

TABLE OF CONTENTS

<u>No</u>	<u>S.No</u>	<u>Topic Name</u>	<u>Page</u>
	1.	Abstract	4
	2.	System requirements	5
	3.	Database design	6
	4.	Coding	7-18
	5.	Output screens	19-23
	6.	Bibliography	24

ABSTRACT

This project focuses on the development of a Library Database Management System using Python and MySQL to streamline library operations such as book inventory, user management, and transaction records. The system employs a MySQL database to store and manage information on books, members, and borrowing transactions. Python is utilized as the programming language to implement functionalities, leveraging libraries like mysql-connector for database connectivity.

The application provides features such as adding, updating, deleting, and searching for books and members, as well as recording book borrow and return transactions. User input is validated to ensure data integrity, while SQL queries are optimized for efficient database operations. The system also includes safeguards against common database vulnerabilities, such as SQL injection, to enhance security.

This project demonstrates how relational databases and Python programming can be integrated to develop a functional, user-friendly system for managing library resources. The system can be extended to include advanced features like graphical user interfaces, automated notifications, and real-time analytics, making it a scalable solution for modern libraries.

SYSTEM REQUIREMENTS

Hardware Components

1. VGA Monitor
2. Qwerty keyboard
3. 2GB RAM
4. 2.6 GHz Processor
5. Graphics card

Software Components

1. Windows 7
2. Python 3.7 with suitable modules
3. MySQL Command Client

DATABASE DESIGN

In the following "Library" table, we will store the details of all books in the library .

Field	Type	Null	Key	Default
entry_id	int(11)	NO	PRI	NULL
book_title	varchar(255)	NO		NULL
author	varchar(255)	NO		NULL
price	decimal(10,2)	NO		NULL
status	varchar(50)	NO		NULL

CODING

```
import pymysql

import matplotlib.pyplot as plt

# Database connection

connection = pymysql.connect(

    host="localhost",

    user="root", # Replace with your MySQL username

    password="root", # Replace with your MySQL

password

    database="library_db"

)

cursor = connection.cursor()

while True:

    print("\nLibrary Management System")

    print("1. Insert New Data")

    print("2. Update Data")
```

```
print("3. Delete Data")
print("4. Search Specific Details")
print("5. Display All Details")
print("6. Graph (Books by Price)")
print("7. Change Book Status")
print("8. Exit")
```

```
choice = input("Enter your choice: ")
if not choice.isdigit() or not (1 <= int(choice) <= 8):
    print("Invalid choice! Please enter a number
between 1 and 8.")
    continue
```

```
choice = int(choice)
```

```
#Insert New Data
```

```
if choice == 1:
    print("\n--- Insert New Data ---")
    book_id = input("Enter Book ID (integer): ")
    if not book_id.isdigit():
```

```
        print("Invalid Book ID. Please enter a valid  
integer ID.")
```

```
        continue
```

```
    book_id = int(book_id)
```

```
    title = input("Enter Book Title: ")
```

```
    author = input("Enter Author: ")
```

```
    price = input("Enter Book Price: ")
```

```
    if not price.replace(".", "", 1).isdigit():
```

```
        print("Invalid price. Please enter a valid  
numeric value.")
```

```
        continue
```

```
    price = float(price)
```

```
    cursor.execute("SELECT * FROM library WHERE  
entry_id = %s", (book_id,))
```

```
    existing_book = cursor.fetchone()
```

```
    if existing_book:
```

```
        print("Book with ID", book_id, "already  
exists. Please choose a different ID.")
```

else:

```
        query = "INSERT INTO library (entry_id,  
book_title, author, price, status) VALUES (%s, %s, %s,  
%s, 'Available')"
```

```
        cursor.execute(query, (book_id, title, author,  
price))
```

```
        connection.commit()
```

```
        print("Book added successfully!")
```

Update Data

elif choice == 2:

```
    print("\n--- Update Data ---")
```

```
    book_id = input("Enter Book ID to Update: ")
```

```
    if not book_id.isdigit():
```

```
        print("Invalid Book ID. Please enter a valid  
ID.")
```

```
        continue
```

```
    book_id = int(book_id)
```

```
    cursor.execute("SELECT * FROM library WHERE  
entry_id = %s", (book_id,))
```



```
result = cursor.fetchone()

if result:

    new_title = input("Enter new Book Title (press
Enter to skip): ") or result[1]

    new_author = input("Enter new Author (press
Enter to skip): ") or result[2]

    new_price = input("Enter new Price (press
Enter to skip): ")

    if new_price:

        if not new_price.replace(".", "",
1).isdigit():

            print("Invalid price. Please enter a
valid numeric value.")

            continue

        new_price = float(new_price)

    else:

        new_price = result[3]

query = """UPDATE library

        SET book_title = %s, author = %s,
price = %s
```

WHERE entry_id = %s''''''

**cursor.execute(query, (new_title, new_author,
new_price, book_id))**

connection.commit()

**print("Book with ID", book_id, "updated
successfully!")**

else:

**print("Book ID does not exist. Update
operation skipped.")**

Delete Data

elif choice == 3:

print("\n--- Delete Data ---")

book_id = input("Enter Book ID to Delete: ")

if not book_id.isdigit():

**print("Invalid Book ID. Please enter a valid
ID.")**

continue

book_id = int(book_id)

```
        cursor.execute("SELECT * FROM library WHERE
entry_id = %s", (book_id,))

        result = cursor.fetchone()

        if result:

            query = "DELETE FROM library WHERE
entry_id = %s"

            cursor.execute(query, (book_id,))

            connection.commit()

            print("Book with ID", book_id, "deleted
successfully!")

        else:

            print("Book ID does not exist. Delete
operation skipped.")
```

Option 4: Search Specific Details

```
elif choice == 4:

    print("\n--- Search Specific Details ---")

    search_id = input("Enter Book ID to Search: ")

    if not search_id.isdigit():

        print("Invalid Book ID. Please enter a valid
ID.")
```

continue

search_id = int(search_id)

**cursor.execute("SELECT * FROM library WHERE
entry_id = %s", (search_id,))**

result = cursor.fetchone()

if result:

print("Book Details:")

print("ID:", result[0])

print("Title:", result[1])

print("Author:", result[2])

print("Price:", result[3])

print("Status:", result[4])

else:

print("Book with ID", search_id, "not found.")

Display All Details

elif choice == 5:

print("\n--- Display All Details ---")

cursor.execute("SELECT * FROM library")

```

records = cursor.fetchall()

if records:

    print("ID | Title | Author
| Price | Status")

    for row in records:

        title = row[1] or "N/A"
        author = row[2] or "N/A"
        price = round(row[3], 2)

        print(row[0], "|", title.ljust(20), "|",
author.ljust(15), "|", price, "|", row[4])

    else:

        print("No books found in the library.")

# Graph(Books by Price)

elif choice == 6:

    print("\n--- Graph (Books by Price) ---")

    cursor.execute("SELECT book_title, price FROM
library ORDER BY price")

    result = cursor.fetchall()

    if result:

```

```
titles = [row[0] for row in result]
```

```
prices = [row[1] for row in result]
```

```
plt.barh(titles, prices, color='skyblue')
```

```
plt.title("Books Sorted by Price")
```

```
plt.xlabel("Price")
```

```
plt.ylabel("Book Title")
```

```
plt.show()
```

```
else:
```

```
    print("No books found to display in the  
graph.")
```

```
# Change Book Status
```

```
elif choice == 7:
```

```
    print("\n--- Change Book Status ---")
```

```
    book_id = input("Enter Book ID to Change Status:  
")
```

```
    if not book_id.isdigit():
```

```
        print("Invalid Book ID. Please enter a valid  
ID.")
```

```
        continue

    book_id = int(book_id)

    cursor.execute("SELECT * FROM library WHERE
entry_id = %s", (book_id,))

    result = cursor.fetchone()

    if result:

        print("Current Status:", result[4])

        new_status = input("Enter new status (e.g.,
'Available', 'Issued'): ")

        query = "UPDATE library SET status = %s
WHERE entry_id = %s"

        cursor.execute(query, (new_status, book_id))

        connection.commit()

        print("Status of Book ID", book_id, "updated
to", new_status)

    else:

        print("Book ID does not exist. Status update
skipped.")

# Exit
```

```
elif choice == 8:
```

```
    print("Exiting the Library Management System.")
```

```
    break
```

```
# Close the connection
```

```
cursor.close()
```

```
connection.close()
```


OUTPUT SCREEN

Screen-1: WELCOME SCREEN

```
Library Management System
1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit
Enter your choice: |
```

Screen-2: Inserting data

```
Library Management System
1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit
Enter your choice: 1

--- Insert New Data ---
Enter Book ID (integer): 100
Enter Book Title: Harry Potter
Enter Author: Jk Rowling
Enter Book Price: 1500
Book added successfully!
```

Screen-3: updating the details(successful change)

Library Management System

1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit

Enter your choice: 2

--- Update Data ---

Enter Book ID to Update: 101

Enter new Book Title (press Enter to skip):

Enter new Author (press Enter to skip):

Enter new Price (press Enter to skip): 1900

Book with ID 101 updated successfully!

Screen-4: updating the details(unsuccesful change)

Library Management System

1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit

Enter your choice: 2

--- Update Data ---

Enter Book ID to Update: 104

Book ID does not exist. Update operation skipped.

Screen-5: deleting the details(successful delete)

```
Library Management System
1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit
Enter your choice: 3

--- Delete Data ---
Enter Book ID to Delete: 103
Book with ID 103 deleted successfully!
```

Screen-6: deleting the details(unsuccesful delete)

```
Library Management System
1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit
Enter your choice: 3

--- Delete Data ---
Enter Book ID to Delete: 105
Book ID does not exist. Delete operation skipped.
```

Screen-7:Searching specific Book details(successful Search)

```
Library Management System
1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit
Enter your choice: 4

--- Search Specific Details ---
Enter Book ID to Search: 100
Book Details:
ID: 100
Title: Harry Potter
Author: Jk Rowling
Price: 1500.00
Status: Available
```

Screen-8:Searching specific Book details(unsuccesful Search)

```
Library Management System
1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit
Enter your choice: 4

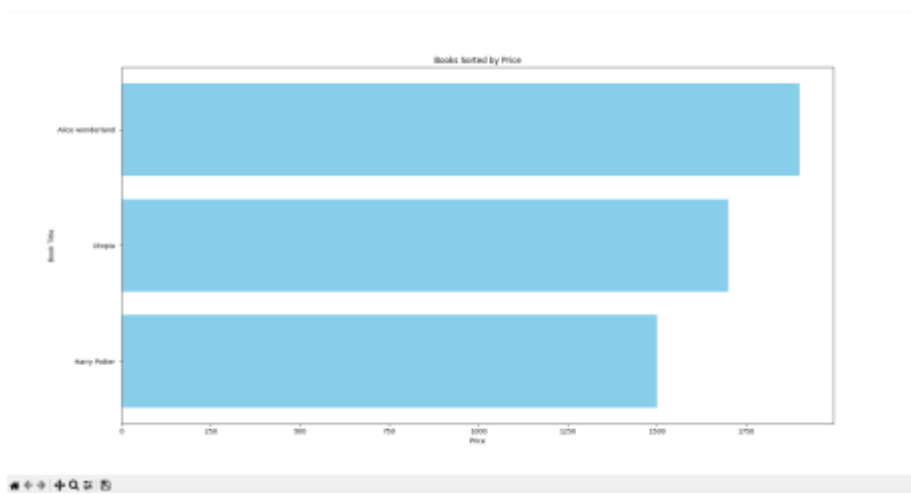
--- Search Specific Details ---
Enter Book ID to Search: 105
Book with ID 105 not found.
```

Screen-9: Display entire table data

```
Library Management System
1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit
Enter your choice: 5

--- Display All Details ---
ID | Title           | Author       | Price | Status
100 | Harry Potter    | Jk Rowling   | 1500.00 | Available
101 | Alice Wonderland | Lewis Carroll | 1900.00 | Available
102 | Utopia          | Thomas Moor  | 1700.00 | Available
```

Screen-10: Displaying bar graph



Screen 11 : Changing Book Status

```
Library Management System
1. Insert New Data
2. Update Data
3. Delete Data
4. Search Specific Details
5. Display All Details
6. Graph (Books by Price)
7. Change Book Status
8. Exit
Enter your choice: 7

--- Change Book Status ---
Enter Book ID to Change Status: 100
Current Status: Available
Enter new status (e.g., 'Available', 'Issued'): Issued
Status of Book ID 100 updated to Issued
```

Bibliography:

www.google.com

www.python.org.

www.geeksforgeeks.org

www.stackoverflow.com

Martin Brown and Martin C Brown, *"Python: The Complete Reference"*, Mc-Graw-Hill, 2001