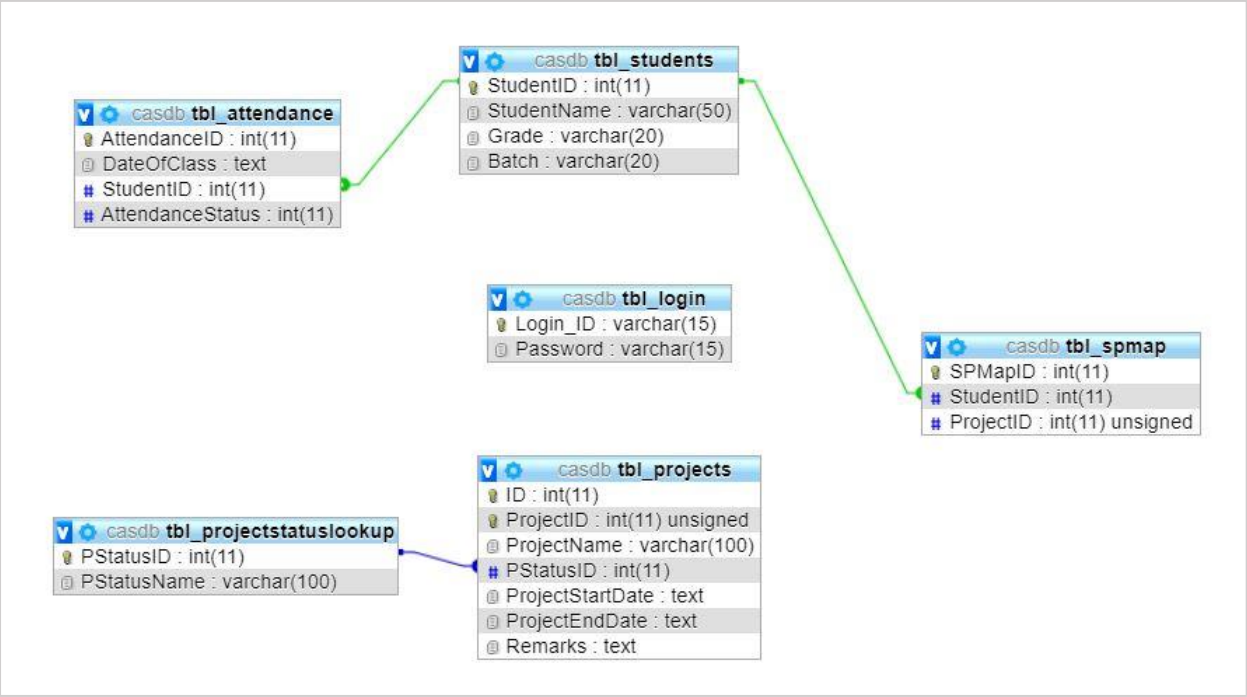
Criterion C

Features: -

- **Login**
- **Student credential storage, display and editing**
- **Project display storage, display, and editing**
- **Attendance of students**
- **Project status allotment**
- **Dashboard**

Database Design:



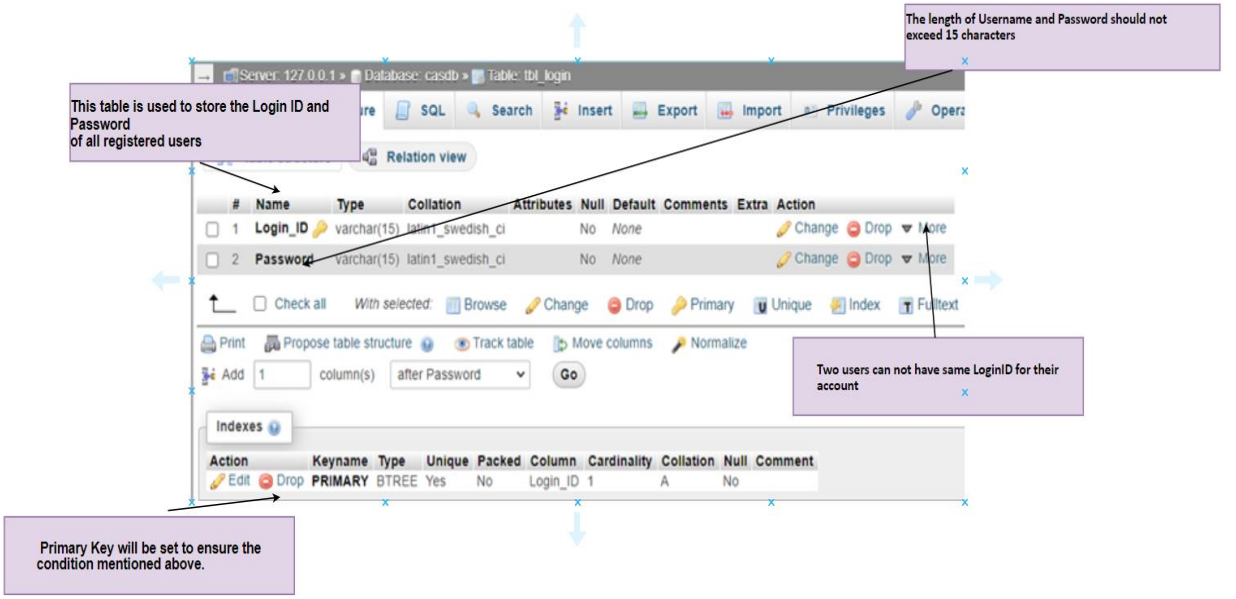
Software used: MySQL Server

Purpose: The relational database is used with normalization to provide referential integrity for the management of the data.

Tables:

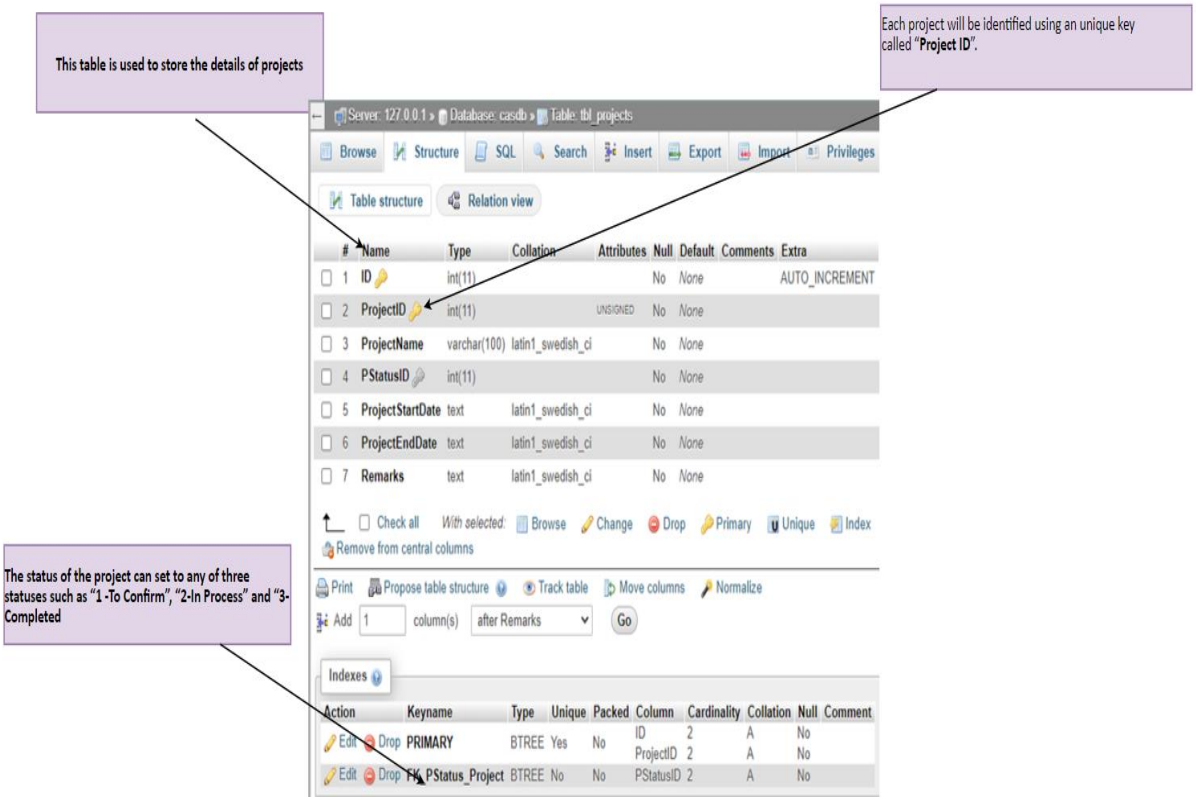
Name: tbl_Login

The length of Username and Password should not exceed 15 characters. Two users can not have same LoginID for their account. Primary Key will be set to ensure the condition mentioned above.



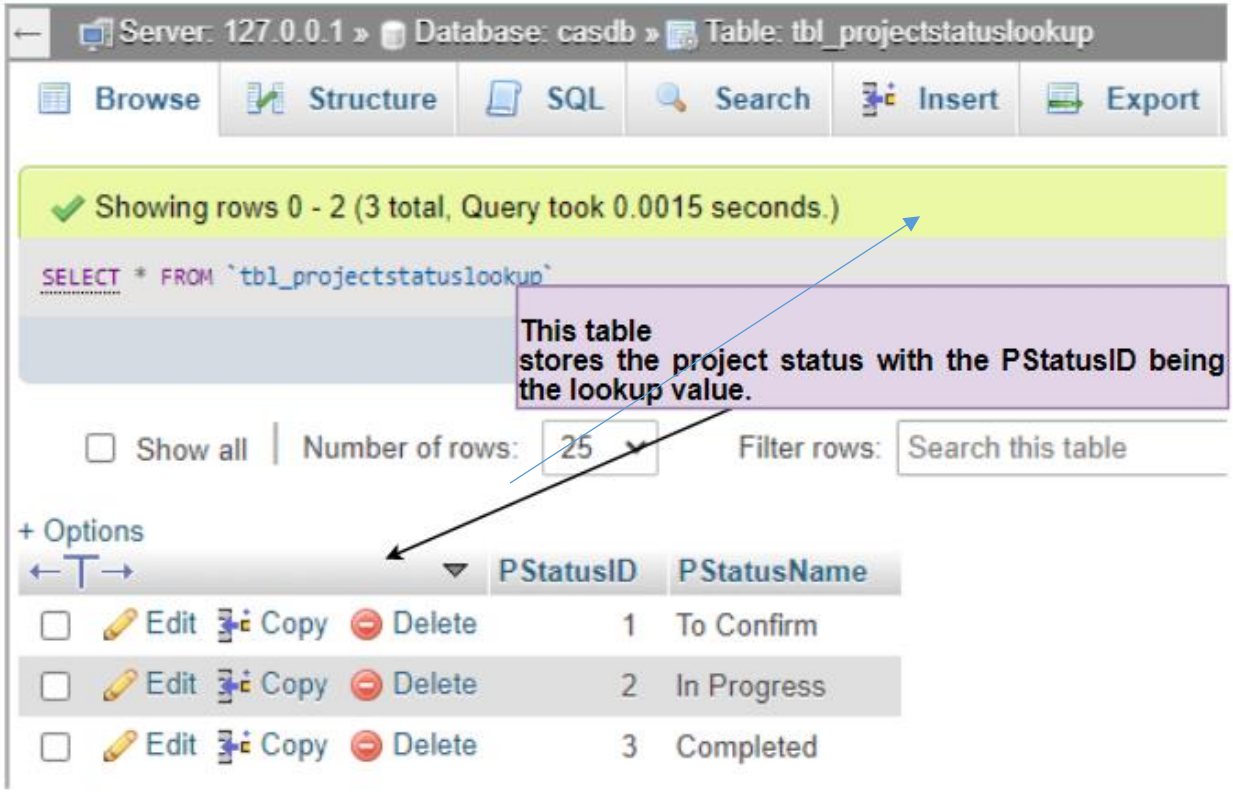
Name: tbl_Projects

This table is used to store the details of projects. Each project will be identified using an unique key called "**Project ID**". The status of the project can set to any of three statuses such as "1 -To Confirm", "2-In Process" and "3- Completed".



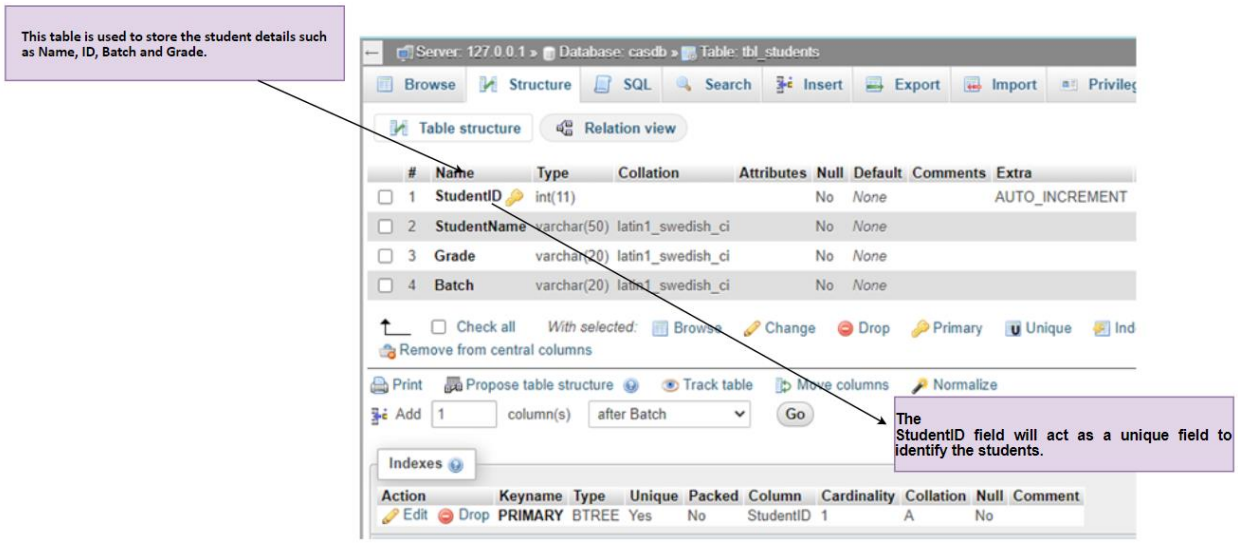
Name: tbl_ProjectStatusLookUp

This table stores the project status with the PStatusID being the lookup value.



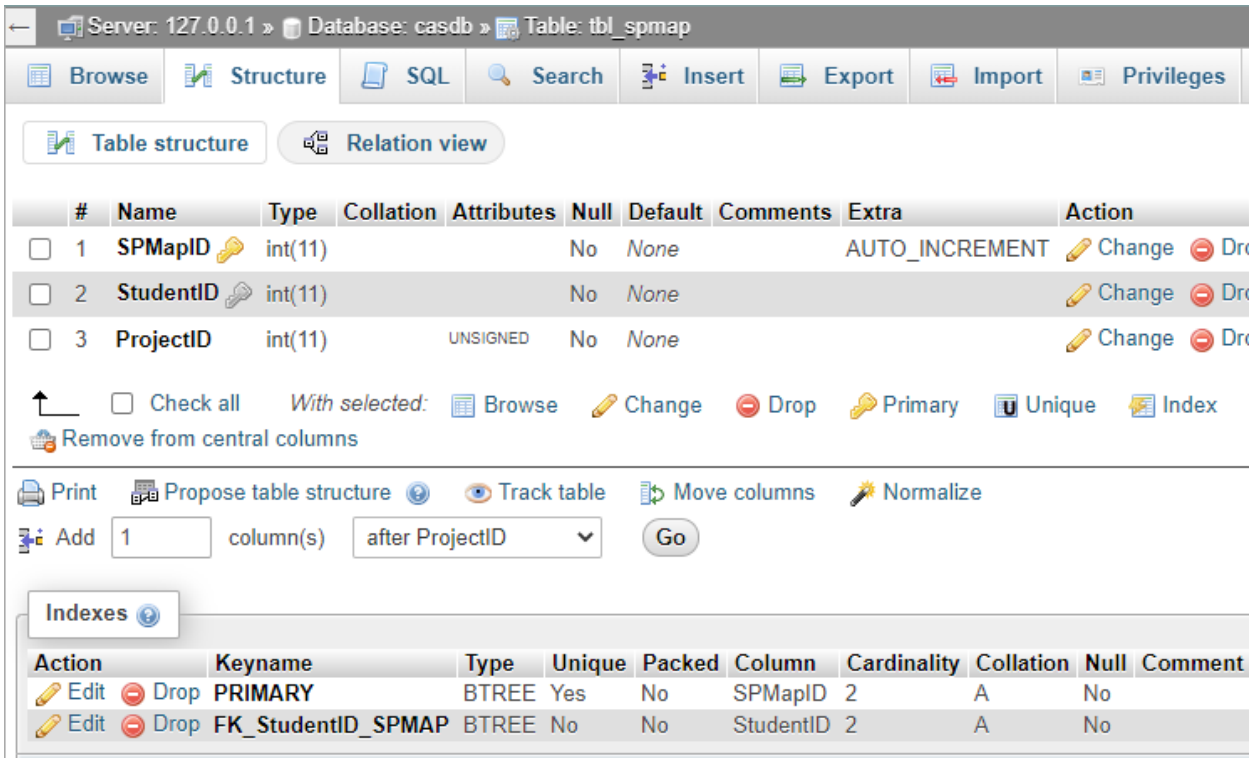
Name: tbl_Students

This table is used to store the student details such as Name, ID, Batch and Grade. The StudentID field will act as a unique field to identify the students.



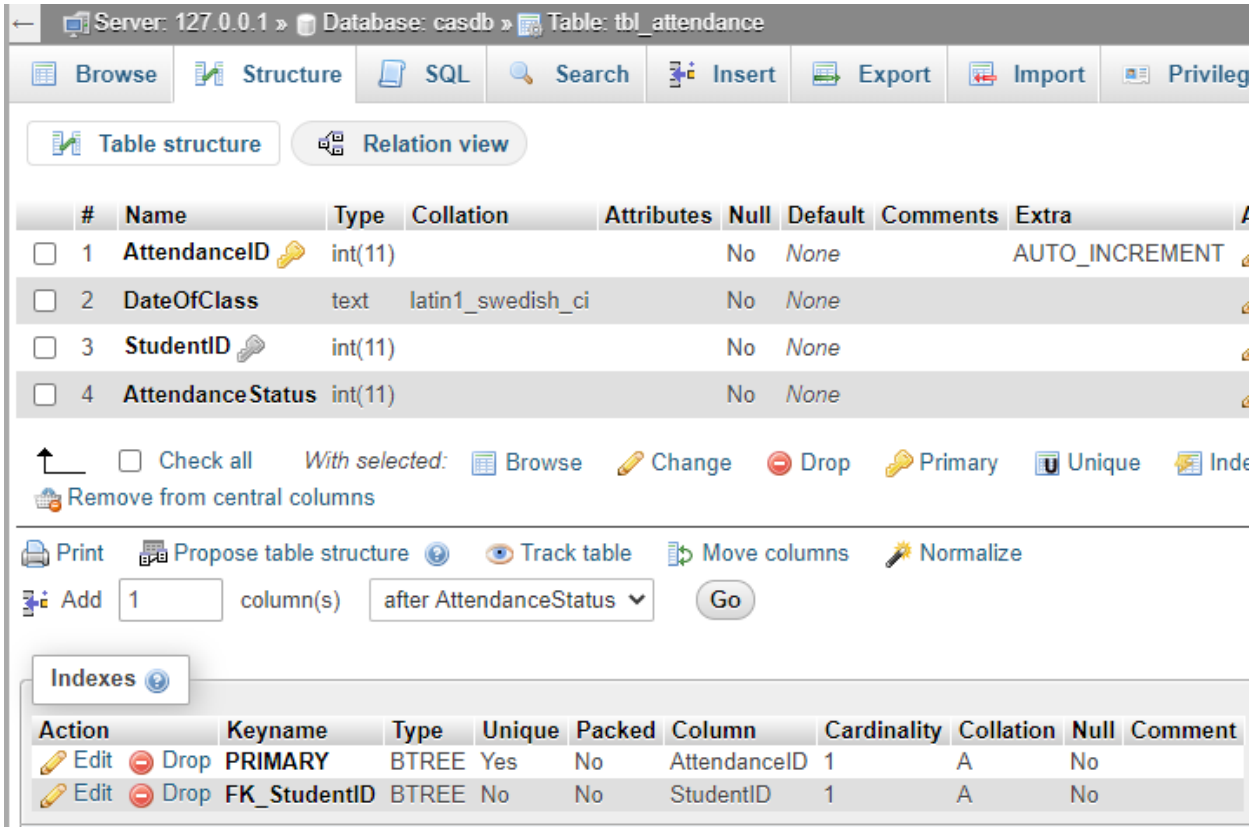
Name: tbl_SPMap

This table is used to map the Projects and Students. The projectID and studentID are the primary keys for the Student and Project table respectively.



Name: tbl_Attendance

This table is used to store the daily attendance record for the students for the CAS class.



Application Development:

For connectivity of Python with MySQL, the following modules to be installed using pip:

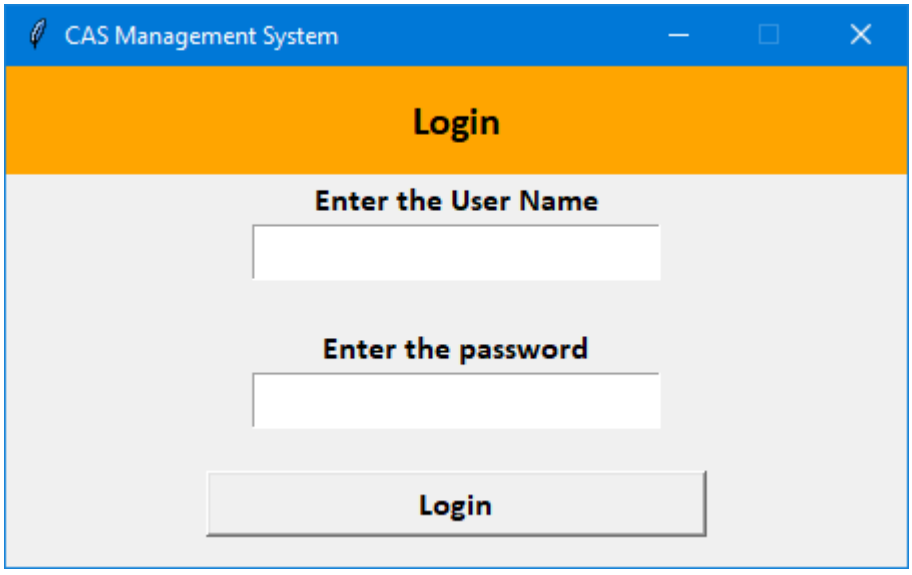
```
pip install mysql-connector-python
```

`Login Window : MainScreen()`

CASMgmt -OOP.py - E:\CASMgmt -OOP.py (3.6.7)

File Edit Format Run Options Window Help

```
def main_account_screen():
  global main_screen
  main_screen = Tk()  # create a GUI window
  main_screen.geometry("450x250") # set the configuration of GUI window
  main_screen.title("CAS Management System") # set the title of GUI window
  main_screen.resizable(height = 0, width = 0)
  global username_verify
  global password_verify
  username_verify = StringVar()
  password_verify = StringVar()

  Label(text="Login", bg="Orange", width="450", height="2", font=("Calibri", 15,"bold")).pack()

  # Set username entry
  Label(text="Enter the User Name", width="450", font=("Calibri", 12,"bold")).pack()
  global username_entry
  username_entry = Entry(main_screen, font=("Calibri", 15,"bold"), textvariable=username_verify)
  username_entry.pack()
  Label(text="").pack()

  # Set password entry
  Label(text="Enter the password", width="450", font=("Calibri", 12,"bold")).pack()
  global password_entry
  password_entry = Entry(main_screen, font=("Calibri", 15,"bold"), textvariable=password_verify,show
  password_entry.pack()
  Label(text="").pack()

  # create Login Button
  Button(text="Login", height="1", width="30",font=("Calibri", 12,"bold"),command=login_verify).pack(
```



Dashboard Window : Dashboard()

CASMgmt -OOP.py - E:\CASMgmt -OOP.py (3.6.7)

File Edit Format Run Options Window Help

```
class Dashboard:
  def __init__(self, master):
    self.master = master
    self.master.geometry("850x500")
    self.master.title("Dashboard - CAS Management System") # set the title of GUI window
    self.master.resizable(height = 0, width = 1000)
    self.project_status_dict = {}
    self.fetch_project_status()
    self.show_dashboard()

  def show_dashboard(self):
    self.frame1 = tk.Frame(self.master)
    # create a Form label
    Label(self.frame1, text="Dashboard", bg="Orange", font=("Calibri", 15,"bold")).grid(row=0, column=
    Label(self.frame1, text="Project List", font=("Calibri", 12,)).grid(row=1, column=0, columnspan=10)
    self.frame1.pack()

    self.frame2 = tk.Frame(self.master)
    cols = ('ProjectID', 'Project Name', 'Start Date','End Date','Remarks','Status','StudentID','SName','Gra
    self.listBox = ttk.Treeview(self.frame2, columns=cols, show='headings',selectmode ='browse')

    for col in cols:
      self.listBox.column(col, width = 10, anchor ='c')
      self.listBox.heading(col, text=col)
    self.listBox.pack(fill='x',expand=True)

    tempList = self.fetch_project_details()
    for i in tempList:
      self.listBox.insert("", "end", values=i)
```

Dashboard - CAS Management System

Dashboard

Project List

ProjectID	Project Name	Start Date	End Date	Remarks	Status	StudentID	SName	Grade	Batch
1	IoT Projects	11	11		To Confirm	1	John	2020	11
1	IoT Projects	11	11		To Confirm	3	Smith Johnsor	2020	15
2	Restful API	11	11	Requires JAVA	In Progress	1	John	2020	11

Manage Projects <== Click this button to open Manage Projects Screen

Manage Students <== Click this button to open Manage Students Screen

Manage Attendance <== Click this button to open Manage Attendance Screen

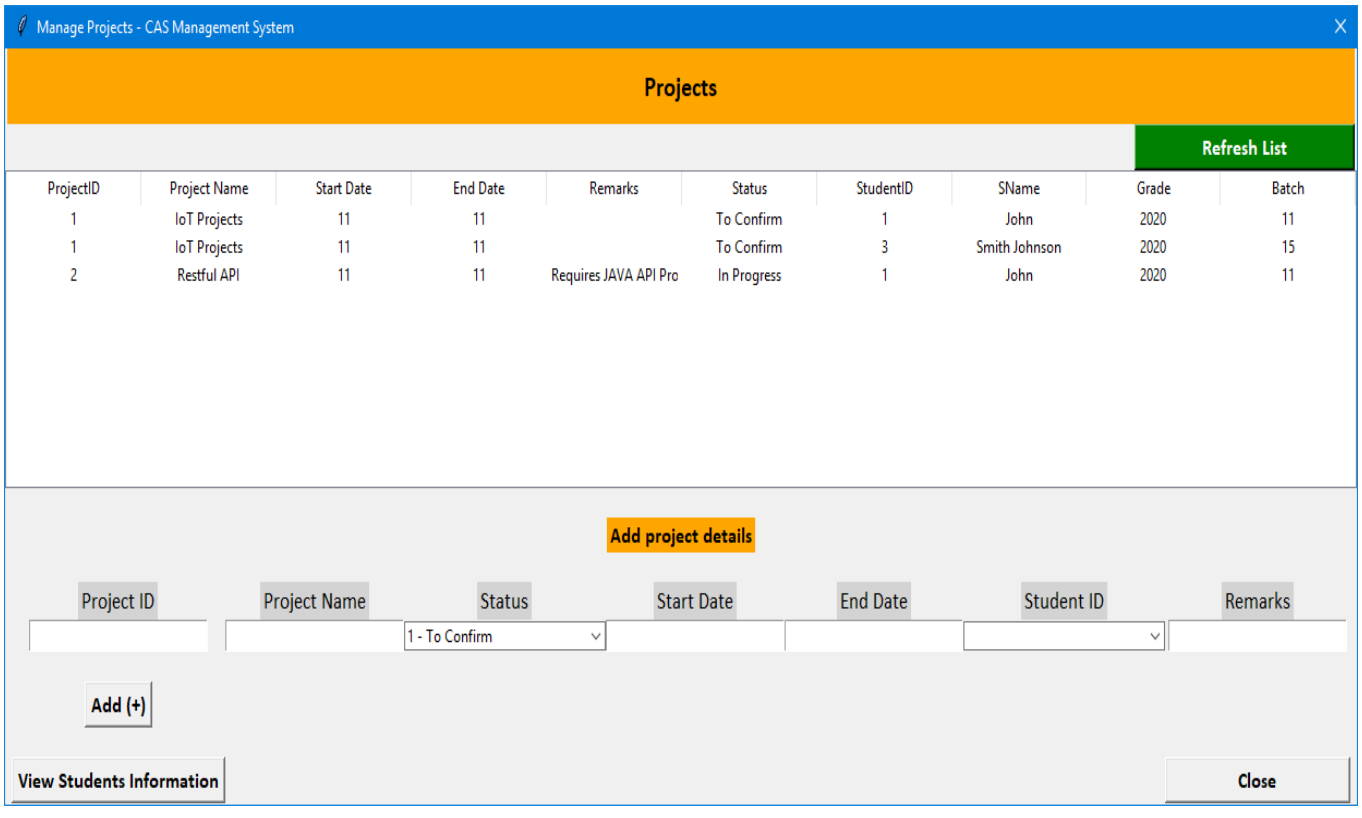
Refresh List <== Click this button to refresh Project List

Project Window : Class Projects()

CASMgmt -OOP.py - E:\CASMgmt -OOP.py (3.6.7)

File Edit Format Run Options Window Help

```
class Projects(Dashboard):
  def __init__(self, master):
    self.master = master
    self.master.title("Manage Projects - CAS Management System") # set the title of GUI window
    self.master.geometry("1240x540")
    self.master.resizable(height = 0, width = 0)
    self.project_status_dict = {}
    self.projectid = StringVar()
    self.projectname = StringVar()
    self.status  = StringVar()
    self.startdate  = StringVar()
    self.enddate  = StringVar()
    self.studentid= StringVar()
    self.remarks = StringVar()
    self.fetch_project_status()
    self.show_projects_window()

  def show_projects_window(self):
    self.frame1 = tk.Frame(self.master)
    # create a Form label
    Label(self.frame1, text="Projects", width=123, height=2, bg="Orange", font=("Calibri", 15,"bold")).g
    refresh_btn = Button(self.frame1, text="Refresh List", width=24, bg='green',fg='white',font=("Calibri'
    self.frame1.pack()

    self.frame2 = tk.Frame(self.master)
    cols = ('ProjectID', 'Project Name', 'Start Date','End Date','Remarks','Status','StudentID','SName','Gra
    self.listBox = ttk.Treeview(self.frame2, columns=cols, show='headings',selectmode ='browse')

    for col in cols:
```

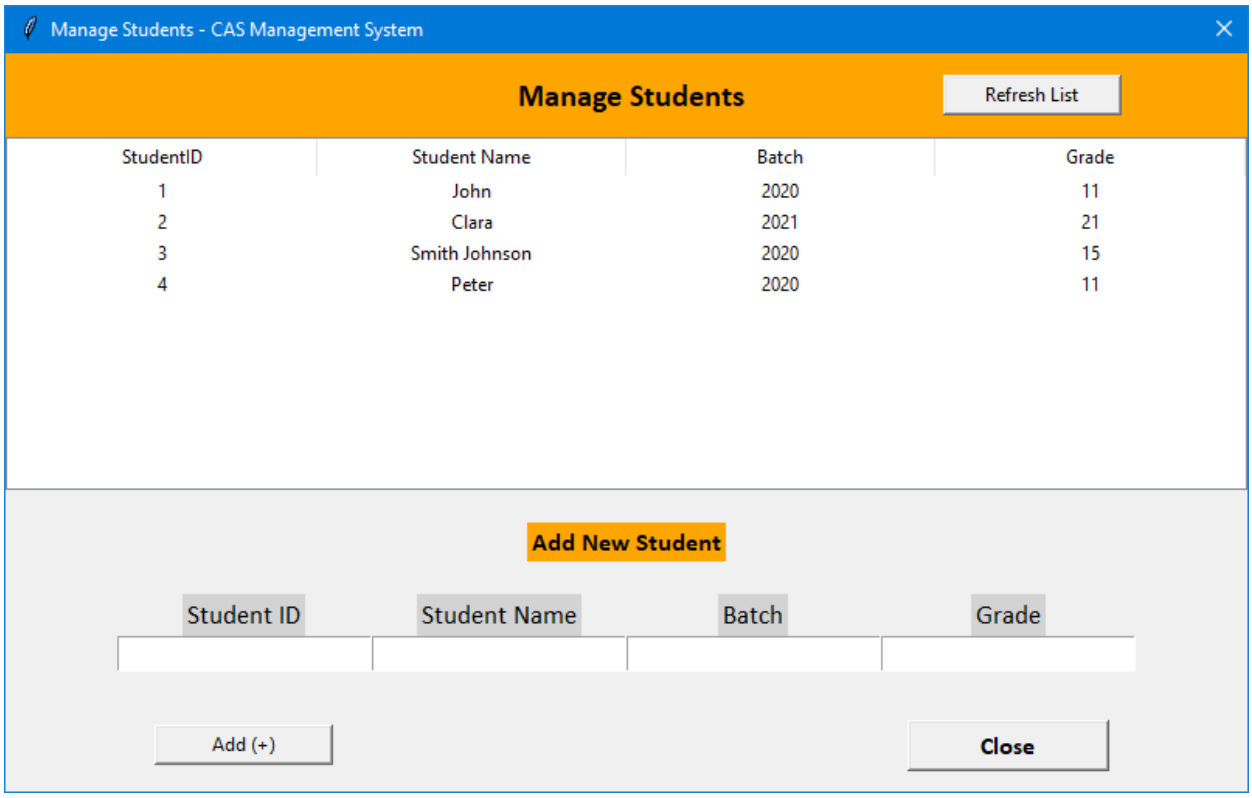


Students Window : Class Students()

CASMgmt -OOP.py - E:\CASMgmt -OOP.py (3.6.7)

File Edit Format Run Options Window Help

```
class Students(Dashboard):
  def __init__(self, master):
    self.master = master
    self.master.title("Manage Students - CAS Management System") # set the title of GUI window
    self.master.geometry("800x475")
    self.master.resizable(height = 0, width = 0)
    self.show_students_window()

  def show_students_window(self):
    self.frame1 = tk.Frame(self.master)
    # create a Form label
    Label(self.frame1, text="Manage Students", width = "80", bg="Orange", height="2", font=("Calibri",
    refresh_btn = Button(self.frame1, text="Refresh List", width=15, command=self.refresh_students_i
    self.frame1.pack()

    self.frame2 = tk.Frame(self.master)
    cols = ('StudentID','Student Name','Batch','Grade')
    self.listBox = ttk.Treeview(self.frame2, columns=cols, show='headings',selectmode ='browse')
    for col in cols:
      self.listBox.column(col, width = 40, anchor ='c')
      self.listBox.heading(col, text=col)

    self.listBox.pack(fill='x',expand=True)
    tempList = self.fetch_student_details()
    for i in tempList:
      self.listBox.insert("", "end", values=i)

#   self.listBox.grid(row=1, column=0, columnspan=4)
    self.frame2.pack(fill='x',expand=True)
```

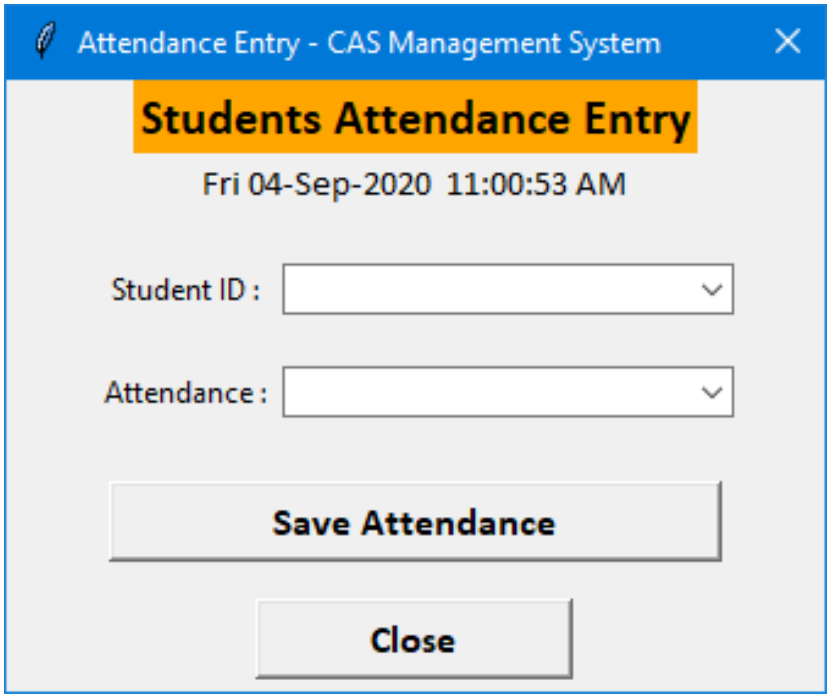


Attendance Window : Class Attendance()

CASMgmt -OOP.py - E:\CASMgmt -OOP.py (3.6.7)

File Edit Format Run Options Window Help

```
class Attendance(Dashboard):
  def __init__(self, master):
    self.master = master
    self.master.title("Window n.2")
    self.master.geometry("335x250")
    self.master.title("Attendance Entry - CAS Management System") # set the title of GUI window
    self.master.resizable(height = 0, width = 0)
    self.studentid = StringVar()
    self.attendance = StringVar()
    self.timestampStr= StringVar()
    self.show_attendance_window()

  def show_attendance_window(self):
    self.frame = tk.Frame(self.master)
    dateTimeObj = datetime.now()
    self.timestampStr = dateTimeObj.strftime("%a %d-%b-%Y  %r")

    # create a Form label
    Label(self.frame, text="Students Attendance Entry", bg="Orange", font=("Calibri", 15,"bold")).grid(r
    Label(self.frame, text=self.timestampStr, font=("Calibri", 11)).grid(row=1, column=0,columnspan=2)

    Label(self.frame, text="").grid(row=2, column=0)
    Label(self.frame, text="Student ID : ").grid(row=3, column=0)

    studentid_combo = ttk.Combobox(self.frame, width = 27,  textvariable = self.studentid ,state="reado
    studentid_combo['values'] = self.fetch_student_detail()
    studentid_combo.grid(row = 3,column = 1)
    studentid_combo.current(0)

    Label(self.frame, text="").grid(row=5, column=0)
```



Bibliography:

1. https://pynative.com/install-mysql-connector-python/
2. https://www.tutorialspoint.com/python/python_gui_programming.htm
3. https://www.geeksforgeeks.org/python-tkinter-tutorial/
4. https://effbot.org/tkinterbook/