COLLEGE MANAGEMENT SYSTEM

By:  
KARTHIKEYAN AS

XII A

NATIONAL PUBLIC SCHOOL GOPALAPURAM

****

**NATIONAL PUBLIC SCHOOL**

**Gopalapuram , Chennai**

**BONAFIDE CERTIFICATE**

Certified to be the bonafide **Project Work** done by

Master/Miss ………………………………………………………….………….

of Class **XII A** in **COMPUTER SCIENCE** during the academic year **2020-2021** at **NATIONAL PUBLIC SCHOOL, CHENNAI- 86.**

**Signature of Internal Examiner:** ………………………………………………

**Date:­­­­­­­­­­­­­­­­** ……………………………………………….

**Submitted for All-India Senior School Certificate Practical Examination.**

**Board Roll number: …………………………………………**

**Date of Examination: ………………………………………..**

**Centre of Examination: ……………………………………..**

**Signature of External Examiner:** **……………………………**

**Date:……………………………..**

**ACKNOWLEDGEMENT**

I gratefully acknowledge my sincere thanks to our Computer Science Teacher for her valuable guidance and supervision throughout the project work. I also thank Madam Principal and the School Management for providing the necessary support to complete the project work.

INDEX

[**1.** **INTRODUCTION** 5](#_Toc59785893)

[**2.** **COLLEGE MANAGEMENT SYSTEM- PROGRAM** 7](#_Toc59785894)

[**3.** **OUTPUTS** 115](#_Toc59785895)

[**4.** **BIBLIOGRAPHY** 116](#_Toc59785896)

# **INTRODUCTION**

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python is often used as a “scripting language” for web applications. This means that it can automate specific series of tasks, making it more efficient. It’s especially great for using on back end web development projects, because Python has pre-built libraries and web frameworks including Pyramid, Django, and Flask, shortening the amount of time you spend on projects by allowing you to repurpose chunks of code. Python is also used for scientific research and computing and even has several science-friendly or science-specific libraries. Consequently, Python is often used in software applications, pages within a web browser, the shells of operating systems and some games.

For many real-life applications, it is necessary to have a database to store data which can be manipulated. For such aspects, Interface of python with SQL Database is very useful. It facilitates us to have multiple separate working environments through the same connection to the database.

The project involves using the interface of python with a SQL database to develop a College Management System. It works on the assumption of having two types of users, Staff and Administrative office. It consists of three main registers: staff register, student register and results register, where the access of the staff register is available only to the Administrative office. The following reports would be generated at the end:

* 1. Details of staff based on Date of Joining
  2. Details of students based on Date of Admission
  3. Details of students based on the Class
  4. Details of students based on the Marks (less than 50, equal to 50, and more than 50)

# **COLLEGE MANAGEMENT SYSTEM - PROGRAM**

import mysql.connector

from columnar import columnar

def menu():#Main menu

print("\t\t||||WELCOME TO COLLEGE MANAGEMENT SYSTEM||||")

while True:#Choosing the designation

print("\t\t\tChoose your designation")

print()

print("\t\t\t1.Admin officer")

print()

print("\t\t\t2.Staff")

print()

print("\t\t\t3.Exit")

print()

try:

ch=int(input("\t\t\tEnter your choice: "))

print()

except:

print()

print("\t\t\tAn Error Has occured")

print()

continue

if ch==1:

print("\t\t\tAdmin officer")

try:

adminid=int(input("\t\t\tEnter admin ID: "))

print()

except:

print()

print("\t\t\tAn error has occured\n")

continue

if searchadminid(adminid)==True:#checking for the existence of admin id in the database using searchadminid function

pwd=getpassword()#Getting the password as user input

print()

if validatepassword(adminid,pwd)==True:#Validating the entered password by comparing it with the password stored in database using the validatepassword function

while True:

print("\t\t\tAdmin Officer Menu")

print()

print("\t\t\t1.Staff Register")

print()

print("\t\t\t2.Student Register")

print()

print("\t\t\t3.Result Register")

print()

print("\t\t\t4.Exit to main menu")

print()

try:

ch1=int(input("\t\t\tEnter the Register Choice: "))

print()

except:

print()

print("\t\t\tAn Error has occured")

print()

break

if ch1==1:

staff()

print()

elif ch1==2:

student()

print()

elif ch1==3:

result()

print()

elif ch1==4:

print()

break

else:

print("\t\t\tInvalid Choice")

print()

else:

print("\t\t\tIncorrect Password")

print()

else:

print("\t\t\tAdmin ID does not exist")

print()

elif ch==2:

print("\t\t\tStaff")

print()

try:

staffid=int(input("\t\t\tEnter staff Id: "))

print()

except:

print()

print("\t\t\tAn error has occured\n")

continue

if searchstaffid(staffid)==True:#checking for the existence of staff id in the database using searchstaffid function

staffpwd=getpassword()

if validatepassword(staffid,staffpwd)==True:#Validating the entered password by comparing it with the password stored in database using the validatepassword function

while True:

print("\t\t\tStaff Menu")

print()

print("\t\t\t1.Student Register")

print()

print("\t\t\t2.Result Register")

print()

print("\t\t\t3.Exit to main menu")

print()

try:

ch2=int(input("\t\t\tEnter the Register choice: "))

print()

except:

print("\t\t\tAn Error Has Occured")

print()

break

if ch2==1:

student()

print()

elif ch2==2:

result()

print()

elif ch2==3:

print()

break

else:

print("\t\t\tInvalid choice")

print()

else:

print("\t\t\tIncorrect password")

print()

elif searchstaffid(staffid)==False:

print("\t\t\tStaff ID does not exist")

print()

elif ch==3:

break

else:

print("\t\t\tInvalid choice")

print()

def searchadminid(adminid):#checking the existence of the adminid entered by the user

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select staffid from staff where jobtitlecode=1")

rec=cursor.fetchall()

flag=False

for i in rec:

if i==(adminid,):

flag=True

mycon.close()

return flag

def searchstaffid(staffids):#checking the existence of the staffid entered by the user

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select staffid from staff where jobtitlecode not in (1)")

rec=cursor.fetchall()

flag=False

for i in rec:

if i==(staffids,):

flag=True

mycon.close()

return flag

def getpassword():#getting password as user input using getpass

import stdiomask#masking the user input and displaying stars

pw=stdiomask.getpass(prompt="\t\t\tEnter Password ",mask="\*")

return pw

def validatepassword(sid,password):#checking if the password entered by the user to login is correct

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

sql="Select password from staff where staffid=%s"%(sid)

cursor.execute(sql)

rec=cursor.fetchall()

flag=False

for i in rec:

if i==(password,):

flag=True

return flag

def createtables():#creating the database and tables

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123")

cursor=mycon.cursor()

cursor.execute("Create database if not exists college")#DatabaseName:College

mycon.close()

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Create table if not exists jobtitles(jobtitlecode int(3) \

not null unique primary key,jobtitle varchar(25))")#Table Name:Jobtitle

#Jobtitle table consits of jobtitles and the respective jobtitle code

cursor.execute("Select \* from jobtitles")

rec=cursor.fetchall()

if len(rec)==0:#Checking for existence of jobtitles.If no jobtitles exists adding 'Admin' as the first job title

cursor.execute("insert into jobtitles values(1,'%s')"%("Admin"))

mycon.commit()

cursor.execute("Create table if not exists department(departmentcode \

int(3) not null unique primary key,department varchar(20))")#Table \

Name:Department

#Department table consists of departments and respective department code

cursor.execute("Create table if not exists subject(subjectcode int(3) \

not null unique primary key,subject varchar(20))")#Table Name:Subject

#Subject table consists of subject and respective subject code

cursor.execute("Create table if not exists staff(name varchar(50) \

not null,staffid int(6) not null unique primary key,gender char(1), \

dateofbirth date,address varchar(100),contactno varchar(10),\

dateofjoining date,qualification varchar(10),departmentcode int(3) \

references department(departmentcode),\

jobtitlecode int(3) not null references jobtitles(jobtitlecode),basicpay \

int(6),password varchar(8) not null)")#Table Name:Staff

#staff table contains staff details

cursor.execute("Create table if not exists student(registrationno int(6) \

not null unique primary key,dateofadmission date not null,rollnumber \

int(6) not null,name varchar(50) not null, gender char(1) not null, \

dateofbirth date not null,address varchar(100) not null,\

parentphonenumber varchar(10) not null,classcode int(3) not null \

references class(classcode),section varchar(5) not null)")#Table Name:Student

#student table contains student details

cursor.execute("Create table if not exists result(registrationno int(6) \

not null references student(resgistrationno),subjectcode int(3) not null \

references subject(subjectcode), markobtained int(3) not null,passmark \

int(3) not null,maximummark int(3) not null,result char(1) not null)") \

#Table Name:Result

#result table contains result details of students

cursor.execute("create table if not exists class(classcode int(3) \

not null unique primary key,class varchar(20))")#Table name:Class

#class table consists of the classes and the respective class code

cursor.execute("Select \* from staff")

rec1=cursor.fetchall()

if len(rec1)==0:#Checking for the existence of records in staff table .If no records exist a default admin with staff id 1 and password

#'admin' is added to staff table which is used for logging in for first time

cursor.execute("insert into staff(name,staffid,jobtitlecode,password)\ values('Default Admin',1,1,'admin')")

mycon.commit()

cursor.execute("Select \* from department")

rec1=cursor.fetchall()

if len(rec1)==0:#Checking for existence of records in department table.If no records exist a default department 'NA'(Not assigned) is added

cursor.execute("insert into department values(1,'NA')")

mycon.commit()

mycon.close()

def staff():#Menu for operations performed in staff register

while True:

print()

print("\t\t\tStaff Register Menu")

print()

print("\t\t\t1.Add a staff")

print()

print("\t\t\t2.Search by ID")

print()

print("\t\t\t3.Search by name")

print()

print("\t\t\t4.Search by contact number")

print()

print("\t\t\t5.Modify address")

print()

print("\t\t\t6.Modify job title")

print()

print("\t\t\t7.Modify department")

print()

print("\t\t\t8.Modify password")

print()

print("\t\t\t9.Modify phone number")

print()

print("\t\t\t10.Add a department")

print()

print("\t\t\t11.Add a jobtitle")

print()

print("\t\t\t12.Staff details based on date of joining")

print()

print("\t\t\t13.Exit")

print()

try:

ch=int(input("\t\t\tEnter your choice: "))

print()

except:

print()

print("\t\t\tAn error has occured")

print()

return

if ch==1:

addstaff()

print()

elif ch==2:

searchbystaffid()

print()

elif ch==3:

searchbystaffname()

print()

elif ch==4:

searchbystaffphno()

print()

elif ch==5:

modifyaddress()

print()

elif ch==6:

modifyjobtitlecode()

print()

elif ch==7:

modifydepartmentcode()

print()

elif ch==8:

modifypassword()

print()

elif ch==9:

modifyphno()

print()

elif ch==10:

addnewdepartment()

print()

elif ch==11:

addnewjobtitle()

print()

elif ch==12:

staffdetailsdoj()

print()

elif ch==13:

print()

break

else:

print("\t\t\tInvalid Choice")

print()

def addstaff():#Adding a staff to the staff table

import datetime

import mysql.connector

print("\t\t\tAdd a staff")

print()

while True:

name=input("\t\t\tEnter Staff Name: ")

print()

f=1

for i in name:

if i in "!@#$%^&\*(){}[]\|\_=+-~`<>,?1234567890?/":#Checking for special characters in the name.Special characters allowed in name are space and full stop.

print("\t\t\tInvalid name.Please re enter name.\n")

f=0

break

if f==1:

break

elif f==0:

continue

while True:

gender=input("\t\t\tEnter Gender(M/F): ")

print()

if gender not in "MF":#Gender can only be 'M'-Male or 'F'-Female

print("\t\t\tInvalid gender.Please re enter gender\n")

continue

else:

break

while True:

dob=input("\t\t\tEnter Date Of Birth.(yyyy-mm-dd): ")

print()

f="%Y-%m-%d"#Default date format

try:

datetime.datetime.strptime(dob,f)#Checking if date is in correct format.If not in correct format value error is raised.

yyyy,mm,dd=dob.split("-")#Obtaining the year,month and date from the date entered by user

yyyy=int(yyyy)

mm=int(mm)

dd=int(dd)

d=str(datetime.date.today())

yyyy1,mm1,dd1=d.split("-")

yyyy1=int(yyyy1)

if mm==1 or mm==3 or mm==5 or mm==7 or mm==8 or mm==10 or mm==12:#getting the number of days in the month

m1=31

elif mm==4 or mm==6 or mm==9 or mm==11:#getting the number of days in the month

m1=30

elif yyyy%4==0 and yyyy%100!=0 or yyyy%400==0:#leap year condition

m1=29

else:#non leap year condition

m1=28

f=1

if mm<1 or mm >12:# checking whether the enterd month is valid 1<=month<=12

f=0

elif dd<1 or dd>m1:#checking whether the date entered is valid .Here m1 is the number of days in the month entered

f=0

elif yyyy<1930:#Only people born after 1930 can join the college

f=0

if dob>=d:

f=0

elif (yyyy1-yyyy)<22:#minimum age of the staff while joining should be 22 years

f=0

if f==0:

print("\t\t\tInvalid date of birth.Please re enter date of birth.\n")

continue

elif f==1:

break

except ValueError:

print("\t\t\tInvalid date of birth.Please re enter date of \

birth.\n")#this message will be printed if the date is in incorrect format

continue

while True:

#conditions for valid address

#minimum length=10

#maximum length=100

address=input("\t\t\tEnter Address.(Less Than 100 characters): ")

print()

if len(address)>=100 :

print("\t\t\tAddress too long.Please Re enter address.\n")

continue

elif len(address)<10:

print("\t\t\tAddress too short.Please Re enter address.\n")

continue

else:

break

while True:

contactno=input("\t\t\tEnter Contact Number: ")

print()

#Conditions for a valid contact number

#should contain 10 digits

if len(contactno)!=10:

print("\t\t\tInvalid contact number.Please Re enter contact number.\n")

continue

f=1

for i in contactno:

if i.isdigit()==False:

f=0

break

if f==0:

print("\t\t\tInvalid contact number.Please Re enter contact \

number.\n")

continue

elif f==1:

break

while True:

qualification=input("\t\t\tEnter Qualification.(In 10 Characters): ")

print()

#conditions for valid qualification

#maximum length=10

#should not contain digits

if len(qualification)>=10:

print("\t\t\tQualification too long.Please re enter qualification.\n")

continue

f=1

for i in qualification:

if i.isdigit():

f=0

break

if f==0:

print("\t\t\tIncorrect format for qualification.Please re enter \

qualification.\n")

continue

elif f==1:

break

while True:

doj=input("\t\t\tEnter Date of Join(yyyy-mm-dd): ")

print()

f="%Y-%m-%d"

try:

datetime.datetime.strptime(doj,f)#The strptime() method creates a datetime object from the given string.

yyyy,mm,dd=doj.split("-")#Date of join

yyyy2,mm2,dd2=dob.split("-")#Date of birth

yyyy2=int(yyyy2)#Birth year

yyyy=int(yyyy)#Join year

mm=int(mm)

dd=int(dd)

d=str(datetime.date.today())#CURRENT DATE

yyyy1,mm1,dd1=d.split("-")

yyyy1=int(yyyy1)#CURRENT YEAR

if mm==1 or mm==3 or mm==5 or mm==7 or mm==8 or \

mm==10 or mm==12:#Deciding the number of days in the month \

entered to check if correct date is entered

m1=31

elif mm==4 or mm==6 or mm==9 or mm==11:

m1=30

elif yyyy%4==0 and yyyy%100!=0 or yyyy%400==0:#Checking\

for leap year to decide number of days in february

m1=29

else:#number of days in a non leap year february

m1=28

f=1

if mm<1 or mm >12:

f=0

elif dd<1 or dd>m1:

f=0

elif yyyy<1900:

f=0

if doj>d:

f=0

elif yyyy-yyyy2<=22:#JOIN YEAR-BIRTH YEAR

f=0

if f==0:

print("\t\t\tInvalid date of join.Please re enter date of join.\n")

continue

elif f==1:

break

except ValueError:#ValueError is raised when date is not in correct format

print("\t\t\tInvalid date of join.Please re enter date of join.\n")

continue

while True:

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select \* from jobtitles")

rec=cursor.fetchall()

print("\t\t\tJobtitle Code","Jobtitle",sep="\t ")

for i in rec:

print(" \t\t\t ",i[0]," \t\t ",i[1])#Displaying choices for jobtitle

print()

try:

jobtitlecode=int(input("\t\t\tEnter Job Title Code: "))

print()

except:

print("\t\t\tInvalid input for Jobtitle code.\n")

continue

cursor.execute("Select jobtitlecode from jobtitles where \

jobtitlecode=%s"%(jobtitlecode))

rec=cursor.fetchall()

if len(rec)!=1:#Checking for existence of jobtitle code entered by \

user

print("\t\t\tJob title code do not exist.\n")

continue

else:

break

while True:

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select \* from department")

rec=cursor.fetchall()

print("\t\t\tDepartment code","Department",sep="\t ")

for i in rec:

print("\t\t\t ",i[0],"\t\t\t ",i[1])#Displaying choices for department

print()

try:

departmentcode=int(input("\t\t\tEnter department code: "))

print()

except:

print("\t\t\tInvalid input for Department code.Please Re enter \

Department code.\n")

continue

cursor.execute("Select departmentcode from department where \

departmentcode=%s"%(departmentcode))#Checking for existence of \

department code entered by user

rec=cursor.fetchall()

if len(rec)!=1:#checking for the existence of department code entered by user

print("\t\t\tDepartment code does not exist.\n")

continue

else:

break

while True:

try:

basicpay=int(input("\t\t\tEnter Basic Pay in INR: "))

print()

except:

print("\t\t\tInvalid input for Basic Pay.Please re enter Basic \

Pay\n")

continue

if basicpay<10000:#minimum basic pay=Rs 10000

print("\t\t\tMinimum Basic Pay is 10000.Please re enter basic \

pay\n")

continue

if type(basicpay)==int:

break

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select max(staffid) from staff")

rec=cursor.fetchall()

staffid=0

for i in rec:

staffid=i[0]+1#Auto generating staff id

while True:

password=getpassword()

print()

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select password from staff where password='%s'"%(password))

rec=cursor.fetchall()

#Conditions for a valid password

#Minimum length=5

#Maximum length=8

#Atleast 1 digit and 1 special character should be there in password

if len(rec)!=0:

print("\t\t\tPassword already exists.Please Re enter Password\n")

continue

if len(password)>8:

print("\t\t\tExceeding Character limit.Only 8 characters \

permitted.Please Re enter password\n")

continue

if len(password)<5:

print("\t\t\tPassword should contain atleast 5 characters.Please \

Re enter password\n")

continue

dc=sc=0

for i in password:

if i.isdigit():

dc+=1

elif i.isalpha():

pass

else:

sc+=1

if dc<1:

print("\t\t\tPassword should contain atleast 1 digit.Please \

Re enter password\n")

continue

elif sc<1:

print("\t\t\tPassword should contain atleast 1 \

special character.Please Re enter password\n")

continue

else:

break

print("\t\t\tStaff ID of",name,"is",staffid)#Displaying the name of \

the staff with staff id generated

print()

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:#Inserting the record into staff table

cursor.execute("Insert into staff \

values('%s',%s,'%s','%s','%s','%s','%s','%s',%s,%s,%s,'%s')"%(name,\

staffid,gender,dob,address,contactno,doj,qualification,departmentcode,\

jobtitlecode,basicpay,password))

mycon.commit()

print("\t\t\tAdded staff Successfully\n")

print()

except:

print("\t\t\tUnable to add to staff table\n")

print()

mycon.rollback()

mycon.close()

def searchbystaffname():#Searching staff by staff name

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

password="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tSearch staff by name\n")

try:

staffname=input("\t\t\tEnter the name of the staff: ")

print()

except :

print()

print("\t\t\tInvalid staff name")

print()

return

f=1

for i in staffname:

if i in "!@#$%^&\*(){}[]\|\_=+-~`<>,?1234567890?/":#Checking for special characters in the name.Special characters allowed in name are space and full stop.

f=0

if f==0:

print("\t\t\tInvalid staff name.\n")

print()

return

try:

cursor.execute("select name,staffid,gender,dateofbirth,address,contactno,dateofjoining,\

qualification,department,jobtitle,basicpay from staff,jobtitles,\

department where name='%s' and \

staff.jobtitlecode=jobtitles.jobtitlecode \

and staff.departmentcode=department.departmentcode"%(staffname))

myrecords=cursor.fetchall()

except:

print()

print("\t\t\tUnable to search staff\n")

return

if len(myrecords)==0:#checking for the existence of staff name entered by user

print()

print("\t\t\tstaff name not found\n")

else:

data=[]#printing the staff details in tabular form

header=["Name","Staff ID","Gender","Date of \

birth","Address","Contact Number","Date of \

joining","Qualification","Department","Job title","Basic Pay(INR)"]

for i in myrecords:

i=list(i)

data.append(i)

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

mycon.close()

def searchbystaffphno():#searching staff by contact number

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tSearch by staff contact number")

print()

phno=input("\t\t\tEnter the contact number: ")

f=1

print()

for i in phno:

if i.isdigit()==False:

f=0

if len(phno)!=10 or f==0:

print("\t\t\tInvalid contact number.\n")

return

else:

try:

cursor.execute("select s.name, s.staffid, s.gender, s.dateofbirth, \

s.address, s.contactno, s.dateofjoining, s.qualification, d.department, \

j.jobtitle, s.basicpay from staff s, department d, jobtitles j where \

s.departmentcode=d.departmentcode and \

s.jobtitlecode=j.jobtitlecode and contactno=%s"%(phno))

d=cursor.fetchall()

r=cursor.rowcount

except:

print("\t\t\tUnable to search the contact number\n")

print()

return

if r==0:#Checking for existence of staff with the given contact number

print("\t\t\tStaff with the contact number does not exist\n.")

print()

return

else:

data=[]#displaying staff details in a tabular form

header=["Name","Staff ID","Gender",\

"Date of birth","Address","Contact Number",\

"Date of joining","Qualification","Department","Job title",\

"Basic Pay(INR)"]

for i in d:

i=list(i)

data.append(i)

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

mycon.close()

def searchbystaffid():#Searching staff by staff id

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tSearch by staff ID\n")

try:

staffid=int(input("\t\t\tEnter the staff ID to search "))

print()

except ValueError:

print()

print("\t\t\tInvalid Input\n")

print()

return

try:

cursor.execute("select name,staffid,gender,dateofbirth,Address,\

contactno,dateofjoining,qualification,department,jobtitle,basicpay from \

Staff,jobtitles,\

department where staffid=%s and staff.jobtitlecode=jobtitles.jobtitlecode and staff.departmentcode=department.departmentcode"%(staffid))

myrecords=cursor.fetchall()

except:

print()

print("\t\t\tUnable to find record\n")

print()

return

if len(myrecords)==0:#Checking for existence of staff with the given staff id

print("\t\t\tStaff ID not found")

else:

data=[]#displaying staff details in a tabular form

header=["Name","Staff ID","Gender","Date of birth","Address",\

"Contact Number","Date of joining","Qualification","Department",\

"Job title","Basic Pay(INR)"]

for i in myrecords:

i=list(i)

data.append(i)

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

mycon.close()

def addnewdepartment():#adding a new department to department table

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tAdd a new department\n")

try:

newdept=input("\t\t\tEnter the new department to be inserted: ")

print()

except:

print()

print("\t\t\tAn error has occured")

print()

return

try:

cursor.execute("Select \* from department where \

department='%s'"%(newdept))#Checking for existence of department entered by user

#New department is added to the department table only if it doesnot exist

except:

print("\t\t\tAn error has occured")

print()

return

rec=cursor.fetchall()

if len(rec)!=0:

print("\t\t\tDepartment already exists")

print()

return

try:

cursor.execute("select max(departmentcode) from department")

except:

print("\t\t\tAn error has occured")

print()

return

a=cursor.fetchall()

deptcode=0

for i in a:#auto generating department code

deptcode=i[0]+1

try:

cursor.execute("insert into department \

values(%s,'%s')"%(deptcode,newdept))

mycon.commit()

print("\t\t\tDepartment code for",newdept,"is",deptcode)

print("\t\t\tNew department inserted successfully")

print()

except:

print("\t\t\tUnable to add department")

print()

mycon.rollback()

mycon.close()

def addnewjobtitle():#adding a new jobtitle to jobtitles table

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tAdd a new jobtitle\n")

try:

newjobtitle=input("\t\t\tEnter the new jobtitle to be inserted: ")

print()

except:

print()

print("\t\t\tAn error has occured")

print()

return

try:

cursor.execute("Select \* from jobtitles where \

jobtitle='%s'"%(newjobtitle))#Checking for existence of jobtitle entered by user

#New jobtitle is added to the jobtitle table only if it doesnot exist

except:

print("\t\t\tAn error has occured")

print()

return

rec=cursor.fetchall()

if len(rec)!=0:

print("\t\t\tJobtitle already exists")

print()

return

try:

cursor.execute("select max(jobtitlecode) from jobtitles")

except:

print("\t\t\tAn error has occured")

print()

return

a=cursor.fetchall()

jtcode=0

for i in a:

jtcode=i[0]+1#Auto generating jobtitle code

try:

cursor.execute("insert into jobtitles \

values(%s,'%s')"%(jtcode,newjobtitle))

mycon.commit()

print("\t\t\tJob title code for",newjobtitle,"is",jtcode)

print("\t\t\tNew jobtitle inserted successfully")

print()

except:

print("\t\t\tUnable to insert jobtitle")

print()

mycon.rollback()

mycon.close()

def staffdetailsdoj():#Generating a report based on date of joining of staffs

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tReport generation based on date of joining of staff\n")

try:

cursor.execute("Select staffid,name,dateofjoining from staff where \

staffid!=1 order by dateofjoining asc")

rec=cursor.fetchall()

except:

print()

print("\t\t\tUnable to display report\n")

return

if len(rec)==0:#when staff table does not contain any records except the default admin

print()

print("\t\t\tNo records in staff table.\n")

return

else:

data=[]

header=['Staff ID','Name','Date of Joining']

for i in rec:

i=list(i)

data.append(i)

table=columnar(data,header,justify="c")

print(table)

def modifyaddress():#Modify address

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tModify Address\n")

try:

staffid=int(input("\t\t\tEnter the staff ID to modify address: "))

print()

except:

print()

print('\t\t\tAn error has occured')

print()

return

cursor.execute("select \* from staff where staffid=%s and staffid!=1"%(staffid))

rec=cursor.fetchall()

if len(rec)==0:#Checking the existence of staff ID entered by user

print("\t\t\tStaff ID does not exist")

print()

return

add=input("\t\t\tEnter new address: ")

if len(add)>100:

print("\t\t\tAddress too long")

print()

return

sql="Update staff set address=%s where staffid=%s";

value=(add,staffid)

try:

cursor.execute(sql,value)

mycon.commit()

print('\t\t\tAddress modified successfully')

print()

except:

print()

print('\t\t\tUnable to modify address')

print()

mycon.rollback()

mycon.close()

def modifyjobtitlecode():#Modify jobtitle

import mysql.connector

from columnar import columnar

print("\t\t\tModify Job title\n")

try:

staffid=int(input("\t\t\tEnter staff ID to modify job title: "))

print()

except :

print()

print("\t\t\tAn error has occured")

print()

return

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("select staffid from staff where staffid=%s and staffid!=1"%(staffid))

rec=cursor.fetchall()

if len(rec)!=1:#Checking the existence of staff ID entered by user

print("\t\t\tStaff ID does not exist")

print()

return

cursor.execute('select \* from jobtitles')

rec=cursor.fetchall()

data=[]

header=[]

for i in rec:

i=list(i)

data.append(i)

header=["Jobtitle code","Jobtitle"]

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

try:

jobtitlecode=int(input("\t\t\tEnter new job title code: "))

print()

except:

print()

print("\t\t\tAn error has occured")

return

cursor.execute("select jobtitlecode from jobtitles where \

jobtitlecode=%s"%(jobtitlecode))

rec=cursor.fetchall()

if len(rec)!=1:

print("\t\t\tJob title code does not exist")

return

try:

cursor.execute("update staff set jobtitlecode=%s where \

staffid=%s"%(jobtitlecode,staffid))

mycon.commit()

print("\t\t\tJobtitle Modified successfully")

print()

except:

print("\t\t\tUnable to modify jobtitle")

print()

mycon.rollback()

mycon.close()

def modifydepartmentcode():#Modify department

import mysql.connector

from columnar import columnar

print("\t\t\tModify Department\n")

try:

staffid=int(input("\t\t\tEnter staff ID to modify department: "))

print()

except:

print()

print("\t\t\tAn error has occured")

print()

return

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("select staffid from staff where staffid=%s and \

staffid!=1"%(staffid))

rec=cursor.fetchall()

if len(rec)!=1:#Checking the existence of staff ID entered by user

print("\t\t\tStaff ID does not exist")

return

cursor.execute('select \* from department')

rec=cursor.fetchall()

data=[]

header=[]

for i in rec:

i=list(i)

data.append(i)

header=["Department Code","Department"]

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

try:

departmentcode=int(input("\t\t\tEnter new department code: "))

print()

except ValueError:

print()

print("\t\t\tAn error has occured")

print()

return

cursor.execute("select departmentcode from department where \

departmentcode=%s"%(departmentcode))#Checking the existence of department code entered by user

rec=cursor.fetchall()

if len(rec)!=1:

print("\t\t\tDepartment code does not exist")

return

try:

cursor.execute("update staff set departmentcode=%s where \

staffid=%s"%(departmentcode,staffid))

mycon.commit()

print("\t\t\tDepartment modified successfully")

except:

print("\t\t\tUnable to modify department")

mycon.rollback()

mycon.close()

def modifypassword():#Modify password

import mysql.connector

import stdiomask

print("\t\t\tModify password\n")

try:

staffid=int(input("\t\t\tEnter Staff ID to modify password: "))

print()

except:

print()

print("\t\t\tAn error has occured")

print()

return

passwordold=stdiomask.getpass(prompt="\t\t\tEnter Old Password: \

",mask="\*")

print()

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select staffid,password from staff where \

staffid=%s and password='%s' and staffid!=1"%(staffid,passwordold))

rec=cursor.fetchall()

#Checking the validity of staff ID and password entered by user

if len(rec)!=1:

print("\t\t\tIncorrect staff ID or Password\n")

print()

return

newpassword=stdiomask.getpass(prompt="\t\t\tEnter New Password: \

",mask="\*")

print()

cursor.execute("Select password from staff where \

password='%s'"%(newpassword))

rec=cursor.fetchall()

if len(rec)!=0:

print("\t\t\tPassword cannot be used.")

print()

return

if len(newpassword)>8:#Checking if the new password satisfies all the conditions for a valid password

print("\t\t\tExceeding Character limit.Only 8 characters permitted.")

print()

return

if len(newpassword)<5:

print("\t\t\tPassword should contain atleast 5 characters.")

print()

return

if newpassword==passwordold:

print("\t\t\tSame as old password.")

print()

return

dc=sc=0

for i in newpassword:

if i.isdigit():

dc+=1

elif i.isalpha():

pass

else:

sc+=1

if dc<1:

print("\t\t\tPassword should contain atleast 1 digit.\n")

return

elif sc<1:

print("\t\t\tPassword should contain atleast 1 special character.\n")

return

try:

cursor.execute("Update staff set password='%s' where \

staffid=%s and password='%s'"%(newpassword,staffid,passwordold))

print("\t\t\tPassword modified successfully")

print()

mycon.commit()

except:

print("\t\t\tUnable to modify password.")

print()

mycon.rollback()

mycon.close()

def modifyphno():#modify phone number

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tModify contact number\n")

try:

staffid=int(input("\t\t\tEnter the staff ID to modify contact number: \

"))

print()

except:

print()

print('\t\t\tAn error has occured')

print()

return

cursor.execute("select \* from staff where staffid=%s and \

staffid!=1"%(staffid))

rec=cursor.fetchall()

if len(rec)==0:#Checking for the existence of staff id entered by user

print("\t\t\tStaff ID does not exist")

print()

return

nphno=input("\t\t\tEnter new contact number: ")

if len(nphno)!=10:

print("\t\t\tInvalid contact number")

print()

return

f=1

for i in nphno:

if i.isdigit()==False:

f=0

break

if f==0:

print("\t\t\tInvalid contact number.\n")

return

sql="Update staff set contactno=%s where staffid=%s";

value=(nphno,staffid)

try:

cursor.execute(sql,value)

mycon.commit()

print('\t\t\tContact number modified successfully')

print()

except:

print()

print('\t\t\tUnable to modify contact number')

print()

mycon.rollback()

def addstudent():#Adding a new student to student table

import datetime

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select\*from class")

rec=cursor.fetchall()

if len(rec)==0:#Checking if a class exists in the class table before adding a student

print("\t\t\tAdd a class before adding student\n")

return

print("\t\t\tAdd a student\n")

print()

while True:

name=input("\t\t\tEnter student Name: ")

print()

f=1

for i in name:

if i in "!@#$%^&\*(){}[]\|\_=+-~`<>,?1234567890?/":#Checking for special characters in the name.Special characters allowed in name are space and full stop.

print("\t\t\tInvalid name.Please re enter name\n")

f=0

break

if f==1:

break

elif f==0:

continue

while True:

try:

rollno=int(input("\t\t\tEnter roll number: "))

print()

except:

print()

print("\t\t\tAn Error has occured\n")

continue

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select rollnumber from student where \

rollnumber=%s"%(rollno))

rec=cursor.fetchall()

if len(rec)==1:#Checking for existence of roll number entered by user

print("\t\t\tRoll number already exists.Please re enter roll \

number\n")

continue

else:

break

while True:

gender=input("\t\t\tEnter Gender(M/F): ")

print()

if gender not in "MF":#Gender can only be 'M'-Male or 'F'-Female

print("\t\t\tInvalid gender.Please re enter gender\n")

continue

else:

break

while True:

dob=input("\t\t\tEnter Date Of Birth.(yyyy-mm-dd): ")

print()

f="%Y-%m-%d"

try:

datetime.datetime.strptime(dob,f)

yyyy,mm,dd=dob.split("-")

yyyy=int(yyyy)

mm=int(mm)

dd=int(dd)

d=str(datetime.date.today())

yyyy1,mm1,dd1=d.split("-")

yyyy1=int(yyyy1)

if mm==1 or mm==3 or mm==5 or mm==7 or mm==8 or \

mm==10 or mm==12:

m1=31

elif mm==4 or mm==6 or mm==9 or mm==11:

m1=30

elif yyyy%4==0 and yyyy%100!=0 or yyyy%400==0:

m1=29

else:

m1=28

f=1

if mm<1 or mm >12:

f=0

elif dd<1 or dd>m1:

f=0

elif yyyy<1900:

f=0

if dob>=d:

f=0

elif (yyyy1-yyyy)<18:

f=0

if f==0:

print("\t\t\tInvalid date of birth.Please re enter date of birth.\n")

continue

elif f==1:

break

except ValueError:

print("\t\t\tInvalid date of birth.Please re enter date of birth.\n")

continue

while True:

address=input("\t\t\tEnter Address.(Less Than 100 characters): ")

print()

if len(address)>=100:

print("\t\t\tAddress too long.Please Re enter address.\n")

continue

if len(address)<10:

print("\t\t\tAddress too short.Please Re enter address.\n")

continue

else:

break

while True:

contactno=input("\t\t\tEnter Parent Contact Number: ")

print()

if len(contactno)!=10:

print("\t\t\tInvalid contact number.Please Re enter contact \

number.\n")

continue

f=1

for i in contactno:

if i.isdigit()==False:

f=0

break

if f==0:

print("\t\t\tInvalid contact number.Please Re enter contact \

number.\n")

continue

elif f==1:

break

while True:

doj=input("\t\t\tEnter Date of admission(yyyy-mm-dd): ")

print()

f="%Y-%m-%d"

try:

datetime.datetime.strptime(doj,f)

yyyy,mm,dd=doj.split("-")#DATE OF JOIN

yyyy2,mm2,dd2=dob.split("-")#DATE OF BIRTH

yyyy2=int(yyyy2)#BIRTH YEAR

yyyy=int(yyyy)#JOIN YEAR

mm=int(mm)

dd=int(dd)

d=str(datetime.date.today())#CURRENT DATE

yyyy1,mm1,dd1=d.split("-")

yyyy1=int(yyyy1)#CURRENT YEAR

if mm==1 or mm==3 or mm==5 or mm==7 or mm==8 or \

mm==10 or mm==12:

m1=31

elif mm==4 or mm==6 or mm==9 or mm==11:

m1=30

elif yyyy%4==0 and yyyy%100!=0 or yyyy%400==0:

m1=29

else:

m1=28

f=1

if mm<1 or mm >12:

f=0

elif dd<1 or dd>m1:

f=0

elif yyyy<1900:

f=0

if doj>d:

f=0

elif yyyy-yyyy2<18:#Minimum age of student=18

f=0

if f==0:

print("\t\t\tInvalid date of join.Please re enter date of join.\n")

continue

elif f==1:

break

except ValueError:

print("\t\t\tInvalid date of join.Please re enter date of join.\n")

continue

while True:

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select \* from class")

rec=cursor.fetchall()

print("\t\t\tClass Code","Class",sep="\t ")#Displaying choice for class

for i in rec:

print(" \t\t\t ",i[0]," \t\t ",i[1])

print()

try:

classcode=int(input("\t\t\tEnter Class Code: "))

print()

except:

print("\t\t\tInvalid input for Class code.\n")

continue

cursor.execute("Select classcode from class where \

classcode=%s"%(classcode))#Checking for existence of class code \

entered by the user in the class table

rec=cursor.fetchall()

if len(rec)!=1:

print("\t\t\t Class code do not exist.\n")

continue

else:

break

while True:

print("\t\t\tSection-year\n")

print("\t\t\tA-1st year\n")

print("\t\t\tB-2nd year\n")

print("\t\t\tC-3rd year\n")

print("\t\t\tD-4th year\n")

sec=input("\t\t\tEnter Section: ")

if sec not in "ABCD":

print("\t\t\tInvalid Section.Please reenter section\n")

continue

else:

break

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select max(registrationno) from student")

rec=cursor.fetchall()

regno=0#Auto generating registration number

for i in rec:

if i[0]==None:#When no records exists in student table

regno=1

else:

regno=i[0]+1

print("\t\t\tRegistration number of",name,"is",regno)#Displaying student name with the autogenerated registration number

print()

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:

cursor.execute("insert into student \

values(%s,'%s',%s,'%s','%s','%s','%s','%s',%s,'%s')"%(regno,doj,rollno,\

name,gender,dob,address,contactno,classcode,sec))

mycon.commit()

print("\t\t\tAdded student Successfully\n")

print()

except:

print("\t\t\tUnable to add to student table\n")

print()

mycon.rollback()

mycon.close()

def modifystudentaddress():#Modify student address

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tModify Address\n")

try:

registrationno=int(input("\t\t\tEnter the Registration Number to \

modify address: "))

print()

except:

print()

print('\t\t\tAn error has occured')

print()

return

cursor.execute("select \* from student where \

registrationno=%s"%(registrationno))

rec=cursor.fetchall()

if len(rec)==0:#Checking existence of registration number entered by user

print("\t\t\tRegistration Number does not exist")

print()

return

add=input("\t\t\tEnter new Address: ")

if len(add)>100:

print("\t\t\tAddress too long.Please re enter address.")

print()

return

if len(add)<10:

print("\t\t\tAddress too short.Please re enter address.")

print()

return

sql="Update student set address=%s where registrationno=%s";

value=(add,registrationno)

try:

cursor.execute(sql,value)

mycon.commit()

print('\t\t\tAddress modified successfully')

print()

except:

print()

print('\t\t\tUnable to modify address')

print()

mycon.rollback()

mycon.close()

def modifyparentphno():#Modify parent phone number

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

#print(mycon.is\_connected()

print("\t\t\tModify parent phone number\n")

try:

registrationnumber=int(input("\t\t\tEnter the registration number to \

modify parent phone number: "))

print()

except:

print()

print('\t\t\tAn error has occured')

print()

return

cursor.execute("select \* from student where \

registrationno=%s"%(registrationnumber))

rec=cursor.fetchall()

if len(rec)==0:#Checking existence of registration number entered by user

print("\t\t\tRegistration number does not exist")

print()

return

npphno=input("\t\t\tEnter new parent phone number: ")

if len(npphno)!=10:

print("\t\t\tInvalid contact number")

print()

return

f=1

for i in npphno:

if i.isdigit()==False:

f=0

break

if f==0:

print("\t\t\tInvalid parent phone number.\n")

return

sql="Update student set parentphonenumber=%s where \

registrationno=%s";

value=(npphno,registrationnumber)

try:

cursor.execute(sql,value)

mycon.commit()

print('\t\t\tParent phone number modified successfully')

print()

except:

print()

print('\t\t\tUnable to modify parent phone number')

print()

mycon.rollback()

mycon.close()

def searchbyregistno():#Search by registration number

print("\t\t\tSearch by registration number\n")

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user='root',\

passwd="sql123",database="college")

cursor=mycon.cursor()

from columnar import columnar

try:

registno=int(input("\t\t\tEnter reistration number to search for: "))

print()

except:

print()

print("\t\t\tInvalid Input\n")

print()

return

try:

cursor.execute("Select registrationno,dateofadmission,rollnumber,name,gender,dateofbirth,\

address,parentphonenumber,class,section from student,class where \

student.classcode=class.classcode and registrationno=%s"%(registno))

rec=cursor.fetchall()

except:

print()

print("\t\t\tUnable to search for registration number\n")

return

if len(rec)==0:#Checking for existence of registration number entered by user

print("\t\t\tRegistration number does not exists\n")

print()

return

else:

data=[]#Displaying student details in tabular form

header=["Registration number","Date of admission","Roll \

number","Name","Gender","Date of birth","Address","Parent contact \

number","Class","Section"]

for i in rec:

i=list(i)

data.append(i)

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

def searchbyrollno():#Search by roll number

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user='root',\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tSearch by roll number\n")

from columnar import columnar

try:

rollno=int(input("\t\t\tEnter roll number to search for: "))

print()

except:

print()

print("\t\t\tInvalid Input\n")

print()

return

try:

cursor.execute("Select registrationno,dateofadmission,rollnumber,name,gender,dateofbirth,\

adderss,parentphonenumber,class,section from student,class where \ student.classcode=class.classcode and rollnumber=%s"%(rollno))

rec=cursor.fetchall()

except:

print()

print("\t\t\tUnable to search for roll number\n")

return

if len(rec)==0:#Checking for existence of roll number entered by user

print("\t\t\tRoll number does not exists\n")

print()

return

else:

data=[]#Displaying student details in tabular form

header=["Registration number","Date of admission","Roll \

number","Name","Gender","Date of birth","Address","Parent contact number","Class","Section"]

for i in rec:

i=list(i)

data.append(i)

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

mycon.close()

def searchbyparentphno():#Search by parent phone number

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tSearch by parent phone number\n")

phno=input("\t\t\tEnter the parent contact number: ")

f=1

print()

for i in phno:

if i.isdigit()==False:

f=0

if len(phno)!=10 or f==0:

print("\t\t\tInvalid contact number.\n")

return

else:

try:

cursor.execute("Select \

registrationno,dateofadmission,rollnumber,name,gender,dateofbirth,\

address,parentphonenumber,class,section from student,class where \

student.classcode=class.classcode and \

parentphonenumber='%s'"%(phno))

d=cursor.fetchall()

except:

print("\t\t\tUnable to search the contact number\n")

return

if len(d)==0:#Checking for existence of phone number entered by user

print("\t\t\tStudent with the contact number does not exist\n.")

return

else:

data=[]#Displaying student detils in tabular form

header=["Registration number","Date of admission","Roll \

Number","Name","Gender","Date Of birth","Address","Parent contact \

number","Class","Section"]

for i in d:

i=list(i)

data.append(i)

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

mycon.close()

def searchbystudentname():#Search by student name

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

password="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tSearch by student name\n")

print()

try:

studentname=input("\t\t\tEnter the Name of The student: ")

print()

except :

print("\t\t\tInvalid student name\n")

return

f=1

for i in studentname:

if i in "!@#$%^&\*(){}[]\|\_=+-~`<>,?1234567890?/":#Checking for special characters in the name.Special characters allowed in name are space and full stop.

f=0

if f==0:

print("\t\t\tInvalid student name.\n")

return

try:

cursor.execute("Select registrationno,dateofadmission,rollnumber,name,gender,dateofbirth,\

address,parentphonenumber,class,section from student,class where \

student.classcode=class.classcode and name='%s'"%(studentname))

myrecords=cursor.fetchall()

except:

print("\t\t\tUnable to search student\n")

return

if len(myrecords)==0:#Checking for existence of student name entered by user

print("\t\t\tStudent name not found")

else:

data=[]#Displaying student details in tabular form

header=["Registration Number","Date of admission","Roll \

number","Name","Gender","Date of birth","Address","Parent Contact \

Number","Class","Section"]

for i in myrecords:

i=list(i)

data.append(i)

table=columnar(data,header,max\_column\_width=30,\

min\_column\_width=5,terminal\_width=500,justify='c')

print(table)

mycon.close()

def studentdetailsdoj():#Report based on date of joining of students

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:

cursor.execute("Select \

registrationno,rollnumber,name,dateofadmission,class from student, \

class where student.classcode=class.classcode order by dateofadmission \

asc")

rec=cursor.fetchall()

except:

print("\t\t\tUnable to display report\n")

return

if len(rec)==0:#Checking for existence of records in student table

print("\t\t\tNo records in student table.\n")

return

else:

data=[]#Displaying student details in tabular form

header=['Registration number','Roll number','Name','Date of admission','Class']

for i in rec:

i=list(i)

data.append(i)

table=columnar(data,header,justify="c")

print("\t\t\tReport based on date of admission\n")

print(table)

def addnewclass():#Adding a new class

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:

classes=input("\t\t\tEnter the new class to be inserted")

except:

print("\t\t\tAn error has occured")

print("\n\n\n")

return

for i in classes:

if i.isdigit():

print("\t\t\tInvalid Input\n")

return

try:

cursor.execute("Select \* from class where class='%s'"%(classes))

except:

print("\t\t\tAn error has occured")

print("\n\n\n")

return

rec=cursor.fetchall()

if len(rec)==1:#Checking for existence of class entered by user

print("\t\t\tClass already exists")

return

try:

cursor.execute("select max(classcode) from class")

except:

print("\t\t\tAn error has occured")

print("\n\n\n")

return

a=cursor.fetchall()

ccode=0

for i in a:#Auto generating class code

if i[0]==None:

ccode=1

else:

ccode=i[0]+1

try:

cursor.execute("insert into class values(%s,'%s')"%(ccode,classes))

mycon.commit()

print("\t\t\tClass code for",classes,"is",ccode)

print("\t\t\tNew class inserted successfully")

print("\n\n\n")

except:

print("\t\t\tUnable to add class")

print("\n\n\n")

mycon.rollback()

def student():#Menu for operations performed in student register

while True:

print("\t\t\tStudent Menu\n")

print("\t\t\t1.Add a student\n")

print("\t\t\t2.Add a class\n")

print("\t\t\t3.Modify address\n")

print("\t\t\t4.Modify parent contact number\n")

print("\t\t\t5.Search by registration number\n")

print("\t\t\t6.Search by roll number\n")

print("\t\t\t7.Search by student name\n")

print("\t\t\t8.Search by phone number\n")

print("\t\t\t9.Display details based on date of admission\n")

print("\t\t\t10.Display details based on class\n")

print("\t\t\t11.Exit\n")

try:

ch=int(input("\t\t\tEnter choice: "))

except:

print("\t\t\tAn Error has occured\n")

return

if ch==1:

addstudent()

elif ch==2:

addnewclass()

elif ch==3:

modifystudentaddress()

elif ch==4:

modifyparentphno()

elif ch==5:

searchbyregistno()

elif ch==6:

searchbyrollno()

elif ch==7:

searchbystudentname()

elif ch==8:

searchbyparentphno()

elif ch==9:

studentdetailsdoj()

elif ch==10:

reportbasedonclass()

elif ch==11:

break

else:

print("\t\t\tInvalid choice\n")

print("\n\n\n")

def reportbasedonclass():#Report based on class

import mysql.connector

from columnar import columnar

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:

cursor.execute("Select name,rollnumber,registrationno,class \

from student,class where student.classcode=class.classcode \

order by student.classcode")

rec=cursor.fetchall()

except:

print("\t\t\tUnable to generate report\n")

return

if len(rec)==0:

print("\t\t\tNo records in student table\n")

return

else:#Displaying studentdetails in tabular form

data=[]

header=['Name','Roll number','Registration number','Class']

for i in rec:

i=list(i)

data.append(i)

table=columnar(data,header,justify="c")

print("\t\t\tReport based on class\n")

print(table)

def addresult():#Adding a result to result table

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

Passwd="sql123",database="college")

cursor=mycon.cursor()

cursor.execute("Select\*from subject")

rec=cursor.fetchall()

if len(rec)==0:#Checking for existence of subject in subject table

print("\t\t\tAdd a subject before adding result\n")

return

while True:

try:

regno=int(input("\t\t\tEnter Registration number: "))

print()

except:

print()

print("\t\t\tInvalid registration number please re enter \

registration number")

print()

continue

cursor.execute("select \* from student where \

registrationno=%s"%(regno))#Checking for existence of registration number enterd by user

rec=cursor.fetchall()

if len(rec)==0:

print("\t\t\tRegistration number does not exist.Please re enter \

registration number\n")

continue

else:

break

while True:

cursor.execute("Select \* from subject")

rec=cursor.fetchall()

print("\t\t\tSubject Code\t\t","Subject")

for i in rec:

print("\t\t\t",i[0],"\t\t",i[1])

try:

subcode=int(input("\t\t\tEnter subject code: "))

print()

except:

print()

print("\t\t\tInvalid subject code.Please re enter subject code")

print()

continue

cursor.execute("Select \* from subject where \

subjectcode=%s"%(subcode))#Checking for existence of subject code entered by user

rec=cursor.fetchall()

if len(rec)==0:

print("\t\t\tSubject code does not exist.Please Re enter subject code\n")

continue

else:

break

while True:

try:

maxmark=int(input("\t\t\tËnter maximum mark: "))

print()

except:

print()

print("Invalid input for maximum mark.Please re enter \

maximum mark\n")

continue

if maxmark<=0:#Maximum mark cannot be less than or equal to 0

print("Maximum mark cannot be less than or equal to zero.Please re enter maximum mark\n")

continue

else:

break

while True:

try:

passmark=int(input("\t\t\tEnter pass mark: "))

print()

except:

print()

print("\t\t\tInvalid input for pass mark.Please re enter pass \

mark\n")

continue

if passmark>maxmark:#Passmark cannot be greater than maximum mark

print("\t\t\tPassmark is greater than maximum mark.Please re enter pass mark\n")

continue

if passmark==maxmark:#Passmark cannot be equal to maximum mark

print("\t\t\tPassmark is equal to maxmark.Please re enter pass mark\n")

continue

if passmark<=0:#Passmark cannot be less than or equal to 0

print("\t\t\tPass mark is less than 0.Please Re enter pass mark\n")

continue

else:

break

while True:

try:

markobtained=int(input("\t\t\tEnter mark obtained: "))

print()

except:

print()

print("\t\t\tInvalid input for mark obtained.Please re enter mark obtained")

continue

if markobtained>maxmark:#Markobtained cannot be greater than maximum mark

print("\t\t\tMark obtained is greater than maximum mark.Please re enter mark obtained\n")

continue

if markobtained<0:#Mark obtained cannot be less than 0

print("\t\t\tMark obtained cannot be less than 0.Please re enter mark obtained\n")

continue

else:

break

while True:

print("\t\t\tResult\n")

print("\t\t\tP-Pass\n")

print("\t\t\tF-Fail\n")

print("\t\t\tA-Absent\n")

try:

result=input("\t\t\tEnter result: ")

print()

except:

print("\t\t\tInvalid input for result.Please re enter result\n")

continue

if result not in ("P","F","A"):#Result can only be 'P'-Pass,'F'-Fail,'A-Absent'

print("\t\t\tInvalid input for result.Please re enter result")

continue

else:

break

cursor.execute("Select \* from result where registrationno=%s and subjectcode=%s"%(regno,subcode))#Checking for existence of result in result table

rec=cursor.fetchall()

if len(rec)!=0:

print("\t\t\tResult already exists\n")

return

try:

cursor.execute("insert into result values(%s,%s,%s,%s,%s,'%s')"%(regno,subcode,markobtained,passmark,maxmark,result))

mycon.commit()

print("\t\t\tAdded result successfully\n")

except:

print()

mycon.rollback()

print("\t\t\tUnable to add result\n")

def reportmarkless50():#Report of students whose marks are lesser than 50

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:

cursor.execute("select \

result.registrationno,name,subject,markobtained,passmark,maximummar\

k,result from student,result,subject \

where student.registrationno=result.registrationno and

subject.subjectcode=result.subjectcode and markobtained<50")

rec=cursor.fetchall()

except:

print("\t\t\tUnable to generate report\n")

return

if len(rec)==0:#Checking for existence of records where mark is less than 50

print("\t\t\tNo records exists\n")

return

else:

data=[]#Display records in tabular form

header=['Registration number','Name','Subject','Mark obtained','Pass mark','Maximum mark','Result']

for i in rec:

i=list(i)

data.append(i)

table=columnar(data,header,justify="c")

print("\t\t\tStudents whose marks are less than 50\n")

print(table)

def modifyresult():#Modify result

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

print("\t\t\tModify Result\n")

try:

registrationno=int(input("\t\t\tEnter the Registration \

Number to modify result: "))

print()

except:

print()

print('\t\t\tAn error has occured')

print()

return

cursor.execute("select student.registrationno from result,student \

where student.registrationno=%s and \

student.registrationno=result.registrationno"%(registrationno))

rec=cursor.fetchall()#Checking for existence of registration number entered by user

if len(rec)==0:

print("\t\t\tRegistration Number does not exist")

print()

return

print("\t\t\tResult\n")

print("\t\t\tP-Pass\n")

print("\t\t\tF-Fail\n")

print("\t\t\tA-Absent\n")

try:

result=input("\t\t\tEnter new result: ")

print()

except:

print()

print("\t\t\tInvalid result\n")

return

f=1

for i in result:

if i not in ["P","F","A"]:#Result can only be 'P'-Pass,'F'-Fail,'A'-\

Absent

f=0

if f==0:

print("\t\t\tInvalid result.\n")

print()

return

sql="Update result set result=%s where registrationno=%s";

value=(result,registrationno)

try:

cursor.execute(sql,value)

mycon.commit()

print('\t\t\tResult modified successfully')

print()

except:

print()

print('\t\t\tUnable to modify result')

print()

mycon.rollback()

mycon.close()

def reportmarkequal50():#Report of students whose marks are equal to 50

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:

cursor.execute("select result.registrationno,name,subject,markobtained,passmark,\

maximummark,result from student,result,subject where \

student.registrationno=result.registrationno \

subject.subjectcode=result.subjectcode and markobtained=50")

rec=cursor.fetchall()

except:

print("\t\t\tUnable to generate report\n")

return

if len(rec)==0:#Checking for existence of records where marks is equal to 50

print("\t\t\tNo records exists\n")

return

else:

data=[]#Displaying details in tabular form

header=['Registration number','Name','Subject','Mark obtained',\n

'Pass mark','Maximum mark','Result']

for i in rec:

i=list(i)

data.append(i)

table=columnar(data,header,justify="c")

print("\t\t\tStudents whose marks are equal to 50\n")

print(table)

def searchresultbyregistrationnumber():#Searching a result by registration number

print("\t\t\tSearch result by registration number\n")

try:

regno=int(input("\t\t\tEnter registration number: "))

print()

except:

print("\t\t\tInvalid registration number\n")

print()

return

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user='root',\

passwd="sql123",database="college")

cursor=mycon.cursor()

from columnar import columnar

cursor.execute("select \* from student where registrationno=%s"%(regno))

rec=cursor.fetchall()

if len(rec)==0:#Checking for existence of registration number entered by user

print("\t\t\tInvalid Registration number\n")

return

cursor.execute("select result.registrationno,name,subject,markobtained,passmark,\

maximummark,result from student,result,subject where \

student.registrationno=result.registrationno and \

subject.subjectcode=result.subjectcode and \

student.registrationno=%s"%(regno))

rec1=cursor.fetchall()

if len(rec1)==0:

print("\t\t\tInvalid Registration number\n")

return

data=[]#Displaying setails in tabular form

header=['Registration number','Name','Subject','Mark obtained',\

'Pass mark','Maximum mark','Result']

for i in rec1:

i=list(i)

data.append(i)

table=columnar(data,header,justify="c")

print(table)

def addsubject():#Adding a new subject to subject table

print("\t\t\tAdd a subject\n")

try:

subname=input("\t\t\tEnter subject: ")

except:

print("\t\t\tInvalid subject name\n")

return

if len(subname)>20:#Subject name should be less than 20 characters

print("\t\t\tSubject name too long\n")

return

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:

cursor.execute("Select \* from subject where subject='%s'"%(subname))

except:

print("\t\t\tAn error has occured")

print("\n\n\n")

return

rec=cursor.fetchall()

if len(rec)==1:#Checking for existence of subject entered by user

print("\t\t\tSubject already exists\n")

return

try:

cursor.execute("select max(subjectcode) from subject")

except:

print("\t\t\tAn error has occured")

print("\n\n\n")

return

a=cursor.fetchall()

scode=0

for i in a:#Auto generate subjectcode

if i[0]==None:

scode=1

else:

scode=i[0]+1

try:

cursor.execute("insert into subject \

values(%s,'%s')"%(scode,subname))

mycon.commit()

print("\t\t\tSubject code for",subname,"is",scode)

print("\t\t\tNew subject inserted successfully")

print("\n")

except:

print("\t\t\tUnable to add subject")

print("\n")

mycon.rollback()

def reportmarkgreater50():#Report of students whose marks are greater than 50

from columnar import columnar

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",\

passwd="sql123",database="college")

cursor=mycon.cursor()

try:

cursor.execute("select \

result.registrationno,name,subject,markobtained,passmark,\

maximummark,result from student,result,subject where \

student.registrationno=result.registrationno and \

subject.subjectcode=result.subjectcode and markobtained>50")

rec=cursor.fetchall()

except:

print("\t\t\tUnable to generate report\n")

return

if len(rec)==0:#Checking for existence of records where mark is greater than 50

print("\t\t\tNo records exists\n")

return

else:#Display data in tabular form

data=[]

header=['Registration number','Name','Subject','Mark obtained',\

'Pass mark','Maximum mark','Result']

for i in rec:

i=list(i)

data.append(i)

table=columnar(data,header,justify="c")

print("\t\t\tStudents whose marks are greater than to 50\n")

print(table)

def searchresultbysubjectcode():#Searching result by subject code

print("\t\t\tSearch result by subject code\n")

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user='root',\

passwd="sql123",database="college")

cursor=mycon.cursor()

from columnar import columnar

cursor.execute("Select \* from subject")

rec=cursor.fetchall()

if len(rec)==0:#Checking for existence of subject in subject table

print("\t\t\tNo Subject exists\n")

return

for i in rec:

print("\t\t\t",i[0],"\t\t",i[1])

try:

subcode=int(input("\t\t\tEnter subject code: "))

print()

except:

print()

print("\t\t\tInvalid subject code.")

print()

return

cursor.execute("select \* from subject where \

subjectcode=%s"%(subcode))

rec=cursor.fetchall()

if len(rec)!=1:#Checking for existence of subject code entered by user

print("\t\t\tInvalid subject code1\n")

return

try:

cursor.execute("select result.registrationno,name,subject,markobtained,passmark,\

maximummark,result from student,result,subject where \

student.registrationno=result.registrationno and \

subject.subjectcode=result.subjectcode and \

result.subjectcode=%s"%(subcode))

rec1=cursor.fetchall()

#print(rec1)

if len(rec1)==0:

print("\t\t\tNo records found\n")

return

except:

print()

print("\t\t\tAn error has occured\n")

print()

return

data=[]#Displaying result details in tabular form

header=['Registration number','Name','Subject','Mark obtained',\

'Pass mark','Maximum mark','Result']

for i in rec1:

i=list(i)

data.append(i)

table=columnar(data,header,justify="c")

print(table)

def result():#Menu for operations performed in result register

while True:

print("\t\t\tResult menu\n")

print("\t\t\t1.Add a subject\n")

print("\t\t\t2.Add a result\n")

print("\t\t\t3.Search result by registration number\n")

print("\t\t\t4.Search result by subject\n")

print("\t\t\t5.Details of students whose marks is less than 50\n")

print("\t\t\t6.Details of students whose marks is equal to 50\n")

print("\t\t\t7.Details of atudents whose marks is greater than 50\n")

print("\t\t\t8.Modify result\n")

print("\t\t\t9.Exit\n")

print()

try:

ch=int(input("\t\t\tEnter your choice: "))

except:

print()

print("\t\t\tAn error has occured\n")

return

if ch==1:

addsubject()

elif ch==2:

addresult()

elif ch==3:

searchresultbyregistrationnumber()

elif ch==4:

searchresultbysubjectcode()

elif ch==5:

reportmarkless50()

elif ch==6:

reportmarkequal50()

elif ch==7:

reportmarkgreater50()

elif ch==8:

modifyresult()

elif ch==9:

break

else:

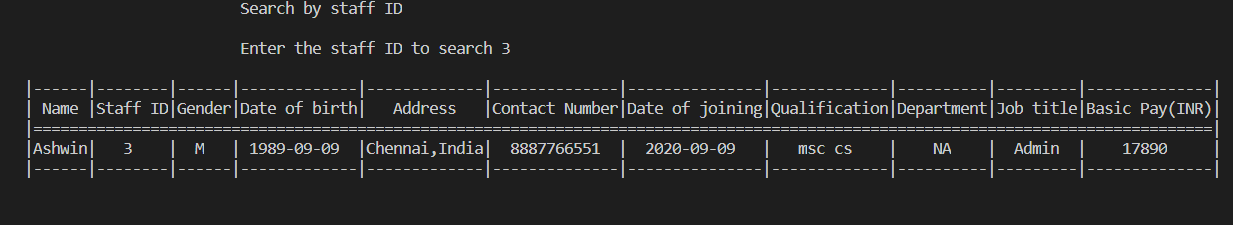
print("\t\t\tInvalid choice\n")

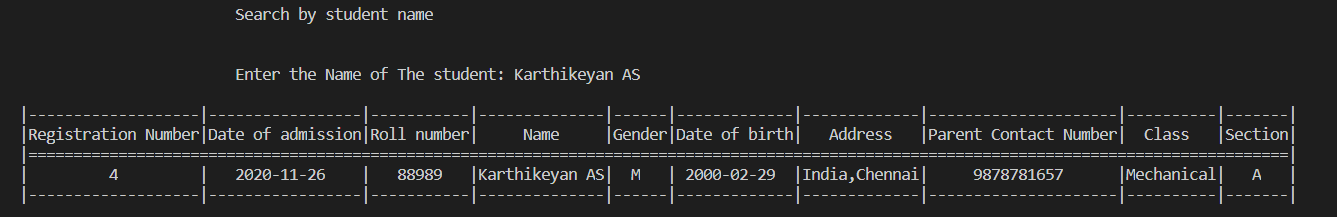
continue

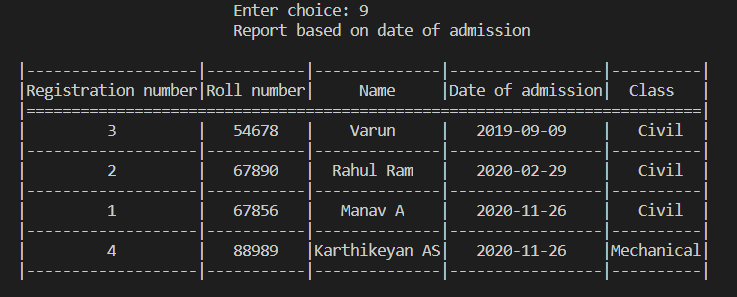
createtables()

menu()

# **OUTPUTS**







# **BIBLIOGRAPHY**

* 1. SUMITA ARORA
  2. PREETI ARORA
  3. <https://www.programiz.com/python-programming/datetime/strptime>
  4. <https://www.programiz.com/python-programming/datetime/strftime>
  5. https://github.com/MaxTaggart/columnar