

**LOKMANYA TILAK INTERNATIONAL SCHOOL**

**COMPUTER SCIENCE**

**INVESTIGATORY PROJECT**

**ON**

**STUDENT MANAGEMENT**

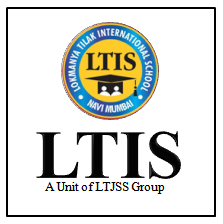
**PORTAL**

**SUBMITTED BY: AYUSH BADGUJAR**

**CLASS: XII**

**CONTENT**

|  |  |  |
| --- | --- | --- |
| **SR.NO.** | **TOPIC** | **PAGE NO.** |
| **1** | **Cover Page** | **1** |
| **2** | **Content** | **2** |
| **3** | **Certificate** | **3** |
| **4** | **Acknowledgement** | **4** |
| **5** | **Objective** | **5** |
| **6** | **Requirements** | **6** |
| **7** | **Introduction** | **7** |
| **8** | **Working** | **8-10** |
| **9** | **Source Code** | **11-17** |
| **10** | **Output** | **18-22** |
| **11** | **Description of tables (variable table)** | **23** |
| **12** | **Conclusion** | **24** |
| **13** | **Bibliography** | **25** |

****

**LOKMANYA TILAK INTERNATIONAL SCHOOL**

**CERTIFICATE**

**This is to certify that the work depicted in this Project is the work of  
AYUSH BADGUJAR of class XII having CBSE Roll no. \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**has satisfactorily completed the “COMPUTER SCIENCE INVESTIGATORY PROJECT” on the topic “STUDENT MANAGEMENT PORTAL” for All India Senior School Certificate Examination (AISSCE) for the Academic year 2022-23.**

***INTERNAL EXAMINER EXTERNAL EXAMINER PRINCIPAL***

***Ms. Ritu Kapur***

****

**ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my **Computer Science** teacher **Ms. Poonam Gupta** as well as our Principal Madam **Ms. Ritu Kapur** who gave me the golden opportunity to do this wonderful project on the topic **“STUDENT MANAGEMENT PORTAL”** which also helped me in doing a lot of research and enhanced my analytical and critical thinking skills. Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project in given time frame.

**OBJECTIVE**

The aim of the project is to apply the basic knowledge of the python language and MySQL work bench learnt, for writing a program which would manage students data in which one can enter there personal details and get their fee status given to school.

**REQUIREMENTS**

**SOFTWARE**

1. PyCharm Community Edition 2020.2.3 x64 and MySQL Workbench 8.0 CE for writing and executing the program.
2. MS word for writing project report.

**HARDWARE**

1. Modern Operating System: Windows 7 or higher.
2. x86 64-bit CPU (Intel/ AMD architecture)
3. Minimum 4GB RAM
4. 5 GB free disk space
5. Printer for taking printout of the report.

**INTRODUCTION**

The emergence of computers has impacted all the spheres of our lives in a big way. This holds true in case of managing details of large number of students in the school. Keeping an account of thousands of students in a school manually would be an arduous task and may involve mistakes also. But with the help of computers these errors are avoided by allowing the system to keep track of information such as student’s personal details, fee paid till now, fee to be paid, and even school’s basic information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes. Thus this system allows smooth functioning of student data by eliminating chances of errors.

**WORKING**

This student management portal can faultlessly manage and store student’s information electronically according to what student has given to school. The system would help the principal and registrar to keep a constant track of all the new students entering the school and their fee status accordingly. Apart from checking one’s data one can register his/her name to schools student list, and update their profile in future, also allowing one to pay fee and get their payment confirmation right away. Hence, having student management portal is a dire need of all the schools and colleges.

**WORKING WITH STUDENT MANAGEMENT PORTAL CODE**

1. MySQL connector is imported as c and connection is established by the using host, password, and user of MySQL workbench and data base that is to be accessed.
2. Colorama is imported to get colored text which help us to write important texts in different colors so that user don’t miss them and hence providing a better feedback from the program.
3. Time and os are imported for small uses in 2-3 places to give a real life experience through the program.
4. If-else condition is used to check if the connection is successful to get into the system of Student Manager
5. welcome() function is defined in order to greet the user and provide a better interface. It uses basic coding to print STUDENT MANAGEMENT PORTAL with ‘#’s surrounding it.
6. In order to put all the functions of the Student Manager, a function menu() is defined and if-elif and print functions are employed to execute it.
7. To login into our student account, student\_login() function is made and it takes information like password and roll no. from the user to go to further code.
8. To add or register our name in school’s list, add\_student() function is made which takes important information like name, section, roll no. , date of birth, address, phone number, password.
9. To verify if the password entered 2 twice is same or not and to save the data to the database, password\_verification1() function is used.
10. To see school’s basic information, about\_school() function is made which allows the user to see school’s moto, director’s message etc.
11. After logging in, to see all the functions inside it, inside\_student\_login() function which shows the options update profile, deposit fee and log out using if-elif-else statements.
12. To update students profile, update\_profile() function is used which takes all the important information from the user again in which the user can change the wrong data inputted before.
13. It again verifies password that user entered twice and saves the updated data using password\_verification2() function.
14. To deposit fee to the school, deposite\_fee() function which shows the fee pending and fee paid till now. It also allows the user to pay the fee and get it confirmed in under 10 seconds.

**SOURCE CODE**

Python Source Code

**import** mysql.connector **as** c  
**from** colorama **import** Fore,Style  
**import** time  
**import** os  
  
con=c.connect(host=**"localhost"**, user=**"root"**, passwd=**"The\_0tt0m0n\_M0nkey"**,database=**"student\_management\_system"**)  
**if** con.is\_connected():  
 print(**"Successfully connected"**)  
**else**:  
 print(**"Problem"**)  
  
mycursor = con.cursor()  
  
  
**def** welcome():  
 print(**"#"**\*188)  
 print(**f"{**Fore.RED**}{**Style.BRIGHT**}LTIS, NAVI MUMBAI, STUDENT MANAGEMENT PORTAL{**Style.RESET\_ALL**}"**.center(201, **"\*"**))  
 print(**"#"**\*188, **"\n\n"**)  
  
  
**def** menu():  
 print(**"Please enter the number given below to access their respective options"**)  
 print(**"1 : STUDENT LOGIN"**)  
 print(**"2 : ADD STUDENT"**)  
 print(**"3 : ABOUT SCHOOL"**)  
 print(**"4 : EXIT WEBSITE"**)  
 choice = int(input(**"Enter the number to proceed : "**))  
 **if** choice == 1:  
 student\_login()  
 **elif** choice == 2:  
 add\_student()  
 **elif** choice == 3:  
 about\_school()  
 **elif** choice == 4:  
 print(**f"\n\n\n\n{**Fore.BLUE**}{**Style.BRIGHT**}THANKYOU FOR USING THE PORTAL !!!{**Style.RESET\_ALL**}"**)  
 **else**:  
 print(**"Please enter numbers from 1 to 4 only.\n\n\n\n"**)  
 menu()  
  
  
**def** student\_login():  
 print(**"Please enter student's roll number and the password to login the portal."**)  
 **global** roll, password  
 roll = int(input(**"STUDENTS ROLL NO. : "**))  
 password = input(**"PASSWORD WITH WHICH YOU HAVE REGISTERED : "**)  
 stu\_pw = **f"""select pw from information where rollno = {**roll**}"""** mycursor.execute(stu\_pw)  
 pwd = mycursor.fetchone()[0]  
 con.commit()  
 **if** password == pwd:  
 print(**f"\n{**Fore.BLUE**}Password is correct, login successful.{**Style.RESET\_ALL**}"**)  
 inside\_student\_login()  
  
 **else**:  
 print(**f"\n{**Fore.BLUE**}Password or roll number is wrong.{**Style.RESET\_ALL**}"**)  
 student\_login()  
  
  
**def** add\_student():  
 **global** name, cls, sec, dob, adrs, phone, pw, apw, rollno, record, record2  
 record = []  
 record2 = []  
 print(**"\nPlease enter the following details to add student information to school's data.\n"**)  
 name = input(**"NAME OF STUDENT : "**)  
 rollno = int(input(**"ROLL NUMBER OF STUDENT : "**))  
 cls = int(input(**"CLASS OF STUDENT (In integers only please): "**))  
 sec = input(**"SECTION OF STUDENT : "**)  
 dob = input(**"DATE OF BIRTH OF STUDENT DDMMYYYY : "**)  
 adrs = input(**"ADDRESS OF STUDENT : "**)  
 phone = int(input(**"PHONE NUMBER OF STUDENT : "**))  
 pw = input(**"ENTER THE PASSWORD WITH WHICH YOU CAN ACCESS THE PORTAL : "**)  
 apw = input(**"PLEASE ENTER THE PASSWORD AGAIN : "**)  
 password\_verification1()  
  
  
**def** password\_verification1():  
  
 **if** apw==pw:  
 print(**"\nSTUDENT REGISTRATION SUCCESSFULL , PLEASE LOGIN TO PROCEED ."**)  
 record.append(name)  
 record.append(rollno)  
 record.append(cls)  
 record.append(sec)  
 record.append(pw)  
 record.append(adrs)  
 record.append(phone)  
 record.append(dob)  
 record2.append(rollno)  
 record2.append(0)  
 sql = **"""insert into information(name, rollno, class, sec, pw, address, phoneno, dob)   
 values(%s,%s,%s,%s,%s,%s,%s,%s)"""** sql2 = **"""insert into rfp(rollno, fees) values(%s,%s)"""** mycursor.execute(sql2, record2)  
 mycursor.execute(sql, record)  
 con.commit()  
 student\_login()  
  
 **if** apw!=pw:  
 print(**"\nPASSWORD ENTERED IN BOTH PLACES IS NOT SAME PLEASE ENTER SAME PASSWORD ."**)  
 add\_student()  
  
  
**def** about\_school():  
 file = (**"C:\\Users\\KIRAN\\Desktop\\about\_school\_sms\_csip.txt"**)  
 os.startfile(file)  
 time.sleep(10)  
 print(**"\n\n\n\n"**)  
 menu()  
  
  
**def** inside\_student\_login():  
 print(**"Student's Data that we have with us is : "**)  
 print(**"Name, Roll no., Class, Sec, Password, Address, Phone no., Date of Birth"**)  
 sql = **f"""select \* from information where rollno = {**roll**}"""** mycursor.execute(sql)  
 stu\_data = mycursor.fetchmany()  
 con.commit()  
 print(stu\_data)  
 print(**"1 : UPDATE PROFILE"**)  
 print(**"2 : DEPOSIT FEE"**)  
 print(**"3 : LOG OUT"**)  
 choice = int(input(**"Enter your choice : "**))  
 **if** choice == 1:  
 update\_profile()  
 **elif** choice == 2:  
 deposite\_fee()  
 **elif** choice == 3:  
 print(**"\n\n"**)  
 menu()  
 **else**:  
 print(**"Please enter above given options only..."**)  
 inside\_student\_login()  
  
  
**def** update\_profile():  
 **global** name, cls, sec, dob, adrs, phone, pw, apw, rollno, record, record2  
 record = []  
 record2 = []  
 print(**"\nPlease enter the following details to update student information to school's data.\n"**)  
 name = input(**"NAME OF STUDENT : "**)  
 rollno = int(input(**"ROLL NUMBER OF STUDENT : "**))  
 cls = int(input(**"CLASS OF STUDENT (In integers only please): "**))  
 sec = input(**"SECTION OF STUDENT : "**)  
 dob = input(**"DATE OF BIRTH OF STUDENT DDMMYYYY : "**)  
 adrs = input(**"ADDRESS OF STUDENT : "**)  
 phone = int(input(**"PHONE NUMBER OF STUDENT : "**))  
 pw = input(**"ENTER THE PASSWORD WITH WHICH YOU CAN ACCESS THE PORTAL : "**)  
 apw = input(**"PLEASE ENTER THE PASSWORD AGAIN : "**)  
 deleter1 = **f"""delete from information where rollno = {**roll**}"""** mycursor.execute(deleter1)  
 con.commit()  
 password\_verification2()  
  
  
**def** password\_verification2():  
  
 **if** apw==pw:  
 print(**"\nSTUDENT PROFILE UPDATED."**)  
 record.append(name)  
 record.append(rollno)  
 record.append(cls)  
 record.append(sec)  
 record.append(pw)  
 record.append(adrs)  
 record.append(phone)  
 record.append(dob)  
 sql = **"""insert into information(name, rollno, class, sec, pw, address, phoneno, dob)   
 values(%s,%s,%s,%s,%s,%s,%s,%s)"""** mycursor.execute(sql, record)  
 con.commit()  
  
 student\_login()  
  
 **if** apw!=pw:  
 print(**"\nPASSWORD ENTERED IN BOTH PLACES IS NOT SAME PLEASE ENTER SAME PASSWORD ."**)  
 update\_profile()  
  
  
**def** deposite\_fee():  
 fee = **f"""select fees from rfp where rollno = {**roll**}"""** mycursor.execute(fee)  
 fee\_paid = mycursor.fetchone()[0]  
  
 con.commit()  
 **if** int(fee\_paid)!=120000:  
 fee\_pending = 120000 - int(fee\_paid)  
 print(**"Fees paid till now = "**, fee\_paid)  
 print(**"Fees pending = "**, fee\_pending)  
 print(  
 **f"{**Fore.GREEN**}To deposite fee kindly use to under given IFS code in the format School Fees of "  
 f"[Full name of student].{**Style.RESET\_ALL**}"**)  
 print(**"IFS Code = ltisnavimumbai@sbi"**)  
 fee\_paid\_rn = int(input(**"Enter the fees you have paid now we will verify your payment and get to "  
 "you and if you haven't paid anything then type 0 = "**))  
 fee\_left = fee\_paid\_rn + int(fee\_paid)  
 fee\_change = **f"""update rfp set fees = {**fee\_left**} where rollno = {**roll**}"""** mycursor.execute(fee\_change)  
 con.commit()  
 print(**"Please wait, verifying your payment......"**)  
 time.sleep(5)  
 print(  
 **f"\n\n{**Fore.RED**}PAYMENT VERIFIED, {**fee\_paid\_rn**}, HAS BEEN DEPOSITED TO THE SCHOOL's "  
 f"BANK ACCOUNT.{**Style.RESET\_ALL**}"**)  
 fee1 = **f"""select fees from rfp where rollno = {**roll**}"""** mycursor.execute(fee1)  
 fee\_paid1 = mycursor.fetchone()[0]  
 fee\_pending1 = 120000 - int(fee\_paid1)  
 con.commit()  
 print(**f"{**Fore.RED**}Fees paid till now = "**, fee\_paid1)  
 print(**"Fees pending = "**, fee\_pending1, **f"{**Style.RESET\_ALL**}\n"**)  
 print(**"The page will be redirected to logged in page in 7 sec"**)  
 **for** i **in** range(1, 8):  
 print(i)  
 time.sleep(1)  
  
 inside\_student\_login()  
  
 **else**:  
 print(**"YOU HAVE PAID THE ENTIRE FEE"**)  
 print(**"The page will be redirected to logged in page in 4 sec"**)  
 **for** i **in** range(1, 5):  
 print(i)  
 time.sleep(1)  
  
 inside\_student\_login()  
  
  
welcome()  
menu()

Workbench Source Code

create database student\_management\_system;

use student\_management\_system;

create table information(name varchar(60) not null,rollno int, class int,sec char(1),pw varchar(60), address varchar(200),phoneno bigint,dob varchar(30));

insert into information value("Alapan", 1, 12, "A", "alap", "elora", "9764319785", "14/05/2006");

insert into information value("Anupam", 2, 12, "A", "anup", "jasmine", "9769319785", "27/11/2005");

insert into information value("Auro", 3, 12, "A", "auro", "garden", "9767819785", "14/04/2005");

insert into information value("Aryan", 4, 12, "A", "arya", "sanchi", "9763619785", "01/08/2006");

insert into information value("Ayush", 5, 12, "A", "why", "ajanta", "9764789785", "22/08/2005");

create table rfp(rollno int, fees int);

insert into rfp value(1, 120000);

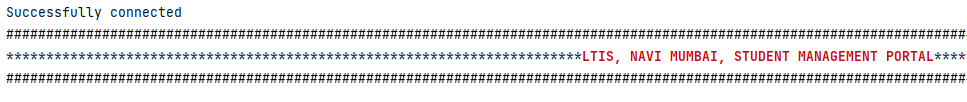
insert into rfp value(2, 50000);

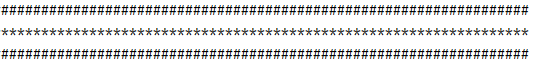
insert into rfp value(3, 20000);

insert into rfp value(4, 60000);

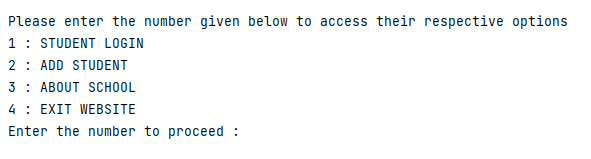
insert into rfp value(5, 80000);

**OUTPUT**

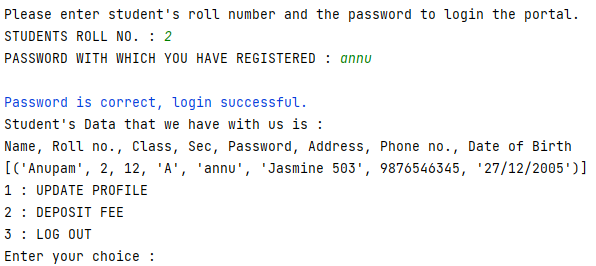
Welcome



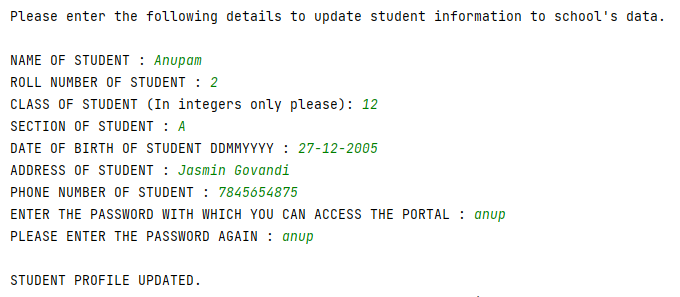
Menu

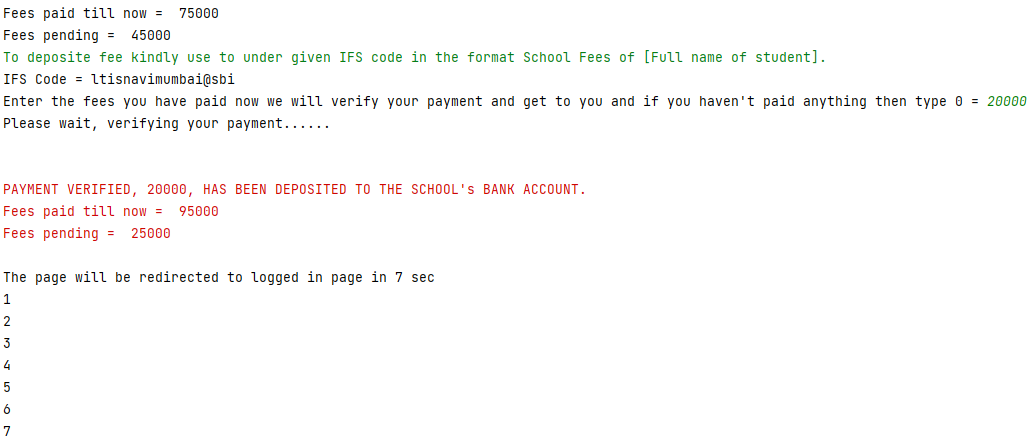


Student Login

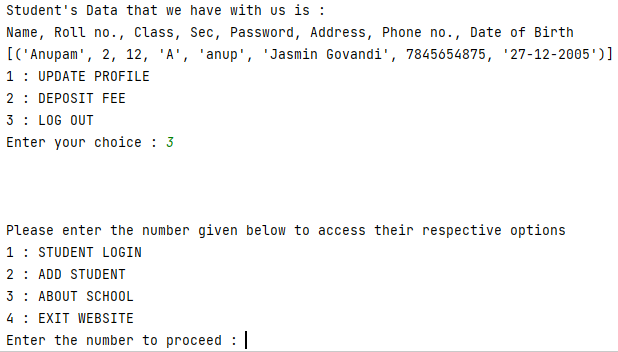


Update profile

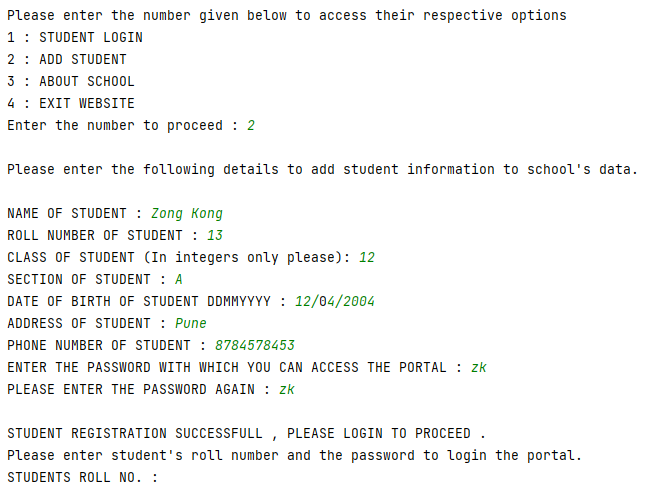


Deposite Fee

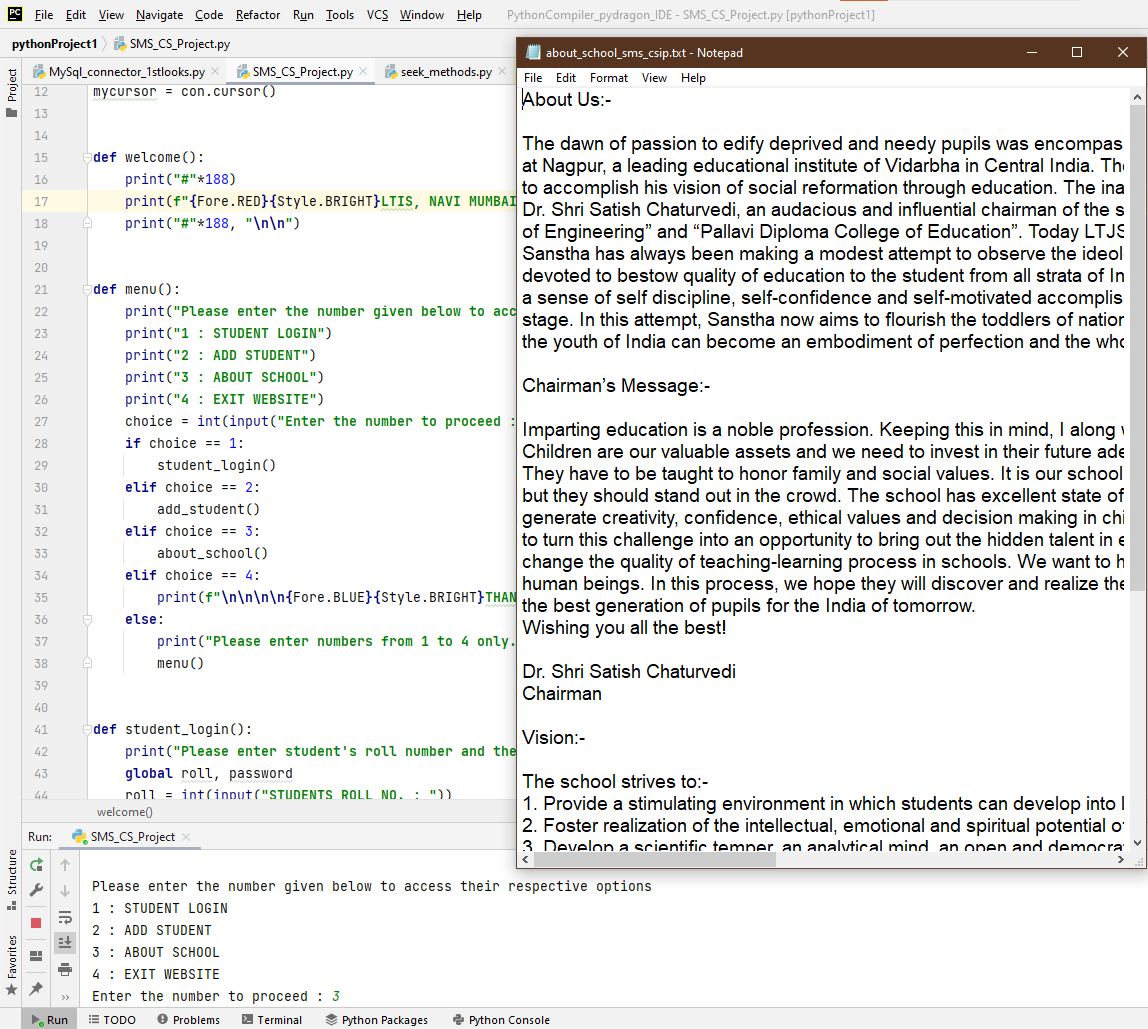
Logout



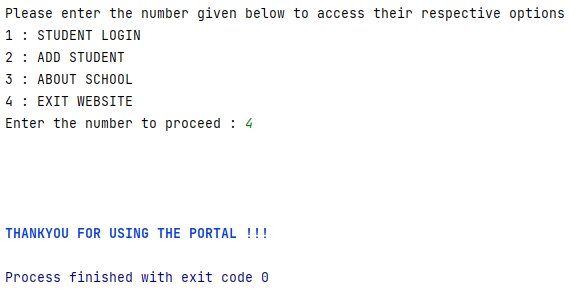
Add Student



About School

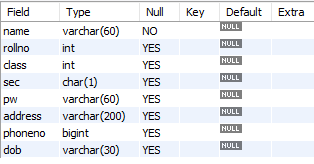


Exit website



**VARIABLE TABLES**

Information table

****

rfp table



**CONCLUSION**

Based on the knowledge acquired during the learning of python program and MySQL work bench in school, I could successfully apply it and write a program on "Student Management Portal". Using this code now one can easy manage a large data on their finger tips and students can easily find their fee pending which they are mostly likely to forget.

This program can be modified to take more and more different type of data ranging from marks to teacher’s reviews regarding the student. This can help parents to track their child’s progress.

**BIBLIOGRAPHY**

1. Computer Science with python. By Sumita Arora
2. <https://www.w3schools.com/sql/sql_delete.asp>
3. <https://www.w3schools.com/sql/sql_update.asp>
4. <https://stackoverflow.com/questions/29772337/python-mysql-connector-unread-result-found-when-using-fetchone>
5. <https://technocrash.online/python-example/create-a-student-table-and-insert-data/>
6. <https://dev.mysql.com/doc/refman/8.0/en/rename-table.html#:~:text=ALTER%20TABLE%20old_table%20RENAME%20new_table,are%20performed%20left%20to%20right>.
7. <https://docs.google.com/document/d/1c2mmPFJOMSYNs_QVAvxzFWzN2N6nb1Oo/edit>
8. <https://csstudy.in/python-projects-for-class-12-with-source-code-mysql-connectivity/>
9. <https://drive.google.com/file/d/1hxMRi-MqSDBq3v8LByj79v8f-DsxibME/view>
10. <https://stackoverflow.com/questions/39473297/how-do-i-print-colored-output-with-python-3>
11. <https://drive.google.com/file/d/1k5_Cm7o5y3ruafgtqP3OH7hrxa7Zy2Ip/view>
12. <https://drive.google.com/file/d/1MeSOvo1I5iRE5z-7bVGOUGcL4OMPRkOq/view>