**Group Members:**

**1.Mohammed Ammaar Khan (60017220106)**

**2.Hammad Khan (60017220022)**

**Introduction**

A Student Management System is a comprehensive software application that facilitates educational institutions in managing their day-to-day operations. It is designed to integrate and automate various administrative tasks of educational institutions, including registration, admissions, attendance tracking, grade management, and examination management.

The Student Management System has become an essential tool for educational institutions of all sizes, from small primary schools to large universities. In this tutorial, we’ll create such a **Student Management System Project in Python with MySQL**.

It will help to manage the following tasks:

1. Add new data of a student.
2. View the information of an existing student.
3. Update or modify a student’s data.
4. Delete the record of a student.

## The Project Details

The **graphical interface** of this project is managed by the [Tkinter](https://docs.python.org/3/library/tkinter.html" \t "_blank) library. The PyM ySQL package is used here to manage database operations using python.

The project folder contains three python files, **main.py**, **custom.py**, and **credentials.py**. As the name of **main.py**, it handles all the tasks. **custom.py** has information about the color and font used by the main program. **credentials.py** or the final file contains credentials to log in to the MySQL server. Here, users need to enter their own Username and Password for the MySQL server.

**Create a Database and Two Tables**

Create a database with this name: “**student\_management**“.

create database student\_management;

Create a table “**student\_register**” under the “**student\_management**” database.

create table student\_register(

f\_name VARCHAR(50) NOT NULL,

l\_name VARCHAR(50) NOT NULL,

course VARCHAR(30) NOT NULL,

subject VARCHAR(50) NOT NULL,

year Int(10) NOT NULL,

age Int(10) NOT NULL,

gender char(10) NOT NULL,

birth DATE NOT NULL,

contact VARCHAR(15) NOT NULL,

email VARCHAR(100) NOT NULL,

PRIMARY KEY ( contact )

);

**Requirements and Installation**

Make sure that you have Python installed on your system. You will also need to install PyMySQL library for connecting to and interacting with MySQL databases.

pip install PyMySQL

Note that, by-default Tkinter comes with most Python installations, so you likely won’t need to install it separately. However, if you don’t have it, you can install it using pip:

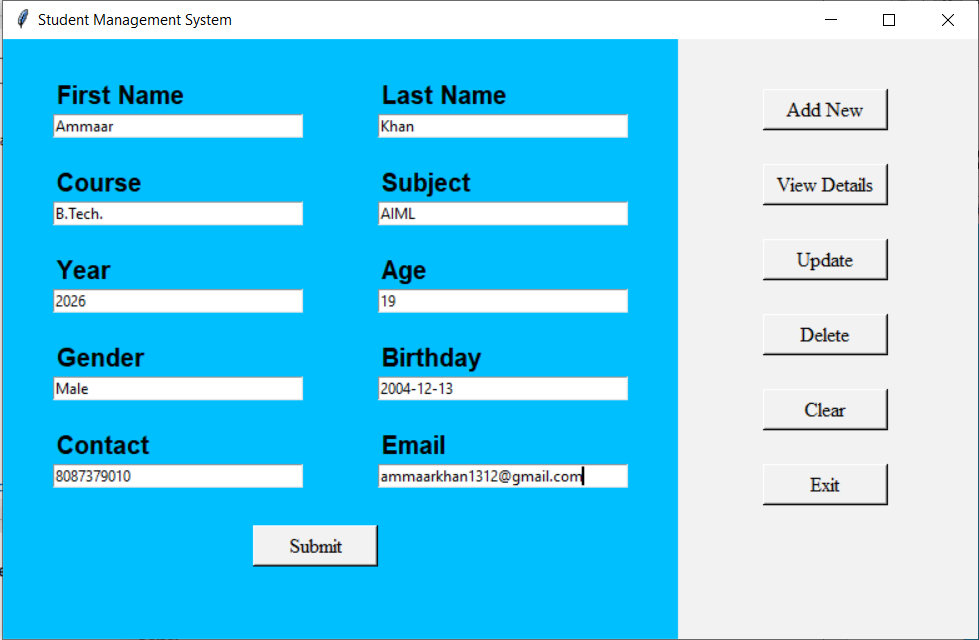
pip install tk

**What can you learn from this project?**

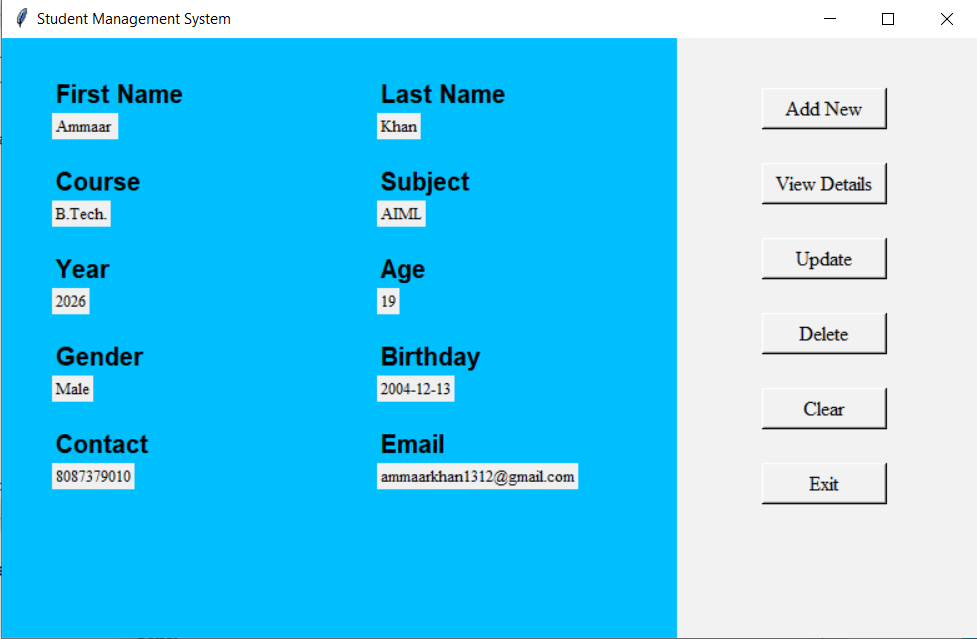
* **Creating a graphical interface with Tkinter**: Creating Tkinter windows, frames, labels, input widgets, buttons, etc.
* **Connecting Python to MySQL**: [How to connect python to MySQL](https://pyseek.blogspot.com/2021/01/how-to-connect-python-with-mysql-pyseek.html).
* **Database operations in python**: Access data from a table, update and delete operations, etc.
* Creating multiple python modules to manage several tasks.
* [Object Orient Programming in Python](https://pyseek.blogspot.com/2023/03/object-oriented-programming-in-python.html): Using classes, objects, etc.

**Output:**

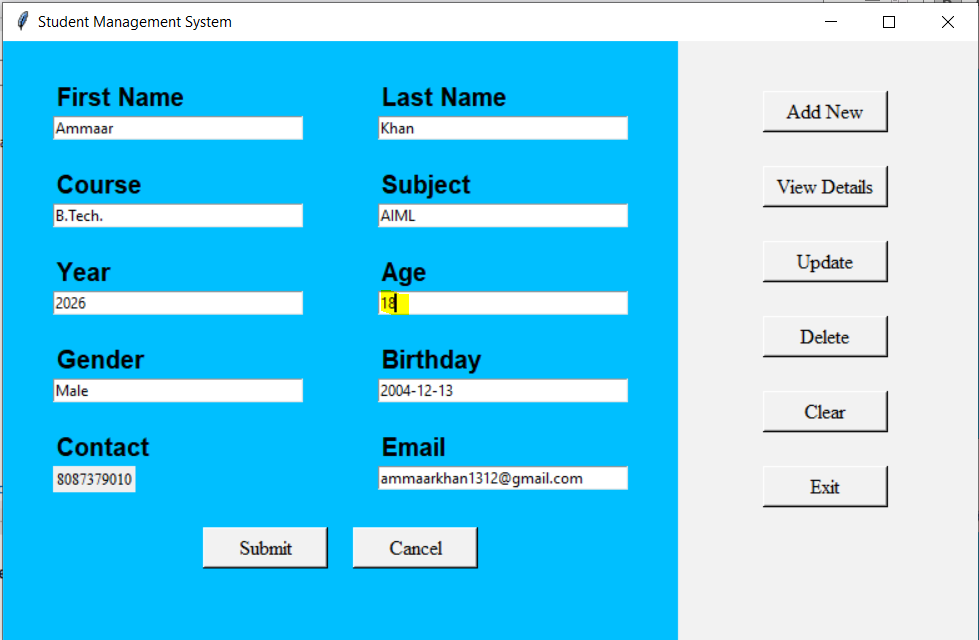
**Add New:**



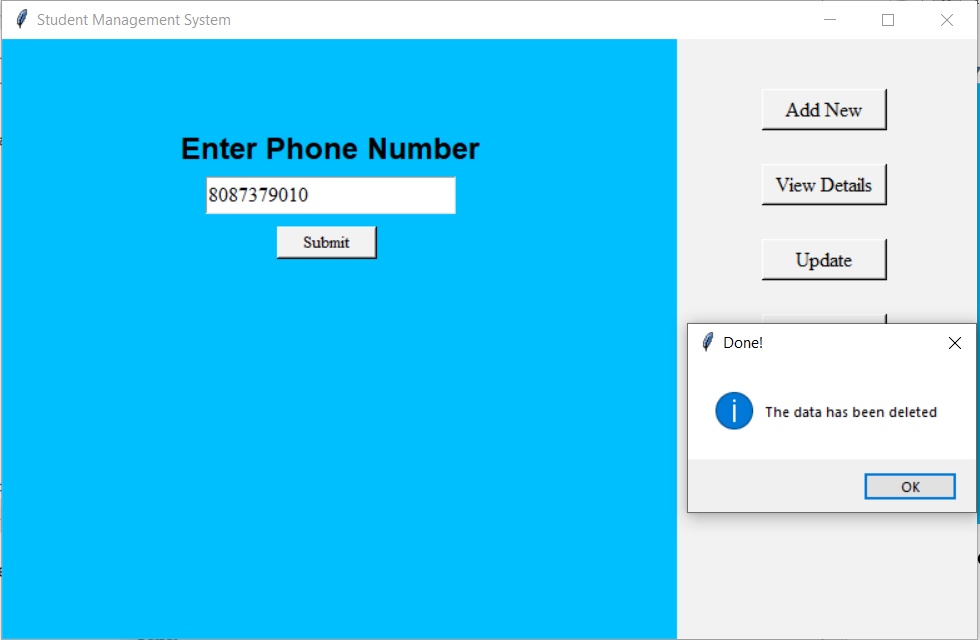
**View Details:**



**Update:**



**Delete:**



**Code:**

**Credentials.py :**

# User Credentials

host = 'localhost'

user = 'root'

password = '808737'

database = 'student\_management'

**Custom.py :**

color\_1 = "deep sky blue"

color\_2 = "gray95"

color\_3 = "black"

color\_4 = "white"

font\_1 = "times new roman"

font\_2 = "helvetica"

**Main.py :**

'''student management system project in python using tkinter and mysql'''

from functools import partial

from tkinter import \*

from tkinter import messagebox

import pymysql

import custom as cs

import credentials as cr

class Management:

    def \_\_init\_\_(self, root):

        self.window = root

        self.window.title("Student Management System")

        self.window.geometry("780x480")

        self.window.config(bg = "white")

        # Customization

        self.color\_1 = cs.color\_1

        self.color\_2 = cs.color\_2

        self.color\_3 = cs.color\_3

        self.color\_4 = cs.color\_4

        self.font\_1 = cs.font\_1

        self.font\_2 = cs.font\_2

        # User Credentials

        self.host = cr.host

        self.user = cr.user

        self.password = cr.password

        self.database = cr.database

        # Left Frame

        self.frame\_1 = Frame(self.window, bg=self.color\_1)

        self.frame\_1.place(x=0, y=0, width=540, relheight = 1)

        # Right Frame

        self.frame\_2 = Frame(self.window, bg = self.color\_2)

        self.frame\_2.place(x=540,y=0,relwidth=1, relheight=1)

        # Buttons

        self.add\_bt = Button(self.frame\_2, text='Add New', font=(self.font\_1, 12), bd=2, command=self.AddStudent, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=68,y=40,width=100)

        self.view\_bt = Button(self.frame\_2, text='View Details', font=(self.font\_1, 12), bd=2, command=self.GetContact\_View, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=68,y=100,width=100)

        self.update\_bt = Button(self.frame\_2, text='Update', font=(self.font\_1, 12), bd=2, command=self.GetContact\_Update, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=68,y=160,width=100)

        self.delete\_bt = Button(self.frame\_2, text='Delete', font=(self.font\_1, 12), bd=2, command=self.GetContact\_Delete,cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=68,y=220,width=100)

        self.clear\_bt = Button(self.frame\_2, text='Clear', font=(self.font\_1, 12), bd=2, command=self.ClearScreen, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=68,y=280,width=100)

        self.exit\_bt = Button(self.frame\_2, text='Exit', font=(self.font\_1, 12), bd=2, command=self.Exit, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=68,y=340,width=100)

    '''Widgets for adding student data'''

    def AddStudent(self):

        self.ClearScreen()

        self.name = Label(self.frame\_1, text="First Name", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=30)

        self.name\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.name\_entry.place(x=40,y=60, width=200)

        self.surname = Label(self.frame\_1, text="Last Name", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=30)

        self.surname\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.surname\_entry.place(x=300,y=60, width=200)

        self.course = Label(self.frame\_1, text="Course", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=100)

        self.course\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.course\_entry.place(x=40,y=130, width=200)

        self.subject = Label(self.frame\_1, text="Subject", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=100)

        self.subject\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.subject\_entry.place(x=300,y=130, width=200)

        self.year = Label(self.frame\_1, text="Year", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=170)

        self.year\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.year\_entry.place(x=40,y=200, width=200)

        self.age = Label(self.frame\_1, text="Age", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=170)

        self.age\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.age\_entry.place(x=300,y=200, width=200)

        self.gender = Label(self.frame\_1, text="Gender", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=240)

        self.gender\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.gender\_entry.place(x=40,y=270, width=200)

        self.birth = Label(self.frame\_1, text="Birthday", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=240)

        self.birth\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.birth\_entry.place(x=300,y=270, width=200)

        self.contact = Label(self.frame\_1, text="Contact", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=310)

        self.contact\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.contact\_entry.place(x=40,y=340, width=200)

        self.email = Label(self.frame\_1, text="Email", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=310)

        self.email\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.email\_entry.place(x=300,y=340, width=200)

        self.submit\_bt\_1 = Button(self.frame\_1, text='Submit', font=(self.font\_1, 12), bd=2, command=self.Submit, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=200,y=389,width=100)

    '''Get the contact number to show a student details'''

    def GetContact\_View(self):

        self.ClearScreen()

        self.getInfo = Label(self.frame\_1, text="Enter Phone Number", font=(self.font\_2, 18, "bold"), bg=self.color\_1).place(x=140,y=70)

        self.getInfo\_entry = Entry(self.frame\_1, font=(self.font\_1, 12), bg=self.color\_4, fg=self.color\_3)

        self.getInfo\_entry.place(x=163, y=110, width=200, height=30)

        self.submit\_bt\_2 = Button(self.frame\_1, text='Submit', font=(self.font\_1, 10), bd=2, command=self.CheckContact\_View, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=220,y=150,width=80)

    '''To update a student details, get the contact number'''

    def GetContact\_Update(self):

        self.ClearScreen()

        self.getInfo = Label(self.frame\_1, text="Enter Phone Number", font=(self.font\_2, 18, "bold"), bg=self.color\_1).place(x=140,y=70)

        self.getInfo\_entry = Entry(self.frame\_1, font=(self.font\_1, 12), bg=self.color\_4, fg=self.color\_3)

        self.getInfo\_entry.place(x=163, y=110, width=200, height=30)

        self.submit\_bt\_2 = Button(self.frame\_1, text='Submit', font=(self.font\_1, 10), bd=2, command=self.CheckContact\_Update, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=220,y=150,width=80)

    '''Get the contact number to delete a student record'''

    def GetContact\_Delete(self):

        self.ClearScreen()

        self.getInfo = Label(self.frame\_1, text="Enter Phone Number", font=(self.font\_2, 18, "bold"), bg=self.color\_1).place(x=140,y=70)

        self.getInfo\_entry = Entry(self.frame\_1, font=(self.font\_1, 12), bg=self.color\_4, fg=self.color\_3)

        self.getInfo\_entry.place(x=163, y=110, width=200, height=30)

        self.submit\_bt\_2 = Button(self.frame\_1, text='Submit', font=(self.font\_1, 10), bd=2, command=self.DeleteData, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=220,y=150,width=80)

    '''Remove all widgets from the frame 1'''

    def ClearScreen(self):

        for widget in self.frame\_1.winfo\_children():

            widget.destroy()

    '''Exit window'''

    def Exit(self):

        self.window.destroy()

    '''

    Checks whether the contact number is available or not. If available,

    the function calls the 'ShowDetails' function to display the result.

    '''

    def CheckContact\_View(self):

        if self.getInfo\_entry.get() == "":

            messagebox.showerror("Error!", "Please enter your contact number",parent=self.window)

        else:

            try:

                connection = pymysql.connect(host=self.host, user=self.user, password=self.password, database=self.database)

                curs = connection.cursor()

                curs.execute("select \* from student\_register where contact=%s", self.getInfo\_entry.get())

                row=curs.fetchone()

                if row == None:

                    messagebox.showerror("Error!","Contact number doesn't exists",parent=self.window)

                else:

                    self.ShowDetails(row)

                    connection.close()

            except Exception as e:

                messagebox.showerror("Error!",f"Error due to {str(e)}",parent=self.window)

    '''

    Checks whether the contact number is available or not. If available,

    the function calls the 'GetUpdateDetails' function to get the new data to perform

    update operation.

    '''

    def CheckContact\_Update(self):

        if self.getInfo\_entry.get() == "":

            messagebox.showerror("Error!", "Please enter your contact number",parent=self.window)

        else:

            try:

                connection = pymysql.connect(host=self.host, user=self.user, password=self.password, database=self.database)

                curs = connection.cursor()

                curs.execute("select \* from student\_register where contact=%s", self.getInfo\_entry.get())

                row=curs.fetchone()

                if row == None:

                    messagebox.showerror("Error!","Contact number doesn't exists",parent=self.window)

                else:

                    self.GetUpdateDetails(row)

                    connection.close()

            except Exception as e:

                messagebox.showerror("Error!",f"Error due to {str(e)}",parent=self.window)

    '''Clears a student record'''

    def DeleteData(self):

        if self.getInfo\_entry.get() == "":

            messagebox.showerror("Error!", "Please enter your contact number",parent=self.window)

        else:

            try:

                connection = pymysql.connect(host=self.host, user=self.user, password=self.password, database=self.database)

                curs = connection.cursor()

                curs.execute("select \* from student\_register where contact=%s", self.getInfo\_entry.get())

                row=curs.fetchone()

                if row == None:

                    messagebox.showerror("Error!","Contact number doesn't exists",parent=self.window)

                else:

                    curs.execute("delete from student\_register where contact=%s", self.getInfo\_entry.get())

                    connection.commit()

                    messagebox.showinfo('Done!', "The data has been deleted")

                    connection.close()

                    self.ClearScreen()

            except Exception as e:

                messagebox.showerror("Error!",f"Error due to {str(e)}",parent=self.window)

    '''Gets the data that the user wants to update to perform the update operation'''

    def GetUpdateDetails(self, row):

        self.ClearScreen()

        self.name = Label(self.frame\_1, text="First Name", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=30)

        self.name\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.name\_entry.insert(0, row[0])

        self.name\_entry.place(x=40,y=60, width=200)

        self.surname = Label(self.frame\_1, text="Last Name", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=30)

        self.surname\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.surname\_entry.insert(0, row[1])

        self.surname\_entry.place(x=300,y=60, width=200)

        self.course = Label(self.frame\_1, text="Course", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=100)

        self.course\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.course\_entry.insert(0, row[2])

        self.course\_entry.place(x=40,y=130, width=200)

        self.subject = Label(self.frame\_1, text="Subject", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=100)

        self.subject\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.subject\_entry.insert(0, row[3])

        self.subject\_entry.place(x=300,y=130, width=200)

        self.year = Label(self.frame\_1, text="Year", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=170)

        self.year\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.year\_entry.insert(0, row[4])

        self.year\_entry.place(x=40,y=200, width=200)

        self.age = Label(self.frame\_1, text="Age", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=170)

        self.age\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.age\_entry.insert(0, row[5])

        self.age\_entry.place(x=300,y=200, width=200)

        self.gender = Label(self.frame\_1, text="Gender", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=240)

        self.gender\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.gender\_entry.insert(0, row[6])

        self.gender\_entry.place(x=40,y=270, width=200)

        self.birth = Label(self.frame\_1, text="Birthday", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=240)

        self.birth\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.birth\_entry.insert(0, row[7])

        self.birth\_entry.place(x=300,y=270, width=200)

        contact = Label(self.frame\_1, text="Contact", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=310)

        contact\_data = Label(self.frame\_1, text=row[8], font=(self.font\_1, 10)).place(x=40, y=340)

        self.email = Label(self.frame\_1, text="Email", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=310)

        self.email\_entry = Entry(self.frame\_1, bg=self.color\_4, fg=self.color\_3)

        self.email\_entry.insert(0, row[9])

        self.email\_entry.place(x=300,y=340, width=200)

        self.submit\_bt\_1 = Button(self.frame\_1, text='Submit', font=(self.font\_1, 12), bd=2, command=partial(self.UpdateDetails, row), cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=160,y=389,width=100)

        self.cancel\_bt = Button(self.frame\_1, text='Cancel', font=(self.font\_1, 12), bd=2, command=self.ClearScreen, cursor="hand2", bg=self.color\_2,fg=self.color\_3).place(x=280,y=389,width=100)

    '''Within frame 1, it displays information about a student'''

    def ShowDetails(self, row):

        self.ClearScreen()

        name = Label(self.frame\_1, text="First Name", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=30)

        name\_data = Label(self.frame\_1, text=row[0], font=(self.font\_1, 10)).place(x=40, y=60)

        surname = Label(self.frame\_1, text="Last Name", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=30)

        surname\_data = Label(self.frame\_1, text=row[1], font=(self.font\_1, 10)).place(x=300, y=60)

        course = Label(self.frame\_1, text="Course", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=100)

        course\_data = Label(self.frame\_1, text=row[2], font=(self.font\_1, 10)).place(x=40, y=130)

        subject = Label(self.frame\_1, text="Subject", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=100)

        subject\_data = Label(self.frame\_1, text=row[3], font=(self.font\_1, 10)).place(x=300, y=130)

        year = Label(self.frame\_1, text="Year", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=170)

        year\_data = Label(self.frame\_1, text=row[4], font=(self.font\_1, 10)).place(x=40, y=200)

        age = Label(self.frame\_1, text="Age", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=170)

        age\_data = Label(self.frame\_1, text=row[5], font=(self.font\_1, 10)).place(x=300, y=200)

        gender = Label(self.frame\_1, text="Gender", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=240)

        gender\_data = Label(self.frame\_1, text=row[6], font=(self.font\_1, 10)).place(x=40, y=270)

        birth = Label(self.frame\_1, text="Birthday", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=240)

        birth\_data = Label(self.frame\_1, text=row[7], font=(self.font\_1, 10)).place(x=300, y=270)

        contact = Label(self.frame\_1, text="Contact", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=40,y=310)

        contact\_data = Label(self.frame\_1, text=row[8], font=(self.font\_1, 10)).place(x=40, y=340)

        email = Label(self.frame\_1, text="Email", font=(self.font\_2, 15, "bold"), bg=self.color\_1).place(x=300,y=310)

        email\_data = Label(self.frame\_1, text=row[9], font=(self.font\_1, 10)).place(x=300, y=340)

    '''Updates student data'''

    def UpdateDetails(self, row):

        if self.name\_entry.get() == "" or self.surname\_entry.get() == "" or self.course\_entry.get() == "" or self.subject\_entry.get() == "" or self.year\_entry.get() == "" or self.age\_entry.get() == "" or self.gender\_entry.get() == "" or self.birth\_entry.get() == "" or self.email\_entry.get() == "":

            messagebox.showerror("Error!","Sorry!, All fields are required",parent=self.window)

        else:

            try:

                connection = pymysql.connect(host=self.host, user=self.user, password=self.password, database=self.database)

                curs = connection.cursor()

                curs.execute("select \* from student\_register where contact=%s", row[8])

                row=curs.fetchone()

                if row==None:

                    messagebox.showerror("Error!","The contact number doesn't exists",parent=self.window)

                else:

                    curs.execute("update student\_register set f\_name=%s,l\_name=%s, course=%s, subject=%s, year=%s, age=%s, gender=%s, birth=%s, email=%s where contact=%s",

                                        (

                                            self.name\_entry.get(),

                                            self.surname\_entry.get(),

                                            self.course\_entry.get(),

                                            self.subject\_entry.get(),

                                            self.year\_entry.get(),

                                            self.age\_entry.get(),

                                            self.gender\_entry.get(),

                                            self.birth\_entry.get(),

                                            self.email\_entry.get(),

                                            row[8]

                                        ))

                    connection.commit()

                    connection.close()

                    messagebox.showinfo('Done!', "The data has been updated")

                    self.ClearScreen()

            except Exception as e:

                messagebox.showerror("Error!",f"Error due to {str(e)}",parent=self.window)

    '''It adds the information of new students'''

    def Submit(self):

        if self.name\_entry.get() == "" or self.surname\_entry.get() == "" or self.course\_entry.get() == "" or self.subject\_entry.get() == "" or self.year\_entry.get() == "" or self.age\_entry.get() == "" or self.gender\_entry.get() == "" or self.birth\_entry.get() == "" or self.contact\_entry.get() == "" or self.email\_entry.get() == "":

            messagebox.showerror("Error!","Sorry!, All fields are required",parent=self.window)

        else:

            try:

                connection = pymysql.connect(host=self.host, user=self.user, password=self.password, database=self.database)

                curs = connection.cursor()

                curs.execute("select \* from student\_register where contact=%s", self.contact\_entry.get())

                row=curs.fetchone()

                if row!=None:

                    messagebox.showerror("Error!","The contact number is already exists, please try again with another number",parent=self.window)

                else:

                    curs.execute("insert into student\_register (f\_name,l\_name,course,subject,year,age,gender,birth,contact,email) values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",

                                        (

                                            self.name\_entry.get(),

                                            self.surname\_entry.get(),

                                            self.course\_entry.get(),

                                            self.subject\_entry.get(),

                                            self.year\_entry.get(),

                                            self.age\_entry.get(),

                                            self.gender\_entry.get(),

                                            self.birth\_entry.get(),

                                            self.contact\_entry.get(),

                                            self.email\_entry.get()

                                        ))

                    connection.commit()

                    connection.close()

                    messagebox.showinfo('Done!', "The data has been submitted")

                    self.reset\_fields()

            except Exception as e:

                messagebox.showerror("Error!",f"Error due to {str(e)}",parent=self.window)

    '''Reset all the entry fields'''

    def reset\_fields(self):

        self.name\_entry.delete(0, END)

        self.surname\_entry.delete(0, END)

        self.course\_entry.delete(0, END)

        self.subject\_entry.delete(0, END)

        self.year\_entry.delete(0, END)

        self.age\_entry.delete(0, END)

        self.gender\_entry.delete(0, END)

        self.birth\_entry.delete(0, END)

        self.contact\_entry.delete(0, END)

        self.email\_entry.delete(0, END)

# The main function

if \_\_name\_\_ == "\_\_main\_\_":

    root = Tk()

    obj = Management(root)

    root.mainloop()

**MySQL:**

