COMPUTER SCIENCE PROJECT

Python Connectivity with MYSQL

Made by :

Persis Anil(16)

Shubhi Sudan (09)

Manas Goyal (40)

CERTIFICATE

This is to certify that this project has been completed by Persis Anil, Manas Goyal and Shubhi Sudan of class XII-F (2021-2022) under the guidance of Mrs. Puja Gupta (PGT: Computer Science, Mount Carmel School, Dwarka).

Teacher's Signature

ACKNOWLEDGEMENT

We would like to thank our computer science teacher Mrs. Puja Gupta for providing her invaluable guidance and knowledge.

We would also like to thank our parents who helped us in finalizing this project in the limited time frame.

We also extend our gratitude to the authors/creators of the various sources we used to obtain relevant information for this project.

INDEX

PROJECT SYNOPSIS

Topic: Python connectivity with MYSQL

Name: Library Management System

About: A library management system keeps track of the books present in the library. It is an important piece of software which is a must at schools and colleges.

Problem Definition: A library database system is an infrastructure that allows users to search books and book content, add/remove, and download selected books. The problem faced is that library users require an efficient method to find a specific book or keyword(s) within a book given a continuously expanding library.

Reason for selecting dataset: A librarian requires maintaining a database of new books and the books that are borrowed by users along with their due dates.

The purpose of a library management system is to provide instant and accurate data regarding any type of book, thereby saving a lot of time and effort.

Source of dataset: Created by team members

Hardware requirement: Laptop with operating system of windows 7 or above

Software requirement: Python

Front end: Python- Python is interpreted, interactive and object-oriented scripting language with less syntactical constructions. It is portable and supports GUI programming with cross platform library.

Back end: MySQL- MySQL is a relational database management system based on SQL — Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications.

Limitations: Network connectivity, power fluctuations, and inadequacy of staff with ICT skills.

Reference: Text book - Class XI and XII NCERT, Google.com (For any online queries), Book details (name and type of various books)

Team members: Shubhi Sudan (roll no. 09), Persis Anil (roll no. 16), Manas Goyal (roll no. 40)

Team details:

FANTASY BOOKS: analysis done by Shubhi Sudan

ADVENTURE BOOKS: analysis done by Persis Anil

CHILDREN'S BOOKS: analysis done by Manas Goyal

PYTHON CONNECTIVITY WITH MYSQL

Python can be used in database applications.One of the most popular databases is MySQL.
MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python

Advantages and benefits of MySQL Connector Python: –

- MySQL Connector Python is written in pure Python, and it is self-sufficient to execute database queries through Python.
- It is an official Oracle-supported driver to work with MySQL and Python.
- It is Python 3 compatible, actively maintained.

Arguments required to connect:-

You need to know the following details of the MySQL server to perform the connection from Python.

ARGUMENT	DESCRIPTION
Username	**The username that you use to work with MYSQL Server. The default username for the MYSQL database is a root.**
Password	**Password is given by the user at the time of installing the MYSQL server. If you are using root then you won't need the password.**
Host Name	**The server name or IP address on which MYSQL is running. If you are running on localhost, then you can use localhost or its IP.**
Database Name	**The name of the database to which you want to connect and perform the operations.**

Creating a database and Creating a table

```
import mysql.connector as mc
DB_NAME='library'
TB_NAME='data'
def create_db():
    global DB_NAME
    try:
        con=mc.connect(user="root",password="2112",host="localhost")
        if con.is_connected():
            cur=con.cursor() # cur user defined name
            DB_NAME=input("enter database name")
            cur.execute("CREATE DATABASE if NOT Exists {}".format(DB_NAME))
        else:
            print('sorry cannot connect')
    except: print("some error")
    finally: con.close()
def create_tb():
    global TB_NAME
    try:
        con=mc.connect(user="root",password="2112",host="localhost",database='library')
        if con.is_connected():
            cur=con.cursor() # cur user defined name
            cur.execute("use {}".format(DB_NAME))
            TB_NAME=input("enter table name")
            cur.execute('''CREATE TABLE if NOT Exists {}(serial_no int Primary key AUTO_INCREMENT,
            Title varchar(35) not null, Author varchar(35) not null,Status varchar(35) not null,
            Genre varchar(35) not null)'''
                        .format(TB_NAME))
        else:
            print('sorry cannot connect')
    except: print("some error")
    finally: con.close()
create_db()
create_tb()
```

```
enter database namelibrary
enter table namedata
```

Output:

```
mysql> use library;
Database changed
mysql> show tables;
+-------------------+
| Tables_in_library |
+-------------------+
| data              |
+-------------------+
1 row in set (1.40 sec)

mysql> desc data;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| serial_no | int         | NO   | PRI | NULL    | auto_increment |
| Title     | varchar(35) | NO   |     | NULL    |                |
| Author    | varchar(35) | NO   |     | NULL    |                |
| Status    | varchar(35) | NO   |     | NULL    |                |
| Genre     | varchar(35) | NO   |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
5 rows in set (1.21 sec)
```

Inserting record(s)

```
def insert_rec():
    while (True):
        try:
            con=mc.connect(user="root",password="2112",host="localhost",database='library')
            if con.is_connected():
                cur=con.cursor() # cur user defined name
                title=input("enter Title")
                author=input("enter The name of author")
                status=input("enter status")
                genre=input("Enter Genre")
                cur.execute("use {}".format(DB_NAME))
                sql="INSERT INTO data(Title,Author,Status,Genre) VALUES(%s,%s,%s,%s)"
                cur.execute(sql,(title,author,status,genre))
                con.commit()
            else:
                print('sorry cannot connect')
        except Exception as err: print("some error"+err)
        con.close()
        m=input("do you want to enter more records?")
        if (m=='no'): break
insert_rec()
```

```
enter TitleGame of Thrones
enter The name of authorGeorge R.R. Martin
enter statusIssued
Enter GenreFantasy
do you want to enter more records?yes
enter TitleTreasure Island
enter The name of authorRobert Louis Stevenson
enter statusAvailable
Enter GenreAdventure fiction
do you want to enter more records?yes
enter TitleCharlie and the Chocolate Factory
enter The name of authorRoald Dahl
enter statusIssued
Enter GenreChildren's books
do you want to enter more records?no
```

Output:

```
mysql> select * from data;
+-----------+-----------------------------------+-----------------------+-----------+-------------------+
| serial_no | Title                             | Author                | Status    | Genre             |
+-----------+-----------------------------------+-----------------------+-----------+-------------------+
|         1 | Game of Thrones                   | George R.R. Martin    | Issued    | Fantasy           |
|         2 | Treasure Island                   | Robert Louis Stevenson | Available | Adventure fiction |
|         3 | Charlie and the Chocolate Factory | Roald Dahl            | Issued    | Children's books  |
+-----------+-----------------------------------+-----------------------+-----------+-------------------+
3 rows in set (0.62 sec)
```

Deleting record(s)

```
def delete_rec():
    try:
        con=mc.connect(user="root",password="2112",host="localhost",database='library')
        if con.is_connected():
            cur=con.cursor()
            title=input("enter name of book")
            sql="delete FROM data WHERE Title = %s"
            cur.execute(sql,(title,))
            con.commit()
        else:
            print('sorry cannot connect')
    except Exception as err: print("some error"+str(err))
    finally: con.close()
delete_rec()
```

```
enter name of bookTreasure Island
```

Output:

```
mysql> select * from data;
+-----------+-----------------------------------+--------------------+--------+------------------+
| serial_no | Title                             | Author             | Status | Genre            |
+-----------+-----------------------------------+--------------------+--------+------------------+
|         1 | Game of Thrones                   | George R.R. Martin | Issued | Fantasy          |
|         3 | Charlie and the Chocolate Factory | Roald Dahl         | Issued | Children's books |
+-----------+-----------------------------------+--------------------+--------+------------------+
2 rows in set (0.07 sec)
```

Displaying all records

```
def display_all_rec():
    try:
        con=mc.connect(user="root",password="2112",host="localhost",database='library')
        if con.is_connected():
            cur=con.cursor()
            cur.execute("use {}".format(DB_NAME))
            sql="select * FROM data"
            cur.execute(sql)
            while True:
                i=cur.fetchone()
                if not i:
                    break
                print("Serial number: {}".format(i[0]))
                print("Title: {}".format(i[1]))
                print("Author: {}".format(i[2]))
                print("Status: {}".format(i[3]))
                print("Genre : {}".format(i[4]))
                print()
        else:
            print('sorry cannot connect')
    except Exception as err: print("some error"+str(err))
    finally: con.close()
display_all_rec()
```

```
Serial number: 1
Title: Game of Thrones
Author: George R.R. Martin
Status: Issued
Genre : Fantasy

Serial number: 3
Title: Charlie and the Chocolate Factory
Author: Roald Dahl
Status: Issued
Genre : Children's books
```

Searching record(s)

```
def search_title_rec():
    try:
        con=mc.connect(user="root",password="2112",host="localhost",database='library')
        if con.is_connected():
            cur=con.cursor()
            title=input("enter name of the book")
            cur.execute("use {}".format(DB_NAME))
            sql="select * FROM data where title = %s"
            cur.execute(sql,(title,))
            while True:
                i=cur.fetchone()
                if not i:
                    if cur.rowcount==0:
                        print("sorry no such book found")
                    break
                print("Serial number: {}".format(i[0]))
                print("Title: {}".format(i[1]))
                print("Author: {}".format(i[2]))
                print("Status: {}".format(i[3]))
                print("Genre : {}".format(i[4]))
                print()
        else:
            print('sorry cannot connect')
    except Exception as err: print("some error"+str(err))
    finally: con.close()
search_title_rec()
```

```
enter name of the bookGame of Thrones
Serial number: 1
Title: Game of Thrones
Author: George R.R. Martin
Status: Issued
Genre : Fantasy
```

```
def search_author_rec():
    try:
        con=mc.connect(user="root",password="2112",host="localhost",database='library')
        if con.is_connected():
            cur=con.cursor()
            cur.execute("use {}".format(DB_NAME))
            author=input("enter the name of author")
            sql="select * FROM data where author = %s"
            cur.execute(sql,(author,))
            while True:
                i=cur.fetchone()
                if not i:
                    if cur.rowcount==0:
                        print("sorry no book found by this author")
                    break
                print("Serial number: {}".format(i[0]))
                print("Title: {}".format(i[1]))
                print("Author: {}".format(i[2]))
                print("Status: {}".format(i[3]))
                print("Genre : {}".format(i[4]))
                print()

        else:
            print('sorry cannot connect')
    except Exception as err: print("some error"+str(err))
    finally: con.close()
search_author_rec()
```

```
enter the name of authorRoald Dahl
Serial number: 3
Title: Charlie and the Chocolate Factory
Author: Roald Dahl
Status: Issued
Genre : Children's books
```