ALL SAINT'S CHURCH SR.SEC. SCHOOL

M.I. ROAD , JAIPUR

A PROJECT REPORT ON

SCHOOL MANAGEMENT SYSTEM

SUBJECT: INFORMATICS PRACTICES(065)

Session: 2023-2024

SUBMITTED BY-

Mohd. Shayyan

SUBMITTED TO-

Mrs. Sharon Amus

ACKNOWLEDGMENT

I would like to express our special thanks of

Gratitude to my teacher Mrs. Sharon Amus for

Mentoring me for this project work. I also thank

our principal Mrs. Shabnam Haque for her motivation and guidance.

My project is titled as School Management System and

It enables me to do a lot of research and I came to Know about so many new things.

Secondly I would also like to thank our parents and friends who helped us a lot in finalizing this project Within the limited time frame.

Mohd. Shayyan

XII SCI

CONTENT

1.Introduction
2.System Implementation
3.Screenshots
4.User Output
5.SQL Queries
6.User Interface Code
7.Testing
8.Bibliography

INTRODUCTION

The "School Management System" undertaken as a project under IP is based on PYTHON AND MYSQL.

It's an attempt to automate the existing system.

The project enables its user to perform few

operations pertaining to management of School.

The Project Enables its user to:

1.) Add a new Student, new Staff and new Fee record's.

2.) Delete Student, Staff and Fee record's.

3.) Update Student, Staff and Fee record's.

4.) View Student, Staff and Fee record's from the Database.

System Implementation

The hardware used =>

----------| System |----------

Processor-	**Intel(R) Core(TM) i5 7300U CPU 2.60GHz 2.71 GHz**
Installed memory[RAM]-	**8.00 GB(7.88 GB usable)**
System Type-	**64-bit operating system, X64-based processor**
Pen and Touch-	**No pen or touch input is Available for this display**

Screenshots

PROJECT TITLE- “SCHOOL MANAGEMENT”
DBMS: MySQL
Host : localhost
User: root
Password: root
Database: School
Table Structure: As per the Screenshot given below:

Table: Student

```
mysql> Desc students;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| Id      | varchar(255) | YES  |     | NULL    |       |
| name    | varchar(255) | YES  |     | NULL    |       |
| age     | varchar(255) | YES  |     | NULL    |       |
| gender  | varchar(255) | YES  |     | NULL    |       |
| room_no | varchar(255) | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
5 rows in set (0.06 sec)
```

Table: Staff

```
mysql> Desc Staff;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Id     | varchar(50) | YES  |     | NULL    |       |
| post   | varchar(50) | YES  |     | NULL    |       |
| name   | varchar(50) | YES  |     | NULL    |       |
| salary | varchar(50) | YES  |     | NULL    |       |
| phone  | varchar(50) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Table: Fee

```
mysql> Desc fee;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| SrNo    | varchar(50) | YES  |     | NULL    |       |
| Name    | varchar(50) | YES  |     | NULL    |       |
| Class   | varchar(50) | YES  |     | NULL    |       |
| Status  | varchar(50) | YES  |     | NULL    |       |
| Quarter | varchar(50) | YES  |     | NULL    |       |
| PaidAmt | varchar(50) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

USER OUTPUT

STUDENT DETAILS:

```
                        ---| Modules in School Management System |---

[1.]->| Student record Module |                          [2.]->| Staff record Module |
[3.]->| Fee record Module |                              [4.]->| Exit |




Enter your choice: 1

[1.]->| Add New Student record |                         [2.]->|  View Student details |
[3.]->| Update Student details |                         [4.]->| Delete Student details |

Enter your choice:
```

STAFF DETAILS:

```
                        ---| Modules in School Management System |---

[1.]->| Student record Module |                          [2.]->| Staff record Module |
[3.]->| Fee record Module |                              [4.]->| Exit |




Enter your choice: 2
[1.]->| Add New Staff record |                          [2.]->| View Staff details |
[3.] ->| Update Staff details |                         [4.]->| Delete Staff details |
Enter your choice:
```

FEE DETAILS:

```
                          ---| Modules in School Management System |---

[1.]->| Student record Module |                                    [2.]->| Staff record Module |
[3.]->| Fee record Module |                                        [4.]->| Exit |




Enter your choice: 3
[1.]->| Add Fee deposit details |                                   [2.]->| View Fee datails |
[3.]->| Update Fee datails |                                       [4.]->| Delete Fee datails |
Enter your choice:
```

EXIT DETAILS:

```
                          ---| Modules in School Management System |---

[1.]->| Student record Module |                                    [2.]->| Staff record Module |
[3.]->| Fee record Module |                                        [4.]->| Exit |




Enter your choice: 4

Exited !
Succesfully,
Thanks
For
Coming :-)
```

SQL QUERIES

```
Create database school;

use school;

CREATE TABLE students (Id VARCHAR(255),name
VARCHAR(255), age VARCHAR(255), gender
VARCHAR(255), room_no VARCHAR(255));

Desc students;

use school;

create table Staff(Id varchar(50),post varchar(50),
name varchar(50),salary varchar(50),phone
varchar(50))

Desc Staff;

use school;

create table fee(SrNo varchar(50),Name
varchar(50),Class varchar(50),Status varchar(50),
Quarter varchar(50),PaidAmt varchar(50));

Desc fee;
```

USER INTERFACE CODE

```
print("*" * 135)

print("                          ---| Welcome to  School Management System by Shayyan|---\n")

print("*" * 135)

# Connecting from the server

userName=input("\n ENTER MYSQL SERVER'S USERNAME: ")

print("*"*135)

password=input(" ENTER MYSQL SERVER'S PASSWORD: ")

print("*"*135)

mydb = mysql.connector.connect(  host="localhost",  user=userName,  password=password,
database="school")

print(mydb,"connected to server")

print("\n")

print("\n")

    print()

    print("                      ---| Modules in School Management System |---")

    print()

    print("[1.]->| Student record Module |                              [2.]->| Staff record Module |")

    print("[3.]->| Fee record Module |                                  [4.]->| Exit |                  \n")

# Get the user's choice:

# if option first:

def getchoice():

    while True:

        menu()

        print("\n\n")

        option = input("Enter your choice: ")
```

```
if option=='1':
        print("\n[1.]->| Add New Student record |                    [2.]->|  View Student details |")
        print("[3.]->| Update Student details |                    [4.]->| Delete Student details \n")
        opp = input("Enter your choice: ")

    if opp=='1':
        add_student()
        input("Press ENTER KEY to continue.....")
        print()

    elif opp=='2':
        view_students()
        input("Press ENTER KEY to continue.....")
        print()

    elif opp=='3':
        update_student()
        input("Press ENTER KEY to continue.....")
        print()

    elif opp=='4':
        delete_student()
        input("Press ENTER KEY to continue.....")
        print()
```

```
## if option Second:
        elif option=='2':
            print("[1.]->| Add New Staff record |                    [2.]->| View Staff details | ")
            print("[3.] ->| Update Staff details |                    [4.]->| Delete Staff details | ")
            opp =input("Enter your choice: ")
            if opp=='1':
                add_staff()
                input("Press ENTER KEY to continue.....")
                print()
            elif opp=='2':
                view_staff()
                input("Press ENTER KEY to continue.....")
                print()
            elif opp=='3':
                update_staff()
                input("Press ENTER KEY to continue.....")
                print()
```

```
            elif opp=='4':
                delete_staff()
                input("Press ENTER KEY to continue.....")
                print()
### if option Third:
        elif option=='3':
            print("[1.]->| Add Fee deposit details |                    [2.]->| View Fee datails | ")
            print("[3.]->| Update Fee datails |                         [4.]->| Delete Fee datails | ")
            opp = input("Enter your choice: ")

           if opp=='1':
                fee()
                input("Press ENTER KEY to continue.....")
                print()
            elif opp=='2':
                view_fee()
                input("Press ENTER KEY to continue.....")
                print()

            elif opp=='3':
                update_fee()
                input("Press ENTER KEY to continue.....")
                print()
```

```
            elif opp=='4':
                delete_fee()
                input("Press ENTER KEY to continue.....")
                print()


#### if option Fourth:
        elif option=='4':
            print()
            print("Exited !")
            print("Succesfully,")
            print("Thanks")
            print("For")
            print("Coming :-)")
            print()
            print()
            print()
            print()
            break
```

ADD STUDENT RECORD

```
# Define the function to add a new student

def add_student():

    Id=input("Enter Student SrNo: ")

    name = input("Enter student Name: ")

    age = input("Enter student DOB: ")

    gender = input("Enter student gender: ")

    room_no = input("Enter student Class: ")

    cursor = mydb.cursor()

 # CREATING A TABLE

cursor.execute('CREATE TABLE students (Id VARCHAR(255) Primary Key ,name VARCHAR(255) ,
VARCHAR(255), gender VARCHAR(255), room_no VARCHAR(255))')


# Inserting Values

    sql = "INSERT INTO students (Id,name, age, gender, room_no) VALUES (%s,%s, %s, %s, %s)"

    val = (Id,name, age, gender, room_no)

    cursor.execute(sql, val)

    mydb.commit()

    print(cursor.rowcount, "record(s) inserted.")
```

DELETE STUDENT RECORD

```
# Define the function to delete student details
def delete_student():
    Id = input("Enter student SrNo: ")
    cursor = mydb.cursor()
    sql = "DELETE FROM students WHERE Id = %s"
    val = (Id,)
    cursor.execute(sql, val)
    mydb.commit()
    print(cursor.rowcount, "record(s) deleted.")
```

VIEW STUDENT RECORD

```
# Define the function to view student details
def view_students():
    cursor = mydb.cursor()
    cursor.execute("SELECT * FROM students")
    result = cursor.fetchall()
    for row in result:
        print(row)
```

UPDATE STUDENT RECORD

```
# Define the function to update student details
def update_student():
    id = input("Enter student SrNo: ")
    name = input("Enter student Name: ")
    age = input("Enter student DOB: ")
    gender = input("Enter student gender: ")
    room_no = input("Enter student Class: ")
    cursor = mydb.cursor()
    sql = "UPDATE students SET name = %s, age = %s, gender = %s, room_no = %s WHERE id = %s"
    val = (name, age, gender, room_no, id)
    cursor.execute(sql, val)   mydb.commit()    print(cursor.rowcount, "record(s) updated.")
```

ADD STAFF RECORD

```
# Define the function to add a new staff
def add_staff():
    Id=input("Enter staff ID: ")
    post=input("Enter staff Post: ")
    name = input("Enter staff Name: ")
    salary = input("Enter staff Salary: ")
    phone = input("Enter staff Phone no: ")
    cursor = mydb.cursor()
```

```
    # CREATING A TABLE

#    cursor.execute('create table Staff(Id varchar(50) Primary Key, post varchar(50),name varchar(50),salary varchar(50),phone varchar(50))')

# Inserting Values

    sqls = "INSERT INTO staff (Id,post,name,salary,phone) VALUES (%s,%s,%s, %s, %s)"

    vals = (Id,post,name,salary,phone)

    cursor.execute(sqls, vals)

    mydb.commit()

    print(cursor.rowcount, "record(s) inserted.")
```

UPDATE STAFF RECORD

```
# Define the function to update staff details

def update_staff():

    Id=input("Enter staff ID: ")

    post=input("Enter staff Post: ")

    name = input("Enter staff Name: ")

    salary = input("Enter staff Salary: ")

    phone = input("Enter staff Phone no: ")

    cursor = mydb.cursor()

    sql = "UPDATE staff SET post= %s, name = %s, salary = %s, phone = %s WHERE Id = %s"

    val = (Id,post,name,salary, phone)

    cursor.execute(sql, val)

    mydb.commit()

   print(cursor.rowcount, "record(s) updated.")
```

DELETE STAFF RECORD

```
# Define the function to delete staff details

def delete_staff():

    Id = input("Enter staff ID: ")

    cursor = mydb.cursor()

    sql = "DELETE FROM staff WHERE Id = %s"

    val = (Id,)

    cursor.execute(sql, val)

    mydb.commit()    print(cursor.rowcount, "record(s) deleted.")
```

VIEW STAFF RECORD

```
# Define the function to view student details

def view_staff():

    cursor = mydb.cursor()

    cursor.execute("SELECT * FROM staff")

    result = cursor.fetchall()

    for row in result:

        print(row)
```

ADD FEE RECORD

```
# Define the function to add Fee details

def fee():

    student_id =input("Enter Payer SrNo: ")

    Name = input("Enter Payer Name: ")

    Class = input("Enter Payer Class: ")

    Status= input("Enter Status(Paid/Due) : ")

    Quarter= input("Enter Quarter : ")

    PaidAmt= input("Enter PaidAmt : ")

 cursor = mydb.cursor()

  # CREATING A TABLE

 cursor.execute('CREATE TABLE fee (student_id VARCHAR(255), Name VARCHAR(255), Class
VARCHAR(255), Status VARCHAR(255), Quarter VARCHAR(255), PaidAmt VARCHAR(255), FOREIGN
KEY (student_id) REFERENCES students(Id))')

# Inserting Values

    msql = "INSERT INTO fee (student_id,Name,Class,Status,Quarter,PaidAmt) VALUES (%s,%s, %s, %s,

    %s,%s)"

    valu = (student_id,Name,Class,Status,Quarter,PaidAmt)

    cursor.execute(msql, valu)

    mydb.commit()

    print(cursor.rowcount, "record(s) inserted.")
```

UPDATE FEE RECORD

```
# Define the function to update Fee details
def update_fee():
    student_id = input("Enter student Id: ")
    Name = input("Enter student Name: ")
    Class = input("Enter student Class: ")
    Status = input("Enter student Status(Paid/Due): ")
    Quarter = input("Enter student Quarter: ")
    PaidAmt = input("Enter student PaidAmt: ")
    cursor = mydb.cursor()
    sqlx = "UPDATE fee SET Name = %s, Class = %s, Status = %s, Quarter = %s,PaidAmt = %s WHERE
    student_id = %s
     valx = (Name,Class,Status,Quarter,PaidAmt, student_id)
     cursor.execute(sqlx, valx)
    mydb.commit()
    print(cursor.rowcount, "record(s) updated.")
```

VIEW FEE RECORD

```
# Define the function to view Fee details
def view_fee():
    cursor = mydb.cursor()
    cursor.execute("SELECT * FROM fee")
    result = cursor.fetchall()
    for row in result:
        print(row)
```

DELETE FEE RECORD

```
# Define the function to delete Fee details
def delete_fee():
    student_id = input("Enter student Id: ")
    cursor = mydb.cursor()
    sqle = "DELETE FROM fee WHERE student_id = %s"
    vale = (student_id,)
    cursor.execute(sqle, vale)  mydb.commit() print(cursor.rowcount, "record(s) deleted.")
```

TESTING

Software testing is an empirical investigation conducted to provide skateholders with information

About the quality of the product or service under test, with respect to the context in which it is

Intended to operate. Software testing also provides an, independent view of the software to allow

The business to appreciate and understand the risk at implementation of the software.

Test techniques include, but are not limited to the process of executing a programme or

Application with the intent of finding software bugbugs.

It can also be stated as the process of validating and verifying that a software programme/

Application / product meets the business and technical requirements that guided the

its design and development, so that it works a expected and can be implemented with

the same characteristics. Software testing, depending on the testing method employed,

can be implemented at anytime in the development process however the most test

effort is employed after the requirements have been defined and coding process has

been completed.

BIBLOGRAPHY

- **Google for Research**
- **www.wikipedia.com**
- **www.geeksforgeeks.org**