

DBMS

INNOVATIVE ASSIGNMENT



TOPIC: EXAM MANAGEMENT SYSTEM

COURSE CODE: 2CS402

MADE BY :

21BCE141 (DEV MAKWANA)

21BCE142 (MALAY ZALAWADIA)

21BCE144 (MANAV PATEL)

INTRODUCTION

OBJECTIVES & PROBLEM STATEMENT :

- The main objective of the project is to design and develop a user friendly-system
- Easy to use and an efficient computerized system.
- To develop an accurate and flexible system, it will eliminate data redundancy.
- To study the functioning of Students management System.
- To make a software fast in processing, with good user interface.
- To make software with good user interface so that user can change it and it should be used for a long time without error and maintenance.
- To provide immediate storage and retrieval of data and information.
- Improving arrangements for students coordination.
- Reducing paperwork.

LIMITATIONS:

- Time consumption in data entry as the records are to be manually maintained faculties a lot of time.
- Lot of paper work is involved as the records are maintained in the files and registers.
- Storage Requires as files and registers are used the storage space requirement is increased.
 - Less Reliable use of papers for storing valuable data information is not at all reliable.
 - Aadhar linkage with the official aadhar database has not been done.

ABOUT THE PROJECT :

Student's Online Exam Database Management System using Python and MySQL Python.

In this project, we have designed a database that will keep record of students, their subjects and the online exam they give. This project is helpful for students to take examination online by giving them the examination link automatically.

This project has been developed using MySQL software and Python Programming language. The required Database is Online Exam and has three Tables that are 'Student', 'Subject' and 'Exam'. These are the three entities in this Database.

In this project we have created multiple Python based modules and function binding MySQL to work on the required database. Using these python functions, we can:

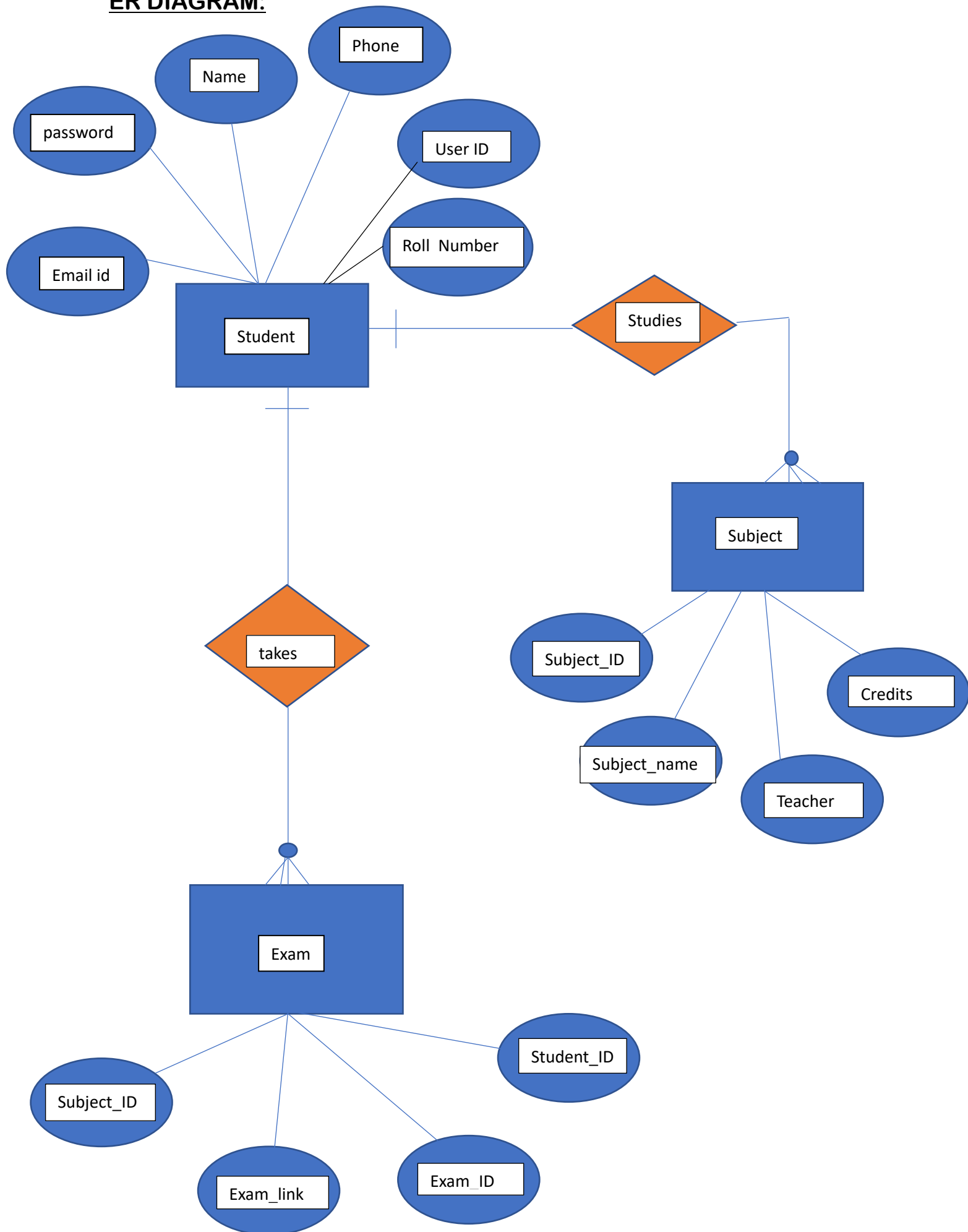
- 1) Append new entries to all three tables
- 2) Delete the existing entries.
- 3) Update the entries
- 4) And select the entries based on the Primary Key.

To each and every database table. To manage, add, change, and delete data entries in the tables with ease, we developed a user-friendly, menu-driven Python programme.

The database on the Local Host server will be automatically connected when the Python file is run.

The user will have a number of choices to change and edit the different tables in the Database while the programme continues to run.

ER DIAGRAM:



INPUT (PYTHON AND MYSQL CODE):

exam_management.py

```
import Database
import Menu
import Student
import Subject
import Exam

Database.DatabaseCreate()
Database.TablesCreate()

while True:
    print("\t\t\t\t\t EXAM MANAGEMENT SYSTEM\n")
    print("=====")
    print("1. Student Management")
    print("2. Subject Management")
    print("3. Exam Management")
    print("4. Exit")
    choice = int(input("Enter your choice: "))

    if choice == 1:
        Menu.student_manage()
    elif choice == 2:
        Menu.subject_manage()
    elif choice == 3:
        Menu.exam_manage()
    elif choice == 4:
        break
    else:
        print("Invalid choice !!!")
        x = input("Press any key to continue")
```

Database.py

File Edit Format Run Options Window Help

```
import mysql.connector as connector

def DatabaseCreate():
    cnx = connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003')
    Cursor = cnx.cursor()
    Cursor.execute("CREATE DATABASE IF NOT EXISTS online_exam")
    Cursor.execute("")
    Cursor.close()
    cnx.close()

def TablesCreate():
    cnx = connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
    Cursor = cnx.cursor()
    Cursor.execute("CREATE TABLE IF NOT EXISTS Student(User_ID varchar(10) PRIMARY KEY, Password varchar(20), Email_Address varchar(40), Name text(40), Roll_Number varchar(20), Phone varchar(20))")
    Cursor.execute("CREATE TABLE IF NOT EXISTS Subject(Subject_ID varchar(10) PRIMARY KEY, Subject_name text(20), Credits int(2), Teacher text(40))")
    Cursor.execute("CREATE TABLE IF NOT EXISTS Exam(Exam_ID varchar(10) PRIMARY KEY, User_ID varchar(10), Subject_ID varchar(10), Exam_Link varchar(50))")
    Cursor.close()
    cnx.close()
```

Menu.py

```
import Student
import Subject
import Exam

def student_manage():
    while True:
        print("\t\t\t STUDENT RECORD MANAGEMENT\n")
        print("=====")
        print("1. Add Student")
        print("2. Search Student Record")
        print("3. Delete Student Record")
        print("4. Update Record")
        print("5. Return to Main Menu")
        print("=====")
        choice = int(input("Enter your choice: "))
        if choice == 1:
            Student.insertData()
        elif choice == 2:
            Student.SearchStudentRec()
        elif choice == 3:
            Student.deleteStudent()
        elif choice == 4:
            Student.UpdateStudent()
        elif choice == 5:
            return
        else:
            print("Invalid choice !!!")
            x = input("Enter any key to continue")

def subject_manage():
    while True:
        print("\t\t\t SUBJECT RECORD MANAGEMENT\n")
        print("=====")
        print("1. Add Subject Record")
        print("2. Search Subject Record")
        print("3. Delete Subject Record")
        print("4. Update Subject Record")
        print("5. Return to Main Menu")
        print("=====")
        choice = int(input("Enter your choice: "))
        if choice == 1:
            Subject.insertSubject()
        elif choice == 2:
            Subject.SearchSubject()
        elif choice == 3:
            Subject.deleteSubject()
```



```

        elif choice == 4:
            Subject.UpdateSubject()
        elif choice == 5:
            return
        else:
            print("Invalid choice !!!")
            x = input("Enter any key to continue")

def exam_manage():
    while True:
        print("\t\t\t\t MEMBER RECORD MANAGEMENT\n")
        print("=====")
        print("1. Add Exam")
        print("2. Delete Exam")
        print("3. View Exam")
        print("4. Return to Main Menu")
        print("=====")
        choice = int(input("Enter your choice: "))
        if choice == 1:
            Exam.AddExam()
        elif choice == 2:
            Exam.DeleteExam()
        elif choice == 3:
            Exam.ViewExam()
        elif choice == 4:
            return
        else:
            print("Invalid choice !!!")
            x = input("Enter any key to continue")

```

Student.py

```

import mysql.connector
from mysql.connector import errorcode
from datetime import date, datetime, timedelta
from mysql.connector import connection
import os
import platform

def clrscreen():
    if platform.system() == "Windows":
        print(os.system("cls"))

def insertData():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        User_ID = input("Enter User ID : ")
        Password = input("Enter Password : ")
        Email_Address = input("Enter Email_Address : ")
        Name = input("Name : ")
        Roll_Number = input("Enter Roll Number : ")
        Phone = input("Enter Phone Number : ")
        Qry = ("INSERT INTO Student VALUES (%s, %s, %s, %s, %s, %s)")
        data = (User_ID, Password, Email_Address, Name, Roll_Number, Phone)
        Cursor.execute(Qry,data)
        cnx.commit()
        Cursor.close()
        cnx.close()
        print("Record Inserted.")
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
    cnx.close()

```

```

def deleteStudent():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        User_ID = input("Enter User_ID of Student to be deleted : ")
        Qry = ("DELETE FROM Student WHERE User_ID = %s")
        del_rec = (User_ID,)
        Cursor.execute(Qry, del_rec)
        cnx.commit()
        Cursor.close()
        cnx.close()
        print(Cursor.rowcount, "Record(s) Deleted Successfully.")
    except mysql.connector.ERROR as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
    cnx.close()

def SearchStudentRec():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        User_ID = input("Enter User_ID of Student to be searched : ")
        query = ("SELECT * FROM Student WHERE User_ID = %s ")
        rec_srch = (User_ID,)
        Cursor.execute(query, rec_srch)
        Rec_count = 0
        for (User_ID, Password, Email_Address, Name, Roll_Number, Phone) in Cursor:
            Rec_count += 1
            print("=====")
            print("User_ID : ", User_ID)
            print("Password : ", Password)
            print("Email_Address : ", Email_Address)
            print("Name: ", Name)
            print("Roll_Number : ", Roll_Number)
            print("Phone : ", Phone)
            print("=====")
            if Rec_count%2 == 0:
                input("Press any key continue")
                clrscr()
                print(Rec_count, "Record(s) found")
        cnx.commit()

```

```

        Cursor.close()
        cnx.close()
    except mysql.connector.ERROR as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
    cnx.close()

def UpdateStudent():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        User_ID = input("Enter User_ID of the Student to be Updated : ")
        print("Enter new data")
        New_User_ID = input("Enter User_ID : ")
        Password = input("Enter Password : ")
        Email_Address = input("Enter Email_Address : ")
        Name = input("Name : ")
        Roll_Number = input("Enter Roll_Number : ")
        Phone = input("Enter Phone Number : ")
        Qry = ("UPDATE Student SET User_ID=%s, Password=%s, Email_Address=%s, Name=%s, Roll_Number=%s, Phone=%s WHERE User_ID = %s")
        data = (New_User_ID, Password, Email_Address, Name, Roll_Number, Phone, User_ID)
        Cursor.execute(Qry,data)
        cnx.commit()
        Cursor.close()
        cnx.close()
        print(Cursor.rowcount, "Record(s) Updated Successfully.")
    except mysql.connector.ERROR as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
    cnx.close()

```

Subject.py

```
import mysql.connector
from mysql.connector import errorcode
from mysql.connector import (connection)
import os

def insertSubject():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        Subject_ID = input("Enter Subject_ID : ")
        Subject_name = input("Enter Subject Name : ")
        Credits = int(input("Enter Credits : "))
        Teacher = input("Enter Teacher Name : ")
        Qry = ("INSERT INTO Subject VALUES(%s, %s, %s, %s)")
        data = (Subject_ID, Subject_name, Credits, Teacher)
        Cursor.execute(Qry, data)
        cnx.commit()
        Cursor.close()
        cnx.close()
        print("Record Inserted.")
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
        cnx.close()

def deleteSubject():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        Subject_ID = input("Enter Subject_ID to be deleted : ")
        Qry = ("DELETE FROM Subject WHERE Subject_ID = %s")
        del_rec = (Subject_ID,)
        Cursor.execute(Qry, del_rec)
        cnx.commit()
        Cursor.close()
        cnx.close()
        print(Cursor.rowcount, "Record(s) Deleted Successfully.")
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
```

```
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
        cnx.close()

def SearchSubject():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        Subject_ID = input("Enter Subject_ID to be Searched : ")
        query = ("SELECT * FROM Subject where Subject_ID = %s")
        rec_srch = (Subject_ID,)
        Cursor.execute(query, rec_srch)
        Rec_count = 0
        for (Subject_ID, Subject_name, Credits, Teacher) in Cursor:
            Rec_count += 1
            print("=====")
            print("Subject_ID : ", Subject_ID)
            print("Subject_name : ", Subject_name)
            print("Credits : ", Credits)
            print("Teacher : ", Teacher)
            print("=====")
            if Rec_count%2 == 0:
                input("Press any key to continue: ")
                clrscr()
                print(Rec_count, "Record(s) found")
        cnx.commit()
        Cursor.close()
        cnx.close()
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
        cnx.close()
```

```

def UpdateSubject():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        Subject_ID = input("Enter Subject_ID of Subject to be Updated : ")
        print("Enter new data: \n")
        New_Subject_ID = input("Enter Subject_ID : ")
        Subject_name = input("Enter Subject Name : ")
        Credits = int(input("Enter Credits : "))
        Teacher = input("Enter Teacher Name : ")
        Qry = ("UPDATE Subject SET Subject_ID=%s, Subject_name=%s, Credits=%s, Teacher=%s WHERE Subject_ID=%s")
        data = (New_Subject_ID, Subject_name, Credits, Teacher, Subject_ID)
        Cursor.execute(Qry, data)
        cnx.commit()
        Cursor.close()
        cnx.close()
        print(Cursor.rowcount, "Record(s) Updated Successfully.")
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
    cnx.close()

```

Exam.py

```

import mysql.connector
from mysql.connector import errorcode
from mysql.connector import (connection)
import os

def AddExam():
    cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
    Cursor = cnx.cursor()
    Exam_ID = input("Enter Exam_ID : ")
    User_ID = input("Enter User_ID : ")
    Subject_ID = input("Enter Subject_ID : ")
    Exam_Link = input("Enter Exam Link: ")
    Qry = ("INSERT INTO Exam VALUES (%s, %s, %s, %s)")
    data = (Exam_ID, User_ID, Subject_ID, Exam_Link)
    Cursor.execute(Qry, data)
    cnx.commit()
    Cursor.close()
    cnx.close()
    print("Record Inserted.")
    cnx.close()

def DeleteExam():
    cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
    Cursor = cnx.cursor()
    Exam_ID = input("Enter Exam_ID of Exam to be deleted: ")
    Qry = ("DELETE FROM Exam WHERE Exam_ID = %s")
    del_rec = (Exam_ID,)
    Cursor.execute(Qry, del_rec)
    cnx.commit()
    Cursor.close()
    cnx.close()
    print(Cursor.rowcount, "Record(s) Deleted Successfully.")
    cnx.close()

```

```

def ViewExam():
    try:
        cnx = mysql.connector.connect(host = 'localhost', port = 3306, user = 'root', password = 'Manav_Patel@2003', database = 'online_exam')
        Cursor = cnx.cursor()
        Exam_ID = input("Enter Exam_ID of Exam to be searched: ")
        query = ("SELECT * FROM Exam WHERE Exam_ID = %s ")
        rec_srch = (Exam_ID,)
        Cursor.execute(query, rec_srch)
        Rec_count = 0
        for (Exam_ID, User_ID, Subject_ID, Exam_Link) in Cursor:
            Rec_count += 1
            print("=====")
            print("Exam ID : ", Exam_ID)
            print("User ID : ", User_ID)
            print("Subject ID : ", Subject_ID)
            print("Exam link: ", Exam_Link)
            print("=====")
            if Rec_count%2 == 0:
                input("Press any key to continue")
                clrscr()
                print(Rec_count, "Record(s) found")
        cnx.commit()
        Cursor.close()
        cnx.close()
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)
        cnx.close()

```

TABLE DESCRIPTIONS:

```
mysql> use online_exam;
Database changed
mysql> show tables;
```

```
+-----+
| Tables_in_online_exam |
+-----+
| exam                   |
| student                |
| subject                |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> desc exam;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Exam_ID    | varchar(10)   | NO   | PRI | NULL     |       |
| User_ID    | varchar(10)   | YES  |     | NULL     |       |
| Subject_ID | varchar(10)   | YES  |     | NULL     |       |
| Exam_Link  | varchar(50)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
mysql> desc student;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| User_ID        | varchar(10)   | NO   | PRI | NULL     |       |
| Password       | varchar(20)   | YES  |     | NULL     |       |
| Email_Address  | varchar(40)   | YES  |     | NULL     |       |
| Name           | tinytext      | YES  |     | NULL     |       |
| Roll_Number    | varchar(20)   | YES  |     | NULL     |       |
| Phone          | varchar(10)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> desc subject;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Subject_ID     | varchar(10)   | NO   | PRI | NULL     |       |
| Subject_name    | tinytext      | YES  |     | NULL     |       |
| Credits         | int           | YES  |     | NULL     |       |
| Teacher        | tinytext      | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

OUTPUT:

1. STUDENT MANAGEMENT

INSERTING DATA

```
STUDENT RECORD MANAGEMENT
=====
1. Add Student
2. Search Student Record
3. Delete Student Record
4. Update Update Record
5. Return to Main Menu
=====
Enter your choice: 1
Enter User_ID : S001
Enter Password : S001@1
Enter Email Address : s001@gmail.com
Name : Manav Pandya
Enter Roll_Number : 001
Enter Phone Number : 1234567890
Record Inserted.

STUDENT RECORD MANAGEMENT
=====
1. Add Student
2. Search Student Record
3. Delete Student Record
4. Update Update Record
5. Return to Main Menu
=====
Enter your choice: 1
Enter User_ID : S002
Enter Password : S002@2
Enter Email Address : s002@gmail.com
Name : Malay Zalawadia
Enter Roll_Number : 002
Enter Phone Number : 7412589633
Record Inserted.
```

```
STUDENT RECORD MANAGEMENT
=====
1. Add Student
2. Search Student Record
3. Delete Student Record
4. Update Update Record
5. Return to Main Menu
=====
Enter your choice: 1
Enter User_ID : S003
Enter Password : S003@3
Enter Email Address : s003@gmail.com
Name : Manav Patel
Enter Roll_Number : 003
Enter Phone Number : 9632587410
Record Inserted.

STUDENT RECORD MANAGEMENT
=====
1. Add Student
2. Search Student Record
3. Delete Student Record
4. Update Update Record
5. Return to Main Menu
=====
Enter your choice: 1
Enter User_ID : S004
Enter Password : S004@4
Enter Email Address : s004@gmail.com
Name : Bhavy Masalia
Enter Roll_Number : 004
Enter Phone Number : 1239874565
Record Inserted.
```

```
STUDENT RECORD MANAGEMENT

=====
1. Add Student
2. Search Student Record
3. Delete Student Record
4. Update Update Record
5. Return to Main Menu
=====
Enter your choice: 1
Enter User_ID : S007
Enter Password : S007@7
Enter Email_Address : s007@gmail.com
Name : Dev Makwana
Enter Roll_Number : 007
Enter Phone_Number : 4561237777
Record Inserted.
```

AFTER INSERTING DATA

| User_ID | Password | Email_Address | Name | Roll_Number | Phone |
|---------|----------|----------------|-------------------|-------------|------------|
| S001 | S001@1 | s001@gmail.com | Manav Pandya | 001 | 1234567890 |
| S002 | S002@2 | s002@gmail.com | Malay Zalawadia | 002 | 7412589633 |
| S003 | S003@3 | s003@gmail.com | Manav Patel | 003 | 9632587410 |
| S004 | S004@4 | s004@gmail.com | Bhavy Masalia | 004 | 1239874565 |
| S005 | S005@5 | s005@gmail.com | Om Mehra | 005 | 3126459788 |
| S006 | S006@6 | s006@gmail.com | Sarthak Methaniya | 006 | 1231231231 |
| S007 | S007@7 | s007@gmail.com | Dev Makwana | 007 | 4561237777 |

SEARCHING

```
STUDENT RECORD MANAGEMENT

=====
1. Add Student
2. Search Student Record
3. Delete Student Record
4. Update Update Record
5. Return to Main Menu
=====
Enter your choice: 2
Enter User_ID of Student to be searched : S002
=====
User_ID : S002
Password : S002@2
Email_Address : s002@gmail.com
Name: Malay Zalawadia
Roll_Number : 002
Phone : 7412589633
=====
```


DELETING AND UPDATING

```
STUDENT RECORD MANAGEMENT

=====
1. Add Student
2. Search Student Record
3. Delete Student Record
4. Update Update Record
5. Return to Main Menu
=====
Enter your choice: 3
Enter User_ID of Student to be deleted : S002
1 Record(s) Deleted Successfully.
```

```
STUDENT RECORD MANAGEMENT

=====
1. Add Student
2. Search Student Record
3. Delete Student Record
4. Update Record
5. Return to Main Menu
=====
Enter your choice: 4
Enter User_ID of the Student to be Updated : S007
Enter new data
Enter User_ID : S007
Enter Password : S007@7
Enter Email_Address : s007@gmail.com
Name : Dev Makwana
Enter Roll_Number : 007
Enter Phone_Number : 1124457788
1 Record(s) Updated Successfully.
```

AFTER DELETING AND UPDATING

| User_ID | Password | Email_Address | Name | Roll_Number | Phone |
|---------|----------|----------------|-------------------|-------------|------------|
| S001 | S001@1 | s001@gmail.com | Manav Pandya | 001 | 3362251145 |
| S003 | S003@3 | s003@gmail.com | Manav Patel | 003 | 9632587410 |
| S004 | S004@4 | s004@gmail.com | Bhavy Masalia | 004 | 1239874565 |
| S005 | S005@5 | s005@gmail.com | Om Mehra | 005 | 3126459788 |
| S006 | S006@6 | s006@gmail.com | Sarthak Methaniya | 006 | 1231231231 |
| S007 | S007@7 | s007@gmail.com | Dev Makwana | 007 | 1124457788 |

2. SUBJECT MANAGEMENT

INSERTING DATA

```

                                SUBJECT RECORD MANAGEMENT
=====
1. Add Subject Record
2. Search Subject Record
3. Delete Subject Record
4. Update Subject Record
5. Return to Main Menu
=====
Enter your choice: 1
Enter Subject_ID : 001
Enter Subject Name : DBMS
Enter Credits : 4
Enter Teacher Name : Dr. Aparna Kumari
Record Inserted.

                                SUBJECT RECORD MANAGEMENT
=====
1. Add Subject Record
2. Search Subject Record
3. Delete Subject Record
4. Update Subject Record
5. Return to Main Menu
=====
Enter your choice: 1
Enter Subject_ID : 002
Enter Subject Name : OS
Enter Credits : 4
Enter Teacher Name : Mr. Ashwin Verma
Record Inserted.

                                SUBJECT RECORD MANAGEMENT
=====
1. Add Subject Record
2. Search Subject Record
3. Delete Subject Record
4. Update Subject Record
5. Return to Main Menu
=====
Enter your choice: 1
Enter Subject_ID : 003
Enter Subject Name : PS
Enter Credits : 3
Enter Teacher Name : Mr. Dhiren Pandit
Record Inserted.
```

AFTER INSERTING DATA

| Subject_ID | Subject_name | Credits | Teacher |
|------------|--------------|---------|-------------------|
| 001 | DBMS | 4 | Dr. Aparna Kumari |
| 002 | OS | 4 | Mr. Ashwin Verma |
| 003 | PS | 3 | Mr. Dhiren Pandit |
| 004 | PSC | 3 | Mr. Vishal Parikh |
| 005 | POM | 2 | Mr. Welala |

SEARCHING

```
SUBJECT RECORD MANAGEMENT

=====
1. Add Subject Record
2. Search Subject Record
3. Delete Subject Record
4. Update Subject Record
5. Return to Main Menu
=====
Enter your choice: 2
Enter Subject_ID to be Searched : 002
=====
Subject_ID : 002
Subject_name : OS
Credits : 4
Teacher : Mr. Ashwin Verma
=====
```

DELETING AND UPDATING

```
SUBJECT RECORD MANAGEMENT

=====
1. Add Subject Record
2. Search Subject Record
3. Delete Subject Record
4. Update Subject Record
5. Return to Main Menu
=====
Enter your choice: 3
Enter Subject_ID to be deleted : 005
1 Record(s) Deleted Successfully.

SUBJECT RECORD MANAGEMENT

=====
1. Add Subject Record
2. Search Subject Record
3. Delete Subject Record
4. Update Subject Record
5. Return to Main Menu
=====
Enter your choice: 4
Enter Subject_ID of Subject to be Updated : 004
Enter new data:

Enter Subject_ID : 004
Enter Subject Name : PSC
Enter Credits : 4
Enter Teacher Name : Mr. Vishal Parikh
1 Record(s) Updated Successfully.
```

AFTER DELETING AND UPDATING

| Subject_ID | Subject_name | Credits | Teacher |
|------------|--------------|---------|-------------------|
| 001 | DBMS | 4 | Dr. Aparna Kumari |
| 002 | OS | 4 | Mr. Ashwin Verma |
| 003 | PS | 3 | Mr. Dhiren Pandit |
| 004 | PSC | 4 | Mr. Vishal Parikh |

3. EXAM MANAGEMENT

INSERTING DATA

```

MEMBER RECORD MANAGEMENT
=====
1. Add Exam
2. Delete Exam
3. View Exam
4. Return to Main Menu
=====
Enter your choice: 1
Enter Exam_ID : E001
Enter User_ID : S002
Enter Subject_ID : 004
Enter Exam Link: PSC004
Record Inserted.

MEMBER RECORD MANAGEMENT
=====
1. Add Exam
2. Delete Exam
3. View Exam
4. Return to Main Menu
=====
Enter your choice: 1
Enter Exam_ID : E002
Enter User_ID : S001
Enter Subject_ID : 001
Enter Exam Link: DBMS001
Record Inserted.

MEMBER RECORD MANAGEMENT
=====
1. Add Exam
2. Delete Exam
3. View Exam
4. Return to Main Menu
=====
Enter your choice: 1
Enter Exam_ID : E003
Enter User_ID : S007
Enter Subject_ID : 003
Enter Exam Link: PS003
Record Inserted.
```

AFTER INSERTING DATA

| Exam_ID | User_ID | Subject_ID | Exam_Link |
|---------|---------|------------|-----------|
| E001 | S002 | 004 | PSC004 |
| E002 | S001 | 001 | DBMS001 |
| E003 | S007 | 003 | PS003 |

SEARCHING

```
MEMBER RECORD MANAGEMENT

=====
1. Add Exam
2. Delete Exam
3. View Exam
4. Return to Main Menu
=====
Enter your choice: 3
Enter Exam_ID of Exam to be searched: E001
=====
Exam ID : E001
User ID : S002
Subject ID : 004
Exam link: PSC004
=====
```

DELETING DATA

```
MEMBER RECORD MANAGEMENT

=====
1. Add Exam
2. Delete Exam
3. View Exam
4. Return to Main Menu
=====
Enter your choice: 2
Enter Exam_ID of Exam to be deleted: E001
1 Record(s) Deleted Successfully.
```

AFTER DELETING DATA

| Exam_ID | User_ID | Subject_ID | Exam_Link |
|---------|---------|------------|-----------|
| E002 | S001 | 001 | DBMS001 |
| E003 | S007 | 003 | PS003 |

CONCLUSION:

The project successfully used various functionalities of python and MySQL also create the fully functional database management system for online portals. Using MySQL as the database is highly beneficial as it is free to download, popular and can be easily customized. The data stored in the MySQL database can easily be retrieved and manipulated according to the requirements with basic knowledge of SQL. Also we got to learn how to connect python to MySQL and maintain crud operations on database. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer.