Vikas Vidyalaya Begusarai

VIKAS VIDYALAYA

विद्यैव बलम्

SESSION:-2021-2022

COMPUTER SCIENCE PROJECT

STD :- 12th

SUBMITTED BY :

PRIYANSHU RAJ

CLASS :- XII - A

ROLL NO : 22602243

SUBMITTED TO:

MR.CHATURANAN JHA

PROJECT BASED ON

SCHOOL MANAGEMENT SYSTEM

Certificate

Roll No: 22602243 Exam No:

This is to certify that **Priyanshu Raj** student of class 12th has successfully completed the research on the below mentioned project under the guidance of **Mr. Chaturanan Jha** during the year of 2021-22 in partial fulfillment of computer science practical examination conducted by Vikas Vidyalaya, Begusarai.

SIGNATURE OF
EXTERNAL
EXAMINER

SIGNATURE OF
INTERNAL
EXAMINER

Principal

ACKNOWLEDGEMENT

In the accomplishment of this project successfully, many people have best owned upon us their blessings and the heart pledge support, this time we are utilizing to thank all the people who have been concerned with this project.

Primarily we would like thank god for being able to complete this project with success. Then we would like to thank my principal **Mr. Manoj Kumar** and our Computer science teacher **Mr. Chaturanan Jha** whose valuable guidance has been the ones that helped us patch this project and make it full proof success, his suggestions and instruction has served as the major contribution towards the completion of this project.

Then we would like to thank our parents who have helped us with their valuable suggestions and guidance has been very helpful in various phases of the completion of the project.

CONTENTS

PRATICULARS

1. Preface
2. Aim and Objective
3. System Discription at a Glance
4. System Requirements Specifications
5. Hardware and Software Requirements
6. Structure of the Table used
7. Forms and data environment used
8. Future Extensions
9. Python Programs
10. Conclusion
11. Limitations

PREFACE

The Central Board Of Secondary Education has included in its course, a full-fledged computer course covering the fundamentals of computer and programming. Exploring the world of computers, and the project is both informative and exciting. The project “School Management System” has been allotted to me.

This project work allocated to me is a part of the entire process involved in computerization of the School Management System.

AIM AND OBJECTIVES

The project “School Management System” fas been allocated to me with the aim of testing the knowledge of computer as a subject and to make them realize the problems that come during the software development process. Thus, the methodology adopted by the Central Board of Secondary Education is to make the examination process more practical based and realistic by familiarizing the students with real fife situations. Its main aims and objectives are:

- To develop a database management system based on MySOL for maintaining the records relating to the management of the organization.

- To develop programs to record the details of the students, fees, and admission of the school

SYSTEM DISCRIPTION AT A GLANCE

A School Management System to record the System of Students Admission, Student records entry and fees deposit that involves several entries relating to New admission of the student of school and enter students records and display, manipulate and also can record students fee details. Thus recording of entries becomes easy and the expenditures incurred in the working of the organization can be easily derived.

SYSTEM REQUIREMENTS OF THE PROJECT

Recommended System Requirements Processors: Intel® Core™ i3 processor 4300M at 2.60 GHz. Disk space: 2 to 4 GB. Operating systems: Windows® 10, MACOS, and UBUNTU. Python Versions: 3.X.X or Higher.

Minimum System Requirements Processors: Intel Atom® processor or Intel® Core™ i3 processor. Disk space: 1 GB. Operating systems: Windows 7 or later, MACOS, and UBUNTU. Python Versions: 2.7.X, 3.6.X.

Prerequisites before installing MySQL Connector Python

You need root or administrator privileges to perform the installation process. Python must be installed on your machine.

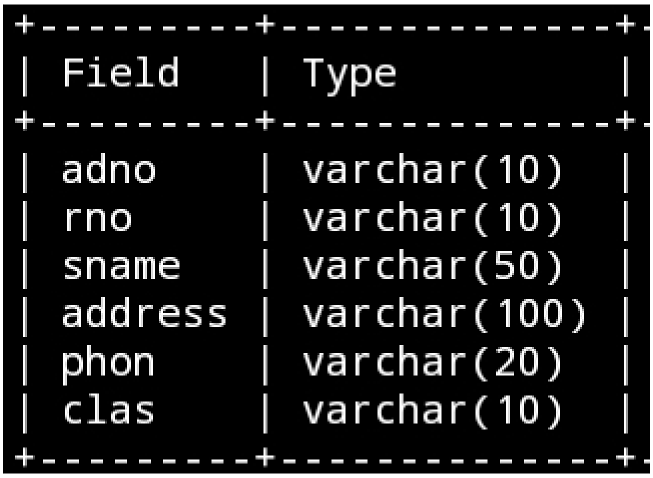
Note: – MySQL Connector Python requires python to be in the system's PATH. Installation fails if it doesn't find Python.

On Windows, If Python doesn't exist in the system's PATH, please manually add the directory containing python.exe yourself.

STRUCTURE OF TABLE

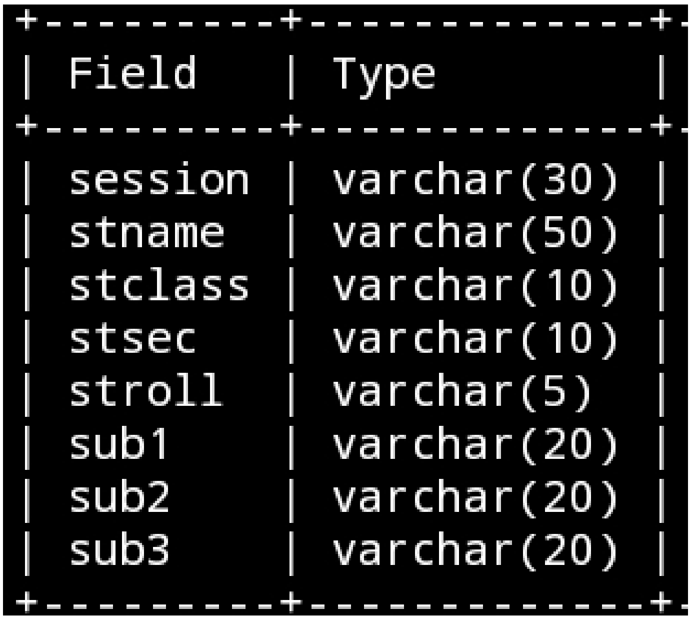
CREATE DATABASE **MPS**

CREATE TABLE Admission



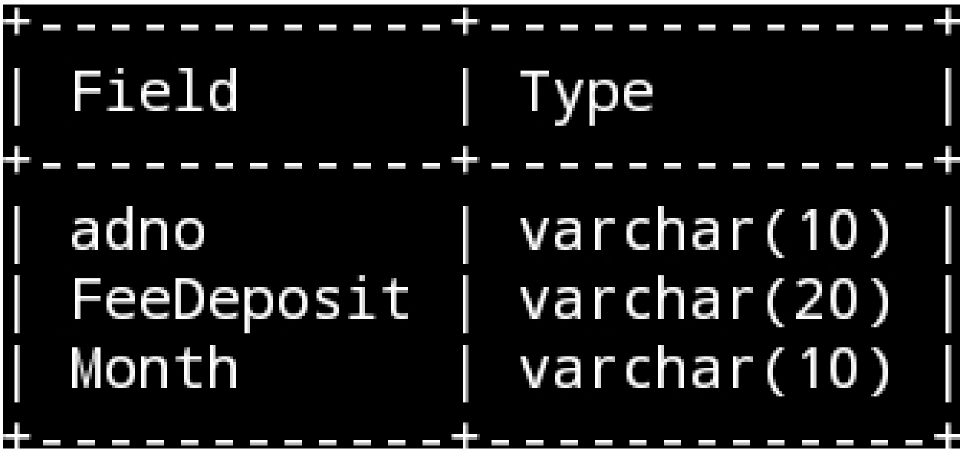
```
+---------+-------------+
| Field   | Type        |
+---------+-------------+
| adno    | varchar(10) |
| rno     | varchar(10) |
| sname   | varchar(50) |
| address | varchar(100)|
| phon    | varchar(20) |
| clas    | varchar(10) |
+---------+-------------+
```

CREATE TABLE Students



```
+---------+-------------+
| Field   | Type        |
+---------+-------------+
| session | varchar(30) |
| stname  | varchar(50) |
| stclass | varchar(10) |
| stsec   | varchar(10) |
| stroll  | varchar(5)  |
| sub1    | varchar(20) |
| sub2    | varchar(20) |
| sub3    | varchar(20) |
+---------+-------------+
```

CREATE TABLE Fees



```
+-------------+--------------+
| Field       | Type         |
+-------------+--------------+
| adno        | varchar(10)  |
| FeeDeposit  | varchar(20)  |
| Month       | varchar(10)  |
+-------------+--------------+
```

FORM AND DATA ENVIRONMENT USED

PYTHON FORMS:

1. **Admission**
    1. **Add new admission details**
    2. **Display admission details**
    3. **Search admission details**
    4. **Delete admission details**
    5. **Update admission details**

1. **Student Data**
    2. **Enter student record**
    3. **Display student record**
    4. **Search student record**
    5. **Delete student record**
    6. **Update student record**

1. **Fees details**
    2. **Deposit fees**
    3. **Display fees details**
    4. **Display fees details of a Particular Student**

MYSQL TABLES:

A. **Admission:** This table records all admission details.
B. **Student:** This table is used to record student details.
C. **Fees:** This table is used to store fees details.

FUTERE EXTENSIONS

- There can be a provtsion for incClding Examination management
- There can be proviszon for including School employee management
- There can be a proviston School transport management
- There can be a provision Online data entry
- There can be a provision for printing am reports
- There can be a provision for entering mufti-level password.
- There can be a provision for receiving feedbacks form the students and parents.

PYTHON PROGRAMS

Main_Menu.py:

```
import main_menu
import admission
import student_data
import fee_details

while                              True:
  print("\t\t....................................................")
  print("\t\t..... *********SCHOOL MANAGEMENT
SYSTEM***********")
  print("\t\t....................................................")
  print("\n\t\t*****************VIKAS VIDYALAYA
BEGUSARAI******************")
  print("*1. Admission*")
  print("*2. Student Data*")
  print("*3. Fee Details*")
  print("*4. Exit*")
  print("\t\t....................................................")
  print("\t\t-------------------------------------------------")
  choice=int(input("Enter your choice : "))
  if choice==1:
    admission.adm_menu()
  elif choice==2:
    student_data.stu_menu()
  elif choice==3:
    fee_details.fee_menu()
  elif choice==4:
    break
  else:
    print("Error: Invalid Choice try again..")
    conti=input("press any key ti continue..")
```

Admission.py:

```
import main_menu
import admission
import mysql.connector as co
def adm_menu():
   while                                    True:
      print("\t\t.............................................. ")
      print("\t\t.....*********School Management
System***********")
      print("\t\t.............................................. ")
      print("\n**Admission**\n")
      print("*1. Add New Admission Details*")
      print("*2. Show Admission Details*")
      print("*3. Search Admission record*")
      print("*4.  Deletion of Record")
      print("*5. Update Admission Details*")
      print("*6. Return*")
      print("\t\t-----------------------------------------------")
      choice=int(input("Enter your choice : "))
      if choice==1:
         admission.admin_details()
      elif choice==2:
         admission.show_admin_details()
      elif choice==3:
         admission.search_admin_details()
```

```
        elif choice==4:
            admission.delete_admin_details()
        elif choice==5:
            admission.edit_admin_details()
        elif choice==6:
            return
        else:
            print("Error: Invalid Choice try again..")
            conti=input("press any key ti continue..")

def admin_details():
    try:
        mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
        cursor=mycon.cursor()
        adno=input("Enter Admission No.: ")
        rno=input("Enter Role No.: ")
        sname=input("Enter Student name No.: ")
        address=input("Enter Address: ")
        phon=input("Enter Mobile No.: ")
        clas=input("Enter Class: ")
        query="insert into Admission(adno,rno,sname,address,
phon,clas) value('{}','{}','{}','{}','{}','{}')".format(adno,rno,
sname,address,phon,clas)
        cursor.execute(query)
        mycon.commit()
        mycon.close()
        cursor.close()
        print('Record has been saved in admission table')
```

```
except:
        print('error')

def show_admin_details():
    mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    cursor.execute("Select * from Admission")
    data = cursor.fetchall()
    for row in data:
        print(row)
def search_admin_details():
    mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    adn=input("Enter Admission Number: ")
    st="select * from Admission where adno='%s'"%(adn)
    cursor.execute(st)
    data = cursor.fetchall()
    print(data)

def delete_admin_details():
    mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    adn=input("Enter Admission Number: ")
    st="delete from admission where adno='%s'"%(adn)
    cursor.execute(st)
    mycon.commit()
    print("Record has been deleted")
```

```
def edit_admin_details():
    mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    print("1: Edit Name: ")
    print("2: Edit Address: ")
    print("3: Phone number: ")
    print("4: Return: ")
    print("\t\t----------------------------------------------------")
    choice = int(input("Enter your choise: "))
    if choice == 1:
        admission.edit_name()
    elif choice == 2:
        admission.edit_address()
    elif choice == 3:
        admission.edit_phno()
    elif choice == 4:
        return
    else:
        print("Error: Invalid Choise try again .... ")
        conti="Press any key to return to "

def edit_name():
    mycon =co.connect(host="localhost",user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    ac=input("Enter Admission no: ")
    nm=input("Enter correct name: ")
    st = "update Admission set sname='%s' where adno =
'%s'"%(nm,ac)
```

```
    cursor.execute(st)
    mycon.commit()
    print('Data updated successfully')
def edit_address():
    mycon =co.connect(host="localhost",user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    ac=input("Enter Admission no: ")
    nm=input("Enter correct address: ")
    st = "update Admission set address='%s' where adno =
'%s'"%(nm,ac)
    cursor.execute(st)
    mycon.commit()
    print('Data updated successfully')
def edit_phno():
    mycon =co.connect(host="localhost",user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    ac=input("Enter Admission no: ")
    nm=input("Enter correct Phone: ")
    st = "update Admission set phon='%s' where adno =
'%s'"%(nm,ac)
    cursor.execute(st)
    mycon.commit()
    print('Data updated successfully')
```

student_data.py:

```
import main_menu
import student_data
import mysql.connector as co

def stu_menu():
   while True:
      print("\t\t............................................. ")
      print("\t\t.....*********SCHOOL MANAGEMENT
SYSTEM***********")
      print("\t\t............................................. ")
      print("\n\t\t******************STUDENT
DATA******************")
      print("*1. Add Student Record*")
      print("*2. Show Student Records*")
      print("*3. Search Student record*")
      print("*4. Deletion of Record")
      print("*5. Update Student Record*")
      print("*6. Return*")
      print("\t\t-----------------------------------------------")
      choice=int(input("Enter your choice : "))
      if choice==1:
         student_data.add_record()
      elif choice==2:
         student_data.show_stu_details()
      elif choice==3:
         student_data.search_stu_details()
      elif choice==4:
         student_data.delete_stu_details()
```

```
        elif choice==5:
           student_data.edit_stu_details()
        elif choice==6:
           return
        else:
           print("Error: Invalid Choice try again..")
           conti=input("press any key ti continue..")

def add_record():
   try:
        mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
        cursor=mycon.cursor()
        session=input("Enter Session: ")
        stname=input("Enter Student Name: ")
        stclass=input("Enter Class: ")
        stsec=input("Enter Section: ")
        stroll=input("Enter Roll No.: ")
        sub = []
        for i in range(3):
           sb = input(f"Enter subject {i+1}: ")
           sub.append(sb)
        query="insert into Student() value('{}','{}','{}','{}','{}','{}','{}',
'{}')".format(session,stname,stclass,stsec,stroll,sub[0],sub[1],
sub[2])
        cursor.execute(query)
        mycon.commit()
        mycon.close()
        cursor.close()
        print('Record has been saved in admission table')
```

```
    except:
        print('error')

def show_stu_details():
    mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    cursor.execute("Select * from Student")
    data = cursor.fetchall()
    for row in data:
        print(row)

def search_stu_details():
    mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    adn=input("Enter Admission Number: ")
    st="select * from Student where stroll='%s'"%(adn)
    cursor.execute(st)
    data = cursor.fetchall()
    print(data)

def delete_stu_details():
    mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    adn=input("Enter Admission Number: ")
    st="delete from Student where stroll='%s'"%(adn)
    cursor.execute(st)
    mycon.commit()
    print("Record has been deleted")
def edit_stu_details():
    mycon=co.connect(host="localhost", user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    print("*1: Edit Name* ")
    print("*1: Edit First Subject* ")
    print("*3: Edit Second Subject* ")
    print("*4: Edit Third Subject* ")
    print("*5: Return* ")
    print("\t\t-------------------------------------------------------")
    choice = int(input("Enter your choise: "))
    if choice == 1:
        student_data.edit_name()
    elif choice == 2:
        student_data.edit_sub1()
    elif choice == 3:
        student_data.edit_sub2()
    elif choice == 4:
         student_data.edit_sub3()
    elif choice == 5:
        return
    else:
        print("Error: Invalid Choise try again .... ")
        conti="Press any key to return to "

def edit_name():
    mycon =co.connect(host="localhost",user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    ac=input("Enter Roll no: ")
    nm=input("Enter Correct name: ")
    st = "update Student set stname='%s' where stroll =
'%s'"%(nm,ac)
    cursor.execute(st)
    mycon.commit()
    print('Data updated successfully')
def edit_sub1():
    mycon =co.connect(host="localhost",user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    ac=input("Enter Roll no: ")
    nm=input("Enter Correct Subject: ")
    st = "update Student set sub1='%s' where stroll = '%s'"%(nm,
ac)
    cursor.execute(st)
    mycon.commit()
    print('Data updated successfully')
def edit_sub2():
    mycon =co.connect(host="localhost",user="root",
passwd="root", database="MPS")
    cursor=mycon.cursor()
    ac=input("Enter Roll no: ")
    nm=input("Enter Correct Subject: ")
    st = "update Student set sub2='%s' where stroll = '%s'"%(nm,
ac)
    cursor.execute(st)
    mycon.commit()
    print('Data updated successfully')
```

```
def edit_sub3():
   mycon =co.connect(host="localhost",user="root",
passwd="root", database="MPS")
   cursor=mycon.cursor()
   ac=input("Enter Roll no: ")
   nm=input("Enter Correct Subject: ")
   st = "update Student set sub3='%s' where stroll = '%s'"%(nm,
ac)
   cursor.execute(st)
   mycon.commit()
   print('Data updated successfully')
```

fee_details.py:

```
import main_menu
import fee_details
import mysql.connector

def fee_menu():
	while True:
		print("\t\t .................................................... ")
		print("\t\t ....*********SCHOOL MANAGEMENT 
SYSTEM***********")
		print("\t\t .................................................... ")
		print("\n **FEE DETAILS**\n")
		print("*1 : Deposit Fee*")
		print("*2 : View Fee of All Students*")
		print("*3 : View Fee of a Particular Student*")
		print("*4 : Return*")
		try:
			userInput = int(input("Please Select An Above 
Option: "))
		except ValueError:
			exit("\nHy! That's Not A Number")
		else:
			print("\n")
			if (userInput==1):
				fee_details.feeDeposit()
			elif (userInput==2):
				fee_details.feeView()
			elif (userInput==3):
				fee_details.feeViewPart()
```

```
                    elif (userInput==4):
                            return
            print("-----------------------------------------------")


def feeDeposit():
        mydb=mysql.connector.connect(host="localhost",
user="root",passwd="root",database="MPS")
        mycursor=mydb.cursor()
        L=[]
        roll=int(input("Enter the Admission number : "))
        L.append(roll)
        feedeposit=int(input("Enter the Fee to be deposited : "))
        L.append(feedeposit)
        month=input("Enter month of fee : ")
        L.append(month)
        fee=(L)
        sql="insert into Fees (adno,FeeDeposit,Month) values (%s,
%s,%s)"
        mycursor.execute(sql,fee)
        mydb.commit()
        print ("Fee has been Deposited Succefully!!!")

def feeView():
        print ("*ALL FEE DETAILS*")
        mydb=mysql.connector.connect(host="localhost",
user="root",passwd="root",database="MPS")
        mycursor=mydb.cursor()
```

```
    sql="Select Admission.adno, Admission.sname,
Admission.clas, sum(Fees.FeeDeposit), count(Fees.month)
from Admission,Fees where Admission.adno=Fees.adno Group
by adno"
    mycursor.execute(sql)
    res=mycursor.fetchall()
    month = ['April','May','June','July','August','September',
'October','November','December','January', 'February','March']
    for x in res:
        x =  list(x)
        a = x.pop()
        x.append(month[a-1])
        print(x,end = ' ')
        print (f"  Fee left from {month[a]}")
    print('\n','\n')

def feeViewPart():
    mydb=mysql.connector.connect(host="localhost",
user="root",passwd="root",database="MPS")
    mycursor=mydb.cursor()
    admno=int(input("Enter the Admission number of the
Student : "))
    sql="Select Admission.adno, Admission.sname,
Admission.clas, sum(Fees.FeeDeposit), count(Fees.month)
from Admission INNER JOIN Fees ON Admission.adno=Fees.
adno and Fees.adno = %s"
    adm=(admno,)
    mycursor.execute(sql,adm)
    res=mycursor.fetchall()
    month = ['April','May','June','July','August','September',
'October','November','December','January', 'February','March']
    for x in res:
        x =  list(x)
        a = x.pop()
        x.append(month[a-1])
        print('\n',x,'\n')
        print (f"Fee left from {month[a]}")
    print('\n','\n')
```

CONCLUSION

This software has its advantages and disadvantages but it can surely help with the record storage system. We don't have to worry about the misplacing of record which is a great clash while storing the record on separate files.

Limitations

1) Does not support

2) If some string is given as input i.e in place where should have been input, the program crashes and data get spoiled.

3) This project can only work in particular (CODEBLOCKS).