PANDIT DEENDAYAL ENERGY UNIVERSITY

SCHOOL OF TECHNOLOGY

B.Tech. (Computer Science and Engineering) : Semester 4

Course: Database Management Systems - Lab

Course Code: 20CP208P

Project report on

Library Management System

Submitted To:

Dr. Sonam Nahar

Submitted By:

Yatri Patel : 21BCP002

Dhruvil Shah: 21BCP426D

Niti Chovatia: 21BCP032

Parth Mistry : 21BCP028

Library Management System

Introduction:

Welcome to the Library Management System project! This database management system project involves the creation of four tables: "Books", "Authors", "Publishers", and "Users". These tables are designed to efficiently manage the data related to books, authors, publishers, and users in a library setting.

The "**Books**" table includes columns such as BookID, Title, AuthorID, PublisherID, Price, NumCopies, and ImageURL. The BookID serves as a unique identifier for each book, while the Title represents the title of the book. The AuthorID and PublisherID columns are used as foreign keys to reference the corresponding tables for authors and publishers. The Price column represents the price of the book, and the NumCopies column indicates the number of copies available. The ImageURL column stores the URL of the book's image.

The "**Authors**" table contains columns like AuthorID and AuthorName. The AuthorID serves as a unique identifier for each author, while the AuthorName stores the name of the author.

The "**Publishers**" table includes columns like PublisherID and PublisherName. The PublisherID is used as a unique identifier for each publisher, and the PublisherName stores the name of the publisher.

The "**Users**" table includes columns like UID, UName, Email, Phone, and pswd. The UID serves as a unique identifier for each user, while the UName, Email, and Phone columns store the name, email, and phone number of the user, respectively. The pswd column stores the password of the user.

By utilizing these tables and their respective columns, this Library Management System aims to provide an efficient and organized way to manage books, authors, publishers, and users in a library setting.

Entities:

1. **Books:**

The "Books" table has the following columns:

- BookID (VARCHAR(10)): Represents the unique identifier for each book.
- Title (VARCHAR(255)): Represents the title of the book and is marked as NOT NULL, meaning it must have a value.
- AuthorID (VARCHAR(10)): Represents the unique identifier of the author of the book.
- PublisherID (INT): Represents the unique identifier of the publisher of the book and is marked as NOT NULL.
- Price (DECIMAL(10,2)): Represents the price of the book and is marked as NOT NULL.
- NumCopies (INT): Represents the number of copies available for the book and is marked as NOT NULL.
- ImageURL (VARCHAR(255)): Represents the URL of the book's image.

- PRIMARY KEY (BookID): Specifies that the BookID column is the primary key for the "Books" table.
- FOREIGN KEY (AuthorID): Specifies that the AuthorID column is a foreign key referencing the AuthorID column in the "Authors" table.
- FOREIGN KEY (PublisherID): Specifies that the PublisherID column is a foreign key referencing the PublisherID column in the "Publishers" table.

2. **Authors:**

The "Authors" table has the following columns:

- AuthorID (VARCHAR(10)): Represents the unique identifier for each author.
- AuthorName (VARCHAR(255)): Represents the name of the author and is marked as NOT NULL.
- PRIMARY KEY (AuthorID): Specifies that the AuthorID column is the primary key for the "Authors" table.

3. **Publishers:**

The "Publishers" table has the following columns:

- PublisherID (VARCHAR(10)): Represents the unique identifier for each publisher.
- PublisherName (VARCHAR(255)): Represents the name of the publisher and is marked as NOT NULL.
- PRIMARY KEY (PublisherID): Specifies that the PublisherID column is the primary key for the "Publishers" table.

4. **Users:**

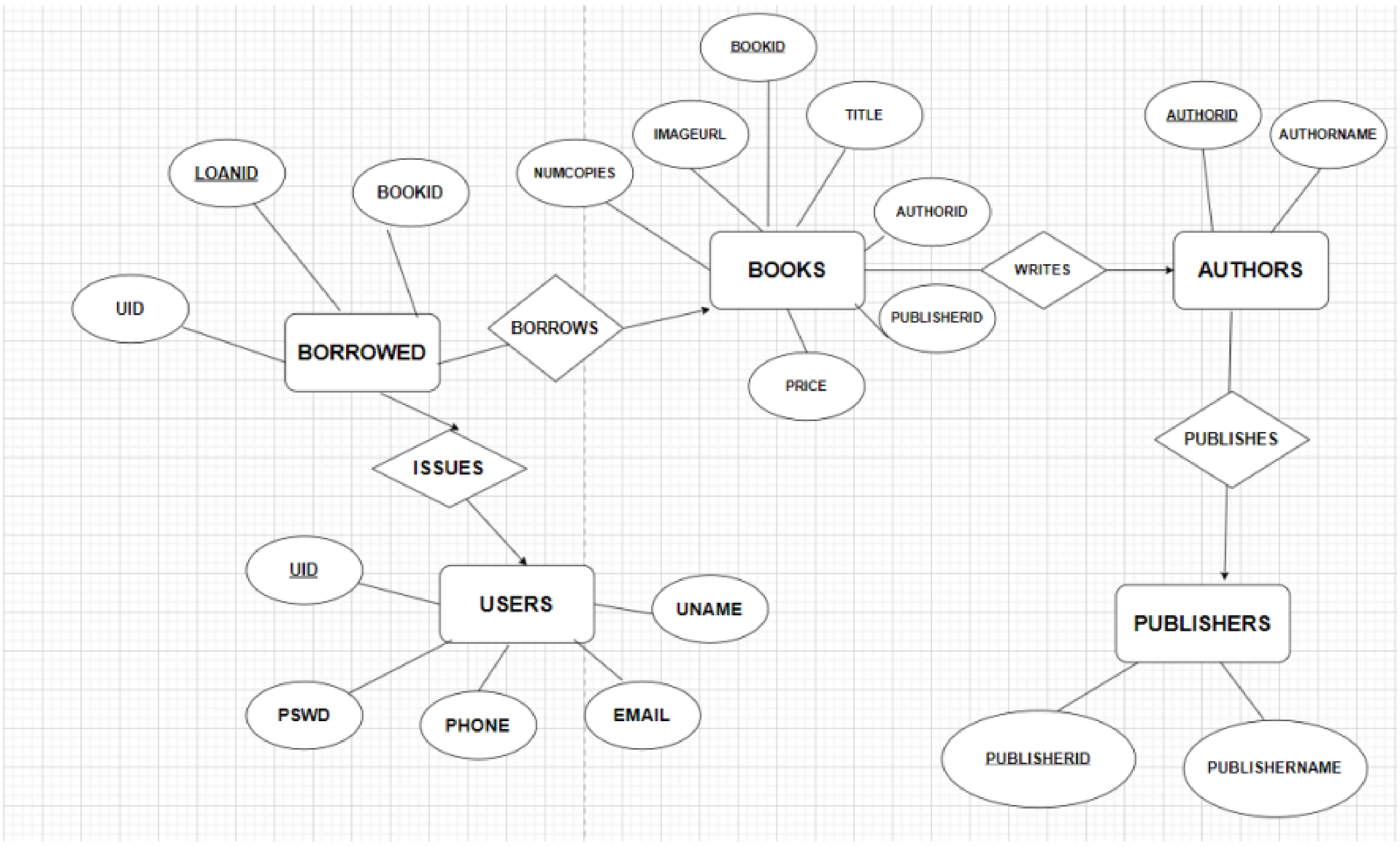
The "Users" table has the following columns:

- UID (VARCHAR(10)): Represents the unique identifier for each user.
- UName (VARCHAR(255)): Represents the name of the user and is marked as NOT NULL.
- Email (VARCHAR(255)): Represents the email of the user and is marked as NOT NULL.
- Phone (VARCHAR(255)): Represents the phone number of the user and is marked as NOT NULL.
- pswd (INT): Represents the password of the user and is marked as NOT NULL.
- PRIMARY KEY (UID): Specifies that the UID column is the primary key for the "Users" table.

Entity – Relationship Diagram:

An ER (Entity-Relationship) diagram is a graphical representation of entities and their relationships to each other. It is commonly used in software development to model the relationships between various entities within a system. An ER diagram can help developers to better understand the data structures of a system, and can serve as a blueprint for database design. Entities are represented as boxes, and relationships between entities are represented as lines connecting the boxes. ER diagrams typically include entities, attributes, and relationships, and can be useful in identifying potential problems or conflicts in the data model.

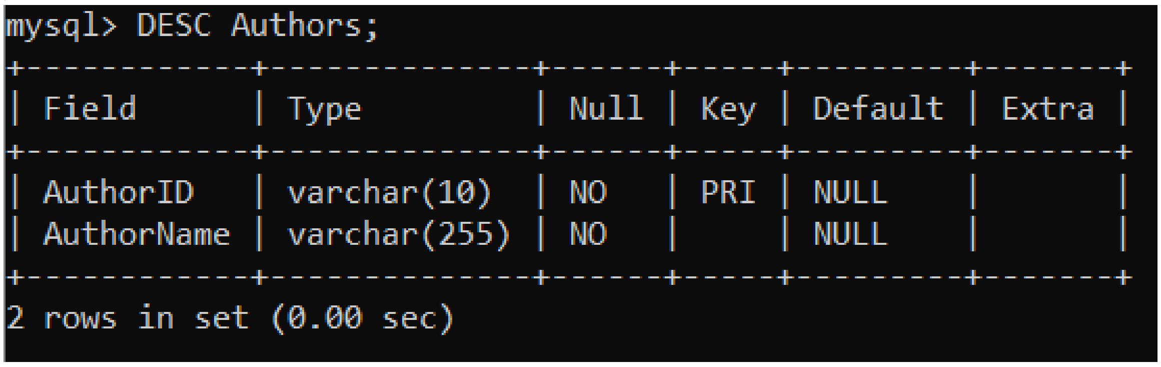
ER DIAGRAM FOR OUR SCHEMA :-



Tables:

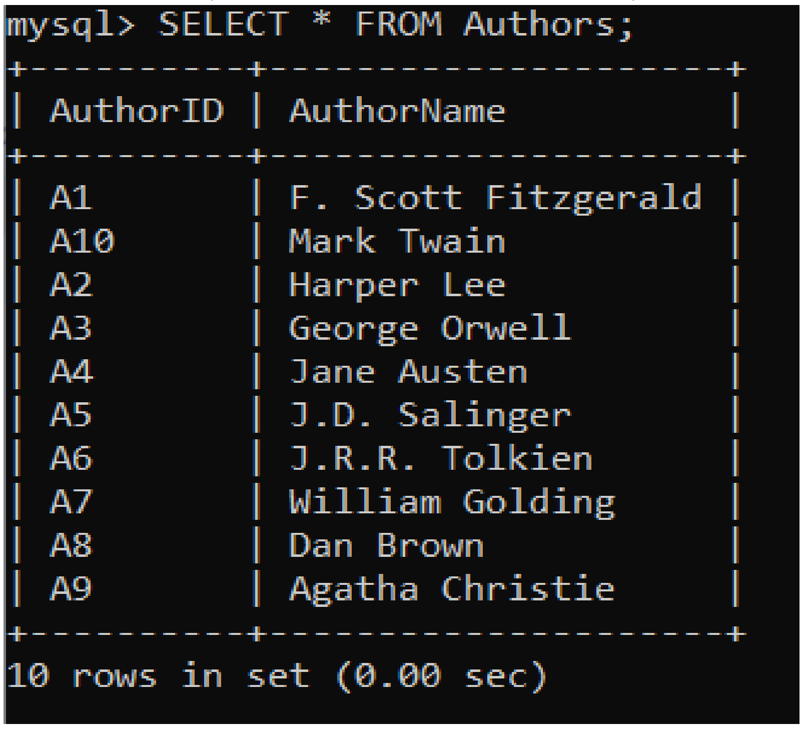
1. Authors:

```
CREATE TABLE Authors (
   AuthorID VARCHAR(10),
   AuthorName VARCHAR(255) NOT NULL,
   PRIMARY KEY (AuthorID)
);
```



```
mysql> DESC Authors;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| AuthorID   | varchar(10)  | NO   | PRI | NULL    |       |
| AuthorName | varchar(255) | NO   |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

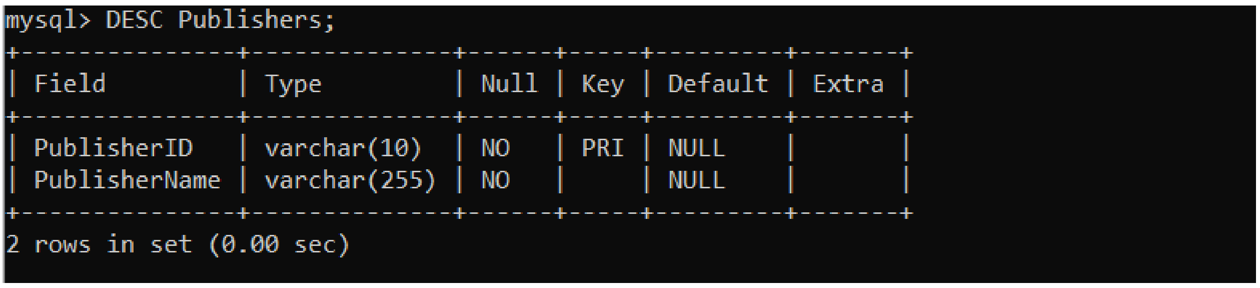
```
INSERT INTO Authors (AuthorID, AuthorName)
    VALUES
        ('A1', 'F. Scott Fitzgerald'),
        ('A2', 'Harper Lee'),
        ('A3', 'George Orwell'),
        ('A4', 'Jane Austen'),
        ('A5', 'J.D. Salinger'),
        ('A6', 'J.R.R. Tolkien'),
        ('A7', 'William Golding'),
        ('A8', 'Dan Brown'),
        ('A9', 'Agatha Christie'),
        ('A10', 'Mark Twain');
```



```
mysql> SELECT * FROM Authors;
+----------+---------------------+
| AuthorID | AuthorName          |
+----------+---------------------+
| A1       | F. Scott Fitzgerald |
| A10      | Mark Twain          |
| A2       | Harper Lee          |
| A3       | George Orwell       |
| A4       | Jane Austen         |
| A5       | J.D. Salinger       |
| A6       | J.R.R. Tolkien      |
| A7       | William Golding     |
| A8       | Dan Brown           |
| A9       | Agatha Christie     |
+----------+---------------------+
10 rows in set (0.00 sec)
```

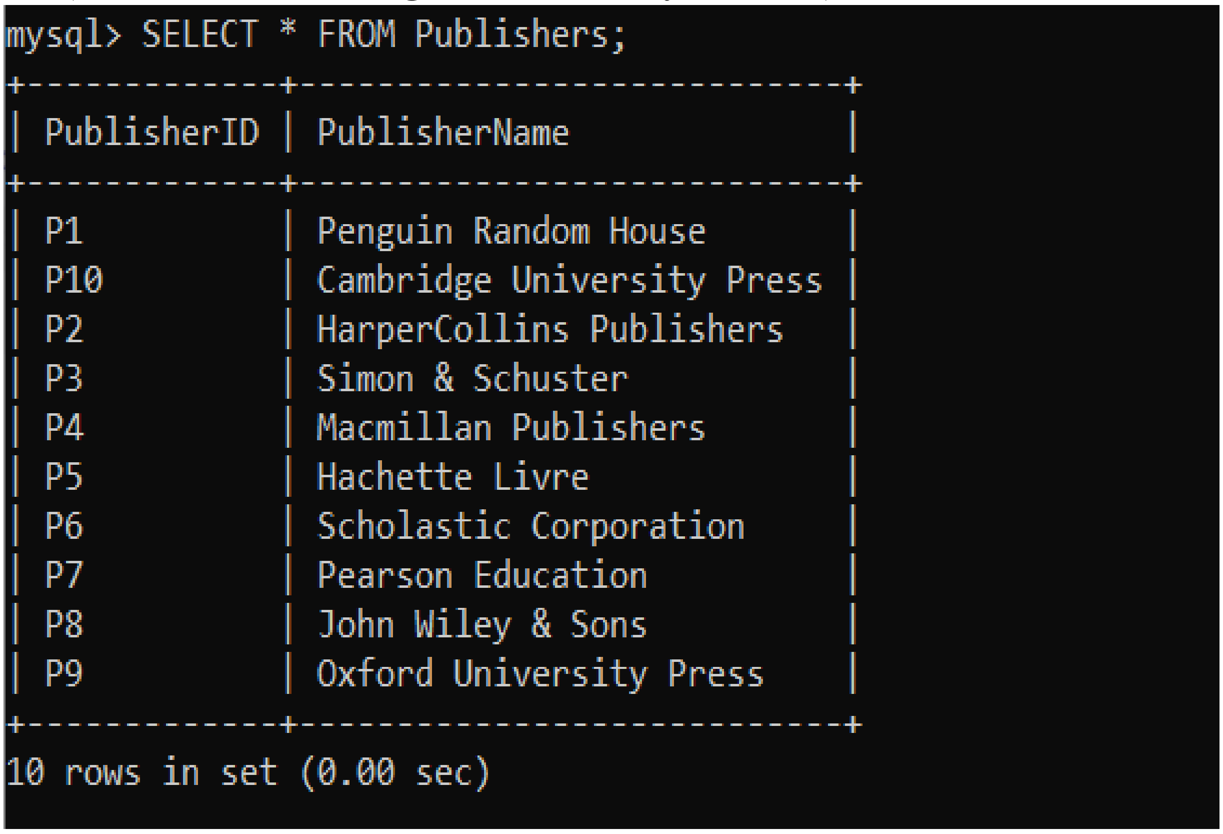
2. **Publishers:**

```
CREATE TABLE Publishers (
  PublisherID VARCHAR(10),
  PublisherName VARCHAR(255) NOT NULL,
  PRIMARY KEY (PublisherID)
);
```



```
mysql> DESC Publishers;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| PublisherID   | varchar(10)  | NO   | PRI | NULL    |       |
| PublisherName | varchar(255) | NO   |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

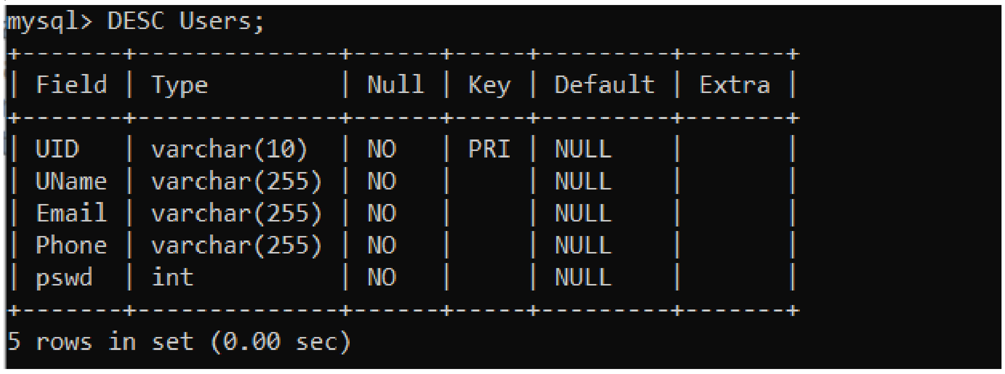
```
INSERT INTO Publishers (PublisherID, PublisherName)
VALUES
   ('P1', 'Penguin Random House'),
   ('P2', 'HarperCollins Publishers'),
   ('P3', 'Simon & Schuster'),
   ('P4', 'Macmillan Publishers'),
   ('P5', 'Hachette Livre'),
   ('P6', 'Scholastic Corporation'),
   ('P7', 'Pearson Education'),
   ('P8', 'John Wiley & Sons'),
   ('P9', 'Oxford University Press'),
   ('P10', 'Cambridge University Press');
```



```
mysql> SELECT * FROM Publishers;
+-------------+----------------------------+
| PublisherID | PublisherName              |
+-------------+----------------------------+
| P1          | Penguin Random House       |
| P10         | Cambridge University Press |
| P2          | HarperCollins Publishers   |
| P3          | Simon & Schuster           |
| P4          | Macmillan Publishers       |
| P5          | Hachette Livre             |
| P6          | Scholastic Corporation     |
| P7          | Pearson Education          |
| P8          | John Wiley & Sons          |
| P9          | Oxford University Press    |
+-------------+----------------------------+
10 rows in set (0.00 sec)
```

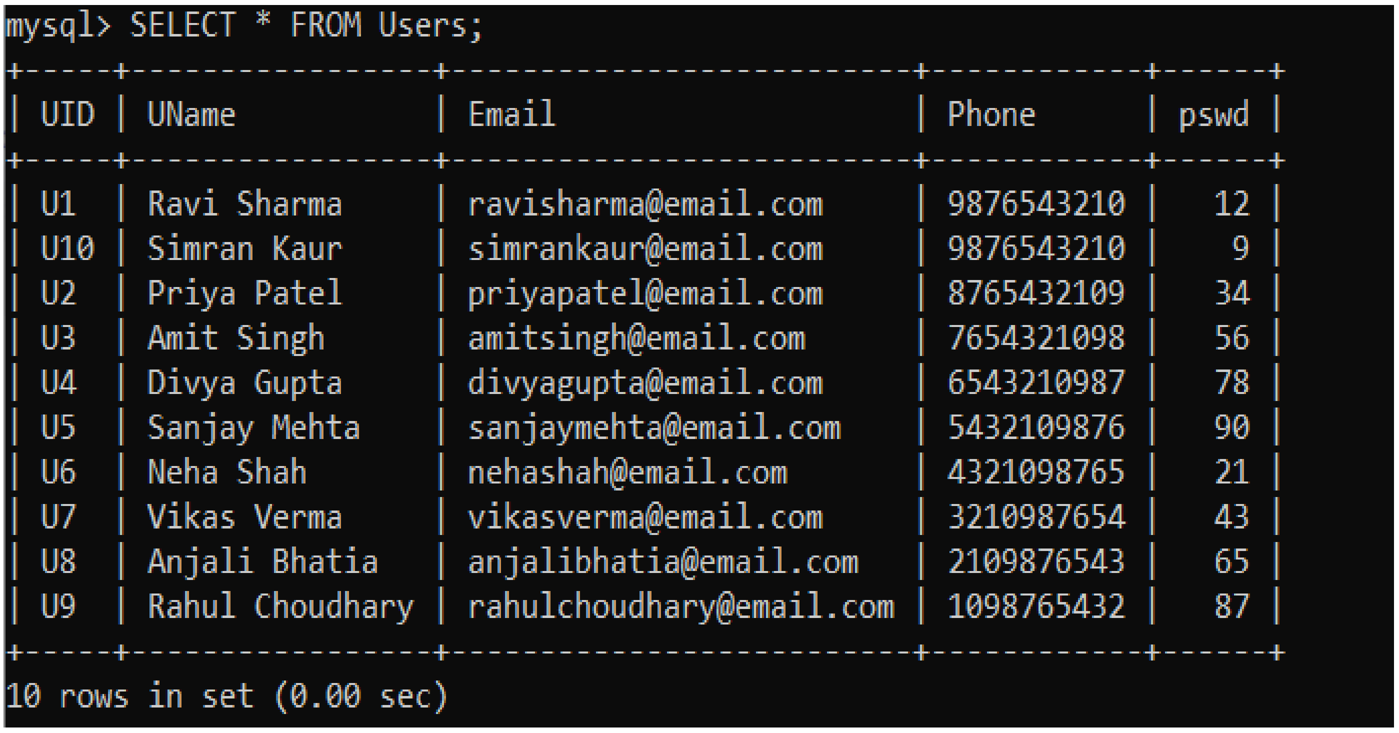
3. **Users:**

```
CREATE TABLE Users (
    UID VARCHAR(10),
    UName VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL,
    Phone VARCHAR(255) NOT NULL,
    pswd INT NOT NULL,
    PRIMARY KEY (UID)
);
```



```
mysql> DESC Users;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| UID   | varchar(10)  | NO   | PRI | NULL    |       |
| UName | varchar(255) | NO   |     | NULL    |       |
| Email | varchar(255) | NO   |     | NULL    |       |
| Phone | varchar(255) | NO   |     | NULL    |       |
| pswd  | int          | NO   |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

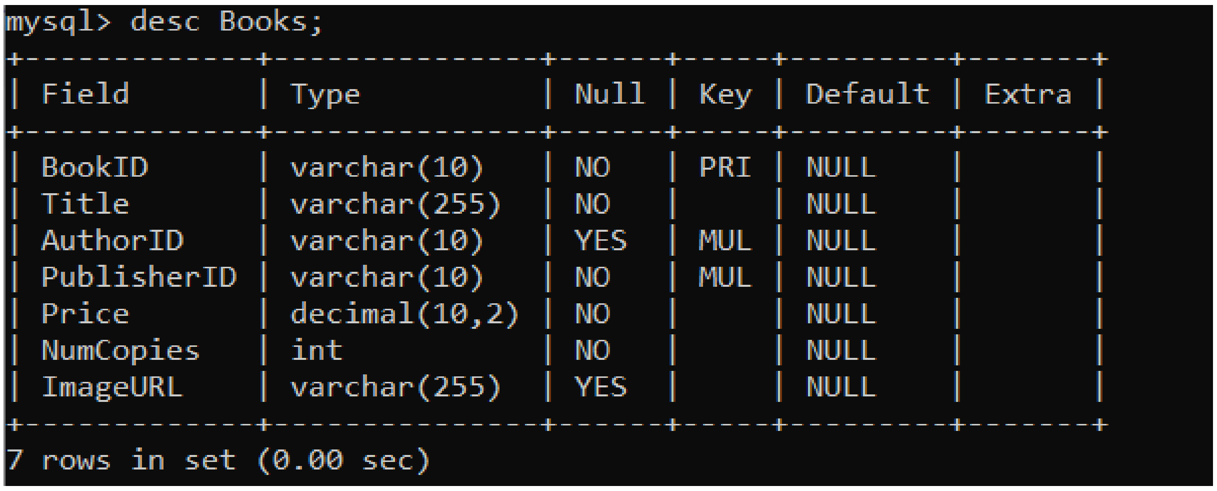
```
INSERT INTO Users (UID, UName, Email, Phone, pswd )
VALUES
    ('U1', 'Ravi Sharma', 'ravisharma@email.com', '9876543210',12),
    ('U2', 'Priya Patel', 'priyapatel@email.com', '8765432109', 34),
    ('U3', 'Amit Singh', 'amitsingh@email.com', '7654321098', 56),
    ('U4', 'Divya Gupta', 'divyagupta@email.com', '6543210987', 78),
    ('U5', 'Sanjay Mehta', 'sanjaymehta@email.com', '5432109876',
90),
    ('U6', 'Neha Shah', 'nehashah@email.com', '4321098765', 21),
    ('U7', 'Vikas Verma', 'vikasverma@email.com', '3210987654', 43),
    ('U8', 'Anjali Bhatia', 'anjalibhatia@email.com', '2109876543',
65),
    ('U9', 'Rahul Choudhary', 'rahulchoudhary@email.com',
'1098765432', 87),
    ('U10', 'Simran Kaur', 'simrankaur@email.com',
'9876543210',09);
```



```
mysql> SELECT * FROM Users;
+-----+-----------------+--------------------------+------------+------+
| UID | UName           | Email                    | Phone      | pswd |
+-----+-----------------+--------------------------+------------+------+
| U1  | Ravi Sharma     | ravisharma@email.com     | 9876543210 |   12 |
| U10 | Simran Kaur     | simrankaur@email.com     | 9876543210 |    9 |
| U2  | Priya Patel     | priyapatel@email.com     | 8765432109 |   34 |
| U3  | Amit Singh      | amitsingh@email.com      | 7654321098 |   56 |
| U4  | Divya Gupta     | divyagupta@email.com     | 6543210987 |   78 |
| U5  | Sanjay Mehta    | sanjaymehta@email.com    | 5432109876 |   90 |
| U6  | Neha Shah       | nehashah@email.com       | 4321098765 |   21 |
| U7  | Vikas Verma     | vikasverma@email.com     | 3210987654 |   43 |
| U8  | Anjali Bhatia   | anjalibhatia@email.com   | 2109876543 |   65 |
| U9  | Rahul Choudhary | rahulchoudhary@email.com | 1098765432 |   87 |
+-----+-----------------+--------------------------+------------+------+
10 rows in set (0.00 sec)
```

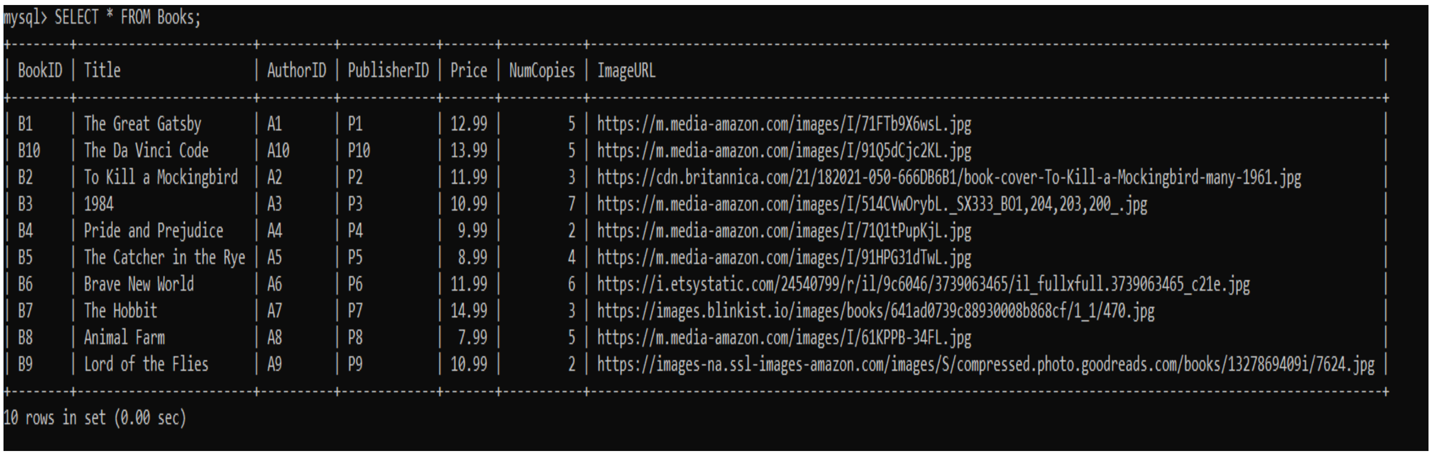
4. **Books:**

```
CREATE TABLE Books (
   BookID VARCHAR(10),
   Title VARCHAR(255) NOT NULL,
   AuthorID VARCHAR(10),
   PublisherID VARCHAR(10) NOT NULL,
   Price DECIMAL(10,2) NOT NULL,
   NumCopies INT NOT NULL,
   ImageURL VARCHAR(255),
   PRIMARY KEY (BookID),
   FOREIGN KEY (AuthorID) REFERENCES
Authors(AuthorID),
   FOREIGN KEY (PublisherID) REFERENCES
Publishers(PublisherID)
);
```



```
mysql> desc Books;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| BookID      | varchar(10)   | NO   | PRI | NULL    |       |
| Title       | varchar(255)  | NO   |     | NULL    |       |
| AuthorID    | varchar(10)   | YES  | MUL | NULL    |       |
| PublisherID | varchar(10)   | NO   | MUL | NULL    |       |
| Price       | decimal(10,2) | NO   |     | NULL    |       |
| NumCopies   | int           | NO   |     | NULL    |       |
| ImageURL    | varchar(255)  | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
INSERT INTO Books (BookID, Title, AuthorID, PublisherID, Price, NumCopies, ImageURL)
VALUES
    ('B1', 'The Great Gatsby', 'A1', 'P1', 12.99, 5, 'https://m.media-amazon.com/images/I/71FTb9X6wsL.jpg'),
    ('B2', 'To Kill a Mockingbird', 'A2', 'P2', 11.99, 3,'https://cdn.britannica.com/21/182021-050-666DB6B1/book-cover-To-Kill-a-Mockingbird-many-1961.jpg'),
    ('B3', '1984', 'A3', 'P3', 10.99, 7, 'https://m.media-amazon.com/images/I/514CVwOrybL._SX333_BO1,204,203,200_.jpg'),
    ('B4', 'Pride and Prejudice', 'A4', 'P4', 9.99, 2, 'https://m.media-amazon.com/images/I/71Q1tPupKjL.jpg'),
    ('B5', 'The Catcher in the Rye', 'A5', 'P5', 8.99, 4,'https://m.media-amazon.com/images/I/91HPG31dTwL.jpg'),
    ('B6', 'Brave New World', 'A6', 'P6', 11.99, 6,'https://i.etsystatic.com/24540799/r/il/9c6046/3739063465/il_fullxfull.3739063465_c21e.jpg'),
    ('B7', 'The Hobbit', 'A7', 'P7', 14.99, 3,'https://images.blinkist.io/images/books/641ad0739c88930008b868cf/1_1/470.jpg'),
    ('B8', 'Animal Farm', 'A8', 'P8', 7.99, 5,'https://m.media-amazon.com/images/I/61KPPB-34FL.jpg'),
    ('B9', 'Lord of the Flies', 'A9', 'P9', 10.99, 2,'https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1327869409i/7624.jpg'),
    ('B10', 'The Da Vinci Code', 'A10', 'P10', 13.99, 5,'https://m.media-amazon.com/images/I/91Q5dCjc2KL.jpg');
```

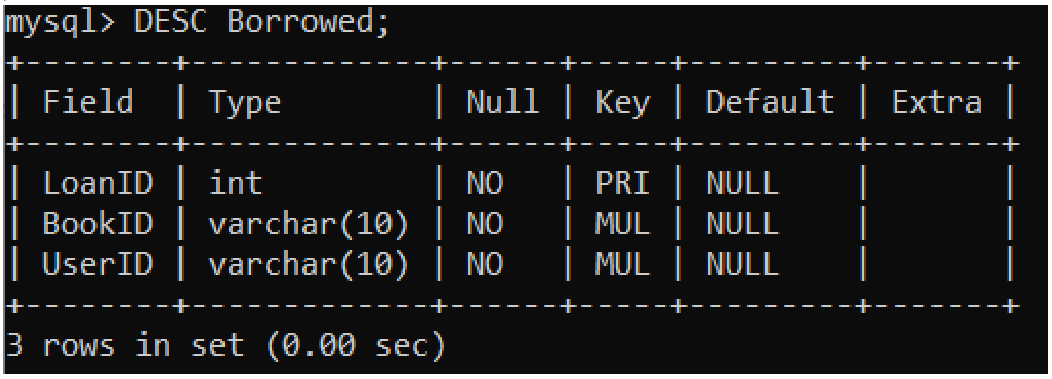


```
mysql> SELECT * FROM Books;
+--------+------------------------+----------+-------------+-------+-----------+------------------------------------------------------------------------------------------------------------+
| BookID | Title                  | AuthorID | PublisherID | Price | NumCopies | ImageURL                                                                                                   |
+--------+------------------------+----------+-------------+-------+-----------+------------------------------------------------------------------------------------------------------------+
| B1     | The Great Gatsby       | A1       | P1          | 12.99 |         5 | https://m.media-amazon.com/images/I/71FTb9X6wsL.jpg                                                        |
| B10    | The Da Vinci Code      | A10      | P10         | 13.99 |         5 | https://m.media-amazon.com/images/I/91Q5dCjc2KL.jpg                                                        |
| B2     | To Kill a Mockingbird  | A2       | P2          | 11.99 |         3 | https://cdn.britannica.com/21/182021-050-666DB6B1/book-cover-To-Kill-a-Mockingbird-many-1961.jpg           |
| B3     | 1984                   | A3       | P3          | 10.99 |         7 | https://m.media-amazon.com/images/I/514CVwOrybL._SX333_BO1,204,203,200_.jpg                                |
| B4     | Pride and Prejudice    | A4       | P4          |  9.99 |         2 | https://m.media-amazon.com/images/I/71Q1tPupKjL.jpg                                                        |
| B5     | The Catcher in the Rye | A5       | P5          |  8.99 |         4 | https://m.media-amazon.com/images/I/91HPG31dTwL.jpg                                                        |
| B6     | Brave New World        | A6       | P6          | 11.99 |         6 | https://i.etsystatic.com/24540799/r/il/9c6046/3739063465/il_fullxfull.3739063465_c21e.jpg                  |
| B7     | The Hobbit             | A7       | P7          | 14.99 |         3 | https://images.blinkist.io/images/books/641ad0739c88930008b868cf/1_1/470.jpg                               |
| B8     | Animal Farm            | A8       | P8          |  7.99 |         5 | https://m.media-amazon.com/images/I/61KPPB-34FL.jpg                                                        |
| B9     | Lord of the Flies      | A9       | P9          | 10.99 |         2 | https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1327869409i/7624.jpg |
+--------+------------------------+----------+-------------+-------+-----------+------------------------------------------------------------------------------------------------------------+

10 rows in set (0.00 sec)
```

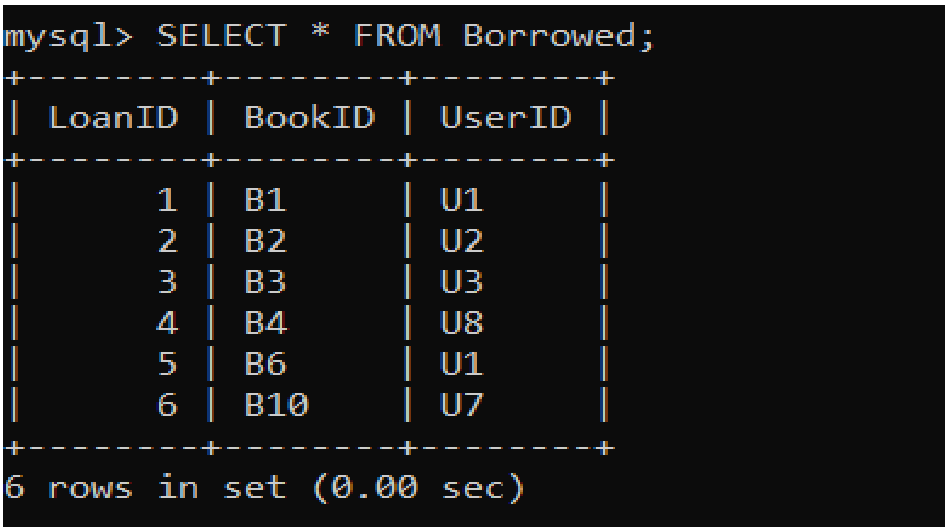
5. Borrowed:

```
CREATE TABLE Borrowed (
    LoanID INT NOT NULL PRIMARY KEY,
    BookID VARCHAR(10) NOT NULL,
    UserID VARCHAR(10) NOT NULL,
    FOREIGN KEY (BookID) REFERENCES Books(BookID),
    FOREIGN KEY (UserID) REFERENCES Users(UID)
);
```



```
mysql> DESC Borrowed;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| LoanID | int         | NO   | PRI | NULL    |       |
| BookID | varchar(10) | NO   | MUL | NULL    |       |
| UserID | varchar(10) | NO   | MUL | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
INSERT INTO Borrowed( LoanID, BookID, UserID) VALUES
(1, 'B1','U1'),
(2, 'B2','U2'),
(3, 'B3','U3'),
(4, 'B4','U8'),
(5, 'B6','U1'),
(6, 'B10','U7');
```



```
mysql> SELECT * FROM Borrowed;
+--------+--------+--------+
| LoanID | BookID | UserID |
+--------+--------+--------+
|      1 | B1     | U1     |
|      2 | B2     | U2     |
|      3 | B3     | U3     |
|      4 | B4     | U8     |
|      5 | B6     | U1     |
|      6 | B10    | U7     |
+--------+--------+--------+
6 rows in set (0.00 sec)
```

<u>**Normalization**</u>**:**

Normalization is the process of structuring and handling the relationship between data to minimize redundancy in the relational table and avoid the unnecessary anomalies properties from the database like insertion, update and delete. It helps to divide large database tables into smaller tables and make a relationship between them. It can remove redundant data and ease to add, manipulate or delete table fields.
To normalize the given schema up to BCNF, we need to identify functional dependencies and remove any partial dependencies and transitive dependencies.
Functional dependencies are:

For the Books table:

- BookID -> Title, AuthorID, PublisherID, Price, NumCopies, ImageURL
- AuthorID -> AuthorName
- PublisherID -> PublisherName
- For the Authors table:
- AuthorID -> AuthorName
- For the Publishers table:

- PublisherID -> PublisherName

For the Users table:

- UID -> UName, Email, Phone, pswd

For the Borrowed table:

- LoanID -> BookID, UserID
- After normalization, the following tables are produced:

Step 1: Checking for 1NF
A relation R is in first normal form (1NF) if and only if it does not contain any composite
attribute or multi-valued attributes or their combinations.

Step 2: Checking for 2NF
A relation R is in second normal form (2NF)

- if and only if it is in 1NF and
- every non-primary key attribute is fully dependent on the primary key.

Step 3: Checking for 3NF
A relation R is in third normal form (3NF)

- if and only if it is in 2NF and
- every non-key attribute is non-transitively dependent on the primary key

All our tables are already in 3NF since there is no transitive dependency.

Step 4: Checking for BCNF
A relation R is in Boyce-Cott normal form (BCNF) if

- It and only if it is in 3NF and

- every determinant should be the primary key.

All the tables are in BCNF already since they follow the above two conditions.

After normalization, the following tables are produced:

- Books (BookID, Title , PublisherID , Price , NumCopies , ImageURL);
- Book_Authors (BookID , AuthorID);
- Authors (AuthorID , AuthorName);
- Publishers (PublisherID , PublisherName);
- Users (UserID , UserName, Email , Phone , Password);
- Borrowed (LoanID, BookID , UserID , BorrowDate, ReturnDate);

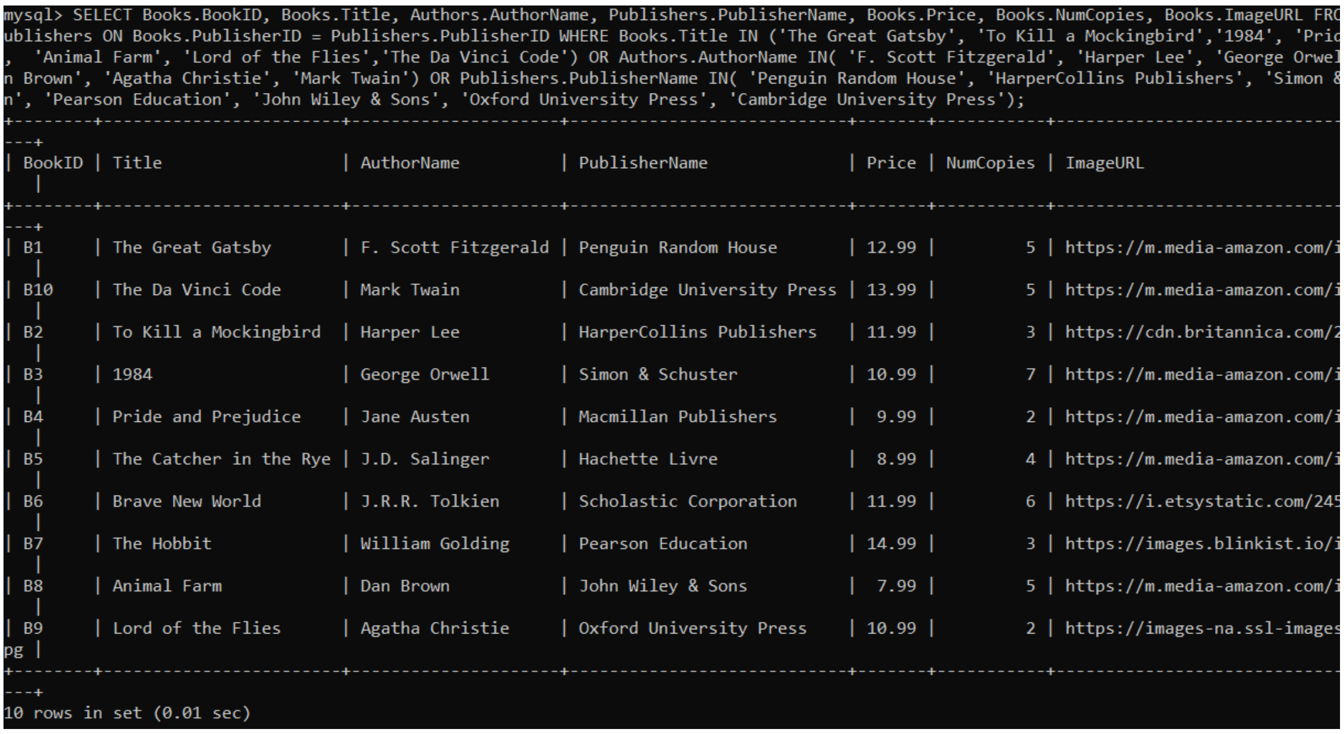
Authors and Book_Authors. This is because the original table was not in third normal form since it contained a transitive dependency between AuthorID and AuthorName. By splitting the table into two, we eliminate this dependency.

<u>**Sample Queries**</u>:

1.
   ```
   SELECT Books.BookID, Books.Title,
   Authors.AuthorName, Publishers.PublisherName,
   Books.Price, Books.NumCopies, Books.ImageURL
   FROM Books
   INNER JOIN Authors ON Books.AuthorID =
   Authors.AuthorID
   ```

```
INNER JOIN Publishers ON Books.PublisherID =
Publishers.PublisherID
WHERE Books.Title IN ('The Great Gatsby', 'To Kill a
Mockingbird','1984', 'Pride and Prejudice', 'The Catcher in
the Rye',  'Brave New World', 'The Hobbit',  'Animal Farm',
'Lord of the Flies','The Da Vinci Code')
OR Authors.AuthorName IN( 'F. Scott Fitzgerald', 'Harper
Lee', 'George Orwell', 'Jane Austen', 'J.D. Salinger', 'J.R.R.
Tolkien', 'William Golding',' Dan Brown', 'Agatha Christie',
'Mark Twain')
OR Publishers.PublisherName IN( 'Penguin Random
House', 'HarperCollins Publishers', 'Simon & Schuster',
'Macmillan Publishers', 'Hachette Livre', 'Scholastic
Corporation', 'Pearson Education', 'John Wiley & Sons',
'Oxford University Press', 'Cambridge University Press');
```

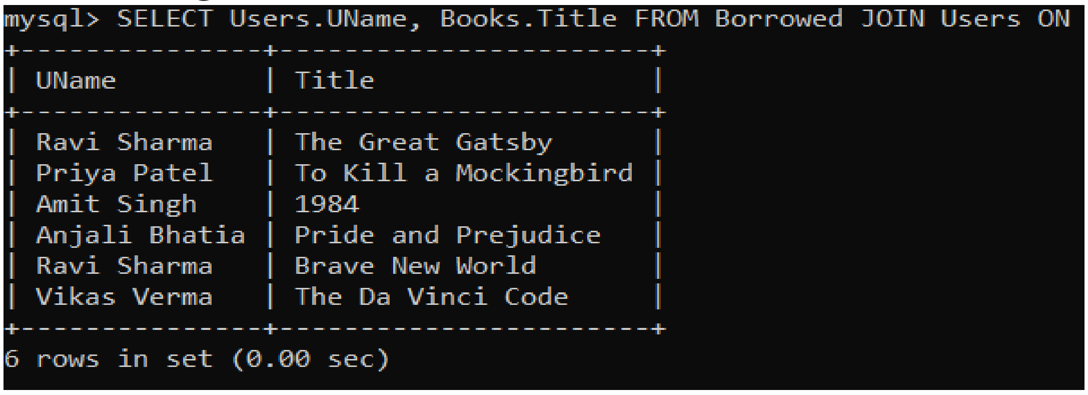
- The above query is a SQL query that retrieves data from the Books, Authors, and Publishers tables. It uses inner joins to combine data from these tables based on specific conditions, and a WHERE clause to filter the results based on certain book titles, author names, and publisher names.
- This query retrieves data that combines information from the Books, Authors, and Publishers tables, and filters the results based on specific book titles, author names, and publisher names listed in the query. This query is likely used to retrieve information about books, along with their associated authors and publishers, based on specific criteria.



```
mysql> SELECT Books.BookID, Books.Title, Authors.AuthorName, Publishers.PublisherName, Books.Price, Books.NumCopies, Books.ImageURL FRO
ublishers ON Books.PublisherID = Publishers.PublisherID WHERE Books.Title IN ('The Great Gatsby', 'To Kill a Mockingbird','1984', 'Pric
,  'Animal Farm', 'Lord of the Flies','The Da Vinci Code') OR Authors.AuthorName IN( 'F. Scott Fitzgerald', 'Harper Lee', 'George Orwel
n Brown', 'Agatha Christie', 'Mark Twain') OR Publishers.PublisherName IN( 'Penguin Random House', 'HarperCollins Publishers', 'Simon &
n', 'Pearson Education', 'John Wiley & Sons', 'Oxford University Press', 'Cambridge University Press');
+--------+------------------------+---------------------+----------------------------+-------+-----------+---------------------------
---+
| BookID | Title                  | AuthorName          | PublisherName              | Price | NumCopies | ImageURL
   |
+--------+------------------------+---------------------+----------------------------+-------+-----------+---------------------------
---+
| B1     | The Great Gatsby       | F. Scott Fitzgerald | Penguin Random House       | 12.99 |         5 | https://m.media-amazon.com/i
   |
| B10    | The Da Vinci Code      | Mark Twain          | Cambridge University Press | 13.99 |         5 | https://m.media-amazon.com/i
   |
| B2     | To Kill a Mockingbird  | Harper Lee          | HarperCollins Publishers   | 11.99 |         3 | https://cdn.britannica.com/2
   |
| B3     | 1984                   | George Orwell       | Simon & Schuster           | 10.99 |         7 | https://m.media-amazon.com/i
   |
| B4     | Pride and Prejudice    | Jane Austen         | Macmillan Publishers       |  9.99 |         2 | https://m.media-amazon.com/i
   |
| B5     | The Catcher in the Rye | J.D. Salinger       | Hachette Livre             |  8.99 |         4 | https://m.media-amazon.com/i
   |
| B6     | Brave New World        | J.R.R. Tolkien      | Scholastic Corporation     | 11.99 |         6 | https://i.etsystatic.com/245
   |
| B7     | The Hobbit             | William Golding     | Pearson Education          | 14.99 |         3 | https://images.blinkist.io/i
   |
| B8     | Animal Farm            | Dan Brown           | John Wiley & Sons          |  7.99 |         5 | https://m.media-amazon.com/i
   |
| B9     | Lord of the Flies      | Agatha Christie     | Oxford University Press    | 10.99 |         2 | https://images-na.ssl-images
pg |
+--------+------------------------+---------------------+----------------------------+-------+-----------+---------------------------
---+
10 rows in set (0.01 sec)
```

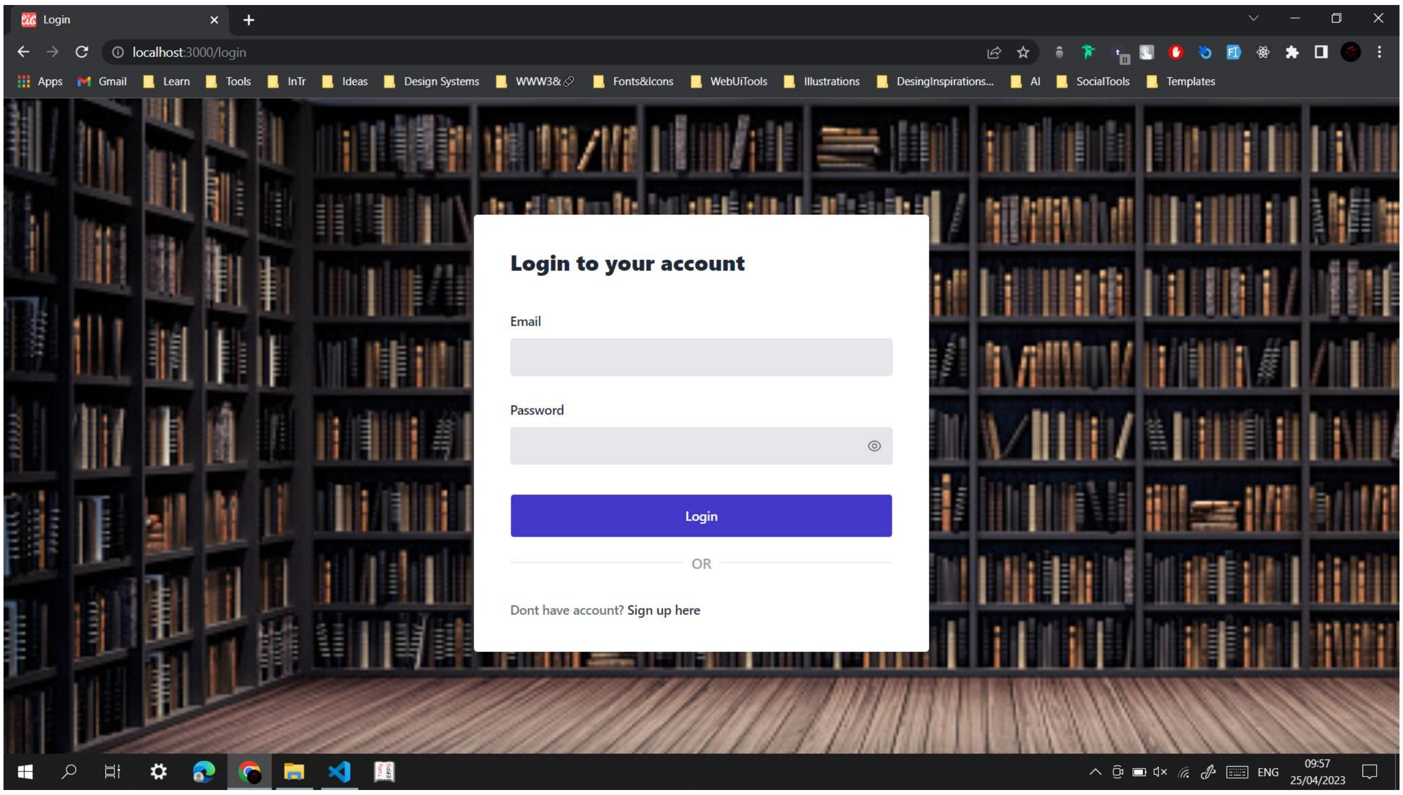
2. SELECT Users.UName, Books.Title
        FROM Borrowed
        JOIN Users ON Borrowed.UserID = Users.UID
        JOIN Books ON Borrowed.BookID = Books.BookID;

- The above query is a SQL query that retrieves data from three tables: Users, Borrowed, and Books. It uses joins to combine data from these tables based on specific conditions.
- This query retrieves data that combines the "UName" column from the Users table with the "Title" column from the Books table, based on matching "UserID" and "BookID" values in the Borrowed table. This query is likely used to retrieve information about users who have borrowed books, along with the titles of the borrowed books.



```
mysql> SELECT Users.UName, Books.Title FROM Borrowed JOIN Users ON
+---------------+-----------------------+
| UName         | Title                 |
+---------------+-----------------------+
| Ravi Sharma   | The Great Gatsby      |
| Priya Patel   | To Kill a Mockingbird |
| Amit Singh    | 1984                  |
| Anjali Bhatia | Pride and Prejudice   |
| Ravi Sharma   | Brave New World       |
| Vikas Verma   | The Da Vinci Code     |
+---------------+-----------------------+
6 rows in set (0.00 sec)
```

Front End:



Login to your account

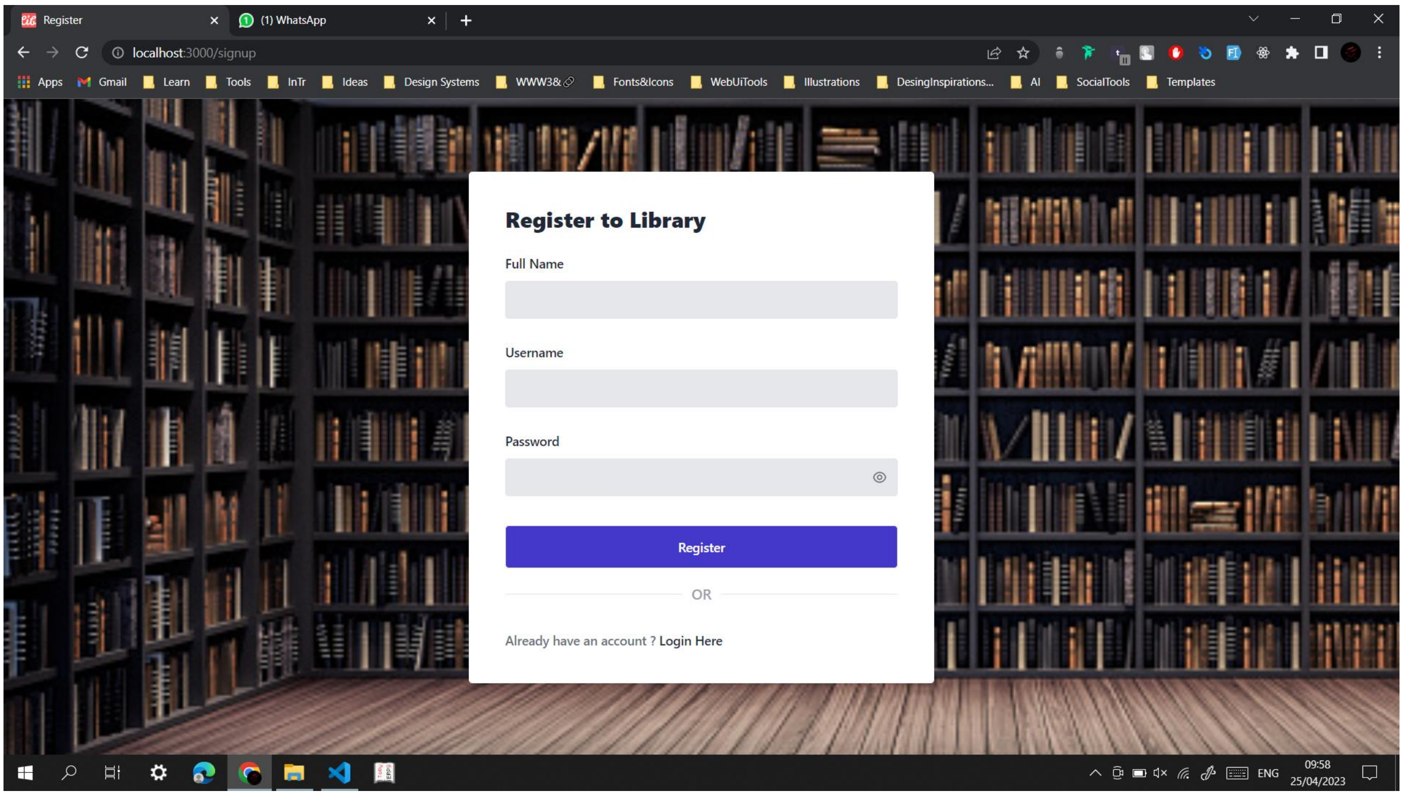
Email

Password

Login

OR

Dont have account? Sign up here



Register to Library

Full Name

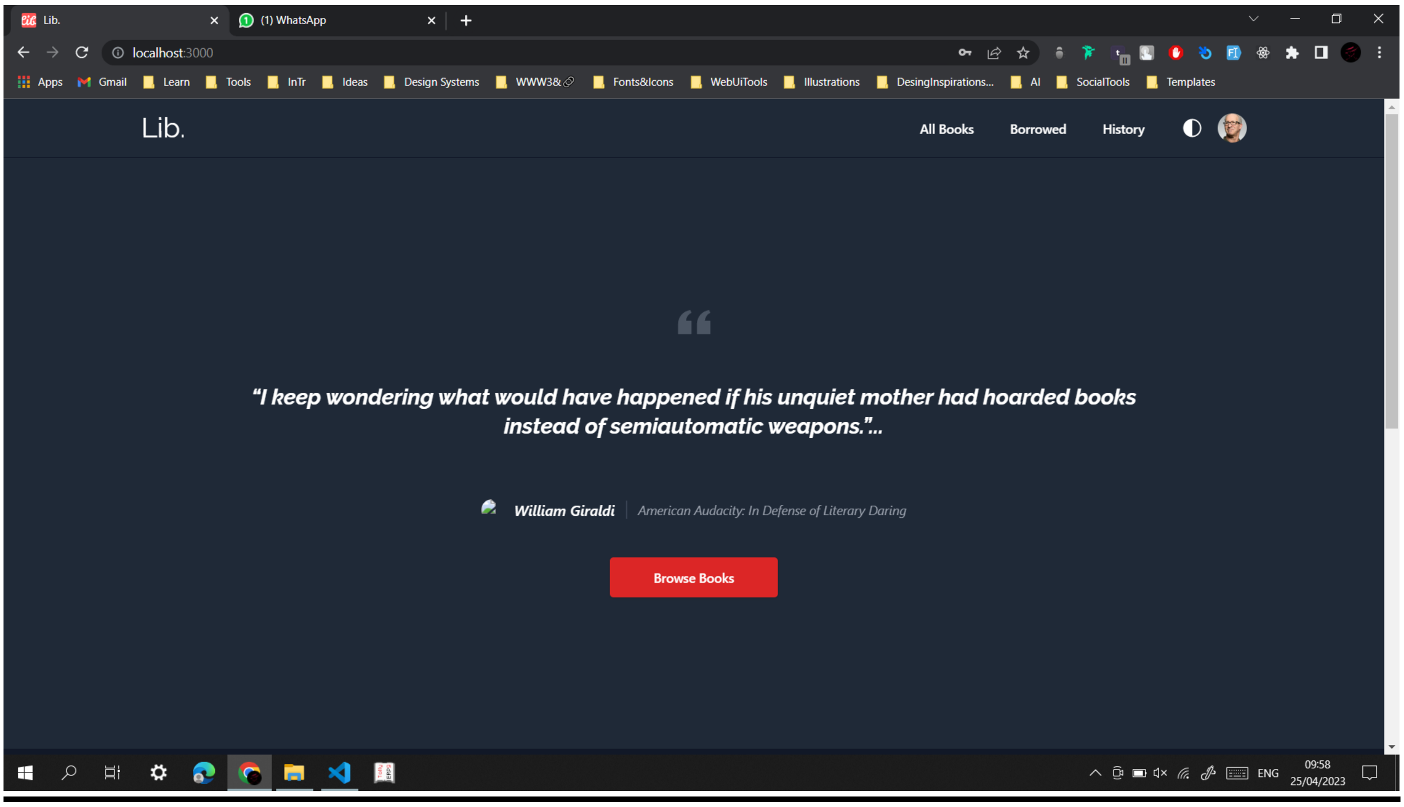
Username

Password

Register

OR

Already have an account ? Login Here

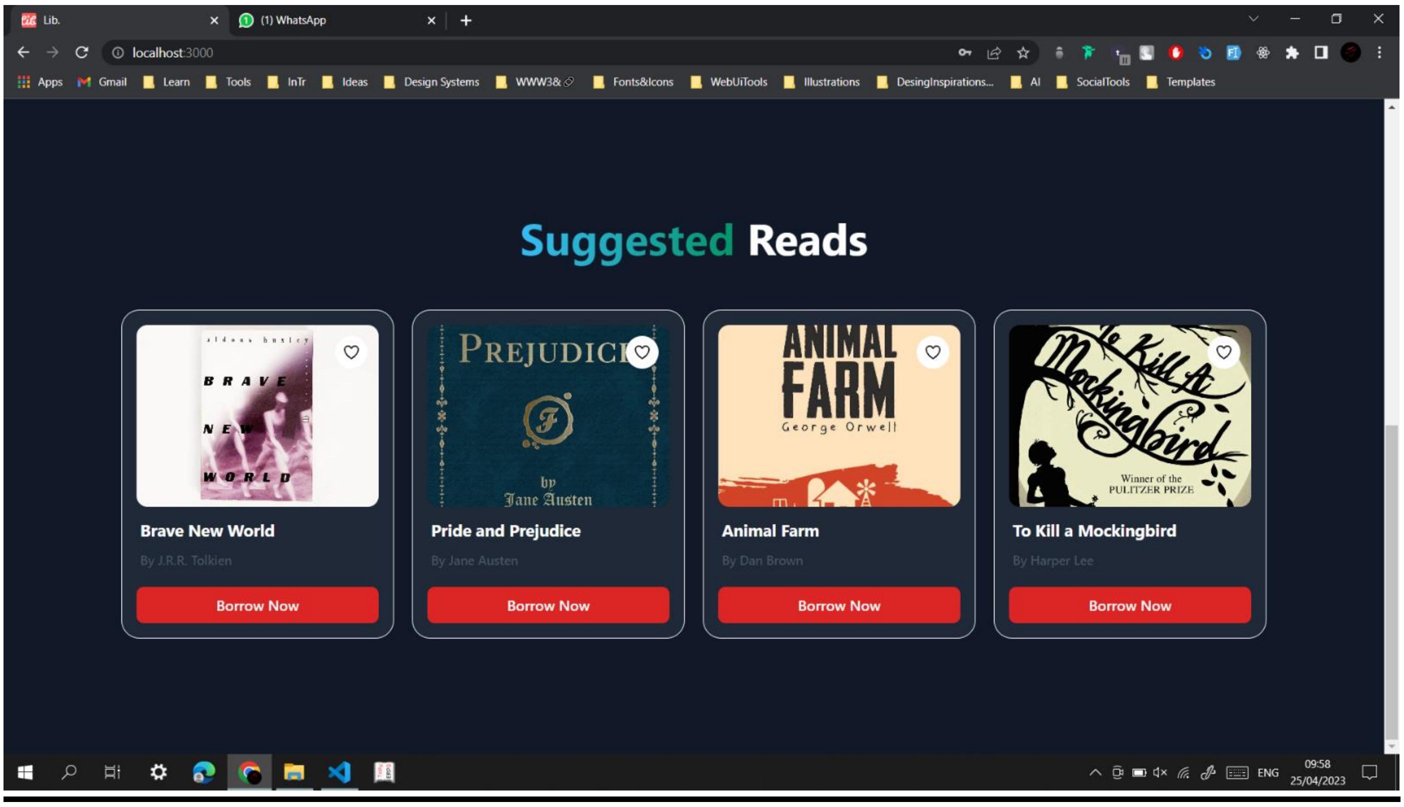

Lib.

All Books Borrowed History

"I keep wondering what would have happened if his unquiet mother had hoarded books instead of semiautomatic weapons."...

William Giraldi | *American Audacity: In Defense of Literary Daring*

Browse Books

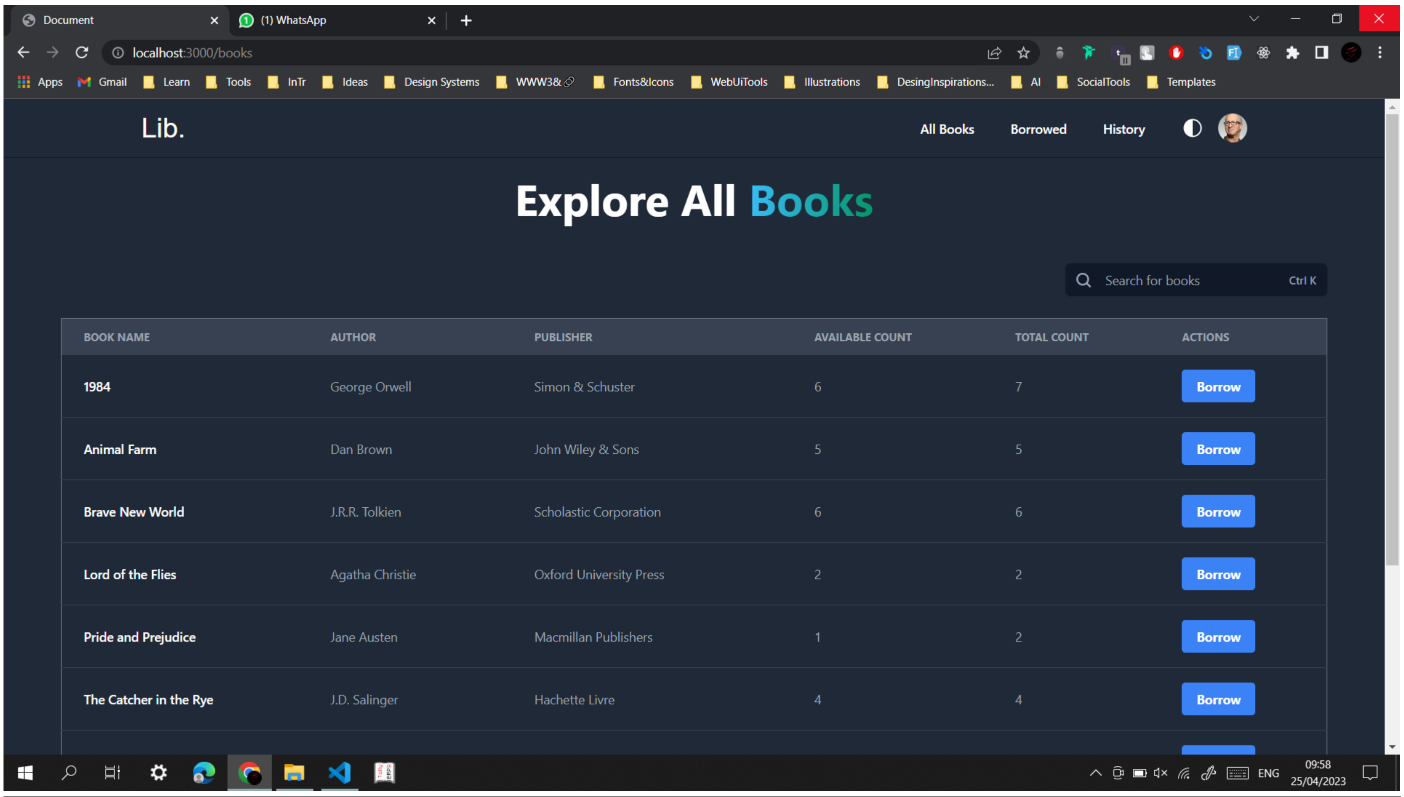

Suggested Reads

Brave New World
By J.R.R. Tolkien
Borrow Now

Pride and Prejudice
By Jane Austen
Borrow Now

Animal Farm
By Dan Brown
Borrow Now

To Kill a Mockingbird
By Harper Lee
Borrow Now


localhost:3000/books

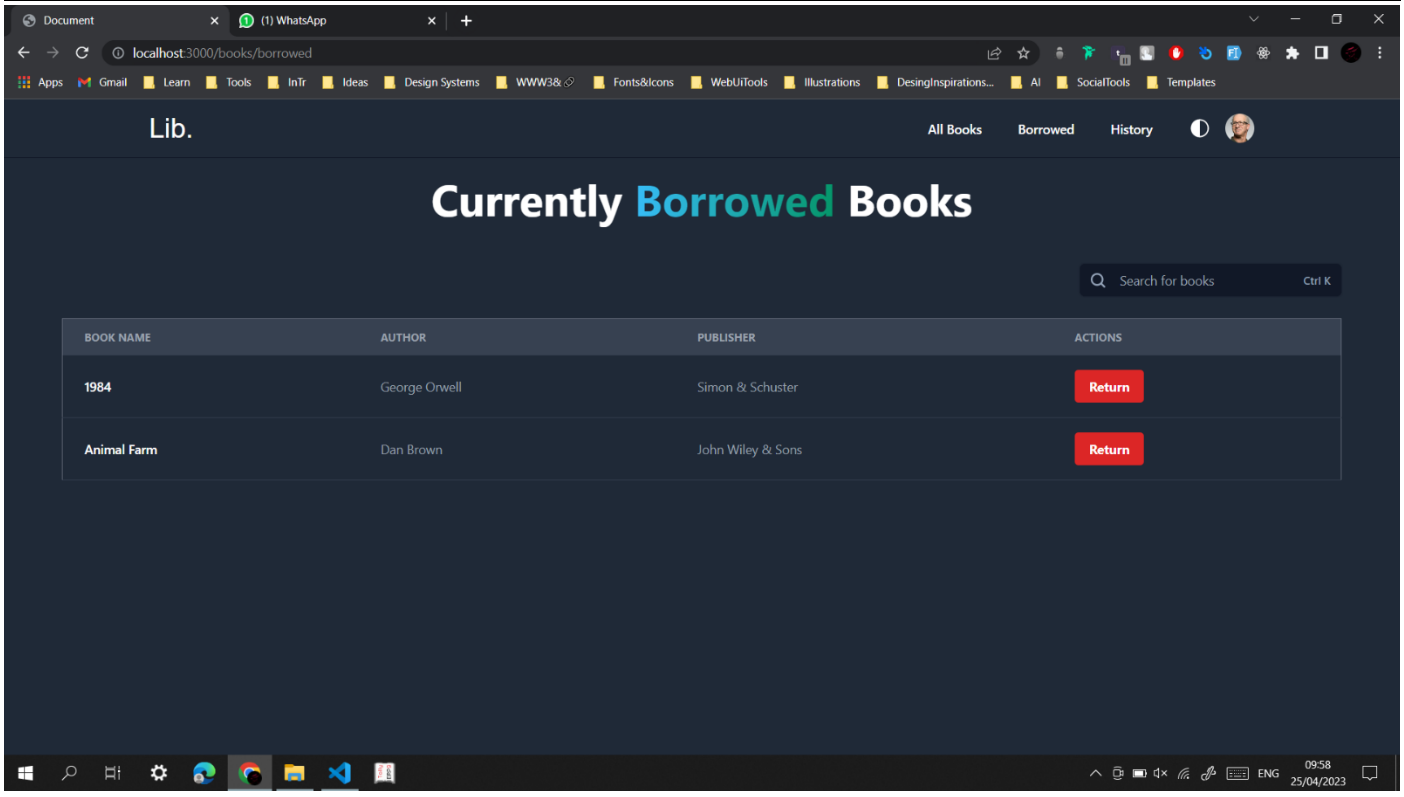
Lib.

All Books Borrowed History

Explore All Books

Search for books Ctrl K

BOOK NAME	AUTHOR	PUBLISHER	AVAILABLE COUNT	TOTAL COUNT	ACTIONS
1984	George Orwell	Simon & Schuster	6	7	Borrow
Animal Farm	Dan Brown	John Wiley & Sons	5	5	Borrow
Brave New World	J.R.R. Tolkien	Scholastic Corporation	6	6	Borrow
Lord of the Flies	Agatha Christie	Oxford University Press	2	2	Borrow
Pride and Prejudice	Jane Austen	Macmillan Publishers	1	2	Borrow
The Catcher in the Rye	J.D. Salinger	Hachette Livre	4	4	Borrow


localhost:3000/books/borrowed

Lib.

All Books Borrowed History

Currently Borrowed Books

Search for books Ctrl K

BOOK NAME	AUTHOR	PUBLISHER	ACTIONS
1984	George Orwell	Simon & Schuster	Return
Animal Farm	Dan Brown	John Wiley & Sons	Return

Conclusion:

In conclusion, the Library Management System project is designed to effectively manage the data related to books, authors, publishers, and users in a library setting. The tables, including "Books", "Authors", "Publishers", and "Users", are structured with unique identifiers, foreign keys, and relevant columns to store and retrieve information efficiently. The project aims to streamline library operations, facilitate book tracking, and enhance user experience. By implementing this robust database management system, libraries can effectively organize and manage their resources, making it easier for users to access and utilize library materials.