NEHRU SMARAKA VIDYALAYA
BANGALORE

INVESTIGATORY PROJECT IN COMPUTER

Topic:

SCHOOL MANAGEMENT SYSTEM

Submitted to

Ms. Prarthana

Submitted by

Anshurup Gupta

Grade XI-D

NEHRU SMARAKA VIDYALAYA
BANGALORE

CERTIFICATE

This is to certify that **Anshurup Gupta**, bona fide student of **class XI-D** has successfully completed the project titled **“School Management System”** prescribed by the Central Board of Secondary Education for the AISSCEfor the year **2021-22**.

Teacher In-charge **Head of Department** **Principal**

Date:

ACKNOWLEDGEMENT

I wish to express my gratitude to my teacher **Ms. Prarthana** who have been instrumental in helping me complete this project.

I also wish to express my sincere thanks to our beloved principal **Dr. Anil Kumar T** and the management of Nehru Smaraka Vidyalaya for having changed our focus from exam-based learning to knowledge-learning, which I now understand will go long way in molding my future ahead for better prospects.

I wish to thank my parents, friends and all those who have directly or indirectly contributed towards the completion of this project successfully and effectively.

INDEX

ABOUT PYTHON

Python is a popular general-purpose programming language that can be used for a wide variety of applications. It includes high-level data structures, dynamic typing, dynamic binding, and many more features that make it as useful for complex application development as it is for scripting or "glue code" that connects components together. It can also be extended to make system calls to almost all operating systems and to run code written in C or C++. Due to its ubiquity and ability to run on nearly every system architecture, Python is a universal language found in a variety of different applications.

First developed in the late 1980s by Guido van Rossum, Python has advanced as an open-source programming language by managing public discussion through Python Enhancement Proposals (PEPs). In 2018, van Rossum stepped down as the language's Benevolent Dictator For Life (BDFL), and, as officially outlined in PEP 13, a steering council was put in place to serve as the leadership of the language.

The Python Software Foundation (PSF) is a 501(c)(3) non-profit corporation that holds the intellectual property rights behind the Python programming language. This includes Python version 2.1 and later, PyPI, the CPython reference implementation, and infrastructure to maintain the language. The PSF also provides grants for software craftship and runs multiple PyCon conferences a year.

Python is currently on its third major version and is regularly updated.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

OVERVIEW OF THE PROJECT

In this project, we are making a School Management System using PYTHON, covering the scope of all functions performed and managed by an Educational Institution management, also introducing some new features. The basic idea is to make and maintain a database of all the data in an educational institution to maximum accuracy, efficiency and precision.

Some key highlights of the project are as follows:

- ❖ Student data collection and management
- ❖ Attendance report
- ❖ Course Management system
- ❖ User dashboard feature with checklists for priority and organizing tasks
- ❖ Tuition fee and other educational and non-educational fee management and alerting
- ❖ Score and grade Management
- ❖ Notification of previously done, ongoing and upcoming events at the institution
- ❖ Time Table and planner management system
- ❖ Three categories are taken into account:
    1. The first category is of the students who can merely access the data but do not have the authorization to edit or delete it.
    2. The second category is of the teachers who can edit or delete data but cannot create it.
    3. The third category is of the admins who can create, edit and delete data.
    4. A suitable login feature is added to determine which category the user belongs to.
- ❖ A very promising feature of this project is its flexibility. In case of any emergency, the timetable can be changed to ensure minimum chaos and optimal usage of student time.

PYTHON LIBRARIES USED

The main theme of our project is around data management and proper storage along with user interface.

Students can easily use our dashboard system to check their schedules, attendance, reports, school fees, events, etc.

It will be an online updated system; the users must have a stable network for proper functioning of the database servers.

We will be using python libraires for accessing inbuilt modules and functions for fast development of our project. Some of the main libraries which we will be using are:

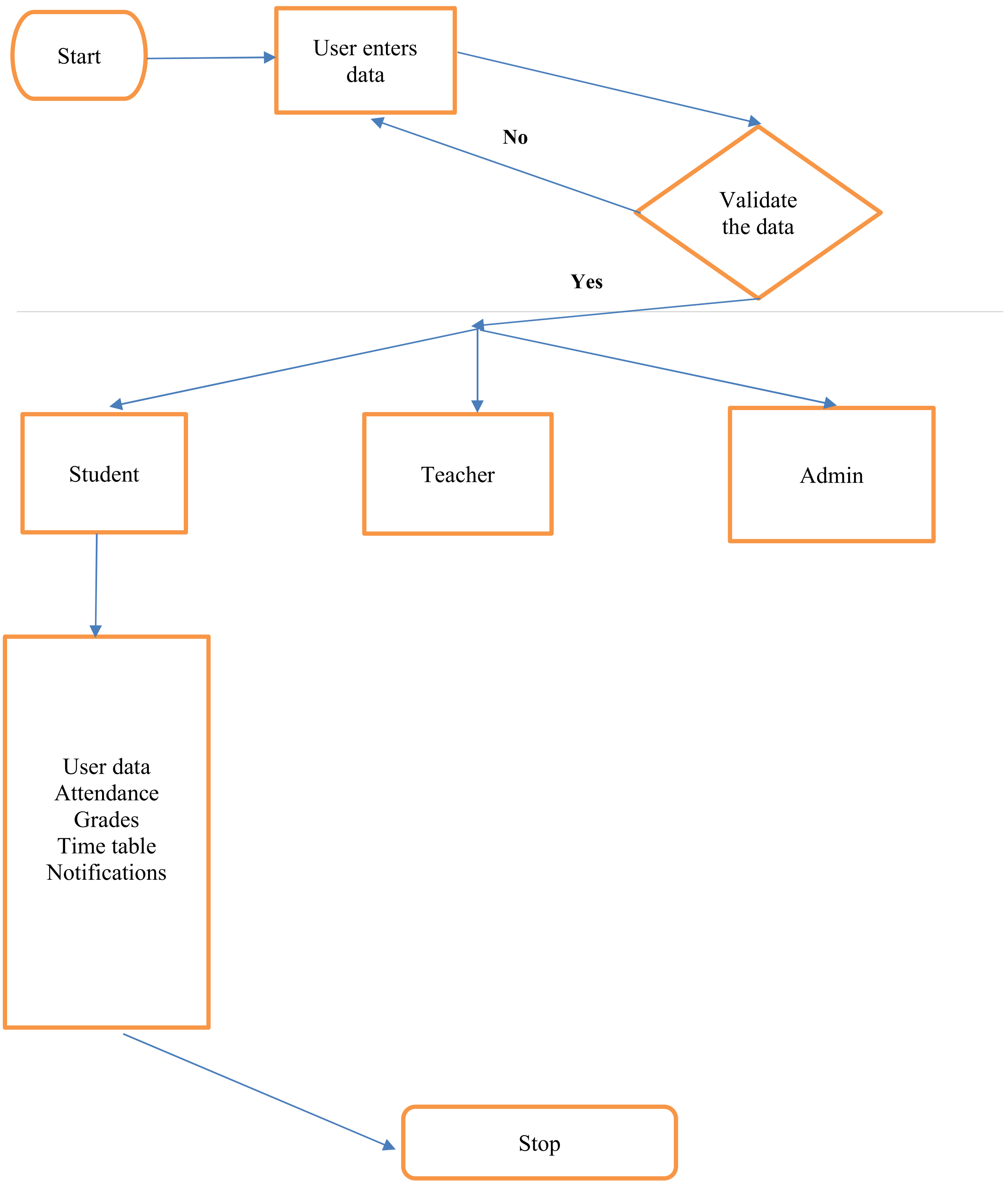
- **Tkinter:** It is a standard python library used for designing user interface. It is the fastest and simplest in terms of design and uses only python. We will be using tkinter for designing our user forms, reports and dashboard.
- **Pillow:** The Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing too

We will be using code editors, VS code, python and online IDEs for our project. Our project design will be divided into independent modules which will allow easy debugging and updating.

We will be creating a python extension file in script mode which could be run by a python interpreter.

Additionally, we will also create a fully executable windows application converted from the python file which could be run by any windows user without any python interpreter.

APPLICATION FLOWCHART



SOURCE CODE

Main.py

```
import tkinter as tk
from PIL import Image,ImageTk
import student_user_interface as sui
#import teacher_user_interface as tui
#import admin_user_interface as aui

#database coding work here
dataname="Demo"
datapassword="Demo"
datauserId="Student"

#global variable
entered_data=[]

store_username="empty"
store_password="empty"
store_val="empty"

#valid check functions
def valid_credentials(user_login):
    #print("checking the wrapped data",user_login)
    My_username,My_userpassword,My_userId=user_login

    if(My_username==dataname and My_userpassword==datapassword and My_userId==datauserId):
        l = list(user_login)
        l=l.copy()

        global entered_data
        entered_data=tuple(l)
        return True
    else:
        return False

#First login frame
my_login_window=tk.Tk()
my_login_window.title("Login Window")


#getting the width and height of screen
scr_width= my_login_window.winfo_screenwidth()
scr_height= my_login_window.winfo_screenheight()
my_login_window.geometry("%dx%d" % (scr_width,scr_height))

#maximum and minimum size of screen window
my_login_window.maxsize(scr_width,scr_height)
my_login_window.minsize(int(scr_width/2),int(scr_height/2))
```

```
#background login image
my_image1= Image.open("login_bg.png")
Myimage_w,Myimage_h=my_image1.size
bg_ratio=Myimage_w/Myimage_h

bg_image= my_image1.resize((int(3*bg_ratio*scr_height/4),int(3*scr_height/4)), Image.ANTIALIAS)
scr_bg= ImageTk.PhotoImage(bg_image)

#image canvas
login_bg=tk.Canvas(my_login_window,bg="#88cffa",width=int(3/4*scr_width))
login_bg.pack(fill="both",expand=True)
login_bg.create_image(int(scr_width/2),int(scr_height/2.05),image=scr_bg,anchor="center")

#text
login_bg.create_text(int(scr_width/2),int(scr_height/4),font=("Helvetica",int(scr_width/40)),text="Login Page")
login_bg.create_text(int(scr_width/2),int(scr_height/6),font=("Helvetica",int(scr_width/35),"underline"),text="Bangalore International Academy")

#login credentials
username_val=tk.StringVar(value="")
password_val=tk.StringVar(value="")

username_label=tk.Label(my_login_window,font=("Arial",int(scr_width/64)),text='Name')
password_label=tk.Label(my_login_window,font=("Arial",int(scr_width/64)),text='Password')

username=tk.Entry(my_login_window,width=int(scr_width/55),font=("default",int(scr_width/65)),textvariable=username_val)
password=tk.Entry(my_login_window,width=int(scr_width/55),font=("default",int(scr_width/65)),textvariable=password_val)

display_username_label=login_bg.create_window(int(scr_width/5),int(scr_height*2/5),window=username_label,anchor="center")
display_username=login_bg.create_window(int(scr_width*4/10),int(scr_height*2/5),window=username,anchor="center")
display_password_label=login_bg.create_window(int(scr_width/5),int(scr_height*1/2),window=password_label,anchor="center")
display_password=login_bg.create_window(int(scr_width*4/10),int(scr_height*1/2),window=password,anchor="center")

#single option buttons
btn_val=tk.StringVar(value="Student")

student_box=tk.Radiobutton(my_login_window,font=("default",int(scr_width/65)),text='Student',value='Student',variable=btn_val)
Teacher_box=tk.Radiobutton(my_login_window,font=("default",int(scr_width/65)),text='Teacher',value='Teacher',variable=btn_val)
Admin_box=tk.Radiobutton(my_login_window,font=("default",int(scr_width/65)),text='Admin',value='Admin',variable=btn_val)

display_student_box=login_bg.create_window(int(scr_width*3/10),int(scr_height*4/7),window=student_box,anchor="center")
display_Teacher_box=login_bg.create_window(int(scr_width*4/10),int(scr_height*4/7),window=Teacher_box,anchor
="center")
display_Admin_box=login_bg.create_window(int(scr_width*5/10),int(scr_height*4/7),window=Admin_box,anchor="
center")

#enter button
def submit():
    global store_username,store_password,store_val
    store_username=username_val.get()
    store_password=password_val.get()
    store_val=btn_val.get()
    t=(store_username,store_password,store_val)

    if(valid_credentials(t)==True):
        my_login_window.destroy()
    else:
        my_Text=login_bg.create_text(int(scr_width/2),int(scr_height/3),text="Incorrect
userdata",font=("default",int(scr_width/67),"underline"),fill="red")
        my_login_window.after(3000,login_bg.delete, my_Text)

login_btn=tk.Button(my_login_window,font=("Helvetica",int(scr_width/65),"underline"),text='LOGIN',bd='5',comma
nd=lambda: [submit()])
display_login_btn=login_bg.create_window(int(scr_width*1/2),int(scr_height*5/7),window=login_btn,anchor="cente
r")

my_login_window.mainloop()

#print("final value",entered_data)

if(entered_data!=[]):
    if(entered_data[2]=="Student"):
        sui.student_data(entered_data[0])
        sui.Student_UI()
    elif(entered_data[2]=="Teacher"):
        pass
    elif(entered_data[2]=="Admin"):
        pass
else:
    print("ERROR")
```

student_user_interface.py

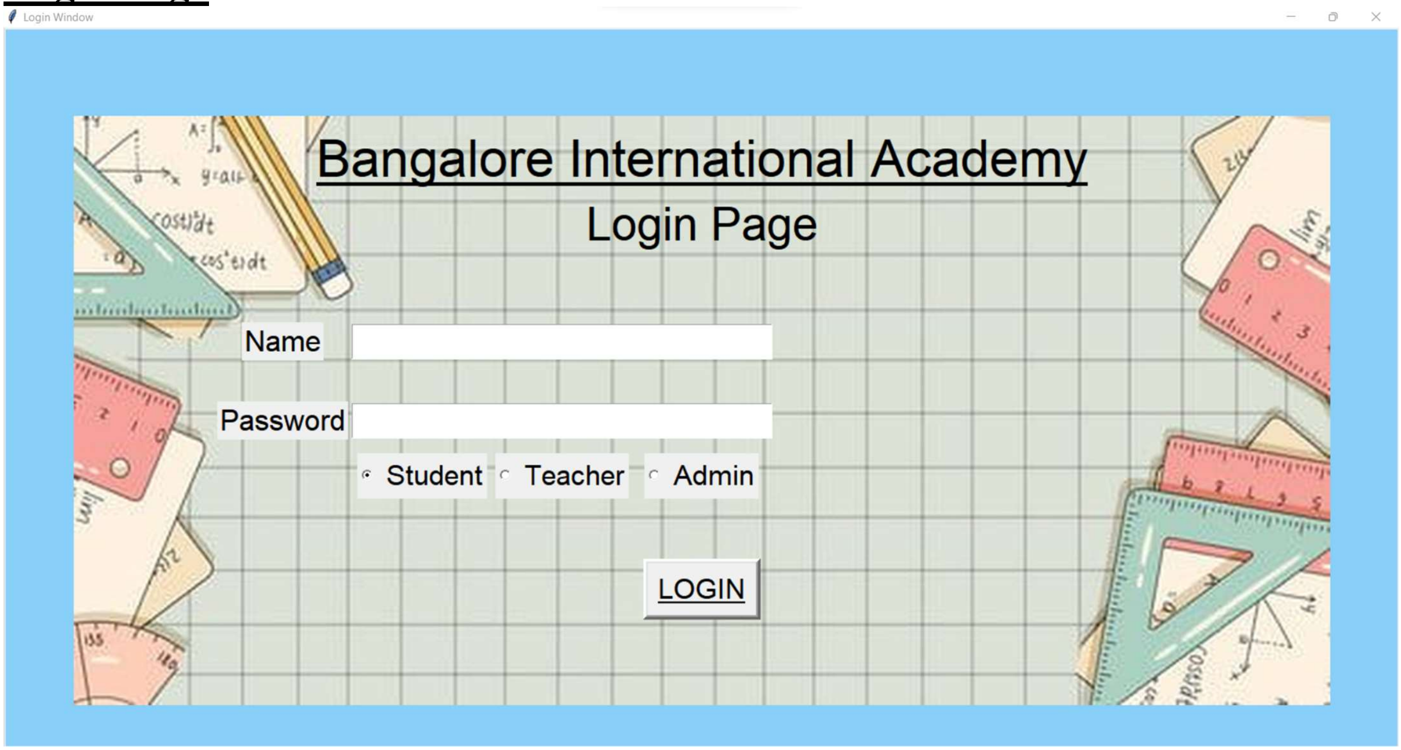
```
import tkinter as tk
from PIL import Image,ImageTk

#global variables
Student_class,Student_sec,Student_name= "empty","empty","empty"
Student_gender,Student_roll="empty","empty"

current_working_days="empty"
attended_working_days="empty"

Student_marks={"English":0,"Physics":0,"Mathematics":0,"Chemistry":0,"Computer":0}

Student_timetable={"Time":["empty"],"Monday":["empty"],"Tuesday":["empty"],"Wednesday":["empty"],"Thursday":["empty"],"Friday":["empty"],"Saturday":["Holiday"],"Sunday":["Holiday"]}

Class_notifications=["empty"]
Student_notifications=["empty"]

events=[]

def student_data(student_name):

    #Database code working here
    if(student_name=="Demo"):
        global Student_name,Class_notifications,Student_notifications,Student_marks
        global Student_class,Student_sec,Student_gender,Student_roll,Student_timetable,events

        Student_name=student_name
        Student_class=11
        Student_sec="D"
        Student_gender="Male"
        Student_roll=1
        Student_timetable={
                "Time":["9:00-9:50","10:00-10:50","11:00-11:30","11:40-12:30","12:40-1:30","1:40-2:30"],
                "Monday":["Eng","Phy","Break","Math","Comp(T)","Chem"],
                "Tuesday":["Math","Phy","Break","Chem","Eng","Comp(P)"],
                "Wednesday":["Chem","Math","Break","Eng","Comp(T)","Phy"],
                "Thursday":["Math","Chem","Break","Eng","Phy","Comp(P)"],
                "Friday":["Phy","Chem","Break","Maths","Comp(P)","Eng"],
                "Saturday":["Holiday"],
                "Sunday":["Holiday"]}
        Student_marks={"English":80,"Physics":80,"Mathematics":100,"Chemistry":90,"Computer":100}
        Student_notifications=["Term 1 Fees paid","Term 2 Fees paid"]
        Class_notifications=["English class changed","Computer Test assigned","Submit all assignments on ptm"]
        events=["Computer Project on 21st"]

def refresh_window():
    Student_UI()

#student user interface function
```

```
def Student_UI():

    #Student window
    Student_window=tk.Tk()
    Student_window.title("Student Dashboard")

    #getting the width and height of screen
    scr_width= Student_window.winfo_screenwidth()
    scr_height= Student_window.winfo_screenheight()
    Student_window.geometry("%dx%d" % (scr_width,scr_height))

    #maximum and minimum size of screen window
    Student_window.maxsize(int(scr_width),int(scr_height))
    Student_window.minsize(int(scr_width/2),int(scr_height/2))

    #Partial frame for canvas and scrollbar
    Student_frame=tk.Frame(Student_window,width=scr_width/2,height=scr_height/2)
    Student_frame.pack(fill="both",expand="true")

    #option menu
    menu_frame=tk.Frame(Student_frame)
    menu_frame.pack(side="top",fill="x")

    m=tk.Menu(menu_frame)
    Student_window.config(menu=m)

    def close_window():
        Student_window.destroy()

    submenu=tk.Menu(m)
    m.add_cascade(label='Options',menu=submenu)
    submenu.add_command(label='Refresh',command=lambda:[close_window(),refresh_window()])
    submenu.add_command(label='Exit', command=close_window)
    submenu.add_separator()

    #main canvas
    Student_canvas=tk.Canvas(Student_frame,bg='#C2E5D3',width=scr_width/3,height=scr_height/3,scrollregion=(0,0
,scr_width,scr_height))

    #vertical scrollbar
    vbar=tk.Scrollbar(Student_frame,orient="vertical")
    vbar.pack(side="right",fill="y")
    vbar.config(command=Student_canvas.yview)

    Student_canvas.config(width=scr_width/3,height=scr_height/3)
    Student_canvas.config(yscrollcommand=vbar.set)
    Student_canvas.pack(side="left",expand=True,fill="both")

    #student images
    my_image1= Image.open("boy_bg.png")
    Myimage1_w,Myimage1_h=my_image1.size
    boyi_ratio=Myimage1_w/Myimage1_h
    boy_image= my_image1.resize((int(boyi_ratio*scr_height*3/10),int(scr_height*3/10)), Image.ANTIALIAS)
    boy_bg= ImageTk.PhotoImage(boy_image)

    my_image2= Image.open("girl_bg.png")
    Myimage2_w,Myimage2_h=my_image2.size
    girli_ratio=Myimage2_w/Myimage2_h

    girl_image= my_image2.resize((int(girli_ratio*scr_height*3/10),int(scr_height*3/10)), Image.ANTIALIAS)
    girl_bg= ImageTk.PhotoImage(girl_image)

    #background image
    my_image3=Image.open("type1_bg.png")
    Myimage3_w,Myimage3_h=my_image3.size
    sui_ratio=Myimage3_w/Myimage3_h

    sui_image=my_image3.resize((int(sui_ratio*scr_height*14/15),int(scr_height*14/15)), Image.ANTIALIAS)
    sui_bg= ImageTk.PhotoImage(sui_image)

    Student_canvas.create_image(int(scr_width/2),int(scr_height/2),image=sui_bg,anchor="center")

    #nameholder display
    my_image4=Image.open("nameholder_bg.png")
    Myimage4_w,Myimage4_h=my_image4.size
    holder_ratio=Myimage4_w/Myimage4_h*1.25

    holder_image=my_image4.resize((int(holder_ratio*scr_height*1/2.25),int(scr_height*1/2.25)),
Image.ANTIALIAS)
    holder_bg= ImageTk.PhotoImage(holder_image)

    Student_canvas.create_image(int(scr_width/2),int(scr_height/3.5),image=holder_bg,anchor="center")

    #Basic info display
    Student_canvas.create_text(int(scr_width/2),int(scr_height/4),font=("Helvetica",int(scr_width/50)),text="Welcome
"+Student_name+",")
    Student_canvas.create_text(int(scr_width/1.95),int(scr_height/3),font=("Helvetica",int(scr_width/60)),text="Class:
"+str(Student_class)+" Section: "+Student_sec+" Roll no: "+str(Student_roll))

    if(Student_gender=="Male"):
        Student_canvas.create_image(int(scr_width/3.15),int(scr_height/3.25),image=boy_bg,anchor="center")
    elif(Student_gender=="Female"):
        Student_canvas.create_image(int(scr_width/3.15),int(scr_height/3.25),image=girl_bg,anchor="center")
    else:
        Student_canvas.create_text(int(scr_width/3.15),int(scr_height/3.25),font=("Helvetica",int(scr_width/75)),text="n
o data",anchor="center")

    def data_window():
        userdata_window=tk.Toplevel(Student_window)
        userdata_window.title("User data")
        userdata_window.geometry("%dx%d" % (scr_width*4/5,scr_height/3))

        #maximum and minimum size of screen window
        userdata_window.maxsize(int(scr_width*4/5),int(scr_height/3))
        userdata_window.minsize(int(scr_width/2),int(scr_height/3))

        #other label
        l=tk.Label(userdata_window,text="Data",font=("Helvetica",int(scr_width/90)))
        l.pack()

        #scrollbar
        s=tk.Scrollbar(userdata_window)
        s.pack(side="right",fill="y")

        #text
        t=tk.Text(userdata_window,wrap="word",yscrollcommand=s.set,font=("Helvetica",int(scr_width/85)),bg="#FFF
DD0")
        t.pack(side="left",fill="both",expand="true")

        #display
        d={"name":Student_name,"class":Student_class,"section":Student_sec,"roll
no":Student_roll,"gender":Student_gender,"marks":Student_marks}

        #display loop
        for i in d:
            t.insert('end',i+": "+str(d[i])+"\n")

        s.config(command=t.yview)

        #buttons
        b=tk.Button(userdata_window,text="EXIT",font=int(scr_width/50),command=lambda:[userdata_window.destroy
()])
        b.pack(side="bottom")
        userdata_window.mainloop()

    Student_user_btn=tk.Button(Student_frame,font=("Helvetica",int(scr_width/85),"underline"),text='User
data',bd='3',command=lambda:[data_window()])
    Student_canvas.create_window(int(scr_width*2/3),int(scr_height/2.45),window=Student_user_btn,anchor="center"
)

    #widget image
    timetable_image= tk.PhotoImage(file='timetable_bg.png')
    attendence_image= tk.PhotoImage(file='attendence_bg.png')
    grades_image= tk.PhotoImage(file='grades_bg.png')
    events_image= tk.PhotoImage(file='events_bg.png')

    timetable_bg= timetable_image.subsample(4,4)
    attendence_bg= attendence_image.subsample(4,4)
    grades_bg= grades_image.subsample(4,4)
    events_bg= events_image.subsample(3,3)

    #widget functions
    def show_mytimetable():
        userdata_window=tk.Toplevel(Student_window)
        userdata_window.title("Schedule")
        userdata_window.geometry("%dx%d" % (scr_width*3/4,scr_height/2))

        #maximum and minimum size of screen window
        userdata_window.resizable(0,0)

        #other label
        l=tk.Label(userdata_window,text="Time table",font=("Helvetica",int(scr_width/90)))
        l.pack()
        s=tk.Scrollbar(userdata_window)
        s.pack(side="right",fill="y")
        t=tk.Text(userdata_window,wrap="word",yscrollcommand=s.set,font=("Helvetica",int(scr_width/85)),bg="#FFF
DD0")
        t.pack(side="left",fill="both",expand="true")

        #display
        for i in (Student_timetable):
            t.insert("end",i+": ")
            for j in ((Student_timetable[i])):
                t.insert('end',j+" | ")
            t.insert('end',"\n")

        s.config(command=t.yview)
        b=tk.Button(userdata_window,text="EXIT",font=int(scr_width/50),command=lambda:[userdata_window.destroy
()])
        b.pack(side="bottom")
        userdata_window.mainloop()

    def show_myevents():
        userdata_window=tk.Toplevel(Student_window)
        userdata_window.title("User data")
        userdata_window.geometry("%dx%d" % (scr_width*3/4,scr_height/3))

        #maximum and minimum size of screen window
        userdata_window.resizable(0,0)

        #other label
        l=tk.Label(userdata_window,text="School Events",font=("Helvetica",int(scr_width/90)))
        l.pack()
        s=tk.Scrollbar(userdata_window)
        s.pack(side="right",fill="y")
        t=tk.Text(userdata_window,wrap="word",yscrollcommand=s.set,font=("Helvetica",int(scr_width/85)),bg="#FFF
DD0")
        t.pack(side="left",fill="both",expand="true")

        #display
        t.insert('end',"Events:"+"\n")
        for i in events:
            t.insert('end',str(i)+"\n")

        s.config(command=t.yview)
        b=tk.Button(userdata_window,text="EXIT",font=int(scr_width/50),command=lambda:[userdata_window.destroy
()])
```

```
        b.pack(side="bottom")
        userdata_window.mainloop()

    def show_myattendence():
        #database and pandas needed
        pass

    def show_mygrades():
        #database and pandas needed
        pass

    #option widgets
    sui_timetable_btn=tk.Button(Student_frame,font=("Helvetica",int(scr_width/60),"underline"),text='Time table',image=timetable_bg,compound="left",bd='5',command=lambda:[show_mytimetable()])
    sui_Attendence_btn=tk.Button(Student_frame,state="disabled",font=("Helvetica",int(scr_width/60),"underline"),text='Attendence',image=attendence_bg,compound="left",bd='5',command=lambda:[show_myattendence()])
    sui_grades_btn=tk.Button(Student_frame,state="disabled",font=("Helvetica",int(scr_width/60),"underline"),text='Grade and score',image=grades_bg,compound="right",bd='5',command=lambda:[show_mygrades()])
    sui_events_btn=tk.Button(Student_frame,font=("Helvetica",int(scr_width/60),"underline"),text='Events',image=events_bg,compound="right",bd='5',command=lambda:[show_myevents()])

    Student_canvas.create_window(int(scr_width*4/11),int(scr_height*7/11),window=sui_timetable_btn,anchor="center")
    Student_canvas.create_window(int(scr_width*4/11),int(scr_height*9/11),window=sui_Attendence_btn,anchor="center")
    Student_canvas.create_window(int(scr_width*7/11),int(scr_height*7/11),window=sui_grades_btn,anchor="center")
    Student_canvas.create_window(int(scr_width*7/11),int(scr_height*9/11),window=sui_events_btn,anchor="center")

    #notifications panel
    my_notificationframe=tk.Frame(Student_canvas)
    Student_canvas.create_window(int(scr_width/8),int(scr_height/2),height=int(scr_height*5/6),width=int(scr_width/5.5),window=my_notificationframe,anchor="center")

    #notifcation box
    mylabel=tk.Label(my_notificationframe,font=("Helvetica",int(scr_width/75),"underline italic"),text="Notifications")
    mytextbox=tk.Text(my_notificationframe,wrap="word",height=int(scr_height*5/6),width=int(scr_width/6),font=("Helvetica",int(scr_width/85)),bg="#D1FFEA")

    mylabel.pack()
    mytextbox.pack(expand="true",fill="both")

    #display notifications
    mytextbox.insert('end',"Class notifications"+"\n")
    for i in range(0,len(Class_notifications)):
        mytextbox.insert('end',"*"+Class_notifications[i])
        mytextbox.insert('end',"\n")
    mytextbox.insert('end','Student notifications'+'\n')
    for i in range(0,len(Student_notifications)):
        mytextbox.insert("end",Student_notifications[i])
        mytextbox.insert('end',"\n")
    Student_window.mainloop()

#Stand alone testing
'''student_data("Demo")
Student_UI()'''
```

Sample outputs

Login Page



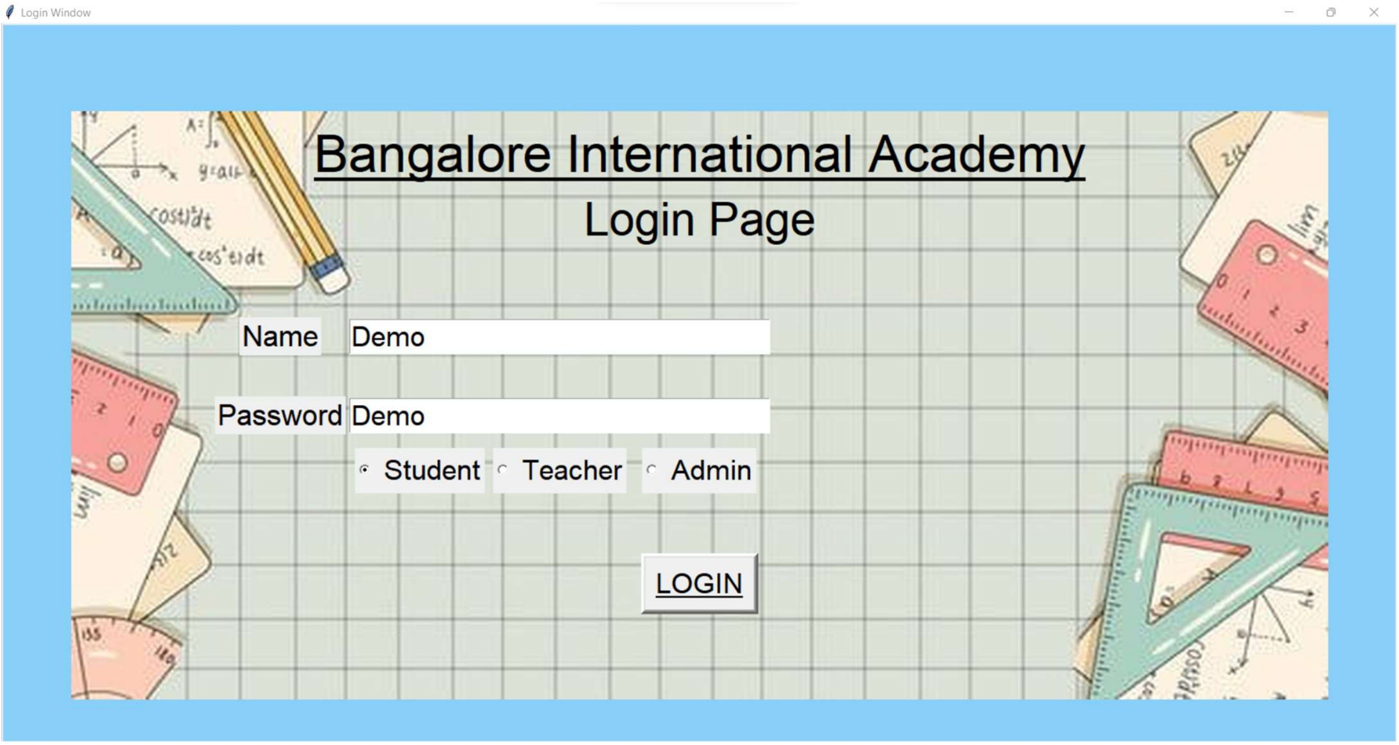
Bangalore International Academy

Login Page

Name

Password

● Student ○ Teacher ○ Admin



Bangalore International Academy

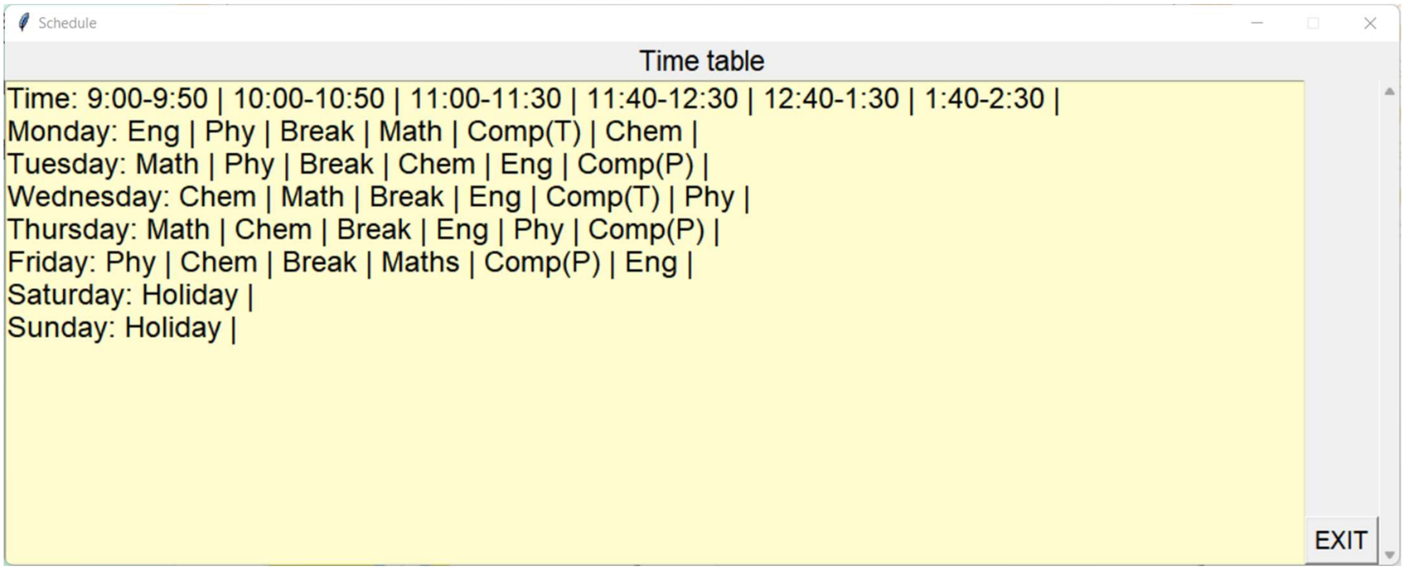
Login Page

Name Demo

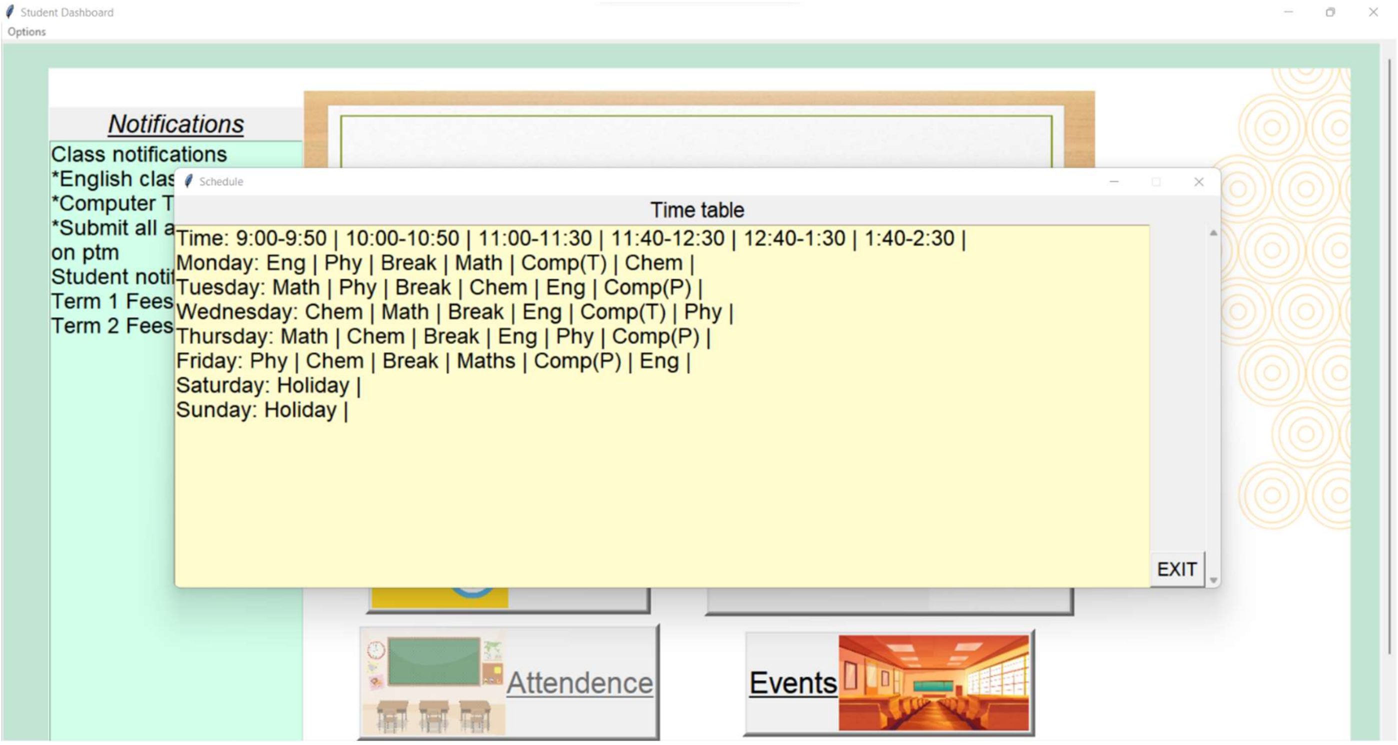
Password Demo

● Student ○ Teacher ○ Admin

Time-Table



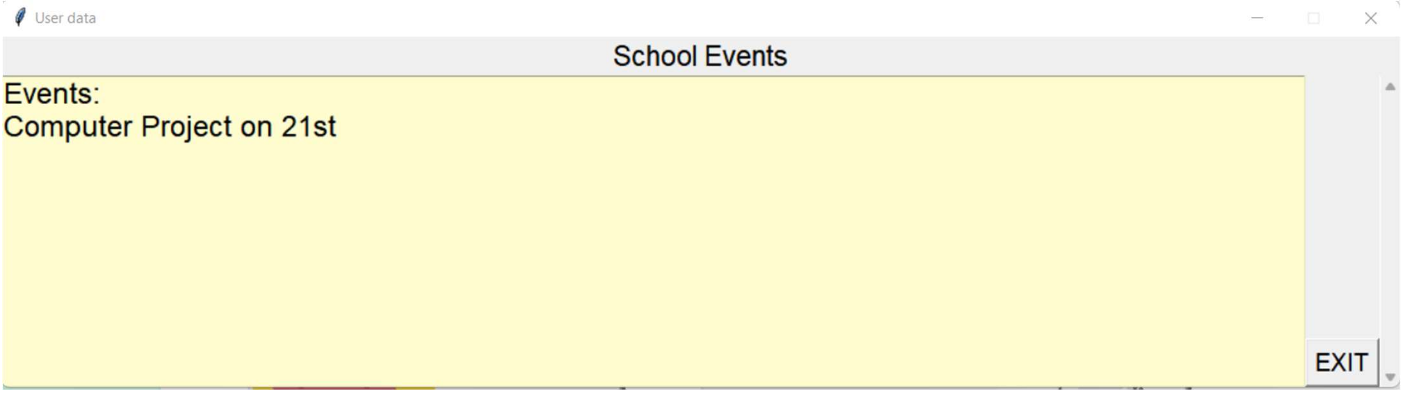
```
Time: 9:00-9:50 | 10:00-10:50 | 11:00-11:30 | 11:40-12:30 | 12:40-1:30 | 1:40-2:30 |
Monday: Eng | Phy | Break | Math | Comp(T) | Chem |
Tuesday: Math | Phy | Break | Chem | Eng | Comp(P) |
Wednesday: Chem | Math | Break | Eng | Comp(T) | Phy |
Thursday: Math | Chem | Break | Eng | Phy | Comp(P) |
Friday: Phy | Chem | Break | Maths | Comp(P) | Eng |
Saturday: Holiday |
Sunday: Holiday |
```



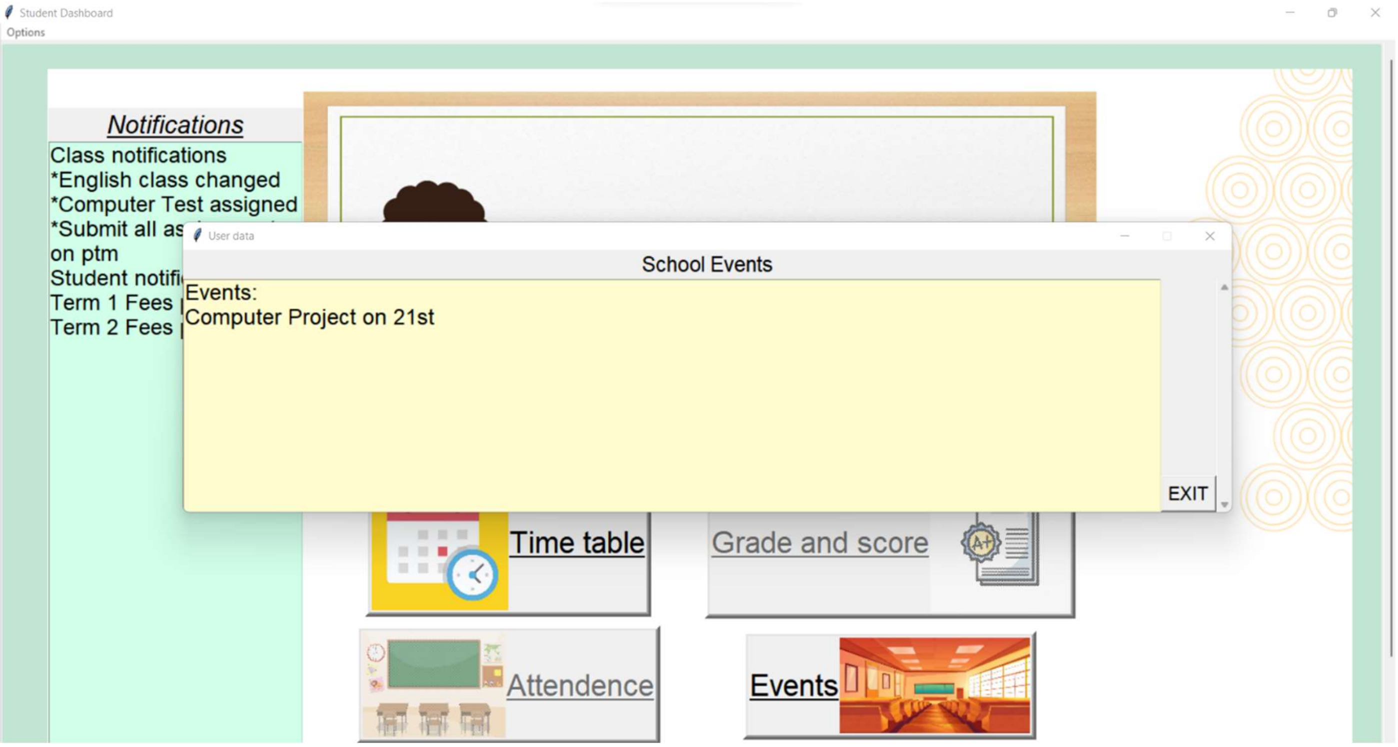
Class notifications
*English clas
*Computer T
*Submit all a
on ptm
Student noti
Term 1 Fees
Term 2 Fees

```
Time: 9:00-9:50 | 10:00-10:50 | 11:00-11:30 | 11:40-12:30 | 12:40-1:30 | 1:40-2:30 |
Monday: Eng | Phy | Break | Math | Comp(T) | Chem |
Tuesday: Math | Phy | Break | Chem | Eng | Comp(P) |
Wednesday: Chem | Math | Break | Eng | Comp(T) | Phy |
Thursday: Math | Chem | Break | Eng | Phy | Comp(P) |
Friday: Phy | Chem | Break | Maths | Comp(P) | Eng |
Saturday: Holiday |
Sunday: Holiday |
```

School Events



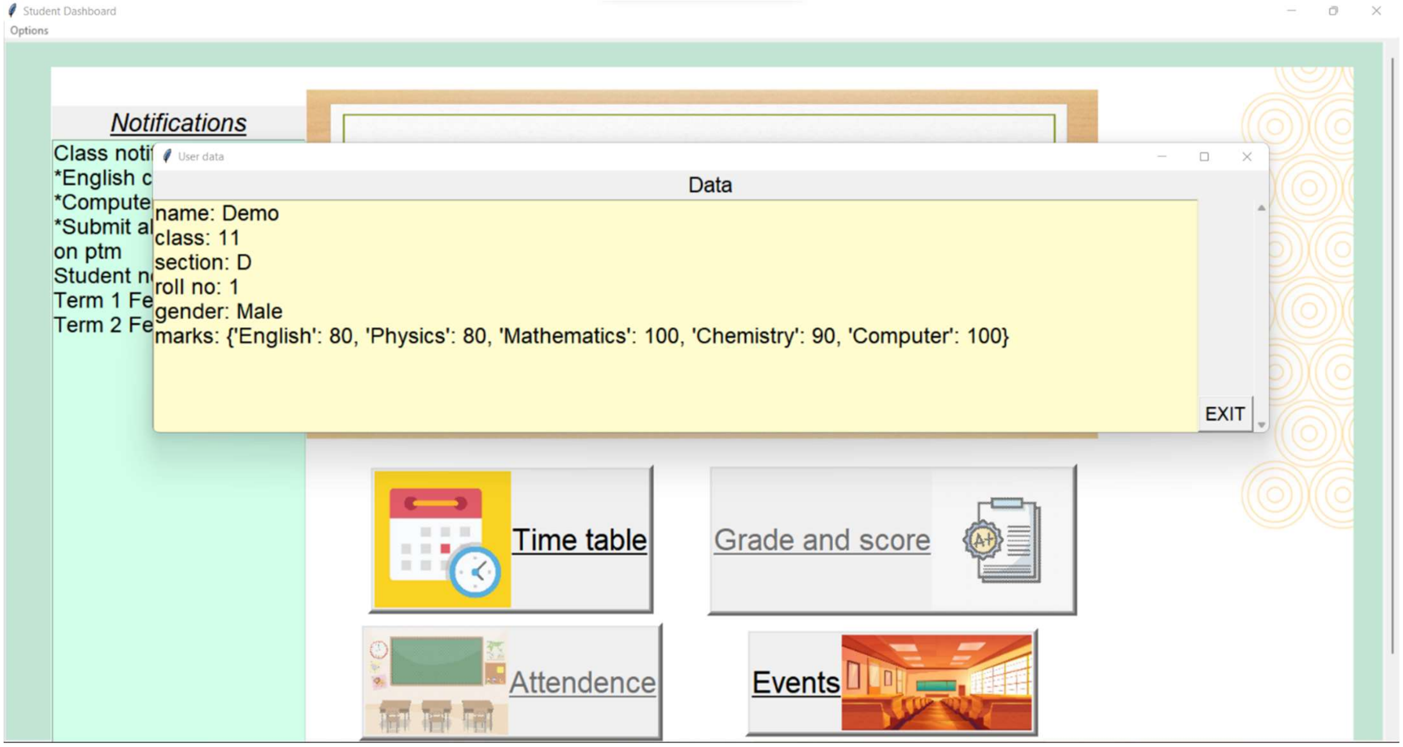
Events:
Computer Project on 21st



Class notifications
*English class changed
*Computer Test assigned
*Submit all as
on ptm
Student notifi
Term 1 Fees
Term 2 Fees

Events:
Computer Project on 21st

User Data

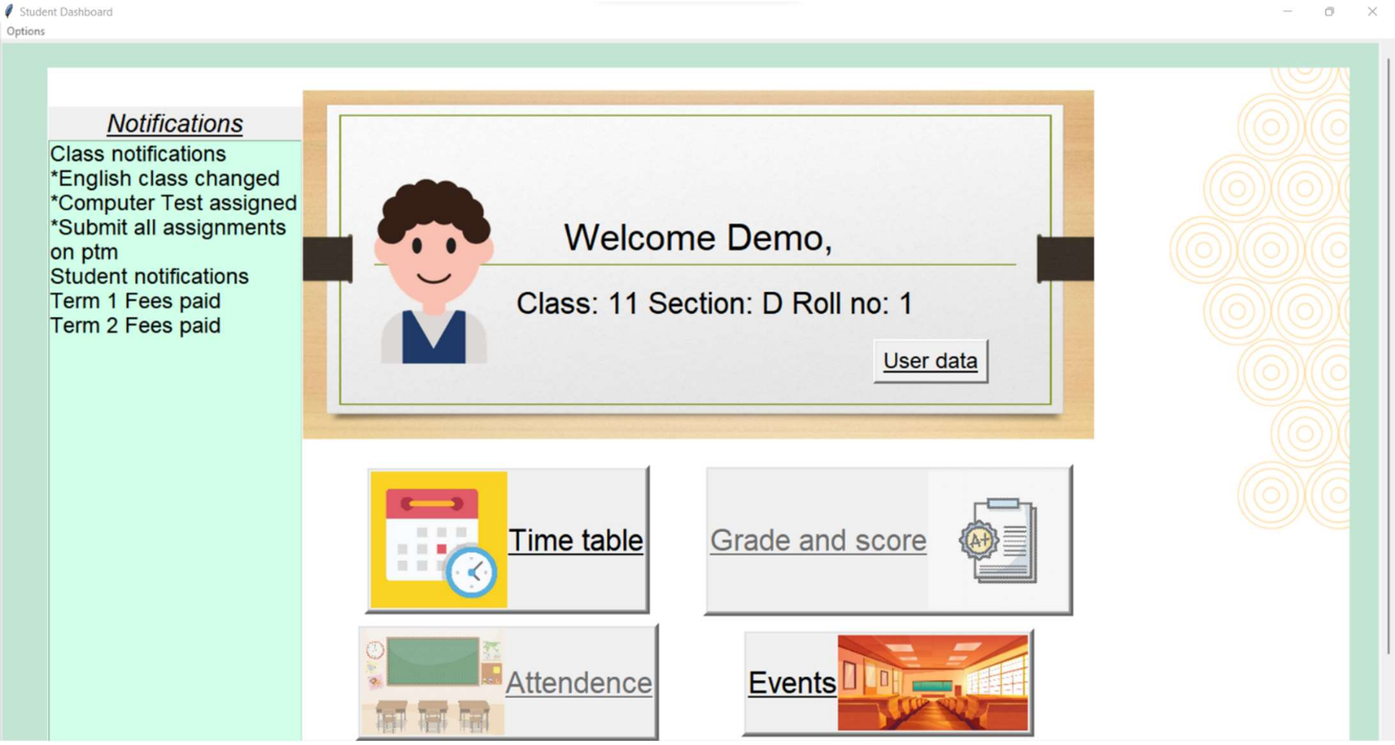


Notifications

Class noti
*English c
*Compute
*Submit al
on ptm
Student n
Term 1 Fe
Term 2 Fe

Data

```
name: Demo
class: 11
section: D
roll no: 1
gender: Male
marks: {'English': 80, 'Physics': 80, 'Mathematics': 100, 'Chemistry': 90, 'Computer': 100}
```



Notifications

Class notifications
*English class changed
*Computer Test assigned
*Submit all assignments
on ptm
Student notifications
Term 1 Fees paid
Term 2 Fees paid

Welcome Demo,

Class: 11 Section: D Roll no: 1

CONCLUSION

Python is an object-oriented, high-level language. It is indeed easy learn and use. It's inbuilt functions and libraries, and multiple other external libraries with which it can be integrated help all developers to easily solve a problem, and provide a suitable and necessary solution to it. We have built a comprehensive School Management System application, which is not only easy to use, but is also flexible, which allows the user, be it Admins, Teachers or Students to use it hassle free. Students can use it to see their timetable, attendance etc. and is also a necessary tool for managing school in these dire times.

This project, though optimized, still has a lot of scope for improvement in the coming future. Some of the ideas we would like to propose are as follows:

- Student behavior and participation analytics
- Integration with cloud-based database management
- Track student activity in class
- Integrate study group discussions among teacher and student or a group of students
- All feature for students to interact with each other according to section and class
- Application version for smart phone users

BIBLIOGRAPHY

- Computer book – Sumita Arora
- https://www.w3schools.com/python/default.asp
- https://opensource.com/resources/python
- https://www.geeksforgeeks.org/python-gui-tkinter/
- https://www.geeksforgeeks.org/python-pillow-a-fork-of-pil/
- https://stackoverflow.com