



## **Project report**

**Project Name**  
**Student Management System**

## **Group Members**

**Jawad Ahmed Soomro (021-18-44532)**  
**Rida Taufeeq (021-20-50636)**

**Submitted To**  
**Sir Anees Ahmed**

**Course**  
**Introduction to Python**

# Table of Contents

<b>Problem Statement:</b> .....	3
<b>Introduction:</b> .....	3
<b>Tools:</b> .....	3
<b>Technologies:</b> .....	4
<b>Libraries:</b> .....	4
<b>Data Flow Diagram</b> .....	5
<b>Code Screenshot:</b> .....	6
<b>Project Output:</b> .....	10
<b>Project video:</b> .....	10

## Problem Statement:

- Student Management System can be used by education institutes to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using this project.

## Introduction:

- The objective of Student Management System is to allow the administrator of any organization to edit and find out the personal details of a student. It'll also facilitate keeping all the records of students, such as their Registration id, First Name, Last Name, Father Name, DOB, gender, phone number, etc. So all the information about a student will be available in a few seconds. Overall, it'll make Student Information Management an easier job for the administrator any organization. The main purpose of this Student Management System document is to illustrate the requirements of the project Student Management System and is intended to help any organization to maintain and manage its student's data.

## Tools:

- **SQLyog Community - 64 bit**

SQLyog Community Edition is a powerful program that enables you to work with MySQL databases using a visual interface. If you are not used to the command line interface, this tool turns the MySQL database management into an easier job.

- **PyCharm IDE**

PyCharm is one of the most intuitive and feature-rich integrated development environments (IDEs) available for Python development.

- **XAMPP server**

XAMPP is a small and light MySQL distribution containing the most common database development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL. XAMPP. The light version is a small package containing Apache HTTP Server, PHP, MySQL, phpMyAdmin, OpenSSL, and SQLite.

## Technologies:

- **Back End: MYSQL**
- **Front End: Python**

## Libraries:

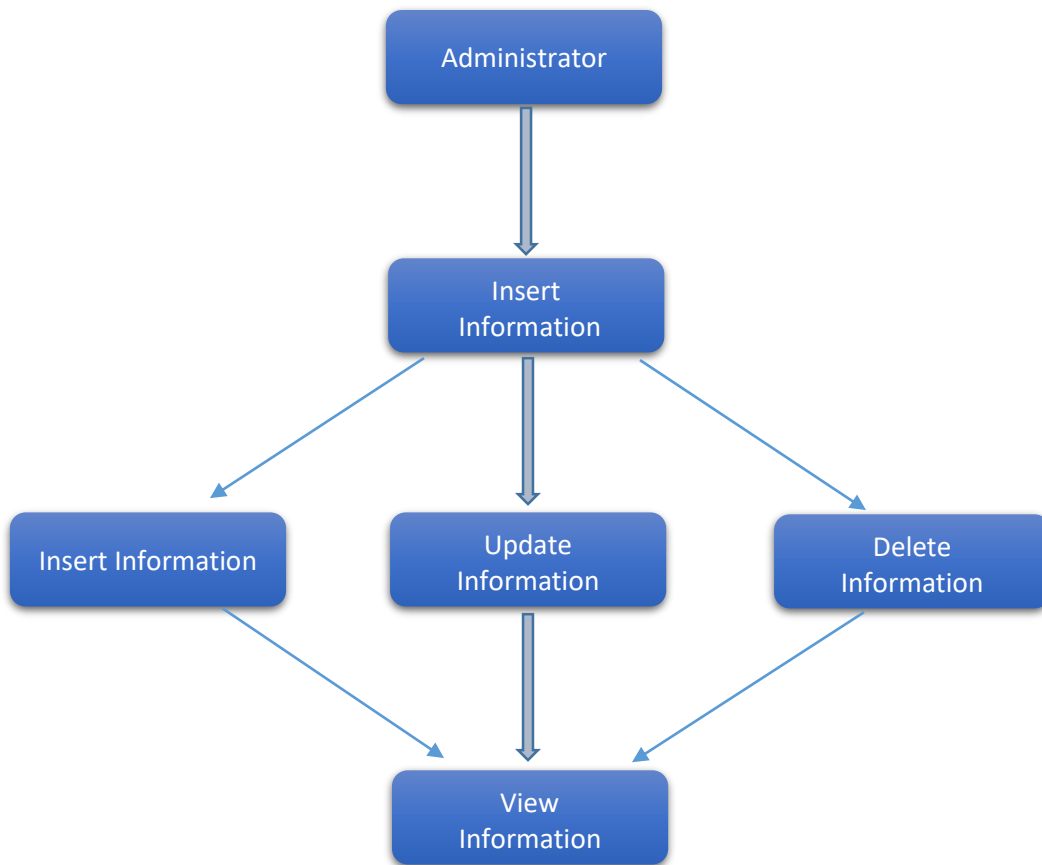
- **Mysql.connector**

The MySQL Connector/ ODBC (Open Database Connectivity) is the name for the family of MySQL ODBC drivers provided by Microsoft to access a MySQL Database using the standard ODBC API. MySQL Connector/ODBC provides both driver-manager based and native interfaces to connect to the MySQL Database.

- **Tkinter**

Tkinter is an open source, portable graphical user interface (GUI) library designed for use in Python scripts. Tkinter relies on the Tk library, the GUI library used by Tcl/Tk and Perl, which is in turn implemented in C. Therefore, Tkinter can be said to be implemented using multiple layers.

## Data Flow Diagram



## Code Screenshot:

```
main.py x
1  import tkinter as tk
2      from tkinter import ttk, messagebox
3      import mysql.connector
4  from tkinter import *
5
6
7  def GetValue(event):
8      e1.delete(0, END)
9      e2.delete(0, END)
10     e3.delete(0, END)
11     e4.delete(0, END)
12     e5.delete(0, END)
13     e6.delete(0, END)
14     e7.delete(0, END)
15     e8.delete(0, END)
16
17     row_id = listBox.selection()[0]
18     select = listBox.set(row_id)
19     e1.insert(0, select['RegNo'])
20     e2.insert(0, select['First_Name'])
21     e3.insert(0, select['Last_Name'])
22     e4.insert(0, select['Fname'])
23     e5.insert(0, select['DOB'])
24     e6.insert(0, select['Gender'])
25     e7.insert(0, select['Contact'])
26     e8.insert(0, select['Address'])
27
28
29  def Add():
30      RegNo = e1.get()
31      First_Name = e2.get()
32      Last_Name = e3.get()
33      Fname = e4.get()
34      DOB = e5.get()
35      Gender = e6.get()
36      Contact = e7.get()
37      Address = e8.get()
38
39      mysql = mysql.connector.connect(host='localhost', user='root', password='', database='management')
40      mycursor = mysql.cursor()
41
42      try:
43          sql = "INSERT INTO registration (RegNo,First_Name,Last_Name,Fname,DOB,Gender>Contact,Address) VALUES (%s,%s,%s,%s,%s,%s,%s,%s)"
44          val = (RegNo,First_Name,Last_Name,Fname,DOB,Gender>Contact,Address)
45          mycursor.execute(sql, val)
46          mysql.commit()
47          lastid = mycursor.lastrowid
48          messagebox.showinfo("INSERT TABLE", "Student Data Inserted Successfully ")
49          e1.delete(0, END)
50          e2.delete(0, END)
51          e3.delete(0, END)
52          e4.delete(0, END)
53          e5.delete(0, END)
54          e6.delete(0, END)
55          e7.delete(0, END)
56          e8.delete(0, END)
57          e1.focus_set()
58      except Exception as e:
59          print(e)
60          mysql.rollback()
61          mysql.close()
62
```

```
main.py
64 def update():
65     RegNo = e1.get()
66     First_Name = e2.get()
67     Last_Name = e3.get()
68     Fname = e4.get()
69     DOB = e5.get()
70     Gender = e6.get()
71     Contact = e7.get()
72     Address = e8.get()
73     mysqldb = mysql.connector.connect(host="localhost", user="root", password="", database="management")
74     mycursor = mysqldb.cursor()
75     try:
76         sql = "Update registration set First_Name=%s,Last_Name=%s,Fname=%s,DOB=%s,Gender=%s,Contact=%s,Address=%s where RegNo=%s"
77         val = (First_Name,Last_Name,Fname,DOB,Gender,Contact,Address,RegNo)
78         mycursor.execute(sql, val)
79         mysqldb.commit()
80         lastid = mycursor.lastrowid
81         messagebox.showinfo("UPDATE TABLE", "Student Data Updated Successfully ")
82
83         e1.delete(0, END)
84         e2.delete(0, END)
85         e3.delete(0, END)
86         e4.delete(0, END)
87         e5.delete(0, END)
88         e6.delete(0, END)
89         e7.delete(0, END)
90         e8.delete(0, END)
91         e1.focus_set()
92
93 except Exception as e:
94     print(e)
95     mysqldb.rollback()
96     mysqldb.close()
97
main.py
99 def delete():
100     RegNo = e1.get()
101
102     mysqldb = mysql.connector.connect(host="localhost", user="root", password="", database="management")
103     mycursor = mysqldb.cursor()
104
105     try:
106         sql = "delete from registration where RegNo = %s"
107         val = (RegNo,)
108         mycursor.execute(sql, val)
109         mysqldb.commit()
110         lastid = mycursor.lastrowid
111         messagebox.showinfo("RECORD TABLE", "Student Data Deleted successfully ")
112
113         e1.delete(0, END)
114         e2.delete(0, END)
115         e3.delete(0, END)
116         e4.delete(0, END)
117         e5.delete(0, END)
118         e6.delete(0, END)
119         e7.delete(0, END)
120         e8.delete(0, END)
121         e1.focus_set()
122
123 except Exception as e:
124
125     print(e)
126     mysqldb.rollback()
127     mysqldb.close()
```

```
main.py x
130 def show():
131     mysqldb = mysql.connector.connect(host="localhost", user="root", password="", database="management")
132     mycursor = mysqldb.cursor()
133     mycursor.execute("SELECT RegNo,First_Name,Last_Name,Fname,DOB,Gender,Contact,Address FROM registration")
134     records = mycursor.fetchall()
135     print(records)
136
137     for i, (RegNo,First_Name,Last_Name,Fname,DOB,Gender,Contact,Address) in enumerate(records, start=1):
138         listBox.insert("", "end", values=(RegNo,First_Name,Last_Name,Fname,DOB,Gender,Contact,Address))
139     mysqldb.close()
140
141
142 root = Tk()
143 root.title("Student Management System")
144 root.geometry("900x580")
145 root.configure(bg='lightblue')
146 global e1
147 global e2
148 global e3
149 global e4
150 global e5
151 global e6
152 global e7
153 global e8
154
155 tk.Label(root,bg='lightblue', text="Student Management System ", fg="blue", font=("Algerian", 30)).place(x=165, y=5)
156
157 tk.Label(root,bg='lightblue', text="Registration No").place(x=110, y=80)
158 Label(root,bg='lightblue', text="First Name").place(x=110, y=110)
159 Label(root,bg='lightblue', text="Last Name").place(x=110, y=140)
160 Label(root,bg='lightblue', text="Father Name").place(x=110, y=170)
161
162 Label(root,bg='lightblue', text="Date of Birth").place(x=550, y=80)
163 Label(root,bg='lightblue', text="Gender").place(x=550, y=110)
164 Label(root,bg='lightblue', text="Contact").place(x=550, y=140)
165 Label(root,bg='lightblue', text="Address").place(x=550, y=170)
166
167
168 e1 = Entry(root)
169 e1.place(x=250, y=80)
170
171 e2 = Entry(root)
172 e2.place(x=250, y=110)
173
174 e3 = Entry(root)
175 e3.place(x=250, y=140)
176
177 e4 = Entry(root)
178 e4.place(x=250, y=170)
```

```
main.py x
177
180 e5 = Entry(root)
181 e5.place(x=650, y=80)
182
183 e6 = Entry(root)
184 e6.place(x=650, y=110)
185
186 e7 = Entry(root)
187 e7.place(x=650, y=140)
188
189 e8 = Entry(root)
190 e8.place(x=650, y=170)
191
192
193 Button(root, text="ADD", command=Add, height=3, width=13).place(x=225, y=250)
194 Button(root, text="UPDATE", command=update, height=3, width=13).place(x=400, y=250)
195 Button(root, text="DELETE", command=delete, height=3, width=13).place(x=575, y=250)
196
197 cols = ('RegNo', 'First_Name', 'Last_Name', 'Fname', 'DOB', 'Gender', 'Contact', 'Address')
198
199 listBox = ttk.Treeview(root, columns=cols, show='headings')
200 listBox.column('RegNo', width=50, anchor=CENTER)
201 listBox.column('First_Name', width=120, anchor=CENTER)
202 listBox.column('Last_Name', width=120, anchor=CENTER)
203
main.py x
197 cols = ('RegNo', 'First_Name', 'Last_Name', 'Fname', 'DOB', 'Gender', 'Contact', 'Address')
198
199 listBox = ttk.Treeview(root, columns=cols, show='headings')
200 listBox.column('RegNo', width=50, anchor=CENTER)
201 listBox.column('First_Name', width=120, anchor=CENTER)
202 listBox.column('Last_Name', width=120, anchor=CENTER)
203 listBox.column('Fname', width=120, anchor=CENTER)
204 listBox.column('DOB', width=120, anchor=CENTER)
205 listBox.column('Gender', width=120, anchor=CENTER)
206 listBox.column('Contact', width=120, anchor=CENTER)
207 listBox.column('Address', width=123, anchor=CENTER)
208
209 for col in cols:
210     listBox.heading(col, text=col)
211     listBox.grid(row=0, column=0, columnspan=2)
212     listBox.place(x=2, y=350)
213
214 show()
215 listBox.bind('<Double-Button-1>', GetValue)
216
217 root.mainloop()
```

Use HTTP/SSH Tunneling to connect to MySQL even if direct connection is disallowed : Reason #36 to upgrade

```
Query 1 +
1 CREATE DATABASE management
2
3 CREATE TABLE registration
4 (
5     RegNo INT(20),
6     First_Name VARCHAR(50),
7     Last_Name VARCHAR(50),
8     Fname VARCHAR(50)
9     DOB DATE,
10    Gender VARCHAR(10),
11    Contact INT(20),
12    Address VARCHAR(50),
13 );
14
```

## Project Output:

**STUDENT MANAGEMENT SYSTEM**

Registration No	<input type="text"/>	Date of Birth	<input type="text"/>
First Name	<input type="text"/>	Gender	<input type="text"/>
Last Name	<input type="text"/>	Contact	<input type="text"/>
Father Name	<input type="text"/>	Address	<input type="text"/>

+ ADD
+ UPDATE
+ DELETE

RegNo	First_Name	Last_Name	Fname	DOB	Gender	Contact	Address
44532	Jawad Ahmed	Soomro	Rasool Bux	1999-11-10	Male	323423422	Karachi
50636	Rida	Taufeeq	Muhammad Taufeeq	2000-05-10	Female	342324324	Karachi

## Project video:

<https://drive.google.com/file/d/1FbMVwmjdctGr1N3kzQ3IKnvapt-YeEys/view?usp=sharing>