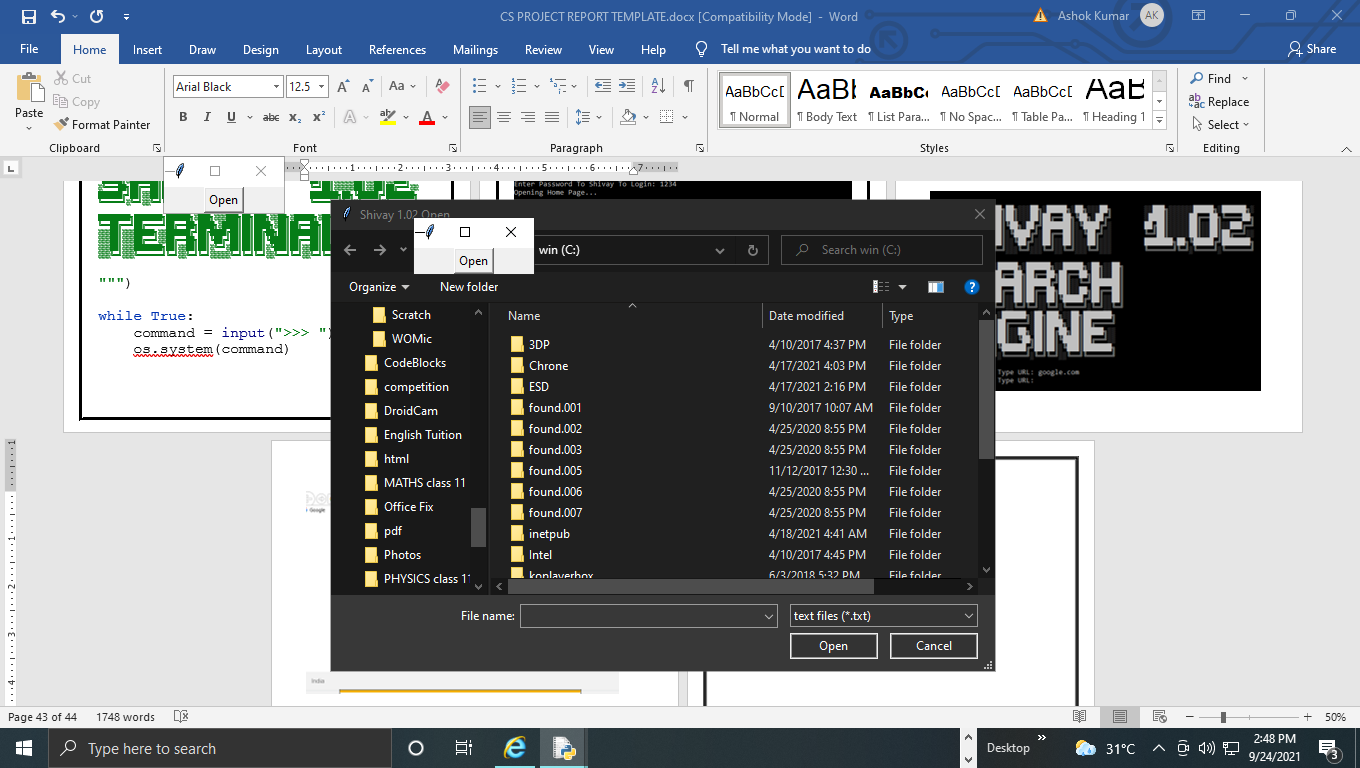


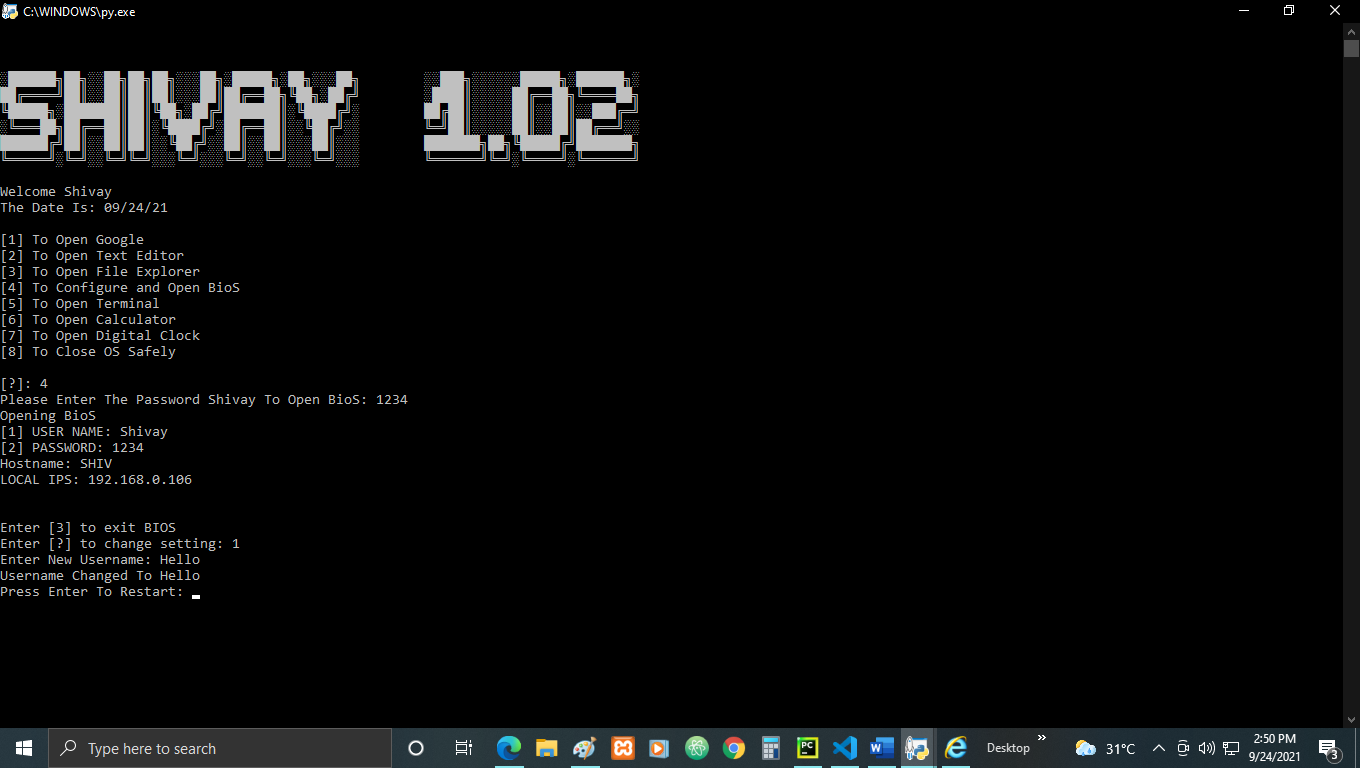


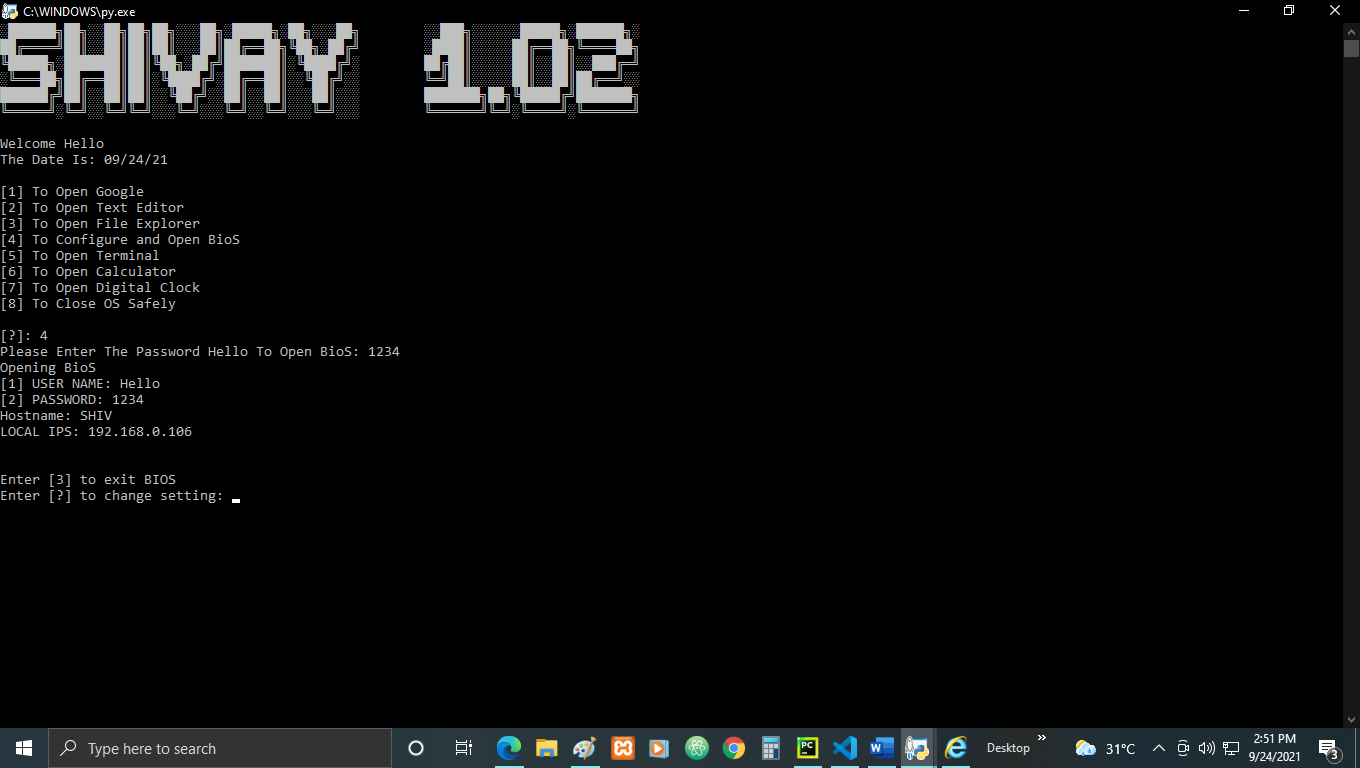


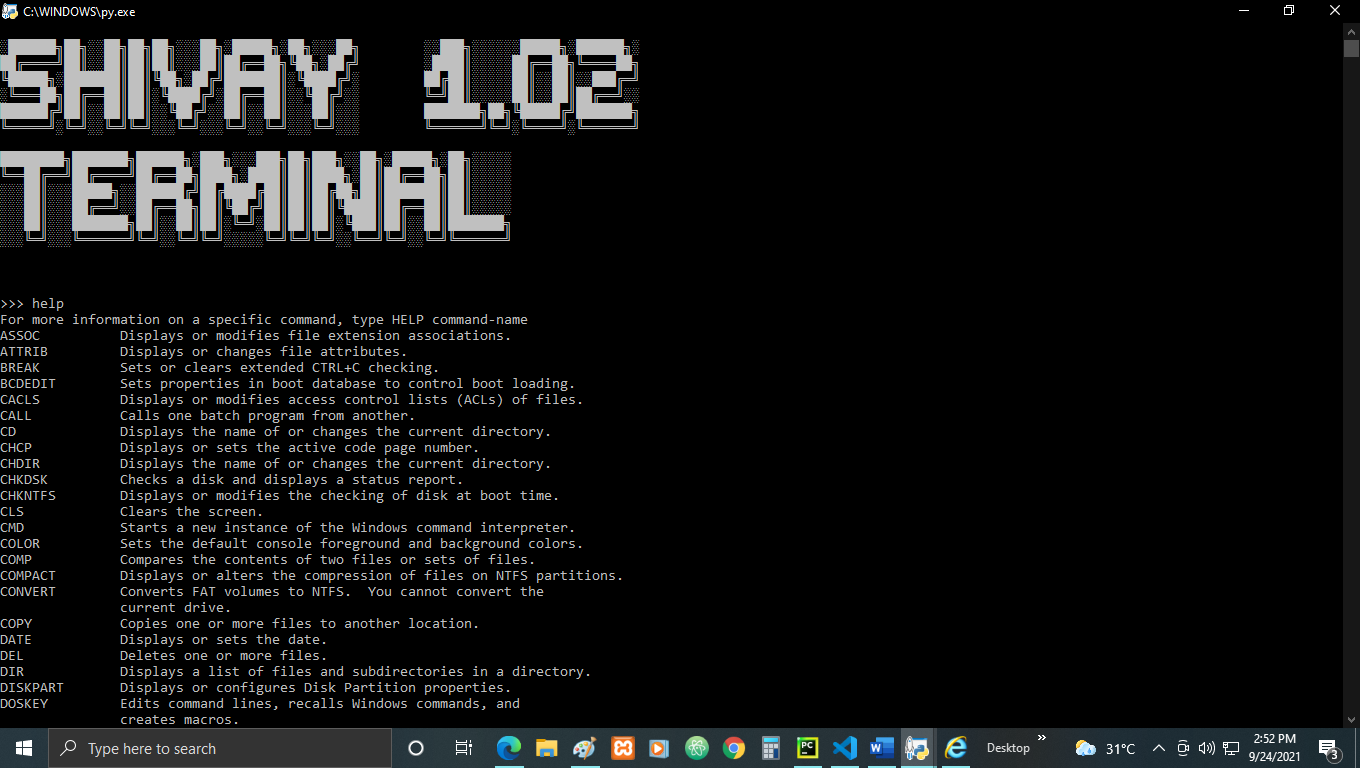


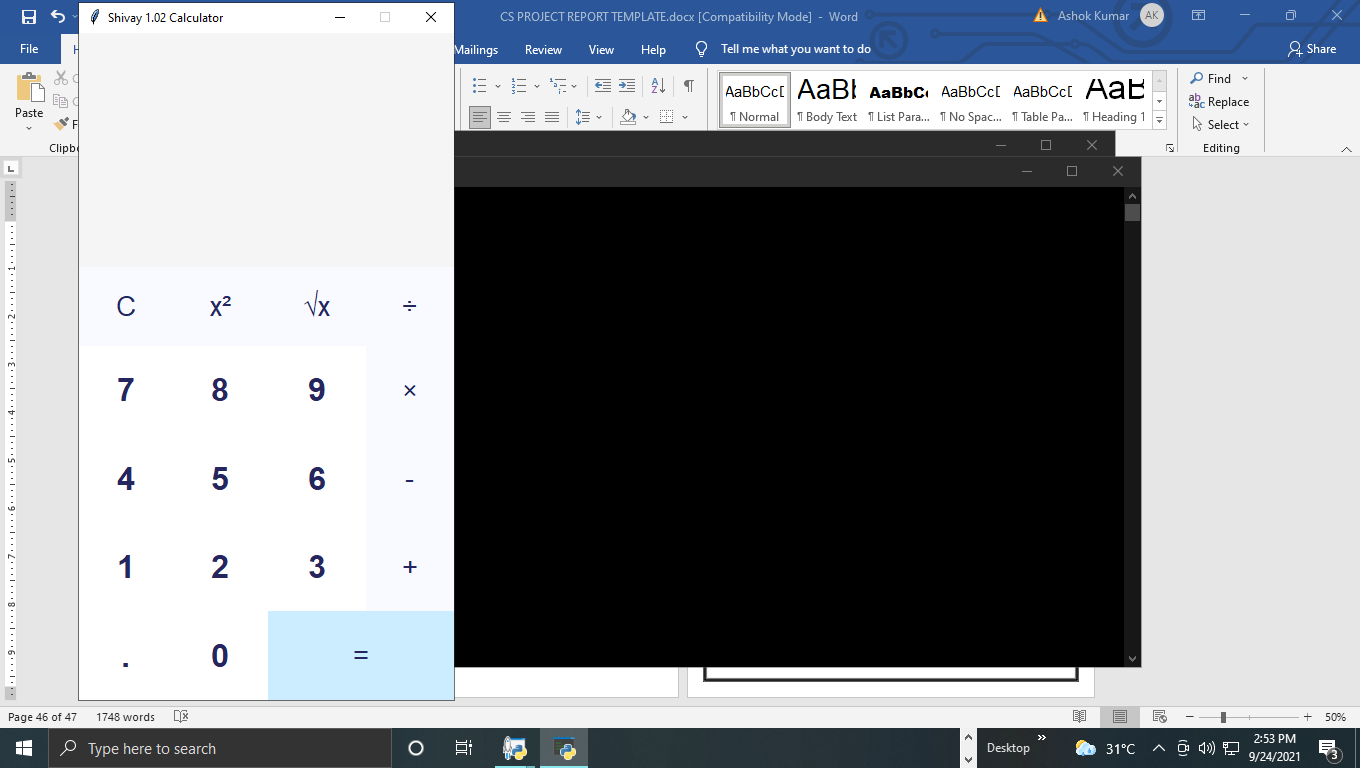


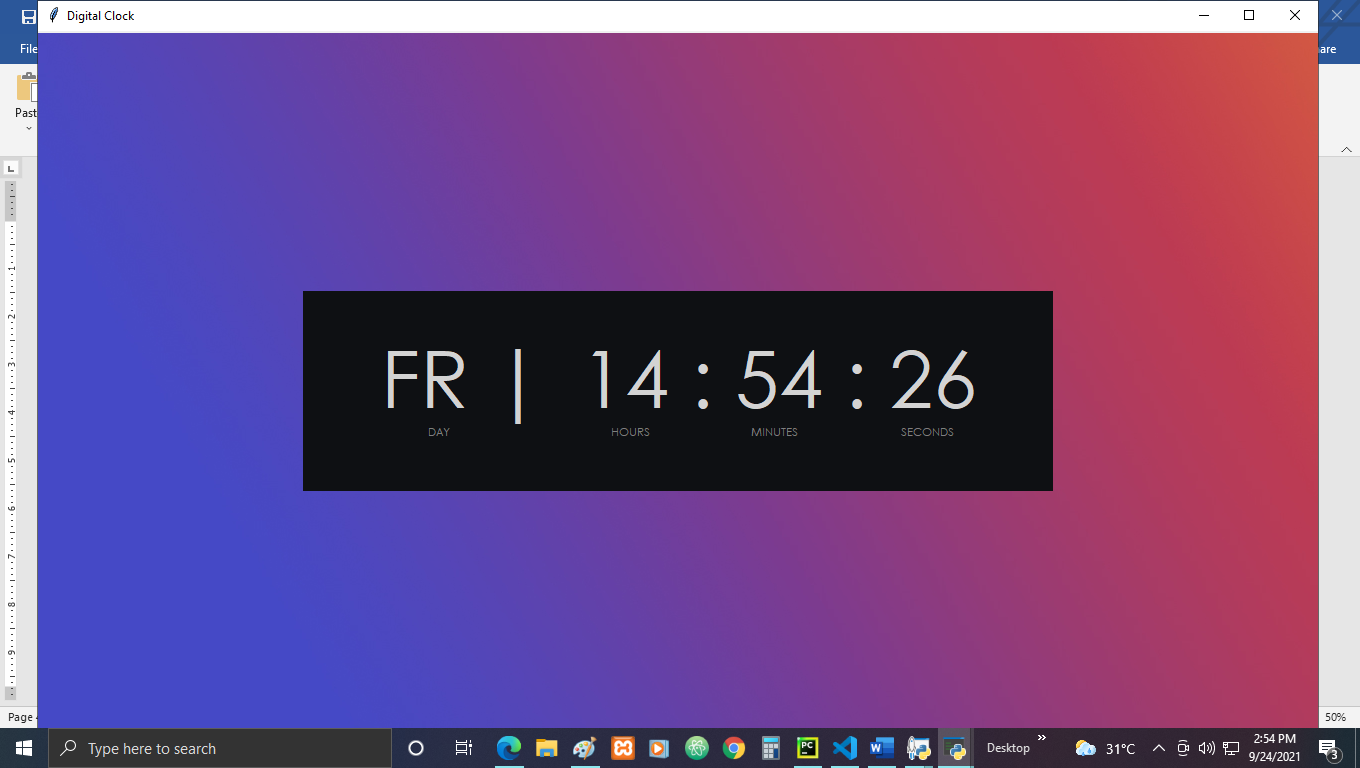












BIBLIOGRAPHY

1. Python.org
2. Geeksforgeeks.com
3. Programiz.com
4. And many more which I have not recorded.