ATTENDANCE MANAGEMENT SYSTEM

SUBMITTED BY:

1. Aya Yousry AboElabas section: 4
2. Habiba Amr Ali section: 5
3. Habiba Abdelsamea Abdelhamid section:5
4. Mohamed Sayed Taha Sakr section: 12
5. Mohamed Adel Mohamed section: 15
6. Maram Nasser Farid section:16
7. Marwa Mohamed Samy section:16

Abstract:

A student attendance management system is a software tool that assists educational institutions in tracking and managing student attendance effectively . tracks student attendance without using paper .

Teachers can quickly and accurately record student attendance without putting in much effort.

Our system contains three modules :

1_admin module: the admin plays an important role in the management of this system as he/she can control and organize the number of teachers and students who can access the system.

2_teacher module: which can manage the attendance of students by Add, Edit or Delete.

3_student module: which can see his/her attendance to class.

_Tools, and Libraries used in the system .

Gui : python .

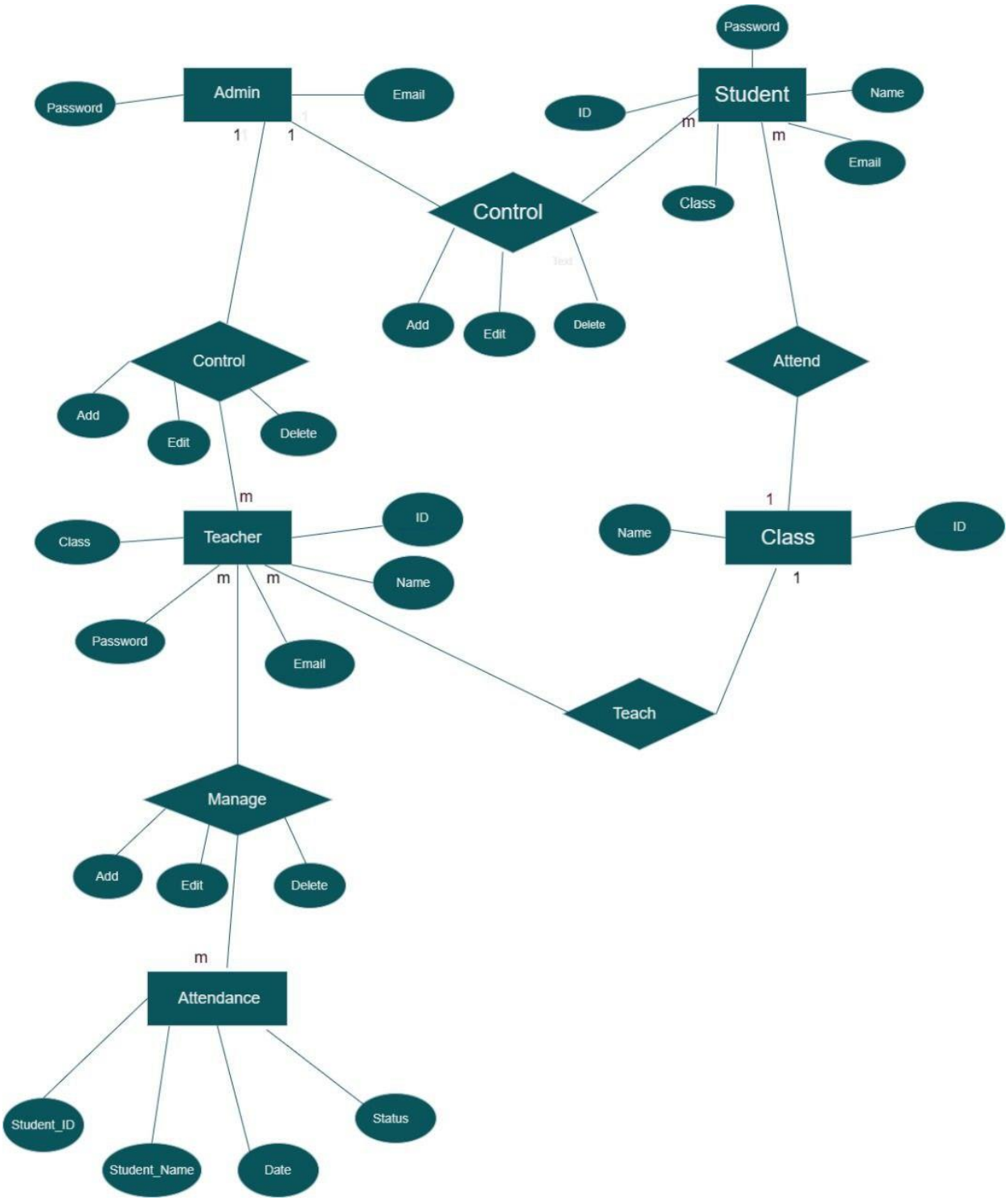
Database: SQL .

Libraries : thinker : used to make GUI.

sqlite3 : used for connection.

Entities Relationship Diagram (ERD)

System database



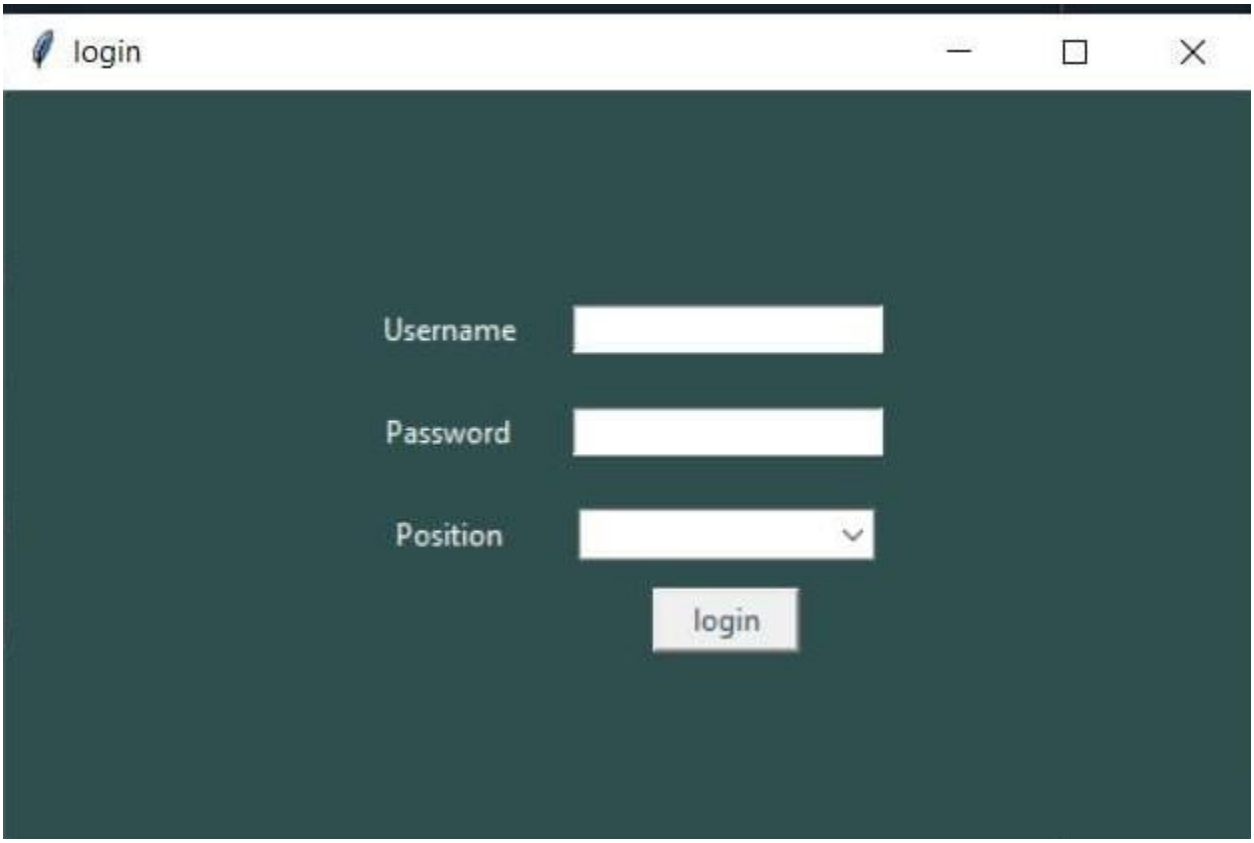
Overview

Project User Interface:

```
import sqlite3

def start():
    global con
    con=sqlite3.connect('System_DB.db')

    global cursor
    cursor=con.cursor()


def end():
    con.commit()
    con.close()


def addStudent(ID,Email,Name,Passowrd,Class):
    start()
    cursor.execute('insert into students values (?,?,?,?,?)',(ID,Email,Name,Passowrd,Class))
    end()


def addTeacher(ID,Email,Name,Passowrd,Class):
    start()
    cursor.execute('insert into teachers values (?,?,?,?,?)',(ID,Email,Name,Passowrd,Class))
    end()
```

This is a part of code that make connection between GUI and data base , we use library Sqlite3 to connect and send command to data base , and use some function.

1_login GUI



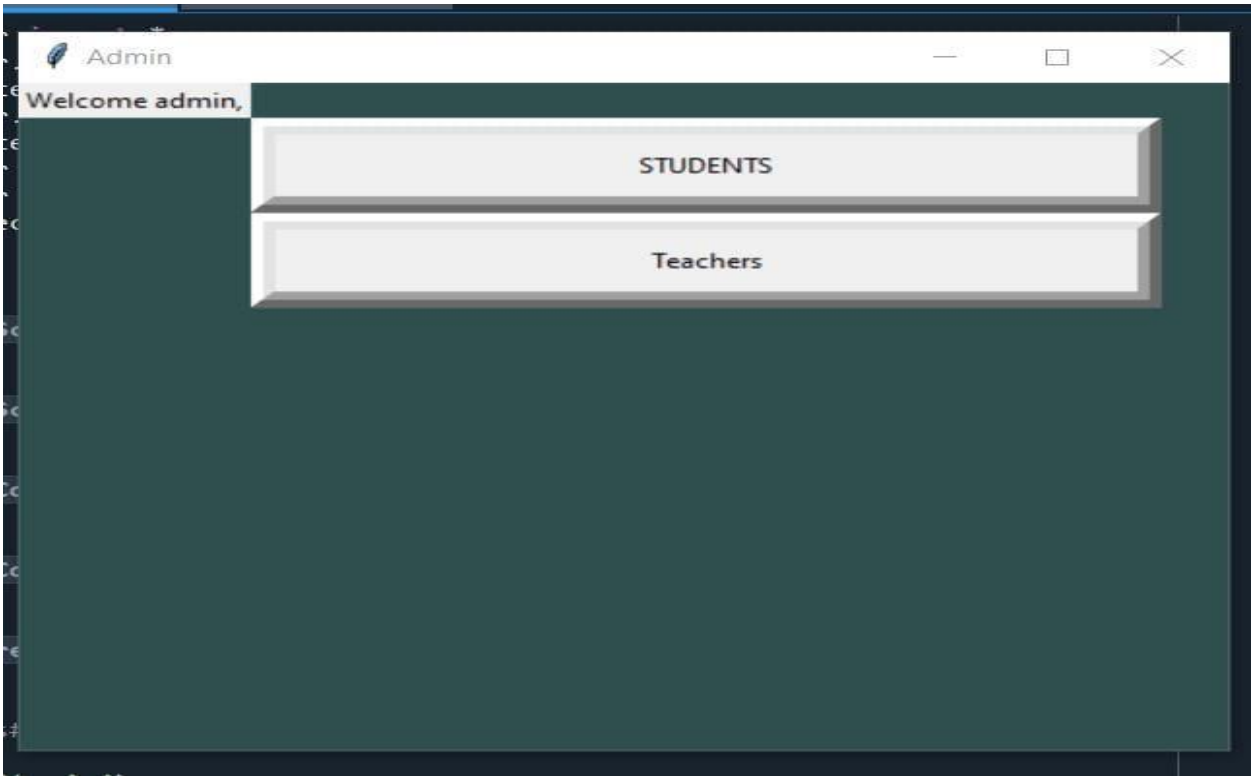
In the Login GUI you can enter your username , password and your position to log in to your homepage. If you entered one of them incorrectly or you didn’t enter data in one of the fields , a warning text will appear to user.

```
# ADMIN
if(Position=='admin'):

    if(Email=='ADMIN' and Passowrd=='1111'):
        root.destroy()
        AdminScreen()
    else: # Wrong Values
        tkinter.messagebox.showwarning('System','Your Email Or Passowrd is Wrong , Try Again!')

# STUDENT
elif(Position=='student'):

    flag = Connection_DB.searchStudentPassowrdAndEmail(Passowrd, Email)

    if(flag==True):
        #config.studentData = Connection_DB.searchStudentName(Passowrd, Email)
        global StudentData
        StudentData=Connection_DB.searchStudentName(Passowrd, Email)
        root.destroy()
        StudentScreen(StudentData)
    else:
        tkinter.messagebox.askokcancel('System','Your Email Or Passowrd is Wrong , Try Again!')

# TEACHER
else: # Position == 'teacher'

    flag = Connection_DB.searchTeacherPassowrdAndEmail(Passowrd, Email)

    if(flag==True):
        teacherClass=Connection_DB.searchTeacherClass(Email, Passowrd)
        root.destroy()
        TeacherScreen(teacherClass)
    else:
        tkinter.messagebox.askokcancel('System','Your Email Or Passowrd is Wrong , Try Again!')
```

This is a piece of code that allows user select his position and check their email address and password in the data base if there are correct. it makes him go to the user's home page,

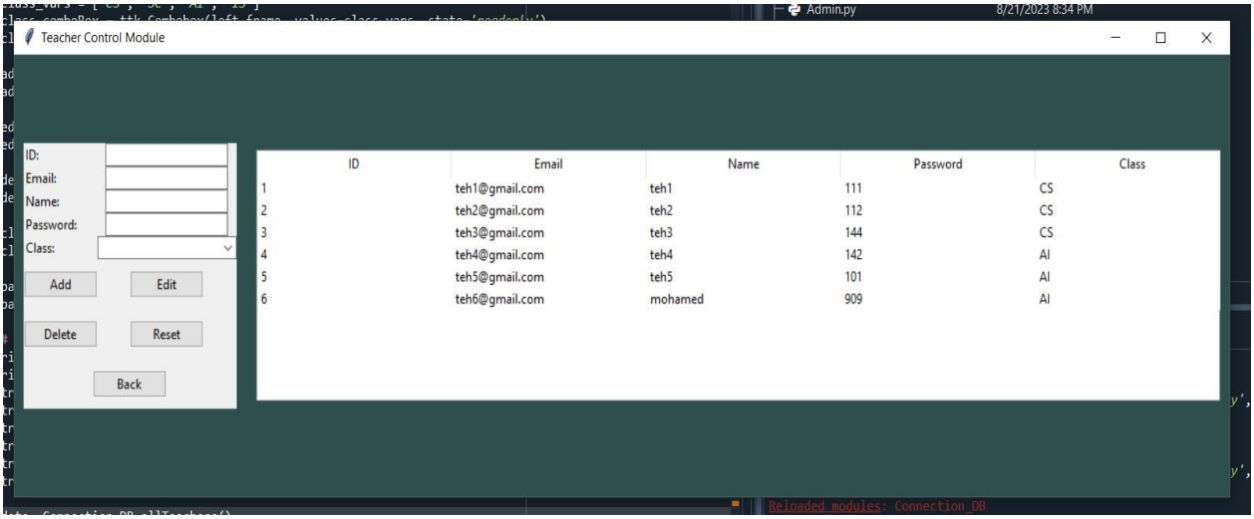
In the event that it is not correct a warning massage appears to user, making him enter the data .

2_Home page for admin



Each button shown on the page takes admin to another page, student button to “student control module” and teacher button to “teacher control module” . which can add, edit and delete the items from it.

3-Teacher control module



In this GUI admin has three buttons to use all of them is connect with database

1_Add: can add a new teacher to the system

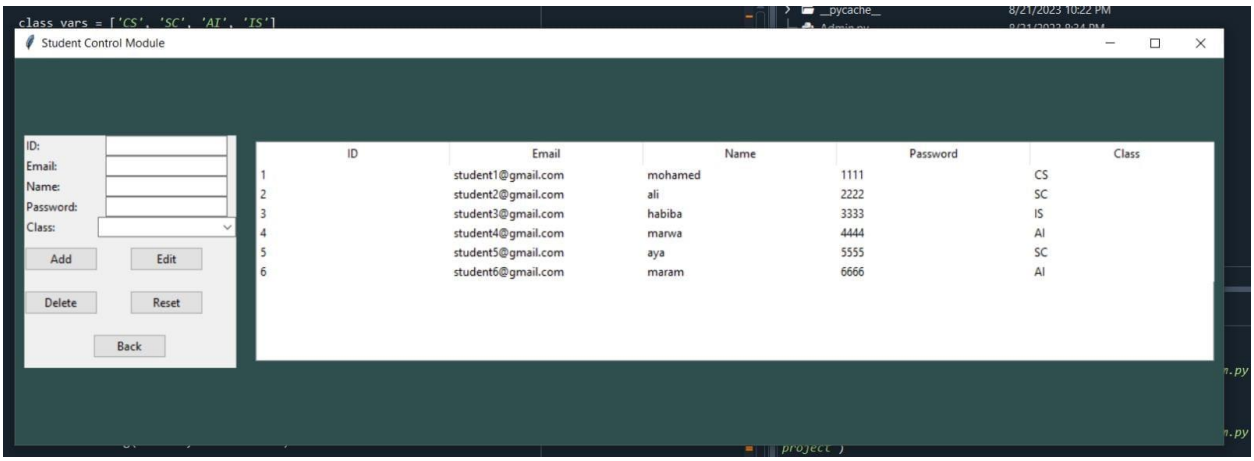
2_ Delete: can delete data of teacher from the system by only choose record

3_Edit: use this when you need to make data modification to record

Also have two buttons 1_reset: to clear fields 2_back: to return admin to home page

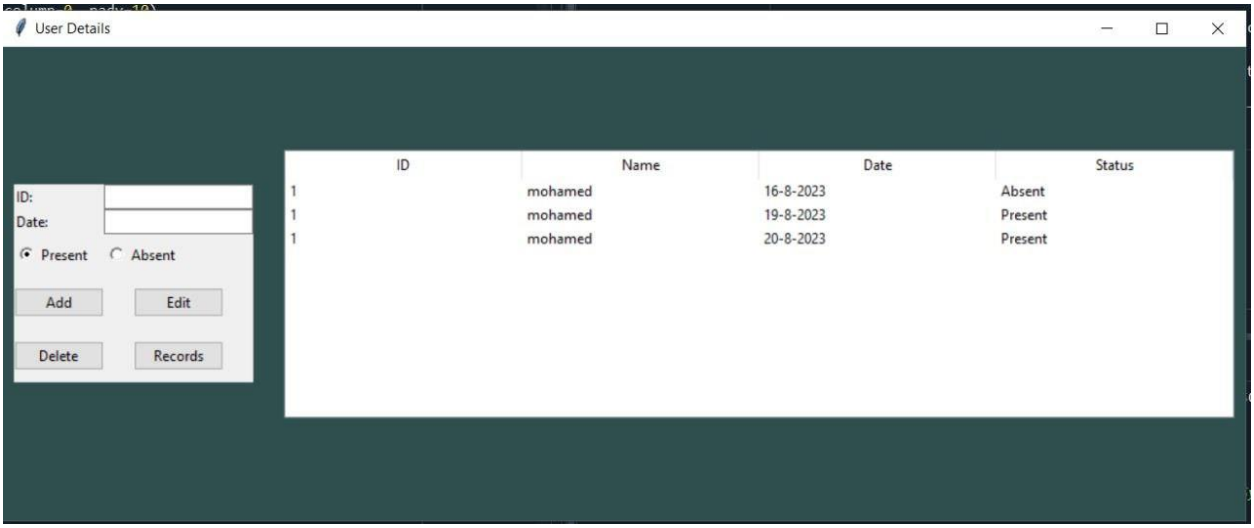
You can add the teacher details here. His /Her name, Username and password which is required to login. In edit or delete button you need to enter ID required .

4-Student control module



This window is the same as the teacher . Here you can create changes or add new student in the student's database.

5_Home page for teacher



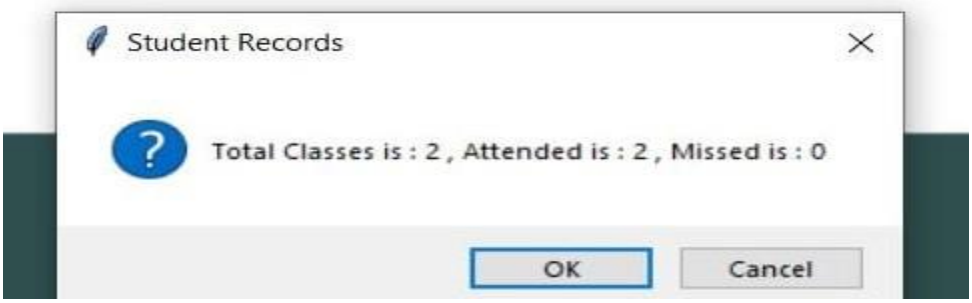
In this GUI teacher can managing student attendance effectively only student in his/her class by using his /her ID . has four buttons to use all of them is connect with database

1_Add: when teacher write id of the student and chose its state , add student to the table and in data base

2_ Delete: can delete data of student from the system by only choose record

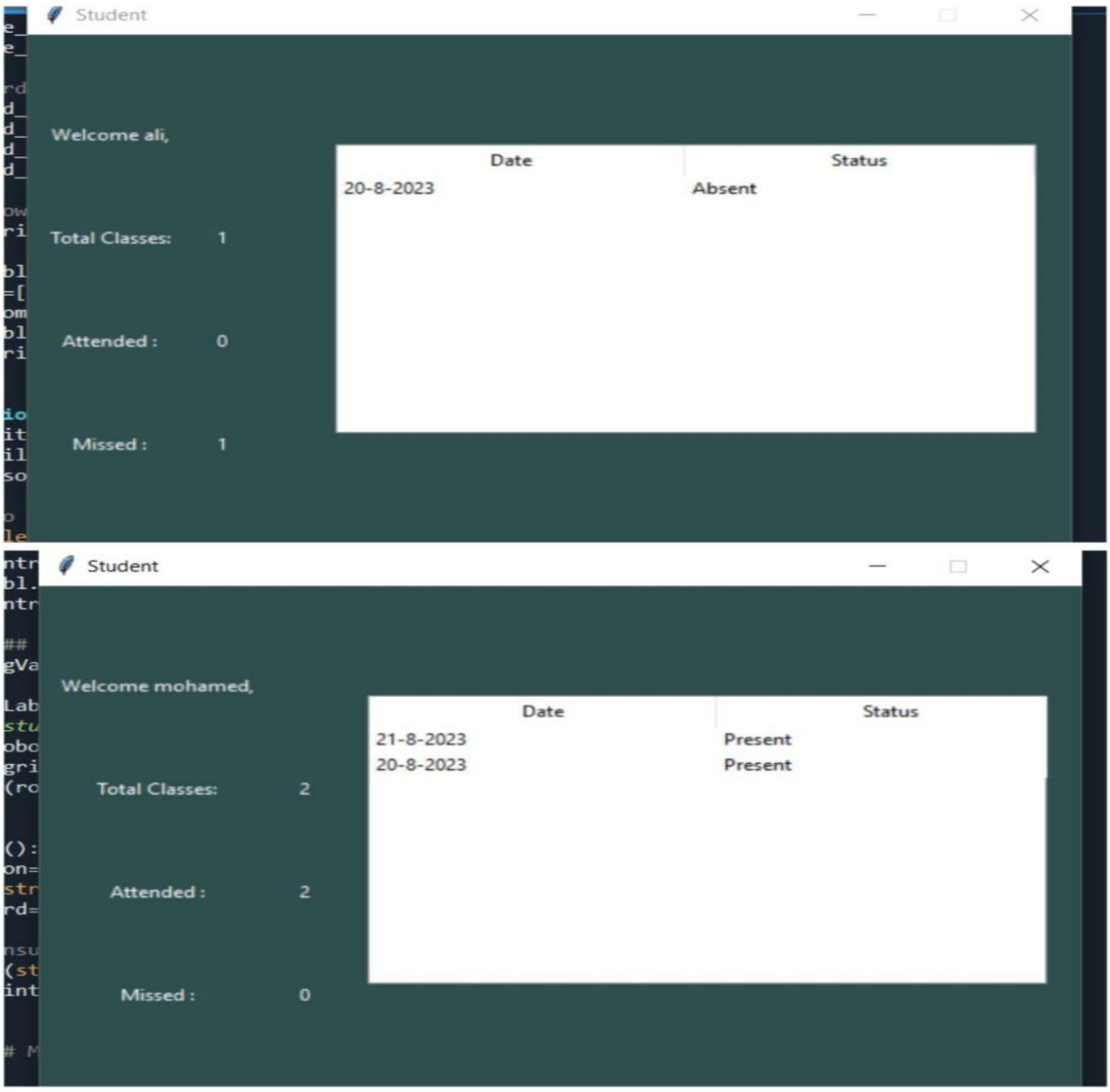
3_Edit: use this when you need to make data modification to record

4_Recorde: when teacher add id of the student , show the teacher the number of classes attended and not attended by the student, Appear as a text.



If teacher enter incorrect ID or not available one a warning text will appear

6_Home page for student



This is the interface for students. No buttons are included. The home page is designed to show all the details of the attendance.

Conclusion

Finally, in the student attendance management system, the outcome of all the hard work done for the project.

It is software to help the user to work with the attendance and updating data.

The user interface of it is very friendly and can be easily used by anyone, It also decreases the amount of time taken to write details and other models.

All the details about students, teachers and their other tasks can only be seen by the verified users. In the end, we can say that this software is performing all the tasks accurately and is doing the work for which it is made and this system can be implemented in N numbers of colleges and schools.