PROJECT REPORT

ON

Student Database Management System

SUBMITTED TO:

CENTRAL BOARD OF SECONDARY EDUCATION

In the partial fulfilment of the requirements

for the award of

AISSCE-2021

UNDER THE GUIDANCE OF:

MR. BALJEET SINGH
HoD
Computer Science Department

SUBMITTED BY:

Divyanshu Chander
Roll - 17692311

CLASS: XII

ARMY PUBLIC SCHOOL CHANDIMANDIR

CERTIFICATE

This is to certify that **Divyanshu Chander** of Class: XII-A has prepared the report on the project entitled "**Student Database Management System**". The project is the result of his efforts and endeavors. The report is found worthy of acceptance as the final project for the Computer Science of Class XII. He has prepared this project under my guidance.

(Mr. Baljeet Singh)
PGT cum HoD
Computer Science

ACKNOWLEDGEMENT

I would like to express a deep sense of thanks to my project guide Mr Baljeet Singh for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice and constant motivation has been responsible for the successful completion of this project.

Last but not the least I would like to thank my parents who helped me a lot in gathering different information, motivating me from time to time in making this project. Despite the busy schedule, they gave different ideas in making this project unique.

Divyanshu Chander

Roll – 17692311

. .

INDEX

INTRODUCTION

Everything nowadays has been digitised, including student management systems. The following project tries to make a program for Student management which involves connectivity of both Python and MySQL using mysql.connector. It enables to set up a new management system with an approach for its wider use.

It enables both Administrators as well as Students to use while making certain restrictions depending upon their user rights.

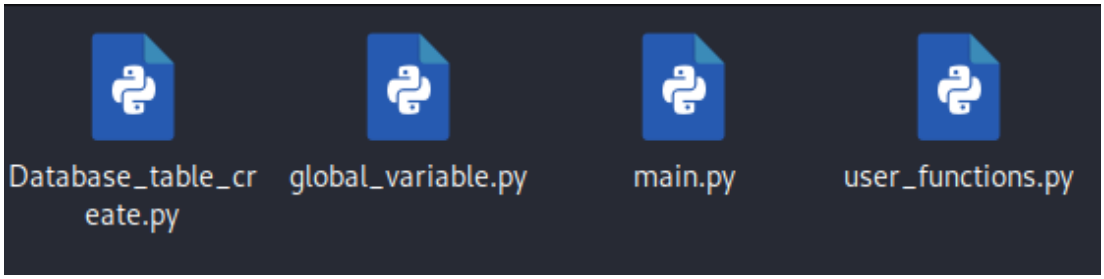
The program provides all the basic features that a data administrator and a student of an organisation would like to use it (program) for.

The operations of adding a student, viewing data, deleting and updating student and fee records, and many more small handy features make this program versatile in many respects.

Moreover, it has been tried to make this project and program simple so that any first time user will be able to understand, manipulate, add and use its features.

Framework of program

The Student Management system involves four python file:



1. **main.py:** It involves the main framework on which the whole program runs including loops, checks,etc.

2. **Database_table_create.py:** It is used by the first time user setting up the management system, making and connecting organisation's database and the required tables.

3. **user_functions.py:** It includes each and every program functions that are being used by the main framework of the program. It is customisable in terms of adding, removing any undesired function that the user might or might not want.

4. **global_variable.py:** It holds several database parameters in the form of variables such as host, user, password, database name and organisation name. These all come handy in the main framework of the program for authentication (password) from administrator, etc.

CODES

1. Database_create_table.py

```
# for sql table creation & table related constants
import mysql.connector as sql
fh = open('global_variable.py', 'w+')
input('Welcome! To create your database: (Press ENTER)')

while True:
        try:
                host = input("Please Enter host name:\t")
                fh.write("host = '{}'\n".format(host))
                user = input('Enter User_Name:\t')
                fh.write("user = '{}'\n".format(user))
                passwd = input('Enter password:\t\t')
                fh.write("passwd = '{}'\n".format(passwd))
                print('Please Wait...')
                connection = sql.connect(host = host, user = user, passwd  = passwd)
                if connection.is_connected():
                        print('Your Database system is connected!!!')
                        cursor = connection.cursor()
                        break
        except:
                end =input('''Try Again: y
Exit:   n\n''').lower()
                if end == 'n':
                        exit()
                continue
organisation_name = input('Enter Name of Organisation(use underscore for space):\t')
fh.write("organisation_name = '{}'\n".format(organisation_name))
database_name = "{}_Student_management".format(organisation_name)
fh.write("database_name = '{}'\n".format(database_name))
while True:
        try:
                query = 'create database {}'.format(database_name)
                cursor.execute(query)
                break
        except:
                print('''Database already exists''')
                exit()
print('Please Wait...')
query = 'use {}'.format(database_name)

cursor.execute(query)
cursor.execute('''create table student_details_basic (Admission_No integer primary key,
        First_name varchar(30) not null,
        Last_name varchar(30) not null,
        Class varchar(5),
        Rollno integer,
        Father_Name varchar(30),
        Mother_Name varchar(30),
        Phone_number bigint,
        Email_id varchar(30)) ''')
connection.commit()

cursor.execute('''create table Login_credentials (Admission_No int(11), foreign key (Admission_No) references
student_details_basic (Admission_No) on delete cascade on update cascade,
         Phone_number bigint,
        passwd varchar(30) not null)''')
connection.commit()
```

```
cursor.execute('''create table student_grades (Admission_No int(11), foreign key (Admission_No) references student_details_basic
(Admission_No)
                                on delete cascade on update cascade,Physics varchar(3) default null,
                                Maths varchar(3) default null, Chemistry varchar(3) default null,
                                English varchar(3) default null, Computer_science varchar(3) default null,
                                PHE varchar(3) default
null)''')
connection.commit()

cursor.execute('''create table attendance (Admission_No int(11),foreign key (Admission_No) references student_details_basic
(Admission_No)
                                on delete cascade on update cascade,
                                Days_Present integer, Days_Absent integer)''')
connection.commit()
cursor.execute('''create table fee (Admission_No int(11),foreign key (Admission_No) references student_details_basic
(Admission_No)
                                on delete cascade on update cascade,
                                fee_due integer, fee_total integer, fee_paid tinyint(1))''')# boolean value to check if fee paid
or not
connection.commit()

print('Your Tables are successfully created')
fh.close()
connection.close()
```

Output:

```
Welcome! To create your database: (Press ENTER)
Please Enter host name: localhost
Enter User_Name:        kali
Enter password:         kali
Please Wait ...
Your Database system is connected!!!
Enter Name of Organisation(use underscore for space):   army_school
Please Wait ...
Your Tables are successfully created
```

- It also creates the **global_variable.py** file.

MySQL Tables:

```
atabase changed
ariaDB [army_school_Student_management]> show tables;
+-----------------------------------------+
| Tables_in_army_school_Student_management |
+-----------------------------------------+
| Login_credentials                       |
| attendance                              |
| fee                                     |
| student_details_basic                   |
| student_grades                          |
+-----------------------------------------+
rows in set (0.001 sec)
```

Table Layouts

- **student_details_basic:**

```
MariaDB [army_school_Student_management]> desc student_details_basic;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Admission_No | int(11)     | NO   | PRI | NULL    |       |
| First_name   | varchar(30) | NO   |     | NULL    |       |
| Last_name    | varchar(30) | NO   |     | NULL    |       |
| Class        | varchar(5)  | YES  |     | NULL    |       |
| Rollno       | int(11)     | YES  |     | NULL    |       |
| Father_Name  | varchar(30) | YES  |     | NULL    |       |
| Mother_Name  | varchar(30) | YES  |     | NULL    |       |
| Phone_number | int(11)     | YES  |     | NULL    |       |
| Email_id     | varchar(30) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

- **Login_credentials:**

```
MariaDB [army_school_Student_management]> desc Login_credentials;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Admission_No | int(11)     | YES  | MUL | NULL    |       |
| Phone_number | bigint(20)  | YES  |     | NULL    |       |
| passwd       | varchar(30) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.001 sec)
```

- **Fee:**

```
+--------------+------------+------+-----+---------+-------+
| Field        | Type       | Null | Key | Default | Extra |
+--------------+------------+------+-----+---------+-------+
| Admission_No | int(11)    | YES  | MUL | NULL    |       |
| fee_due      | int(11)    | YES  |     | NULL    |       |
| fee_total    | int(11)    | YES  |     | NULL    |       |
| fee_paid     | tinyint(1) | YES  |     | NULL    |       |
+--------------+------------+------+-----+---------+-------+
4 rows in set (0.001 sec)
```

- **student_grades:**

```
MariaDB [army_school_Student_management]> desc student_grades;
+------------------+------------+------+-----+---------+-------+
| Field            | Type       | Null | Key | Default | Extra |
+------------------+------------+------+-----+---------+-------+
| Admission_No     | int(11)    | YES  | MUL | NULL    |       |
| Physics          | varchar(3) | YES  |     | NULL    |       |
| Maths            | varchar(3) | YES  |     | NULL    |       |
| Chemistry        | varchar(3) | YES  |     | NULL    |       |
| English          | varchar(3) | YES  |     | NULL    |       |
| Computer_science | varchar(3) | YES  |     | NULL    |       |
| PHE              | varchar(3) | YES  |     | NULL    |       |
+------------------+------------+------+-----+---------+-------+
```

2. global_variable.py(automatically created)

```
host = 'localhost'
user = 'kali'
passwd = 'kali'
organisation_name = 'army_school'
database_name = 'army_school_Student_management'
```

3. user_functions.py

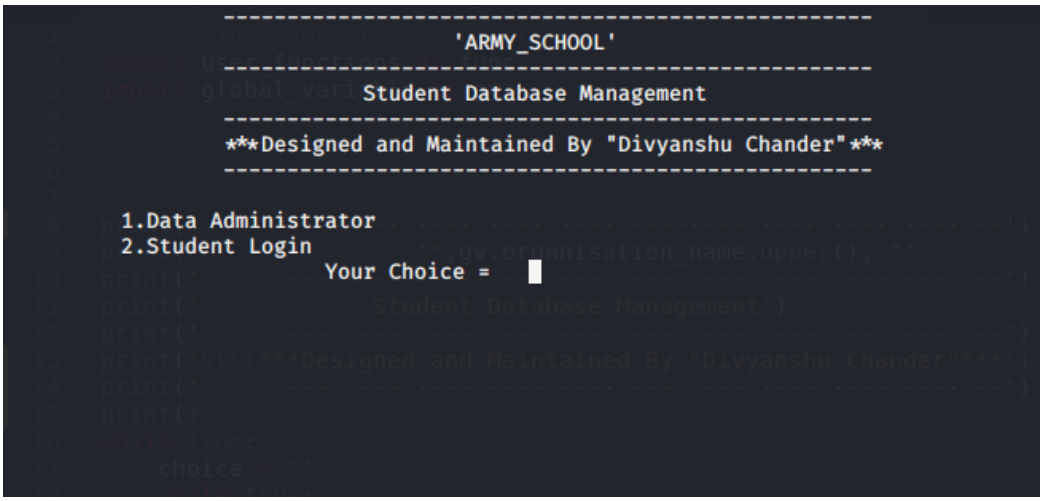
```
  1  #Various framework functions
  2  def sqlconnect():# To connect sql with a cursor ...
  7
  8  def adminmenu():#Administrator menu ...
 24
 25  def end():#program exit statement ...
 29
 30  def showdatabases(): ...
 37
 38  def adminadd(): ...
 77
 78  def admindelete(admiss): ...
 99
100  def adminupdate(): ...
156
157  def mysqlrun(): ...
186
187  def newoldcheck(admiss): ...
199
200  def oldstulog(admiss): ...
223
224  def newstulog(admiss): ...
238
239  def gradecheck(admiss): ...
252
253  def feecheck(admiss): ...
278
279  def basicinfo(admiss): ...
303
304  def passupdate(admiss): ...
314
315  def student(admiss): ...
343
344  def adminsearch(): ...
374
375  def fulldetails(admiss): ...
405
```

4. Main.py(_main_)

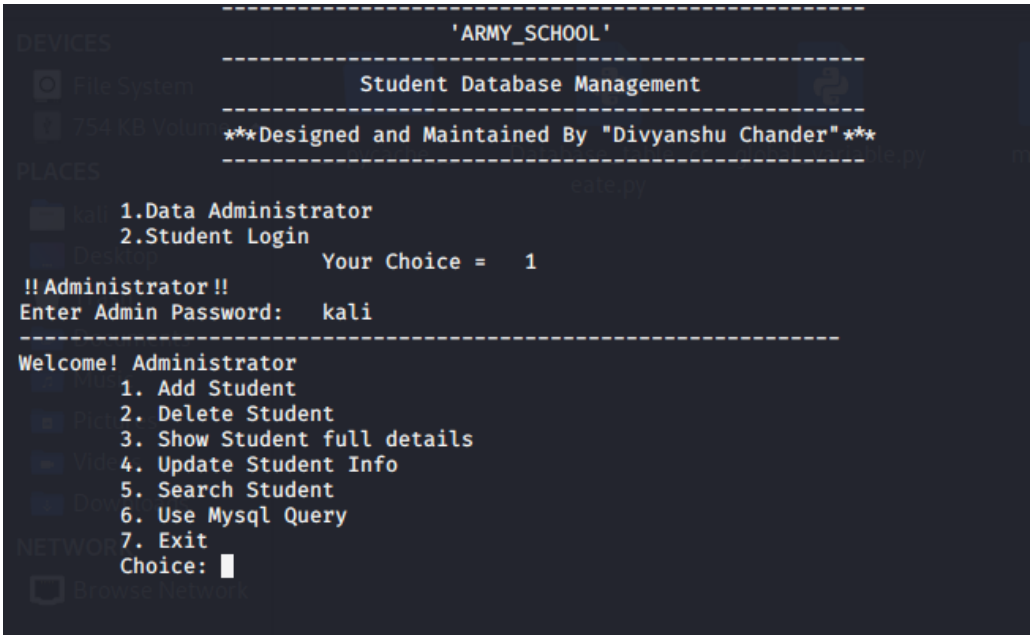
```
# the main framework for programming
import user_functions as func
import global_variable as gv



print('         ----------------------------------------------------')
print("                          '",gv.organisation_name.upper(), "'                    ",sep ='')
print('         ----------------------------------------------------')
print('                    Student Database Management')
print('         ----------------------------------------------------')
print('\t\t***Designed and Maintained By "Divyanshu Chander"***')
print('         ----------------------------------------------------')
print()
while True:
        choice = ''
        while True:
                choice = input('''      1.Data Administrator
        2.Student Login
                        Your Choice =\t''')
                if choice != '1' and choice!='2':
                        print("invalid input- TRY AGAIN")
                        continue
                break
        if choice == '1':
                print('!!Administrator!!')
                adminpass = input('Enter Admin Password:\t')
                if adminpass == gv.passwd:
                        while True:
                                n = func.adminmenu()
                                if n == '1':
                                        func.adminadd()
                                elif n=='2':
                                        print("Caution: It will delete student records")
                                        no = input("Enter admission no:\t ")
                                        func.admindelete(no)
                                elif n == '3':
                                        admiss = input('Enter Admission Number:\t')
                                        func.fulldetails(admiss)
                                elif n=='4':
                                        func.adminupdate()
                                elif n=='5':
                                        func.adminsearch()
                                elif n=='6':
                                        func.mysqlrun()
                                else:
                                        print('invalid choice')
                else:
                        print('Invalid Password')
                        print()
                        continue


        elif choice =='2':

                print('!!Student Login!!')
                print("Welcome to Student Database!!!")
                admission = input('Enter your Admission Number:\t')
                a = func.newoldcheck(admission)

                if a is True:
                        print('Welcome student')
                        b = func.oldstulog(admission)
                        if b is True:
                                func.student(admission)


                elif a is False:
                        print('New Student Registration')
                        func.newstulog(admission)
                        func.student(admission)
```

Output

Main Screen:



1. **Data Administrator:** It asks the user for password which is the same as that of user password stored in global_variable.py and only proceeds after the authentication has been done or else it would return back to the Main Menu.



- Adding student: using **user_functions.adminadd()**

```
def adminadd():
        print('-'*65)
        sql=sqlconnect()
        cursor = sql.cursor()
        print('Enter the following details:')
        while True:

                admin = input('Enter admissionno:\t')
                fname = input('Enter first name:\t')
                lname = input('Enter last name:\t')
                Class = input('Enter class Assigned(like VIII,XI,etc):\t')
                Class.upper()
                roll = int(input('Enter Roll No:\t'))
                pfname = input("Enter Father's name:\t")
                pmname = input("Enter Mother's name:\t")
                phno = input("Enter Phone No.(mandatory):\t")
                email = input('Enter your Email-id:\t')
                a = (admin,fname,lname,Class,roll,pfname,pmname,phno,email)
                query = "insert into student_details_basic values(%s,'%s','%s','%s',%s,'%s','%s',%s,'%s')"%a
                cursor.execute(query)
                query = "insert into student_grades(Admission_No) values({})".format(admin)
                cursor.execute(query)
                query = "insert into fee values({},0,0,1)".format(admin)
                cursor.execute(query)
                print('Student Added')
                sql.commit()

                print()
                loop = input('''        1.Add Again
        2.Main menu
        3.Exit          :\t''')
                if loop =='1':
                        continue
                elif loop =='2':
                        break
                elif loop =='3':
                        end()
        sql.commit()
        sql.close()
```

Output:

```
-----------------------------------------------------------------
Enter the following details:
Enter admissionno:      1234
Enter first name:       Divyanshu
Enter last name:        Chander
Enter class Assigned(like VIII,XI,etc): XII
Enter Roll No:  49
Enter Father's name:    abcd
Enter Mother's name:    qwerty
Enter Phone No.(mandatory):     12345678
Enter your Email-id:    abcd@gmail.com
Student Added

        1.Add Again
        2.Main menu
        3.Exit          :
```

```
MariaDB [army_school_Student_management]> select * from student_details_basic;
+-------------+------------+-----------+-------+--------+-------------+-------------+--------------+----------------+
| Admission_No | First_name | Last_name | Class | Rollno | Father_Name | Mother_Name | Phone_number | Email_id       |
+-------------+------------+-----------+-------+--------+-------------+-------------+--------------+----------------+
|        1234 | Divyanshu  | Chander   | XII   |     49 | abcd        | qwerty      |     12345678 | abcd@gmail.com |
+-------------+------------+-----------+-------+--------+-------------+-------------+--------------+----------------+
```

MySQL: It also adds entries in other tables as well.

- Deleting Student: using **user_functions.admindelete()**

```
def admindelete(admiss):
        while True:
                sql = sqlconnect()
                cursor = sql.cursor()
                cursor.execute("select First_name,Last_name from student_details_basic where Admission_No = {}".format(admiss))
                a = cursor.fetchone()
                cursor.execute("delete from student_details_basic where Admission_No ={}".format(admiss))
                sql.commit()
                print("AdmissionNo",admiss,"successfully deleted whose details are\n",a)
                print()
                loop = input('''        1.Remove Again
        2.Main menu
        3.Exit          :\t''')
                if loop =='1':
                        continue
                elif loop =='2':
                        break
                elif loop =='3':
                        end()
        sql.commit()
        sql.close()
```

Output:

```
Caution: It will delete student records
Enter admission no:      100
AdmissionNo 100 successfully deleted whose details are
 ('Random', 'Name')

        1.Remove Again
        2.Main menu
        3.Exit          :
```

- Searching for a student : **user_function.adminsearch()**

```
def adminsearch():
        sql = sqlconnect()
        cursor = sql.cursor()
        searchdomain_intial = {'Admission_No':'','First_name':'','Father_name':'','Mother_name':'','Phone_number':'','Email':''}
        searchdomain_final = {}
        for d in searchdomain_intial:
                a = input("Enter {} (Else skip):\t".format(d))
                if a == '':
                        continue
                else:
                        searchdomain_final[d] = a
        result = []
        rowcount = 0
        res = []

        for n in searchdomain_final:
                cursor.execute("select*from student_details_basic where {} Like '%{}%' ".format(n, searchdomain_final[n]))
                a = cursor.fetchall()
                result.extend(a)
        for x in result:
                if x not in res:
                        res.append(x)
        result = res


        print(searchdomain_final.keys())
        for s  in result :
                print(s)
```

Output:

```
Enter Admission_No (Else skip): 1234
Enter First_name (Else skip):
Enter Father_name (Else skip):
Enter Mother_name (Else skip):
Enter Phone_number (Else skip):
Enter Email (Else skip):
dict_keys(['Admission_No'])
(1234, 'Divyanshu', 'Chander', 'XII', 49, 'abcd', 'qwerty', 123456, 'abcd@gmail.com')
----------------------------------------------------------------
```

Updating Student Info: It uses a series of multiple **user_functions.adminupdate()**

```
def adminupdate():
        sql = sqlconnect()
        cursor = sql.cursor()
        choice = input('''select option to update:
                1.Student information
                2.New fee update

                                                :
                ''')
        if choice == '1':
                print("STUDENT INFORMATION UPDATE")
                admno = input("Enter Student's Admission No:\t")

                print('Select From the following options to update:')
                opt = input('''1.Update Phone number
                                        2Update E-mail ID
                                        3.Update Student's Grades''')
                if opt == '1':
                        phno = input("Enter New Phone number:\t")
                        cursor.execute("update student_details_basic set Phone_number = {} where Admission_No
={}".format(phno,admno))
                        print('Phone number updated')
                elif opt =='2':
                        email = input("Enter New Email-ID:\t")
                        cursor.execute("update student_details_basic set Email_id = '{}' where Admission_No
={}".format(email,admno))
                        print('Email-ID updated')
                elif opt == '3':
                        x = input('''Do you have
                                1.Computer Science
                                2.Physical Education\n:\t''')
                        if x =='1':
                                subjects = ['Physics','Maths','Chemistry','English','Computer_science']
                        elif x=='2':
                                subjects = ['Physics','Maths','Chemistry','English','PHE']
                        for a in subjects:
                                marks = input('Enter {} Grades/%:\t'.format(a))
                                cursor.execute("Update student_grades set {} = '{}' where Admission_No =
{}".format(a,marks,admno))
                        print("GRADES UPDATED!")
                        sql.commit()




        elif choice == '2':#### LOGIC TO BE WRITTEN
                newamount = input("Enter monthly amount to update fee database:\t")

                cursor.execute("update fee set fee_due = {} where fee_paid = 1".format(newamount))
                sql.commit()
                cursor.execute("update fee set fee_paid = 0 where fee_paid = 1")
                sql.commit()
                cursor.execute("update fee set fee_due = fee_due+{} where fee_paid = 0".format(newamount))
                sql.commit()
        sql.commit()
        sql.close() #LEFT
```

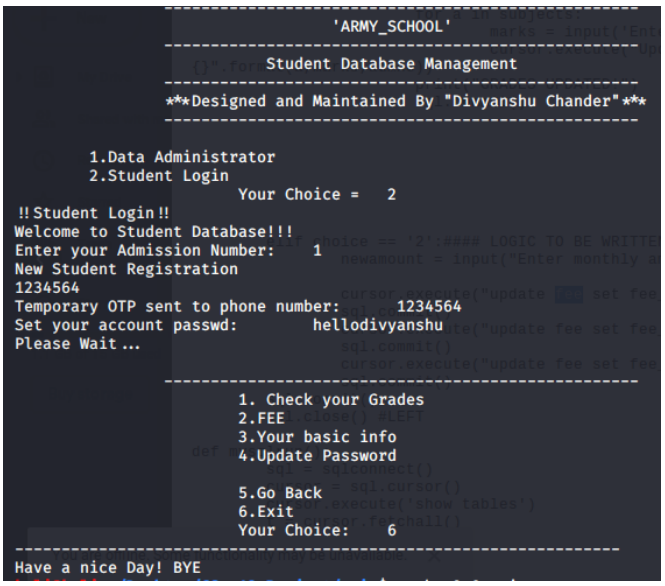
```
select option to update:
                1.Student information
                2.New fee update

                                                    :
                1
STUDENT INFORMATION UPDATE
Enter Student's Admission No:   1
Select From the following options to update:
1.Update Phone number
                                                2Update E-mail ID
                                                3.Update Student's Grade
Do you have
                                    1.Computer Science
                                    2.Physical Education
:       1
Enter Physics Grades/%: 99
Enter Maths Grades/%:   99
Enter Chemistry Grades/%:       95
Enter English Grades/%: 96
Enter Computer_science Grades/%:        98
GRADES UPDATED!
------------------------------------------------------------------------
```

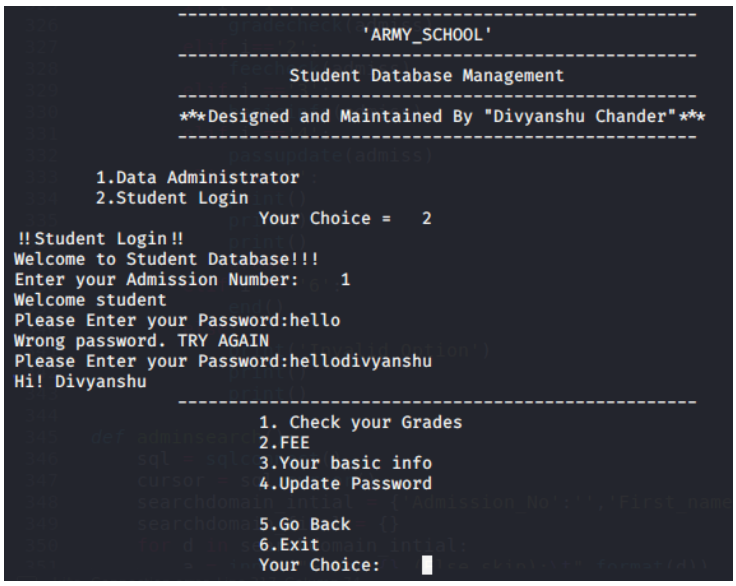
2. Student Login: For student login it follows a general login or sign in process where student's logging in for the first time are required to set up their password after a depicted OTP(on provided number in database) authentication. Old students already having their account are just required to enter their password that they have set up during their first time registration.

```
def newstulog(admiss):
        sql = sqlconnect()
        cursor = sql.cursor()
        cursor.execute('select Phone_number from student_details_basic where Admission_No ={}'.format(admiss))
        ph = cursor.fetchall()
        a = ph[0][0]
        print(a)
        print('Temporary OTP sent to phone number:\t', a)
        passwd = input('Set your account passwd:\t')
        cursor.execute("insert into Login_credentials values({},{},'{}')".format(admiss,a,passwd))
        print('Please Wait... ')
        sql.commit()
        sql.close()
        print()
```

```
def oldstulog(admiss):

        sql = sqlconnect()
        cursor = sql.cursor()
        cursor.execute('Select passwd from Login_credentials where Admission_No = {}'.format(admiss))
        a = cursor.fetchone()
        for i in a:
                p = i
        cursor.execute('Select First_name from student_details_basic where Admission_No = {}'.format(admiss))
        n=cursor.fetchone()
        sql.close()
        for i in n:
                name = i
        while True:
                passwd = input('Please Enter your Password:')


                if p == passwd:
                        print('Hi!',name)
                        return True

                else:
                        print('Wrong password. TRY AGAIN')
```

The above codes are from user_functions.py





New student Login **Old Student Login**

The **password** gets stored in **Login_credentials Table** along with phone number:

```
MariaDB [army_school_Student_management]> select*from Login_credentials;
+--------------+--------------+-----------------+
| Admission_No | Phone_number | passwd          |
+--------------+--------------+-----------------+
|            1 |      1234564 | hellodivyanshu  |
+--------------+--------------+-----------------+
1 row in set (0.001 sec)
```

Student functions:

1. Checking grades
2. Paying/viewing Fee
3. Updating details and account password

Checking Grades:

Using **user_functions.gradecheck()**

```
def gradecheck(admiss):
        sql = sqlconnect()
        cursor = sql.cursor()
        cursor.execute('select * from student_grades where Admission_No = {}'.format(admiss))
        a = cursor.fetchall()
        print('''                        +-------+-------+-----------+---------+------------------+------+---------
                        | Physics | Maths | Chemistry | English | Computer_science | PHE  |
                        +---------+-------+-----------+---------+------------------+------+
                        |  {}     | {}    |     {}    |   {}    |        {}        |  {} |
                        +---------+-------+-----------+---------+------------------+------+
        '''.format(a[0][1],a[0][2],a[0][3],a[0][4],a[0][5],a[0][6]))
        print('')
        sql.close()
```

Output:

```
Hi! Divyanshu
        ------------------------------------------------
                1. Check your Grades
                2.FEE
                3.Your basic info
                4.Update Password

                5.Go Back
                6.Exit
                Your Choice:    1
                +-------+-------+-----------+---------+------------------+------+------
                | Physics | Maths | Chemistry | English | Computer_science | PHE  |
                +---------+-------+-----------+---------+------------------+------+
                |  98     | 99    |     95    |   96    |        99        |  None |
                +---------+-------+-----------+---------+------------------+------+
```

Paying/Viewing Fee

Checks whether the fee has been paid or not and if not asks if you want to pay fee now or else skip.

```
    def feecheck(admiss):
            sql = sqlconnect()
            cursor = sql.cursor()
            cursor.execute('select*from fee where Admission_No = {}'.format(admiss))
            a = cursor.fetchall()
            a=a[0]
            fdue  = a[1]
            ftotal = a[2]
            fpaid = a[3]
            if fpaid == 0:
                    print("                          YOU HAVE NOT PAID THE FEE")
                    print("Total Due Amount:\t", fdue)
                    pay = input('1.To pay fee(else skip):\t')
                    if pay == '1':
                            print('Please Wait... Processing Payment')
                            cursor.execute('update fee set fee_due = 0 where Admission_No = {}'.format(admiss))
                            sql.commit()
                            cursor.execute('update fee set fee_paid = 1 where Admission_No = {}'.format(admiss))
                            sql.commit()
                            print('Fee of', fdue, 'has been paid.\n ThankYou!')

            elif fpaid == 1:
                    print('Fee of has been paid')
```

Output:

```
                        1. Check your Grades
                        2.FEE
                        3.Your basic info
                        4.Update Password

                        5.Go Back
                        6.Exit
                        Your Choice:    2
                                YOU HAVE NOT PAID THE FEE
Total Due Amount:         10000
1.To pay fee(else skip):        1
Please Wait ...  Processing Payment
Fee of 10000 has been paid.
 ThankYou!
```

Viewing/Updating Details

```
def basicinfo(admiss):
        sql = sqlconnect()
        cursor = sql.cursor()
        cursor.execute('select * from student_details_basic where Admission_No = {}'.format(admiss))
        a = cursor.fetchall()
        print(a)
        ph = input('''1. Want to Update Phone number
                2. Want to update Email(else skip):\t''')
        if ph =='1':
                newnumber = input('Please Enter your new Phone number:\t')
                cursor.execute('update student_details_basic set Phone_number ={} where Admission_No
={}'.format(newnumber,admiss))
                sql.commit()
                cursor.execute('update Login_credentials set Phone_number ={} where Admission_No ={}'.format(newnumber,admiss))
                sql.commit()
                print('Information updated!!!')
                print()
        elif ph =='2':
                newmail = input('Please Enter your new Email:\t')
                cursor.execute("update student_details_basic set Email_id ='{}' where Admission_No ={}".format(newmail,admiss))
                print('Information updated!!!')
                print()
                sql.commit()
        sql.close()
        print()
```

Output:

```
[(1, 'Divyanshu', 'Chander', 'XII', 49, 'Mr Chander', 'Mrs Chander', 1234564, 'divyanshu@gmail.com')]
1. Want to Update Phone number
                2. Want to update Email(else skip):     2
Please Enter your new Email:    newemail@gmail.com
Information updated!!!
```

Exiting Program

Using user_functions.end()

```
def end():#program exit statement
        print('-'*65)
        print('Have a nice Day! BYE')
        exit()
```

Output:

```
                    1. Check your Grades
                    2.FEE
                    3.Your basic info
                    4.Update Password

                    5.Go Back
                    6.Exit
                    Your Choice:    6
-----------------------------------------------------------------
Have a nice Day! BYE
```

Backend Tables

1. Student_details_basic

Admission_No	First_name	Last_name	Class	Rollno	Father_Name	Mother_Name	Phone_number	Email_id
1	Divyanshu	Chander	XII	49	Mr Chander	Mrs Chander	1234564	divyanshu@gmail.com
7	Harry	Potter	XII	7	James Potter	Lily Potter	123456	theboywhosurvived@gmail.com
1121	Ankit	Sharma	XII	12	Mr Sharma	Mrs Sharma	123123	ankit@gmail.com
1122	Narayanan	Chari	XII	20	N Chari	Mrs Chari	11223344	narayanan@gmail.com
1222	Sahil	Bhardwaj	XII	32	Mr Bhardwaj	Mrs Bhardwaj	12121	sahilbhardwaj@gmail.com
1278	Random	Singh	XII	11	Mr Probability	Mrs Probability	13123213	probabilityparivar@gmail.com
5258	Tenet	Nolan	XII	1001	Mr Christopher Nolan	Mrs Nolan	1001	tenet@gmail.com
14785	Baburao	Ganpatrao Apte	XII	18	Mr Apte	Mrs Apte	123421	memelegend@gmail.com
123321	Ansh	Mishra	XII	14	Mr Mishra	Mrs Mishra	12345678	ansh@gmail.com
1236598	Katherine	Langford	XII	1	Mr Langford	Mrs Langford	123214	katherine@gmail.com

10 rows in set (0.000 sec)

2. Login_credentials

```
MariaDB [army_school_Student_management]> select*from Login_credentials;
+--------------+--------------+----------------+
| Admission_No | Phone_number | passwd         |
+--------------+--------------+----------------+
|            1 |      1234564 | hellodivyanshu |
+--------------+--------------+----------------+
```

3. Fee Table

```
MariaDB [army_school_Student_management]> select*from fee;
+--------------+---------+-----------+----------+
| Admission_No | fee_due | fee_total | fee_paid |
+--------------+---------+-----------+----------+
|         1122 |   10000 |         0 |        0 |
|         1222 |   10000 |         0 |        0 |
|         1121 |   10000 |         0 |        0 |
|       123321 |   10000 |         0 |        0 |
|         1278 |   10000 |         0 |        0 |
|      1236598 |   10000 |         0 |        0 |
|         5258 |   10000 |         0 |        0 |
|            7 |   10000 |         0 |        0 |
|        14785 |   10000 |         0 |        0 |
|            1 |       0 |         0 |        1 |
+--------------+---------+-----------+----------+
```

4.student_grades

```
MariaDB [army_school_Student_management]> select * from student_grades;
+--------------+---------+-------+-----------+---------+------------------+------+
| Admission_No | Physics | Maths | Chemistry | English | Computer_science | PHE  |
+--------------+---------+-------+-----------+---------+------------------+------+
|         1122 | 45      | 89    | 547       | 98      | NULL             | 96   |
|         1222 | 78      | 79    | 48        | 796     | 82               | NULL |
|         1121 | 47      | 86    | 91        | 73      | NULL             | 91   |
|       123321 | 91      | 76    | 84        | 65      | 88               | NULL |
|         1278 | 78      | 94    | 666       | 77      | NULL             | 79   |
|      1236598 | 92      | 94    | 95        | 99      | 97               | NULL |
|         5258 | 88      | 86    | 84        | 89      | NULL             | 79   |
|            7 | 89      | 89    | 79        | 88      | 85               | NULL |
|        14785 | 78      | 97    | 64        | 55      | 44               | NULL |
|            1 | 98      | 99    | 95        | 96      | 99               | NULL |
+--------------+---------+-------+-----------+---------+------------------+------+
10 rows in set (0.001 sec)
```

Bibliography:

- Python.org
- Computer Science with Python-Sumita Arora
- Stackoverflow

print(“THANKYOU!!!”)