**COMPUTER SCIENCE INVESTIGATORY PROJECT:**

**SESSION: 2019-20**

**SUBMITTED BY: PRITI SHARMA**

**CLASS: XII-A**

**ROLL NO:**

**SUBMITTED TO: MR SANJEEV KUMAR**

CERTIFICATE

This is to certify that **PRITI SHARMA** of class **XII-A** has successfully completed her project under the guidance of subject teacher **MR SANJEEV KUMAR** from **KENDRIYA VIDYALAYA NO.2** **AFS TEZPUR** in partial fulfillment of Computer Science practical examination conducted by **CBSE** for session 2019-20.

Principal’s signature: Teacher’s signature:

Examiner’s signature:

ACKNOWLEDGEMENT

I extend my appreciation and thanks to **MR SANJEEV KUMAR**. It was his unfailing pushing and constant inspiration that this work could take the present shape.

Turning aspirations into realities is easier when people are supportive of your efforts. Among such persons are my parents to whom I am deeply indebted for the positive feedback regarding contribution to my work and for their unfailing steps and ungrudging help at every step in this project.

-Priti Sharma

Class: XII-A

INDEX

|  |  |
| --- | --- |
| Sno. | Topic |
| 1 | System requirements |
| 2 | Feasibility study |
| 3 | Errors and its types |
| 4 | Testing |
| 5 | Maintenance |
| 6 | Flow chart of program |
| 7 | Code |
| 8 | Output |
| 9 | Appendix |
| 10 | Bibliography |

**SYSTEM REQUIREMENTS**

**1. HARDWARE:**

* + Processor
  + Keyboard
  + Minimum memory - 2GB

**2. SOFTWARE:**

* + Operating System –OS7, OS8
  + Python IDLE
  + MYSQL

**FEASIBILITY STUDY**

Feasibility study is a system proposal according to its work, ability, impact on the operation ability to meet the needs of users and efficient use of resources. An important outcome of preliminary investigations the determination of that system requested feasible.

**ECONOMICAL FEASIBILITY:**

Economics analysis is the most frequent use method for evaluating the effectiveness of the candidates the benefits and savings that are expected from system and compare them with cost.

This software is not very costly. It just worth Rs.5500/-.So users records can be maintained at a cheaper cost and every school would like to use this software so that the student’s records can be managed easily.

**TECHNICAL FEASIBILTY:**

Technical feasibility centre on the existing computer system and to what extent it can support the proposed task. This involves financial consideration to accommodate technical enhancements.

It is technically feasible because whatever technology is needed to develop this software is easily available.

**ERRORS AND ITS TYPES**

An error, some time called “A BUG” is anything in the code that prevents a program from compiling and running correctly. There are broadly three types of errors as follows:

1. **Compile- time errors**: Errors that occurs during compilation of a program is called compile time error. It has two types as follows:
2. **Syntax error**: It refers to formal rules governing the construction of valid statements in a language.
3. **Semantics error**: It refers to the set of rules which give the meaning of a statement.
4. **Run time Errors:** Errors that occur during the execution of program are run time errors. These are harder to detect errors. Some run-time error stop the execution of program which is then called program “Crashed”.
5. **Logical Errors:** Sometimes, even if you don’t encounter any error during compiling-time and runtime, your program does not provide the correct result. This is because of the programmer’s mistaken analysis of the problem he or she is trying to solve.Such errors are called logical error.

**TESTING**

1. **Alpha Testing**: It is the most common type of testing used in the software industry. The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user. It is conducted at the developer’s site.
2. **Beta Testing**: It is a formal type of software testing which is carried out by the customers. It is performed in a real environment before releasing the products into the market for the actual end-users. It is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirement. Beta Testing is successful when the customer accepts the software.
3. **White Box Testing**: White box testing is based on the knowledge about the internal logic of an application’s code. It is also known as Glass box Testing. Internal Software and code working should be known for performing this type of testing. These tests are based on the coverage of the code statements, branches, paths, conditions etc.
4. **Black Box Testing**: It is a software testing, method in which the internal structure or design of the item to be tested is not known to the tester. This method of testing can be applied virtually to every level of the software testing.

**MAINTENANCE**

Programming maintenance refers to the modifications in the program. After it has been completed, in order to meet changing requirement or to take care of the errors that shown up. There are four types of maintenance:

1. **Corrective Maintenance**: When the program after compilation shows error because of some unexpected situations, untested areas such errors are fixed up by Corrective maintenance.
2. **Adaptive Maintenance**: Changes in the environment in which an information system operates may lead to system management. To accommodate changing needs time to time maintenance is done and is called Adaptive maintenance.
3. **Preventive Maintenance**: If possible the errors could be anticipated before they actually occur; the maintenance is called Preventive maintenance.
4. **Perfective Maintenance**: In this rapidly changing world, information technology is the fastest growing area. If the existing system is maintained to keep tuned with the new features, new facilities, new capabilities, it is said to be Perfective maintenance.

**FLOW CHART OF THE PROGRAM**

**EXIT**

CODE

**import mysql.connector as mcon**

**import sys**

**con = mcon.connect(host="localhost",port="3306",user="root",passwd="root")**

**mycursor = con.cursor()**

**if con.is\_connected():**

**print("MySql DataBase is connected Successfully.")**

**mycursor.execute("create database if not exists LOC")**

**mycursor.execute("use LOC")**

**mycursor.execute("create table if not exists user \**

**(uname varchar(20) primary key,upwd varchar(20)\**

**,utype char(5),ustatus char(5))")**

**Q = "insert into user(uname,upwd,utype) values (\'LOC\',\'LOC\',\'S\')"**

**#print(Q)**

**#mycursor.execute(Q)**

**con.commit()**

**at = 1**

**while at <= 3:**

**at += 1**

**uid = input("Enter User Name : ")**

**pwd = input("Enter User Password : ")**

**status = 'A'**

**mycursor.execute("select \* from user where uname = '{}' and upwd = '{}' and ustatus = '{}'".format(uid,pwd,status))**

**data = mycursor.fetchone()**

**count = mycursor.rowcount**

**#print(count)**

**if count == 1:**

**print("Login Successfully.")**

**print("Perform CRUD Operations.")**

**#--------------------------\*CHOICES\*-----------------------------**

**while True:**

**print("Input 'I' for Insertion a New Record.")**

**print("Input 'U' for Update an Existing Record.")**

**print("Input 'R' for Removal an Existing Record.")**

**print("Input 'S' for Searching a Record.")**

**print("Input 'D' for Display All Records.")**

**print("Input 'E' for Exit the Program.")**

**ch = input("Enter Your Option: ")**

**#--------------------------\*TABLE CREATION\*-----------------------------**

**if ch == 'I' or ch == 'i':**

**ins = "create table if not exists students(\**

**reg\_num int(20) primary key, loc\_sr\_num integer NOT NULL, yr\_pass\_xi int(5) NOT NULL,\**

**exam\_cat char(5) NOT NULL,cand\_name char(50) NOT NULL,mother\_name char(50) NOT NULL,\**

**father\_name char(50) NOT NULL,gender varchar(5),category1 varchar(5),minority varchar(5),\**

**PwD\_status varchar(20),mob\_num bigint NOT NULL,email\_id varchar(50),aadhar\_num bigint,\**

**sub\_1 char(15),sub\_2 char(15) NOT NULL,sub\_3 char(15) NOT NULL,\**

**sub\_4 char(15) NOT NULL,sub\_5 char(15) NOT NULL,add\_sub\_6 char(15) NOT NULL,\**

**int\_grade\_sub1 char(30),int\_grade\_sub2 char(30),int\_grade\_sub3 char(30),annual\_income varchar(25),roll\_num\_of\_equi\_exam\_passed integer,\**

**exam\_of\_equi\_exam\_passed char(20),board\_of\_equi\_exam\_passed char(20),single\_child char(5),\**

**migration\_certificate char(5),adm\_no integer,adm\_date date)"**

**#print(ins)**

**mycursor.execute(ins)**

**#--------------------------\*INSERTION OF RECORDS\*-----------------------------**

**print("Insertion Operation.")**

**reg = int(input("Enter student's registration\_num: "))**

**locsr = int(input("Enter student's loc\_sr\_num: "))**

**yrpassc11 = int(input("Enter student's year\_passing\_class11: "))**

**ecat = input("Enter student's exam\_cat: ")**

**cname = input("Enter student's Name: ")**

**mname = input("Enter student's mother's name: ")**

**fname = input("Enter student's father's name: ")**

**gender = input("Enter student's gender: ")**

**cat = input("Enter student's category: ")**

**minor = input("Enter if student belongs to minority section(y/n): ")**

**pwdis = input("Enter if student have disability (type of disability): ")**

**mnum = int(input("Enter student's mobile\_num: "))**

**email = input("Enter student's email\_id: ")**

**ad\_num = int(input("Enter student's addhar number: "))**

**s1 = input("Enter subject1(compulsory language): ")**

**s2 = input("Enter subject2: ")**

**s3 = input("Enter subject3: ")**

**s4 = input("Enter subject4: ")**

**s5 = input("Enter subject5: ")**

**s6 = input("Enter subject6(additional): ")**

**intsub1 = input("Enter name of internal grade subject1: ")**

**intsub2 = input("Enter name of internal grade subject2: ")**

**intsub3 = input("Enter name of internal grade subject3: ")**

**aninc = int(input("Enter annual income of student's parents: "))**

**eexrnum = int(input("Enter student's rollnum of equivalent exam passed:"))**

**eexam = input("Enter student's exam of equivalent exam passed:")**

**eexboard = input("Enter student's board of equivalent exam passed:")**

**sch = input("Enter if student is single girl child or not:")**

**mgcr = input("Enter if migration certificate is required or not:")**

**adm\_num = int(input("Enter student's admission num:"))**

**adm\_date = input("Enter student's admission date as (yyyy-mm-dd):")**

**q = "insert into students (reg\_num,loc\_sr\_num,yr\_pass\_xi,exam\_cat,cand\_name,\**

**mother\_name,father\_name,gender,category1,minority,PwD\_status,mob\_num,email\_id,\**

**aadhar\_num,sub\_1,sub\_2,sub\_3,sub\_4,sub\_5,add\_sub\_6,int\_grade\_sub1,int\_grade\_sub2,int\_grade\_sub3,\**

**annual\_income,roll\_num\_of\_equi\_exam\_passed,exam\_of\_equi\_exam\_passed,board\_of\_equi\_exam\_passed,\**

**single\_child,migration\_certificate,adm\_no,adm\_date) values ({},{},{},'{}',\**

**'{}','{}','{}','{}','{}','{}','{}',{},'{}',{},'{}','{}','{}','{}','{}','{}','{}','{}','{}',{},\**

**{},'{}','{}','{}','{}',{},'{}')\**

**".format(reg,locsr,yrpassc11,ecat,cname,mname,fname,gender,cat,minor,pwdis,mnum,email,ad\_num,s1,s2,s3,s4,s5,s6,intsub1,intsub2,intsub3,aninc,eexrnum,eexam,eexboard,sch,mgcr,adm\_num,adm\_date)**

**mycursor.execute(q)**

**con.commit()**

**print("Record is inserted Successfully.")**

**#--------------------------\*UPDATION\*-----------------------------**

**elif ch == 'U' or ch == 'u':**

**print("Updation of Record.")**

**reg = input("Enter Student's registration Number: ")**

**sn = input("Enter New student's Name: ")**

**mn = input("Enter New student Mother's Name: ")**

**fn = input("Enter New student Father's Name: ")**

**qry = "update students set cand\_name = '{}' , mother\_name = '{}' , father\_name = '{}' where reg\_num = {}".format(sn,mn,fn,reg)**

**mycursor.execute(qry)**

**con.commit()**

**print("Record is updated Successfully.")**

**#--------------------------\*DELETION\*-----------------------------**

**elif ch == 'R' or ch == 'r':**

**print("Removal of Record.")**

**reg = input("Enter Student's registration Number: ")**

**qry = "delete from students where reg\_num = {}".format(reg)**

**mycursor.execute(qry)**

**con.commit()**

**print("Record is deleted Successfully.")**

**#--------------------------\*SEARCHING\*-----------------------------**

**elif ch == 'S' or ch == 's':**

**print("Searching Operation.")**

**reg = input("Enter Student's registration Number: ")**

**qry = "select \* from students where reg\_num = {} ".format(reg)**

**#print(qry)**

**mycursor.execute(qry)**

**print("Record is found Successfully.")**

**data = mycursor.fetchone()**

**count = mycursor.rowcount**

**print("Total No. of Record:",count)**

**for row in data:**

**print(row)**

**#--------------------------\*DISPLAY\*-----------------------------**

**elif ch == 'D' or ch == 'd':**

**print("Display ALl Records.")**

**qry = "select \* from students"**

**mycursor.execute(qry)**

**data = mycursor.fetchall()**

**count = mycursor.rowcount**

**print("Total No. of Record: ",count)**

**print("{0:<9s}{1:<9s}{2:<9s}{3:<9s}{4:<9s}{5:<9s}{6:<9s}{7:<9s}{8:<9s}{9:<9s}".format('Sl.No','Name','MName','FName','Subject1','Subject2','Subject3','Subject4','Subject5','Subject6'))**

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

**for row in data:**

**print("{0:<9s}{1:<9s}{2:<9s}{3:<9s}{4:<9s}{5:<9s}{6:<9s}{7:<9s}{8:<9s}{9:<9s}".format(str(row[1]),row[4],row[5],row[6],row[14],row[15],row[16],row[17],row[18],row[19]))**

**elif ch == 'E' or ch == 'e':**

**print("Exiting Program.")**

**sys.exit(0)**

**else:**

**print("Wrong Input. Try Again!!!!!")**

**else:**

**print("Login Failed")**

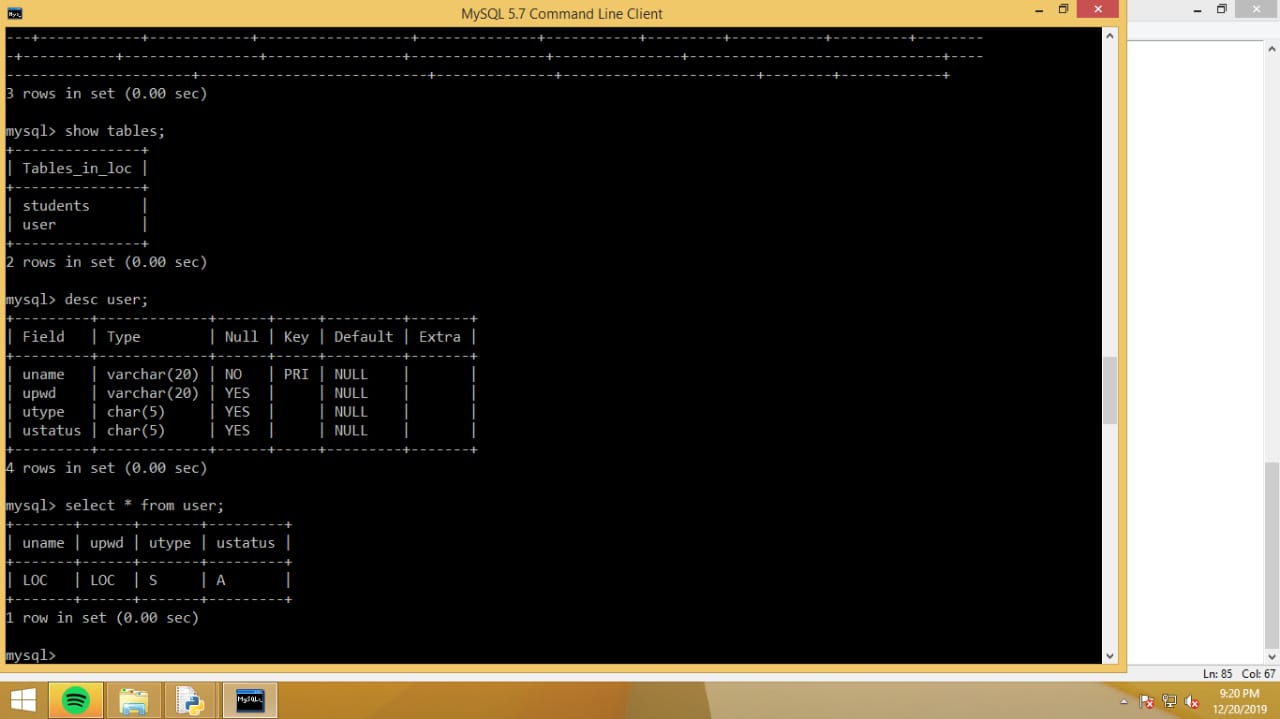
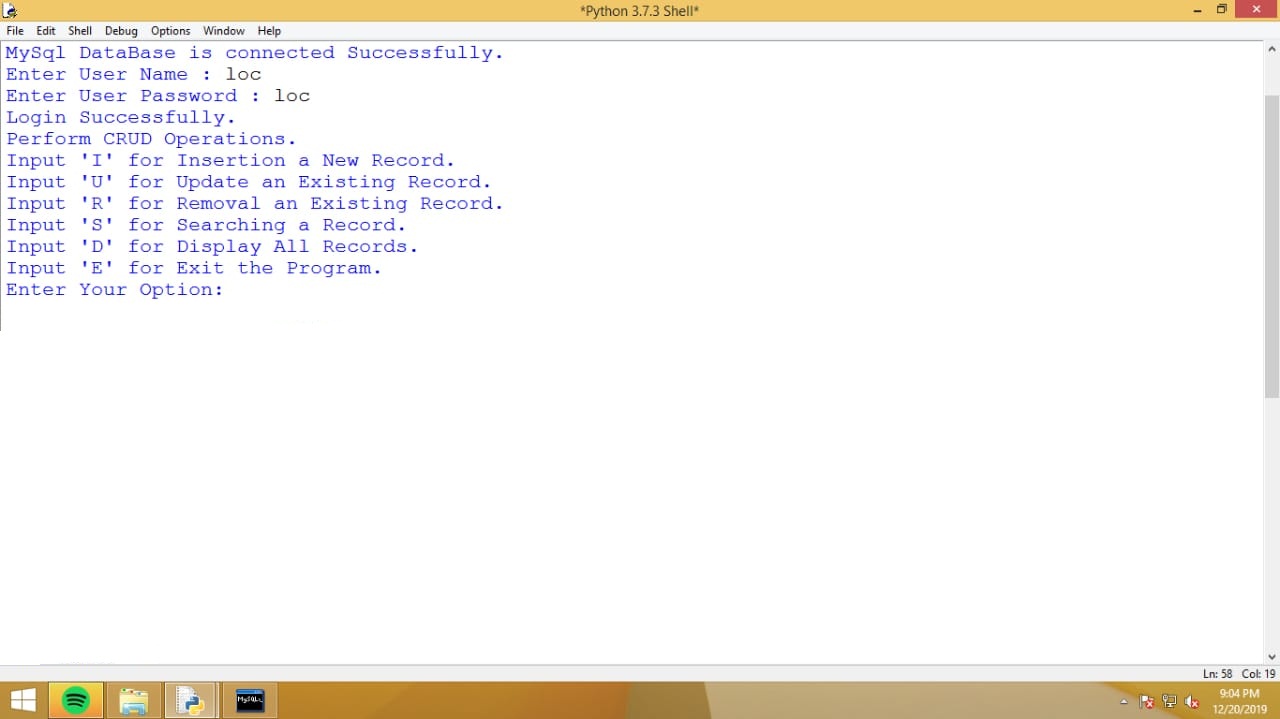
**if at !=4:**

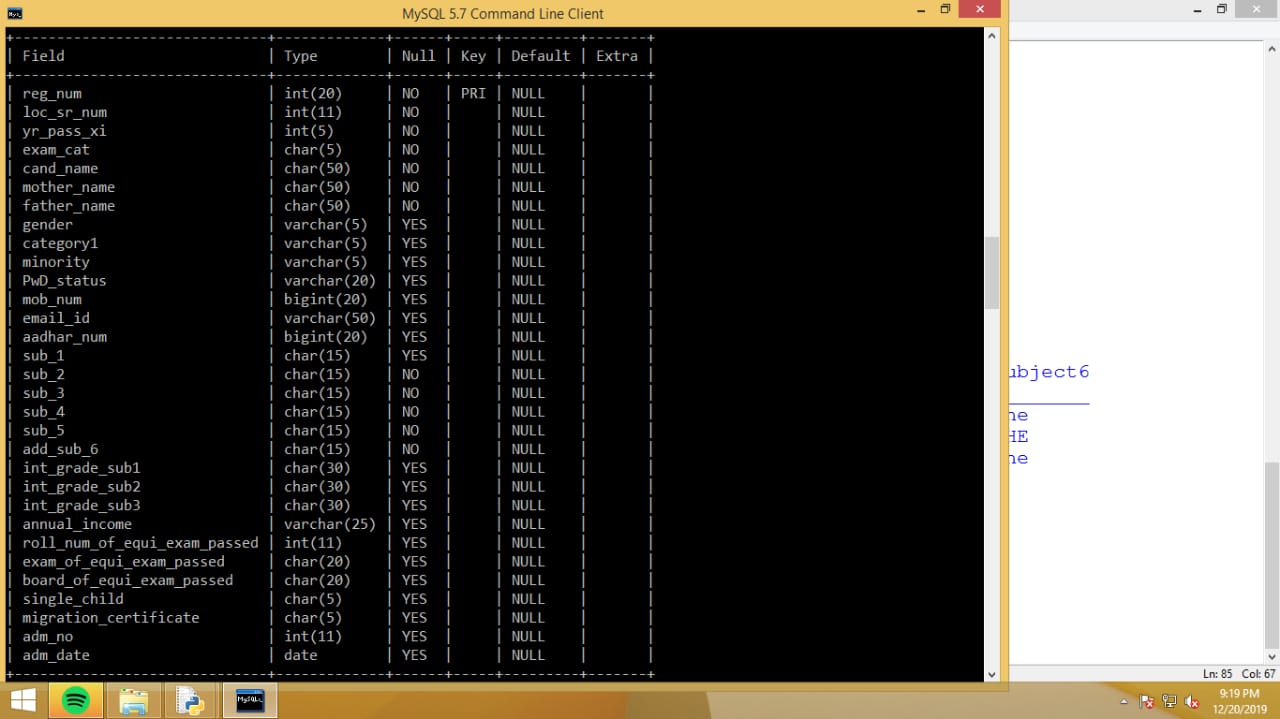
**print("Try Again")**

**else:**

**print("MySql DataBase Connection Failed.Terminating....")**

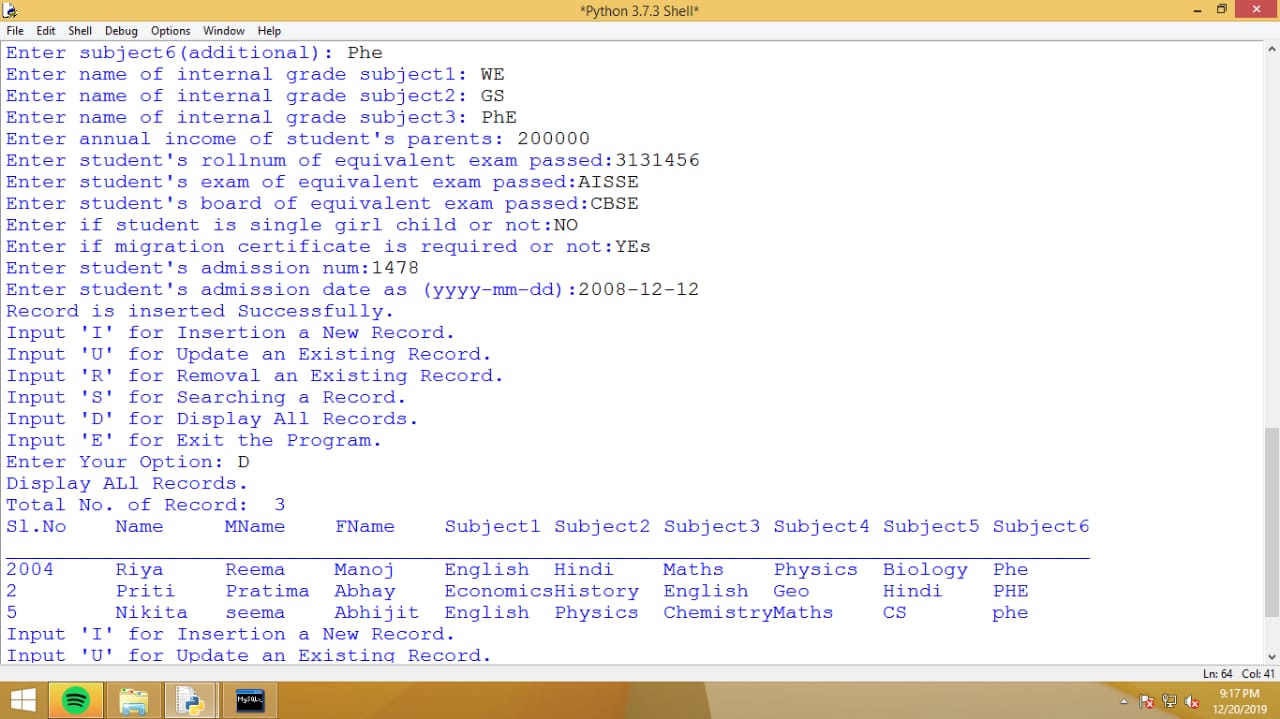
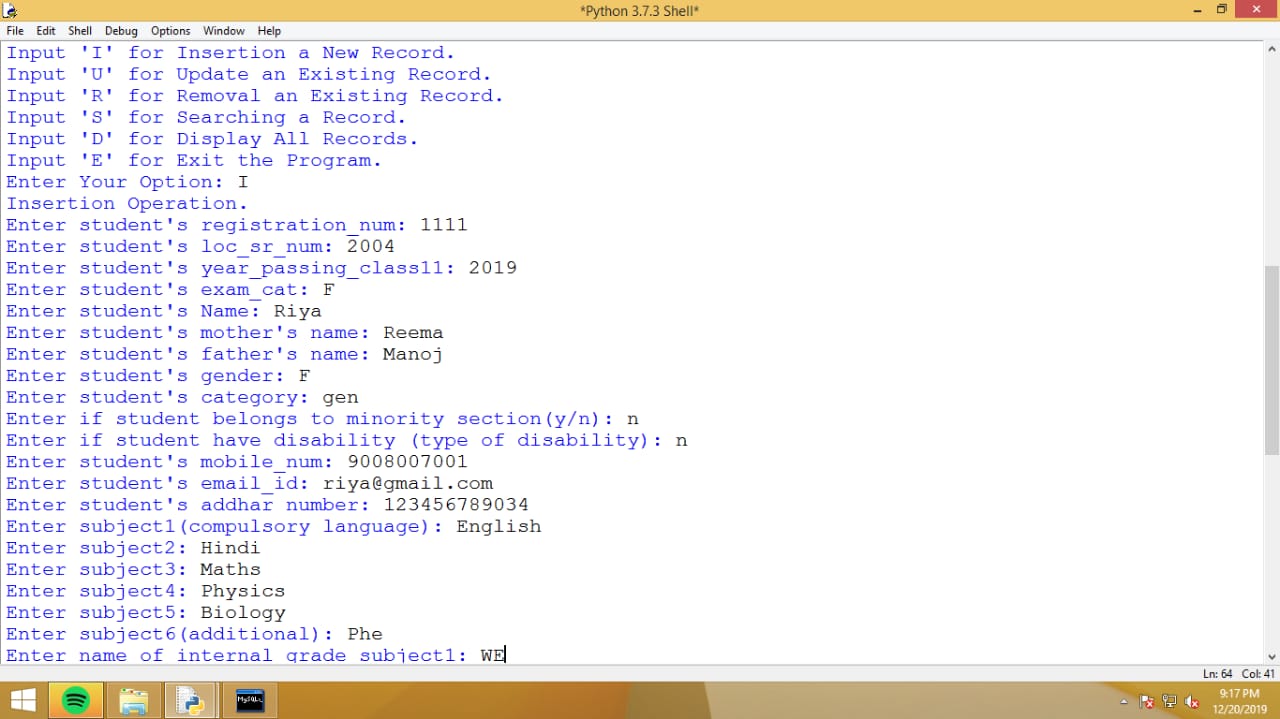
OUTPUT

**1.Table created successfully.**

****

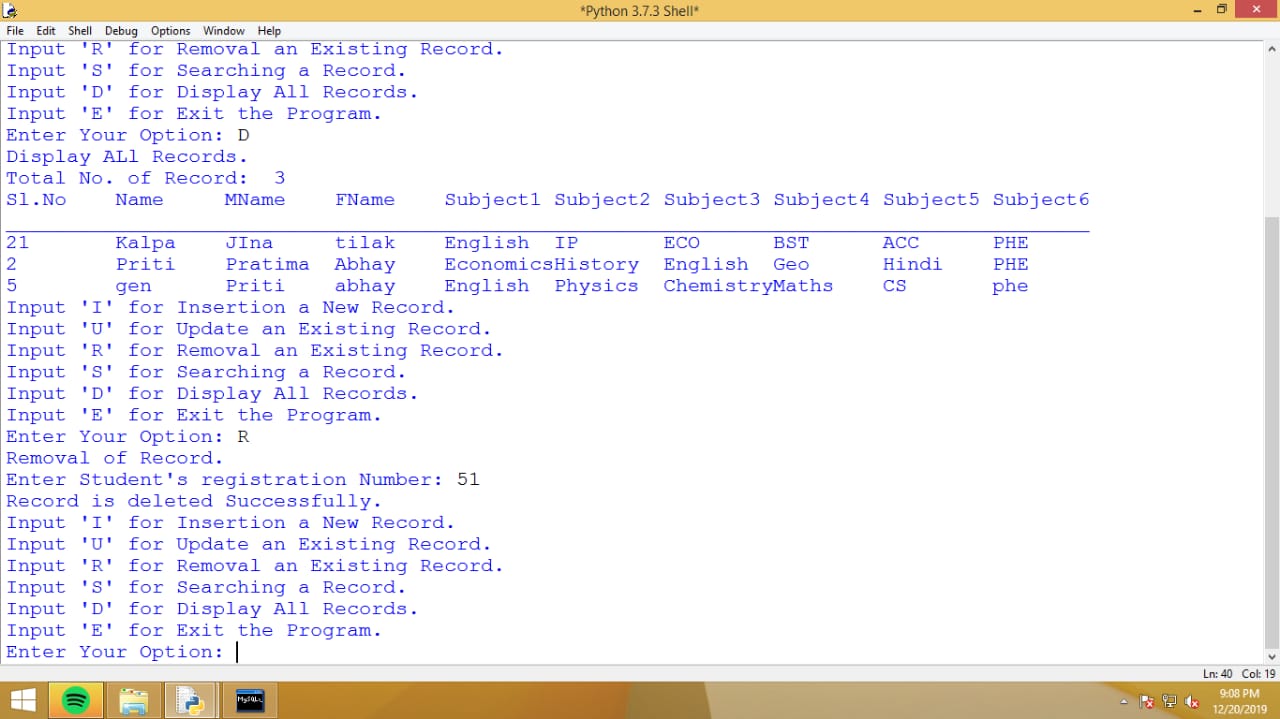
**STRUCTURE OF THE TABLE**

**2. Insertion of record:**

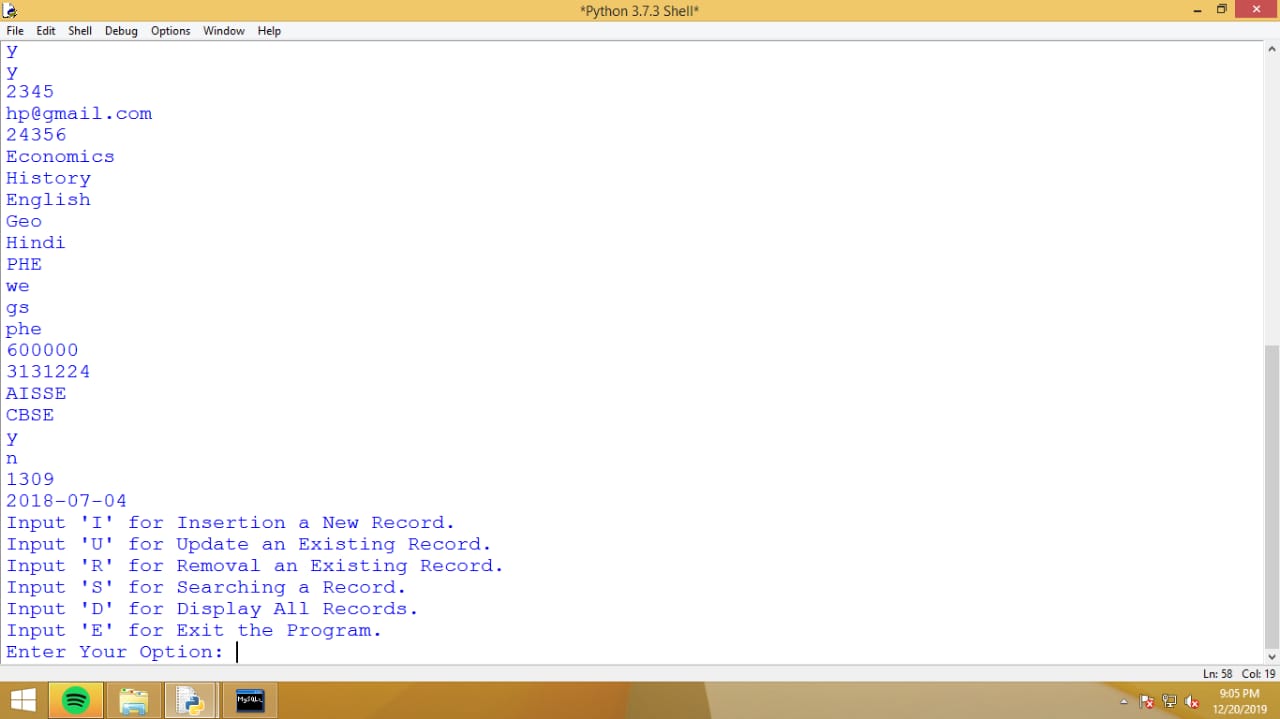
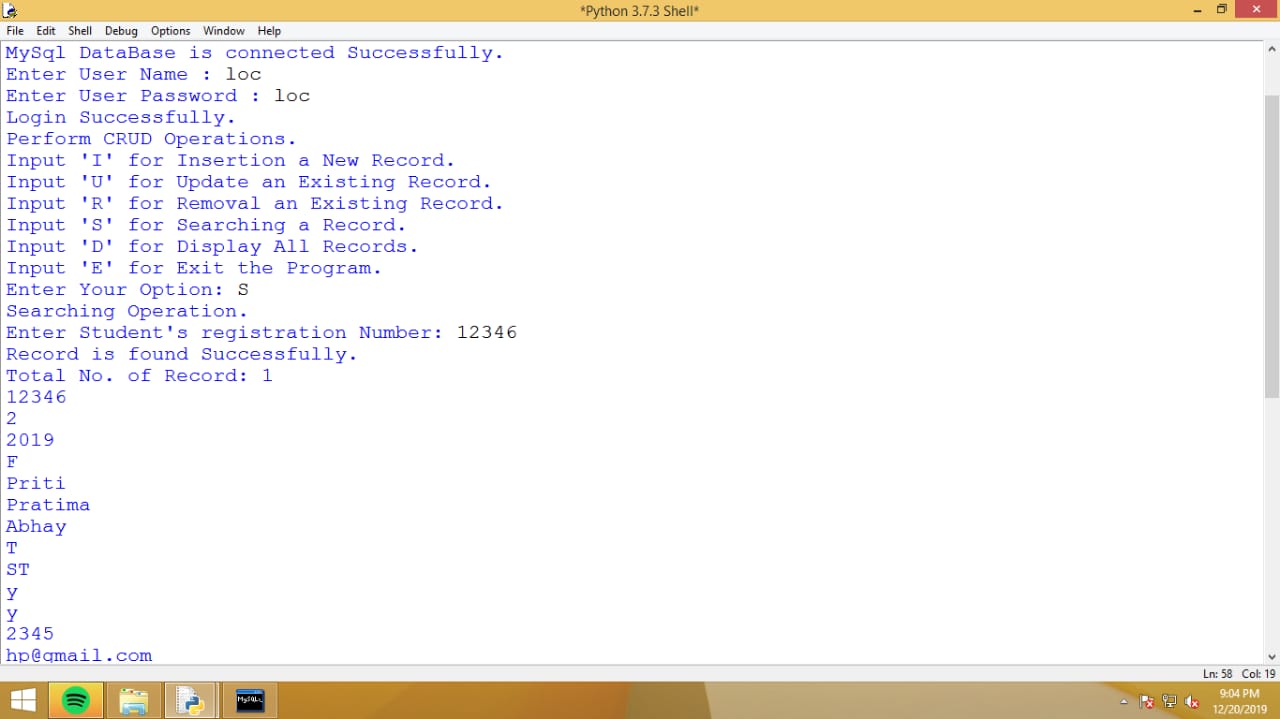
****

**3. Updating record: **

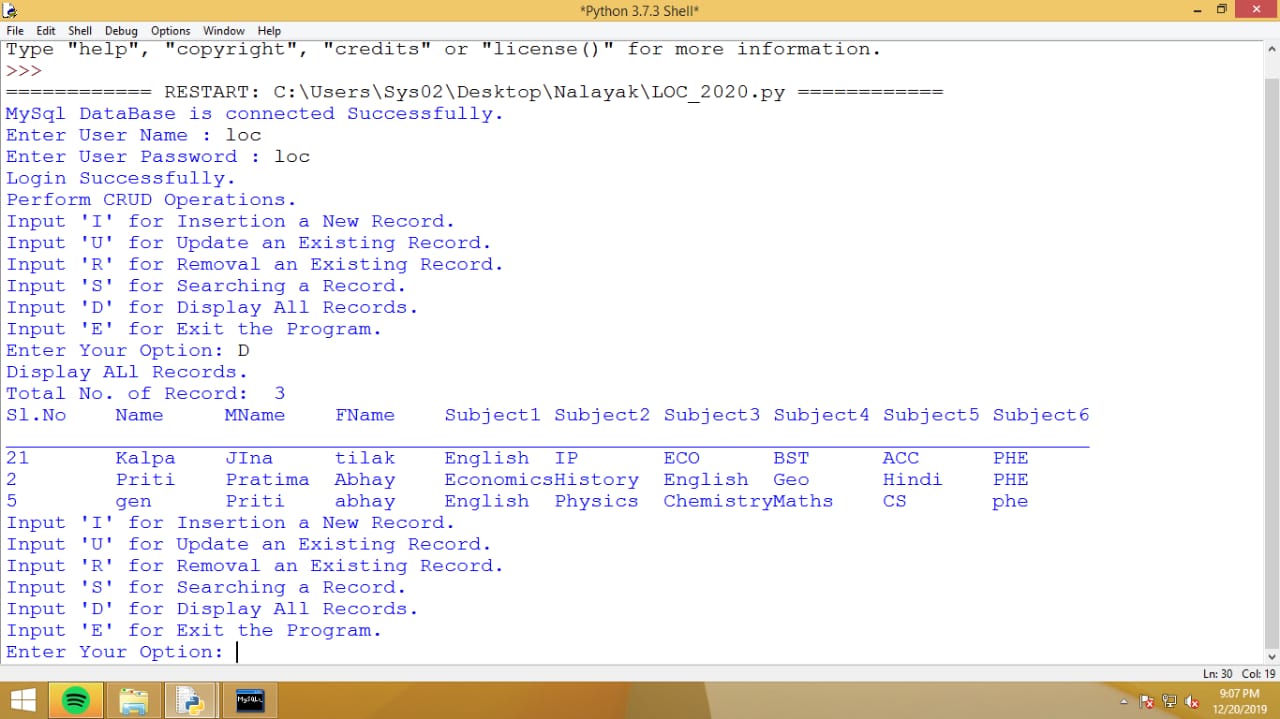
**4. Deleting a record:**

****

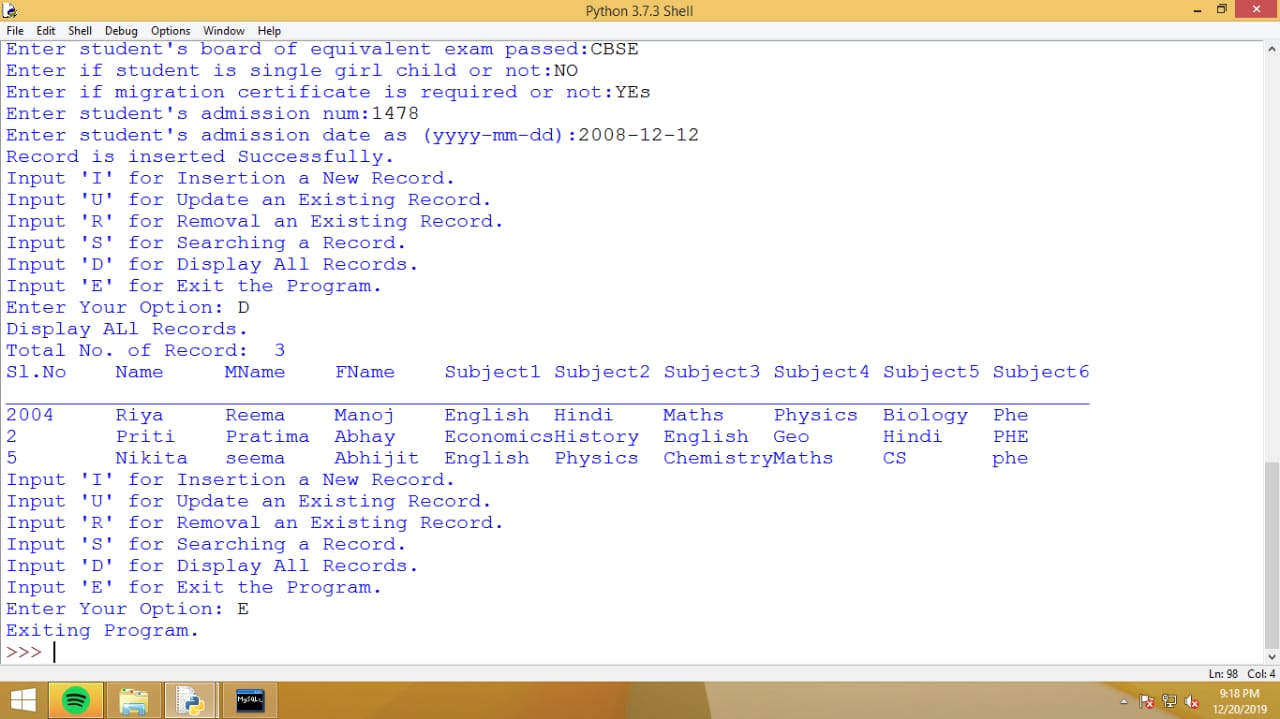
**5. Searching a record:**



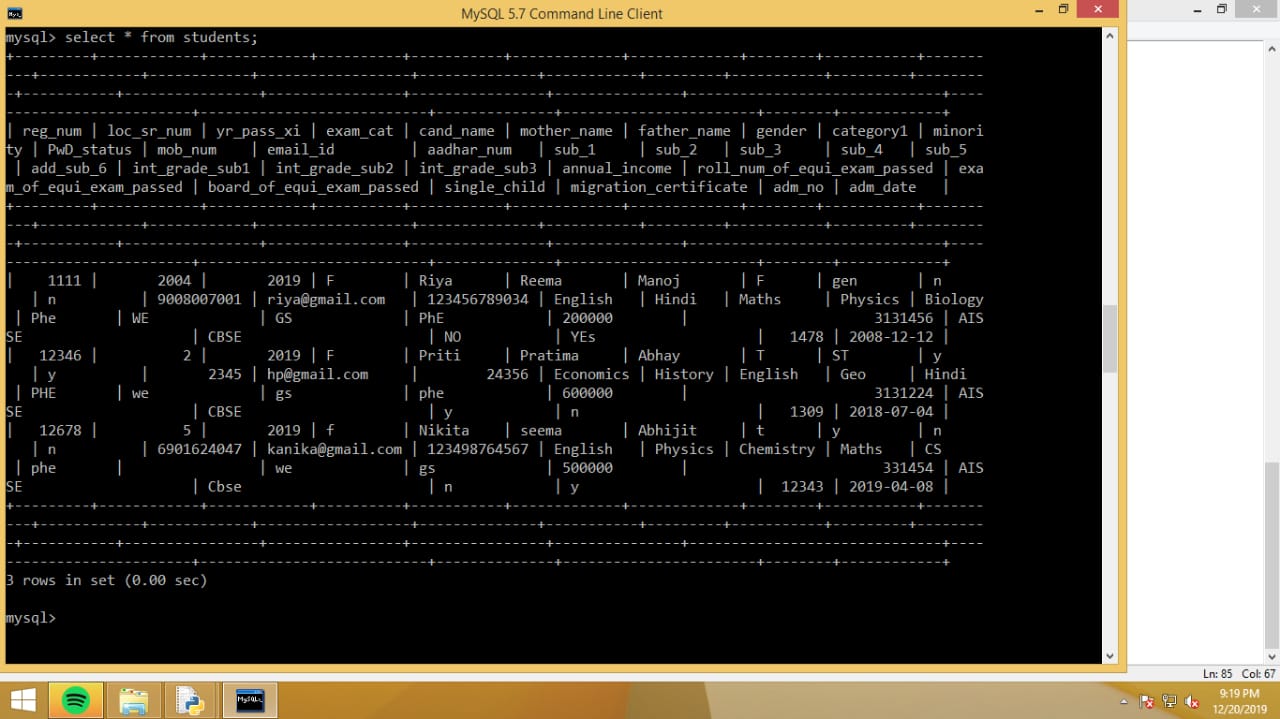
**6. Display all Records:**

****

**7. Exit:**

****

**RECORDS:**

****

**APPENDIX**