

# **Library Management System**

## **A PROJECT REPORT**

*Submitted by*

**Manik Chauhan [Reg No: RA2211003030245]**

**Arya Gupta[Reg No: RA2211003030247]**

**Khushi Joshi [Reg No: RA2211003030249]**

**Vishwas Upadhyay[Reg No: RA2211003030255]**

*Under the guidance of*

**Ms. Neetu Bansla**

(Assistant Professor, Department of Computer Science & Engineering)



**SRM INSTITUTE OF SCIENCE & TECHNOLOGY, NCR CAMPUS**

**APRIL 2024**

# **SRM INSTITUTE OF SCIENCE & TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## **BONAFIDE CERTIFICATE**

*Certified to be the bonafide record of work done by **Khushi Joshi(RA2211003030249)** of 4<sup>th</sup> semester 2<sup>nd</sup> year **B.TECH** degree course in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, NCR Campus** of Department of Computer Science & Engineering in Database Management Systems Lab, during the academic year 2023-2024.*

### **SIGNATURE**

Dr. Neetu Bansla  
Assistant Professor  
Computer Science & Engineering

### **SIGNATURE**

Dr. Avneesh Vashistha  
Head of the Department  
Computer Science & Engineering

# **SRM INSTITUTE OF SCIENCE & TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## **BONAFIDE CERTIFICATE**

*Certified to be the bonafide record of work done by **Arya Gupta(RA2211003030247)** of 4<sup>th</sup> semester 2<sup>nd</sup> year **B.TECH** degree course in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, NCR Campus** of Department of Computer Science & Engineering in Database Management Systems Lab, during the academic year 2023-2024.*

### **SIGNATURE**

Dr. Neetu Bansla  
Assistant Professor  
Computer Science & Engineering

### **SIGNATURE**

Dr. Avneesh Vashistha  
Head of the Department  
Computer Science & Engineering

# **SRM INSTITUTE OF SCIENCE & TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## **BONAFIDE CERTIFICATE**

*Certified to be the bonafide record of work done by **Manik Chauhan(RA2211003030245)** of 4<sup>th</sup> semester 2<sup>nd</sup> year **B.TECH** degree course in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, NCR Campus** of Department of Computer Science & Engineering in Database Management Systems Lab, during the academic year 2023-2024.*

### **SIGNATURE**

Dr. Neetu Bansla  
Assistant Professor  
Computer Science & Engineering

### **SIGNATURE**

Dr. Avneesh Vashistha  
Head of the Department  
Computer Science & Engineering

# **SRM INSTITUTE OF SCIENCE & TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## **BONAFIDE CERTIFICATE**

*Certified to be the bonafide record of work done by **Vishwas Upadhyay(RA2211003030255)** of 4<sup>th</sup> semester 2<sup>nd</sup> year **B.TECH** degree course in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, NCR Campus** of Department of Computer Science & Engineering in Database Management Systems Lab, during the academic year 2023-2024.*

### **SIGNATURE**

Dr. Neetu Bansla  
Assistant Professor  
Computer Science & Engineering

### **SIGNATURE**

Dr. Avneesh Vashistha  
Head of the Department  
Computer Science & Engineerin

## INDEX

<b>S. No.</b>	<b>Title</b>	<b>Page No.</b>
1	Abstract	6
2	Problem Identification	7
3	Introduction	8
4	ER Diagram	9
5	Conversion of ER to Relational Table	11
6	SQL Queries/Relational Algebra	13
7	Normalization	17
8	Result	19
9	Conclusion	22

# **LIBLIVE**

## **Library Management Online on Phone**

### **Abstract:**

The contemporary digital era has revolutionized the way libraries manage their resources and interact with patrons. Our project introduces an avant-garde Library Management System (LMS), meticulously crafted to leverage the power of MySQL and Python, and ingeniously integrated with a WhatsApp Twilio chatbot. This symbiosis of technologies transforms user experience by harnessing the ubiquity of WhatsApp for both administrative and user-related library functions.

The system uses secure phone number authentication to distinguish between admin and user roles, each endowed with a tailored set of commands accessible through an intuitive chat interface. Users enjoy the liberty to reserve books, which automatically updates the system's inventory, rendering the reserved books inaccessible to other patrons. This mechanism ensures an equitable distribution of library resources and maximizes book circulation efficiency.

Administrators, on the other hand, are empowered with comprehensive controls over the library's catalog. They can seamlessly add or remove books, manage user privileges, issue or recall books, and maintain order with the capability to ban and unban users as necessary. Such administrative tasks, traditionally cumbersome, are now executed with unprecedented ease and agility.

This LMS is not only a testament to technological innovation but also an embodiment of user-centric design. By facilitating real-time communication and transactions via WhatsApp, it capitalizes on a platform with vast global reach, effectively democratizing library services. This report encapsulates the strategic development, deployment, and operation of the LMS, emphasizing the system's robust database structure, secure authentication processes, and user-friendly interface—all of which coalesce to redefine library management for the 21st century. We delve into the architectural intricacies of the system, its Python-driven logic, and the MySQL database that acts as the bedrock of this advanced management tool.

In sum, this LMS stands as a beacon of modern library management, offering a case study in how digital tools can transform traditional services into an interactive, user-friendly, and efficient experience. Our exploration in the following pages underscores the pivotal role of well-integrated technology in elevating educational and literary institutions to meet the dynamic demands of today's digital clientele.

## **Problem Identification:**

Traditional library management systems have been hampered by several limitations that have impaired their ability to meet the evolving needs of users and administrators alike. First and foremost, conventional systems often rely on physical presence for tasks such as book reservations, issuance, and returns, which can be inconvenient and time-consuming for patrons. Additionally, these systems are typically bound to specific locations and operating hours, limiting access to library services.

Another significant challenge is the complexity and lack of intuitiveness in many legacy systems. Users and administrators frequently face steep learning curves and cumbersome processes when navigating through the various functionalities of the system. This complexity can deter engagement and reduce the overall efficiency of library operations.

Moreover, the issue of accessibility has become increasingly prominent. With the ubiquitous nature of smartphones and the global penetration of messaging platforms, users expect to access services on-the-go, without the constraints of traditional systems. Libraries have been slow to adapt to this digital shift, resulting in a gap between user expectations and the services provided.

From the perspective of library administrators, the inability to swiftly update and manage inventory, as well as enforce compliance through user restrictions, has often led to operational inefficiencies. The lack of real-time communication tools within the LMS has also presented challenges in promptly addressing user queries and ensuring timely returns of library materials.

These problems highlight the need for a modernized LMS that is accessible, user-friendly, and efficient in its operation. Such a system should be capable of integrating with widely used communication platforms to provide real-time interactions and transactions, thus aligning with contemporary expectations of service delivery. Our project identifies these gaps and seeks to bridge them with a novel solution that leverages the technological advances of our time.



## **Introduction:**

The essence of a library lies in its ability to disseminate knowledge effectively. However, the digital tide has altered the landscape of information exchange, rendering traditional library management systems inadequate. In response to this challenge, we embarked on a project to design an innovative Library Management System (LMS) that integrates the relational database capabilities of MySQL with the powerful scripting of Python, interfacing with users through a WhatsApp Twilio chatbot.

Our introduction delves into the crux of the project, addressing the need for a transformation in how libraries manage their resources and engage with their patrons. We examine the shifting paradigms in library utilization, with a growing demand for remote access, digital reservations, and instant communication. The project was initiated to fulfill these contemporary requirements, offering a solution that encapsulates the ease of messaging with the robustness of a traditional LMS.

The proposed system operates on two levels: it serves the users who seek convenience and immediacy in reserving and accessing books, and it serves the administrators who require an agile and reliable platform to manage library operations. By utilizing phone number authentication, the system ensures a secure and personalized experience for each user, while also providing administrators with a suite of powerful tools to manage the library's inventory, oversee user activity, and maintain the library's service standards.

This introduction sets the stage for the subsequent chapters, where we will explore the system's architecture, its user interface design, and the functionalities that make it an exceptional tool for modern libraries. We will also discuss the system's deployment, user adoption strategies, and the impact of this LMS on overall library management efficiency. The intention of this project is not merely to develop a new library system but to revolutionize the library experience for all stakeholders through the application of modern technology.

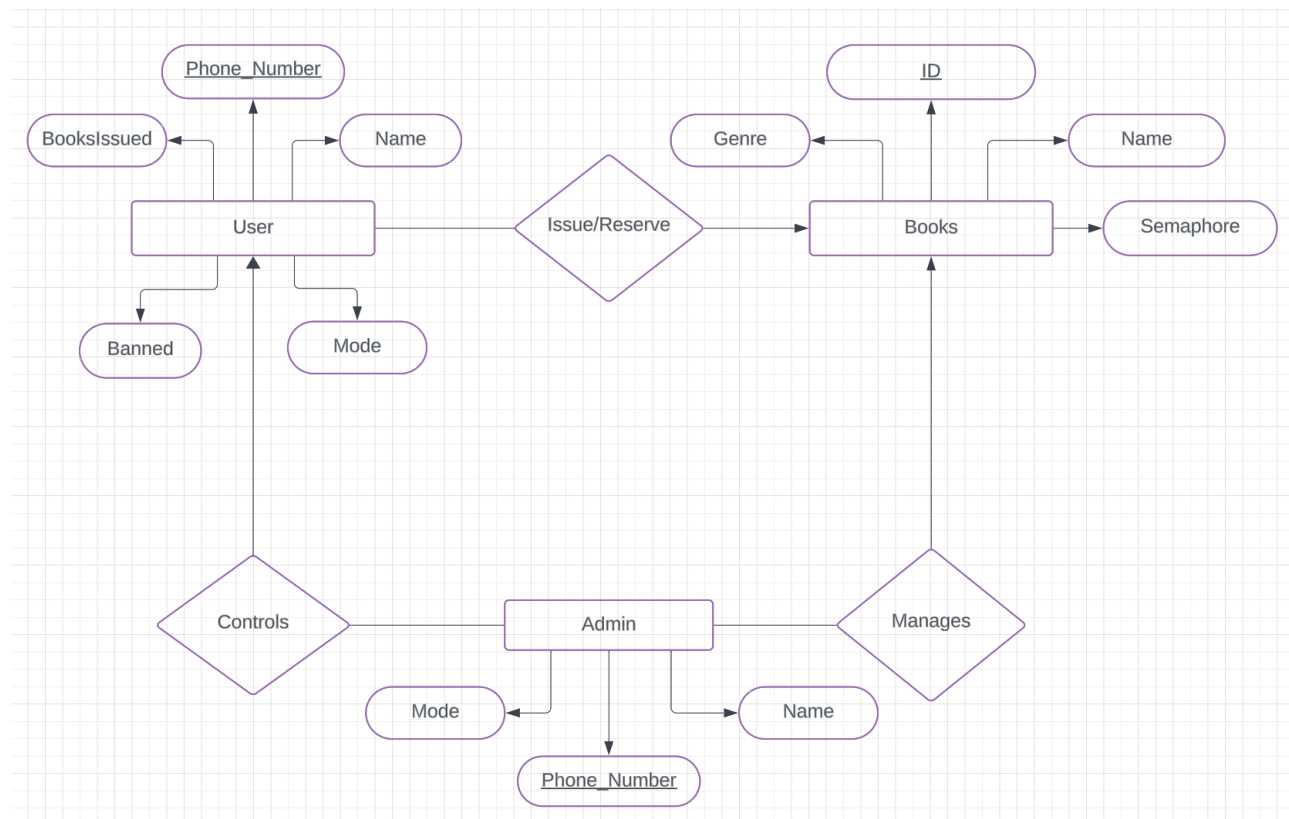
## ER Diagram:

### Introduction

Here we designed an Entity-Relationship (ER) diagram for a library management system. The goal was to establish a structured database model that could efficiently handle various aspects of library operations including user management, book cataloging, issue and reservation tracking, and administrative controls.

### Design Process

We began by identifying the primary entities that are integral to the library system: **User**, **Books**, **Admin**, and the relationship entities **Issue/Reserve**, **Controls**, and **Manages**. These entities were analyzed for their attributes and relationships with one another.



### Entities and Attributes

- **User:** This entity represents the library's patrons and includes attributes such as **Name**, **Phone\_Number**, **BooksIssued**, **Banned**, and **Mode**. These attributes allow the system to store personal details, contact information, and the current process status of the user in the library system.
- **Books:** This entity is critical for cataloging purposes and includes **ID**, **Name**, **Genre**, and **Semaphore**. These attributes help in identifying books, categorizing them by genre, and maintaining a semaphore for tracking book availability.

- **Admin:** The admin entity is designed to handle administrative tasks and has attributes like **Name**, **Phone\_Number**, and **Mode**. It facilitates contact with library staff and denotes their operational mode within the system.

### Relationships

- **Issue/Reserve:** This relationship entity links **User** and **Books** to depict the action of issuing or reserving books. It is a dynamic entity that reflects the current transactions between users and the library's inventory.
- **Controls:** This relationship entity illustrates the control that **Admin** has over **User** actions, influencing whether a user is banned or allowed certain access within the library system.
- **Manages:** Another relationship entity that connects **Admin** to **Books**, signifying the administrative tasks of managing the book inventory, such as adding new books, updating book status, and catalog maintenance.

The ER diagram serves as a blueprint for the development of the library management database. It ensures that the system is capable of handling complex interactions between the entities and provides a comprehensive overview of how data is interconnected within the library management context. The design emphasizes scalability, efficient data retrieval, and the ability to manage library operations effectively.

# Conversion of ER diagram to Relational Model

## Introduction

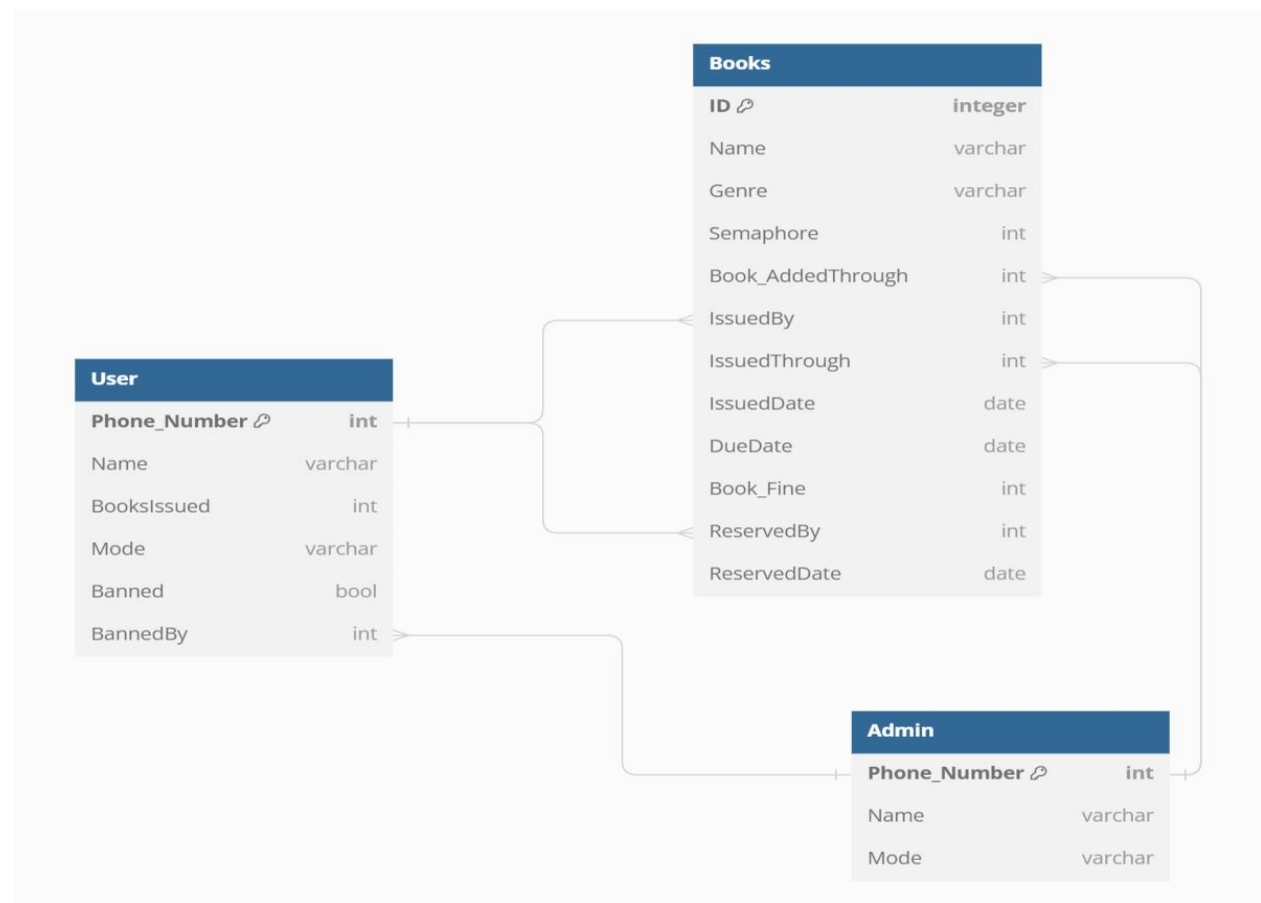
Following the initial creation of an ER diagram for a library management system, the subsequent phase involved the transformation of this conceptual schema into a practical relational database model. The study chronicles the process of converting the entities and relationships identified in the ER diagram into relational tables suitable for implementation in a database.

## Objective

The primary goal was to create a set of relational tables that accurately represent the data structure of the library system. This involved defining tables with appropriate fields, ensuring proper data types, and establishing relationships through foreign keys.

## Methodology

The process began with the deconstruction of the ER diagram into its constituent entities and associations. Each entity was then translated into a relational table, with entity attributes becoming table fields. Data types were carefully assigned to each field to ensure integrity and efficiency of data storage. The One to Many Relationship between entities were converted by adding the attributes of the relation to the 'Many' Relationship Entity with a Foreign key connecting to the 'One' Relationship Entity.



## Relational Tables and Fields

- **User Table:** This table holds the library patrons' information with fields such as **Phone\_Number**, **Name**, **BooksIssued**, **Mode**, **Banned**, and **BannedBy**. The **Phone\_Number** serves as the primary key.
- **Books Table:** Central to the library's inventory management, this table includes **ID**, **Name**, **Genre**, **Semaphore**, **Book\_AddedThrough**, **IssuedBy**, **IssuedThrough**, **IssuedDate**, **DueDate**, **Book\_Fine**, **ReservedBy**, and **ReservedDate**. The **ID** is the primary key, with foreign keys linking to the **User** table where applicable.
- **Admin Table:** The Admin table maintains administrator details with **Phone\_Number**, **Name**, and **Mode**. Here too, **Phone\_Number** is used as a primary key.

## Case Scenarios:

### Admin:

Phone_Number	Name	Mode
+919560405533	Harshit	IssueMenu
+919999367202	Vibhor	AdminMenu
NULL	NULL	NULL

### Books:

ID	Name	Genre	Semaphore	Book_AddedThrough	IssuedBy	IssuedThrough	IssuedDate	DueDate	Book_Fine	ReservedBy	ReservedDate
8	Spiderman	Action	1	+919999367202	NULL	NULL	NULL	NULL	0	NULL	NULL
9	Dragon Ball	Action	1	+919999367202	NULL	NULL	NULL	NULL	0	NULL	NULL
10	Demon Slayer	Action	1	+919999367202	+919667040342	+9199993672...	2024-04-10	2024-05-06	0	NULL	NULL
11	Steins Gate	Action	1	+919999367202	NULL	NULL	NULL	NULL	0	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0	NULL	NULL

### User:

Phone_Number	Name	BooksIssued	Mode	Banned	BannedBy
+917882345621	Rahul	NULL	UserMenu	NULL	NULL
+919667040342	Rohit	NULL	UserMenu	1	+919999367202
NULL	NULL	NULL	NULL	NULL	NULL

Sample data was inserted into the tables to simulate real-world operations:

- Users like Harshit and Vibhor with their respective modes of operation (**IssueMenu** and **AdminMenu**).
- A catalog of books such as "Spiderman", "Dragon Ball", and "Demon Slayer" with their statuses and transaction records.
- A user scenario where Rohit is banned by the admin Vibhor, demonstrating the system's capability to track user restrictions.

The study successfully demonstrates the transition from a theoretical ER diagram to a practical relational database model for a library management system. The resultant relational tables are poised to handle various library operations, from user management to book transactions, ensuring a seamless flow of data across the system. The integration of sample data helps validate the functionality of the database design, ensuring that it meets the system's requirements and user expectations.

# SQL Queries/Relational Algebra

## Introduction

In a library management system, efficiently managing the flow of books and user information is critical. This study focuses on the use of relational algebra to perform complex queries within a library's database to enhance the management of book allocations, user status, and administrative controls.

## Database Structure

The database consists of three main tables:

1. **Admin Table:** Contains data on the library administrators, including phone number, name, and mode.
2. **Books Table:** Records details of books including ID, name, genre, status, and the transaction history of issue and return.
3. **User Table:** Holds information about users, such as phone number, name, the books issued, operational mode, and ban status.

## Objective

The objective is to demonstrate the application of relational algebra in various scenarios, such as finding banned users, locating books currently issued, and understanding administrative operations through the use of projection, selection, and join operations.

## Relational Algebra Queries and Expected Outputs

### Query 1: Find the Names of Banned Users

Using the User table, we want to identify users who have been banned.

*Relational Algebra:*

```
 $\pi_{Name}(\sigma_{Banned=1}(User))$ 
```

*SQL:*

```
SELECT Name
FROM User
WHERE Banned = 1;
```

*Expected Output:*

Name
Rohit

### Query 2: List Books Currently Issued

We aim to find the names of the books that are currently issued out to users.

*Relational Algebra:*

```
 $\pi_{\text{Name}}(\sigma_{\text{IssuedBy} \neq \text{NULL}}(\text{Books}))$ 
```

*SQL:*

```
SELECT Name
FROM Books
WHERE IssuedBy IS NOT NULL;
```

*Expected Output:*

Name
Demon Slayer

### Query 3: Retrieve User Information Who Have Issued a Specific Book

For a book named "Demon Slayer," we want to find which user has it issued.

*Relational Algebra:*

```
 $(\pi_{\text{Phone\_Number}}, \text{Name}(\text{User})) \bowtie (\pi_{\text{IssuedBy}}(\sigma_{\text{Name} = \text{'Demon Slayer'}}(\text{Books})))$ 
```

*SQL:*

```
SELECT User.Phone_Number, User.Name
FROM User
JOIN Books ON User.Phone_Number = Books.IssuedBy
WHERE Books.Name = 'Demon Slayer';
```

*Expected Output:*

Phone_Number	Name
+919667040342	Rohit

### Query 4: List of Users Who Have Not Been Banned

We want to find all users who have not been banned.

*Relational Algebra:*

```
 $\pi_{\text{Name}}(\sigma_{\text{Banned} = \text{NULL}}(\text{User}))$ 
```

SQL:

```
SELECT Name
FROM User
WHERE Banned IS NULL;
```

Expected Output:

Name
Rahul

#### Query 5: Find All Action Genre Books

Retrieve the list of books that belong to the 'Action' genre.

Relational Algebra:

```
 $\pi_{\text{Name}}(\sigma_{\text{Genre}='Action'}(\text{Books}))$ 
```

SQL:

```
SELECT Name
FROM Books
WHERE Genre = 'Action';
```

Expected Output:

Name
Spiderman
Dragon Ball
Demon Slayer
Steins Gate

#### Query 6: Join Users with the Books They Have Issued

We want to find all the information about books and the corresponding users who have issued them. To do this, we need to join the **User** table with the **Books** table on the **Phone\_Number** from **User** and **IssuedBy** from **Books**.

Relational Algebra:

```
 $\pi_{\text{User.Name}, \text{Books.Name}}(\text{User} \bowtie_{\{\text{User.Phone\_Number} = \text{Books.IssuedBy}\}} \text{Books})$ 
```



*SQL:*

```
SELECT User.Name, Books.Name AS Book_Name
FROM User
INNER JOIN Books ON User.Phone_Number = Books.IssuedBy;
```

*Expected Output:*

Name	Book_Name
Rohit	Demon Slayer

These queries demonstrate the power of relational algebra in extracting and correlating data across multiple tables in a library management system. By leveraging these operations, library staff can efficiently handle administrative tasks and ensure a smooth operation.

## Normalization:

### Introduction

The process of normalization in database design involves organizing tables and their relationships in a way that reduces redundancy and dependency. It is a systematic approach that aims to minimize data anomalies and maintain data integrity by dividing large tables into smaller, more manageable pieces of related data.

### Original Unnormalized Table:

The original table structure contains multiple fields related to users, books, their issuance, reservation, and admin activities, all within a single table. This conglomeration of data leads to numerous issues such as redundancy, update anomalies, and a lack of scalability.

### First Normal Form (1NF):

To achieve 1NF, we ensure that the table has a primary key and that all attributes contain atomic (indivisible) values. We might begin by separating book-related information from user-related details, resulting in individual tables where each row is unique, and repeating groups (such as multiple genres for a single book) are eliminated. This would involve creating tables like BookGenres, which relate books to their genres.

### Second Normal Form (2NF):

For a table to be in 2NF, it must first satisfy 1NF and then move all fields that are not dependent on the primary key to separate tables. This would involve creating tables like Reserve and Issue, which stores information about which user has issued or reserved which book. The relationship between books and users would be based on foreign keys that reference the primary keys in the books and users tables.

### Normalization Outcome:

- **BookGenres Table:** Created to separate book genres from the main books table to avoid repeating the book name for each genre. Each book can have multiple genres, and each genre can include many books, establishing a many-to-many relationship between books and genres.

ID	Genre
16	Action
16	Superhero

- **Issue Table:** Extracted from the main books table to track the issuance of books independently. This table includes the user who has issued the book, the admin through whom the book was issued, and relevant dates (issued and due). It helps manage the circulation of books effectively without storing redundant user or book data in the books table.

Issue_ID	ID	IssuedBy	IssuedThrough	IssuedDate	DueDate	Book_Fine
6	16	+919871405533	+919999367202	2024-03-24	2024-04-09	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Reserve Table:** Separated to manage reservations of books by users. It holds information about which user has reserved which book and the reservation date, ensuring that a book's availability is dynamically managed without data duplication.

Reservation_ID	ID	ReservedBy	ReservedDate
1	15	+919560405533	2024-04-25
NULL	NULL	NULL	NULL

### New Books Table:

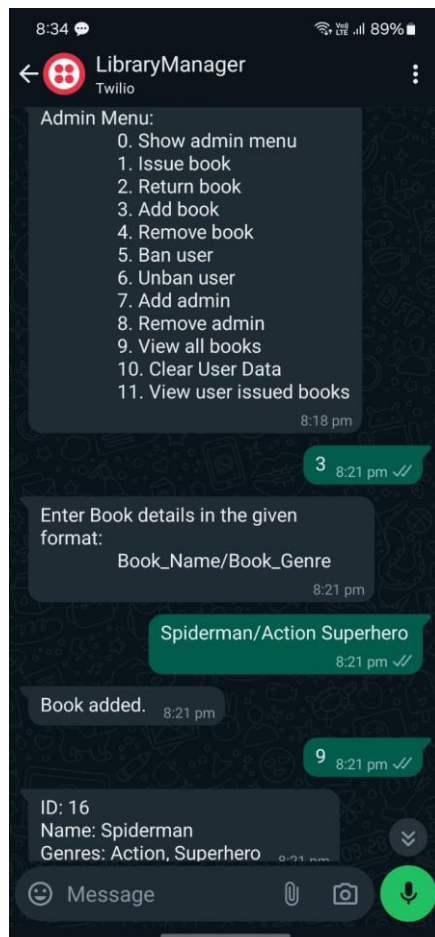
ID	Name	Semaphore	Book_AddedThrough
16	Spiderman	0	+919999367202
NULL	NULL	NULL	NULL

Through this normalization process, we ensure that each table serves a single purpose, representing an entity or a relationship in the library management domain. The result is a database design that is more maintainable, scalable, and efficient, with reduced redundancy and minimized potential for anomalies.

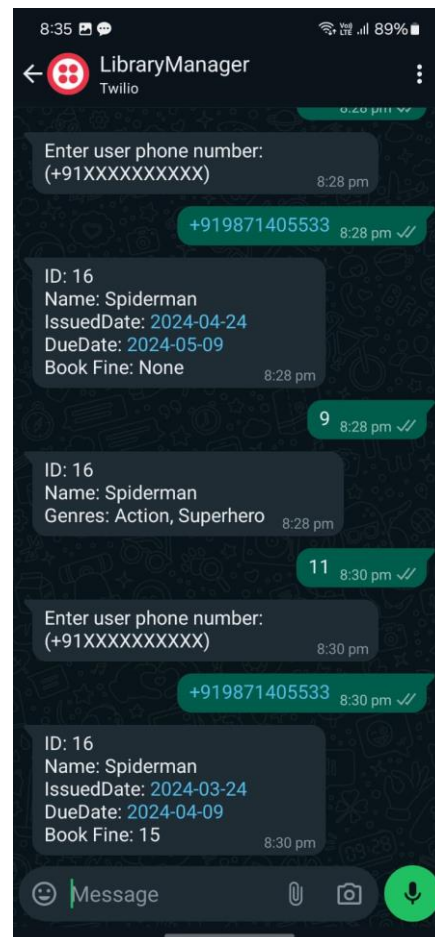
## Result

### Admin:

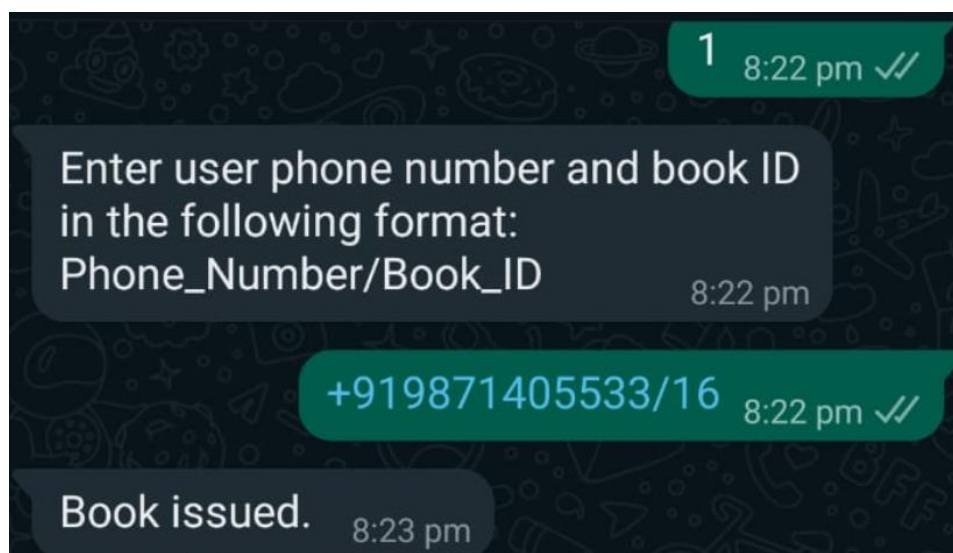
#### Adding book:



#### Viewing issued book of user:

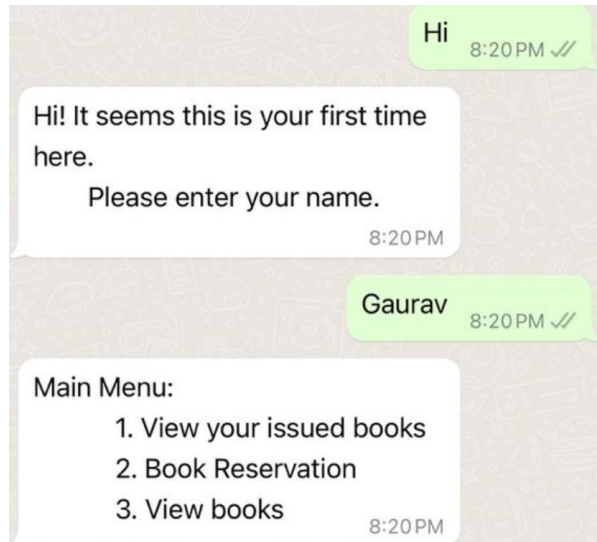


#### Issuing Book:

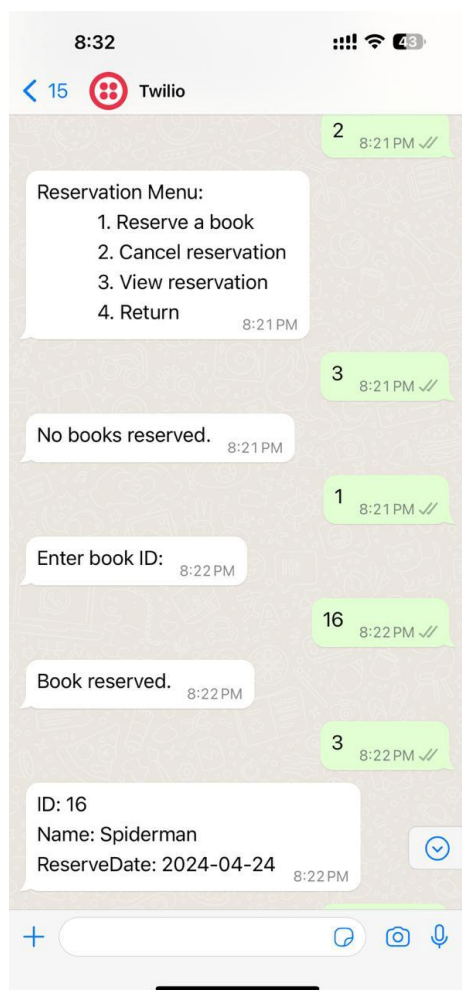


## User:

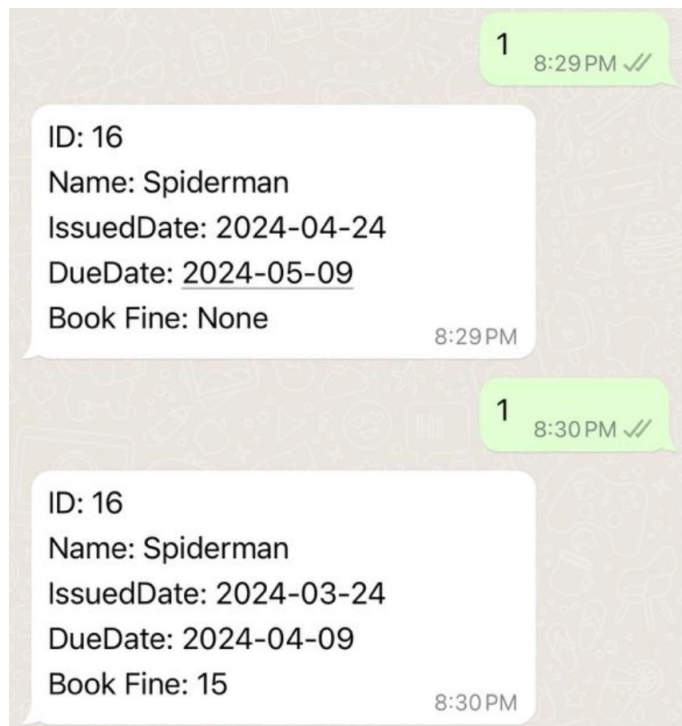
### First time User:



### Reserve Menu and Reserving/Viewing reserved books:



**Viewing issued book after admin issues:**



## Conclusion

The Library Management System (LMS) integrated with the WhatsApp Twilio chatbot, as delineated in this report, exemplifies the transformative potential of combining traditional library services with modern communication technology. By adopting a well-structured MySQL database and the versatility of Python scripting, the system presents a compelling case for how digital convergence can significantly enhance the accessibility and efficiency of library operations.

Throughout the project's development, a keen focus on user experience and operational effectiveness drove the design decisions. The normalization process was particularly crucial in refining the database schema, leading to the creation of well-defined tables such as BookGenres, Issue, and Reserve. These changes resolved numerous data redundancy issues and paved the way for a more robust and scalable system architecture.

As we conclude, it is evident that the project's objectives have been met with distinction. Users can now interact with the

library system through a familiar messaging interface, making book reservations and interactions with the library as simple as sending a chat message. Administrators are equipped with a powerful suite of tools to manage library resources efficiently, and the LMS itself has become more resilient against common data issues such as duplication and inconsistency.

The project's success lies not only in the technical execution but also in its adherence to the principles of user-centered design and the commitment to improving library management practices. This LMS stands as a beacon for future developments, showcasing the significant benefits of integrating digital platforms with library services.

In conclusion, this project report chronicles the journey from a concept to a fully functional Library Management System that leverages WhatsApp for its user interface. This innovative approach has the potential to be replicated in various other contexts, demonstrating that with the right blend of technology and vision, traditional services can be reimagined to meet and exceed contemporary user expectations. The success of this project opens the door to further innovation in library management, encouraging continuous improvement and the embracing of technological advances in service of knowledge dissemination and access.