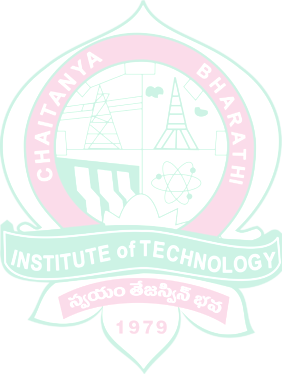




**E-School Management System**

**Engineering Exploration (20MEC03)**

**Project Seminar**

**By: -**

Aarthi – 160120737001

Akshita – 160120737002

Harini – 160120737003

Krishna Prasanna – 160120737004

Manasa – 160120737005

## B.E. Information Technology Semester II

**Under the guidance of**

I.Sucharitha,

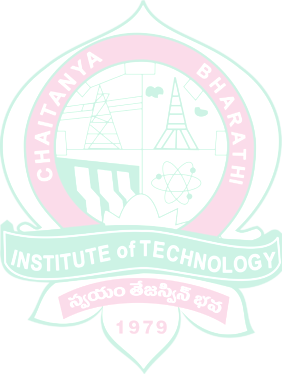
Assistant Professor

**Affiliation**

Department of Information Technology, CBIT

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **S. No** | **TITLE** | **PAGE** |
| 1. | **Abstract** | 1 |
| 2. | **List of Figures** | 2 |
| 3. | **Introduction** | 3 |
| 4. | **Problem Statement** | 4 |
| 5. | **System Design** | 5 |
| 6. | **Implementation** | 6 |
| 7. | **Results** | 37 |
| 8. | **Conclusion** | 47 |
| 9. | **Future Scope** | 48 |
| 10. | **References** | 49 |

**ABSTRACT**

Today, in this competitive world every work has become computerized. The manual way of working has become very hectic, time consuming as well as very difficult. An E-School Management System is an information management system for educational institutions to manage all students’ data. The school management system will manage all the work in any school in particular order so that the time requirement and complexity of the system will be reduced. The main objective of this project is to computerize the paperwork in the system and automate the work. E-School management system works as a centralized database and application that schools can easily access the system from anywhere based on the login credentials. School management system has equipped features that make it possible for admin to update and view their profile, attendance records, grade checks, report cards and fee receipts in just a few clicks. In SMS, Admin can access all the student details just by entering username and password. School management system being web based system, helps to update and record all the student data and to know about the status of the student’s fee, score and attendance.

This project uses python language, tkinter framework and SQLite database. The modules in this school management system are online registration, student information system, school fee management, attendance management and result management. With the E-School Management system, the management can get accurate results and efficient information can be processed in less time. The computerization is done such that the storage of all the details regarding students will be stored in the system which makes system centralized. The E-School Management System ensures data integrity, privacy and security in an open-access environment.

# LIST OF FIGURES

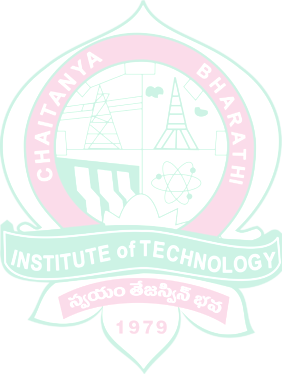
|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No** |
| Fig 5.1  Fig 6.1  Fig 6.2  Fig 6.3  Fig 6.4  Fig 7.1 | Flowchart  Admin Features  Student Features  Teacher Features  Parent Features  Start window | 5  8  16  23  33  38 |
| Fig 7.2 | Log in As window | 38 |
| Fig 7.3 | Login window | 39 |
| Fig 7.4 | Dashboard of Admin | 39 |
| Fig 7.5 | Admin registering student details | 39 |
| Fig 7.6 | Message of successful insertion | 40 |
| Fig 7.7 | Admin registering teacher details | 40 |
| Fig 7.8 | Message of successful insertion | 40 |
| Fig 7.9 | Admin updating student fee | 40 |
| Fig 7.10 | Message of successful fee update | 41 |
| Fig 7.11 | Viewing student fee | 41 |
| Fig 7.12 | Displaying fee | 41 |
| Fig 7.13 | Student Dashboard | 41 |
| Fig 7.14 | Student viewing personal info | 42 |
| Fig 7.15 | Viewing fee details | 42 |
| Fig 7.16 | Student viewing attendance | 42 |
| Fig 7.17 | Student viewing marks | 42 |
| Fig 7.18 | Access denying message | 43 |
| Fig 7.19 | Changing password | 43 |
| Fig 7.20 | Message of successful password change | 43 |
| Fig 7.21 | Parent dashboard window | 43 |
| Fig 7.22 | Viewing fee details | 44 |
| Fig 7.23 | Parent viewing attendance | 44 |
| Fig 7.24 | Student viewing marks | 44 |
| Fig 7.25 | Teacher Dashboard window | 45 |
| Fig 7.26 | Teacher viewing student details | 45 |
| Fig 7.27 | Student details | 45 |
| Fig 7.28 | Viewing student details of same class | 45 |
| Fig 7.29 | Student details belonging to same class | 46 |
| Fig 7.30 | Updating attendance | 46 |
| Fig 7.31  Fig 7.32  Fig 7.33  Fig 7.34  Fig 7.35  Fig 7.36 | Entering attendance  Updating marks  Entering marks  Changing password window  Message of successful password change  Logout confirmation message for all users | 46  46  47  47  47  47 |

# INTRODUCTION

E-School is a complete school information management solution. Today's schools need to manage more information than ever before. Without a solid internal infrastructure for teachers and administrators to share data, critical school and student information can be lost, or worse leading to a host of problems that can effect of a school's image and endurance. To remain competitive, school needs a simple solution that can run individual function, connect their entire operation, use the web as a key communication tool and simplify day to day operational responsibilities, giving staff more time with students. E-School automates various activities of school and optimizes the use of premium resources. Concerned authorities can now easily and seamlessly use the system to access student details, otherwise a time consuming and tedious task. School Management Software offers a strong backbone to the education system. It integrates information from various areas and brings them under one roof. This reduces the hassle involved in performing the task.

# PROBLEM STATEMENT

In educational Institutions the administrators and the teachers have to handle records of many students and the maintenance of such details is very difficult. There is scope for errors and retrieving of student details also becomes difficult. The space and cost consumed in physical storage of student records is huge. There is also a risk of files getting lost because of human error. Hence there is a need to upgrade the system with a computer based information management system.

**SYSTEM DESIGN**

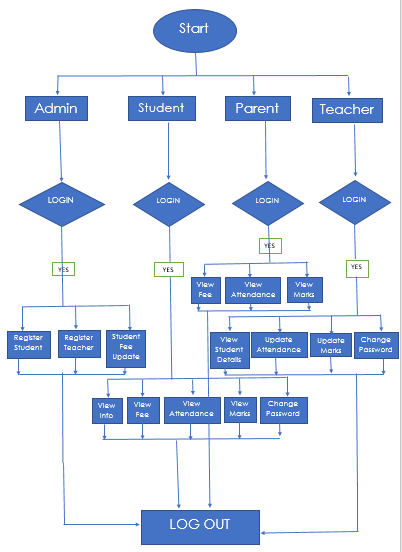
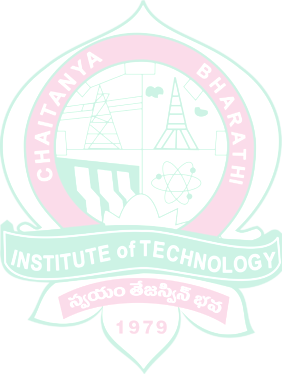


Fig 5.1: Flowchart

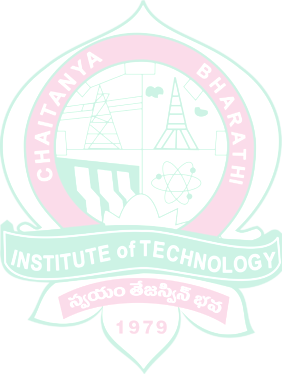
**IMPLEMENTATION**

There are four main users for this system; these are Admin, student, teacher and the parent. Each user can perform several different functions during the use of the system. These are user-friendly functions to make the system more effective and efficient.

Users have their respective usernames and passwords using which they can login.

**LOG-IN:**

* This will help users to login into the system using institute id and password. A user who has the valid id and password can only log in to their respective accounts.
* It will help the authentication of the user who enters the system. The module provides a layer of security over the system as only authorized personnel can login into the system.
* This prevents any anonymous person to enter the system and mishandle the records. It is better than the manual method as they do not have any security measure of who can access the system and who cannot.



## main.py:

## In main module, we will be importing all modules which are Admin, student, teacher and parent.

## Code of main.py

import Admin

import student

import teacher

import parent

from tkinter import \*

window = Tk()

window.geometry('1180x600')

window.title('Start Window')

def start():

    window = Toplevel()

    window.geometry('1180x600')

    Label(window, text='Login As', font='Times 20 bold').place(x=450,y=80)

    Button(window, text='Admin', command=Admin.start, width=25, font=("Times New", 10), bd=5,

           bg="light blue").place(x=450,y=120)#.grid(row=2, column=0, sticky=N + S + E + W)

    Button(window, text='Student', command=student.start, width=25, font=("Times New", 10), bd=5,

           bg="light blue").place(x=450,y=150)#.grid(row=3, column=0, sticky=N + S + E + W)

    Button(window, text='Parent', command=parent.start, width=25, font=("Times New", 10), bd=5,

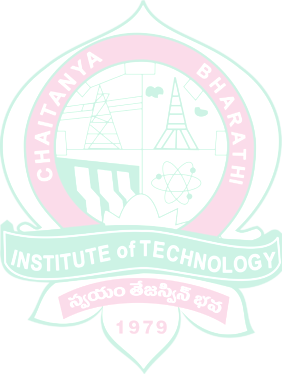
           bg="light blue").place(x=450,y=180)#.grid(row=4, column=0, sticky=N + S + E + W)

    Button(window, text='Teacher', command=teacher.start, width=25, font=("Times New", 10), bd=5,

           bg="light blue").place(x=450,y=210)#.grid(row=5, column=0, sticky=N + S + E + W)

##############################################################################################

img=PhotoImage(file="background1.png")

label=Label(window,image=img)

label.place(x=0,y=0)

Button(window, text='start', command=start, width=20, font=("Times New", 15),bd=-5).place(x=410, y=290)

Label(window,text='E-SCHOOL',font=("Times New",60)).place(x=320,y=120)

window.mainloop()

## Admin.py:

## As shown, the admin can register student details , teacher details and update the fee status as per the fee payment.

## The admin will register the student as the student is new in the educational institute. It will be formed like a structure where all the their details such as full name , class , mobile number , student password will be filled.

## 

Fig 6.1: Admin Features

## Code of Admin.py

from tkinter.messagebox import \*

import sqlite3

con = sqlite3.Connection('student\_records.db')

cur = con.cursor()

con1 = sqlite3.Connection('teacher\_records.db')

cur1 = con1.cursor()

from tkinter import \*

def start():

    window = Toplevel()

    window.geometry('1080x600')

    window.title('Login Window')

    img = PhotoImage(file="background1.png")

    label = Label(window, image=img)

    label.place(x=0, y=0)

    Label(window, text='Username  ', font=("Times New", 20)).grid(row=1, column=1)  # v=id

    v = Entry(window, width=25, font=("Times New", 18), bd=5, bg="light blue")

    v.grid(row=1, column=2)

    Label(window, text='Password  ', font=("Times n=New", 20)).grid(row=2, column=1)

    vv = Entry(window, show='\*', width=25, font=("Times New", 18), bd=5, bg="light blue")  # vv=password

    vv.grid(row=2, column=2)

    def login():

        if (((int(v.get()) != 123) or (int(vv.get()) != 123))):

            showwarning('Login Failed', 'Incorrect Id Or Password Used Or ')

        else:

            menu = Toplevel()

            menu.geometry('1080x600')

            menu.title('Menu Window')

            l = Label(menu, text='Dashboard :', font='Times 20 bold')

            l.grid(row=0, column=0)

            Label(menu).grid(row=1, column=100, rowspan=30)

            def option1():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('Register Students Details')

                    l = Label(root, text='Register Student Details :', font='Times 20 bold')

                    l.grid(row=0, column=3)

                    l = Label(root, text=' ')

                    l.grid(row=1, column=0)

                    l = Label(root, text='Enter Student ID : ')

                    l.grid(row=2, column=0)

                    c = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # c=sid

                    c.grid(row=2, column=1)

                    l = Label(root, text='Enter full name : ')

                    l.grid(row=3, column=0)

                    d = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # d=fname

                    d.grid(row=3, column=1)

                    l = Label(root, text='Enter father name : ')

                    l.grid(row=4, column=0)

                    e = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # e=fname

                    e.grid(row=4, column=1)

                    l = Label(root, text='Enter mother name : ')

                    l.grid(row=5, column=0)

                    g = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # e=mname

                    g.grid(row=5, column=1)

                    l = Label(root, text='Enter class : ')

                    l.grid(row=6, column=0)

                    f = StringVar(root)  # f=class no.

                    f.set("Select class")  # default value

                    w = OptionMenu(root, f, "Class 1", "Class 2", "Class 3", "Class 4", "Class 5", "Class 6", "Class 7",

                                   "Class 8", "Class 9", "Class 10", "Class 11", "Class 12")

                    w.grid(row=6, column=1)

                    l = Label(root, text='Enter mobile no. : ')

                    l.grid(row=7, column=0)

                    q = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # q=total fee

                    q.grid(row=7, column=1)

                    l = Label(root, text='Enter year due fee : ')

                    l.grid(row=8, column=0)

                    m = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # q=mobile no.

                    m.grid(row=8, column=1)

                    l = Label(root, text='Enter Password : ')

                    l.grid(row=9, column=0)

                    p = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # q=pswd

                    p.grid(row=9, column=1)

                    cur.execute(

                        "create table if not exists student(sid number primary key, name varchar(20), fname varchar(20),mname varchar(20), clno number(4),mbno number(10), yearduefee number(6),pswd number(6),paidfee number(6),duefee number(6), eng\_at number(4), hin\_at number(4), mat\_at number(4), sc\_at number(4), ssc\_at number(4), san\_at number(4), eng\_m number(4), hin\_m number(4), mat\_m number(4), sc\_m number(4), ssc\_m number(4), san\_m number(4))")

                    def insertinstudent():

                        cur.execute("select \* from student where sid=?", (int(c.get()),))

                        cpy = cur.fetchone()

                        try:

                            cur.execute("insert into student values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)", (

                            int(c.get()), d.get(), e.get(), g.get(), f.get(), int(q.get()),int(m.get()),int(p.get()), 0, int(m.get()), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0))

                            con.commit()

                            showinfo('Information', 'Student Insertion Successful')

                        ################################################################################################################

                        except sqlite3.IntegrityError as error:

                            showwarning('Error', 'Already exists!')

                    Button(root, text='Insert Data Of Student', command=insertinstudent, width=25,

                           font=("Times New", 10), bd=5, bg="light blue").grid(row=10, column=0, sticky=N + S + E + W)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=11, column=0, sticky=N + S + E + W)

                    root.mainloop()

            Button(menu, text='Register Student details', command=option1, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=1, column=0, sticky=N + S + E + W)

################################################################################################

            def option2():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('Register Teacher Details')

                    l = Label(root, text='Register Teacher Details :', font='Times 20 bold')

                    l.grid(row=0, column=3)

                    l = Label(root, text=' ')

                    l.grid(row=1, column=0)

                    l = Label(root, text='Enter Teacher ID : ')

                    l.grid(row=2, column=0)

                    c = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # c=sid

                    c.grid(row=2, column=1)

                    #xp = c.get()

                    l = Label(root, text='Enter first name : ')

                    l.grid(row=3, column=0)

                    d = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # d=fname

                    d.grid(row=3, column=1)

                    l = Label(root, text='Enter last name : ')

                    l.grid(row=4, column=0)

                    e = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # e=lname

                    e.grid(row=4, column=1)

                    l = Label(root, text='Enter Password : ')

                    l.grid(row=5, column=0)

                    q = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # q=total fee

                    q.grid(row=5, column=1)

                    cur1.execute(

                        "create table if not exists teacher(tid number primary key, fname varchar(20), lname varchar(20), pswd number(6))")

                    def insertinteacher():

                        cur1.execute("select \* from teacher where tid=?", (int(c.get()),))

                        cpy = cur1.fetchone()

                        try:

                            cur1.execute("insert into teacher values(?,?,?,?)", (

                            int(c.get()), d.get(), e.get(), int(q.get())))

                            con1.commit()

                            showinfo('Information', 'Teacher Insertion Successful')

                        ################################################################################################################

                        except sqlite3.IntegrityError as error:

                            showwarning('Error', 'Already exists!')

                    Button(root, text='Insert Data Of Teacher', command=insertinteacher, width=25,

                           font=("Times New", 10), bd=5, bg="light blue").grid(row=7, column=0, sticky=N + S + E + W)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=8, column=0, sticky=N + S + E + W)

                    root.mainloop()

            Button(menu, text='Register Teacher details', command=option2, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=2, column=0, sticky=N + S + E + W)

            ##################################################################

            def option3():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('Student Fee Updation')

                    l = Label(root, text='Student Fee Updation :', font='Times 20 bold')

                    l.grid(row=0, column=3)

                    l = Label(root, text=' ')

                    l.grid(row=1, column=0)

                    l = Label(root, text='Enter Student ID : ')

                    l.grid(row=2, column=0)

                    g = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # g=sid

                    g.grid(row=2, column=1)

                    l = Label(root, text='Enter fee paid : ')

                    l.grid(row=3, column=0)

                    h = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # h=fee paid

                    h.grid(row=3, column=1)

                    def update1():

                        p = int(g.get())

                        cur.execute("update student set duefee=duefee-? where sid=?", (int(h.get()), p))

                        con.commit()

                        cur.execute("update student set paidfee=paidfee+? where sid=?", (int(h.get()), p))

                        con.commit()

                        showinfo('Information', 'Fee Updation Successful')

                    Button(root, text='Update Fee', command=update1, width=25, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=4, column=0, sticky=N + S + E + W)

                    l = Label(root, text=' ')

                    l.grid(row=5, column=0)

                    l = Label(root, text='Enter Student ID to see details : ')

                    l.grid(row=6, column=0)

                    i = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # i=sid

                    i.grid(row=6, column=1)

                    def show3():

                        w = [(int(i.get()))]

                        cur.execute("select duefee from student where sid=?", w)

                        x = cur.fetchall()

                        showinfo('Result', x)

                    Button(root, text='Show Student due fee ', command=show3, width=25, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=7, column=0, sticky=N + S + E + W)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=8, column=0)

                    root.mainloop()

            Button(menu, text='Student Fee Updation', command=option3, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=3, column=0, sticky=N + S + E + W)

            ######################################################################################################################################################################

            def logout():

                ans = askyesno('Confirmation', 'Do You Want To Logout?')

                if (ans == True):

                    menu.destroy()

            Button(menu, text='Logout', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                row=8, column=0, sticky=N + S + E + W)

            menu.mainloop()

    Button(window, text='Login', command=login, width=20, font=("Times New", 15), bd=5, bg="light blue").grid(row=3,

                                                                                                              column=2,

                                                                                                              sticky=N + S + E + W)

# student.py

# As shown, the student can view their personal details, view fee status, view attendance , view marks and change password. Their attendance and marks/results can be viewed with respect to their subjects . The students can also change their password by entering the password initially given by the admin.

# 

Fig 6.2: Student Features

# Code of student.py

from tkinter import \*

from tkinter.messagebox import \*

import sqlite3

con = sqlite3.Connection('student\_records.db')

cur = con.cursor()

def start():

    window = Toplevel()

    window.geometry('1080x600')

    window.title('Login Window')

    Label(window, text='Username  ', font=("Times New", 20)).grid(row=1, column=1)  # v=id

    v = Entry(window, width=25, font=("Times New", 18), bd=5, bg="light blue")

    v.grid(row=1, column=2)

    Label(window, text='Password  ', font=("Times n=New", 20)).grid(row=2, column=1)

    vv = Entry(window, show='\*', width=25, font=("Times New", 18), bd=5, bg="light blue")  # vv=password

    vv.grid(row=2, column=2)

    def login():

        id=[(int(v.get()))]

        z = cur.execute("select pswd from student where sid=?", id)

        x=z.fetchone()

        if ((int(vv.get()) != x[0])):

            showwarning('Login Failed', 'Incorrect Id Or Password Used Or ')

        else:

            menu = Toplevel()

            menu.geometry('1080x600')

            menu.title('Menu Window')

            l = Label(menu, text='Dashboard :', font='Times 20 bold')

            l.grid(row=0, column=0)

            Label(menu).grid(row=1, column=100, rowspan=30)

            #############################################################

            def option1():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('View Window')

                    column\_number = 0

                    Label(root, text="Student ID :", font=("Times New", 20)).grid(row=1, column=column\_number)

                    Label(root, text="Name       :", font=("Times New", 20)).grid(row=2, column=column\_number)

                    Label(root, text="Father Name:", font=("Times New", 20)).grid(row=3, column=column\_number)

                    Label(root, text="Mother Name:", font=("Times New", 20)).grid(row=4, column=column\_number)

                    Label(root, text="Class      :", font=("Times New", 20)).grid(row=5, column=column\_number)

                    Label(root, text="Mobile no. :", font=("Times New", 20)).grid(row=6, column=column\_number)

                    z = cur.execute("select \*from student where sid=?", id)

                    for column\_number, row in enumerate(z):

                        Label(root, text="" + str(row[0]), font=("Times New", 18)).grid(row=1, column=column\_number+1)

                        Label(root, text="" + str(row[1]), font=("Times New", 18)).grid(row=2, column=column\_number+1)

                        Label(root, text="" + str(row[2]), font=("Times New", 18)).grid(row=3, column=column\_number+1)

                        Label(root, text="" + str(row[3]), font=("Times New", 18)).grid(row=4, column=column\_number+1)

                        Label(root, text="" + str(row[4]), font=("Times New", 18)).grid(row=5, column=column\_number+1)

                        Label(root, text="" + str(row[5]), font=("Times New", 18)).grid(row=6, column=column\_number+1)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                        row=8, column=0, sticky=N + S + E + W)

                    root.mainloop()

            Button(menu, text='View info', command=option1, width=25, font=("Times New", 10), bd=5,

                        bg="light blue").grid(row=2, column=0, sticky=N + S + E + W)

            #########################################################################################################################################

            def option2():

                root = Toplevel()

                root.geometry('1080x600')

                root.title('View Fee')

                l = Label(root, text='FEE:', font='Times 20 bold')

                l.grid(row=0, column=3)

                l = Label(root, text=' ')

                l.grid(row=4, column=0)

                cur.execute("select yearduefee from student where sid=?", id)

                x = cur.fetchall()

                Label(root, text=' TOTAL FEE : ' + str(x[0][0]), font='Times 16 bold').grid(column=1, row=1)

                cur.execute("select duefee from student where sid=?", id)

                y = cur.fetchall()

                Label(root, text=' DUE FEE : ' + str(y[0][0]), font='Times 16 bold').grid(column=1, row=2)

                cur.execute("select paidfee from student where sid=?", id)

                z = cur.fetchall()

                Label(root, text=' PAID FEE : ' + str(z[0][0]), font='Times 16 bold').grid(column=1, row=3)

                def logout():

                    ans = askyesno('Confirmation', 'Do You Want To Exit?')

                    if (ans == True):

                        root.destroy()

                Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5,

                       bg="light blue").grid(row=5, column=0)

                root.mainloop()

            Button(menu, text='View Fee', command=option2, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=3, column=0, sticky=N + S + E + W)

            ##################################################################

            def option3():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('View Window')

                    Label(root, text="Attendance", font=("Times New", 20)).grid(row=0, column=4)

                    column\_number = 0

                    Label(root, text="English    :", font=("Times New", 20)).grid(row=1, column=column\_number)

                    Label(root, text="Hindi      :", font=("Times New", 20)).grid(row=2, column=column\_number)

                    Label(root, text="Mathematics:", font=("Times New", 20)).grid(row=3, column=column\_number)

                    Label(root, text="Science    :", font=("Times New", 20)).grid(row=4, column=column\_number)

                    Label(root, text="Social     :", font=("Times New", 20)).grid(row=5, column=column\_number)

                    Label(root, text="Sanskrit   :", font=("Times New", 20)).grid(row=6, column=column\_number)

                    z = cur.execute("select eng\_at,hin\_at,mat\_at,sc\_at,ssc\_at,san\_at from student where sid=?", id)

                    for column\_number, row in enumerate(z):

                        Label(root, text="" + str(row[0]), font=("Times New", 18)).grid(row=1, column=column\_number + 1)

                        Label(root, text="" + str(row[1]), font=("Times New", 18)).grid(row=2, column=column\_number + 1)

                        Label(root, text="" + str(row[2]), font=("Times New", 18)).grid(row=3, column=column\_number + 1)

                        Label(root, text="" + str(row[3]), font=("Times New", 18)).grid(row=4, column=column\_number + 1)

                        Label(root, text="" + str(row[4]), font=("Times New", 18)).grid(row=5, column=column\_number + 1)

                        Label(root, text="" + str(row[5]), font=("Times New", 18)).grid(row=6, column=column\_number + 1)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                        row=8, column=0)

                    root.mainloop()

            Button(menu, text='View Attendance', command=option3, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=5, column=0, sticky=N + S + E + W)

            #########################################################################################################################################

            def option4():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('View Window')

                    Label(root, text="Marks", font=("Times New", 20)).grid(row=0, column=4)

                    column\_number = 0

                    Label(root, text="English    :", font=("Times New", 20)).grid(row=1, column=column\_number)

                    Label(root, text="Hindi      :", font=("Times New", 20)).grid(row=2, column=column\_number)

                    Label(root, text="Mathematics:", font=("Times New", 20)).grid(row=3, column=column\_number)

                    Label(root, text="Science    :", font=("Times New", 20)).grid(row=4, column=column\_number)

                    Label(root, text="Social     :", font=("Times New", 20)).grid(row=5, column=column\_number)

                    Label(root, text="Sanskrit   :", font=("Times New", 20)).grid(row=6, column=column\_number)

                    z = cur.execute("select eng\_m,hin\_m,mat\_m,sc\_m,ssc\_m,san\_m from student where sid=?", id)

                    for column\_number, row in enumerate(z):

                        Label(root, text="" + str(row[0]), font=("Times New", 18)).grid(row=1, column=column\_number + 1)

                        Label(root, text="" + str(row[1]), font=("Times New", 18)).grid(row=2, column=column\_number + 1)

                        Label(root, text="" + str(row[2]), font=("Times New", 18)).grid(row=3, column=column\_number + 1)

                        Label(root, text="" + str(row[3]), font=("Times New", 18)).grid(row=4, column=column\_number + 1)

                        Label(root, text="" + str(row[4]), font=("Times New", 18)).grid(row=5, column=column\_number + 1)

                        Label(root, text="" + str(row[5]), font=("Times New", 18)).grid(row=6, column=column\_number + 1)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                        row=8, column=0)

                    root.mainloop()

            Button(menu, text='View Marks', command=option4, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=6, column=0, sticky=N + S + E + W)

            ##########################################################################

            def option5():

             z = cur.execute("select clno from student where sid=?", id)

             x = z.fetchone()

             if ('Class 11' != x[0] and 'Class 12' != x[0]):

                    showwarning('Access denied', 'You can\'t change your password')

             else :

                root = Toplevel()

                root.geometry('1080x600')

                root.title('Change Password')

                l = Label(root, text='Change Password :', font='Times 20 bold')

                l.grid(row=0, column=3)

                l = Label(root, text=' ')

                l.grid(row=1, column=0)

                l = Label(root, text='Enter New Password : ')

                l.grid(row=3, column=0)

                h = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # h=fee paid

                h.grid(row=3, column=1)

                def update1():

                    cur.execute("update student set pswd=? where sid=?", (int(h.get()), id[0]))

                    con.commit()

                    showinfo('Information', 'Password Changed Successfully')

                Button(root, text='save & change', command=update1, width=25, font=("Times New", 10), bd=5,

                       bg="light blue").grid(row=4, column=0, sticky=N + S + E + W)

                def logout():

                    ans = askyesno('Confirmation', 'Do You Want To Exit?')

                    if (ans == True):

                        root.destroy()

                Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5,

                       bg="light blue").grid(row=8, column=0)

                root.mainloop()

            Button(menu, text='Change Password', command=option5, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=7, column=0, sticky=N + S + E + W)

            ############################################################################

            def logout():

                ans = askyesno('Confirmation', 'Do You Want To Logout?')

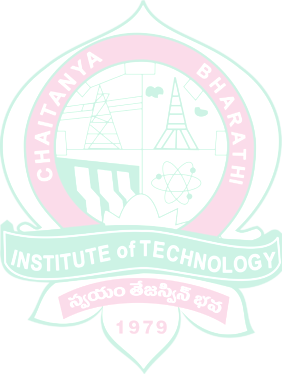
                if (ans == True):

                    menu.destroy()

            Button(menu, text='Logout', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                row=8, column=0, sticky=N + S + E + W)

            menu.mainloop()



    Button(window, text='Login', command=login, width=20, font=("Times New", 15), bd=5, bg="light blue").grid(

                row=3,

                column=2,

                sticky=N + S + E + W)

# teacher.py

# As shown, the teacher can view student’s details, update student’s attendance for the respective subject, update student marks, and change password. The teacher can view a particular student details by entering the student ID. They can also view the details of the entire class by entering the class Number. The teacher can also change their password by entering the password initially given by the admin.

# 

Fig 6.3: Teacher Features

**Code of teacher.py**

from tkinter.messagebox import \*

import sqlite3

con = sqlite3.Connection('student\_records.db')

cur = con.cursor()

con1 = sqlite3.Connection('teacher\_records.db')

cur1 = con1.cursor()

from tkinter import \*

def start():

    window = Toplevel()

    window.geometry('1080x600')

    window.title('Login Window')

    Label(window, text='Username  ', font=("Times New", 20)).grid(row=1, column=1)  # v=id

    v = Entry(window, width=25, font=("Times New", 18), bd=5, bg="light blue")

    v.grid(row=1, column=2)

    Label(window, text='Password  ', font=("Times n=New", 20)).grid(row=2, column=1)

    vv = Entry(window, show='\*', width=25, font=("Times New", 18), bd=5, bg="light blue")  # vv=password

    vv.grid(row=2, column=2)

    def login():

        id=[int(v.get())]

        z = cur1.execute("select pswd from teacher where tid=?", id)

        x=z.fetchone()

        if ((int(vv.get()) != x[0])):

            showwarning('Login Failed', 'Incorrect Id Or Password Used Or ')

        else:

            menu = Toplevel()

            menu.geometry('1080x600')

            menu.title('Menu Window')

            l = Label(menu, text='Dashboard :', font='Times 20 bold')

            l.grid(row=0, column=0)

            Label(menu).grid(row=1, column=100, rowspan=30)

            #############################################################

            def option1():

                root = Toplevel()

                root.geometry('1080x600')

                root.title('Student Details View')

                l = Label(root, text='View Student Details :', font='Times 20 bold')

                l.grid(row=0, column=3)

                l = Label(root, text=' ')

                l.grid(row=1, column=0)

                l = Label(root, text='Enter Student Id : ')

                l.grid(row=2, column=0)

                ui = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # ui=sid

                ui.grid(row=2, column=1)

                l = Label(root, text='Enter class : ')

                l.grid(row=4, column=0)

                f = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")  # f=clno.

                f.grid(row=4, column=1)

                def show1():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('View Window')

                    val = [(int(ui.get()))]

                    row\_number = 0

                    Label(root, text="Student ID :", font=("Times New", 20)).grid(row=1, column=row\_number)

                    Label(root, text="Name       :", font=("Times New", 20)).grid(row=2, column=row\_number)

                    Label(root, text="Father Name:", font=("Times New", 20)).grid(row=3, column=row\_number)

                    Label(root, text="Mother Name:", font=("Times New", 20)).grid(row=4, column=row\_number)

                    Label(root, text="Class      :", font=("Times New", 20)).grid(row=5, column=row\_number)

                    Label(root, text="Mobile no. :", font=("Times New", 20)).grid(row=6, column=row\_number)

                    z = cur.execute("select \*from student where sid=?", val)

                    for row\_number, row in enumerate(z):

                        Label(root, text="" + str(row[0]), font=("Times New", 18)).grid(row=1, column=row\_number + 1)

                        Label(root, text="" + str(row[1]), font=("Times New", 18)).grid(row=2, column=row\_number + 1)

                        Label(root, text="" + str(row[2]), font=("Times New", 18)).grid(row=3, column=row\_number + 1)

                        Label(root, text="" + str(row[3]), font=("Times New", 18)).grid(row=4, column=row\_number + 1)

                        Label(root, text="" + str(row[4]), font=("Times New", 18)).grid(row=5, column=row\_number + 1)

                        Label(root, text="" + str(row[5]), font=("Times New", 18)).grid(row=6, column=row\_number + 1)

                    root.mainloop()

                Button(root, text='Show Student Data', command=show1, width=15, font=("Times New", 10), bd=5,

                       bg="light blue").grid(row=2, column=2, sticky=N + S + E + W)

                ##########################################################################################

                def show2():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('View Window')

                    v = [(f.get())]

                    z = cur.execute("select \*from student where clno=?", v)

                    row\_number = 0

                    Label(root, text="Student ID").grid(column=1, row=row\_number)

                    Label(root, text="Name").grid(column=2, row=row\_number)

                    Label(root, text="Father Name").grid(column=3, row=row\_number)

                    Label(root, text="Mother Name").grid(column=4, row=row\_number)

                    Label(root, text="Class").grid(column=5, row=row\_number)

                    Label(root, text="Mobile no.").grid(column=6, row=row\_number)

                    y = -1

                    for row\_number, row in enumerate(z):

                        y = y + 1

                        Label(root, text="" + str(row[0])).grid(column=1, row=row\_number + 1+y)

                        Label(root, text="" + str(row[1])).grid(column=2, row=row\_number + 1+y)

                        Label(root, text="" + str(row[2])).grid(column=3, row=row\_number + 1+y)

                        Label(root, text="" + str(row[3])).grid(column=4, row=row\_number + 1+y)

                        Label(root, text="" + str(row[4])).grid(column=5, row=row\_number + 1+y)

                        Label(root, text="" + str(row[5])).grid(column=6, row=row\_number + 1+y)

                    root.mainloop()

                Button(root, text='Show All Student Data Of Class', command=show2, width=25, font=("Times New", 10),

                       bd=5, bg="light blue").grid(row=4, column=2, sticky=N + S + E + W)

                ###########################################################################

                def logout():

                    ans = askyesno('Confirmation', 'Do You Want To Exit?')

                    if (ans == True):

                        root.destroy()

                Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                    row=8, column=0, sticky=N + S + E + W)

                root.mainloop()

            Button(menu, text='View Student details', command=option1, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=1, column=0, sticky=N + S + E + W)

            #############################################################

            def option2():

                    win = Toplevel()

                    win.geometry('1080x600')

                    win.title('Update Attendance')

                    l = Label(win, text='Enter Student Id : ')

                    l.grid(row=2, column=0)

                    f = Entry(win, font=("Times New", 10), bd=5, bg="light blue")  # f=sid.

                    f.grid(row=2, column=1)

                    def fill():

                        Label(win, text="English").grid(column=1, row=3, sticky=N + S + E + W)

                        eng = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                        eng.grid(row=4, column=1, sticky=N + S + E + W)

                        Label(win, text="Hindi").grid(column=2, row=3, sticky=N + S + E + W)

                        hin = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                        hin.grid(row=4, column=2, sticky=N + S + E + W)

                        Label(win, text="Mathematics").grid(column=3, row=3, sticky=N + S + E + W)

                        mat = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                        mat.grid(row=4, column=3, sticky=N + S + E + W)

                        Label(win, text="Science").grid(column=4, row=3)

                        sc = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                        sc.grid(row=4, column=4, sticky=N + S + E + W)

                        Label(win, text="Social Science").grid(column=5, row=3, sticky=N + S + E + W)

                        ssc = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                        ssc.grid(row=4, column=5, sticky=N + S + E + W)

                        Label(win, text="Sanskrit").grid(column=6, row=3, sticky=N + S + E + W)

                        san = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                        san.grid(row=4, column=6, sticky=N + S + E + W)

                        def regist():

                            cur.execute("update student set eng\_at=? where sid=?", (eng.get(), f.get()))

                            con.commit()

                            cur.execute("update student set hin\_at=? where sid=?", (hin.get(), f.get()))

                            con.commit()

                            cur.execute("update student set mat\_at=? where sid=?", (mat.get(), f.get()))

                            con.commit()

                            cur.execute("update student set sc\_at=? where sid=?", (sc.get(), f.get()))

                            con.commit()

                            cur.execute("update student set ssc\_at=? where sid=?", (ssc.get(), f.get()))

                            con.commit()

                            cur.execute("update student set san\_at=? where sid=?", (san.get(), f.get()))

                            con.commit()

                            showinfo('Information', 'Attendance Locked')

                        #######################################################################################################

                        Button(win, text='Submit', command=regist, width=25, font=("Times New", 10), bd=5,

                               bg="light blue").grid(row=5, column=0)

                    Button(win, text='Show', command=fill, width=15, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=2, column=2)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            win.destroy()

                    Button(win, text='Exit', command=logout, width=15, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=2, column=3)

                    win.mainloop()

            Button(menu, text='Update Attendance', command=option2, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=2, column=0, sticky=N + S + E + W)

            #########################################################################################################################################

            def option3():

                win = Toplevel()

                win.geometry('1080x600')

                win.title('Update Student Marks')

                l = Label(win, text='Enter Student Id : ')

                l.grid(row=2, column=0)

                f = Entry(win, font=("Times New", 10), bd=5, bg="light blue")  # f=sid.

                f.grid(row=2, column=1)

                def fill():

                    Label(win, text="English").grid(column=1, row=3, sticky=N + S + E + W)

                    eng = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                    eng.grid(row=4, column=1, sticky=N + S + E + W)

                    Label(win, text="Hindi").grid(column=2, row=3, sticky=N + S + E + W)

                    hin = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                    hin.grid(row=4, column=2, sticky=N + S + E + W)

                    Label(win, text="Mathematics").grid(column=3, row=3, sticky=N + S + E + W)

                    mat = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                    mat.grid(row=4, column=3, sticky=N + S + E + W)

                    Label(win, text="Science").grid(column=4, row=3)

                    sc = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                    sc.grid(row=4, column=4, sticky=N + S + E + W)

                    Label(win, text="Social Science").grid(column=5, row=3, sticky=N + S + E + W)

                    ssc = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                    ssc.grid(row=4, column=5, sticky=N + S + E + W)

                    Label(win, text="Sanskrit").grid(column=6, row=3, sticky=N + S + E + W)

                    san = Entry(win, width=15, font=("Times New", 10), bd=5, bg="light blue")

                    san.grid(row=4, column=6, sticky=N + S + E + W)

                    def regist():

                        cur.execute("update student set eng\_m=? where sid=?", (eng.get(), f.get()))

                        con.commit()

                        cur.execute("update student set hin\_m=? where sid=?", (hin.get(), f.get()))

                        con.commit()

                        cur.execute("update student set mat\_m=? where sid=?", (mat.get(), f.get()))

                        con.commit()

                        cur.execute("update student set sc\_m=? where sid=?", (sc.get(), f.get()))

                        con.commit()

                        cur.execute("update student set ssc\_m=? where sid=?", (ssc.get(), f.get()))

                        con.commit()

                        cur.execute("update student set san\_m=? where sid=?", (san.get(), f.get()))

                        con.commit()

                        showinfo('Information', 'Marks Locked')

                    ######################################################################################################

                    Button(win, text='Submit', command=regist, width=15, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=5, column=0)

                Button(win, text='Show', command=fill, width=15, font=("Times New", 10), bd=5,

                       bg="light blue").grid(row=2, column=2)

                def logout():

                    ans = askyesno('Confirmation', 'Do You Want To Exit?')

                    if (ans == True):

                        win.destroy()

                Button(win, text='Exit', command=logout, width=15, font=("Times New", 10), bd=5,

                       bg="light blue").grid(row=2, column=3)

                win.mainloop()

            Button(menu, text='Update Student Marks', command=option3, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=3, column=0, sticky=N + S + E + W)

            ####################################################################

            def option5():

                root = Toplevel()

                root.geometry('1080x600')

                root.title('Change Password')

                l = Label(root, text='Change Password :', font='Times 20 bold')

                l.grid(row=0, column=3)

                l = Label(root, text=' ')

                l.grid(row=1, column=0)

                l = Label(root, text='Enter New Password : ')

                l.grid(row=3, column=0)

                h = Entry(root, width=25, font=("Times New", 10), bd=5, bg="light blue")

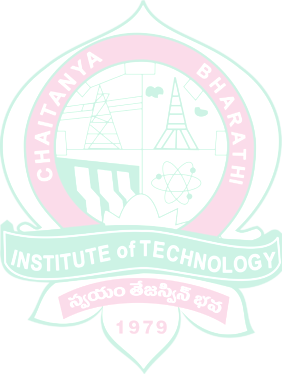
                h.grid(row=3, column=1)

                def update1():

                    cur1.execute("update teacher set pswd=? where tid=?", (int(h.get()), id[0]))

                    con1.commit()

                    showinfo('Information', 'Password Changed Successfully')

                Button(root, text='save & change', command=update1, width=25, font=("Times New", 10), bd=5,

                       bg="light blue").grid(row=4, column=0, sticky=N + S + E + W)

                def logout():

                    ans = askyesno('Confirmation', 'Do You Want To Exit?')

                    if (ans == True):

                        root.destroy()

                Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5,

                       bg="light blue").grid(row=8, column=0)

                root.mainloop()

            Button(menu, text='Change Password', command=option5, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=6, column=0, sticky=N + S + E + W)

            ######################################################################################################################################################################

            def logout():

                ans = askyesno('Confirmation', 'Do You Want To Logout?')

                if (ans == True):

                    menu.destroy()

            Button(menu, text='Logout', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                row=8, column=0, sticky=N + S + E + W)

            menu.mainloop()

            ####################################

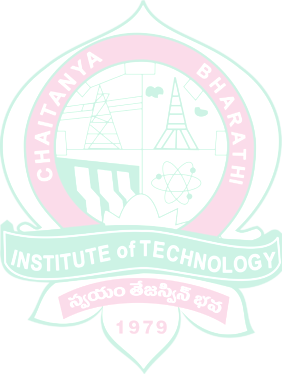
    Button(window, text='Login', command=login, width=20, font=("Times New", 15), bd=5, bg="light blue").grid(

        row=3,

        column=2,

        sticky=N + S + E + W)

**parent.py**

The parents can login to the system by giving their ward’s ID and mobile number. As shown, the parent can view fee status, attendance and marks of their ward. The student’s attendance and marks/results can be viewed with respect to their subjects.

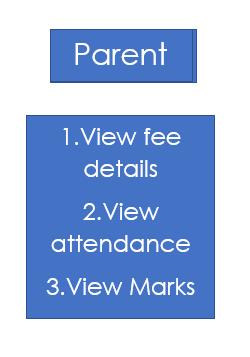


Fig 6.4: Parent Features

**Code of parent.py**

from tkinter.messagebox import \*

import sqlite3

con = sqlite3.Connection('student\_records.db')

cur = con.cursor()

from tkinter import \*

def start():

    window = Toplevel()

    window.geometry('1080x600')

    window.title('Login Window')

    Label(window, text='ID  ', font=("Times New", 20)).grid(row=1, column=1)  # v=id

    v = Entry(window, width=25, font=("Times New", 18), bd=5, bg="light blue")

    v.grid(row=1, column=2)

    Label(window, text='Mobile no.  ', font=("Times n=New", 20)).grid(row=2, column=1)

    vv = Entry(window, show='\*', width=25, font=("Times New", 18), bd=5, bg="light blue")  # vv=password

    vv.grid(row=2, column=2)

    def login():

        id=[int(v.get())]

        z = cur.execute("select mbno from student where sid=?", id)

        x=z.fetchone()

        print(x)

        if ((int(vv.get()) != x[0])):

            showwarning('Login Failed', 'Incorrect Id Or Password Used Or ')

        else:

            menu = Toplevel()

            menu.geometry('1080x600')

            menu.title('Menu Window')

            l = Label(menu, text='Dashboard :', font='Times 20 bold')

            l.grid(row=0, column=0)

            Label(menu).grid(row=1, column=100, rowspan=30)

            ################################################################################

            def option5():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('View Window')

                    Label(root, text="Attendance", font=("Times New", 20)).grid(row=0, column=4)

                    row\_number = 0

                    Label(root, text="English    :", font=("Times New", 20)).grid(row=1, column=row\_number)

                    Label(root, text="Hindi      :", font=("Times New", 20)).grid(row=2, column=row\_number)

                    Label(root, text="Mathematics:", font=("Times New", 20)).grid(row=3, column=row\_number)

                    Label(root, text="Science    :", font=("Times New", 20)).grid(row=4, column=row\_number)

                    Label(root, text="Social     :", font=("Times New", 20)).grid(row=5, column=row\_number)

                    Label(root, text="Sanskrit   :", font=("Times New", 20)).grid(row=6, column=row\_number)

                    z = cur.execute("select eng\_at,hin\_at,mat\_at,sc\_at,ssc\_at,san\_at from student where sid=?", id)

                    for row\_number, row in enumerate(z):

                        Label(root, text="" + str(row[0]), font=("Times New", 18)).grid(row=1, column=row\_number + 1)

                        Label(root, text="" + str(row[1]), font=("Times New", 18)).grid(row=2, column=row\_number + 1)

                        Label(root, text="" + str(row[2]), font=("Times New", 18)).grid(row=3, column=row\_number + 1)

                        Label(root, text="" + str(row[3]), font=("Times New", 18)).grid(row=4, column=row\_number + 1)

                        Label(root, text="" + str(row[4]), font=("Times New", 18)).grid(row=5, column=row\_number + 1)

                        Label(root, text="" + str(row[5]), font=("Times New", 18)).grid(row=6, column=row\_number + 1)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                        row=8, column=0)

                    root.mainloop()

            Button(menu, text='View Attendance', command=option5, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=5, column=0, sticky=N + S + E + W)

 #########################################################################################################################################

            def option7():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('View Window')

                    Label(root, text="Marks", font=("Times New", 20)).grid(row=0, column=4)

                    row\_number = 0

                    Label(root, text="English    :", font=("Times New", 20)).grid(row=1, column=row\_number)

                    Label(root, text="Hindi      :", font=("Times New", 20)).grid(row=2, column=row\_number)

                    Label(root, text="Mathematics:", font=("Times New", 20)).grid(row=3, column=row\_number)

                    Label(root, text="Science    :", font=("Times New", 20)).grid(row=4, column=row\_number)

                    Label(root, text="Social     :", font=("Times New", 20)).grid(row=5, column=row\_number)

                    Label(root, text="Sanskrit   :", font=("Times New", 20)).grid(row=6, column=row\_number)

                    z = cur.execute("select eng\_m,hin\_m,mat\_m,sc\_m,ssc\_m,san\_m from student where sid=?", id)

                    for row\_number, row in enumerate(z):

                        Label(root, text="" + str(row[0]), font=("Times New", 18)).grid(row=1, column=row\_number + 1)

                        Label(root, text="" + str(row[1]), font=("Times New", 18)).grid(row=2, column=row\_number + 1)

                        Label(root, text="" + str(row[2]), font=("Times New", 18)).grid(row=3, column=row\_number + 1)

                        Label(root, text="" + str(row[3]), font=("Times New", 18)).grid(row=4, column=row\_number + 1)

                        Label(root, text="" + str(row[4]), font=("Times New", 18)).grid(row=5, column=row\_number + 1)

                        Label(root, text="" + str(row[5]), font=("Times New", 18)).grid(row=6, column=row\_number + 1)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                        row=8, column=0)

                    root.mainloop()

            Button(menu, text='View Marks', command=option7, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=7, column=0, sticky=N + S + E + W)

            def logout():

                ans = askyesno('Confirmation', 'Do You Want To Logout?')

                if (ans == True):

                    menu.destroy()

            Button(menu, text='Logout', command=logout, width=25, font=("Times New", 10), bd=5, bg="light blue").grid(

                row=8, column=0, sticky=N + S + E + W)

            ################################################################################

            def option3():

                    root = Toplevel()

                    root.geometry('1080x600')

                    root.title('View Fee')

                    l = Label(root, text='FEE:', font='Times 20 bold')

                    l.grid(row=0, column=3)

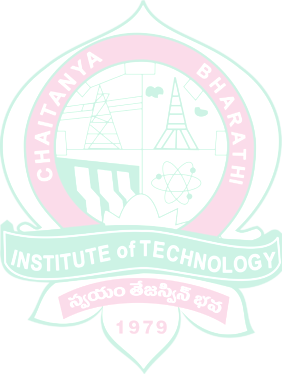
                    l = Label(root, text=' ')

                    l.grid(row=5, column=0)

                    cur.execute("select yearduefee from student where sid=?", id)

                    x = cur.fetchall()

                    Label(root, text=' TOTAL FEE : ' + str(x[0][0]), font='Times 16 bold').grid(column=1, row=1)

                    cur.execute("select duefee from student where sid=?", id)

                    y = cur.fetchall()

                    Label(root, text=' DUE FEE : ' + str(y[0][0]), font='Times 16 bold').grid(column=1, row=2)

                    cur.execute("select paidfee from student where sid=?", id)

                    z = cur.fetchall()

                    Label(root, text=' PAID FEE : ' + str(z[0][0]), font='Times 16 bold').grid(column=1, row=3)

                    def logout():

                        ans = askyesno('Confirmation', 'Do You Want To Exit?')

                        if (ans == True):

                            root.destroy()

                    Button(root, text='Exit', command=logout, width=25, font=("Times New", 10), bd=5,

                           bg="light blue").grid(row=8, column=0)

                    root.mainloop()

            Button(menu, text='View Fee', command=option3, width=25, font=("Times New", 10), bd=5,

                   bg="light blue").grid(row=3, column=0, sticky=N + S + E + W)

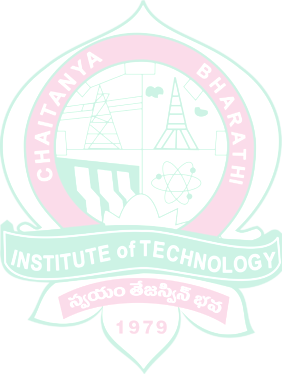
            menu.mainloop()

    Button(window, text='Login', command=login, width=20, font=("Times New", 15), bd=5, bg="light blue").grid(

        row=3,

        column=2,

        sticky=N + S + E + W)

**RESULTS**

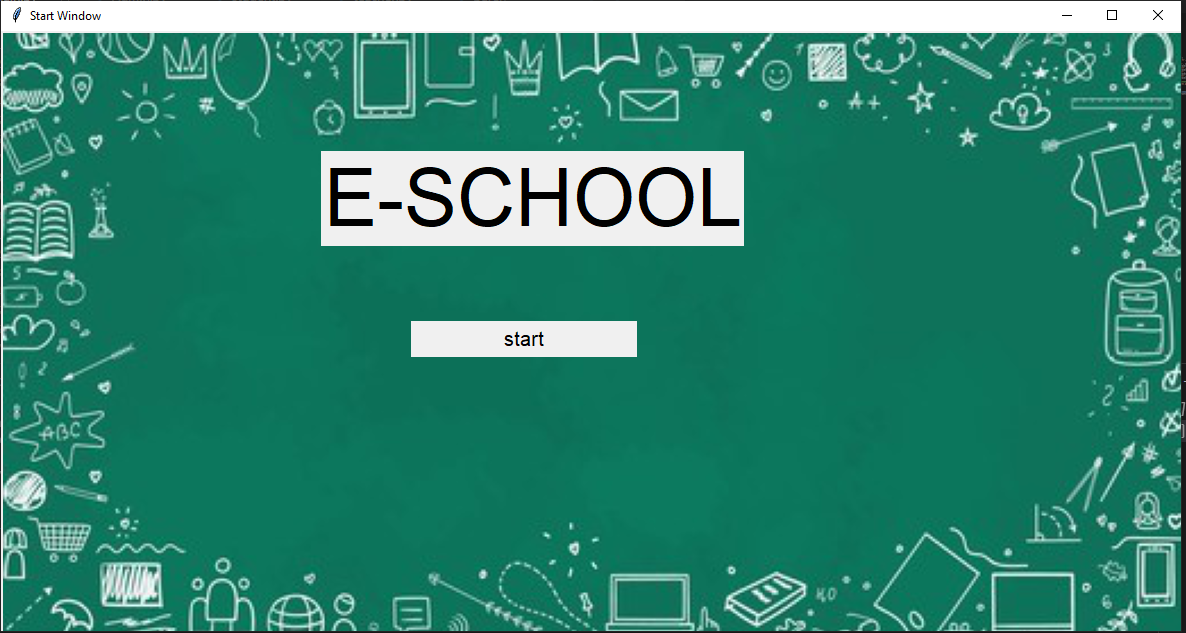
****

Fig 7.1: Start window

**LOGIN:** It is a multi-user login Interface where you can login as Admin , Student , Parent , Teacher . User can login by giving the username and password.

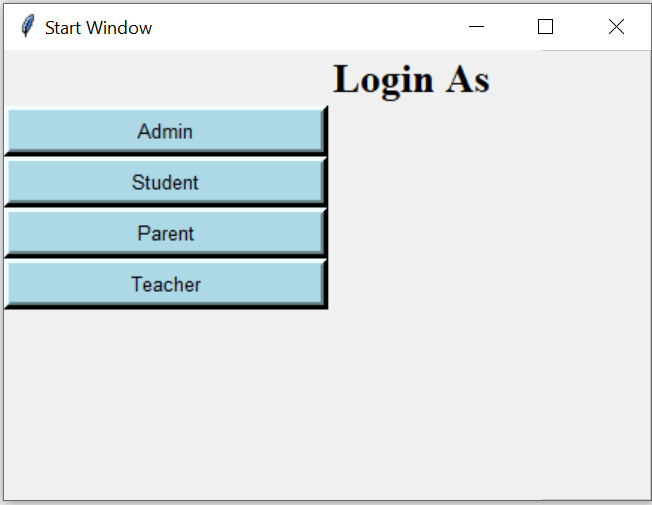


Fig 7.2: Log in As window

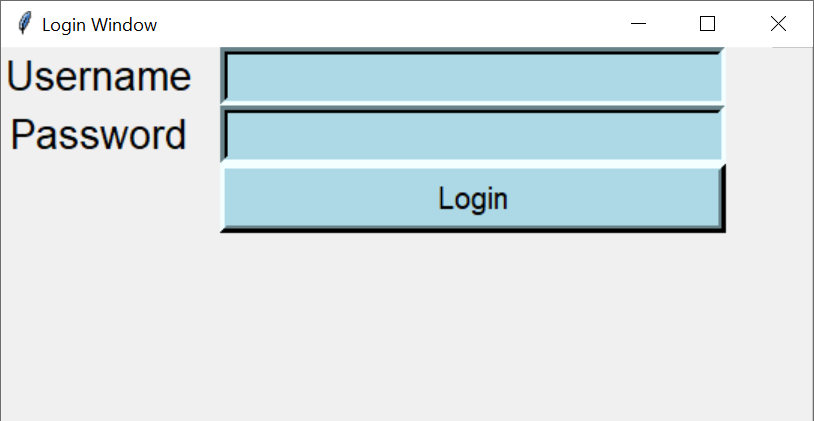
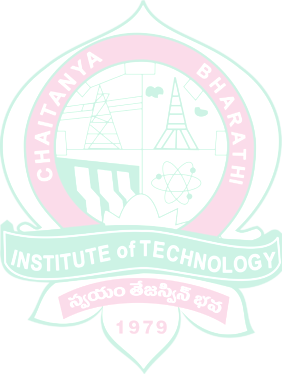


Fig 7.3: Login window

**ADMIN MENU WINDOW**:

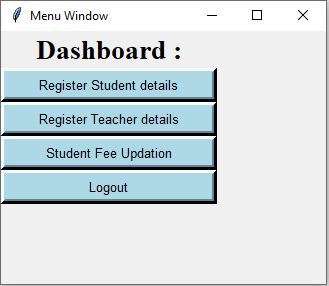


Fig 7.4: Dashboard of Admin

**ADMIN: Register Student Details**

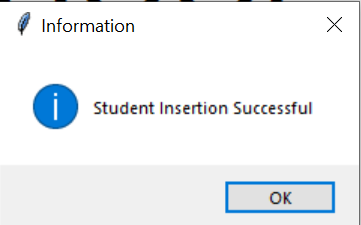
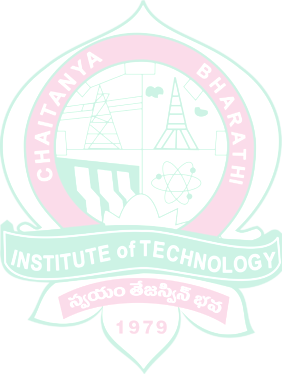
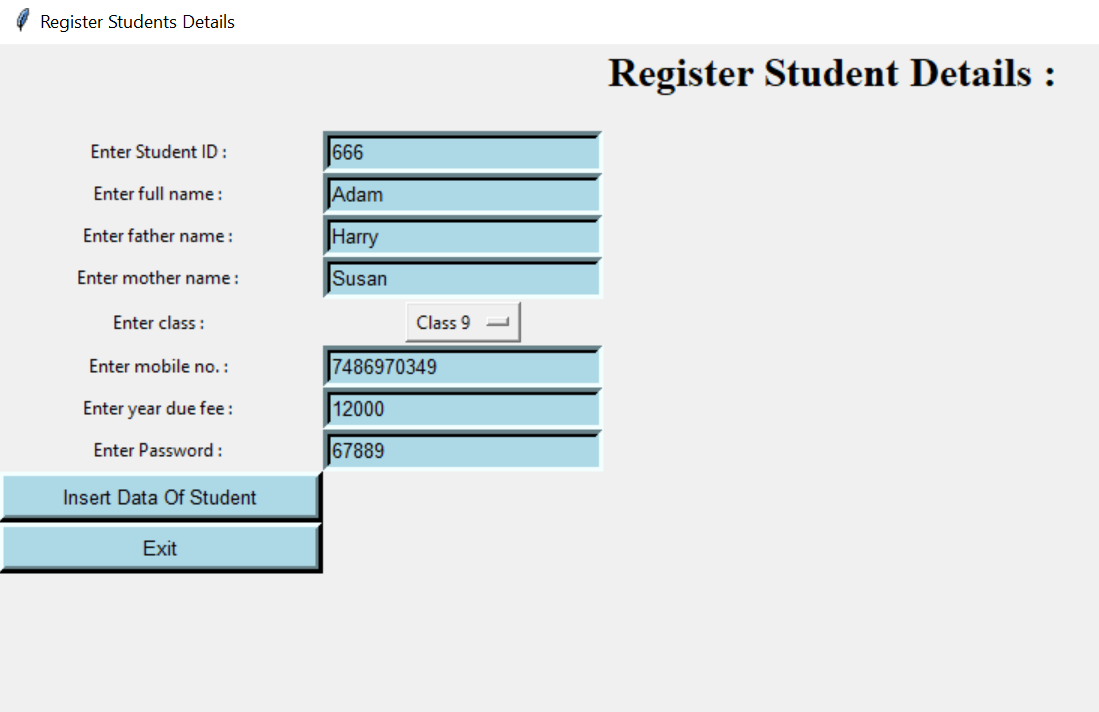


Fig 7.5: Admin registering student details

Fig 7.6: Message of successful insertion

**ADMIN: Register Teacher Details**

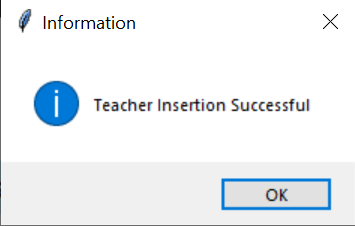
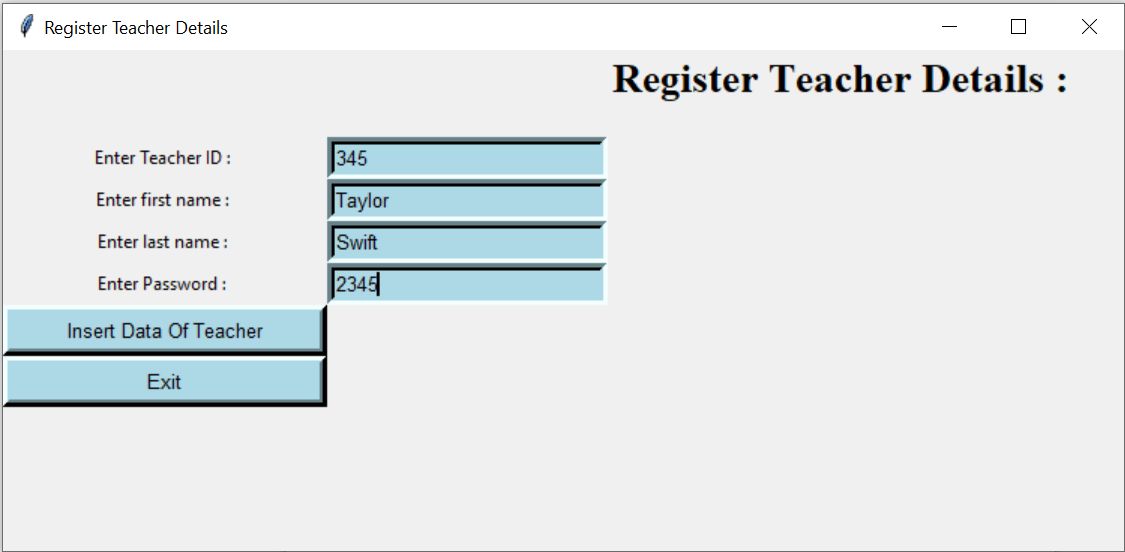
****

Fig 7.7: Admin registering teacher details

Fig 7.8: Message of successful insertion

**ADMIN: Student fee Update**

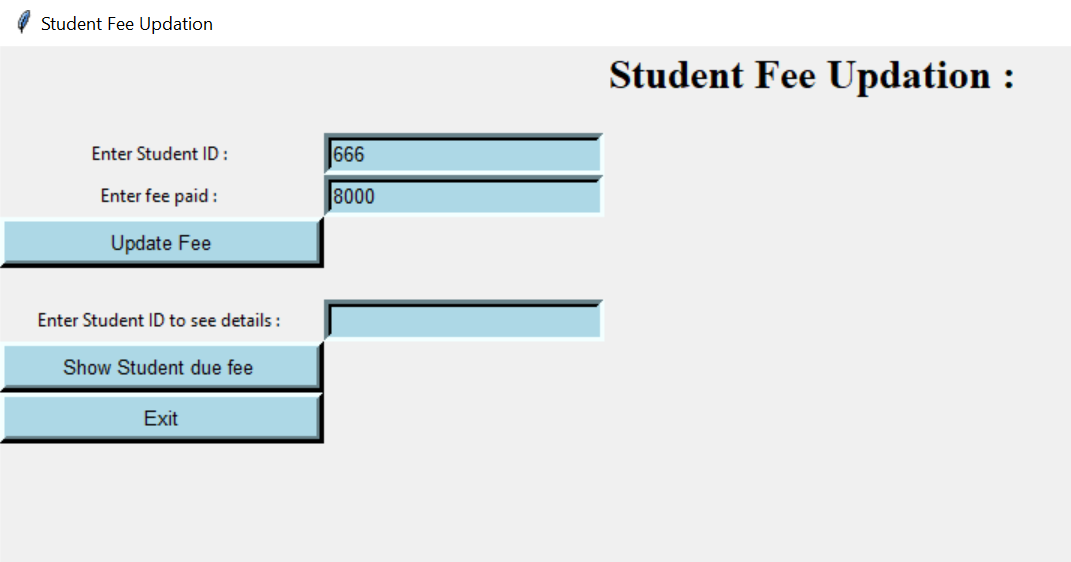
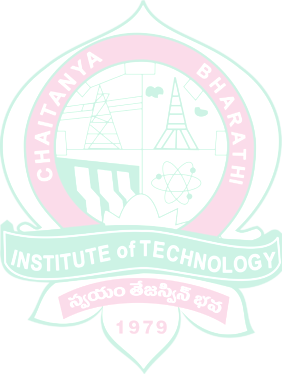
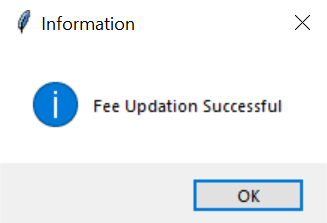
********

Fig 7.9: Admin updating student fee

Fig 7.10: Message of successful fee update

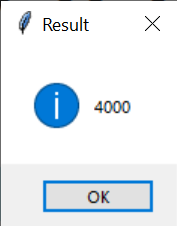
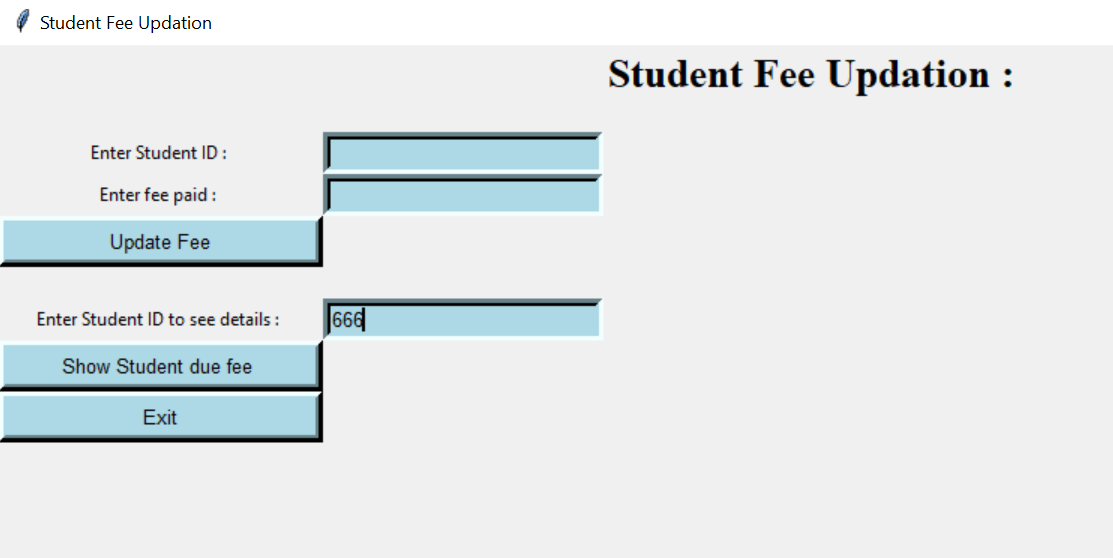


Fig 7.11: Viewing student fee

Fig 7.12: Displaying fee

**STUDENT MENU WINDOW:**

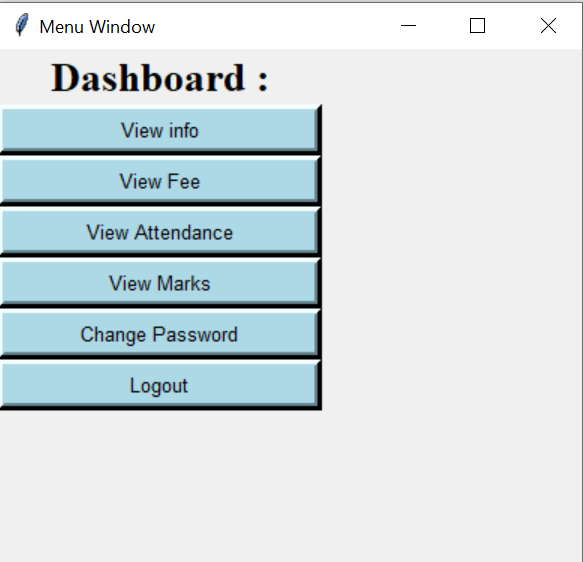
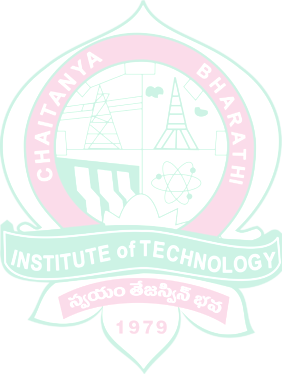
****

Fig 7.13: Student Dashboard

**STUDENT: View info**

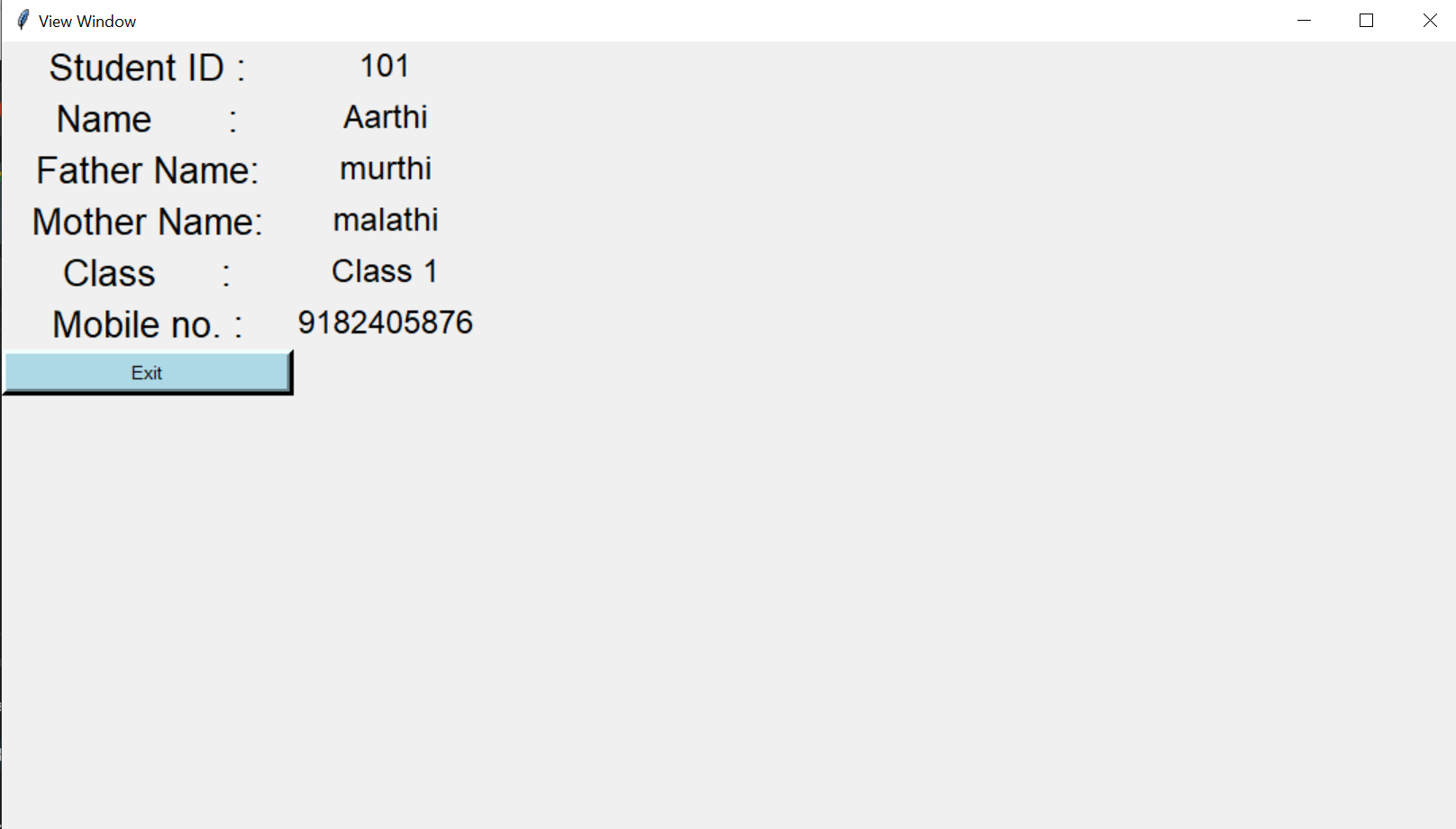


Fig 7.14: Student viewing personal info

**STUDENT: View Fee Details**



Fig 7.15: Viewing fee details

**STUDENT: View Attendance**

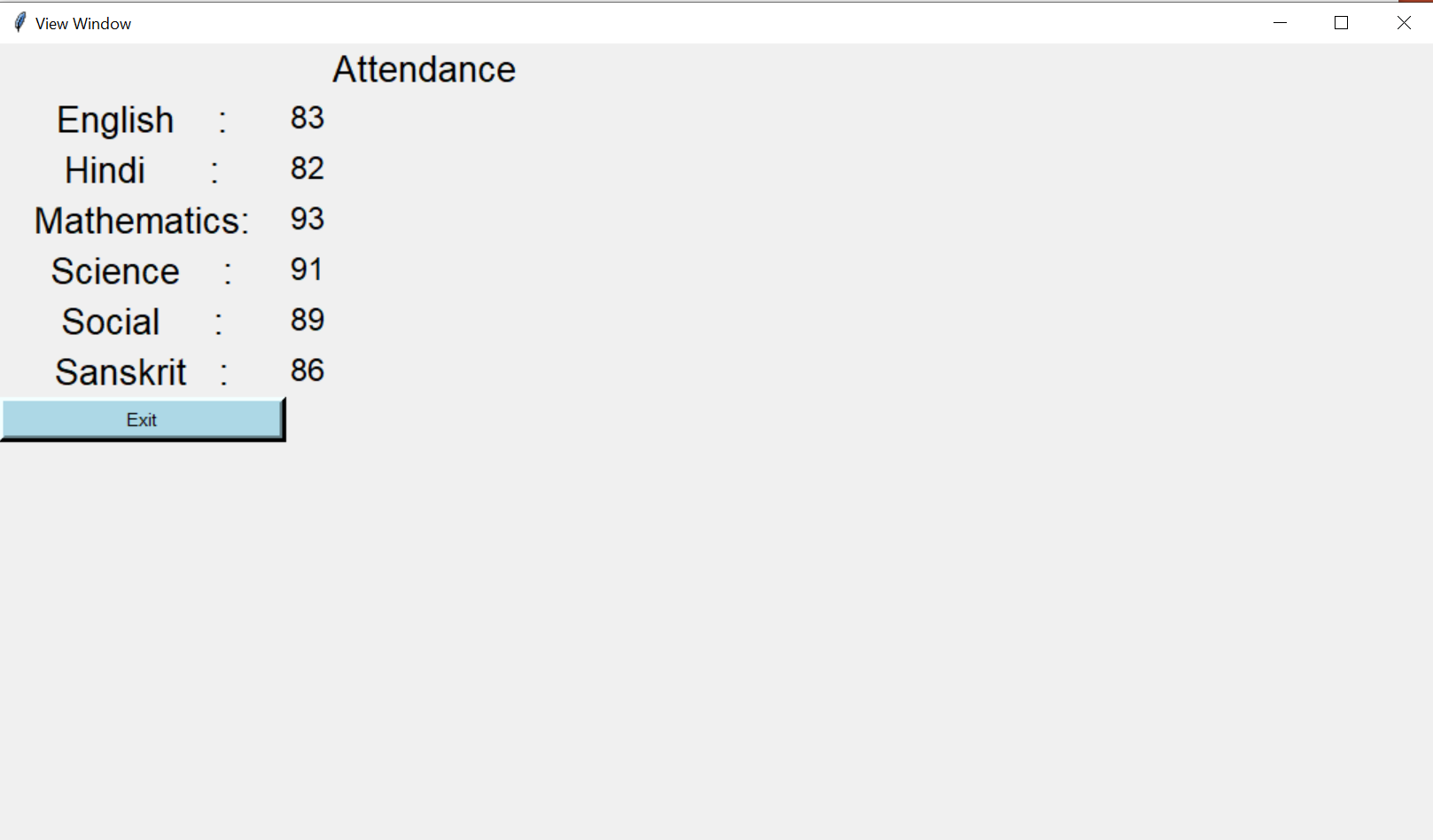


Fig 7.16: Student viewing attendance

**STUDENT: View Marks**

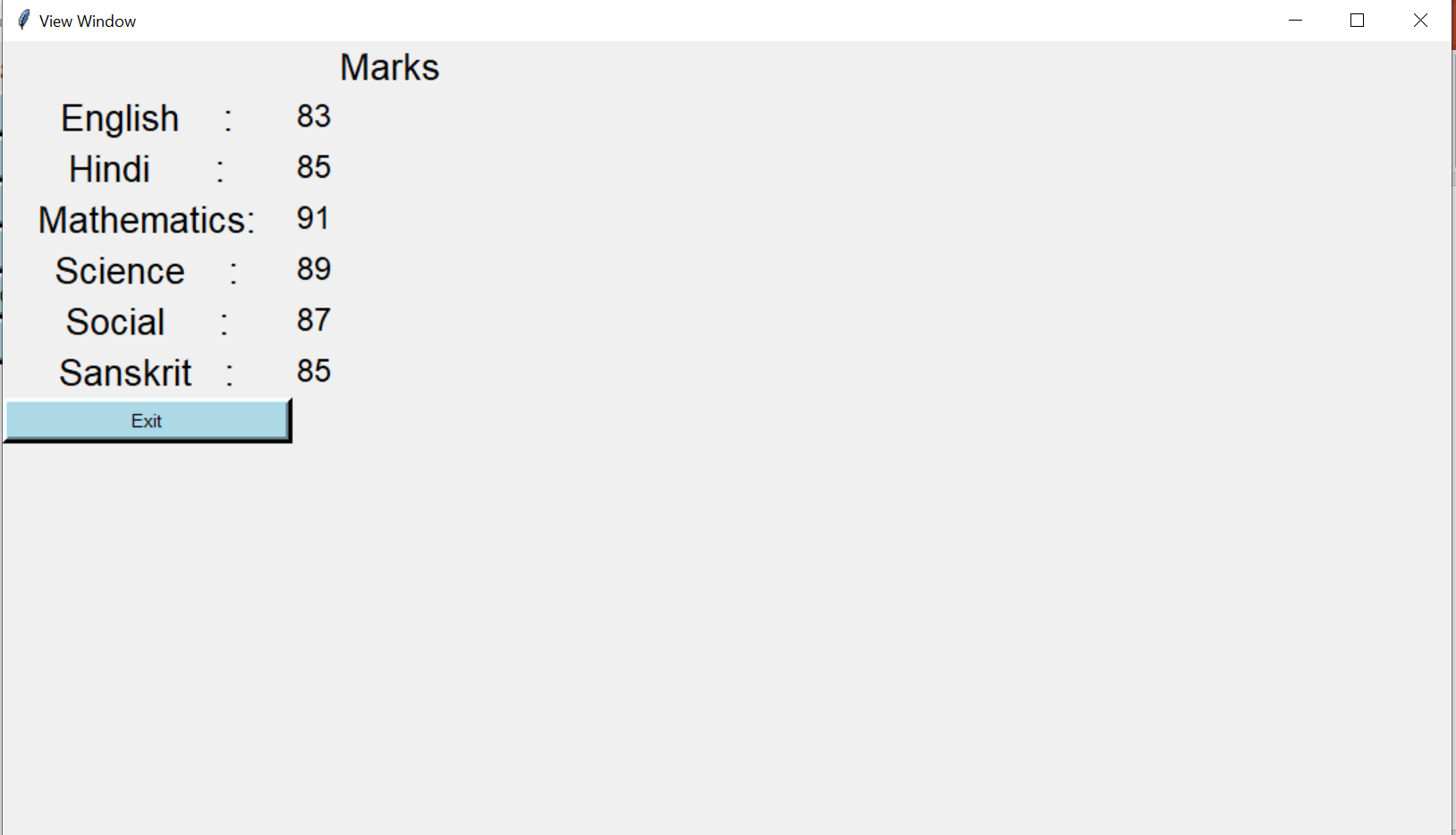
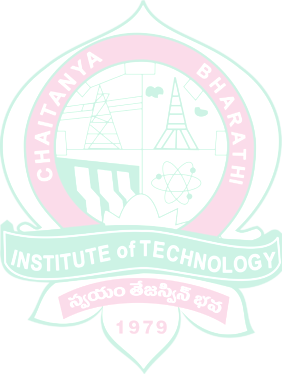


Fig 7.17: Student viewing marks

**STUDENT: Change Password**

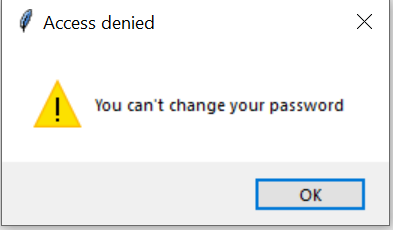


Fig 7.18: Access denying message

For Class 1 to 10:

For class 11 and 12:

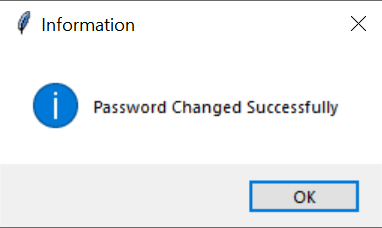
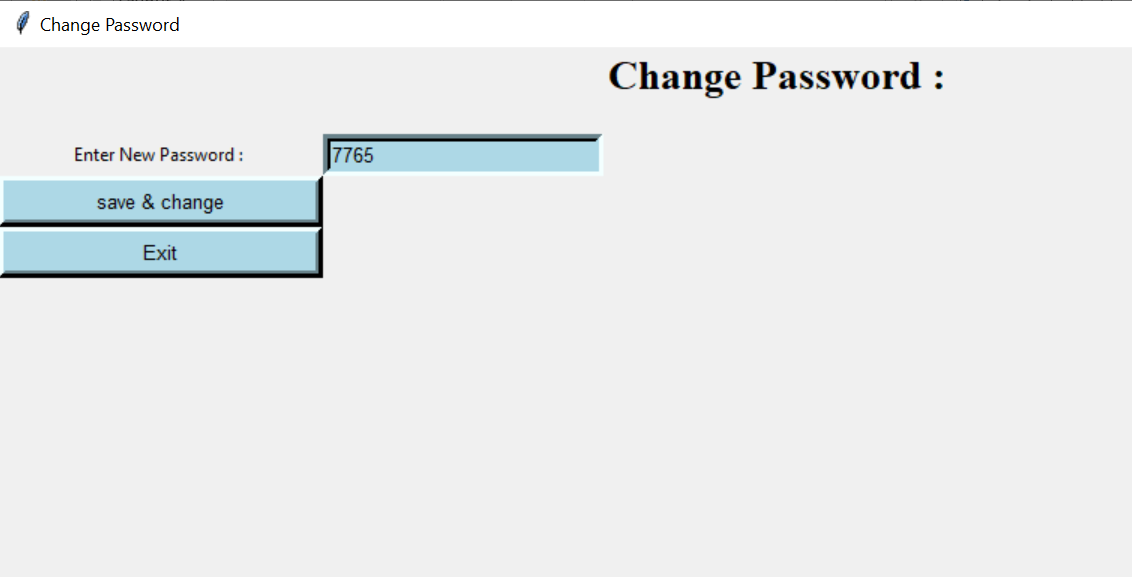


Fig 7.19: Changing password

Fig 7.20: Message of successful password change

**PARENT MENU WINDOW:**

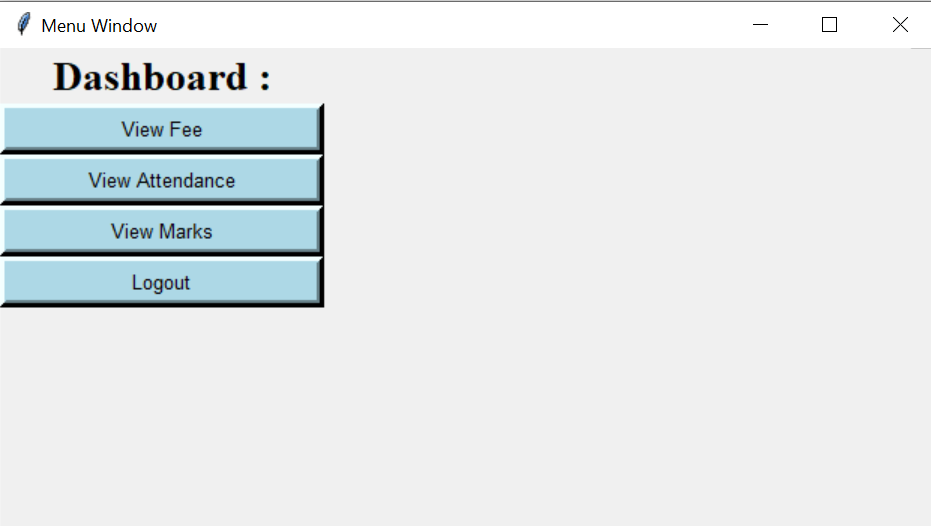
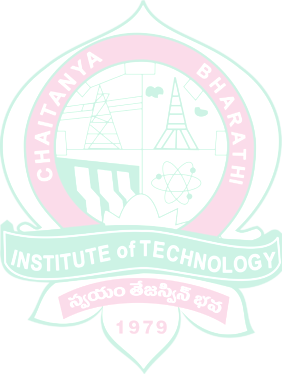


Fig 7.21: Parent dashboard window

**PARENT: View Fee Status**



Fig 7.22: Viewing fee details

**PARENT: View Student Attendance**

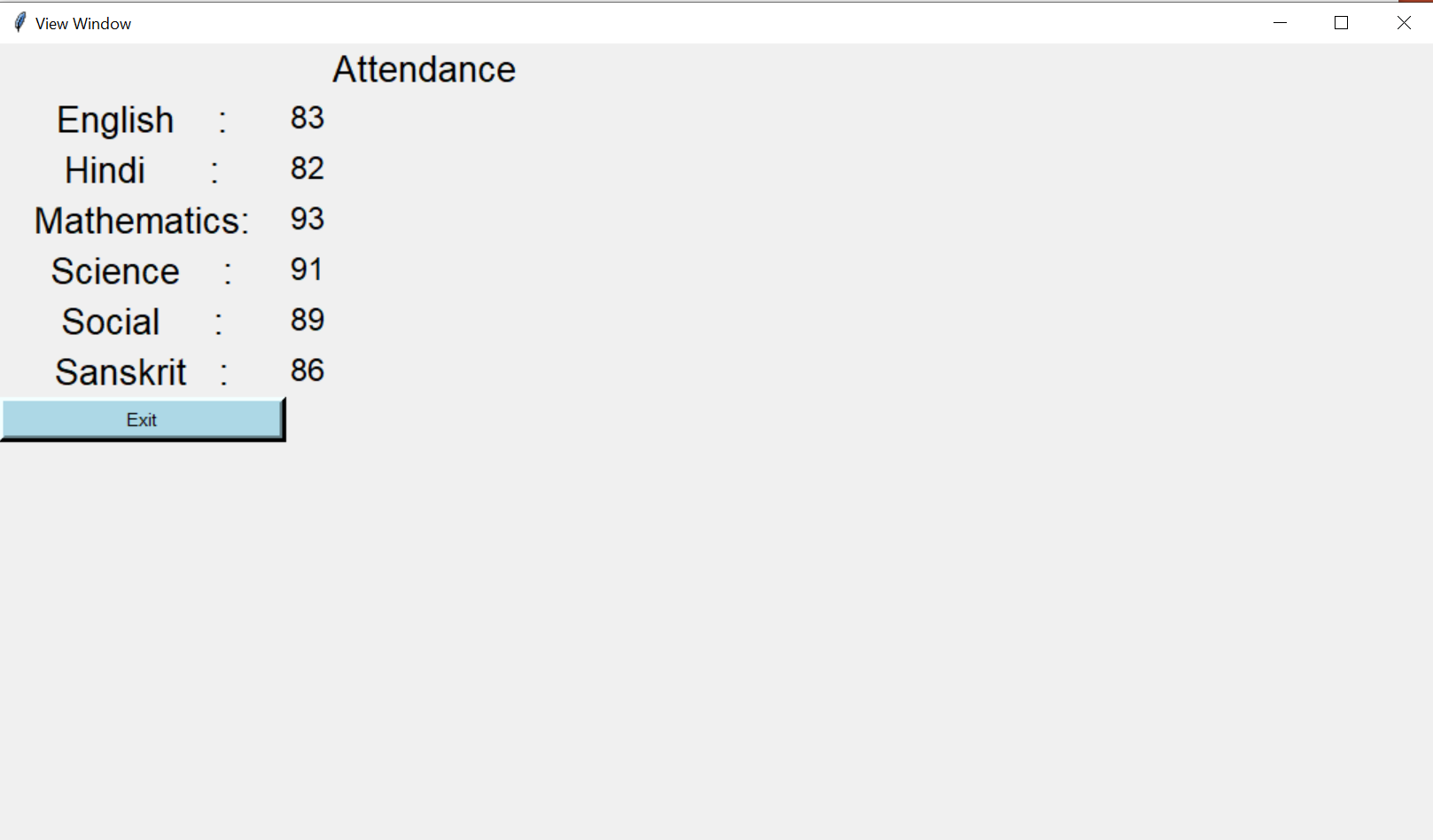
****

Fig 7.23: Parent viewing attendance

**PARENT: View Student Marks**

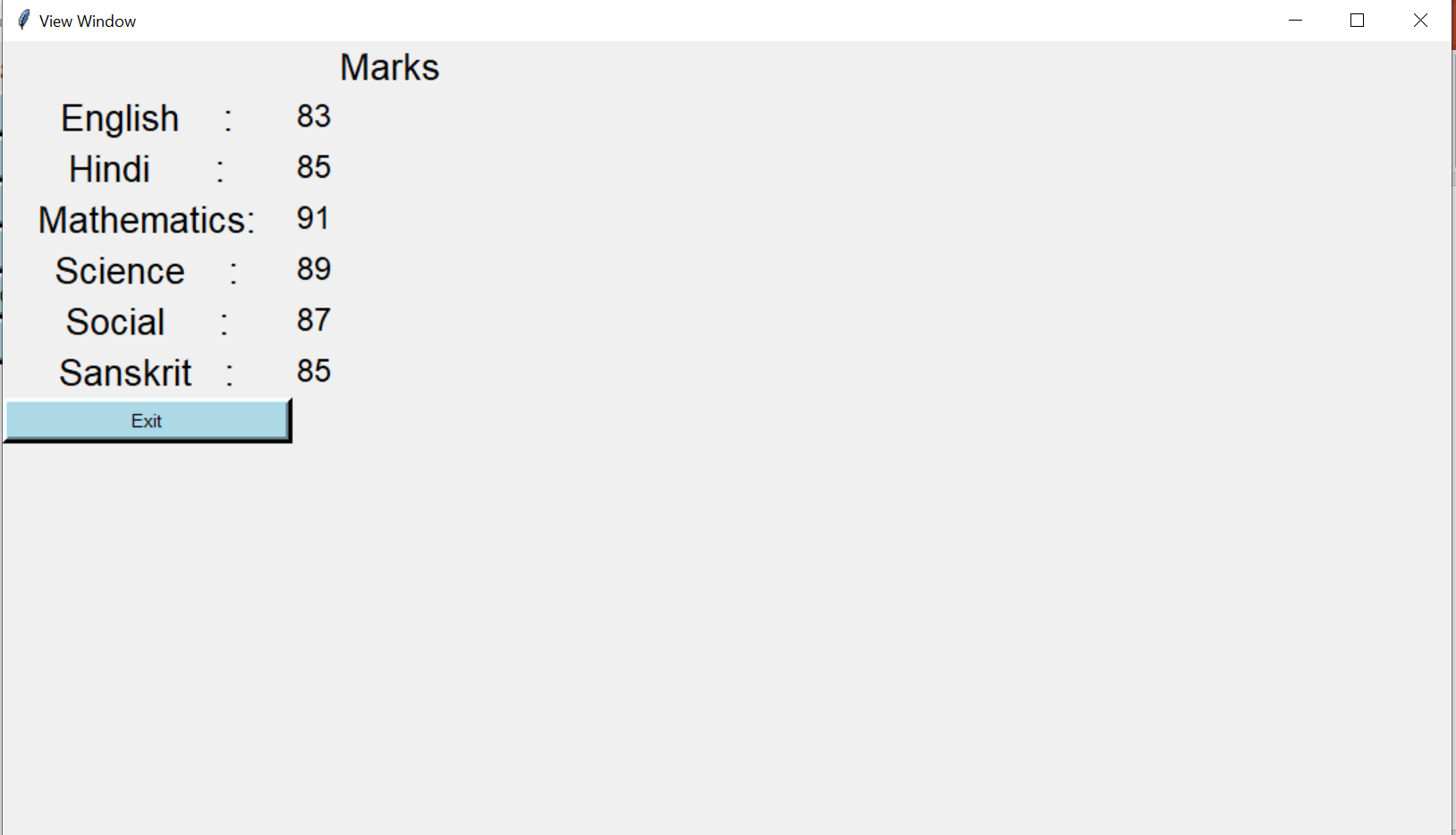
****

Fig 7.24: Student viewing marks

**TEACHER MENU WINDOW:**

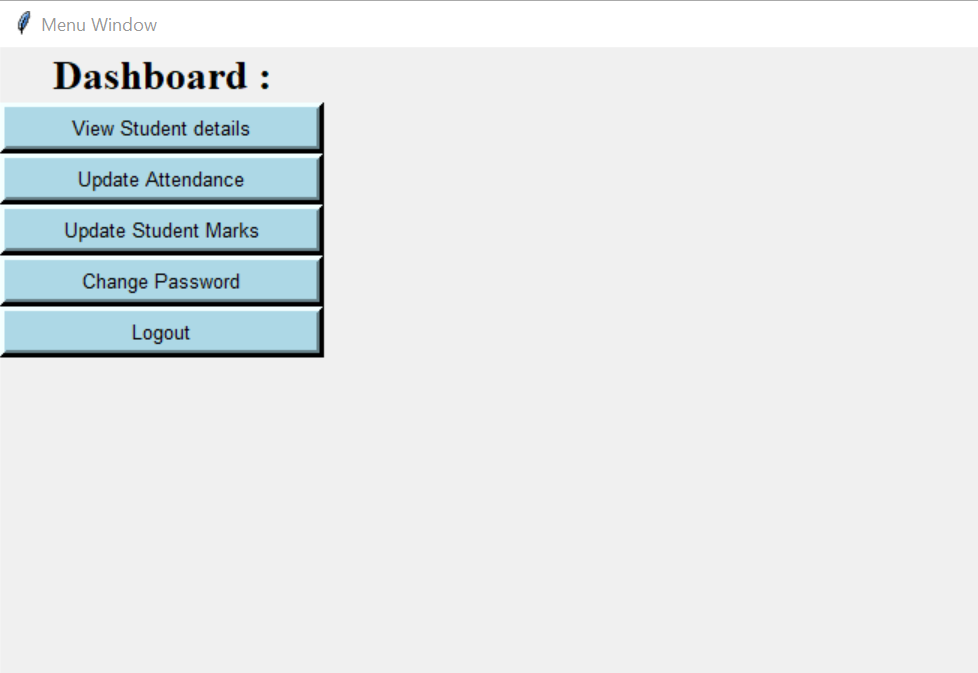
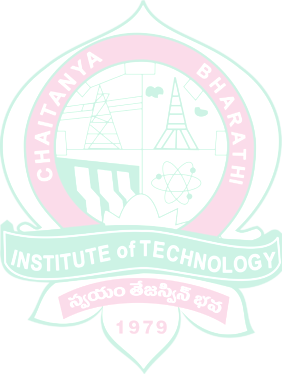
****

Fig 7.25: Teacher Dashboard window

**TEACHER: View Student Details**

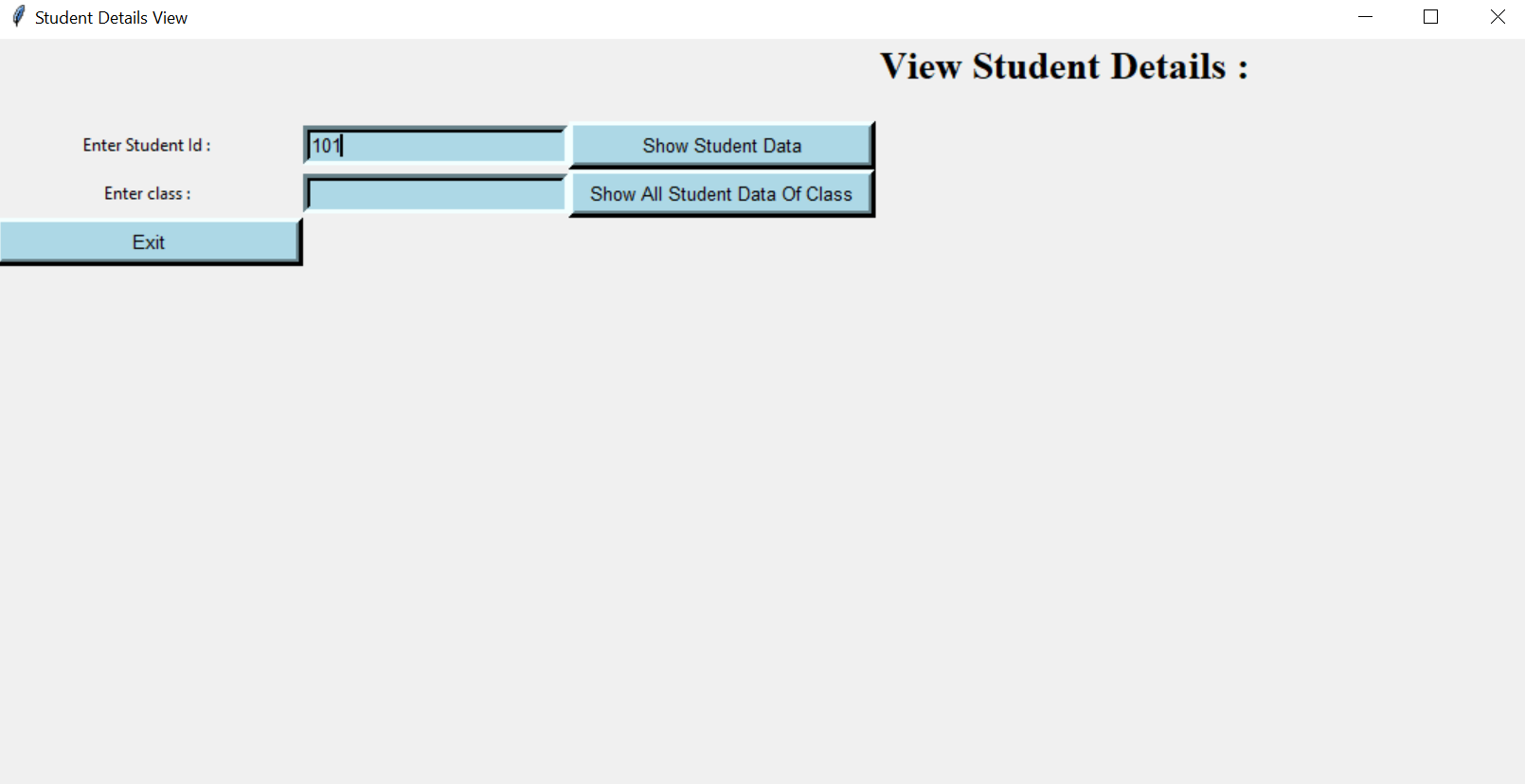
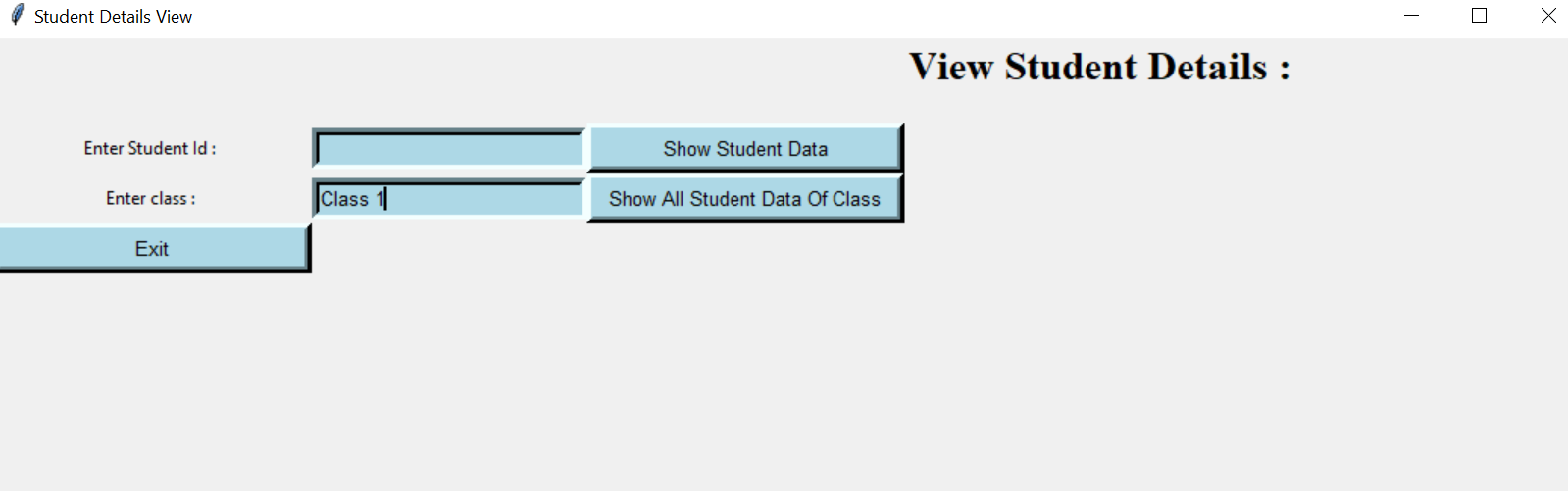
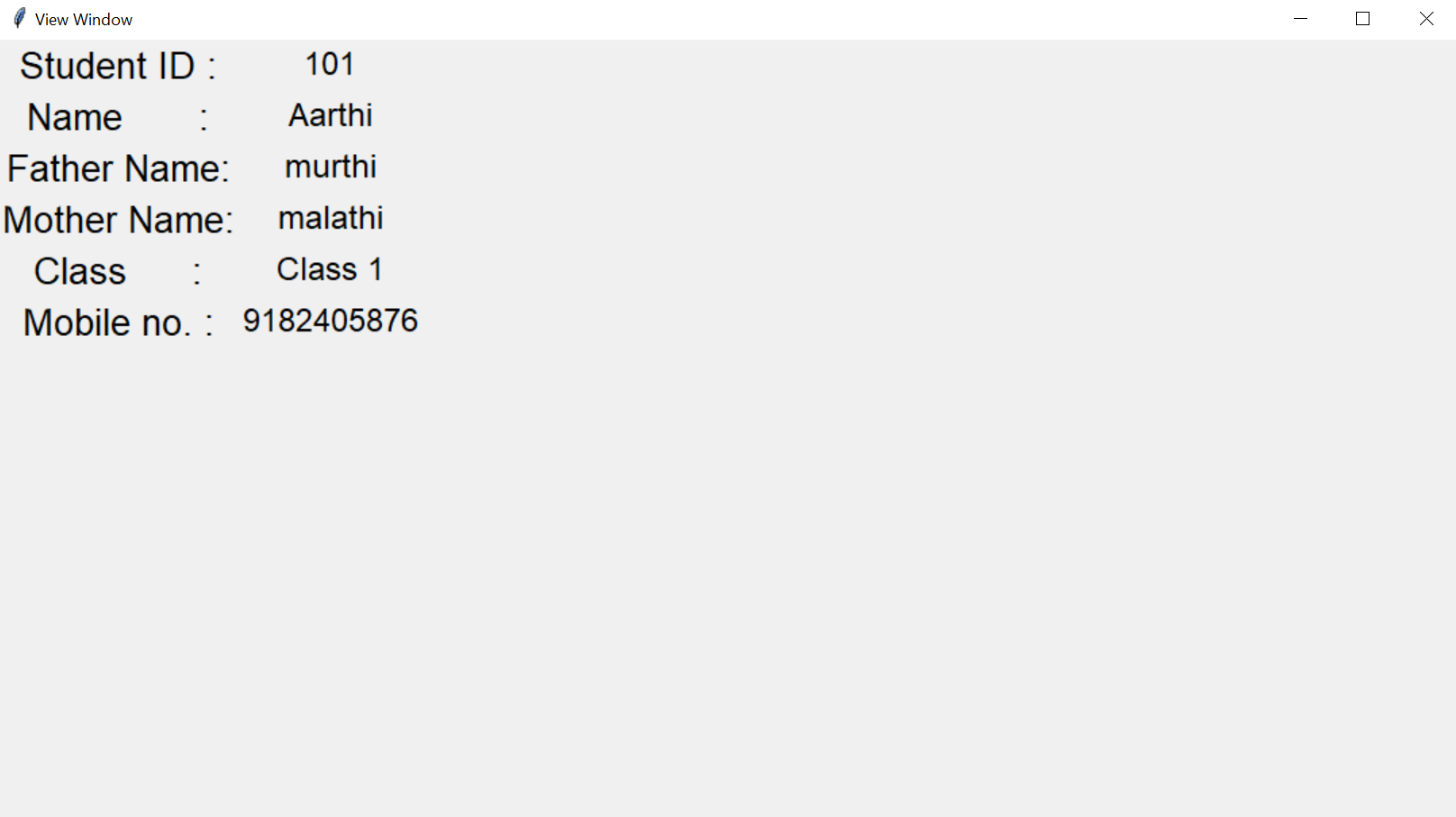
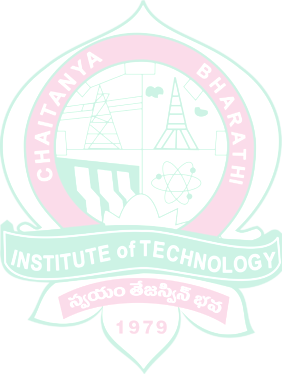
  

Fig 7.26: Teacher viewing student details

Fig 7.28: Viewing student details of same class

Fig 7.29: Student details belonging to same class

Fig 7.27: Student details

**TEACHER: Update Student Attendance**

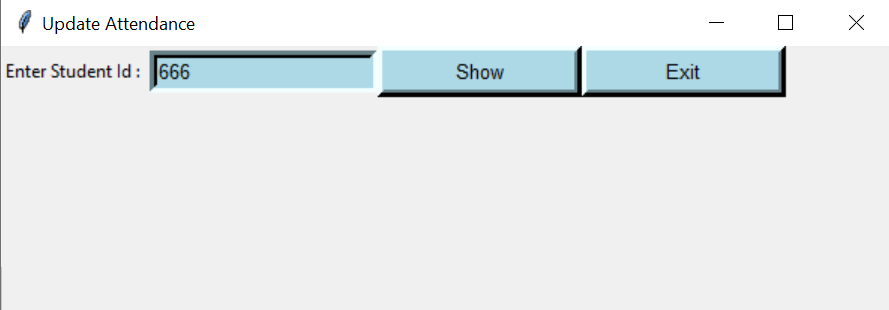


Fig 7.30: Updating attendance

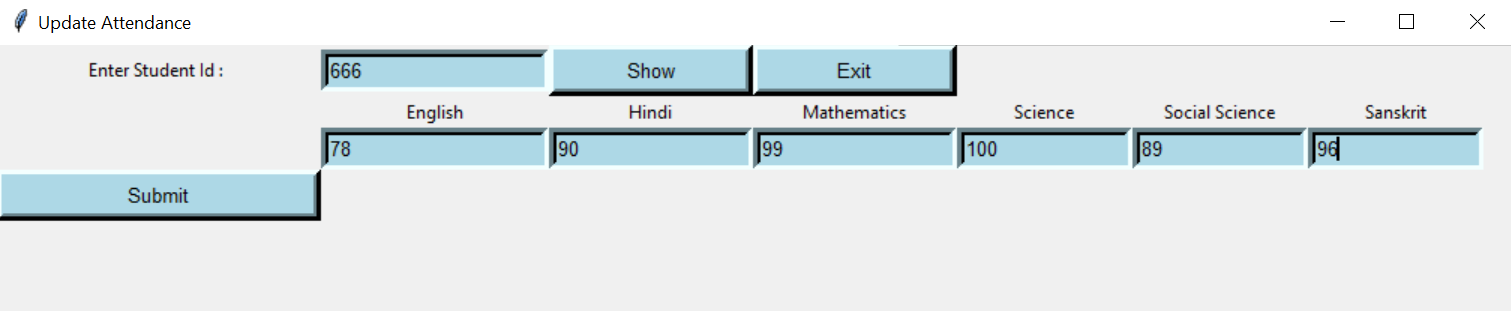


Fig 7.31: Entering attendance

**TEACHER: Update Student Marks**

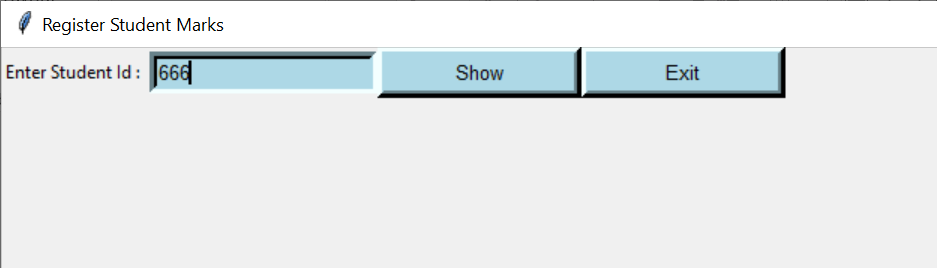


Fig 7.32: Updating marks

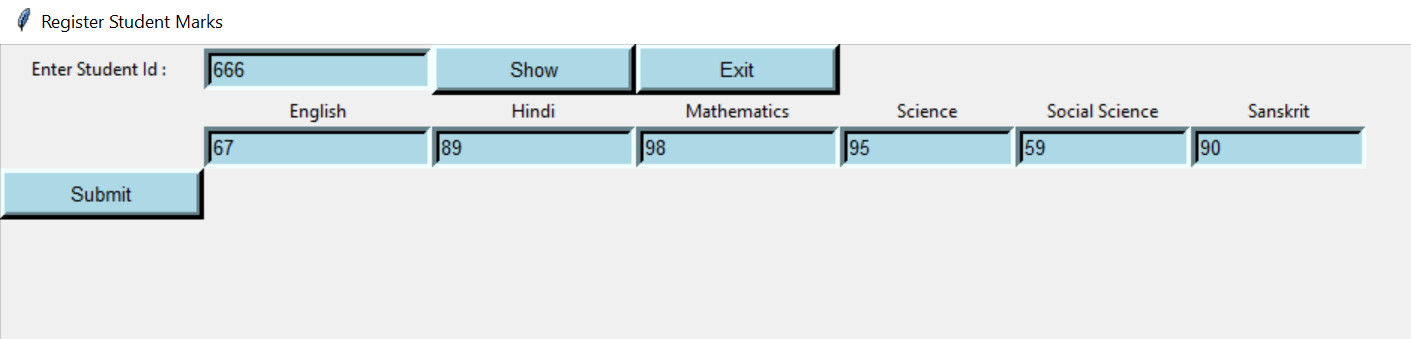
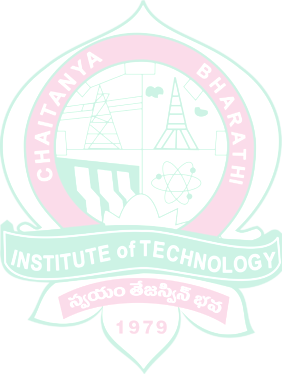


Fig 7.33: Entering marks

**TEACHER: Change Password**

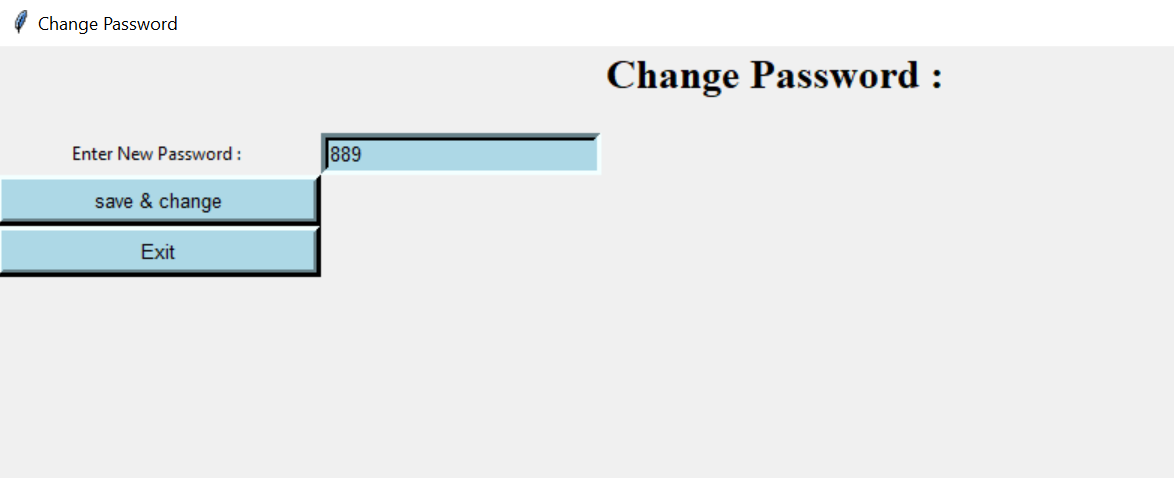
****

Fig 7.34: Changing password window

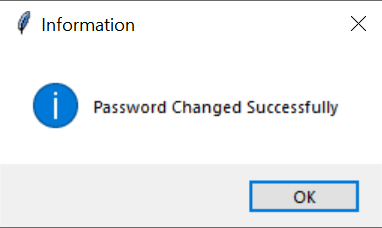
****

Fig 7.35: Message of successful password change

**LOGOUT:**

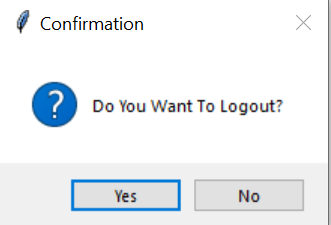
****

Fig 7.36: Logout confirmation message for all users

# CONCLUSION

# In recent years, with the pace of technological development, people have become more and more demanding in terms of quality of life, and the school’s managers in recent years look to improve a performance in their schools to get the highest rate of knowledge and experience in their student. The school management system is bringing a great difference in the lives of students, teachers, parents, and the admin. Good management offers better productivity and hence more progress towards development. It helps the school to achieve the target, reduce work, increase efficiency, eliminating error, and monitoring progress.

# FUTURE SCOPE

# Some ideas and features can be considered as a future work for this project. These features can be summarized in the following points:

# Student – Enable student take exams Online, Enable the student take an online certification courses.

# Teacher – Enable teacher to add notes for the respective subject.

# Parent - Let parent concatenation with other parents, let parent pay fee online

# REFERENCES

* http://dspace.up.edu.ps/jspui/bitstream/123456789/209/1/School%20Management%20System.pdf
* https://www.techprofree.com/school-management-system-project-in-python/
* https://www.sourcecodester.com/python/14520/school-management-system-project-python.html
* https://core.ac.uk/download/pdf/84656452.pdf
* https://itsourcecode.com/free-projects/python-projects/school-management-system-project-in-python-with-source-code/