# Project Report On

**School Management**

**System**

Submitted to

**Jayshree Periwal High School**

**3, Chitrakoot Scheme**

**JAIPUR**

In partial fulfilLment of

the requirements for

**All India Senior School**

**Certificate Examination 2021**

of

**Central Board of Secondary Education**

Submitted By: Yashashawi Bharadwaj

I would like to thank everyone who helped me to accomplish this project.

**Acknowledgement**

My sincere thanks to my *family members* and *respected teachers*, who have helped me with their valuable suggestions and support throughout the development of the project.

I am highly thankful to my project guide **Ms. Ranjeeta** for providing guidance and support at every stage of the project.

I am extremely grateful to **Mrs. Jayshree Periwal**, *Director* and **Mrs. Madhu Maini**, *Principal* of *JAYSHREE PERIWAL HIGH SCHOOL*, Jaipur, for providing us a very good computer lab, due to which this project became possible.

**YASHASHAWI BHARADWAJ**

**Certificate of Originality**

This is to certify that the Project Report entitled “School Management System” submitted to JAYSHREE PERIWAL HIGH SCHOOL in partial fulfilment of the requirement for all India Senior Certificate Examination (AISSCE) 2021 of CBSE, is an original work carried out by YASHASHAWI BHARADWAJ under my guidance.

The matter embodied in this project is genuine work done by the students and has not been submitted of any course of study.

.....................................................

Signature of The Guide

Date:..........................................

Name : Mr. S. Laxman Rao

(HOD – Computer)

JAYSHREE PERIWAL HIGH SCHOOL

Jaipur

**CONTENT**

|  |  |  |
| --- | --- | --- |
| **SNo**. | **Topic** | **Page No.** |
| 1. | Objective & Scope of the Project | 5 |
| 2. | Problem Definition | 7 |
| 3. | Life Cycle of the Project | 8 |
| 4. | Details of Hardware and  Software used | 10 |
| 5. | Input Screen Designs | 12 |
| 6. | Source Code of the Project | 30 |
| 7. | Data Dictionary | 53 |

1. **Objective & Scope of the Project**

**Objective**

The main objective of the project is to computerize the processing of few tasks related to Administration of a School

This system helps the user to maintain the records of Teachers and Students.

This application is used by the School authorities to maintain details of Teachers and students.

Users can also edit, add and delete details of the Teacher and Students.

Users also get the function of comparing marks of students with the average marks through Graphical Representation.

Students can also find out the semester wise fees of each class.

**Scope**

This project is developed for a School

It can further be easily optimized for the use of other educational institutions

This project will help the school administration to easily organise and access data of teachers and students. It will also provide them student’s marks comparison with the average marks in a Graphical Representation which will help them analyse each child’s strength and weaknesses.

This will be a time efficient way for the school authorities to handle large amount of data digitally and make a “Paperless” workplace.

1. **Problem Definition**

The project “School Management System” shows a simple School Management procedure conducted therein.

The system should be developed to maintain the students’ and teachers’ records, marks records, etc.

Each detail must be filled properly and options must be provided.

A proper database should be maintained in the RDBMS and the front end to be developed using advanced GUI interfaces.

The system should be able to handle exceptional situations.

1. **Life Cycle of Project**

System Development Life Cycle (SDLC)

The System Development Life Cycle (SDLC) is a set of activities that analysts, designers and users carry out to develop and implement an Information System.

The SDLC consists of the following activities:

Feasibility Study

Requirement Definition

Design (Database & Program)

Development of Software

Unit Testing

System Testing

Implementation

Evaluation

Maintenance

**Context Diagram**

**School Admin**

Enter Change/Delete

View Details

Details Details

Teacher Details

Student Details

**TEACHERS**

**STUDENTS**

**MARKS**

STUDENT MARKS

**FEES**

Fee details

Enter class

View Details

**Student/Parent**

Get Marks comparison in Graphical Form

Enter student marks

**Teacher**

1. **Details of Hardware and Software used**

Platform : Windows 10

Tools : IDLE (Python 3.8)

RDBMS : MySQL

**Hardware Specifications**

Microprocessor (CPU) : Intel Core i7

Memory (RAM) : 16 GB

Virtual Memory : 64-Bit

Hard Disk : 1 TB

VDU : SVGA

Keyboard : Standard 104 Keys

Mouse : Standard 2-Buttons Mouse

(Scroll)

Printer : Inkjet/Laser

**Software Specifications**

Operating System : Windows 10

Front-End Design : IDLE (Python 3.8)

Back-End : MySQL

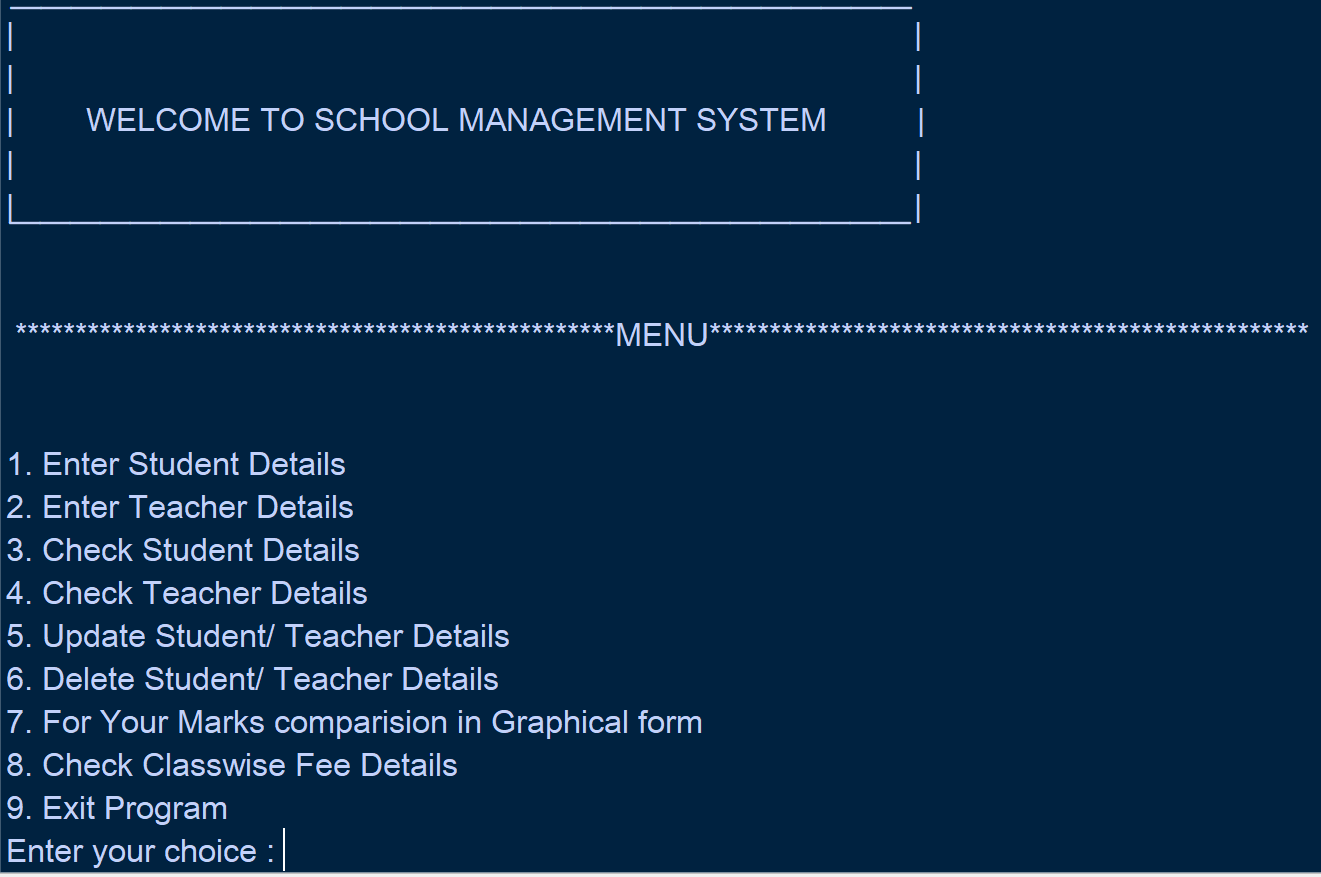
Documentation : Microsoft Word 2016 and MS

Paint

1. **Input Screen Designs**

**Input Forms**

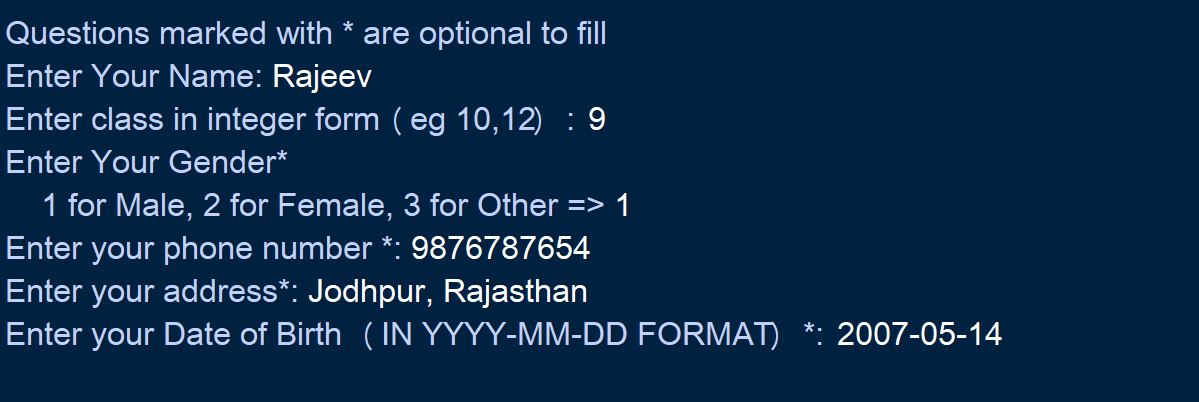
The **First Screen** of the project is as shown below:



There are **9 menu options available** in the main form and any one of them can be selected by entering their corresponding number.

First menu Option “Enter Student Details” is used to enter student details.

Screenshot of the Options of the First menu option:

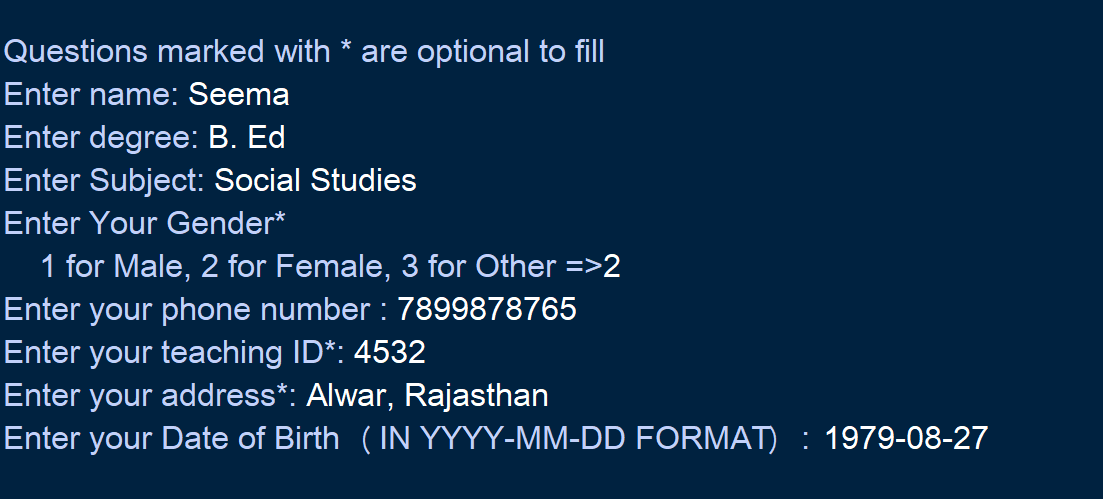


The above screen is used to enter student details.

The details include Name, Class, Gender, Phone Number, Address and Date of Birth of the student.

**Second menu** Option “Enter Student Details” is used to enter student details.

Screenshot of the Options of the Second menu option:

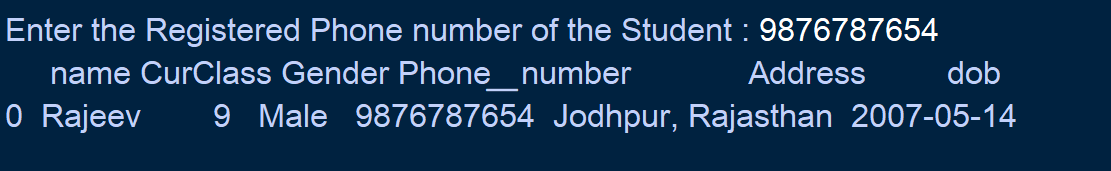


The above screen is used to enter teacher details.

The details include Name, Degree, Subject, Gender, Phone Number, Teaching ID, Address and Date of Birth of the teacher.

**Third menu** Option “Check Student Details” is used to get the details of a particular student.

Screenshot of the Options of the Third menu option:



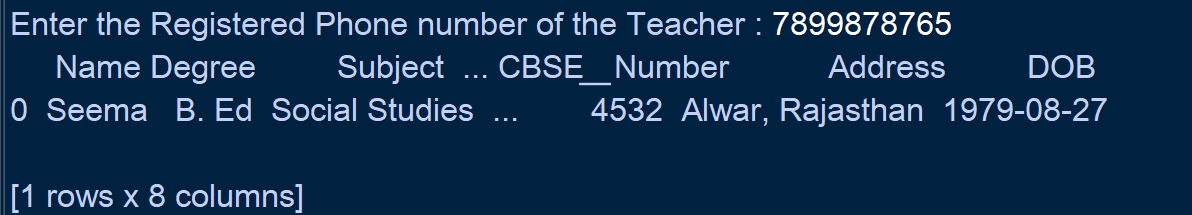
Firstly, the user will have to enter the registered mobile number of the student.

Then we get the details of the particular student as shown in the above screenshot.

The details include Name, Current Class, Gender, Phone number, Address and Date of Birth of the particular student.

**Fourth** menu Option “Check Teacher Details” is used to get the details of a particular teacher.

Screenshot of the Options of the Third menu option:



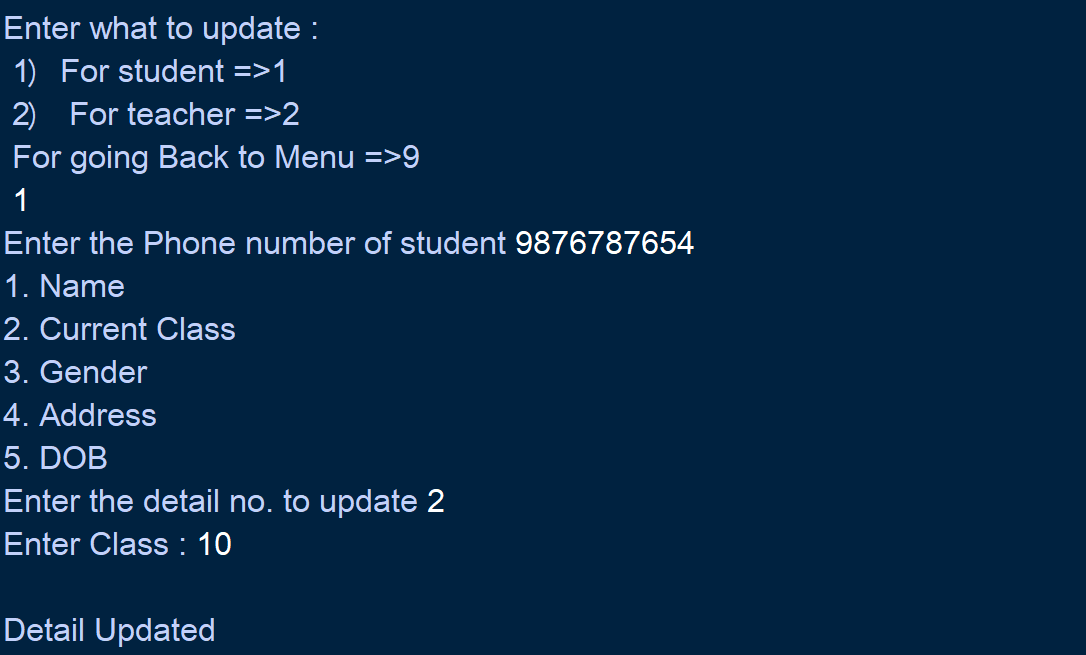
Firstly, the user will have to enter the registered mobile number of the teacher.

Then we get the details of the particular teacher as shown in the above screenshot.

The details include Name, Degree, Subject, Gender, Phone number, Teaching ID, Address and Date of Birth of the particular teacher.

**Fifth** menu Option “Update Student/Teacher Details” is used to get the details of a particular teacher.

Screenshot of the Options of the Third menu option:



The first internal choice is whether the user has to update the details of a student or a teacher.

As shown in the screenshot above, the entered choice is 1 (i.e. Student)

Then the user will have to input the Registered Mobile number of the Student.

The user then gets options to choose what details they want to update.

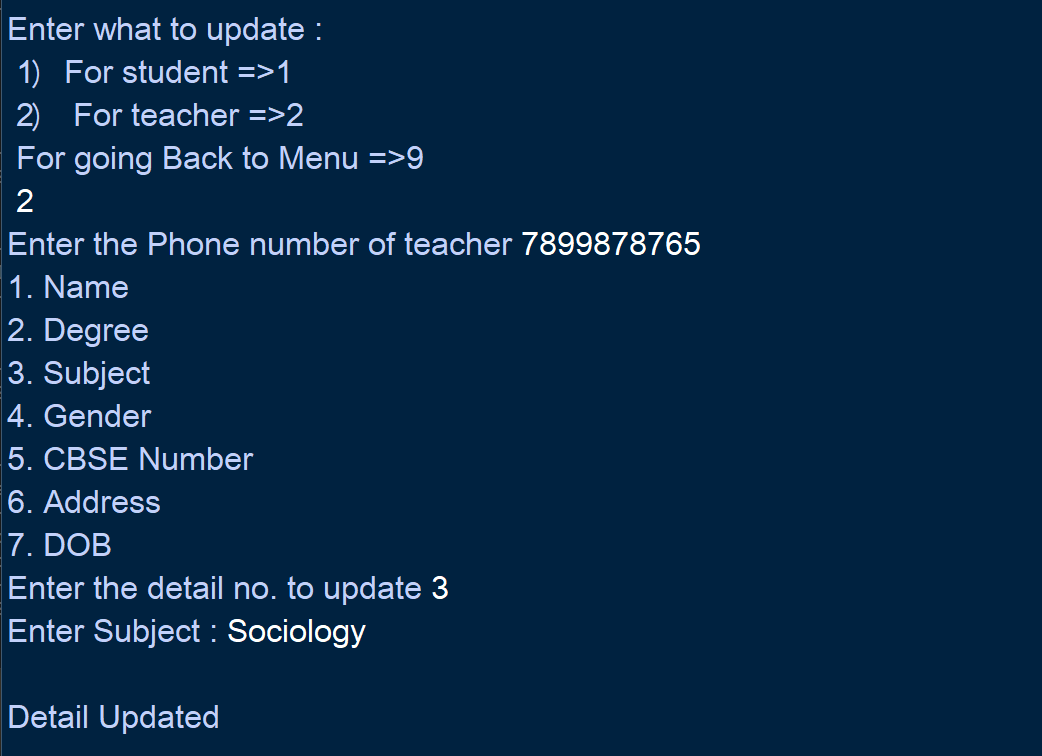
In case of student the options are :-

1. Name
2. Current Class
3. Gender
4. Address
5. DOB

After updating the details, we can see the updated information as shown in the screenshot below.



As shown in the screenshot below, now we have entered the choice as 2 (i.e. Teacher)



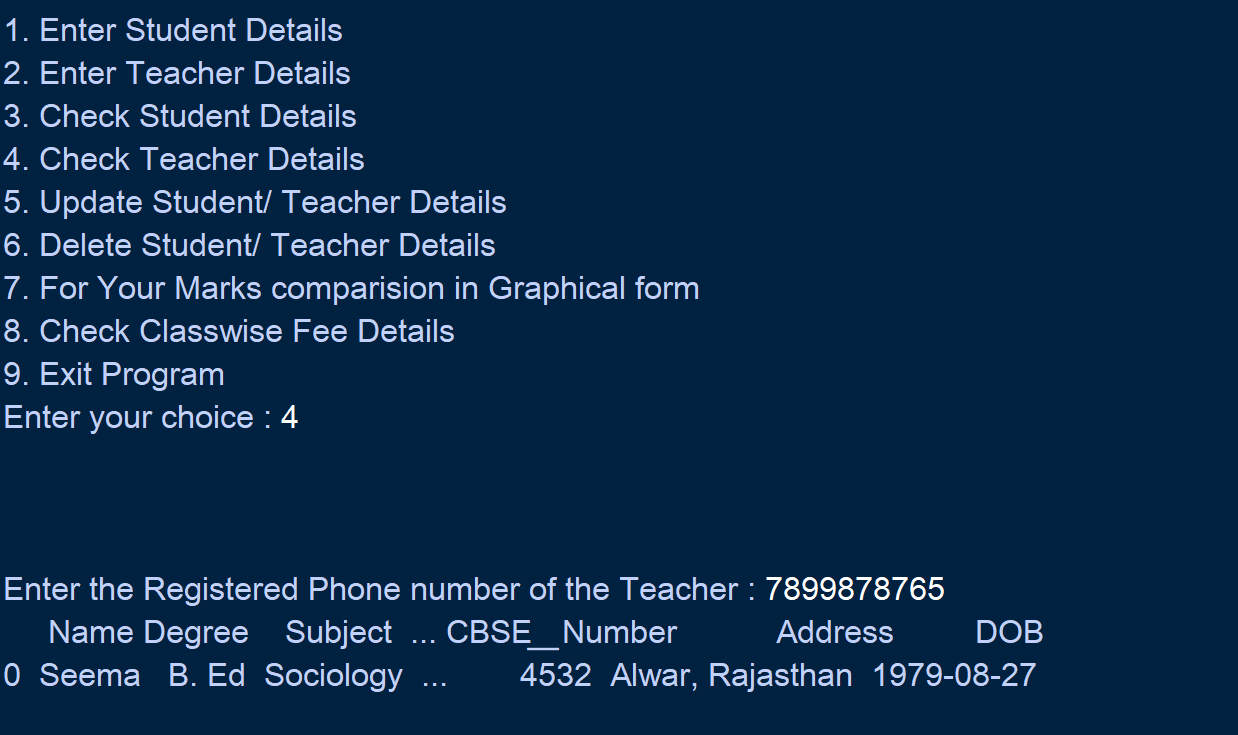
Then the user will have to input the Registered Mobile number of the Teacher.

The user then gets options to choose what details they want to update.

In case of teacher the options are :-

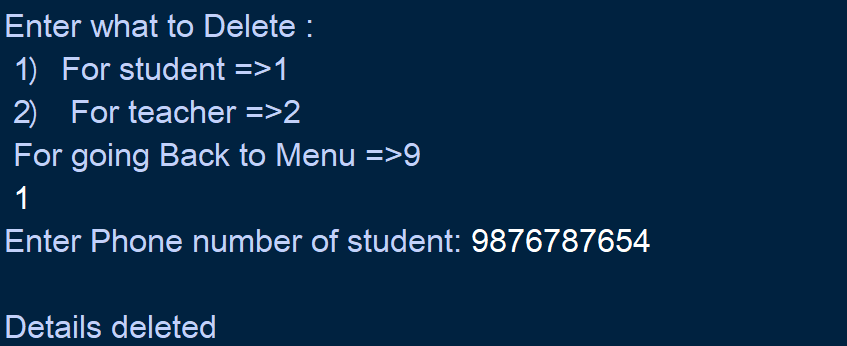
1. Name
2. Degree
3. Subject
4. Gender
5. CBSE Number
6. Address
7. DOB

After updating the details, we can see the updated information as shown in the screenshot below.



**Sixth** menu Option “Delete Student/Teacher Details” is used to delete the details of a particular teacher.

Screenshot of the Options of the Third menu option:



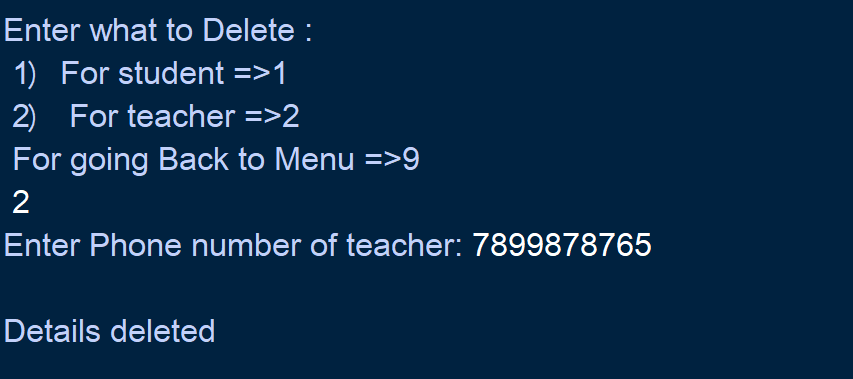
The first internal choice is whether the user has to update the details of a student or a teacher.

As shown in the screenshot above, the entered choice is 1 (i.e. Student)

Then the user will have to input the Registered Mobile number of the Student.

As soon as you enter the phone number the details of the particular student will get deleted.

As shown in the screenshot below, now we have entered the choice as 2 (i.e. Teacher)

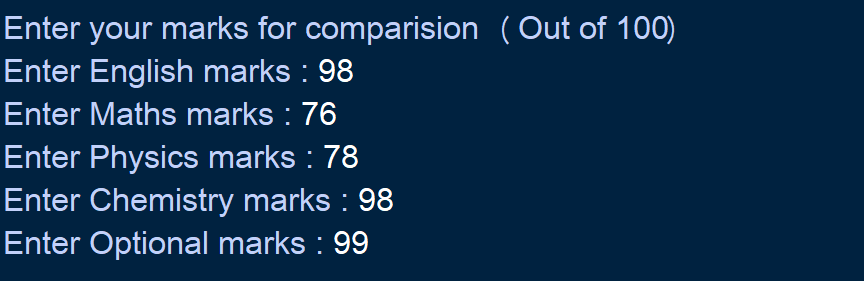


Then the user will have to input the Registered Mobile number of the teacher.

As soon as you enter the phone number the details of the particular teacher will get deleted.

**Seventh** menu Option “For your Marks comparison in Graphical form” is used to compare marks of a student with the average marks.

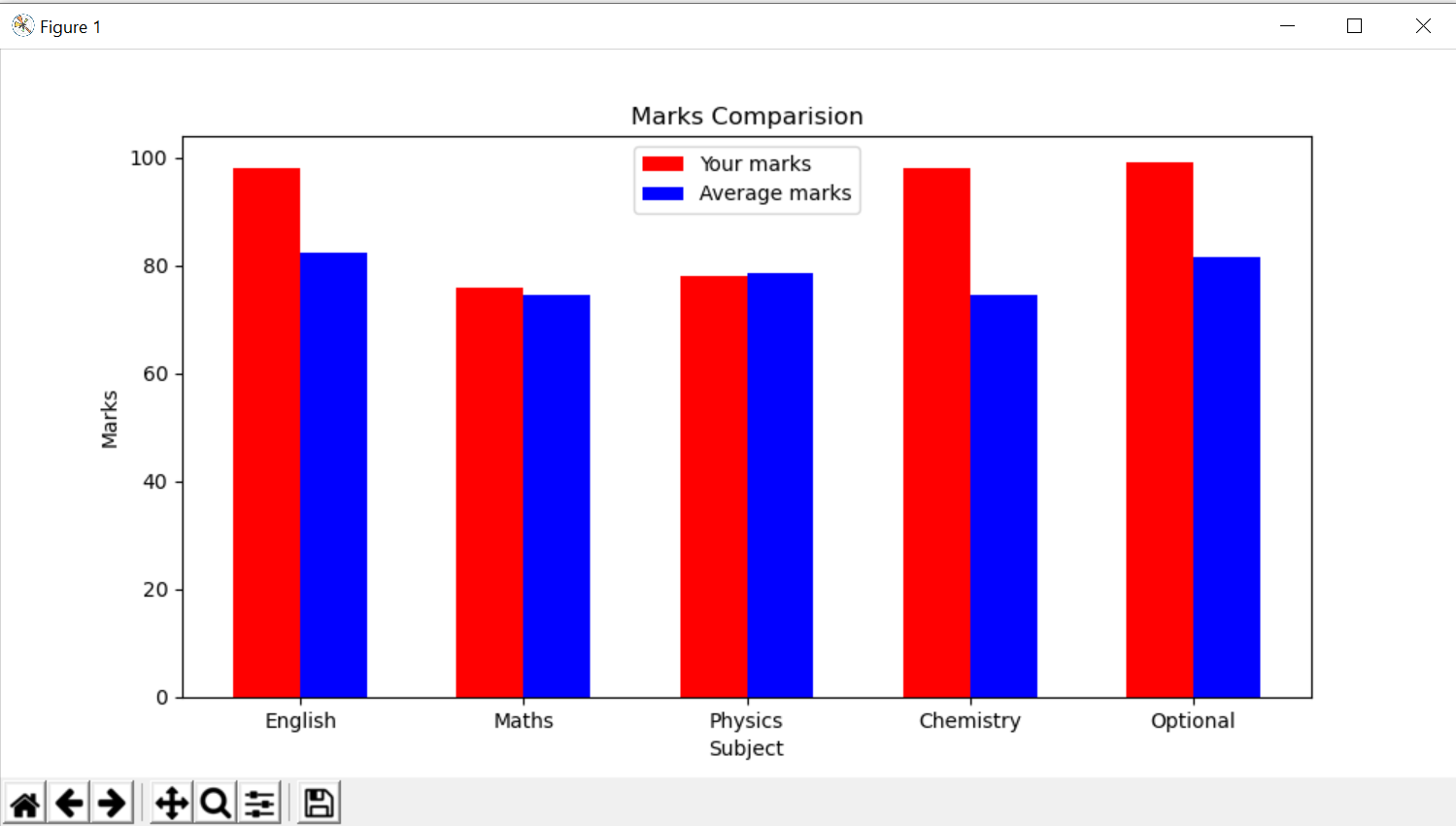
Screenshot of the Options of the Third menu option:



First the user will have to enter student marks out of 100 as seen in the screenshot above.

As soon as the user enters the marks, a graphical comparison window pops up.

The screenshot of the graphical comparison is given below:



The red bars are the students marks

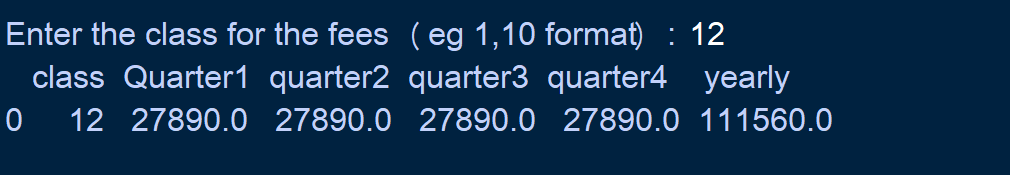
The blue bars are the average marks.

The x axis of the graph has all the subjects marked

The y axis of the graph has the marks up to 100 with an interval of 20 marks.

**Eighth** menu Option “Check Classwise Fee Details” is used get details of semester wise fee for students for their respective classes.

The screenshot of the eighth menu option is given below:



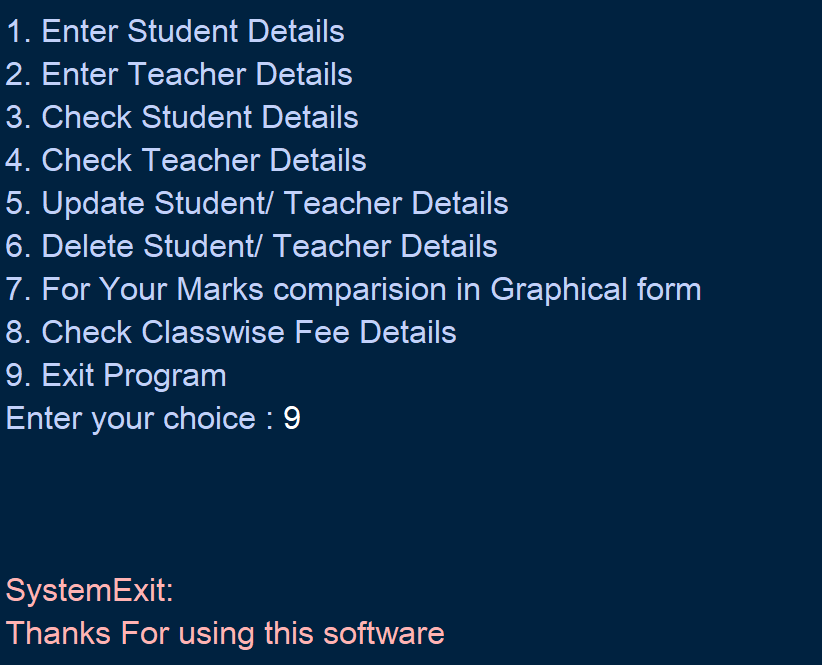
The user has entered the class as 12.

So, the user will get the fee details for class 12.

The user gets fee details of each quarter and also the total fee per annum.

**Ninth** menu Option “Exit Program” is used to exit the program.

The screenshot of the ninth menu option is given below:



1. **Source Code of the Project**

**Source code**

* **Modules Used**

import pandas as pd

import random

import itertools

import numpy as np

import mysql.connector

from random import randint

import matplotlib.pyplot as plt

import sys

mydb = mysql.connector.connect(host="localhost", user ="root",passwd ="12345678", database ="appdata")

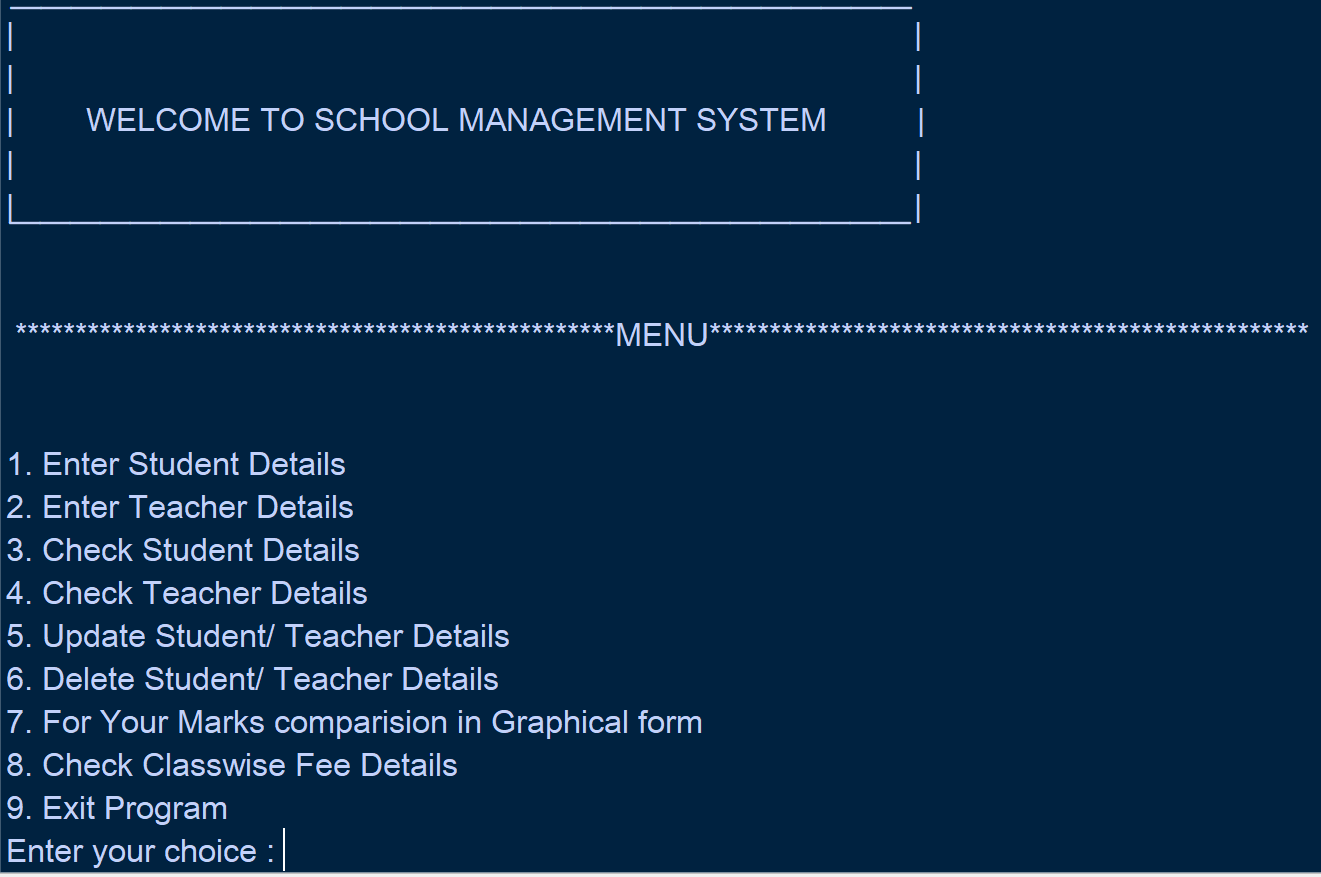
mycursor=mydb.cursor()

#For conversion in 1D array

def oneDArray(x):

return list(itertools.chain(\*x))

* **Main menu (Front page)**

****

def menu():

print("\n\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n\n")

print("1. Enter Student Details ")

print("2. Enter Teacher Details ")

print("3. Check Student Details")

print("4. Check Teacher Details ")

print("5. Update Student/ Teacher Details ")

print("6. Delete Student/ Teacher Details ")

print("7. For Your Marks comparison in Graphical form ")

print("8. Check Classwise Fee Details ")

print("9. Exit Program ")

inp=int(input("Enter your choice : "))

print("\n\n")

if inp==1:

StudentData()

elif inp==2:

teacherdata()

elif inp==3:

studet()

elif inp==4:

teadet()

elif inp==5:

Updatedata()

elif inp==6:

deldata()

elif inp==7:

marks()

elif inp==8:

fees()

elif inp==9:

sys.exit("\nThanks For using this software")

else:

print("Please Enter a Valid number \n\n")

menu()

print(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print("| |")

print("| |")

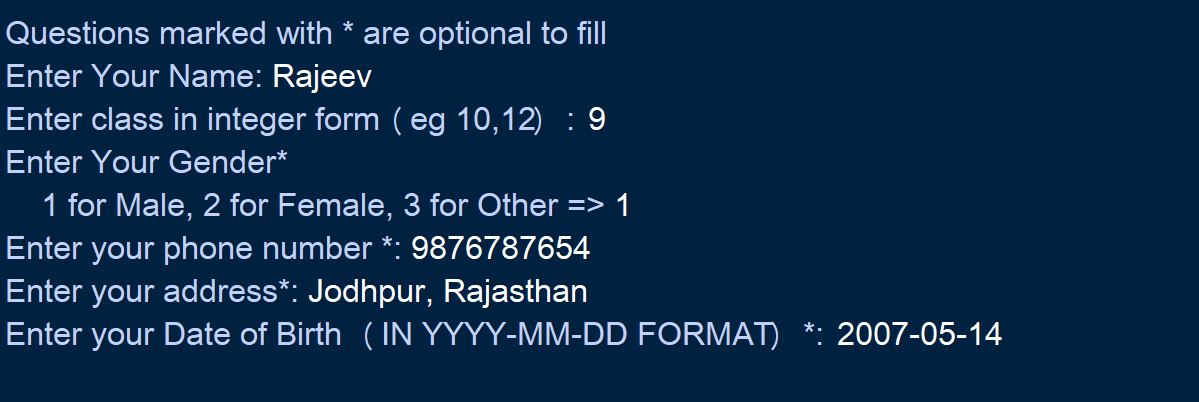
print("| WELCOME TO SCHOOL MANAGEMENT SYSTEM |")

print("| |")

print("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|")

menu()

1. **Enter student details**

****

def StudentData():

l=[]

print("Questions marked with \* are optional to fill")

name=input("Enter Your Name: ")

l.append(name)

curclass=int(input("Enter class in integer form(eg 10,12): "))

l.append(curclass)

gender=int(input("""Enter Your Gender\*

1 for Male, 2 for Female, 3 for Other => """))

if gender==1:

gender="Male"

elif gender==2:

gender="Female"

elif gender==3:

gender="Other"

l.append(gender)

phone\_number=int(input("Enter your phone number \*: "))

l.append(phone\_number)

address=input("Enter your address\*: ")

l.append(address)

dob=input("Enter your Date of Birth (IN YYYY-MM-DD FORMAT)\*: ")

l.append(dob)

stu=(l)

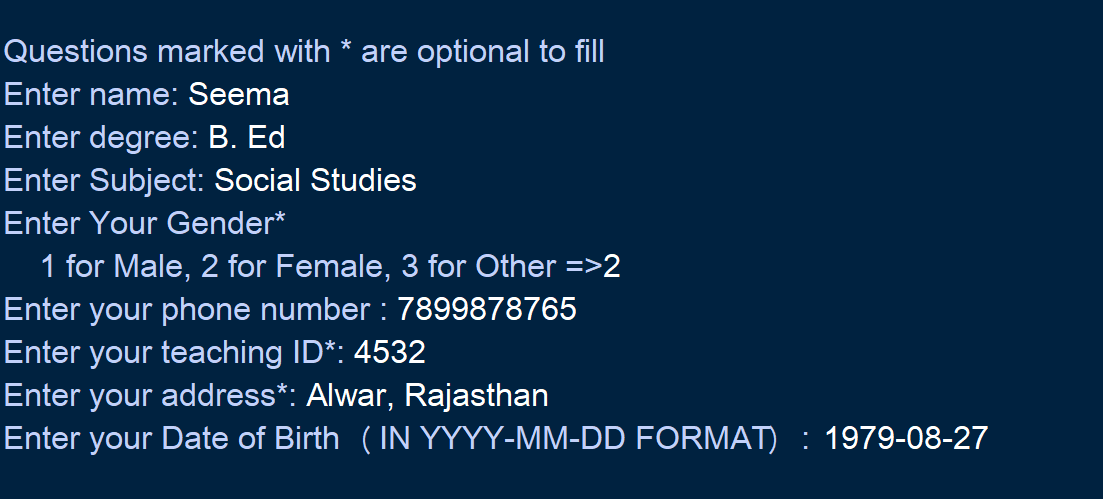
sql="insert into students(name,curclass,gender,phone\_number,address,dob)values(%s,%s,%s,%s,%s,%s)"

mycursor.execute(sql,stu)

mydb.commit()

menu()

1. **Enter Teacher Details**

****

def teacherdata():

k=[]

print("Questions marked with \* are optional to fill")

name=input("Enter name: ")

k.append(name)

degree=input("Enter degree: ")

k.append(degree)

subject=input("Enter Subject: ")

k.append(subject)

gender=int(input("""Enter Your Gender\*

1 for Male, 2 for Female, 3 for Other =>"""))

if gender==1:

gender="Male"

elif gender==2:

gender="Female"

elif gender==3:

gender="Other"

k.append(gender)

phone\_number=int(input("Enter your phone number : "))

k.append(phone\_number)

CBSE\_number=input("Enter your teaching ID\*: ")

k.append(CBSE\_number)

address=input("Enter your address\*: ")

k.append(address)

dob=input("Enter your Date of Birth (IN YYYY-MM-DD FORMAT): ")

k.append(dob)

teacher=(k)

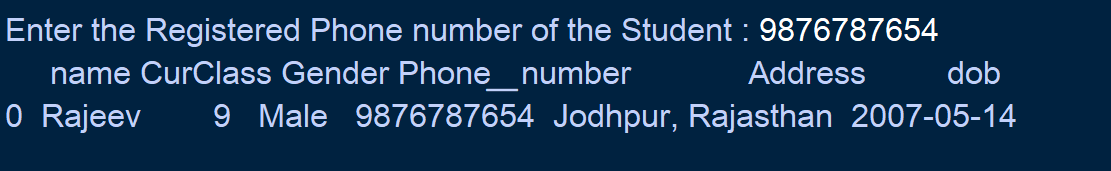
sql="insert into teachers(name,degree,subject,gender,phone\_number,CBSE\_number,address,dob)values(%s,%s,%s,%s,%s,%s,%s,%s)"

mycursor.execute(sql,teacher)

mydb.commit()

menu()

1. **Check Student Details**

****

def studet():

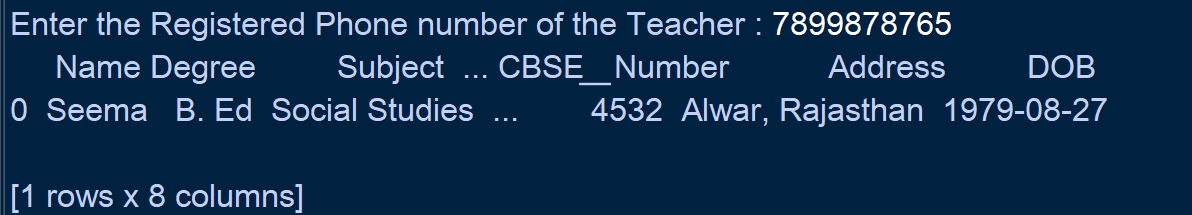
ph=input("Enter the Registered Phone number of the Student : ")

det=pd.read\_sql("select \* from students where phone\_number='%s';"%(ph),mydb)

print(det)

menu()

1. **Check Teacher Details**

****

def teadet():

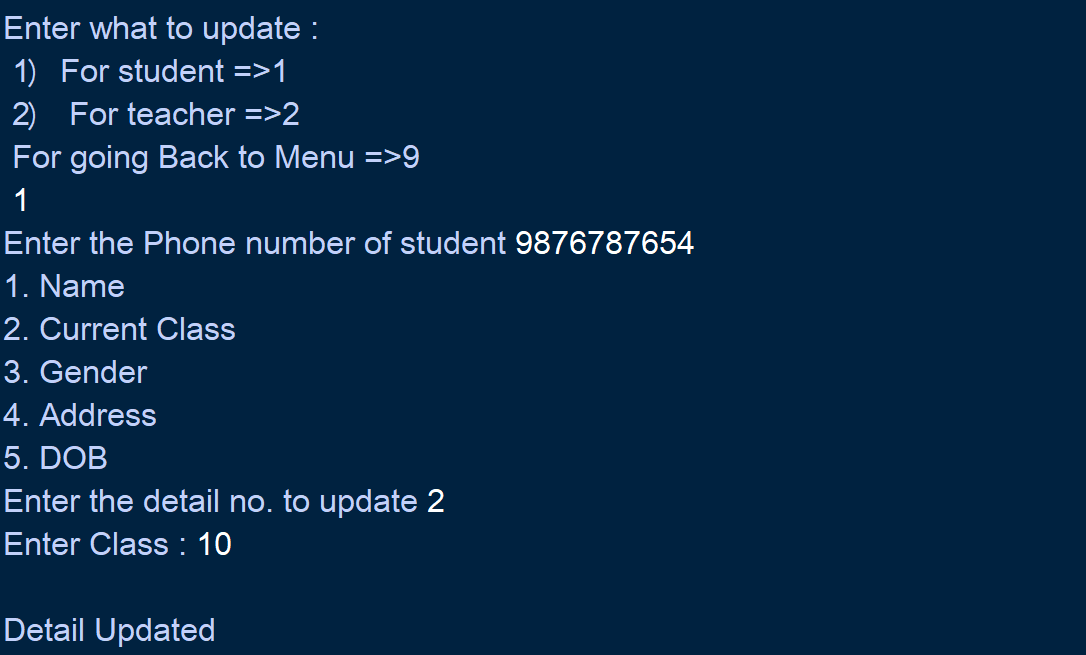
ph=input("Enter the Registered Phone number of the Teacher : ")

det=pd.read\_sql("select \* from teachers where phone\_number='%s';"%(ph),mydb)

print(det)

menu()

1. **Update Student/ Teacher Details**

****

def Updatedata():

x=int(input("Enter what to update : \n 1)For student =>1 \n 2) For teacher =>2 \n For going Back to Menu =>9 \n "))

if x==1:

n=input("Enter the Phone number of student ")

print("1. Name")

print("2. Current Class")

print("3. Gender")

print("4. Address")

print("5. DOB")

y=int(input("Enter the detail no. to update "))

if y==1:

s=input("Enter Name : ")

sql="update students set name='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==2:

s=input("Enter Class : ")

sql="update students set Curclass='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==3:

s=input("Enter Gender : ")

sql="update students set Gender='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==4:

s=input("Enter Address : ")

sql="update students set address='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==5:

s=input("Enter DOB : ")

sql="update students set DOB='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

else:

print("\nPlease enter a valid number")

menu()

menu()

elif x==2:

n=input("Enter the Phone number of teacher ")

print("1. Name")

print("2. Degree")

print("3. Subject")

print("4. Gender")

print("5. CBSE Number")

print("6. Address")

print("7. DOB")

y=int(input("Enter the detail no. to update "))

if y==1:

s=input("Enter Name : ")

sql="update teachers set name='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==2:

s=input("Enter Degree : ")

sql="update teachers set degree='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==3:

s=input("Enter Subject : ")

sql="update teachers set subject='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==4:

s=input("Enter Gender : ")

sql="update teachers set Gender='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==5:

s=input("Enter CBSE Number : ")

sql="update teachers set CBSE\_Number='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==6:

s=input("Enter Address : ")

sql="update teachers set address='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

elif y==7:

s=input("Enter DOB (YYYY-MM-DD): ")

sql="update teachers set dob='%s' where phone\_number='%s';"%(s,n)

mycursor.execute(sql)

mydb.commit()

print("\nDetail Updated")

else:

print("\nPlease enter a valid number")

menu()

menu()

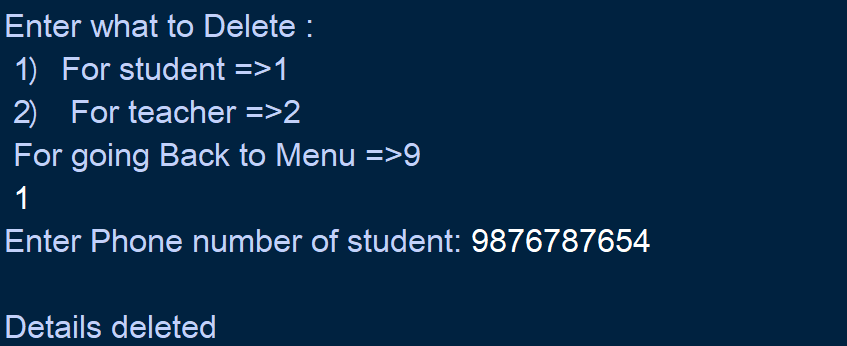
elif x==9:

menu()

else:

menu()

1. **Delete Student/ Teacher Details**

****

def deldata():

x=int(input("Enter what to Delete : \n 1)For student =>1 \n 2) For teacher =>2 \n For going Back to Menu =>9 \n "))

if x==1:

d=int(input("Enter Phone number of student: "))

qry="delete from students where Phone\_number='%s';"%(d)

mycursor.execute(qry)

mydb.commit()

print("\nDetails deleted ")

menu()

elif x==2:

d=int(input("Enter Phone number of teacher: "))

qry="delete from teachers where Phone\_number='%s';"%(d)

mycursor.execute(qry)

mydb.commit()

print("\nDetails deleted ")

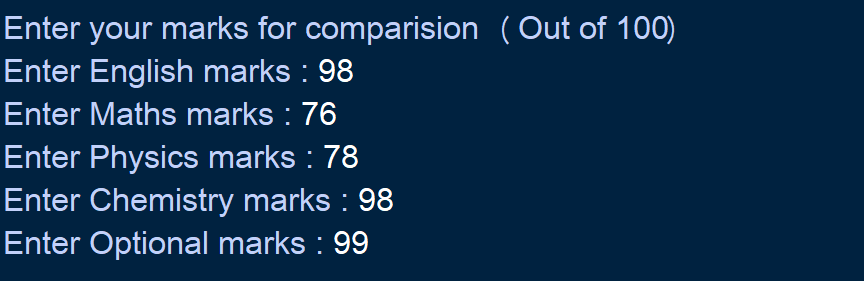
menu()

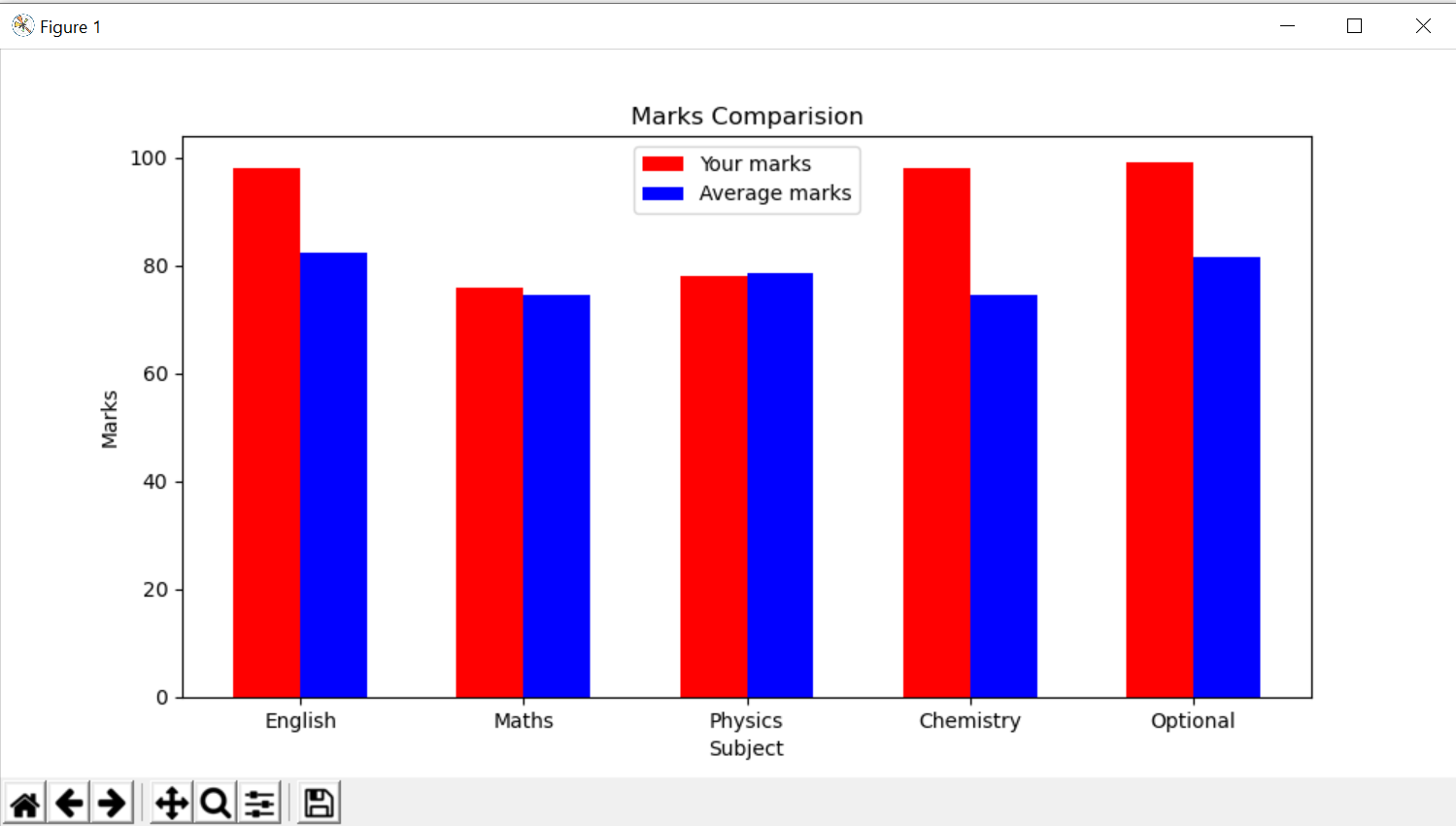
else:

print("\nPlease enter a valid number")

menu()

1. **For Your Marks comparison in Graphical form**

****

****

#For conversion in 1D array

def oneDArray(x):

return list(itertools.chain(\*x))

# For pyplot

def plot(m,fin):

n=5

sub=("English","Maths","Physics","Chemistry","Optional")

y\_pos=np.arange(len(sub))

ind=np.arange(n)

width=0.3

plt.figure(figsize=(10,5))

plt.bar(ind,m,width,color='r',label='Your marks')

plt.bar(ind+width,fin,width,color='b',label='Average marks')

plt.xlabel("Subject")

plt.ylabel("Marks")

plt.xticks(ind + width / 2, sub)

plt.title("Marks Comparison")

plt.legend()

plt.show()

def marks():

print("Enter your marks for comparison (Out of 100)")

m=[]

eng=int(input("Enter English marks : "))

m.append(eng)

math=int(input("Enter Maths marks : "))

m.append(math)

phy=int(input("Enter Physics marks : "))

m.append(phy)

chem=int(input("Enter Chemistry marks : "))

m.append(chem)

opt=int(input("Enter Optional marks : "))

m.append(opt)

marks=(m)

sql="insert into marks(english,maths,physics,chemistry,optional)values(%s,%s,%s,%s,%s);"

mycursor.execute(sql,marks)

mydb.commit()

avg=pd.read\_sql("select avg(english),avg(maths),avg(physics),avg(chemistry),avg(optional)from marks;",mydb)

ab=avg.values.tolist() #gives nested list

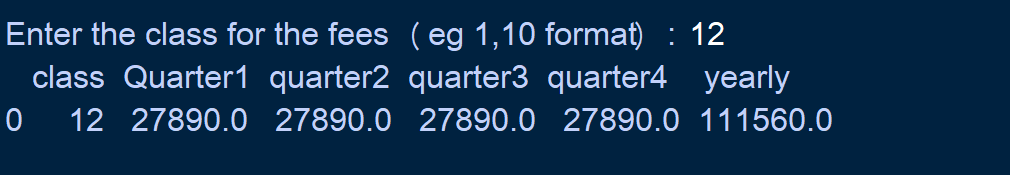
#For converting to 1D list

fin=oneDArray(ab)

grph=plot(marks,fin)

menu()

1. **Check Classwise Fee Details**

****

def fees():

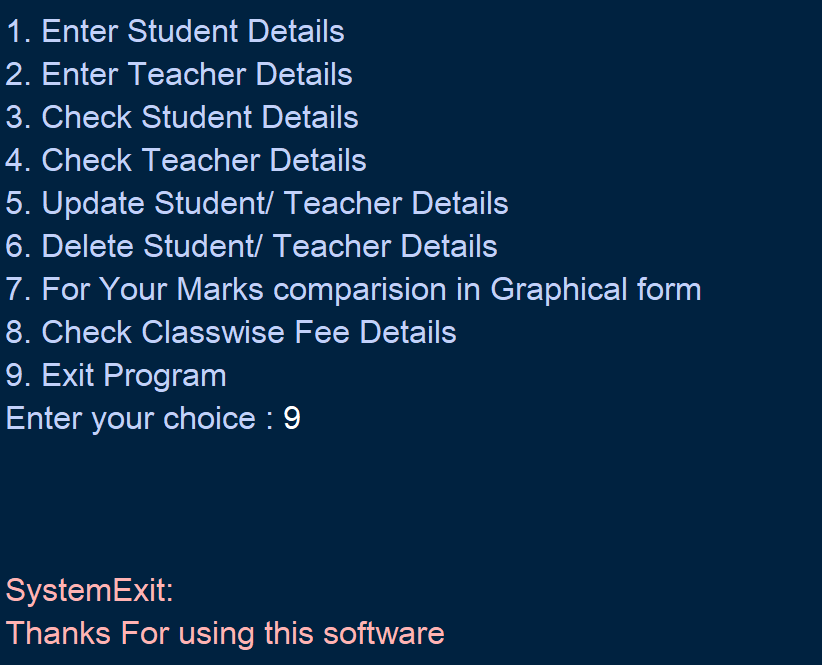
cl=int(input("Enter the class for the fees (eg 1,10 format): "))

f=pd.read\_sql("select class, Quarter1, quarter2, quarter3, quarter4, yearly from fees where class='%s';"%(cl),mydb)

print (f)

menu()

1. **Exit Program**

****

sys.exit("\nThanks For using this software")

1. **DATA DICTIONARY**

Tables used in this project are:

1. Students

+--------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------+--------------+------+-----+---------+-------+

| name | varchar(60) | NO | | NULL | |

| CurClass | varchar(2) | NO | | NULL | |

| Gender | varchar(20) | YES | | NULL | |

| Phone\_number | varchar(13) | NO | PRI | NULL | |

| Address | varchar(100) | YES | | NULL | |

| dob | varchar(12) | NO | | NULL | |

+--------------+--------------+------+-----+---------+-------+

1. Teachers

+--------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------+--------------+------+-----+---------+-------+

| Name | varchar(60) | NO | | NULL | |

| Degree | varchar(150) | NO | | NULL | |

| Subject | varchar(50) | NO | | NULL | |

| Gender | varchar(20) | YES | | NULL | |

| Phone\_number | varchar(13) | NO | PRI | NULL | |

| CBSE\_Number | varchar(30) | YES | | NULL | |

| Address | varchar(100) | YES | | NULL | |

| DOB | varchar(12) | NO | | NULL | |

+--------------+--------------+------+-----+---------+-------+

1. Marks

+----------+---------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+---------------+------+-----+---------+-------+

| Class | int | NO | PRI | NULL | |

| Quarter1 | decimal(8,2) | YES | | NULL | |

| Quarter2 | decimal(8,2) | YES | | NULL | |

| Quarter3 | decimal(8,2) | YES | | NULL | |

| Quarter4 | decimal(8,2) | YES | | NULL | |

| Yearly | decimal(10,2) | YES | | NULL | |

+----------+---------------+------+-----+---------+-------+

1. Fees

+----------+---------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+---------------+------+-----+---------+-------+

| Class | int | NO | PRI | NULL | |

| Quarter1 | decimal(8,2) | YES | | NULL | |

| Quarter2 | decimal(8,2) | YES | | NULL | |

| Quarter3 | decimal(8,2) | YES | | NULL | |

| Quarter4 | decimal(8,2) | YES | | NULL | |

| Yearly | decimal(10,2) | YES | | NULL | |

+----------+---------------+------+-----+---------+-------+