CNC-314 Database Systems

Dr/Mohamed Isaa

Library Management System

Name	ID	Dep	Section	M#1	M#2	M#3	M#4	Bonus	Total
Jana Khaled Beshir	320220019	AID	1						
Sara Samy El-wakeel	320220020	AID	1						
Sara Aly Ahmed	320220024	AID	1						
Omar Khaled Ashour	320220049	AID	1						
Mohamed Gamal Hussein	320220197	CNC	3						

Table of contents

Database Description

The Library Management System (LMS) database is designed to manage and track the library's operations, including Book, Member, borrowing activities, Reservation, and financial transactions. The system has nine entities: **Book, Genre, Shelf, Member, Borrow, Reservation, Library Branch, Librarian, and Transaction**.

The **Book** entity holds details about each book, such as *Book ID (Primary Key), Title, Author, ISBN, Genre ID (Foreign Key to Genre), Shelf ID (Foreign Key to **Shelf**), Publication Year, Stock, and Status (Available or Borrowed)*. Book is linked to **Genre** through a one-to-many **(1: N) relationship** and to Shelf via a one-to-one **(1:1) relationship**, ensuring that each shelf uniquely holds a book.

The library manages members using the **Member** entity, which includes *Member ID (Primary Key), Name, Email, Phone, Membership Date, and Branch ID (Foreign Key to Library Branch)*. Member can borrow books, with the borrowing activity recorded in the **Borrow** entity. This entity *tracks Borrow ID (Primary Key), Book ID (Foreign Key to Book), Member ID (Foreign Key to Member), Borrow Date, Due Date, Return Date, and Fine Amount*. Each Member can borrow multiple Book records, resulting in a one-to-many (1: N) relationship between Member and Borrow. Book records can also be reserved by Member records using the **Reservation** entity, which stores *Reservation ID (Primary Key), Member ID (Foreign Key to Member), Book ID (Foreign Key to Book), Reservation Date, and Status (Pending, Approved, or Cancelled)*. Member and Reservation share a one-to-many **(1: N) relationship**, with each Member being able to reserve multiple Book records.

Library operations occur at different branches, represented by the **Library Branch** entity, which has *Branch ID (Primary Key), Name, Location, and Phone*. Each Member is associated with a specific Library Branch, forming a one-to-many **(1: N) relationship** between Library Branch and Member. Additionally, branches employ librarians, managed via the **Librarian** entity, which includes *Librarian ID (Primary Key), Name, Email, Phone, Hire Date, and Branch ID (Foreign Key to Library Branch)*. Each Library Branch can have multiple Librarian records, creating another one-to-many **(1: N) relationship.**

Financial transactions are managed through the **Transaction** entity, which stores *Transaction ID (Primary Key), Borrow ID (Foreign Key to Borrow), Payment Date, and Amount Paid*. Each Borrow record corresponds to one Transaction, establishing a one-to-one **(1:1) relationship** between Borrow and Transaction. This database design ensures that all library operations are efficiently tracked and managed, with clear relationships and cardinalities between entities. Participation constraints ensure data integrity; for example, every Book must belong to a Genre, every Borrow must reference a Book and Member, and every Member must be associated with a Library Branch. This design is scalable, secure, and optimized to handle all library-related functionalities.

Business Requirements

Efficient Resource Management

- Ensure effective cataloging, tracking, and management of library resources such as books, shelves, and digital media.

Enhanced Member Experience

- Provide members with seamless access to library services, including borrowing, , and returning books with minimal delays.

Data-Driven Decision Making

- Implement reporting features to generate insights on library operations, such as popular books, borrowing trends, and financial summaries.

Operational Scalability

- Support expansion to additional library branches and integration of new resource types, such as eBooks or audiobooks.

Financial Integrity

- Maintain transparency and accuracy in all financial transactions, including fine collection and membership fee management.

Stakeholders' Requirements

Primary Stakeholders

1. Members:
   - Ability to search for, borrow easily.
   - Access to borrowing history and transaction records.
2. Librarians:
   - Efficient tools to manage book inventory, member registrations, and borrowing activities.
   - Capabilities to update book locations, handle reservations, and track overdue fines.
   - User-friendly reporting tools to monitor branch performance.
3. Library Administrators:
   - Role-based access control to manage staff operations.
   - Comprehensive reports on borrowing patterns, member activity, and revenue.
   - Scalability options for adding new branches and resource types.

Secondary Stakeholders

1. IT and Database Administrators:
   - Secure and scalable database architecture to support the growing needs of the library.
   - Minimal downtime and robust disaster recovery mechanisms.
   - Tools for troubleshooting system issues efficiently.
2. Library Branch Managers:
   - Oversight capabilities to monitor branch-specific operations, including active members, book stock, and librarian schedules.
   - Easy-to-use dashboards for performance tracking.

3. Policy Makers:
    - Compliance with local and international data protection and library management standards.
    - Support for enforcing borrowing limits and fine policies.

Functional Requirements

1. Book Management:
    - Add, update, delete, and view books.
    - Track book details such as title, author, ISBN, genre, shelf, publication year, stock, and status.
    - Locate books on specific shelves and update their location when needed.

2. Genre Management:
    - Add and manage genres.
    - Associate books with genres and ensure genre-based categorization.

3. Shelf Management:
    - Add, update, and delete shelves.
    - Track the location of shelves and associate them with specific genres.

4. Member Management:
    - Register new members with details such as name, email, phone, membership date, and branch association.
    - Update or delete member profiles.
    - View borrowing and reservation history for each member.

5. Borrowing Management:
    - Borrow books and assign due dates.
    - Track overdue books and calculate late fees.
    - Update the return status of borrowed books.

6. Reservation Management:
    - Allow members to reserve books.
    - Track reservation statuses (Pending, Approved, or Cancelled).
    - Notify members of changes in reservation status.

7. Library Branch Management:
   - Add, update, and delete library branches.
   - Track branch details, including name, location, and contact information.
     - Associate members and librarians with specific branches.

8. Librarian Management:
   - Add, update, and delete librarians.
   - Track librarian details, including their branch association.
   - Manage librarian schedules and responsibilities.

9. Transaction Management:
   - Process payments for fines or borrowing activities.
   - Track transaction details such as payment date and amount paid.
   - Generate reports on financial transactions.

10. Reporting:
    - Generate reports on popular books, borrowing patterns, genre preferences, and overdue books.
    - Track library branch statistics, such as total members and active librarians.

Non-Functional Requirements

1. Performance:
   - Ensure that search and retrieval of books, members, or transactions are completed within 2 seconds.
   - Support up to 1000 concurrent users without performance degradation.

2. Scalability:
   - Allow future expansion to include digital resources (eBooks, audiobooks) or additional library branches.
   - Support large datasets, such as thousands of books and members.

3. Security:
   - Encrypt sensitive data such as member contact details, librarian details, and transaction records.
   - Implement role-based access control to restrict operations based on user roles (e.g., librarian, admin).

4. Availability:
   - Guarantee 99.9% system uptime to ensure seamless access for members and librarians.
   - Implement robust data backups to prevent data loss in case of failure.
   - Usability:
   - Provide a user-friendly interface for members, librarians, and administrators.
   - Ensure the system is accessible on multiple devices, including desktops and mobile devices.

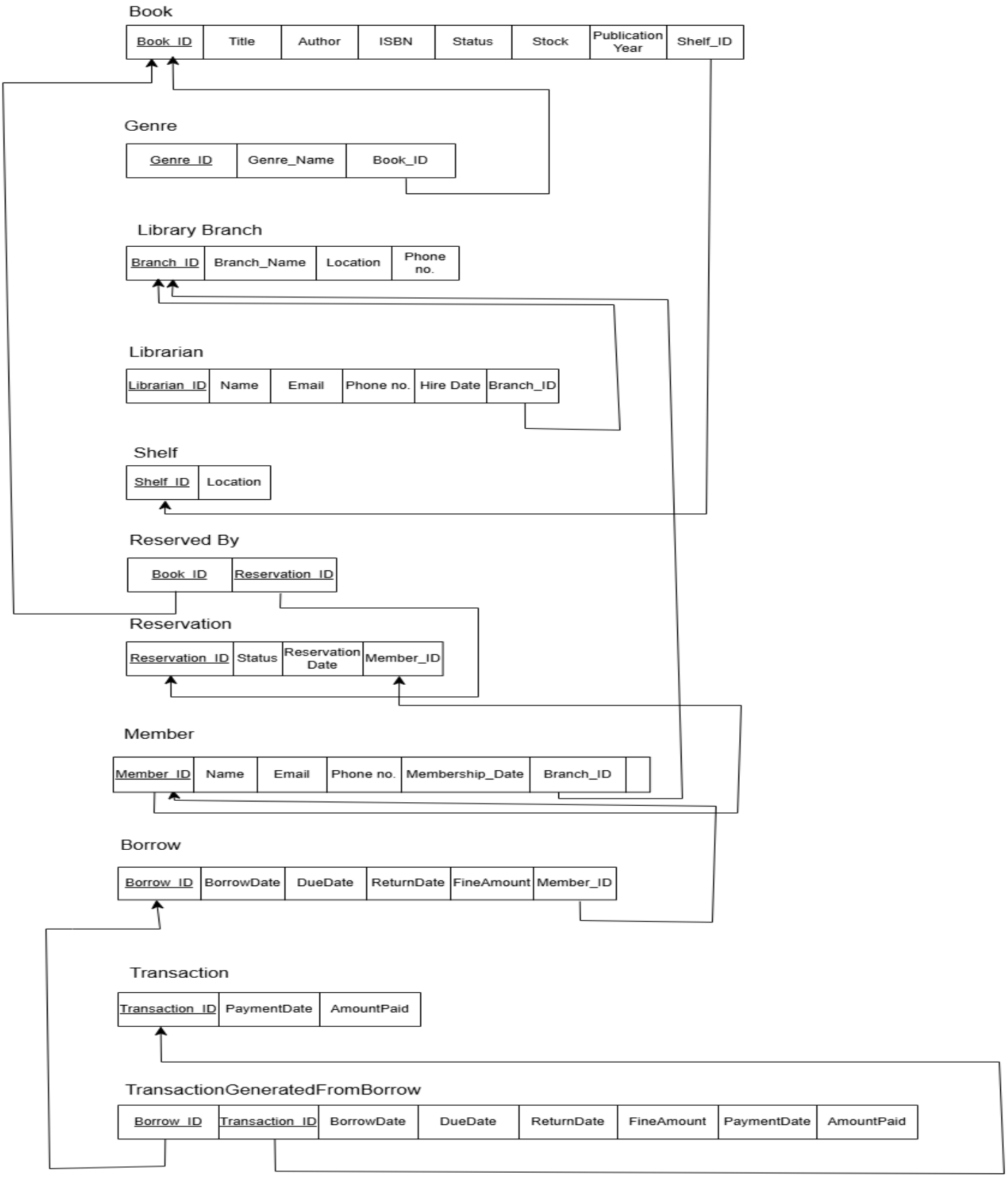
5. Reliability:
   - Maintain data consistency and integrity across entities to prevent issues like double-booking or invalid transactions.
   - Automatically update book stock and status in real-time during borrowing or return operations.

6. Data Integrity:
   - Enforce validation rules, such as preventing members from borrowing more books than allowed or reserving unavailable books.
   - Ensure accurate tracking of overdue books and calculation of fines.

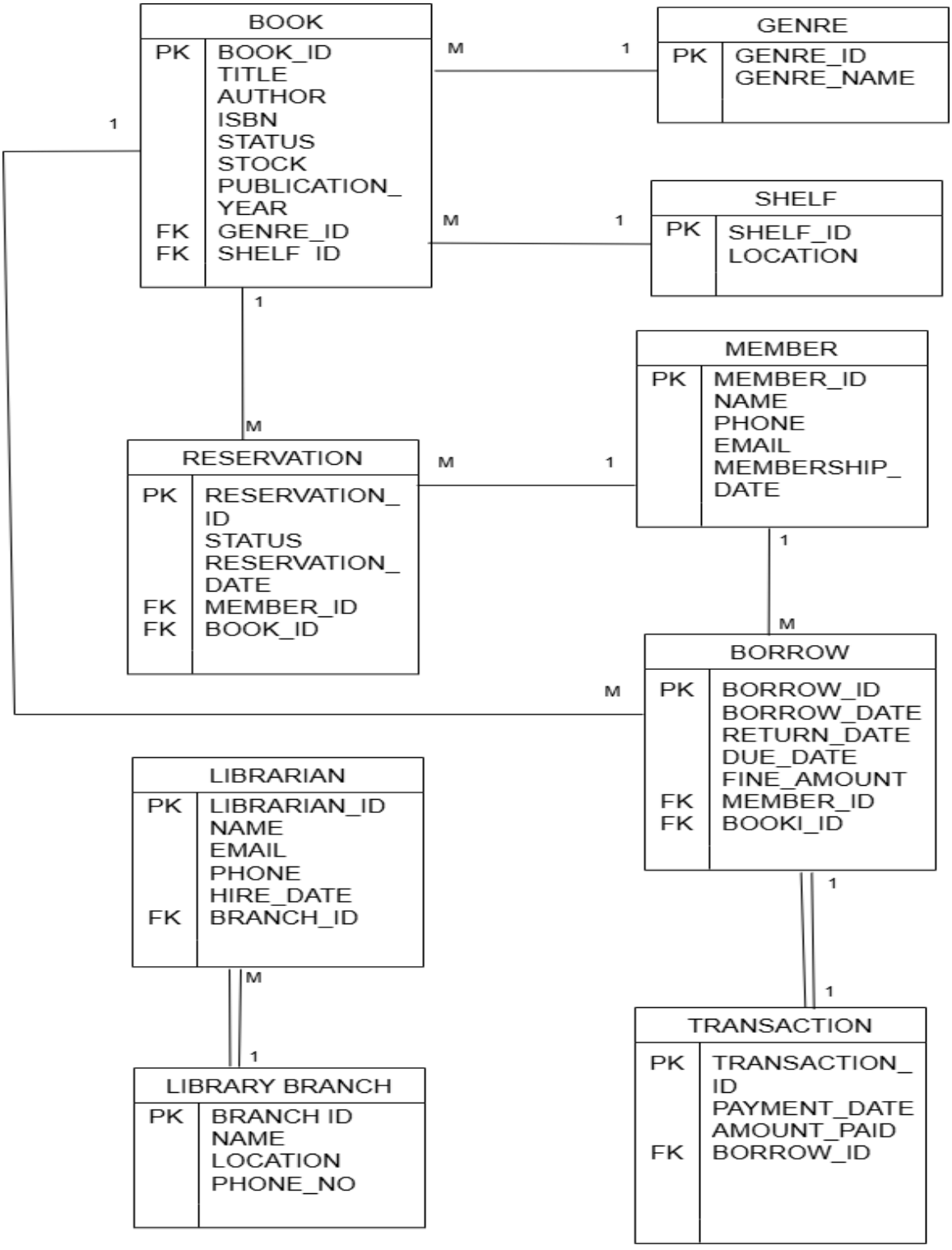
7. Compliance:
   - Comply with relevant data protection regulations to safeguard member and librarian information.
   - Adhere to library policies regarding borrowing limits and overdue fines.
   - This set of requirements reflects the updated system's entities and functionalities, ensuring a robust and scalable library management solution. Let me know if you need further assistance!

ERD

Mapping

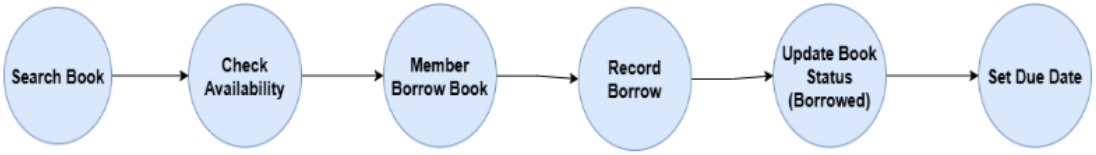

Book
Book_ID
Title
Author
ISBN
Status
Stock
Publication Year
Shelf_ID
Genre
Genre_ID
Genre_Name
Book_ID
Library Branch
Branch_ID
Branch_Name
Location
Phone no.
Librarian
Librarian_ID
Name
Email
Phone no.
Hire Date
Branch_ID
Shelf
Shelf_ID
Location
Reserved By
Book_ID
Reservation_ID
Reservation
Reservation_ID
Status
Reservation Date
Member_ID
Member
Member_ID
Name
Email
Phone no.
Membership_Date
Branch_ID
Borrow
Borrow_ID
BorrowDate
DueDate
ReturnDate
FineAmount
Member_ID
Transaction
Transaction_ID
PaymentDate
AmountPaid
TransactionGeneratedFromBorrow
Borrow_ID
Transaction_ID
BorrowDate
DueDate
ReturnDate
FineAmount
PaymentDate
AmountPaid

Schema

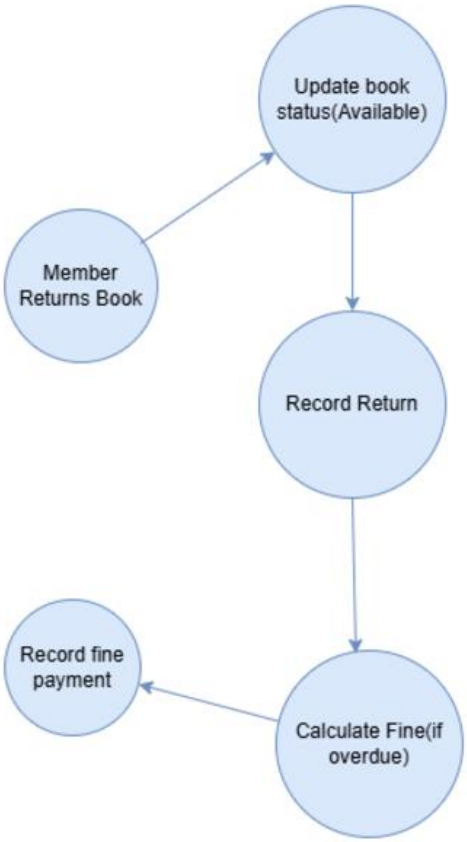

BOOK
PK BOOK_ID
TITLE
AUTHOR
ISBN
STATUS
STOCK
PUBLICATION_ YEAR
FK GENRE_ID
FK SHELF_ID
GENRE
PK GENRE_ID
GENRE_NAME
SHELF
PK SHELF_ID
LOCATION
MEMBER
PK MEMBER_ID
NAME
PHONE
EMAIL
MEMBERSHIP_ DATE
RESERVATION
PK RESERVATION_ ID
STATUS
RESERVATION_ DATE
FK MEMBER_ID
FK BOOK_ID
BORROW
PK BORROW_ID
BORROW_DATE
RETURN_DATE
DUE_DATE
FINE_AMOUNT
FK MEMBER_ID
FK BOOKI_ID
LIBRARIAN
PK LIBRARIAN_ID
NAME
EMAIL
PHONE
HIRE_DATE
FK BRANCH_ID
LIBRARY BRANCH
PK BRANCH ID
NAME
LOCATION
PHONE_NO
TRANSACTION
PK TRANSACTION_ ID
PAYMENT_DATE
AMOUNT_PAID
FK BORROW_ID
M
1

Processes

Book Borrowing Process



Book Returning Process



REPORTS

1. Total Number of Books by Genre

- SQL Query:

```
SELECT Genre.GenreName, COUNT(Book.BookID) AS TotalBooks
FROM Genre
JOIN Book ON Genre.GenreID = Book.GenreID
GROUP BY Genre.GenreName
ORDER BY TotalBooks DESC;
```

- Relational Algebra:

π GenreName, COUNT(BookID) (Genre ⋈ Book.GenreID = Genre.GenreID)
GROUP BY GenreName

Fiction	2
Graphic Novel	1
History	1
Horror	1
Mystery	1
Non-Fiction	1
Poetry	1
Romance	1
Science Fiction	1
Self-Help	1
Thriller	1
Adventure	1
Biography	1
Children's	1
Fantasy	1

2. Members with Borrowed Books:

- SQL Query:

```
SELECT Member.Name, Member.Email, COUNT(Borrow.BorrowID) AS BorrowedBooks
FROM Member
JOIN Borrow ON Member.MemberID = Borrow.MemberID
GROUP BY Member.Name, Member.Email
HAVING COUNT(Borrow.BorrowID) > 0;
```

- Relational Algebra:

π Name, Email, COUNT(BorrowID) (Member ⋈ Borrow.MemberID = Member.MemberID)

GROUP BY Name, Email

HAVING COUNT(BorrowID) > 0

John Smith	john.smith@example.com	1
Alice Johnson	alice.johnson@example.com	1
Bob Brown	bob.brown@example.com	1
Emma Davis	emma.davis@example.com	1
Chris Wilson	chris.wilson@example.com	1
Olivia Taylor	olivia.taylor@example.com	1
Liam Anderson	liam.anderson@example.com	1
Sophia Martinez	sophia.martinez@example.com	1
Ethan Garcia	ethan.garcia@example.com	1
Mia Robinson	mia.robinson@example.com	1
Jacob Clark	jacob.clark@example.com	1
Isabella Lewis	isabella.lewis@example.com	1
James Walker	james.walker@example.com	1
Charlotte Young	charlotte.young@example.com	1
Benjamin Hall	benjamin.hall@example.com	1
dalal	dalal@gamal.com	1
abosara	saraaly@bestie.com	5

3. Books Borrowed and Their Return Status

- SQL Query:

```
SELECT Book.Title, Borrow.BorrowDate, Borrow.DueDate, Borrow.ReturnDate
FROM Book
JOIN Borrow ON Book.BookID = Borrow.BookID
WHERE Borrow.ReturnDate IS NULL;
```

- Relational Algebra:

π Title, BorrowDate, DueDate, ReturnDate

(Book ⋈ Borrow.BookID = Book.BookID)

σ ReturnDate IS NULL

The Great Adventure	1/1/2024	1/15/2024	NULL
Mystery of the Mind	1/5/2024	1/20/2024	NULL
Fantasy World	1/10/2024	1/25/2024	NULL
Science Quest	1/15/2024	1/30/2024	NULL
Historical Tales	1/20/2024	2/5/2024	NULL
Romantic Evenings	1/25/2024	2/10/2024	NULL
Thrilling Heights	2/1/2024	2/15/2024	NULL
Haunted Stories	2/5/2024	2/20/2024	NULL
Child's Play	2/10/2024	2/25/2024	NULL
Poetry Bliss	2/15/2024	3/1/2024	NULL
Adventurous Tales	2/20/2024	3/5/2024	NULL
Life Guide	2/25/2024	3/10/2024	NULL
Graphic Universe	3/1/2024	3/15/2024	NULL
Sunny Days	3/5/2024	3/20/2024	NULL
Green Fields	3/10/2024	3/25/2024	NULL
Life Guide	12/22/2024	1/5/2025	NULL
Mystery of the Mind	12/25/2024	1/8/2025	NULL
The Great Adventure	12/25/2024	1/8/2025	NULL

4. List of Books Available for Borrowing

- SQL Query:

```
SELECT Title, Author, Stock
FROM Book
WHERE Stock > 0;
```

- Relational Algebra:

π Title, Author, Stock (σ Stock > 0 (Book))

The Great Adventure	John Doe	11
Mystery of the Mind	Jane Smith	5
Fantasy World	Alice Wonderland	8
Science Quest	Albert Newton	7
Historical Tales	Henry History	4
Romantic Evenings	Emily Heart	9
Thrilling Heights	Thriller Writer	6
Haunted Stories	Ghost Writer	5
Child's Play	Toy Maker	12
Poetry Bliss	Poet Soul	15
Adventurous Tales	Daring Explorer	11
Life Guide	Wise Guru	9
Graphic Universe	Artist Mind	14
Sunny Days	Happy Soul	13
Green Fields	Nature Lover	7
sa3et el shokhakh	SARA ALY	1000000000

5. Total Fines Collected by Branch

- SQL Query:

```
SELECT LibraryBranch.Name, SUM(Borrow.FineAmount) AS TotalFines
FROM LibraryBranch
JOIN Member ON LibraryBranch.BranchID = Member.BranchID
JOIN Borrow ON Member.MemberID = Borrow.MemberID
```

GROUP BY LibraryBranch.Name;

- **Relational Algebra:**

π Name, SUM(FineAmount)

(LibraryBranch ⋈ Member.BranchID = LibraryBranch.BranchID ⋈ Borrow.MemberID = Member.MemberID)

GROUP BY Name

Central Branch	NULL
Eastside Branch	NULL
Greenfield Branch	NULL
Hilltop Branch	NULL
Lakeside Branch	NULL
Main Branch	0
Meadow Branch	NULL
New City Branch	NULL
North Branch	NULL
Old Town Branch	NULL
Riverside Branch	NULL
South Branch	NULL
Sunnyvale Branch	NULL
Valley Branch	NULL
Westside Branch	NULL

6. Members Who Have Not Borrowed Any Books

- **SQL Query:**

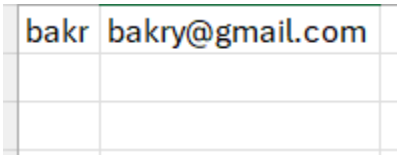
SELECT Name, Email

FROM Member

WHERE MemberID NOT IN (SELECT MemberID FROM Borrow);

- **Relational Algebra:**

π Name, Email (Member) − π MemberID (Borrow)



bakr	bakry@gmail.com

7. Books Borrowed by Genre

- **SQL Query:**

```
SELECT Genre.GenreName, COUNT(Borrow.BorrowID) AS BorrowedBooks
FROM Genre
JOIN Book ON Genre.GenreID = Book.GenreID
JOIN Borrow ON Book.BookID = Borrow.BookID
GROUP BY Genre.GenreName;
```

- **Relational Algebra:**

π GenreName, COUNT(BorrowID)

(Genre ⋈ Book.GenreID = Genre.GenreID ⋈ Borrow.BookID = Book.BookID)

GROUP BY GenreName

Adventure	1
Biography	1
Children's	2
Fantasy	1
Fiction	4
Graphic Novel	1
History	1
Horror	1
Mystery	1
Non-Fiction	3
Poetry	1
Romance	1
Science Fiction	1
Self-Help	1
Thriller	1

8. Overdue Books:

- **SQL Query:**

```
SELECT Book.Title, Borrow.BorrowDate, Borrow.DueDate, DATEDIFF(DAY, Borrow.DueDate, GETDATE()) AS OverdueDays
FROM Book
JOIN Borrow ON Book.BookID = Borrow.BookID
WHERE Borrow.DueDate < GETDATE() AND Borrow.ReturnDate IS NULL;
```

- **Relational Algebra:**

π Title, BorrowDate, DueDate, OverdueDays

(Book ⋈ Borrow.BookID = Book.BookID)

σ DueDate < TODAY AND ReturnDate IS NULL

The Great Adventure	1/1/2024	1/15/2024	345
Mystery of the Mind	1/5/2024	1/20/2024	340
Fantasy World	1/10/2024	1/25/2024	335
Science Quest	1/15/2024	1/30/2024	330
Historical Tales	1/20/2024	2/5/2024	324
Romantic Evenings	1/25/2024	2/10/2024	319
Thrilling Heights	2/1/2024	2/15/2024	314
Haunted Stories	2/5/2024	2/20/2024	309
Child's Play	2/10/2024	2/25/2024	304
Poetry Bliss	2/15/2024	3/1/2024	299
Adventurous Tales	2/20/2024	3/5/2024	295
Life Guide	2/25/2024	3/10/2024	290
Graphic Universe	3/1/2024	3/15/2024	285
Sunny Days	3/5/2024	3/20/2024	280
Green Fields	3/10/2024	3/25/2024	275

Data Dictionary

BOOK Table:

- **BOOK_ID**: Primary Key (*Number*) , Unique identifier for each book in the library.
- **TITLE**: (*Varchar2*),Title of the book.
- **AUTHOR**: (*Varchar2*) ,Name of the author.
- **ISBN**: (*Varchar2*) ,International Standard Book Number for unique book identification.
- **GENRE_ID**: Foreign Key (*Number*) ,References the GENRE table to categorize books.
- **SHELF_ID**: Foreign Key (*Number*), References the SHELF table for the book's location.
- **PUBLICATION_YEAR**: (*Number*) ,Year the book was published.
- **STOCK**: (*Number*) ,Number of copies available.
- **STATUS**: (*Varchar2*),Indicates if the book is "Available" or "Borrowed."

Column Name	Data Type	Nullable	Data Default	Column ID	Comment
BOOK_ID	NUMBER(30,0)	No	Null	1	Unique identifier for book
TITLE	VARCHAR2(100)	Yes	null	2	Title of the book
AUTHOR	VARCHAR2(100)	Yes	null	3	Author's name
ISBN	VARCHAR2(20)	Yes	null	4	ISBN identifier
GENRE_ID	NUMBER(10,0)	Yes	null	5	Foreign key to GENRE
SHELF_ID	NUMBER(10,0)	Yes	null	6	Foreign key to SHELF
PUBLICATION_YEAR	NUMBER(4,0)	Yes	null	7	Year of publication
STOCK	NUMBER(5,0)	Yes	null	8	Available stock count
STATUS	VARCHAR2(20)	Yes	Available	9	Book status

GENRE Table:

- **GENRE_ID:** Primary Key (*Number*) ,Unique identifier for each genre.

  GENRE_NAME: (*Varchar2*), Name of the genre (e.g., Fiction, Science, Biography)

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
GENRE_ID	NUMBER(30,0)	No	null	1	Unique genre identifier
GENRE_NAME	VARCHAR2(50)	Yes	null	2	Name of the genre

SHELF Table:

- **SHELF_ID:** Primary Key (*Number*) ,Unique identifier for each shelf.
- **LOCATION: (***Varchar2*) ,Location details of the shelf in the library**.**

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
SHELF_ID	NUMBER(30,0)	No	null	1	Unique shelf identifier
LOCATION	VARCHAR2(50)	Yes	null	2	Shelf location in library

MEMBER Table:

- **MEMBER_ID**: Primary Key (*Number*) ,Unique identifier for each library member.
- **NAME**: (*Varchar2*), Full name of the member.
- **EMAIL**: (*Varchar2*) ,Email address of the member.
- **PHONE**: (*Varchar2*) ,Phone number of the member.
- **MEMBERSHIP_DATE**: (*Date*) ,The date the membership was initiated.
- **BRANCH_ID**: Foreign Key (*Number*) ,References the LIBRARY_BRANCH table for branch association.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
MEMBER_ID	NUMBER(30,0)	No	null	1	Unique member identifier
NAME	VARCHAR2(100)	Yes	null	2	Member's full name
EMAIL	VARCHAR2(100)	Yes	null	3	Email address
PHONE	VARCHAR2(15)	Yes	null	4	Contact number
MEMBERSHIP_DATE	DATE	Yes	null	5	Membership start date
BRANCH_ID	NUMBER(10,0)	Yes	null	6	Foreign key to Branch

BORROW Table:

- **BORROW_ID**: Primary Key (*Number*),Unique identifier for each borrowing transaction.
- **BOOK_ID**: Foreign Key (*Number*) ,References the BOOK table.
- **MEMBER_ID**: Foreign Key (*Number*) ,References the MEMBER table.
- **BORROW_DATE**: (*Date*) ,The date the book was borrowed.
- **DUE_DATE**: (*Date*) ,Due date for returning the book.
- **RETURN_DATE**: (*Date*), Date the book was returned.
- **FINE_AMOUNT**: (Number),Penalty amount if the book was returned late.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
BORROW_ID	NUMBER(30,0)	No	null	1	Unique borrowing record ID
BOOK_ID	NUMBER(30,0)	Yes	null	2	Foreign key to BOOK
MEMBER_ID	NUMBER(30,0)	Yes	null	3	Foreign key to MEMBER
BORROW_DATE	DATE	Yes	null	4	Date of borrowing
DUE_DATE	DATE	Yes	null	5	Due date for return
RETURN_DATE	DATE	Yes	null	6	Date of return
FINE_AMOUNT	NUMBER(10,2)	Yes	null	7	Penalty for late return

RESERVATION Table:

- **RESERVATION_ID**: Primary Key (*Number*) ,Unique identifier for each reservation.
- **MEMBER_ID**: Foreign Key (*Number*) ,References the MEMBER table.
- **BOOK_ID**: Foreign Key (*Number*) ,References the BOOK table.
- **RESERVATION_DATE**: (*Date*) ,Date the reservation was made.
- **STATUS**: (*Varchar2*) ,Current status of the reservation (Pending, Approved, or Cancelled

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
RESERVATION_ID	NUMBER(30,0)	No	null	1	Unique reservation ID
MEMBER_ID	NUMBER(30,0)	Yes	null	2	Foreign key to MEMBER
BOOK_ID	NUMBER(30,0)	Yes	null	3	Foreign key to BOOK
RESERVATION_DATE	DATE	Yes	null	4	Date of reservation
STATUS	VARCHAR2(20)	Yes	Pending	5	Status of reservation

LIBRARY_BRANCH Table:

- **BRANCH_ID**: Primary Key (*Number*) ,Unique identifier for each library branch.
- **NAME**: (*Varchar2*),Name of the branch.
- **LOCATION**: (*Varchar2*) ,Address or location details of the branch.
- **PHONE**: (*Varchar2*) ,Contact number of the branch.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
BRANCH_ID	NUMBER(30,0)	No	null	1	Unique branch identifier
NAME	VARCHAR2(50)	Yes	null	2	Branch name
LOCATION	VARCHAR2(100)	Yes	null	3	Branch location
PHONE	VARCHAR2(15)	Yes	null	4	Contact number of branch

LIBRARIAN Table:

- **LIBRARIAN_ID**: Primary Key (*Number*) ,Unique identifier for each librarian.
- **NAME**: (*Varchar2*) ,Name of the librarian.
- **EMAIL**: (*Varchar2*) ,Email address of the librarian.
- **PHONE**: (*Varchar2*) ,Contact number of the librarian.
- **HIRE_DATE**: (*Date*) ,The date the librarian was hired.
- **BRANCH_ID**: Foreign Key (*Number*) ,References the LIBRARY_BRANCH table.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
LIBRARIAN_ID	NUMBER(30,0)	No	null	1	Unique librarian ID
NAME	VARCHAR2(50)	Yes	null	2	Librarian's full name
EMAIL	VARCHAR2(50)	Yes	null	3	Email address
PHONE	VARCHAR2(15)	Yes	null	4	Contact number
HIRE_DATE	DATE	Yes	null	5	Date of hire
BRANCH_ID	NUMBER(30,0)	Yes	null	6	Foreign key to Branch

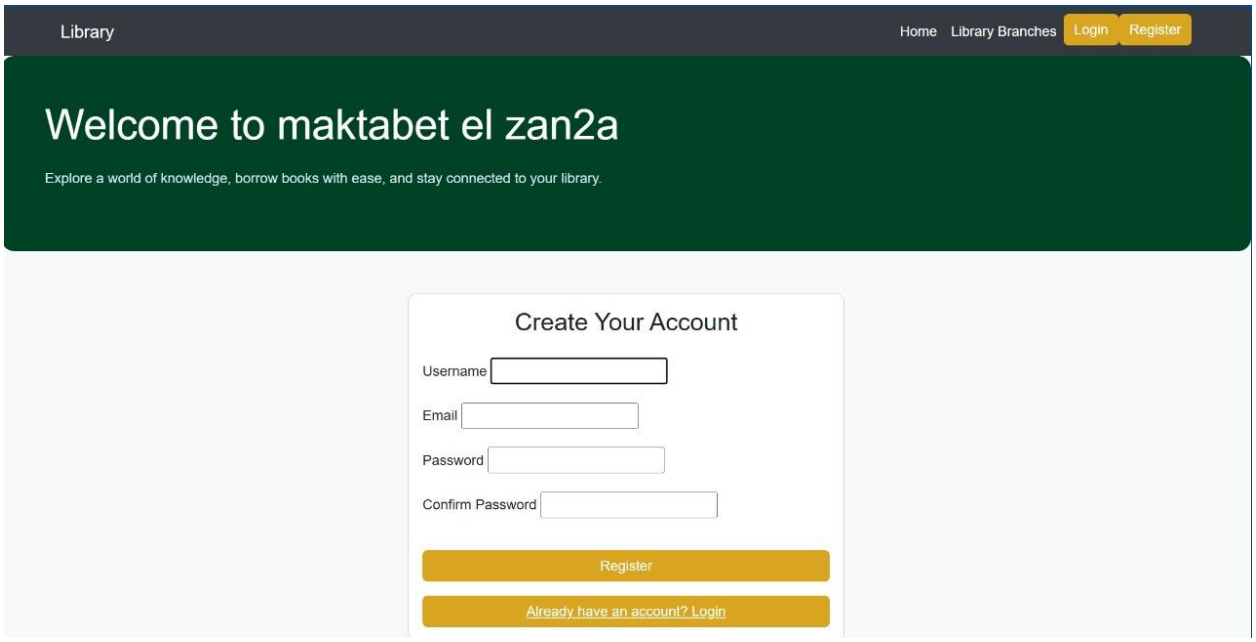
TRANSACTION Table:

- **TRANSACTION_ID**: Primary Key (*Number*) ,Unique identifier for each transaction.
- **BORROW_ID**: Foreign Key (*Number*) ,References the BORROW table.
- **PAYMENT_DATE**: (Date) ,The date of payment.
- **AMOUNT_PAID**: (Number) ,The amount paid during the transaction.

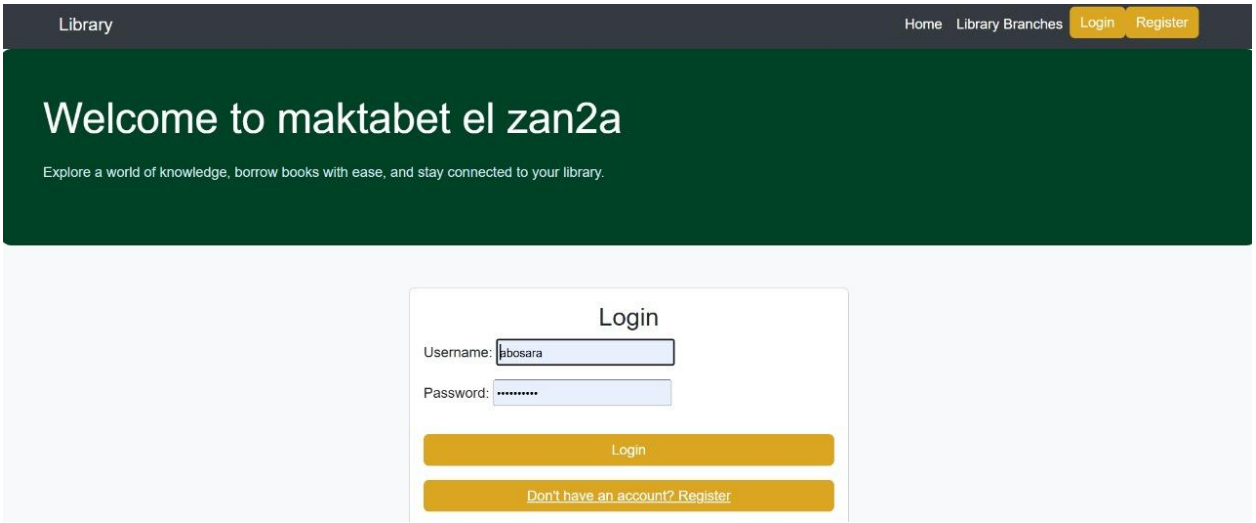
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
TRANSACTION_ID	NUMBER(30,0)	No	null	1	Unique transaction ID
BORROW_ID	NUMBER(30,0)	Yes	null	2	Foreign key to BORROW
PAYMENT_DATE	DATE	Yes	null	3	Date of payment
AMOUNT_PAID	NUMBER(10,2)	Yes	null	4	Transaction amount paid

GUIs of Library Website

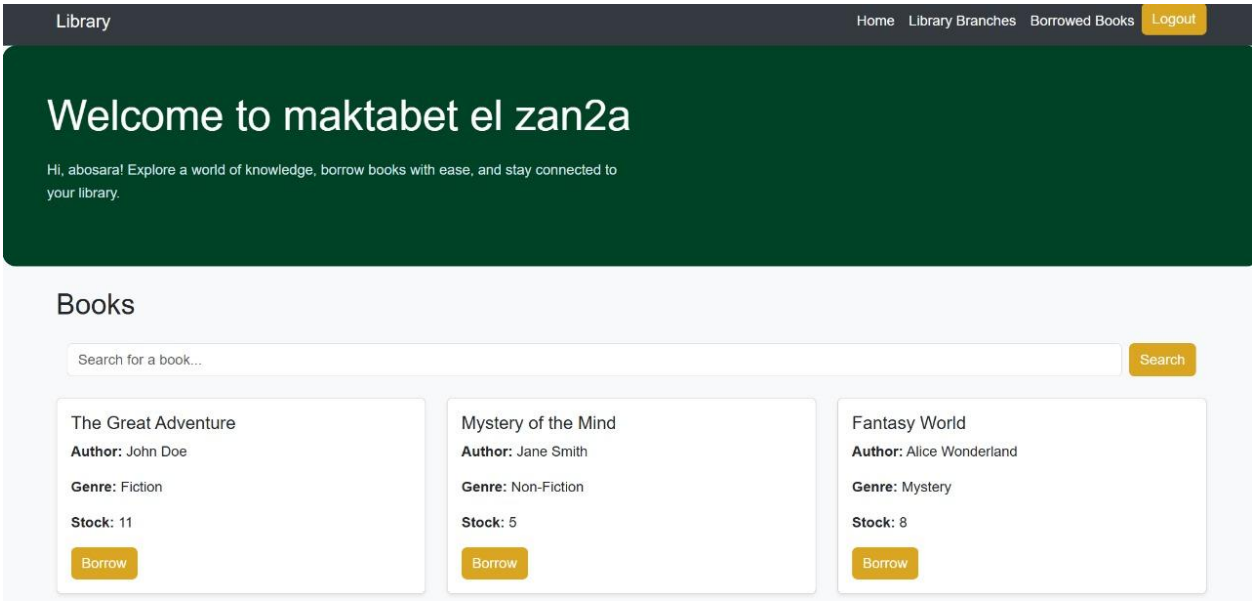
1.Register page:



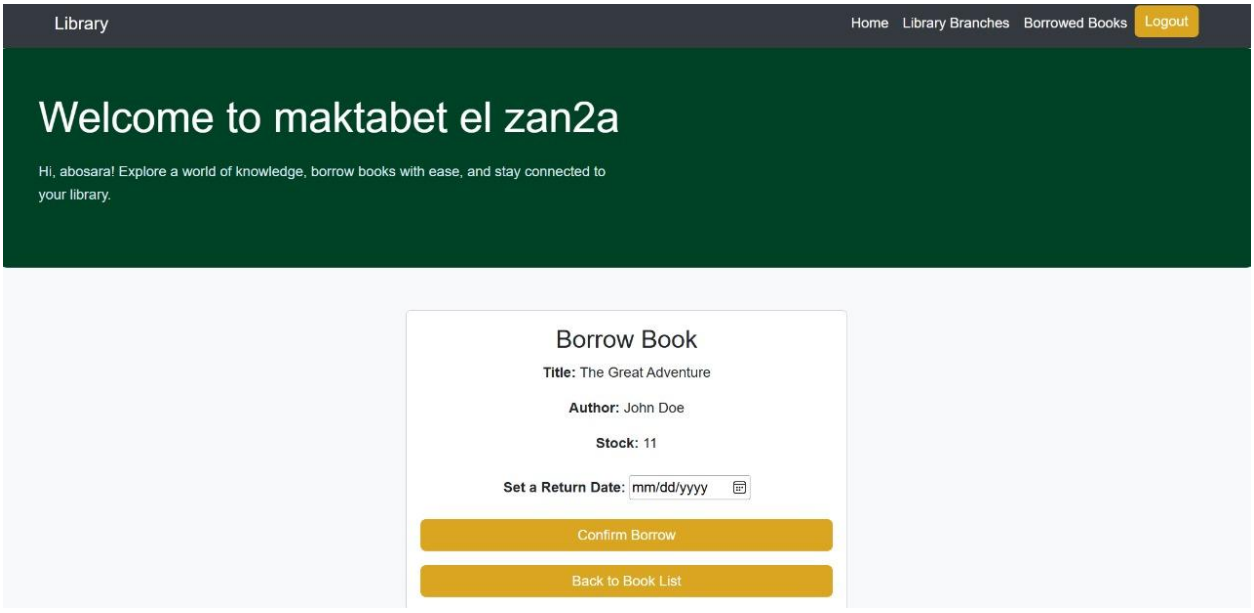
2.Login Page:



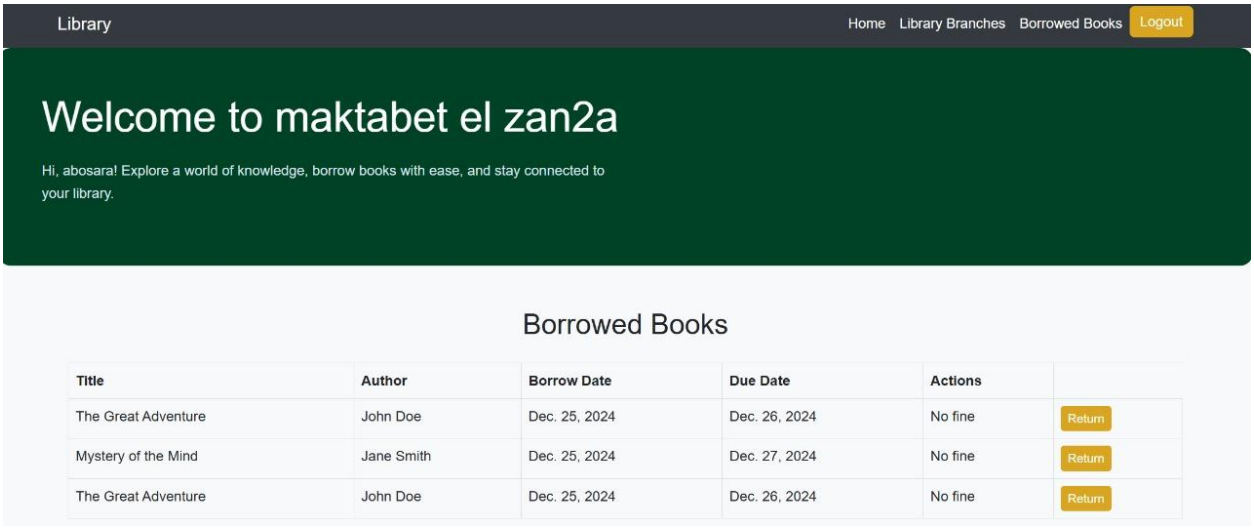
3.Books Page:



4.Borrow Book:



4. History of Borrowed Books:



5. Confirmation on borrowing:

6.Return Book

7.Library Branches Page: