**Goal Manager**

****

A PROJECT REPORT

FOR

STUDENTS

**FOR CBSE 2021 EXAMINATION**

[AS A PART OF THE COMPUTER SCIENCE COURSE (083)]

**ACADEMIC YEAR- 2020-2021**

**SUBMITTED BY**: -

**NAME**: -Nishchal -

-Aneesh Kabra -

-Anay Singh -

**UNDER THE GUIDANCE OF**: Ms. Tarang Malhotra

**MIT LICENSE**

|  |
| --- |
|  |
| Copyright (c) 2020 Nishchal, Aneesh Kabra, Anay Singh  Permission is hereby granted, free of charge, to any person obtaining a copy  of this software and associated documentation files (the "Software"), to deal  in the Software without restriction, including without limitation the rights  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  copies of the Software, and to permit persons to whom the Software is  furnished to do so, subject to the following conditions:  The above copyright notice and this permission notice shall be included in all  copies or substantial portions of the Software.  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  SOFTWARE. |
|  |

**ACKNOWLEDGEMENT**

We undertook this project work, as the part of our XII-Computer Science course(083). We tried our best to apply our knowledge to create a system. However, developing software system is generally a quite complex and time-consuming process. It requires a systematic study, insight vision

and professional approach during the design and development.

Moreover, the developer always feels the need **to help those around them in any way they can.**

We would like to sincerely thank our friends who supported us till the end of this project.

We would like to extend our sincere thanks and gratitude to our teacher, Ms. Tarang Malhotra, for giving valuable time and moral support to develop this

software.

We would like to express our gratitude towards our principal, Ms. Anuradha Handa for extending every possible support for the completion of this project.

We thank everyone who had a hand in the planning and execution of this project.

-Nishchal

Aneesh Kabra

Anay Singh

**CERTIFICATE**

This is to certify that Nishchal, Aneesh Kabra and Anay Singh have of class XII have prepared the report on the project entitled-“**GOAL MANAGER”**. The report is the result of their efforst and endeavours. The report is found worthy of acceptance as their final project report for the subject COMPUTER SCIENCE of subject code 083. They have prepared the report under my guidance:

**Ms. Tarang Malhotra**

**INTRODUCTION**

**GOAL MANAGER** is a python project with SQL connectivity. It is a project to help school students focus and help them track their Goals. Gives a Career path and helps to set and achieve goals and keeps motivating students.

During coding and design of the software Project, **Python IDLE**, as a powerful front-end tool is used for getting Graphical User Interface ,GUI(Using **Tkinter Library**) based integrated platform and coding simplicity. As a back-end a powerful, open source RDBMS, **MySQL** is used as per requirement of the CBSE curriculum of Computer Science Course(083).

**REQUIREMENTS / INSTALLATION**

* Pip install mysql.connector
* Pip install time
* Pip install Json
* Pip install scheduler
* Pip install Tkinter
* Pip install Random(Usually Per-installed )

**MODULES/LIBRARIES USED IN THE PROJECT**

Throughout the development of the project, many python modules have been used.

1. Mysql.connector

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library.

This module does not come built-in with Python and has to be downloaded separately.

1. Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

1. Random

Python offers random module that can generate random numbers. These are pseudo-random numbers. It also contains functions which randomly selects an integer from a range or a random element from a list.

1. Json

JSON stands for JavaScript Object Notation. It is a lightweight data interchange format. It is a language-independent and cross platform text format, supported by many programming languages. This format is used for data exchange between the web server and clients.

JSON format is similar to pickle. However, pickle serialization is Python specific whereas JSON format is implemented by many languages hence has become universal standard. Functionality and interface of json module in Python’s standard library is similar to pickle and marshal modules.

1. Time

The Python time module provides many ways of representing time in code, such as objects, numbers, and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

This module begins the recording time from the epoch. Epoch means time in history and it begins on 1st January 1970.

1. Scheduler

An in-process scheduler for periodic jobs that uses the builder pattern for configuration. Schedule lets you run Python functions (or any other callable) periodically at pre-determined intervals. Schedule Library matches your systems time to that of scheduled time set by you. Once the scheduled time and system time matches the job function (command function that is scheduled) is called.

**PACKAGES USED**

* Database

A python code which helps users create a new id, manage their existing ids and lets admins see all ids. It provides a menu for users to login to their ids also providing them with a guide option. It overall helps us to maintain records of the various users using mysql.connector.

* **Tkdatabse.py-**

import mysql.connector

from tkinter import \*

from tkinter import messagebox

from tkcalendar import Calendar

from quotess import \*

from currenttime import \*

db=mysql.connector.connect(host="localhost",user="root",passwd="251103",database="gmanager")

c = db.cursor()

c.execute("""CREATE TABLE IF NOT EXISTS Students(

ID int(12),

Category varchar(25),

Stream varchar(25),

Name varchar(15),

Qualification varchar(15),

Aim varchar(30),

Skills varchar(30),

password varchar(25))""")

def create\_student\_if\_not\_exists(Id, category, stream, name, qualification, aim, skills,password = 'school@123'):

c.execute("SELECT COUNT(\*) FROM Students WHERE ID=%s",( Id,))

studentCount = c.fetchone()[0]

if studentCount < 1:

c.execute("INSERT INTO Students VALUES (%s, %s, %s, %s, %s, %s, %s,%s)", (Id, category, stream, name, qualification, aim, skills,password))

db.commit()

messagebox.showinfo(title='Thank You', message='User Created, Login from login page')

else:

messagebox.showinfo(title='Sorry', message='User already exists, try another ID')

def impdate():

date = cal.selection\_get()

xnote = note.get()

ID\_temp

c.execute("INSERT INTO DATES VALUES (%s, %s, %s)", (ID\_temp, xnote, date))

db.commit()

messagebox.showinfo(title='Added', message='Event Added')

top.destroy()

def importantevents():

c.execute("""CREATE TABLE IF NOT EXISTS Students(

ID int(12),

Category varchar(25),

Stream varchar(25),

Name varchar(15),

Qualification varchar(15),

Aim varchar(30),

Skills varchar(30),

password varchar(25))""")

c.execute("""CREATE TABLE IF NOT EXISTS DATES(

ID int(12),

NOTE varchar(25),

Imp\_Date DATE)""")

db.commit()

global top

top = Toplevel()

top.geometry("800x650")

global cal

cal = Calendar(top,

font="Arial 14", selectmode='day',

cursor="hand1", day=1, month=1, year=2021,dateformat = 10)

cal.place(rely=0.5, relx=0.68, anchor=CENTER)

global note

note = StringVar(top)

Label(top, text="Enter Label",width = 25,height=2,font=('Helvetica', '14')).place(rely=0.44, relx=0.24, anchor=CENTER)

ID = Entry(top,width="25",textvariable =note)

ID.place(rely=0.5, relx=0.25, anchor=CENTER)

Button(top, text="Done", width = 20,height=2,font=('Helvetica', '14'), command=impdate).place(relx = 0.4, rely = 0.70)

top.mainloop()

def showallevent():

tempwin = Toplevel()

tempwin.title("My Events")

tempwin.geometry("800x650")

c.execute("SELECT NOTE,Imp\_Date FROM DATES WHERE ID=%s",(ID\_temp,))

eventdata = c.fetchall()

y = 0.12

x = 0.22

z = 1

Sno = 1

for i in eventdata:

Label(tempwin, text=('Event',Sno,':',i[0],'on',i[1]),height=3,font=('Helvetica', '14')).place(rely=y, relx=x, anchor=CENTER)

Sno +=1

y += 0.08

z +=1

if z >= 12:

x +=0.40

z = 1

y = 0.12

def get\_student\_from\_database(Id,xpassword):

global ID\_temp

ID\_temp = Id

c.execute("SELECT \* FROM Students WHERE ID=%s", (Id,))

studentdata = c.fetchone()

if studentdata[7] == xpassword:

global studentmainpage

studentmainpage = Toplevel()

studentmainpage.geometry("800x650")

studentmainpage.title("Home")

filename = PhotoImage(file = "homepage-background.png")

background\_label = Label(studentmainpage, image=filename)

background\_label.place(x=0, y=0, relwidth=1, relheight=1)

userID = Label( studentmainpage,font=('Helvetica', '15'),text = f'''Login ID: {studentdata[0]}''')

userID.place(rely=0.04, relx=0.10, anchor=CENTER)

NameD = Label( studentmainpage,font=('Helvetica', '15'),text = f'''Welcome, {studentdata[3]}''')

NameD.place(rely=0.03, relx=0.84, anchor=CENTER)

aimD = Label( studentmainpage,font=('Helvetica', '15'),text = f'''Aim = {studentdata[5]}''')

aimD.place(rely=0.2, relx=0.5, anchor=CENTER)

quotebutton = Button(studentmainpage, width = 10,height=1,text='QUOTE',bg='black',

fg='white',font=('Helvetica', '14'),command=Quotez,)

quotebutton.place(rely=0.10, relx=0.84)

ctime = Button(studentmainpage, width = 10,height=1,text='Current Time',bg='black',

fg='white',font=('Helvetica', '14'),command=currenttimee,)

ctime.place(rely=0.18, relx=0.84)

cimportantevents = Button(studentmainpage,height=1,text='Add Important Events',bg='black',

fg='white',font=('Helvetica', '14'),command= importantevents,)

cimportantevents.place(rely=0.26, relx=0.75)

showevents = Button(studentmainpage, width = 10,height=1,text='My Events',bg='black',

fg='white',font=('Helvetica', '14'),command= showallevent,)

showevents.place(rely=0.34, relx=0.84)

studentmainpage.mainloop()

else:

messagebox.showwarning(title='Incorrect Password', message='Kindly check your password or Contact Admin')

* **tkmenu.py-**

from tkinter import \*

from TKdatabase import \*

from time import \*

from Admindb import \*

#def masterpage():

root = Tk()

root.geometry("800x650")

root.title('Home Page')

filename = PhotoImage(file = "homepage-background.png")

background\_label = Label(root, image=filename)

background\_label.place(x=0, y=0, relwidth=1, relheight=1)

menubar = Menu(root)

filemenu = Menu(menubar, tearoff=0)

filemenu.add\_command(label="Detailed chart about each field",command=None)

filemenu.add\_command(label="Issue with creating new ID",command=None)

filemenu.add\_command(label="Help with different features",command=None )

filemenu.add\_command(label="New Feature Updates", command=None)

filemenu.add\_separator()

filemenu.add\_command(label="Exit", command=root.destroy)

menubar.add\_cascade(label="Guide", menu=filemenu)

filemenu2 = Menu(menubar, tearoff=0)

filemenu2.add\_command(label="Contact Admin",command=None)

filemenu2.add\_command(label="Contact Teacher",command=None )

filemenu2.add\_command(label="Contact counselor",command=None )

filemenu2.add\_command(label="Contact Developer",command=None )

menubar.add\_cascade(label="Contact Request ", menu=filemenu2)

def submitstudent():

global xID,xpassword

xID = user.get()

xpassword = passw.get()

get\_student\_from\_database(xID,xpassword)

def submitadmin():

global a\_ID,a\_password

a\_ID = user.get()

a\_password = passw.get()

adminpower(a\_ID,a\_password)

def createstudent():

global ID, category, stream, name, qualification, aim, skills

ID = xID.get()

category = xcategory.get()

stream = xstream.get()

name = xname.get()

qualification = xqualification.get()

aim = xaim.get()

skills = xskills.get()

password = xpassword.get()

create\_student\_if\_not\_exists(ID, category, stream, name, qualification, aim, skills,password)

def create\_id():

subwindow1 = Toplevel()

subwindow1.geometry("800x650")

subwindow1.title("New User")

menubar = Menu(subwindow1)

filemenu = Menu(menubar, tearoff=0)

filemenu.add\_command(label="Login",command=loginpg )

filemenu.add\_separator()

filemenu.add\_command(label="Exit", command=subwindow1.destroy)

menubar.add\_cascade(label="Main Page ", menu=filemenu)

filename = PhotoImage(file = "homepage-background.png")

s1background\_label = Label(subwindow1, image=filename)

s1background\_label.place(x=0, y=0, relwidth=1, relheight=1)

global xID, xcategory, xstream, xname, xqualification, xaim, xskills, xpassword

xID = IntVar(subwindow1)

xcategory = StringVar(subwindow1)

xstream = StringVar(subwindow1)

xname = StringVar(subwindow1)

xqualification = StringVar(subwindow1)

xaim = StringVar(subwindow1)

xskills = StringVar(subwindow1)

xpassword = StringVar(subwindow1)

Label(subwindow1, text="Enter New ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.24, relx=0.42, anchor=CENTER)

mID = Entry(subwindow1,width="25",textvariable =xID)

mID.place(rely=0.24, relx=0.67, anchor=CENTER)

Label(subwindow1, text="Enter Category",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.32, relx=0.42, anchor=CENTER)

mcategory = Entry(subwindow1,width="25",textvariable =xcategory)

mcategory.place(rely=0.32, relx=0.67, anchor=CENTER)

Label(subwindow1, text="Enter your Stream",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

mstream = Entry(subwindow1,width="25",textvariable =xstream)

mstream.place(rely=0.40, relx=0.67, anchor=CENTER)

Label(subwindow1, text="Enter your name",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.48, relx=0.42, anchor=CENTER)

mname = Entry(subwindow1,width="25",textvariable =xname)

mname.place(rely=0.48, relx=0.67, anchor=CENTER)

Label(subwindow1, text="Enter your Qualification",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.56, relx=0.42, anchor=CENTER)

mqualification = Entry(subwindow1,width="25",textvariable =xqualification)

mqualification.place(rely=0.56, relx=0.67, anchor=CENTER)

Label(subwindow1, text="Enter your Aim",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.64, relx=0.42, anchor=CENTER)

maim = Entry(subwindow1,width="25",textvariable =xaim)

maim.place(rely=0.64, relx=0.67, anchor=CENTER)

Label(subwindow1, text="Enter your skills",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.72, relx=0.42, anchor=CENTER)

mskills = Entry(subwindow1,width="25",textvariable =xskills)

mskills.place(rely=0.72, relx=0.67, anchor=CENTER)

Label(subwindow1, text="Enter a STRONG password",width = 21,height=2,font=('Helvetica', '14')).place(rely=0.80, relx=0.42, anchor=CENTER)

mpassword = Entry(subwindow1,width="25",textvariable =xpassword, show = '\*')

mpassword.place(rely=0.80, relx=0.67, anchor=CENTER)

submit = Button(subwindow1, text ="Create ID",

command = createstudent, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.88)

subwindow1.config(menu=menubar)

subwindow1.mainloop()

def adminpg():

win = Toplevel()

win.geometry("800x650")

win.title('Admin Login')

filename = PhotoImage(file = "abcdef.png")

background\_label = Label(win, image=filename)

background\_label.place(x=0, y=0, relwidth=1, relheight=1)

global user,passw

user = IntVar(win)

passw = StringVar(win)

Label(win, text="Enter Admin ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

ID = Entry(win,width="25",textvariable =user)

ID.place(rely=0.40, relx=0.67, anchor=CENTER)

Label(win, text="Enter Passcode",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

password= Entry(win,textvariable = passw,width="25",show = '\*')

password.place(rely=0.52, relx=0.67, anchor=CENTER)

submit = Button(win, text ="Login",

command = submitadmin, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

win.mainloop()

def studentpg():

if subwindow2.winfo\_exists() == True:

subwindow2.destroy()

global win1

win1 = Toplevel()

win1.geometry("800x650")

win1.title('Student World')

filename = PhotoImage(file = "abcdef.png")

background\_label = Label(win1, image=filename)

background\_label.place(x=0, y=0, relwidth=1, relheight=1)

global user,passw

user = IntVar(win1)

passw = StringVar(win1)

Label(win1, text="Enter your Unique ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

ID = Entry(win1,width="25",textvariable =user)

ID.place(rely=0.40, relx=0.67, anchor=CENTER)

Label(win1, text="Enter your Password",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

password= Entry(win1,textvariable = passw,width="25",show = '\*')

password.place(rely=0.52, relx=0.67, anchor=CENTER)

submit = Button(win1, text ="Login",

command = submitstudent, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

win1.mainloop()

def loginpg():

global subwindow2

subwindow2 = Toplevel()

subwindow2.geometry("800x650")

subwindow2.title("Login")

menubar = Menu(subwindow2)

filemenu = Menu(menubar, tearoff=0)

filemenu.add\_command(label="SignUp",command=create\_id )

filemenu.add\_separator()

filemenu.add\_command(label="Exit", command=subwindow2.destroy)

menubar.add\_cascade(label="Main Page ", menu=filemenu)

filename = PhotoImage(file = "loginp.png")

s2background\_label = Label(subwindow2, image=filename)

s2background\_label.place(x=0, y=0, relwidth=1, relheight=1)

widgt1 = Label(subwindow2,text='Login as?',bg='#ADD8E6')

widgt1.config(font=("Courier", 44))

widgt1.pack()

S2button1 = Button(subwindow2, width = 25,height=3,text='Admin',bg='black', fg='white',

font=('Helvetica', '14'),command=adminpg,)

S2button1.place(rely=0.40, relx=0.5, anchor=CENTER)

S2button2 = Button(subwindow2, width = 25,height=3,text='Student',bg='black',

fg='white',font=('Helvetica', '14') ,command=studentpg )

S2button2.place(rely=0.57, relx=0.5, anchor=CENTER)

subwindow2.config(menu=menubar)

subwindow2.mainloop()

button1 = Button(root, text='Create ID',width = 25,height=3,font=('Helvetica', '14'),

command=create\_id,)

button1.place(rely=0.40, relx=0.5, anchor=CENTER)

button2 = Button(root, width = 25,height=3,text='Login',font=('Helvetica', '14')

, command=loginpg)

button2.place(rely=0.57, relx=0.5, anchor=CENTER)

root.config(menu=menubar)

root.mainloop()

* **Admin.py-**

import mysql.connector

from tkinter import \*

from tkinter import messagebox

db=mysql.connector.connect(host="localhost",user="root",passwd="251103",database="gmanager")

c = db.cursor()

'''c.execute("""Create table if not exists admin\_details(

admin\_username varchar(15),

admin\_id int(12),

admin\_passwords varchar(25))""")'''

def submitfinal():

if sql\_task == 'category':

c.execute('''update students

set Category=%s

where ID=%s''', (cat.get(),idd.get()))

elif sql\_task == 'stream':

c.execute('''update students

set stream=%s

where ID=%s''', (cat.get(),idd.get()))

elif sql\_task == 'name':

c.execute('''update students

set name=%s

where ID=%s''', (cat.get(),idd.get()))

elif sql\_task == 'qualification':

c.execute('''update students

set Qualification=%s

where ID=%s''', (cat.get(),idd.get()))

elif sql\_task == 'aim':

c.execute('''update students

set aim=%s

where ID=%s''', (cat.get(),idd.get()))

elif sql\_task == 'skills':

c.execute('''update students

set skills=%s

where ID=%s''', (cat.get(),idd.get()))

elif sql\_task == 'password':

c.execute('''update students

set password=%s

where ID=%s''', (cat.get(),idd.get()))

elif sql\_task == 'All ':

print('abc')

elif sql\_task == 'delete':

print('abc')

db.commit()

messagebox.showinfo(title='Sucessful', message='data modified successfully')

tempwin.destroy()

def update\_category():

global tempwin

tempwin = Toplevel()

tempwin.title("Modify")

tempwin.geometry("800x650")

global cat, idd

idd = IntVar(tempwin)

cat = StringVar(tempwin)

Label(tempwin, text="Enter new category",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

category = Entry(tempwin,width="25",textvariable = cat)

category.place(rely=0.40, relx=0.64, anchor=CENTER)

Label(tempwin, text="Enter ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

IDa= Entry(tempwin,width="25",textvariable = idd)

IDa.place(rely=0.52, relx=0.64, anchor=CENTER)

submit = Button(tempwin, text ="Update",

command = submitfinal, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

global sql\_task

sql\_task = 'category'

def update\_stream():

global tempwin

tempwin = Toplevel()

tempwin.title("Modify")

tempwin.geometry("800x650")

global cat, idd

idd = IntVar(tempwin)

cat = StringVar(tempwin)

Label(tempwin, text="Enter new Stream",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

category = Entry(tempwin,width="25",textvariable = cat)

category.place(rely=0.40, relx=0.64, anchor=CENTER)

Label(tempwin, text="Enter ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

IDa= Entry(tempwin,width="25",textvariable = idd)

IDa.place(rely=0.52, relx=0.64, anchor=CENTER)

submit = Button(tempwin, text ="Update",

command = submitfinal, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

global sql\_task

sql\_task = 'stream'

def update\_name():

global tempwin

tempwin = Toplevel()

tempwin.title("Modify")

tempwin.geometry("800x650")

global cat, idd

idd = IntVar(tempwin)

cat = StringVar(tempwin)

Label(tempwin, text="Eenter Corrected Name",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

category = Entry(tempwin,width="25",textvariable = cat)

category.place(rely=0.40, relx=0.64, anchor=CENTER)

Label(tempwin, text="Enter ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

IDa= Entry(tempwin,width="25",textvariable = idd)

IDa.place(rely=0.52, relx=0.64, anchor=CENTER)

submit = Button(tempwin, text ="Update",

command = submitfinal, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

global sql\_task

sql\_task = 'name'

def update\_qualification():

global tempwin

tempwin = Toplevel()

tempwin.title("Modify")

tempwin.geometry("800x650")

global cat, idd

idd = IntVar(tempwin)

cat = StringVar(tempwin)

Label(tempwin, text="enter updated qualification",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

category = Entry(tempwin,width="25",textvariable = cat)

category.place(rely=0.40, relx=0.64, anchor=CENTER)

Label(tempwin, text="Enter ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

IDa= Entry(tempwin,width="25",textvariable = idd)

IDa.place(rely=0.52, relx=0.64, anchor=CENTER)

submit = Button(tempwin, text ="Update",

command = submitfinal, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

global sql\_task

sql\_task = 'qualification'

def update\_aim():

global tempwin

tempwin = Toplevel()

tempwin.title("Modify")

tempwin.geometry("800x650")

global cat, idd

idd = IntVar(tempwin)

cat = StringVar(tempwin)

Label(tempwin, text="enter your latest aim",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

category = Entry(tempwin,width="25",textvariable = cat)

category.place(rely=0.40, relx=0.64, anchor=CENTER)

Label(tempwin, text="Enter ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

IDa= Entry(tempwin,width="25",textvariable = idd)

IDa.place(rely=0.52, relx=0.64, anchor=CENTER)

submit = Button(tempwin, text ="Update",

command = submitfinal, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

global sql\_task

sql\_task = 'aim'

def update\_skills():

global tempwin

tempwin = Toplevel()

tempwin.title("Modify")

tempwin.geometry("800x650")

global cat, idd

idd = IntVar(tempwin)

cat = StringVar(tempwin)

Label(tempwin, text="enter all your skills",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

category = Entry(tempwin,width="25",textvariable = cat)

category.place(rely=0.40, relx=0.64, anchor=CENTER)

Label(tempwin, text="Enter ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

IDa= Entry(tempwin,width="25",textvariable = idd)

IDa.place(rely=0.52, relx=0.64, anchor=CENTER)

submit = Button(tempwin, text ="Update",

command = submitfinal, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

global sql\_task

sql\_task = 'skills'

def update\_password():

global tempwin

tempwin = Toplevel()

tempwin.title("Modify")

tempwin.geometry("800x650")

global cat, idd

idd = IntVar(tempwin)

cat = StringVar(tempwin)

Label(tempwin, text="enter new password",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

category = Entry(tempwin,width="25",textvariable = cat)

category.place(rely=0.40, relx=0.64, anchor=CENTER)

Label(tempwin, text="Enter ID",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.52, relx=0.42, anchor=CENTER)

IDa= Entry(tempwin,width="25",textvariable = idd)

IDa.place(rely=0.52, relx=0.64, anchor=CENTER)

submit = Button(tempwin, text ="Update",

command = submitfinal, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

global sql\_task

sql\_task = 'password'

def all\_details():

global tempwin

tempwin = Toplevel()

tempwin.title("ID's")

tempwin.geometry("800x650")

c.execute("SELECT Id FROM Students ")

studentdata = c.fetchall()

y = 0.17

x = 0.14

z = 1

for i in studentdata:

Label(tempwin, text=('ID:',i),width = 15,height=2,font=('Helvetica', '14')).place(rely=y, relx=x, anchor=CENTER)

y += 0.05

z +=1

if z >= 9:

x +=0.20

z = 1

y = 0.17

def deleteacc():

c.execute("DELETE FROM students where Id=%s",(idd.get(),))

db.commit()

messagebox.showinfo(title='Sucessful', message='ID deleted successfully')

tempwin.destroy()

def delete\_student():

global tempwin

tempwin = Toplevel()

tempwin.title("Delete Account")

tempwin.geometry("800x650")

global idd

idd = IntVar(tempwin)

Label(tempwin, text="enter ID to delete",width = 20,height=2,font=('Helvetica', '14')).place(rely=0.40, relx=0.42, anchor=CENTER)

idx = Entry(tempwin,width="25",textvariable = idd)

idx.place(rely=0.40, relx=0.64, anchor=CENTER)

submit = Button(tempwin, text ="Delete",

command = deleteacc, width = 20,height=2,font=('Helvetica', '14'))

submit.place(relx = 0.67, rely = 0.60)

def updatecmd():

global subadmn

subadmn = Toplevel()

subadmn.geometry("800x650")

subadmn.title("Update")

filename1 = PhotoImage(file = "abcdef.png")

background\_label1 = Label(subadmn, image=filename1)

background\_label1.place(x=0, y=0, relwidth=1, relheight=1)

button2=Button(subadmn,text='Update Category',width=20, height=1,font=('Helvetica', '14'),command=update\_category,)

button2.place(rely=0.24, relx=0.5, anchor=CENTER)

button3=Button(subadmn,text='Update Stream',width=20, height=1,font=('Helvetica', '14'),command=update\_stream,)

button3.place(rely=0.32, relx=0.5, anchor=CENTER)

button4=Button(subadmn,text='Update Name',width=20, height=1,font=('Helvetica', '14'),command=update\_name,)

button4.place(rely=0.40, relx=0.5, anchor=CENTER)

button5=Button(subadmn,text='Update Qualification',width=20, height=1,font=('Helvetica', '14'),command=update\_qualification,)

button5.place(rely=0.48, relx=0.5, anchor=CENTER)

button6=Button(subadmn,text='Update Aim',width=20, height=1,font=('Helvetica', '14'),command=update\_aim,)

button6.place(rely=0.56, relx=0.5, anchor=CENTER)

button7=Button(subadmn,text='Update Skills',width=20, height=1,font=('Helvetica', '14'),command=update\_skills,)

button7.place(rely=0.64, relx=0.5, anchor=CENTER)

button8=Button(subadmn,text='Update Password',width=20, height=1,font=('Helvetica', '14'),command=update\_password,)

button8.place(rely=0.72, relx=0.5, anchor=CENTER)

def adminpower(a\_ID,a\_password):

c.execute("SELECT \* FROM ADMIN\_DETAILS WHERE admin\_id=%s",(a\_ID,))

admindata= c.fetchone()

if admindata[2]==a\_password:

global admn

admn = Toplevel()

admn.geometry("800x650")

admn.title("Admin Controls")

filename = PhotoImage(file = "homepage-background.png")

background\_label = Label(admn, image=filename)

background\_label.place(x=0, y=0, relwidth=1, relheight=1)

button2=Button(admn,text='All ID', width=25, height=2,font=('Helvetica', '14'),command=all\_details,)

button2.place(rely=0.32, relx=0.5, anchor=CENTER)

button3=Button(admn,text='Delete an ID', width=25, height=2,font=('Helvetica', '14'),command=delete\_student,)

button3.place(rely=0.44, relx=0.5, anchor=CENTER)

button1=Button(admn,text='Modify the Account', width=25, height=2,font=('Helvetica', '14'),command=updatecmd,)

button1.place(rely=0.56, relx=0.5, anchor=CENTER)

admn.mainloop()

else:

messagebox.showwarning(title='Incorrect Password', message='Kindly check you password or Contact Admin')

* **Calendar.py-**

import tkinter as tk

from tkinter import ttk

from tkcalendar import Calendar, DateEntry

root = tk.Tk()

top = tk.Toplevel(root)

cal = Calendar(top,

font="Arial 14", selectmode='day',

cursor="hand1", year=2018, month=2, day=5)

cal.pack(fill="both", expand=True)

def getdate():

print(cal.selection\_get())

ttk.Button(top, text="ok", command=getdate).pack()

root.mainloop()

Currenttime.py-

from datetime import datetime

import pygame

def currenttimee():

try:

pygame.init()

#icon = pygame.image.load('digitalClock.png')

#pygame.display.set\_icon(icon)

screen = pygame.display.set\_mode((430,180))

pygame.display.set\_caption('Digital Clock')

bigFont = pygame.font.SysFont('DS-Digital',130)#Comic Sans MS

smallFont = pygame.font.SysFont('DS-Digital',30)

white = (255,255,255)

black = (0,0,0)

green = (0,255,0)

months = ['January', 'February', 'March', 'April', 'May',

'June', 'July', 'August', 'September', 'October', 'November', 'December']

days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday' ]

running = True

while running:

screen.fill(black)

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

pygame.quit()

now = datetime.now()

today = datetime.today()

minute = now.strftime('%M:%S')

hour = int(now.strftime('%H'))

month = int(now.strftime('%m'))

year = now.strftime("%d %Y")

day = today.weekday()

day = days[day]

month = months[month-1]

am = 'AM'

if hour > 12:

hour = hour-12

am = 'PM'

time = f'{hour}:{minute}'

timeText = bigFont.render(time,True,green)

monthText = smallFont.render(month,True,green)

yearText = smallFont.render(year,True,green)

amText = smallFont.render(am,True,green)

dayText = smallFont.render(day,True,green)

screen.blit(timeText, (15,20))

screen.blit(monthText, (15,150))

screen.blit(yearText, (145,150))

screen.blit(amText,(380,0))

screen.blit(dayText,(280,150))

pygame.display.update()

except:

return None

* **Quotes.py-**

import json

import random

from tkinter import \*

def Quotez():

with open('data.json') as f:

data = json.load(f)

counter = 0

for i in data['quote']:

if i:

counter += 1

quoteno= random.randint(0,counter-1)

quote = data['quote'][quoteno]

for y in quote:

authorname = y

x = quote[y] + '\n -by ' + y

quotewindow = Toplevel()

quotewindow.title('Quote for you to stay motivated')

quotewindow.geometry("300x150")

text = Text(quotewindow)

text.insert(INSERT, x)

text.pack(side='top', fill='both', expand='yes')

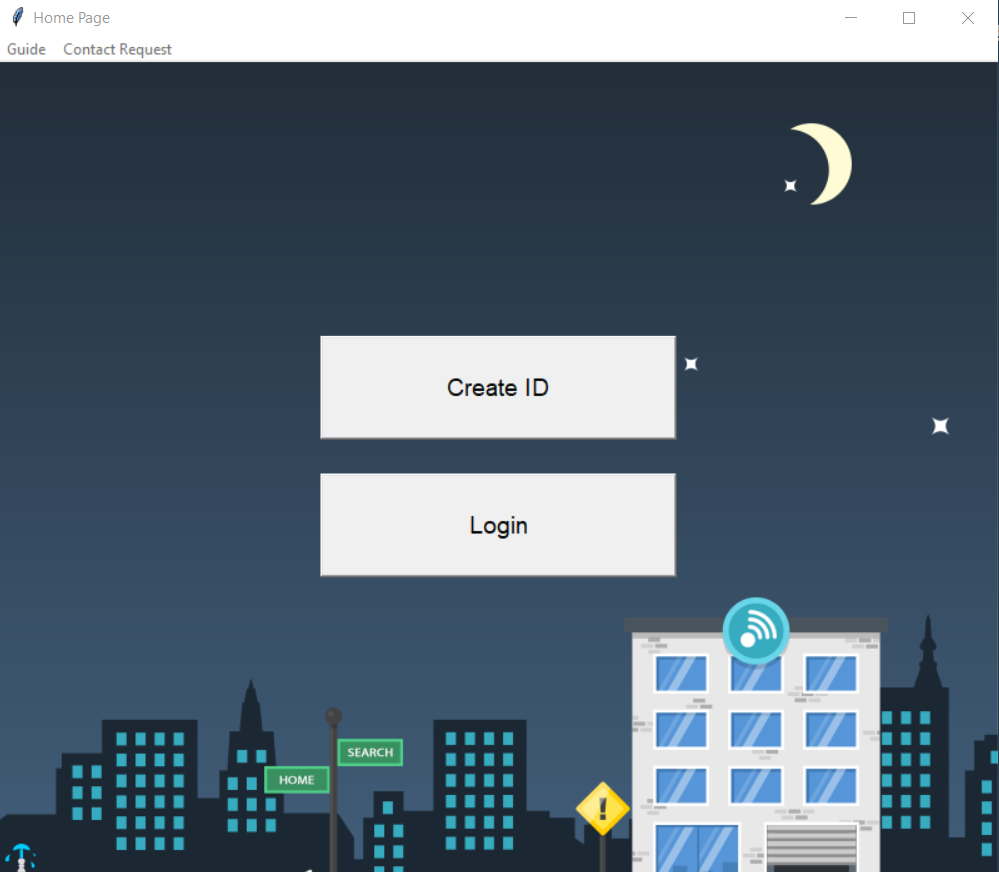
menubar = Menu(quotewindow)

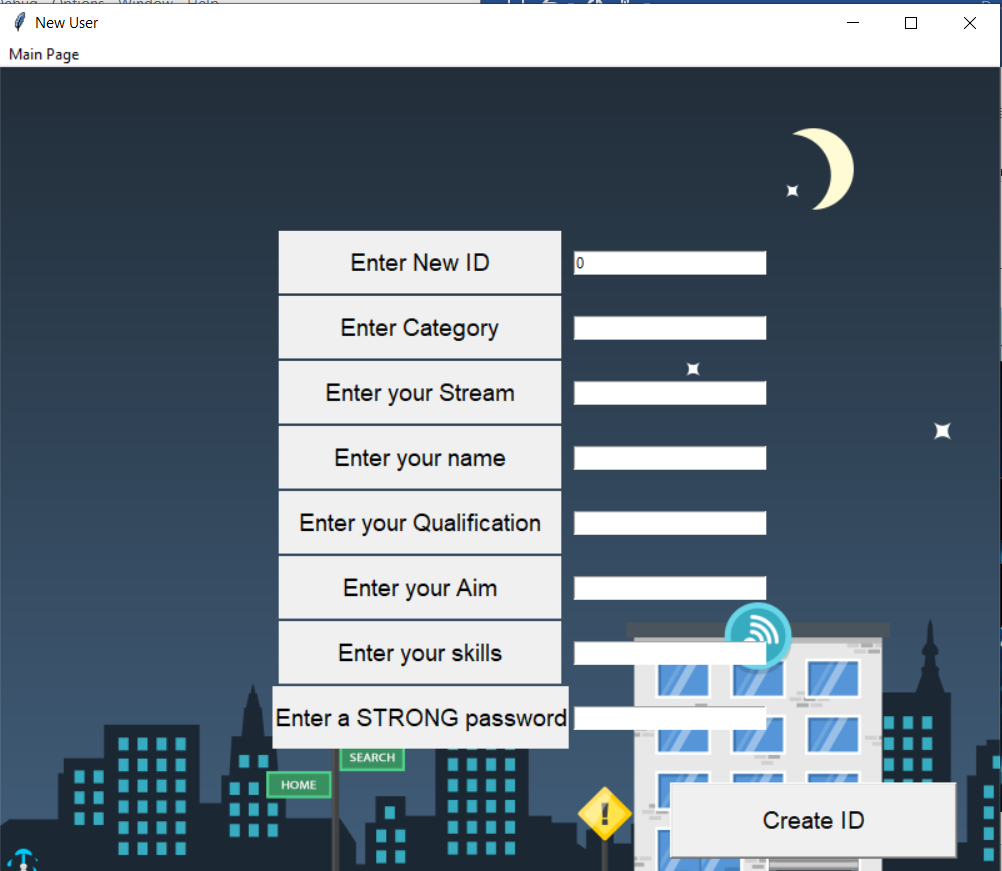
filemenu = Menu(menubar, tearoff=0)

menubar.add\_cascade(label="Exit",command=quotewindow.destroy)

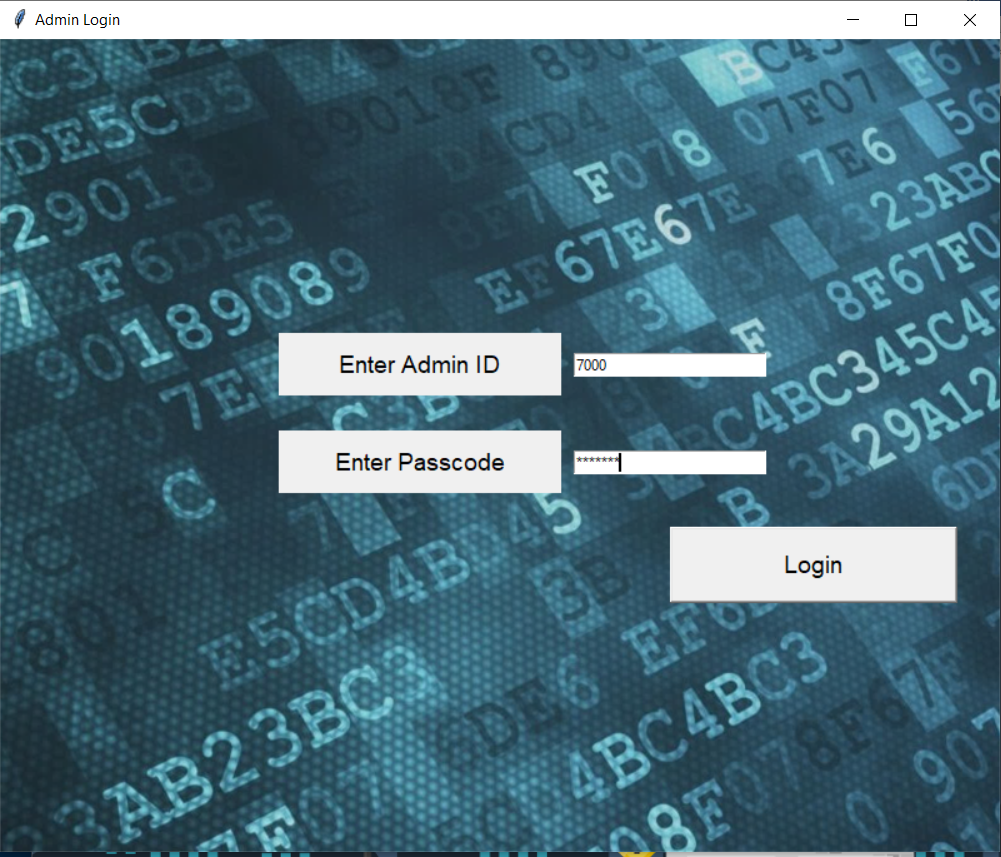
quotewindow.config(menu=menubar)

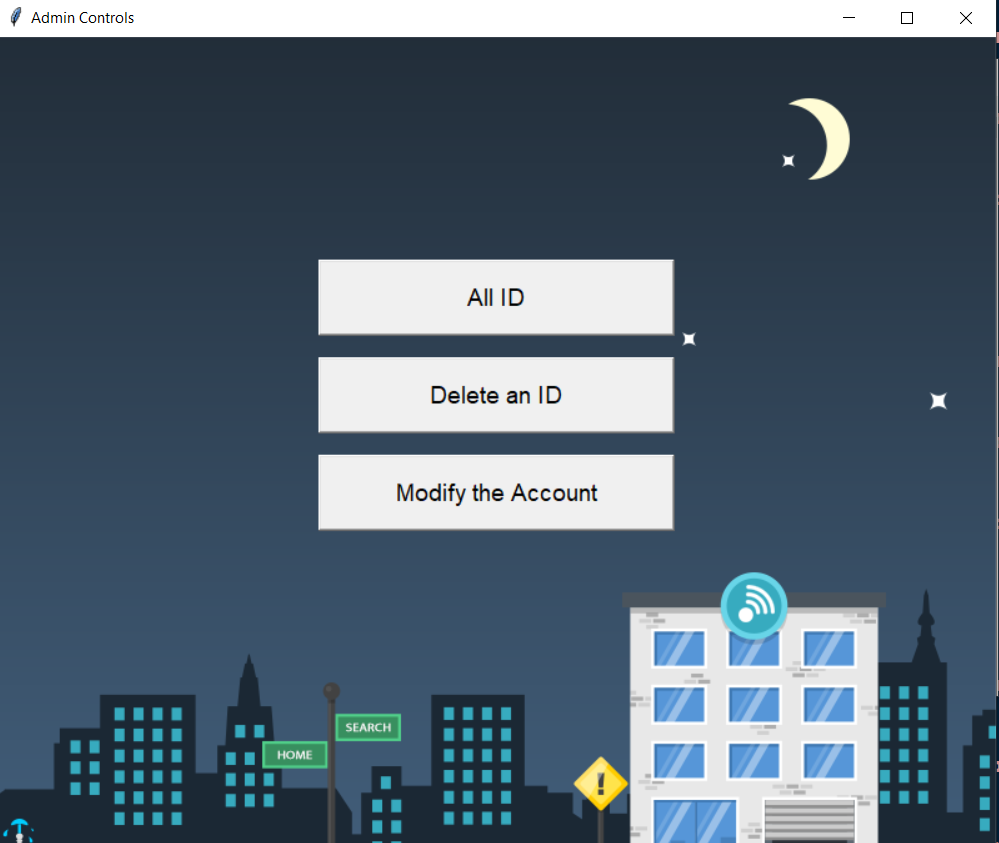
quotewindow.mainloop()

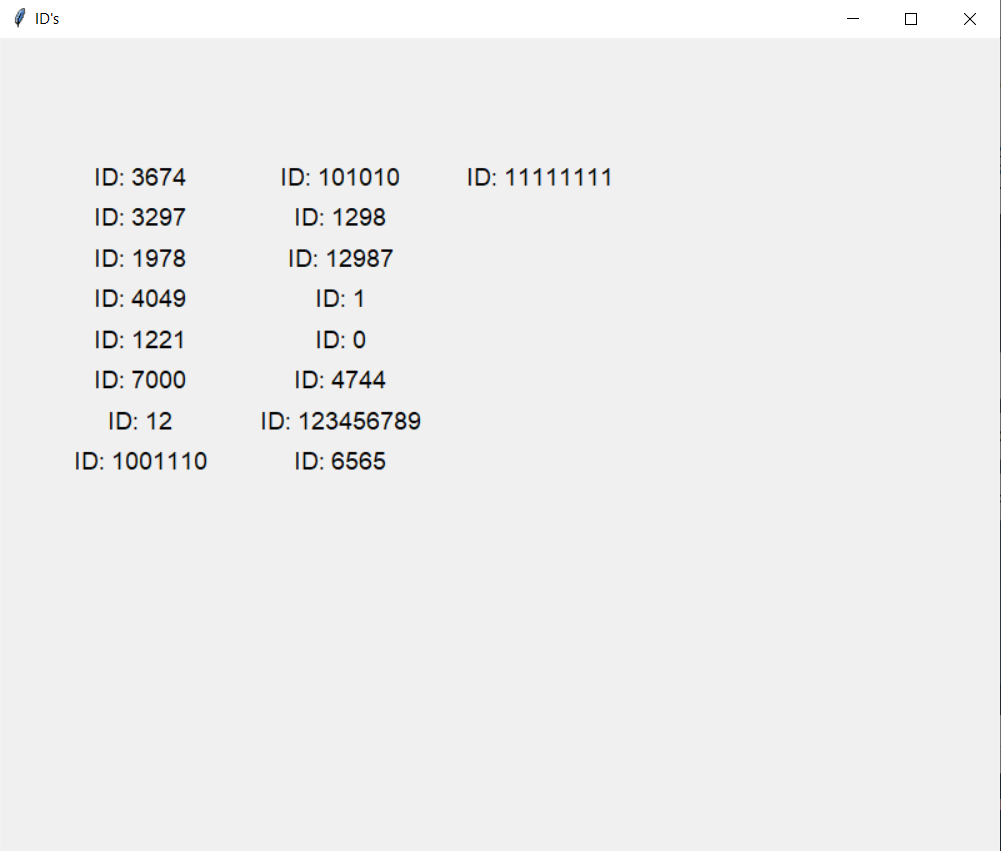
**OUTPUT-**

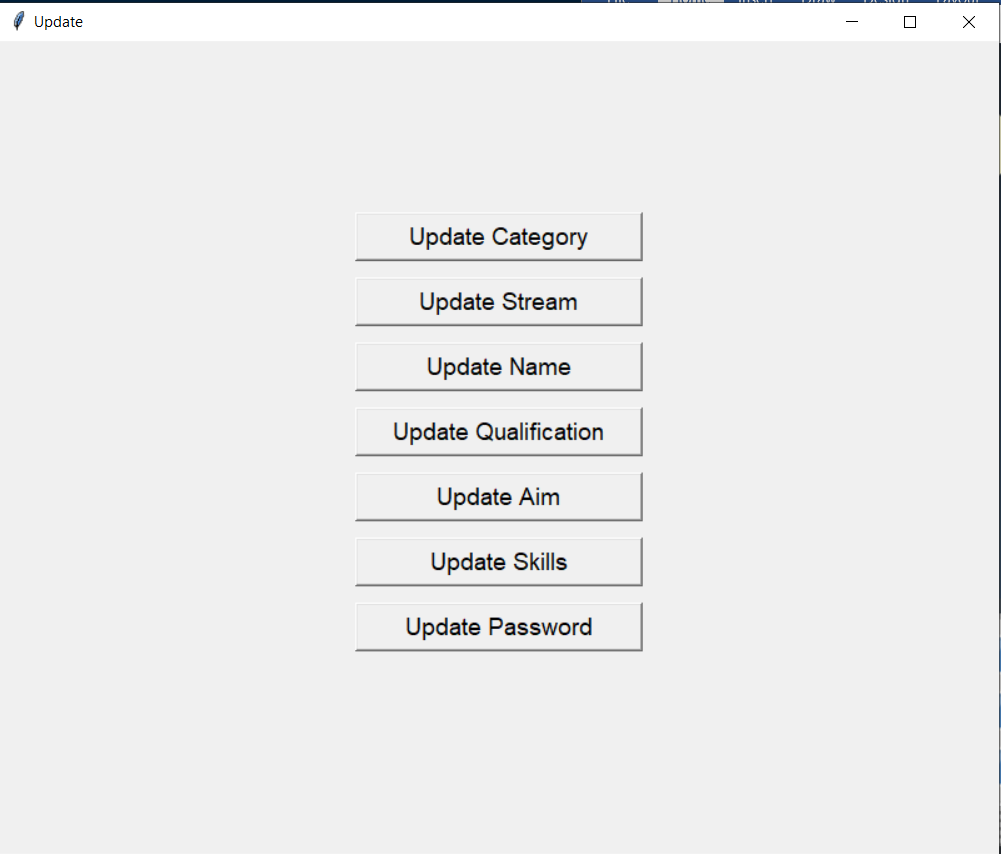
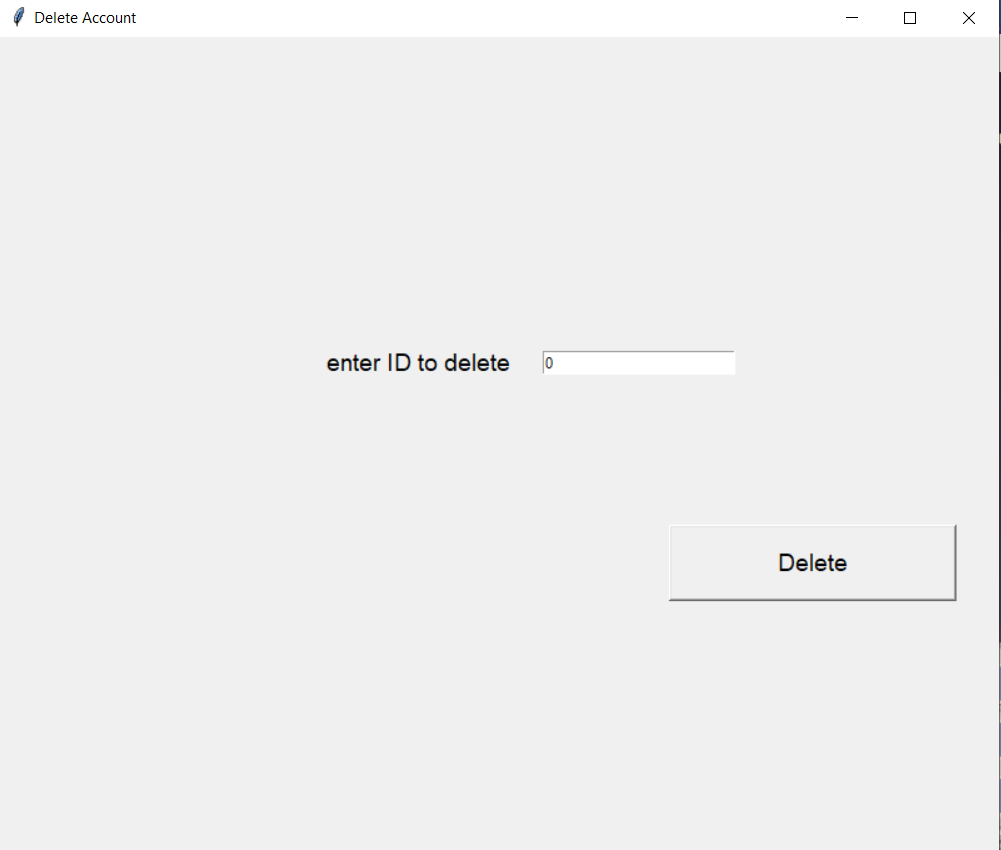
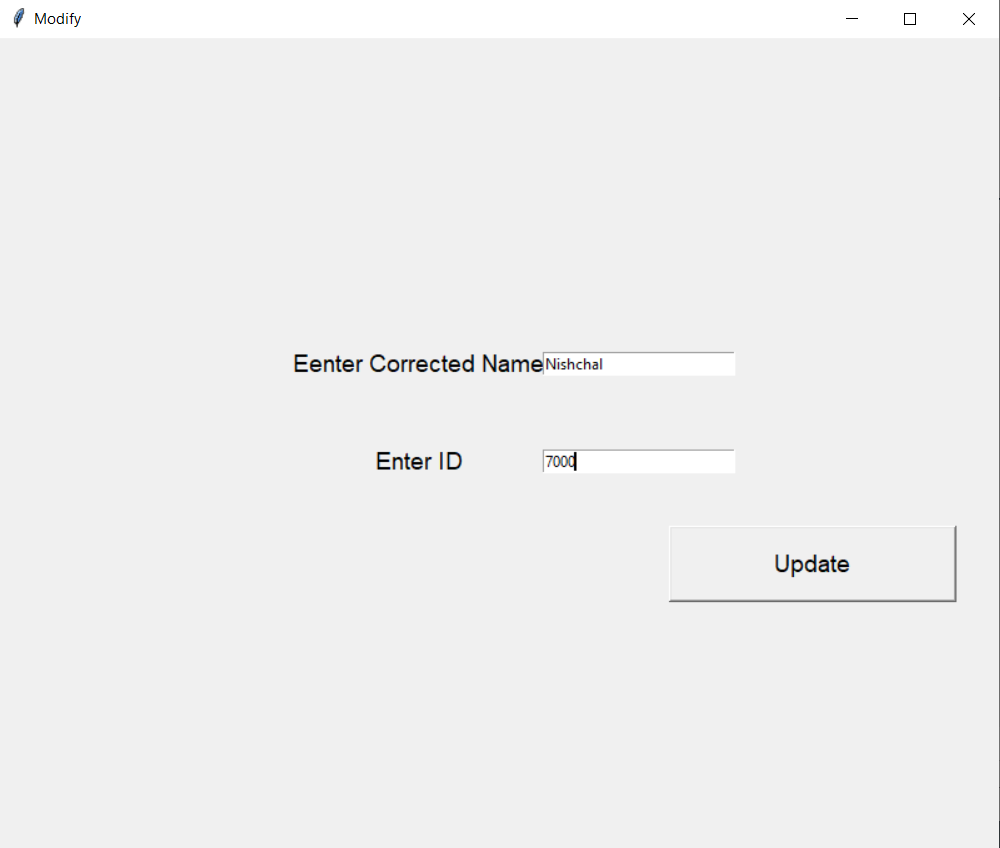


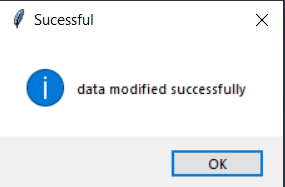


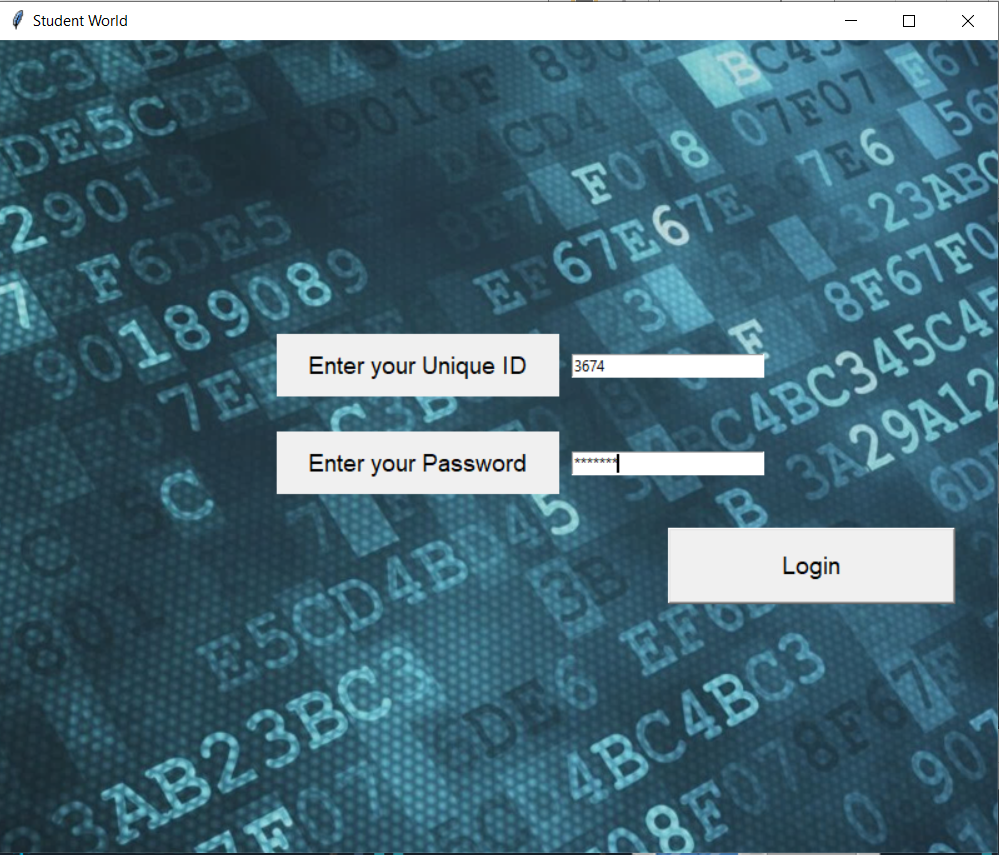


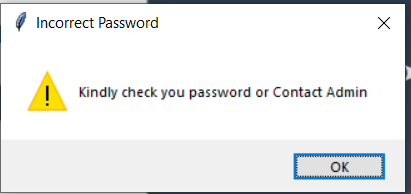


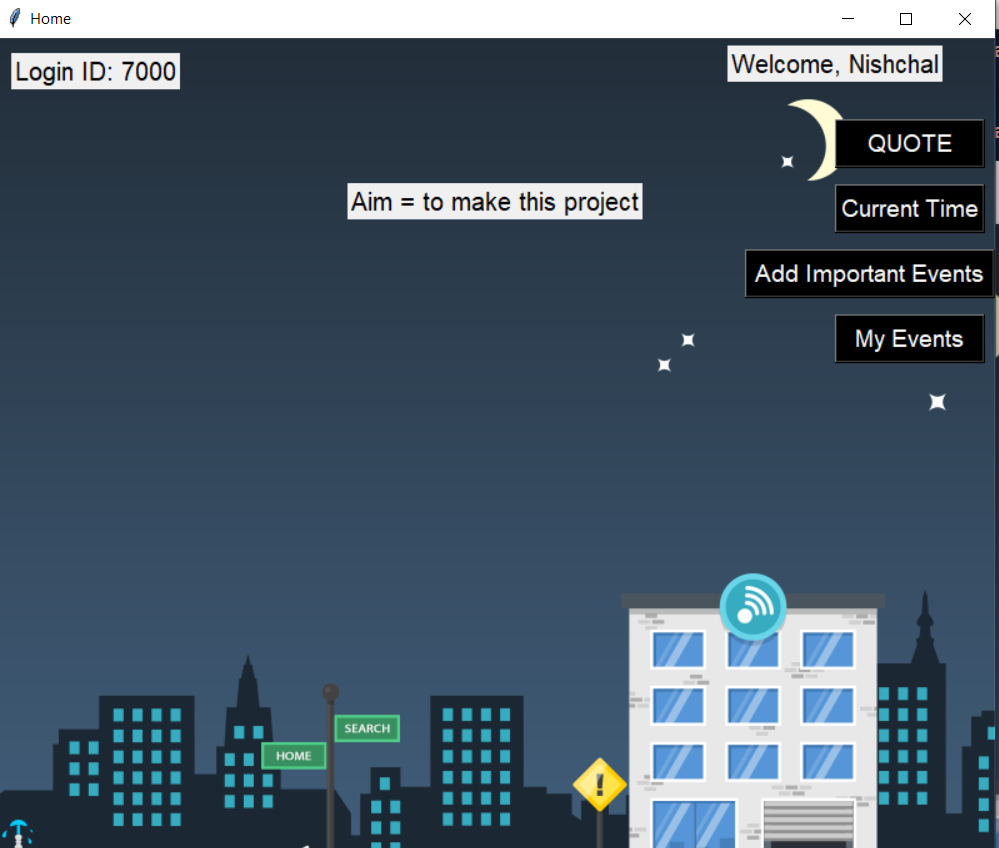


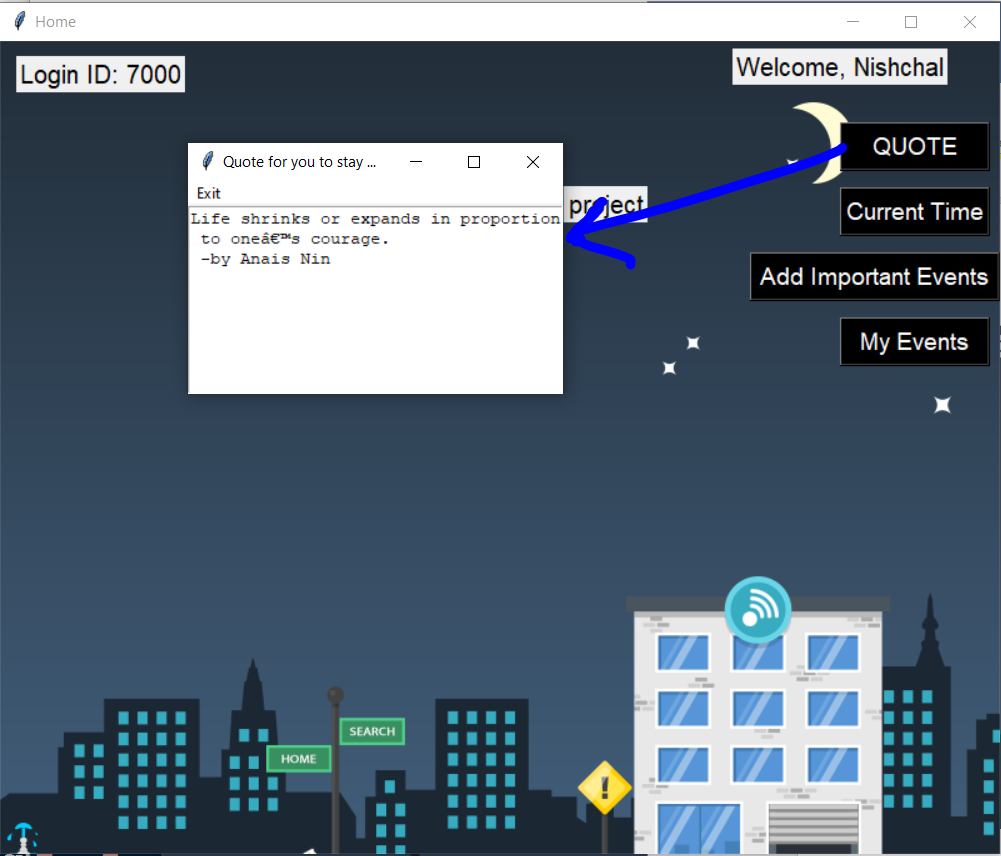
 

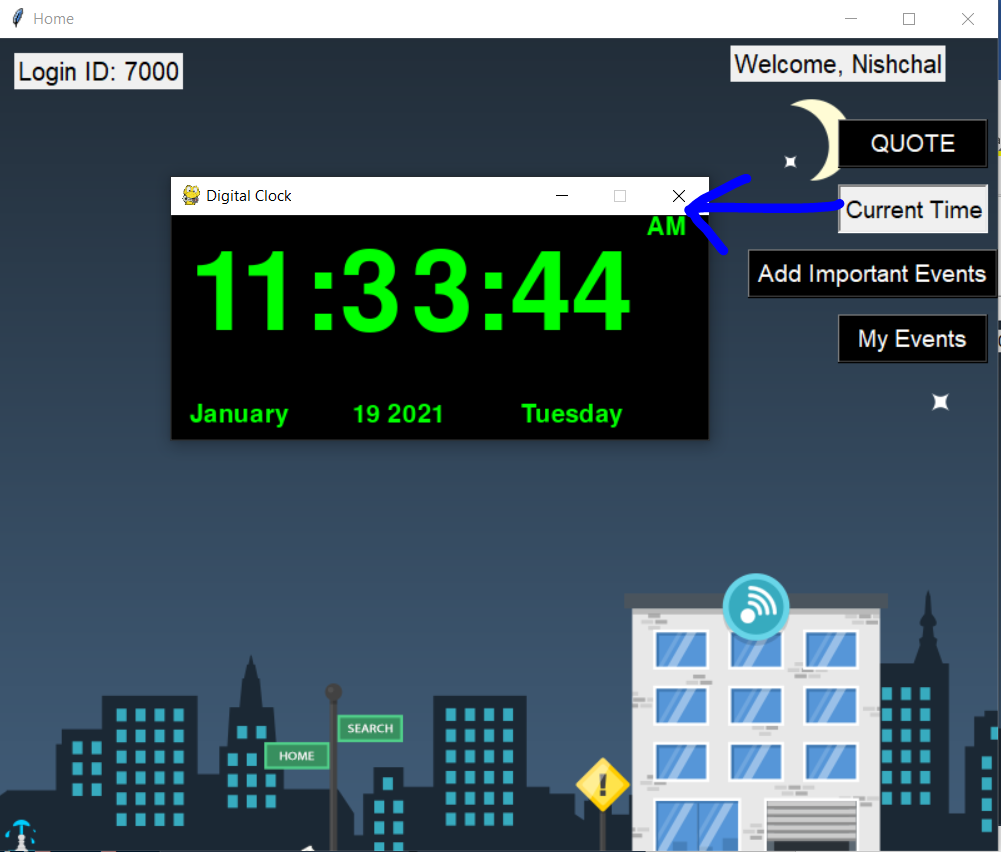


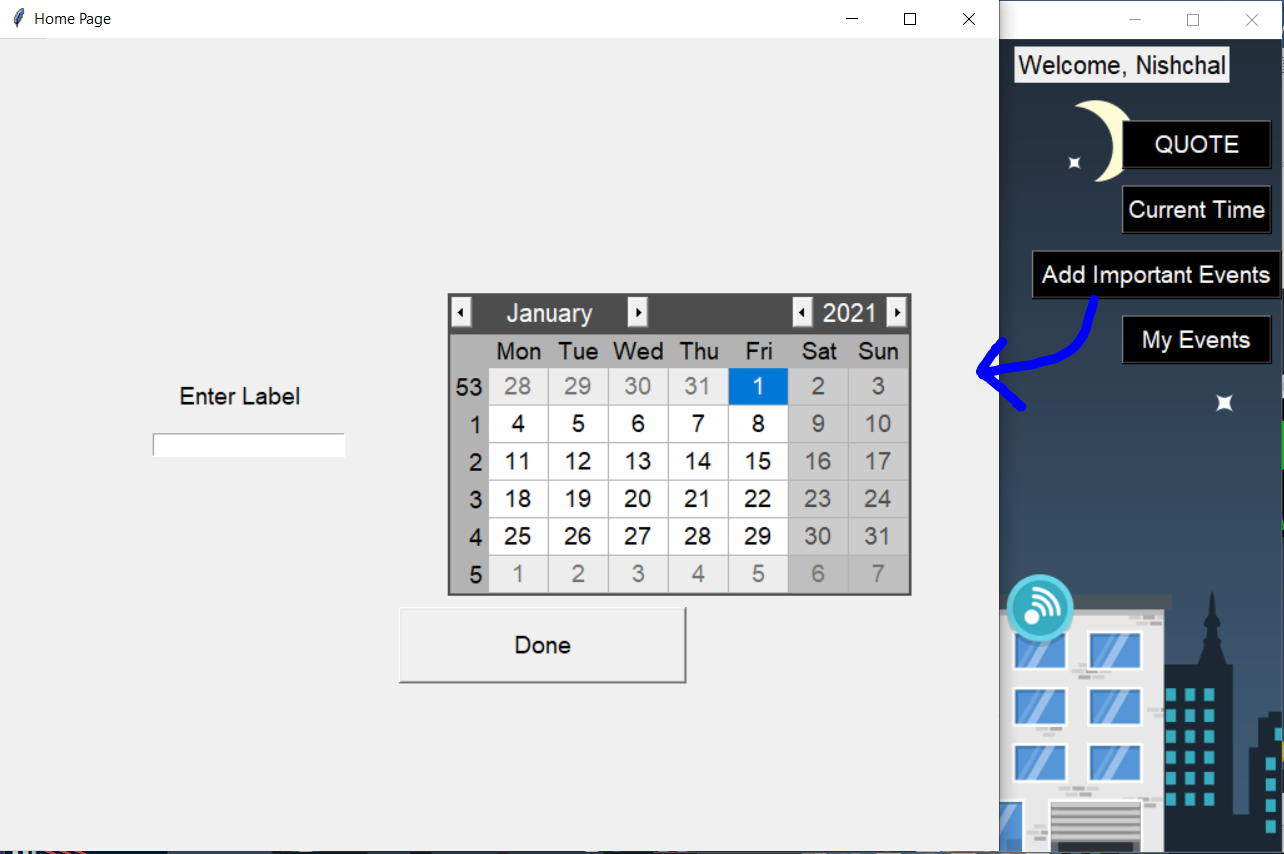


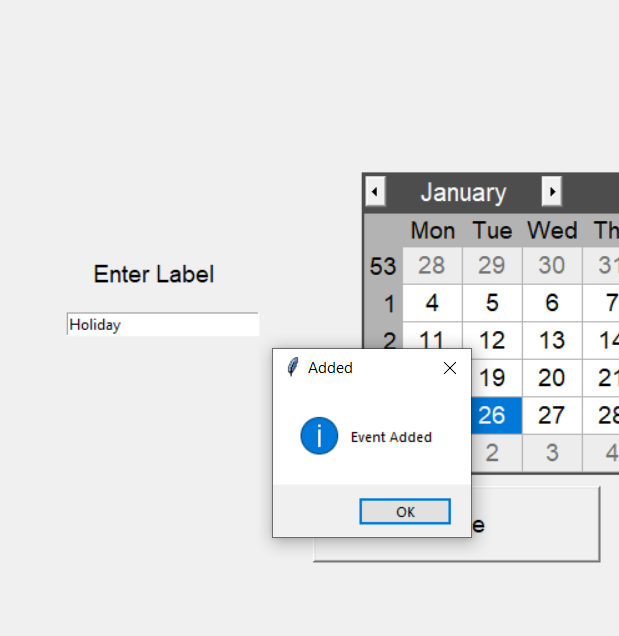


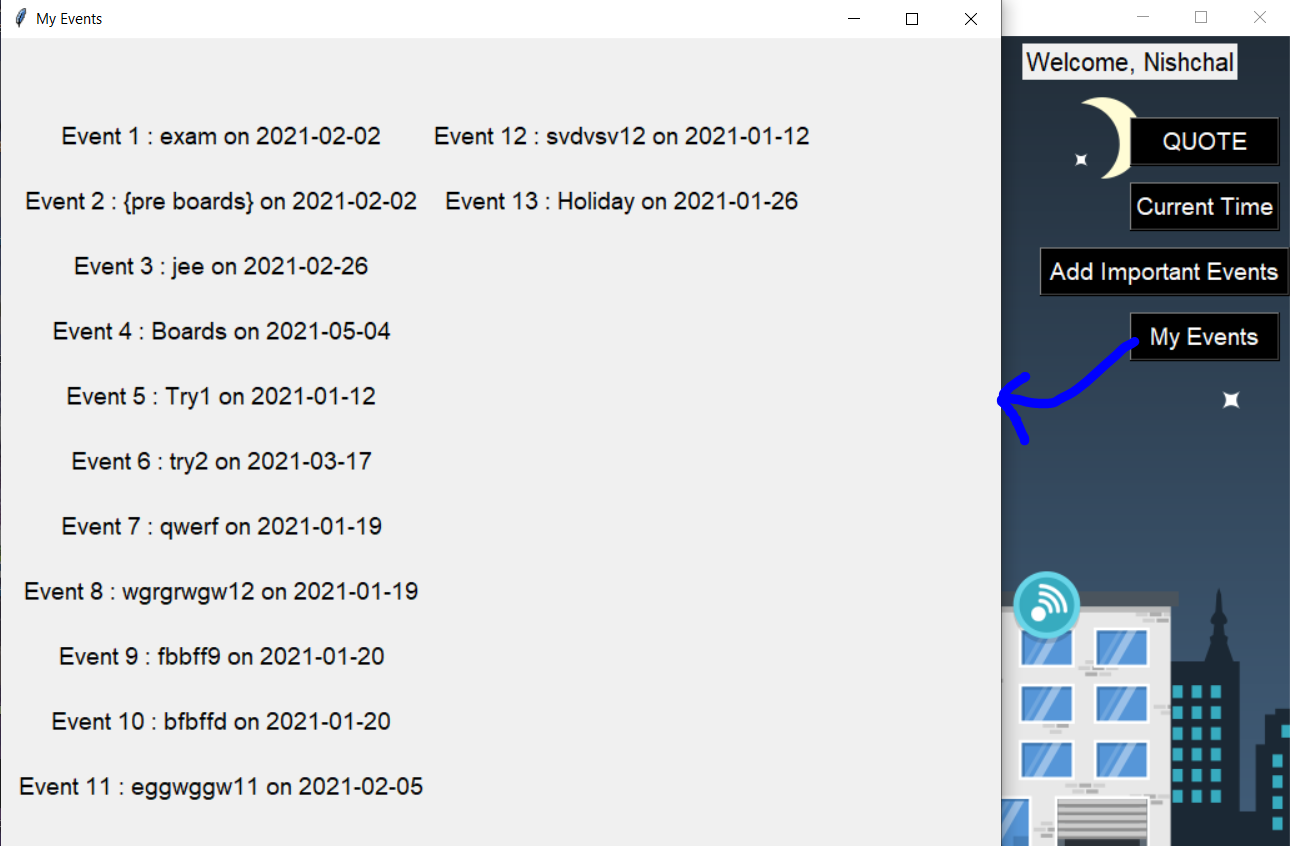












**LIMITATION**

1. SQL Injection

As we are using SQL as a database and without using any cyber security measures (two step authentication). It is likely to be hacked using SQL Injection’s

1. Duplicate Code Rule

Duplicate code is a computer programming term for a sequence of source code that occurs more than once, sequences of duplicate code are sometimes known as code clones or just clones. While this project had a complex code and required same set of functions for multiple sections in the project it was tough for us to follow duplicate code rule.

1. Database Type –

We are using MySql as a database which requires storage in the local system space and it is not on cloud.

**FUTURE IMPLICATIONS**

1. Using a cloud database instead of MySql.
2. Using two step Authentication making the application more secure.
3. Using Pyinstaller to enable the download and usage of the application on any system without requiring any python IDE.
4. Using discord webhook it get instant updates about user’s progress.
5. Using Selenium and webdriver’s to automate online or web-based tasks relevant to the concept of a goal manager.