##### Library Management System Report

| **SIGNATURE BLOCK** | | | | |
| --- | --- | --- | --- | --- |
| **Statement** | I did my share of the work, and I have a general understanding of the contents of the assignment. | | | |
| **Team Member** | **Contribution** | **% of Total** | **Signature** | **Date** |
| *Salam, Michael* | *Created pages for UI , assisted in creating database and assisted in creating database queries* | %50 |  |  |
| *Rodriguez, Kemrick* | *Connected the backend to the UI and created the database.* | %50 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

1. Introduction………………………………………………………………………………………….3
2. System Design Criteria…………………………………………………………………………….4

2.1. UI Planning and Designing…………………………………………………………………..5

2.2. Database Schema…………………………………………………………………………….6

1. Functional Components……………………………………………………………………………6

3.1. Members Page ………………………………………………………………………………..6

3.2 Books Page……………………………………………………………………………………..7

3.3 Borrowings Page……………………………………………………………………………….8

3.4 Librarians Page…………………………………………………………………………………9

1. Backend Integration………………………………………………………………………………...10
2. ERD Diagram………………………………………………………………………………………..11

5.1. Functional Dependencies………………………………………………………………………12

5.2. Stored Procedures, Triggers, Functions……………………………………………………….12

5.4 Normalization & Integrity Constraints…………………………………………………………14

5.5 Challenges Faced………………………………………………………………………………14

1. Conclusion……………………………………………………………………………………………15
2. References……………………………………………………………………………………………15

**Introduction**

The Library Management System project was developed to provide a well-structured , organized and fully functional system for managing library operations.Traditional library processes often face challenges such as manual record-keeping , data inconsistency , and difficulty in tracking the borrowing and returning of books.This project addresses these issues by combining a modern user interface with a structured relational database.

The frontend of the Library Management System is designed with the use of HTML , CSS , and Bootstrap, ensuring that the front end is responsive, visually consistent and user friendly. The backend is constructed and powered by PostgreSQL, with normalized tables for categories, members, authors, books, librarians, and borrowings, each linked through appropriate constraints and relationships to maintain data integrity.

Overall, the system emphasizes usability , consistency , and reliability. It provides a scalable solution for library management, ensuring that both staff and users can interact with the system efficiently.

**2.System Design Criteria**

The Design of the Library management system was guided by the key criteria to ensure the system is capable of efficiency and reliability.These criteria included the usability , reliability, and consistency, which helped to construct the front-end and back-end database.

**2.1 User Interface Design**

The user interface was developed using HTML , CSS , and Bootstrap ,the frameworks chosen to ensure the responsiveness and user-friendly design for the system. These frameworks help to maintain a more modern and consistent appearance.

A unified color scheme, navigation bar, and table layouts were applied across all pages to provide clarity and usability for the users. This consistency ensures that users can easily navigate through the UI , such as navigating through Members, Books, Borrowings, and Librarians without confusion.

To enhance the accessibility and usability of the UI , features which include the search functionality and “Add” buttons were included. These elements improve navigation and allow users to locate information or perform actions , making the interface intuitive and efficient.

**2.2 Database schema**

The backend database was built in PostgreSQL to provide a more reliable and structured foundation for the Library Management System. The schema was designed to ensure normalization , data integrity and scalability.

The Database consists of six core tables: Categories, Members, Authors , Books , Librarians , and Borrowings. Each of the tables was created to represent a distinct entity with the library system.

* Categories : Stores books based on the theme of the book, for example if the book is based on coding or programming then it is included in the Technology Category.
* Members: Holds the user’s information such as name , email, contacts , and membership date.
* Authors: Contains the author’s name and nationalities.
* Books: Records the titles of the books , ISBN , published year , copies available , and links to categories and authors table.
* Librarians: Stores library staff information such as name , contact information , and hire date.
* Borrowings: Keeps track of borrowing transactions , including member, book , borrow date ,due date, return date , and status.

Relationships were established with the use of foreign keys to maintain data integrity:

* Books are linked to Categories and Authors.
* Borrowings are linked to Members and Books.

This ensures that each borrowing record references valid members and books, and each book belongs to a defined category and author.

Constraints were also applied to the tables so that it enforces uniqueness and prevents invalid data:

* Primary Keys ensure each record is unique.
* Unique constraints such as ISBN and Email prevent duplication.
* Foreign keys enforce relationships between tables.

Overall the database schema provides a normalized and reliable structure that supports the functionality of the Library Management System while also ensuring data consistency and scalability

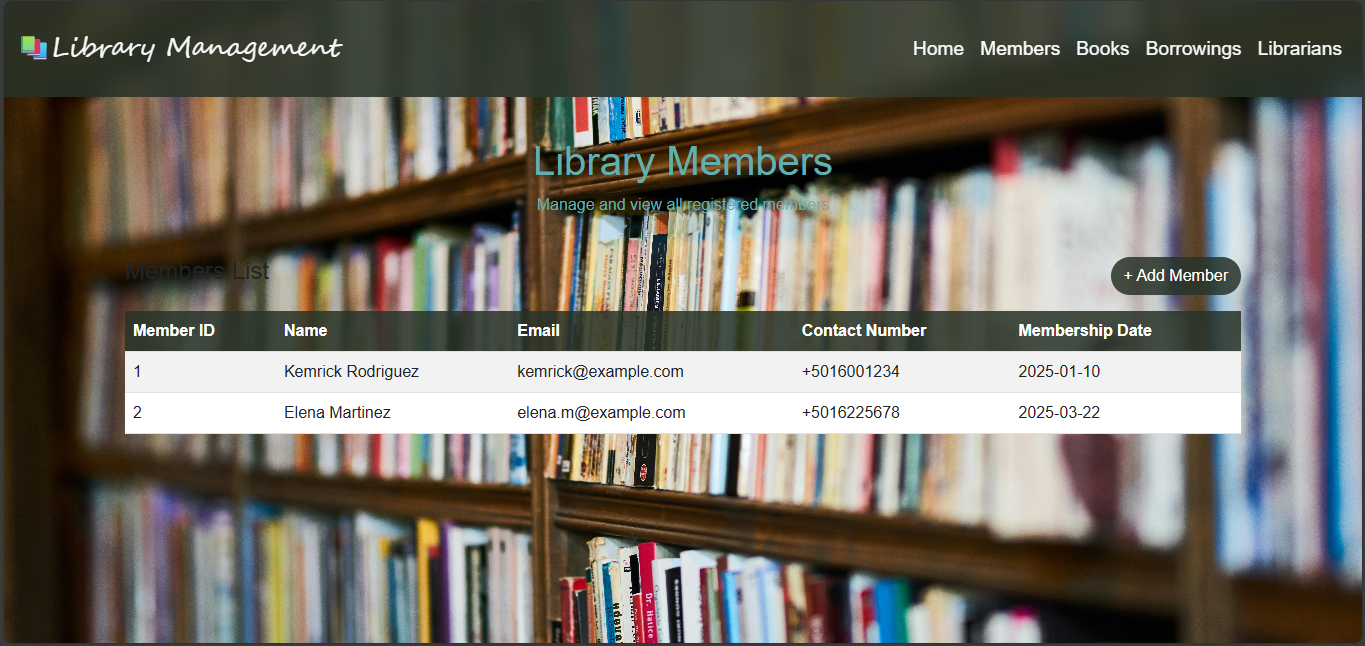
**3 Functional Components**

The library management system is composed of four main functional components , each is designed to handle a specific aspect of the library functions. These components are integrated though the user interface and also connected to the back end database to ensure smooth and efficient management.

**3.1 Members Page**

The members pages allow the registration and management of library members. It displays members details such as ID , name , email , contact , and membership date. Features include:

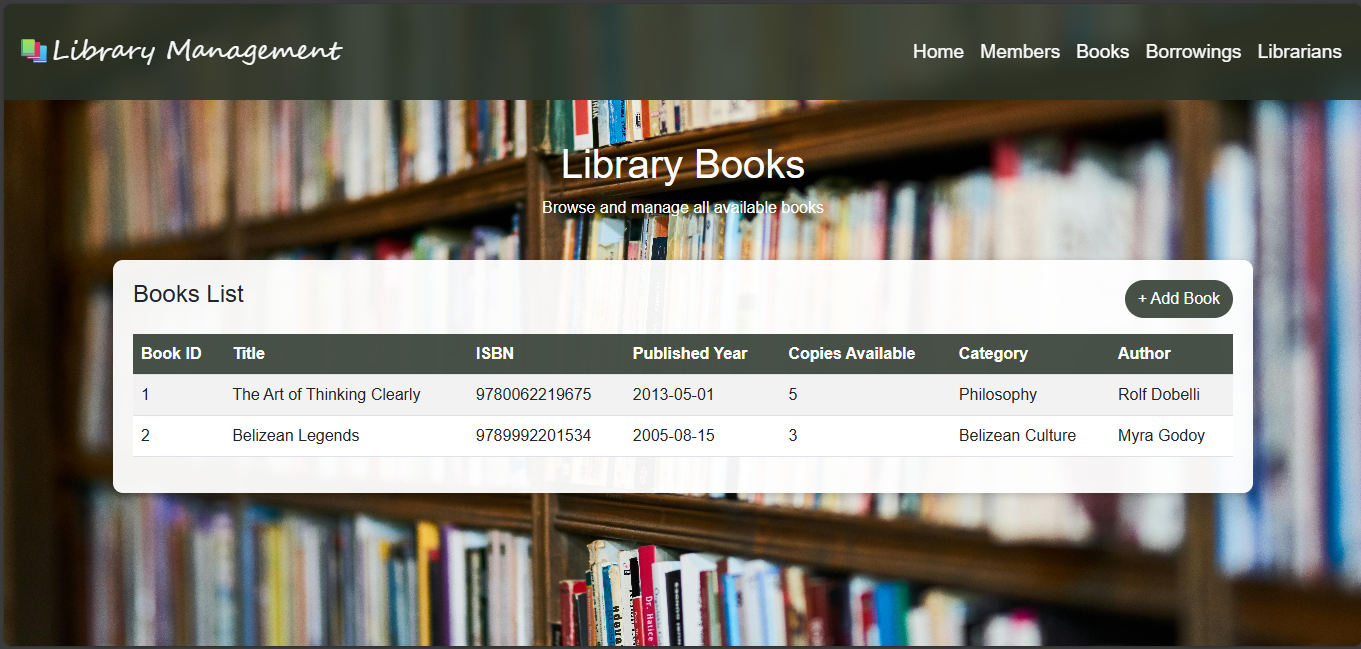
* A searchable table of all registered members in the database.
* An “Add Member” button to register members.
* Consistent layout and navigation for easier use.



**3.2 Books Page**

The Books page provides a catalog of all the books in the library database. Details include the Book ID , Book Title , ISBN , Published Year , Copies Available , Category and Author. Features include:

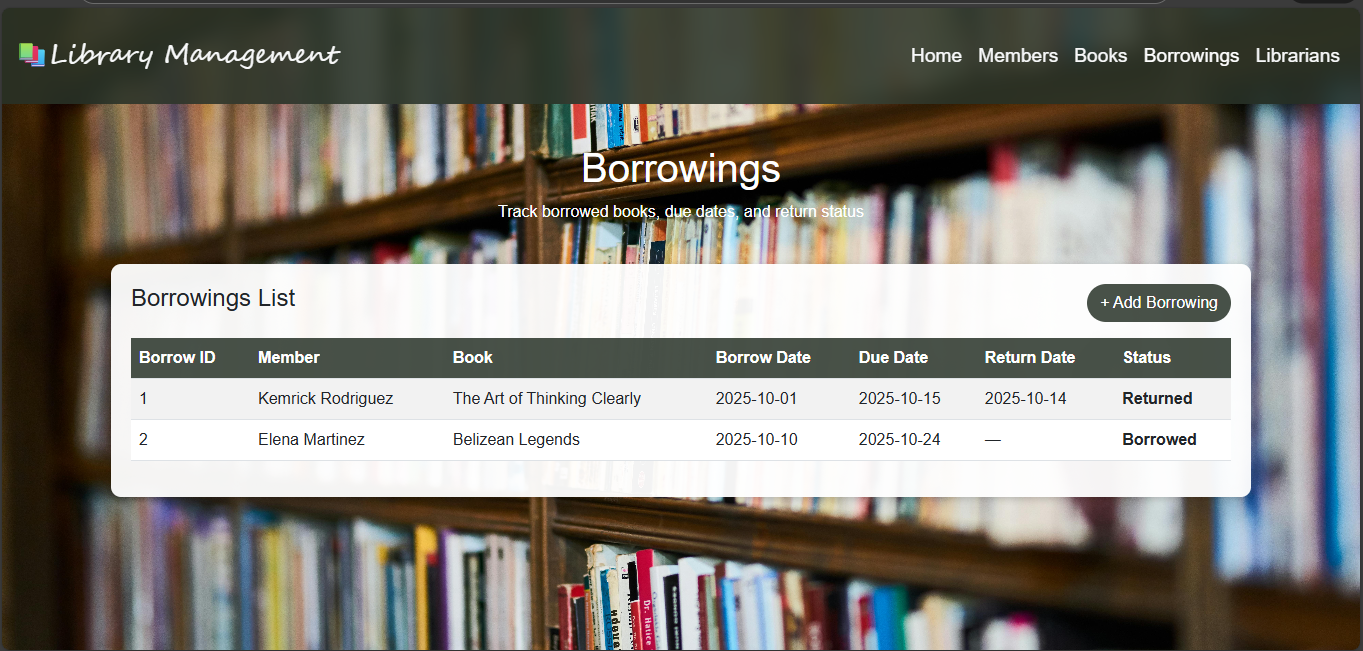
* A Table in the center displaying the book records with the link of category and author information
* An “Add Book” button for adding new titles into the library database.
* Integration with the database prevents duplicate ISBNs and ensures accurate categorizations.



**3.3 Borrowings Page**

The Borrowings page tracks book lending information and transactions. It records the Borrow ID , Member , Book , Borrow Date , Due Date , Return Date and status(Borrowed , Returned , Overdue).Features Include :

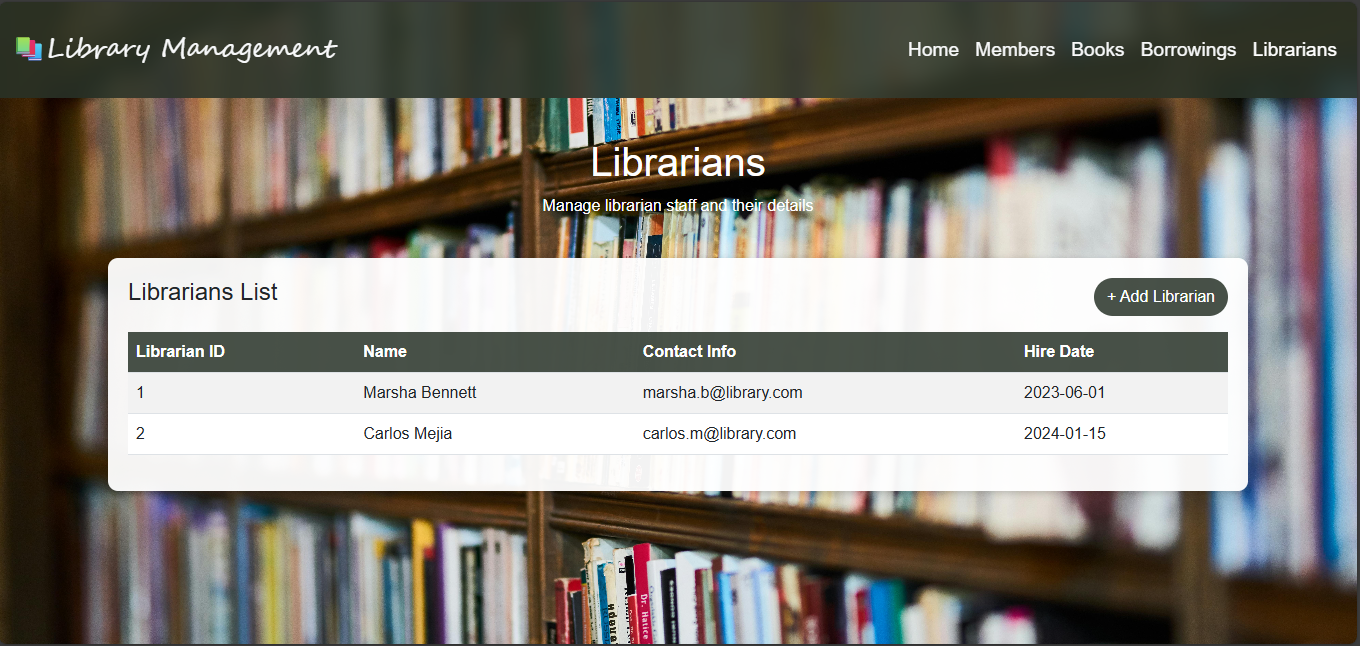
* A table showing the current and past borrowings of books.
* Automatic status updates based on due dates and return dates.
* An “Add Borrowing” button to record new transactions in the library Database.



**3.4 Librarians Page**

The Librarians Page manages the library staff information. It displays the librarian ID , librarian name , contact information , and Hire date. Features Include

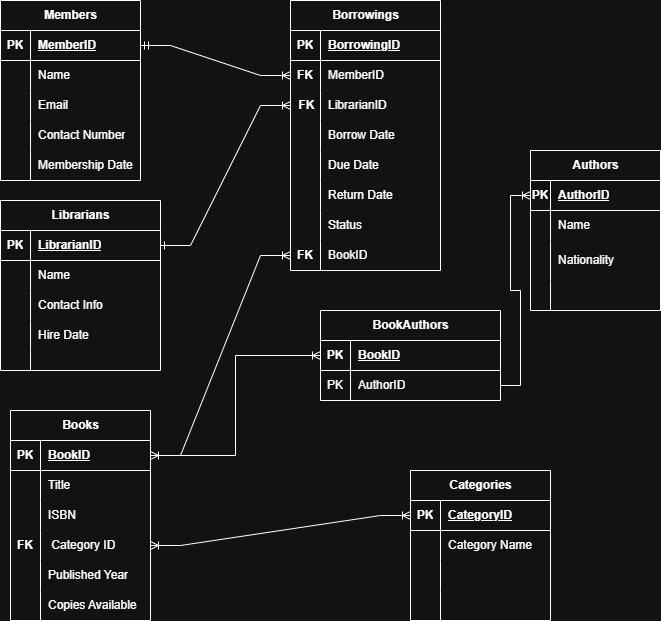
* A table displaying information on library staff.
* An “Add librarian” button to insert new librarian entries.
* Consistent design with the other pages to maintain usability.



Together , all these functional components provide a complete solution for managing library operations, ensuring all members , books , borrowings , and staff are managed efficiently.

**4.Back-End Integration**

**5.ERD Diagram**

****

The ERD diagram illustrates the structure of the Library Management System consists of six entities: Categories, Members, Authors, Books, Librarians, and Borrowings. Relationships are defined through foreign keys: Books are linked to Authors and Categories, while Borrowings are linked to Members and Books,also Books and Authors are linked through a bridge Table. This design ensures normalization and data integrity.

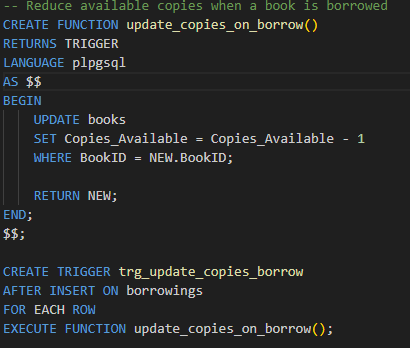
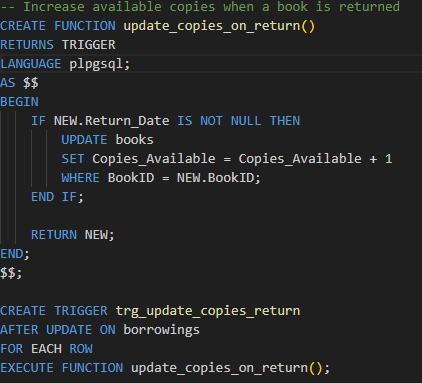
**5.1 Functional Dependencies**

These functional dependencies confirm that each non-key attribute depends only on the primary key, supporting normalization to 3NF.

* Members: MembersID -> Name, Email, Contact\_Number, Membership\_Date
* Books: BookID -> Book\_title, ISBN, Published\_Year, Copies\_Available, CategoryID
* Authors: Author-> Name, Nationality
* Categories: CategoryID -> Category\_Name
* Librarians: LibrarianID ->Name, Contact\_Info, Hire\_date
* Borrowings: BorrowingID ->MemberID, BookID, Borrow\_Date, Due\_Date, Return\_Date, Status

**5.2. Stored Procedures, Triggers, Functions**

To enhance automation and maintain data integrity, stored procedures, triggers, and functions were implemented in the Library Management System. These features reduce manual work, enforce rules automatically, and ensure consistency across the database.

(This trigger reduces the available copies (This trigger increases the available copies when a when a borrowing record is inserted.) borrowing record is updated with a return date.)

**5.3 Normalization & Integrity Constraints**

The database schema was normalized to 3NF (Third Normal Form). Each non-key attribute depends only on the primary key , and there are no transitive dependencies.For example, author nationality is stored in the Authors table rather than the Books table, preventing redundancy. Integrity constraints applied include primary keys, foreign keys, unique constraints (ISBN, Email), and check constraints (valid dates, non-negative copies). Together, these ensure data consistency and reliability.

**5.4 Challenges Faced**

Challenges faced during the making of the Library Management System included without redundancy, ensuring ISBN and Email uniqueness, debugging triggers, connecting Flask routes to PostgreSQL, and maintaining consistent UI across all pages.These challenges were resolved through testing, applying constraints, and refining the schema.Connecting the backend to the frontend was an incredible challenge but with the help of researching and watching youtube videos, the challenge was resolved.

**6. Conclusion**

The Library Management System project successfully integrates a modern user interface with a normalized relational database to provide a complete solution for managing library operations. By combining HTML, CSS, and Bootstrap for the frontend with PostgreSQL for the backend, the system achieves both usability and reliability.

The design criteria of consistency, usability , and scalability guided the development of the user interface and database schema, ensuring that members, books, borrowings, and librarians are managed efficiently. Functional components such as searchable tables, “Add” buttons, and automatic status updates enhance accessibility and reduce manual effort.

Backend integration through allows dynamic rendering of data, while stored procedures, triggers, and functions automate tasks such as updating book availability and calculating overdue days. The ERD and functional dependencies confirm that the database is normalized to Third Normal Form(3NF) with integrity constraints applied to maintain data accuracy and consistency.

Challenges faced during development, enforcing uniqueness, and ensuring smooth frontend‑backend communication, were resolved through iterative testing and refinement. The final system demonstrates how concepts of the relational model can be applied in practice to deliver a scalable, reliable, and user‑friendly application.

Overall, the Library Management System provides a structured and efficient platform for library management, showcasing the successful application of database design principles and modern web technologies.

**7. REFERENCES**

<https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css>

<https://github.com/MikaelSalam/Library-Management-Project/blob/main/Library%20Management%20Database.sql>