

LIBRARY MANAGEMENT

SYSTEM

Mini project of Database Systems Lab (CSE 2262)

Submitted By:

Name: Moksha Deepak Kothari

Roll No: 4

Reg No: 210905017

Section: A

Department of Computer Science and Engineering

Manipal Institute of Technology, Manipal.

April 2023



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Manipal

09/05/2023

CERTIFICATE

This is to certify that the project titled Library Management System is a record of the bonafide work done by Moksha Deepak Kothari (Reg. No. 210905017) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech.) in COMPUTER SCIENCE & ENGINEERING of Manipal Institute of Technology, Manipal, Karnataka, (A Constituent Institute of Manipal Academy of Higher Education), during the academic year 2022-2023.

Name and Signature of Examiners:

1. Dr. Anup Bhat, Assistant Professor, CSE Dept.
2. Mr. Gowardhan Hegde, Assistant Professor Selection Grade, CSE Dept.

TABLE OF CONTENTS

Introduction………………………………………………………………….4

Abstract………………………………………………………………………..4

Problem Statement & Objectives……………………………………4

ER Diagram…………………………………………………………………...5

Relational Schema…………………………………………………………6

DDL Commands…………………………………………………………….6

Constraints……………………………………………………………………7

Methodology…………………………………………………………………8

Views……………………………………………………………………………9

Triggers………………………………………………………………………..9

Queries…………………………………………………………………….…10

Database connectivity….……………………………………………...11

UI Design (With sample data displayed) ……………………....12

Conclusion…………………………………………………………………..16

Limitations and Future Work……………………………………….16

References…………………………………………………………………..17

Introduction

The work of a Librarian can be quite tedious. They need to keep a track of all books the library holds, the publishers and authors of the books and all the borrowings and returns of books that occur daily. The accuracy of a librarian in record keeping is crucial for the proper management of books. Doing so manually can be inefficient, inaccurate, and inconsistent. The computerisation of data and record keeping can make the tracking and management of books in the library more fluid. It removes all the manual errors and reduces the probability of finding incompatible records. It also eliminates the issue of storing piles of record books and going through each page to look for a record.

Abstract

A certain library has several branches and wants to computerize its data. Multiple borrowers borrow books from the libraries. Each borrower can borrow several books at a time. A record of all these borrowings needs to be kept. A record of the authors and publishers of the books also needs to be kept.

An author can write several books but a book can only have one author. A publisher can publish several books, but a book can only be published by one publisher.

This project aims to create a database to store the data pertaining to the books as well as the borrowers of all the libraries. Along with this, overdues also need to be tracked and penalized accordingly.

Problem Statement & Objectives

Design and implement a book management service for a library with several branches. Include information about authors and publishers who have published the books the libraries carry along with information about books carried by each library. The database should be able to record information about the borrowers borrowing the books. The books should have a due date of return, and if not returned by then, the librarian should be able to see the amount the borrower owes for the overdue. The library charges two rupees per day of delay. Allow the user to see how many copies of a book each branch owns and how many copies are currently in the branch. The librarians should be able to add new books coming into the libraries.

For the branches, include information such as branch id, branch name and branch address.

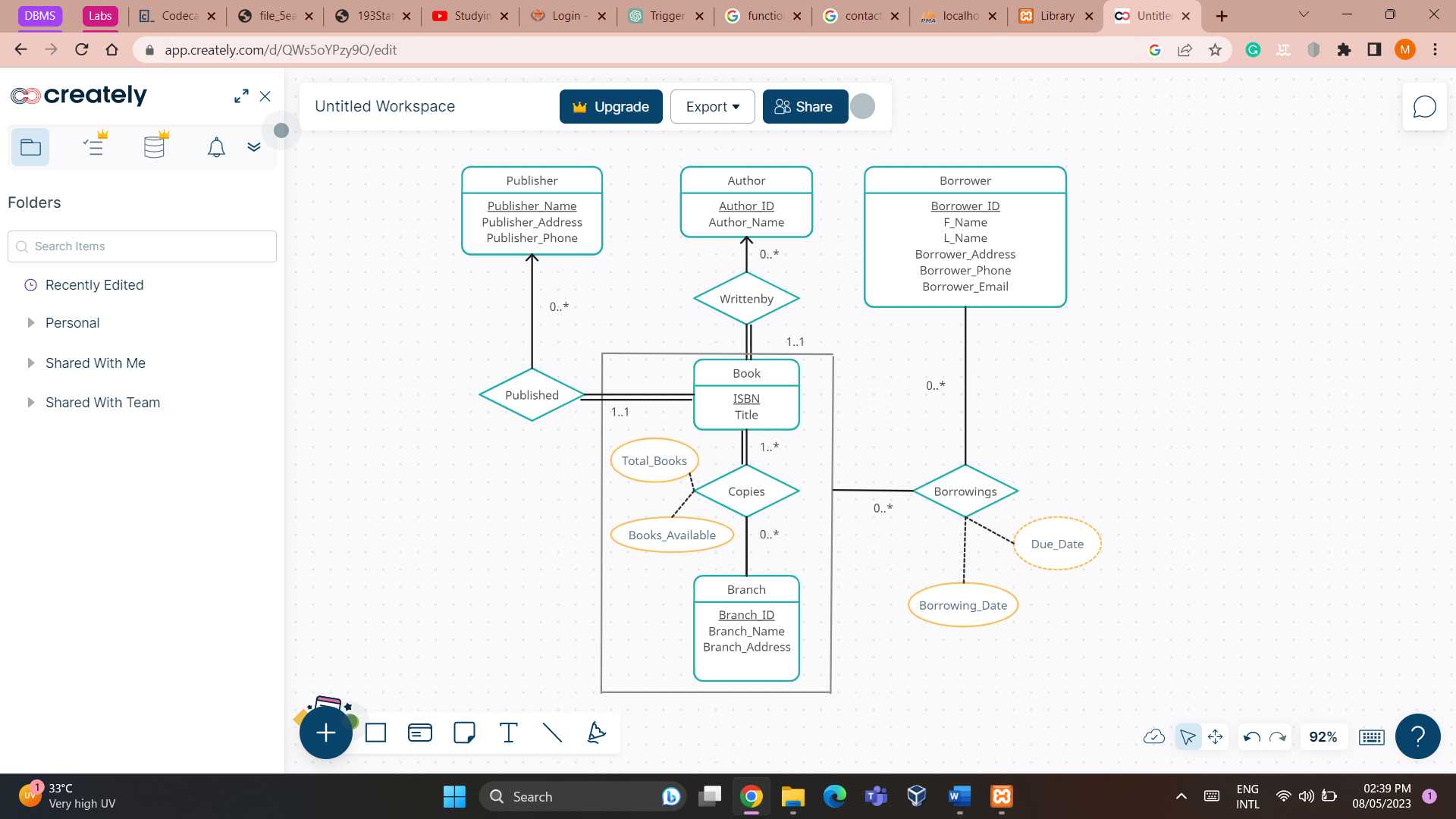
For the publishers and authors, include their name, address, and contact details.

For the borrowers include information such as their id, name address and contact details.

For the borrowings, include information such as the borrower, date of borrowing, due date and ISBN of book borrowed.

For the books, include information such as their title, author, publisher, and copies owned by each branch.

Entity-Relationship Diagram



Relational Schema

A computer screen shot of a computer

Description automatically generated with low confidence

DDL Commands

create table publisher( Publisher\_Name varchar(30) primary key, Publisher\_Address varchar(100) not null, Publisher\_Phone numeric(10) not null);

create table author( Author\_ID numeric(5) primary key, Author\_Name varchar(100) not null);

create table book( ISBN numeric(13) primary key, Title varchar(30) not null, Publisher\_Name varchar(30) not null);

[create](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [table](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) writtenby(ISBN numeric(13), Author\_ID numeric(5), primary key(ISBN, Author\_ID));

create table borrower(Borrower\_ID numeric(5) primary key, F\_Name varchar(20) not null, L\_Name varchar(20) not null, Borrower\_Address varchar(100), Borrower\_Phone numeric(10) not null, Borrower\_Email varchar(20));

create table branch(Branch\_ID varchar(5) primary key, Branch\_Name varchar(20) not null, Branch\_Address varchar(20) not null);

create table copies(Branch\_ID varchar(5), ISBN numeric(13), Total\_Copies numeric(3), Copies\_Available numeric(3) ,primary key(Branch\_ID, ISBN));

create table borrowings(Borrower\_ID numeric(5), ISBN numeric(13), Branch\_ID varchar(5), Borrowing\_Date date, Due\_Date date, primary key(Borrower\_ID, ISBN, Branch\_ID));

Integrity Constraints

alter table book add constraint pubnname\_fk foreign key(Publisher\_Name) references publisher(Publisher\_Name);

alter table writtenby add constraint isbn\_fk foreign key(ISBN) references book(ISBN);

alter table writtenby add constraint aut\_fk foreign key(Author\_ID) references author(Author\_ID);

alter table copies add constraint cop\_isbn\_fk foreign key(ISBN) references book(ISBN);

alter table copies add constraint cop\_branch\_fk foreign key(Branch\_ID) references branch(Branch\_ID);

alter table borrowings add constraint bor\_isbn\_fk foreign key(ISBN) references book(ISBN);

alter table borrowings add constraint bor\_id\_fk foreign key(Borrower\_ID) references borrower(Borrower\_ID);

alter table borrowings add constraint bor\_branch\_fk foreign key(Branch\_ID) references branch(Branch\_ID);

alter table publisher add constraint pub\_phone\_check check (length(Publisher\_Phone)=10);

alter table borrower add constraint bor\_phone\_check check (length(Borrower\_Phone)=10);

alter table book add constraint isbn\_check check(length(ISBN)=13);

**Participation Constraints**

1 book must have exactly 1 publisher, but a publisher may have published multiple books. The library may also not hold any books from the publisher.

1 book must have exactly one author, but an author may have authored multiple books. The library may also not hold any books from the author

A book may be at many branches, a branch can have many books, a branch may not have a particular book.

A borrower can borrow as many books as they would like (provided they do not borrow multiple copies of the same book), a book can be borrowed by multiple borrowers, and a book may not be borrowed by any borrower.

A borrower may not have any borrowings at a given point in time.

A borrower can only borrow 1 copy of a book from any of the branches.

**Methodology**

The tables that can be derived from the entity sets of the ER Diagram are:

* Publisher(Publisher\_Name, Publisher\_Address, Publisher\_Phone)
* Author(Author\_Id, Author\_Name)
* Borrower(Borrower\_ID, F\_Name, L\_Name, Borrower\_Address, Borrower\_Phone, Borrower\_Email)
* Book(ISBN, Title)
* Branch(Branch\_ID, Branch\_Name, Branch\_Address)

The tables that can be derived from relationship sets are:

* Writtenby(Author\_ID, ISBN)

The current constraint is that the library contains books, such as novels, which have one author. However, the library might decide to add books such as textbooks in the future which can have multiple authors. Separating the two tables in the future would be tedious, hence it is taken as a separate schema.

* Copies(Branch\_ID, ISBN, Total\_Copies, Copies\_Available)

Copies is a many-to-many relationship between book and branch, hence a separate schema is created for it.

* Borrowings(Borrower\_ID, ISBN, Branch\_ID, Borrowing\_Date, Due\_Date)

Borrowings is a relation between the copies relation and borrower table. It is taken as a relationship between the aggregation of books and branch tables and the borrower table. Since it is a many-to-many relationship, a separate schema is created for it.

Since Published is a one-to-many relationship, it does not require a separate schema. The primary key, Publisher\_Name, can be added to the Book table, so the modified schema of books is Book(ISBN, Title, Publisher\_Name)

The functional dependencies in the database are:

ISBN Title, Publisher\_Name

Publisher\_Name Publisher\_Address, Publisher\_Phone

Author\_ID Author\_Name

Borrower\_ID F\_Name, L\_Name, Borrower\_Address, Borrower\_Phone,

Borrower\_Email

Branch\_ID Branch\_Name, Branch\_Address

ISBN Author\_Name

ISBN, Borrower\_ID Branch\_ID

None of the constraints are violated by the schema. Hence the schema is in Boyce-Codd normal form.

Views

CREATE VIEW borr\_with\_amount AS SELECT Borrower\_ID, ISBN, Branch\_ID, Borrowing\_Date, Due\_Date, GREATEST(DATEDIFF(CURRENT\_DATE,Due\_Date) \* 2,0) AS Amount\_Due FROM borrowings GROUP BY Borrower\_ID, ISBN, Branch\_ID, Borrowing\_Date, Due\_Date;

Triggers

DELIMITER //

CREATE OR REPLACE TRIGGER `borrow` BEFORE INSERT ON `borrowings` FOR EACH ROW

BEGIN

DECLARE n INT;

SELECT Copies\_Available INTO n FROM copies WHERE ISBN = NEW.ISBN AND Branch\_ID = NEW.Branch\_ID;

IF n > 0 THEN

UPDATE copies SET Copies\_Available = Copies\_Available - 1 WHERE ISBN = NEW.ISBN AND Branch\_ID = NEW.Branch\_ID;

END IF;

END//

DELIMITER ;

DELIMITER //

CREATE OR REPLACE TRIGGER `return`

BEFORE DELETE ON `borrowings` FOR EACH ROW

BEGIN

UPDATE copies SET Copies\_Available = Copies\_Available + 1 WHERE ISBN = OLD.ISBN AND Branch\_ID = OLD.Branch\_ID;

END//

DELIMITER ;

DELIMITER //

CREATE OR REPLACE TRIGGER `ret\_date` AFTER INSERT ON `borrowings` FOR EACH ROW

BEGIN

UPDATE borrowings SET NEW.Due\_Date = DATE\_ADD(NEW.Borrowing\_Date, INTERVAL 14 DAY) WHERE Borrower\_ID=NEW.Borrower\_ID and ISBN = NEW.ISBN AND Branch\_ID = NEW.Branch\_ID;

END//

DELIMITER ;

Queries

select \* from copies natural join book;

select \* from author where Author\_ID=$authorid;

select \* from publisher where Publisher\_Name='$publishername';

select Branch\_ID from branch where Branch\_ID=’$branchid’;

insert into book values('$isbn','$title','$publishername');

insert into writtenby values('$isbn','$authorid');

insert into copies values('$branchid','$isbn','$num','$num');

select \* from book where ISBN='$isbn';

select b.isbn, b.title, a.author\_name, p.publisher\_name, c.branch\_id, c.total\_copies, c.copies\_available from book as b,writtenby as w, author as a, publisher as p, copies as c where b.publisher\_name=p.publisher\_name and b.isbn=w.isbn and a.author\_id=w.author\_id and b.isbn=c.isbn and b.isbn='$isbnentr';

select b.isbn, b.title, a.author\_name, p.publisher\_name, c.branch\_id, c.total\_copies, c.copies\_available from book as b,writtenby as w, author as a, publisher as p, copies as c where b.publisher\_name=p.publisher\_name and b.isbn=w.isbn and a.author\_id=w.author\_id and b.isbn=c.isbn and b.title like '%$titleentr%';

select b.isbn, b.title, a.author\_name, p.publisher\_name, c.branch\_id, c.total\_copies, c.copies\_available from book as b,writtenby as w, author as a, publisher as p, copies as c where b.publisher\_name=p.publisher\_name and b.isbn=w.isbn and a.author\_id=w.author\_id and b.isbn=c.isbn and a.author\_name like '%$author%';

select b.isbn, b.title, a.author\_name, p.publisher\_name, c.branch\_id, c.total\_copies, c.copies\_available from book as b,writtenby as w, author as a, publisher as p, copies as c where b.publisher\_name=p.publisher\_name and b.isbn=w.isbn and a.author\_id=w.author\_id and b.isbn=c.isbn and p.publisher\_name like '%$pubentr%';

select \* from borrower;

insert into borrower values($id,'$fame','$lname','$addr',$phone,'$email');

SELECT bor.Borrower\_ID, bor.F\_Name, bor.L\_Name, bor.Borrower\_Address, bor.Borrower\_Phone, bor.Borrower\_Email, b.ISBN, b.Due\_Date, b.Amount\_Due FROM borr\_with\_amount AS b, borrower as bor WHERE b.Borrower\_ID=bor.Borrower\_ID and bor.Borrower\_ID='$borid';

SELECT bor.Borrower\_ID, bor.F\_Name, bor.L\_Name, bor.Borrower\_Address, bor.Borrower\_Phone, bor.Borrower\_Email, b.ISBN, b.Due\_Date, b.Amount\_Due FROM borr\_with\_amount AS b, borrower as bor WHERE b.Borrower\_ID=bor.Borrower\_ID and bor.F\_Name like '%$borfname%';

SELECT bor.Borrower\_ID, bor.F\_Name, bor.L\_Name, bor.Borrower\_Address, bor.Borrower\_Phone, bor.Borrower\_Email, b.ISBN, b.Due\_Date, b.Amount\_Due FROM borr\_with\_amount AS b, borrower as bor WHERE b.Borrower\_ID=bor.Borrower\_ID and bor.L\_Name like '%$borlname%';

SELECT bor.Borrower\_ID, bor.F\_Name, bor.L\_Name, bor.Borrower\_Address, bor.Borrower\_Phone, bor.Borrower\_Email, b.ISBN, b.Due\_Date, b.Amount\_Due FROM borr\_with\_amount AS b, borrower as bor WHERE b.Borrower\_ID=bor.Borrower\_ID and b.Amount\_Due='$amtdue';

SELECT bor.Borrower\_ID, bor.F\_Name, bor.L\_Name, bor.Borrower\_Address, bor.Borrower\_Phone, bor.Borrower\_Email, b.ISBN, b.Due\_Date, b.Amount\_Due FROM borr\_with\_amount AS b, borrower as bor WHERE b.Borrower\_ID=bor.Borrower\_ID and b.Amount\_Due>='$amtdue' ORDER BY Amount\_Due;

select \* from borr\_with\_amount order by Borrowing\_Date desc;

insert into borrowings(Borrower\_ID, ISBN, Branch\_ID, Borrowing\_Date) values($id,'$isbn','$branch','$date');

delete from borrowings where Borrower\_ID='$id' and ISBN='$isbn' and Branch\_ID='$branch';

select \* from author;

insert into author values($authorid,'$authorname');

select \* from publisher;

insert into publisher values('$pubname','$pubaddress',$pubphone);

select \* from branch;

insert into branch values($brid,'$brname','$braddr');

**Database Connectivity**

I used a combination of HTML and PHP to create a user interface, which was then connected to my database to enable data retrieval and storage. This allowed me to create a dynamic and interactive web application that could easily interact with the database. The code snippet for database connectivity is:

$connection = mysqli\_connect("localhost:4306", "root","");

$db = mysqli\_select\_db($connection,'library\_db');

**UI Design**

The following are screenshots of the various pages that make up the user interface. They also display the sample data.

Homepage

A picture containing text, screenshot, software, multimedia software

Description automatically generated

Books Page

Displays information about all books in all the branches

A computer screen shot of a library

Description automatically generated with low confidence

Page to add a new book

A computer screen shot of a library

Description automatically generated with medium confidence

Page to search for a book

A computer screen shot of a library

Description automatically generated with medium confidence

Borrowers Page

Displays information about all borrowers

A computer screen shot of a library

Description automatically generated with low confidence

Page to add a new borrower

A computer screen shot of a library

Description automatically generated with medium confidence

Page to search for a borrower

A computer screen shot of a library

Description automatically generated with low confidence

Borrowings Page

Displays information about all present borrowings

A computer screen shot of a library

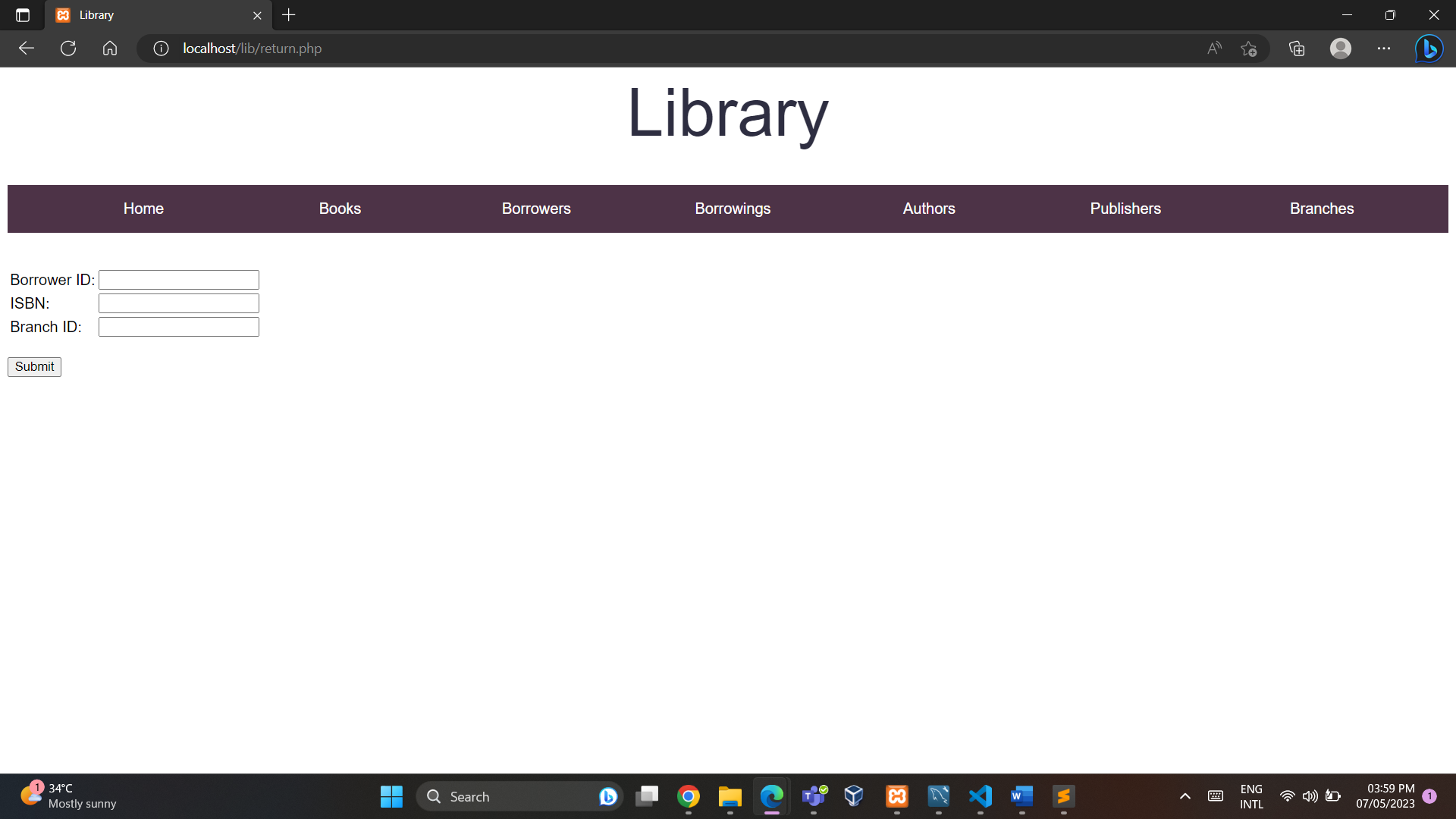
Description automatically generated with low confidence

Page to issue a book

A picture containing screenshot, text, software, multimedia software

Description automatically generated

Page to return a book



Authors Page

Displays information about all authors

A computer screen shot of a library

Description automatically generated with medium confidence

Page to add a new author

A computer screen shot of a library

Description automatically generated with medium confidence

Publishers Page

Displays information about all publishers

A computer screen shot of a library

Description automatically generated with medium confidence

Page to add a new publisher

A picture containing screenshot, text, software, multimedia software

Description automatically generated

Branches Page

Displays information about all branches

A computer screen shot of a library

Description automatically generated with medium confidence

Page to add a new branch

A picture containing screenshot, text, software, multimedia software

Description automatically generated

**Conclusion**

In conclusion, the manual management of books in a library can be a challenging task. It can lead to inconsistencies and errors in record-keeping. The implementation of a computerized book management service can help librarians keep track of books, borrowers, authors, publishers, and overdue books. The proposed project aims to design and implement a database that stores all the necessary information for efficient management of books in the library. It will allow librarians to add new books, keep track of borrowings and returns, and penalize borrowers for overdue books. By implementing such a system, librarians can efficiently manage the library, save time, and provide better services to the borrowers.

**Limitations And Future Work**

The project does not implement authentication and backup measures so both users and librarians currently have access to all data.

The project is also a simplified version of the real-life scenario. It does not allow users to search for books using quotes and the several categories such as genre and publication date. It only allows searching using publisher name, author name, book name and ISBN.

The database needs to be expanded further to have all these capabilities, making it more suitable for real-life use.

**References**

Database System Concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan

<https://www.studocu.com/en-gb/document/university-of-east-london/advanced-software-engineering/library-management-system-final-report/10662691>

<https://www.w3schools.com/php/>

<https://www.w3schools.com/html/html_intro.asp>