INTRODUCTION AND PROBLEM DEFINITION

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally not available in normal library management systems like facility of user login and a facility of teachers login .It also has a facility of admin login through which the admin can monitor the whole system .It also has facility of an online notice board where teachers can student can put up information about workshops or seminars being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the seminar can add it to the notice board . It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date and also the students can request the librarian to add new books by filling the book request form. The librarian after logging into his account i.e. admin account can generate various reports such as student report, issue report, teacher report and book report.

Overall this project of ours is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts.

LIBRARAY DATABASE MANAGEMENT

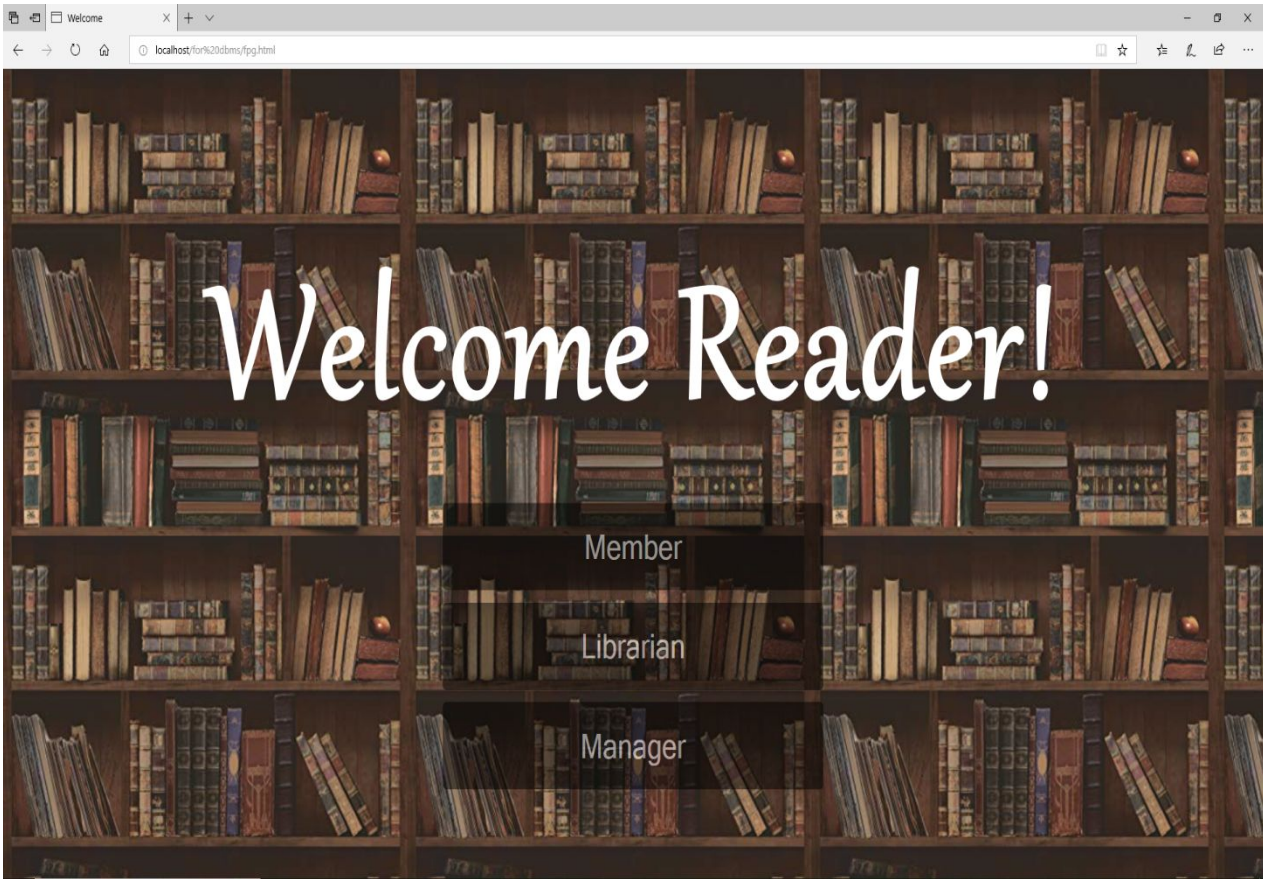
The ***miniworld*** for this project is a library that only lends novels and periodicals. A person needs to visit the library in real time and there, he pays for the ***membership*** and librarian enters the details for the particular member. The member is then given a ***member card*** with a *member ID*, which he can use to login to the website. On the website, the member can view the books/periodicals he has *borrowed*, the *due date* for the same and also the fine on it, if any. Through the website, the member can check for the availability of a novel/periodical and then he/she can ***reserve*** it if they want to.

Then, he can go to the library and borrow book where it is available (in particular offline branch of the whole organization).

Our entities are: **NOVEL, AUTHOR, PUBLISHER, PERIODICAL, MEMBER, MEMBER CARD, LIBRARIAN**. The relations between the Entities are: ***AUTHOR 'writes' PERIODICAL and NOVEL. MEMBER can borrow a NOVEL/PERIODICAL. A NOVEL is published by a PUBLISHER. A MEMBER has a MEMBER_CARD****.* ***A LIBRARIAN has data on the MEMBERs****.* There are 2 views in this database, ***the librarian's view and the member's view. The Librarian's view is used to change and update the database. The Member's view is used to only view the front-end.***

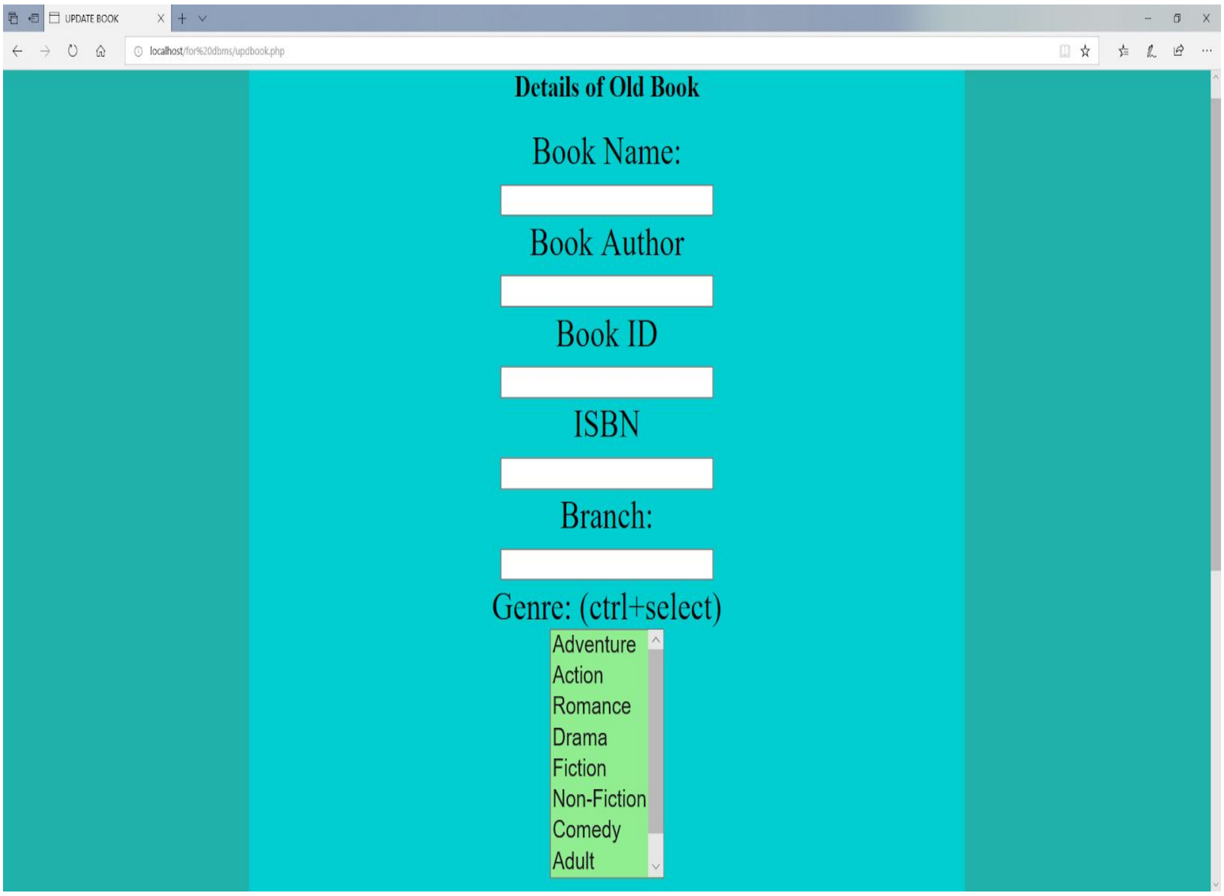
REQUIREMENTS

- INSERT – USER CAN INSERT ALL THE RELEVENT INFORMATION ACCORDING TO HIS USER PREFERENCES. IF HE IS A MEMBER OR LIBRARIAN. HE CAN INSERT HIS NAME, CONTACT, E-MAIL ADDRESS ETC.
- UPDATE-USER CAN UPDATE ALL OF HIS INFORMATION ALSO, WHATEVER PREVIOUSLY HAS BEEN INDERTED.
- DELETED – INFORMATION AND USER CAN AS WELL BE DELETED.
- ALL THE INFORMATION OF THEE BOOK ISSUES IS ALSO PROBIDED SUCH AS PRICE, DUE DATE, DATE OF ISSUE ETC.
- ALL THE INFORMATION RELATED TO USER IS PRESENT SUCH AS DATE OF ISSUE OF A PARTICULAR BOOK, ALSO FINE OR DUE DATE ISSUES.
- PUBLISHER INFORMATION.
- AUTHOR DETAILS SUCH AS AUTHOR NAME, NUMBER OF BOOKS.
- PERIODICAL LOCATION.
- GENRE OF BOOK IS ALSO PROVIDED.
- MEMBERCARD DETAILS.
- LIBRARY LOCATION – IT GIVES YOU DETAILED INFORMATION ABOUT WHICH PERIODICAL IS AVAILABLE AT WHICH LIBRARY.
- BOOK LOCATION.
- YOU CAN ALSO VIEW WHAT BOOKS ARE HELD BY WHAT PERSON AT A PARTICULAR TIME.
- NAME AND WHAT BOOKS ARE ISSUED.
- PRICE WHITIN THE RANGE.



Welcome Reader!

USER INTERFACE IMAGES



Details of Old Book

Book Name:

Book Author

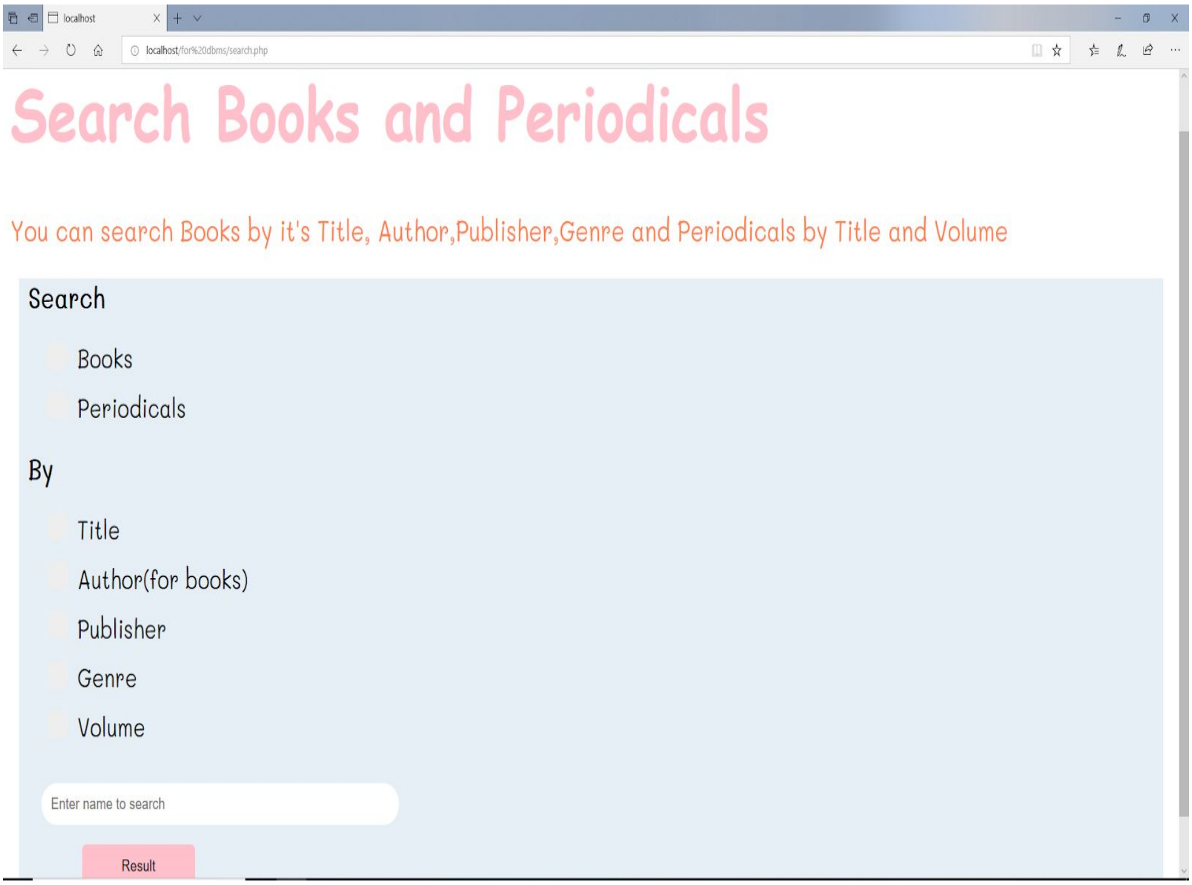
Book ID

ISBN

Branch:

Genre: (ctrl+select)

Adventure
Action
Romance
Drama
Fiction
Non-Fiction
Comedy
Adult



Search Books and Periodicals

You can search Books by it's Title, Author,Publisher,Genre and Periodicals by Title and Volume

Search

○ Books

○ Periodicals

By

○ Title

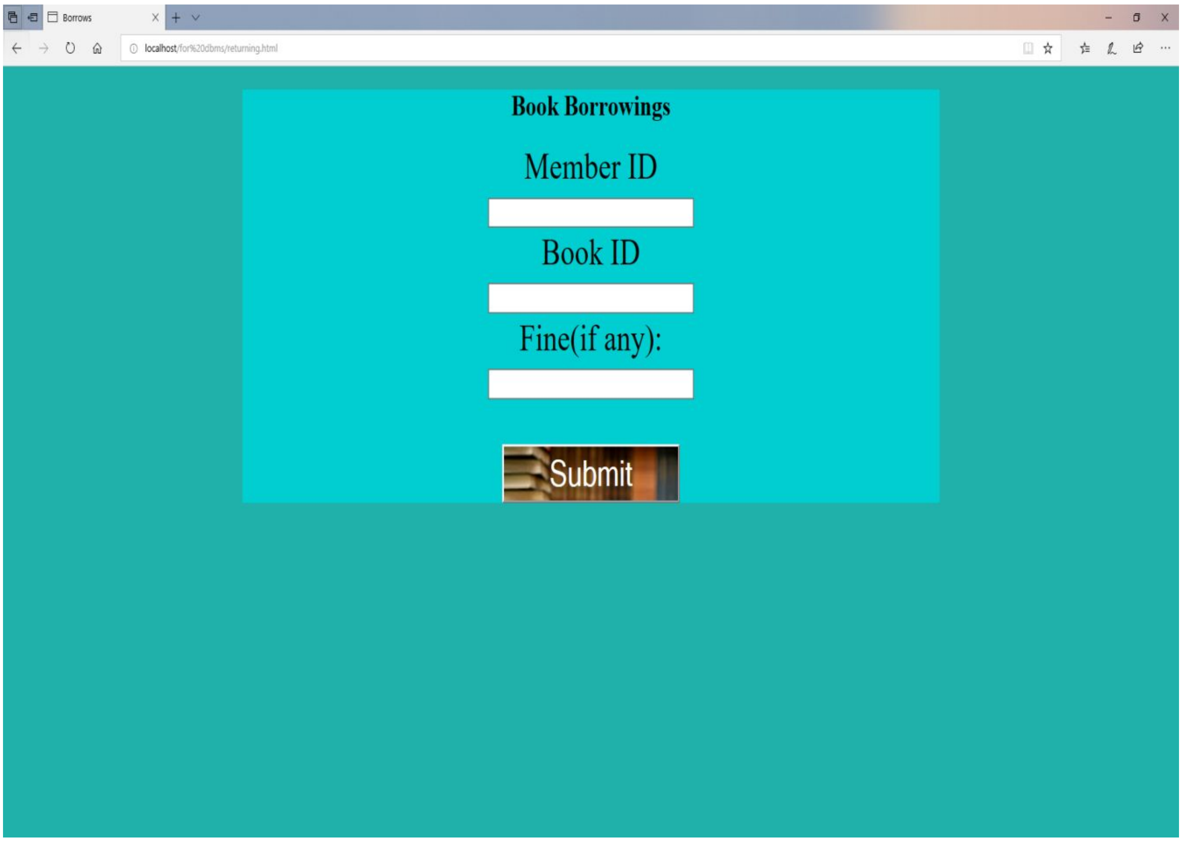
○ Author(for books)

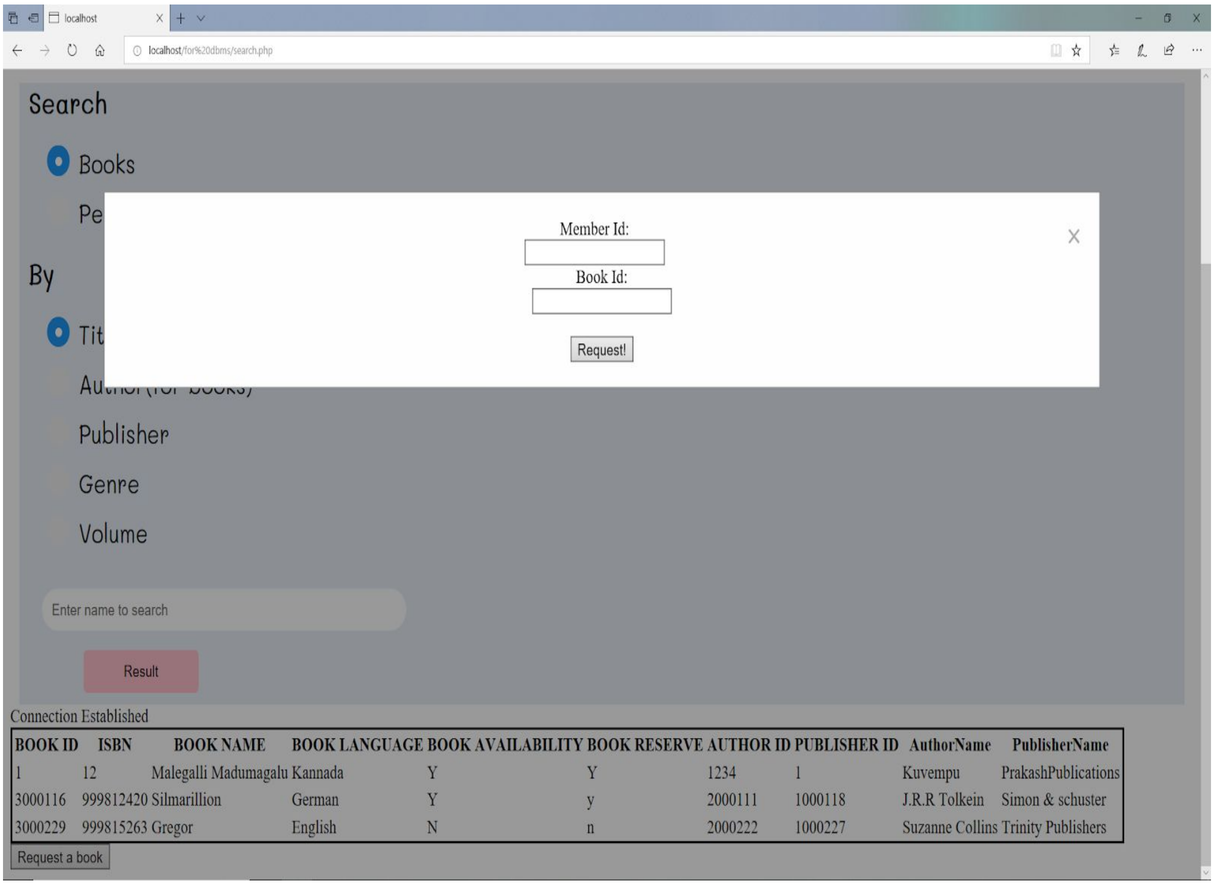
○ Publisher

○ Genre

○ Volume

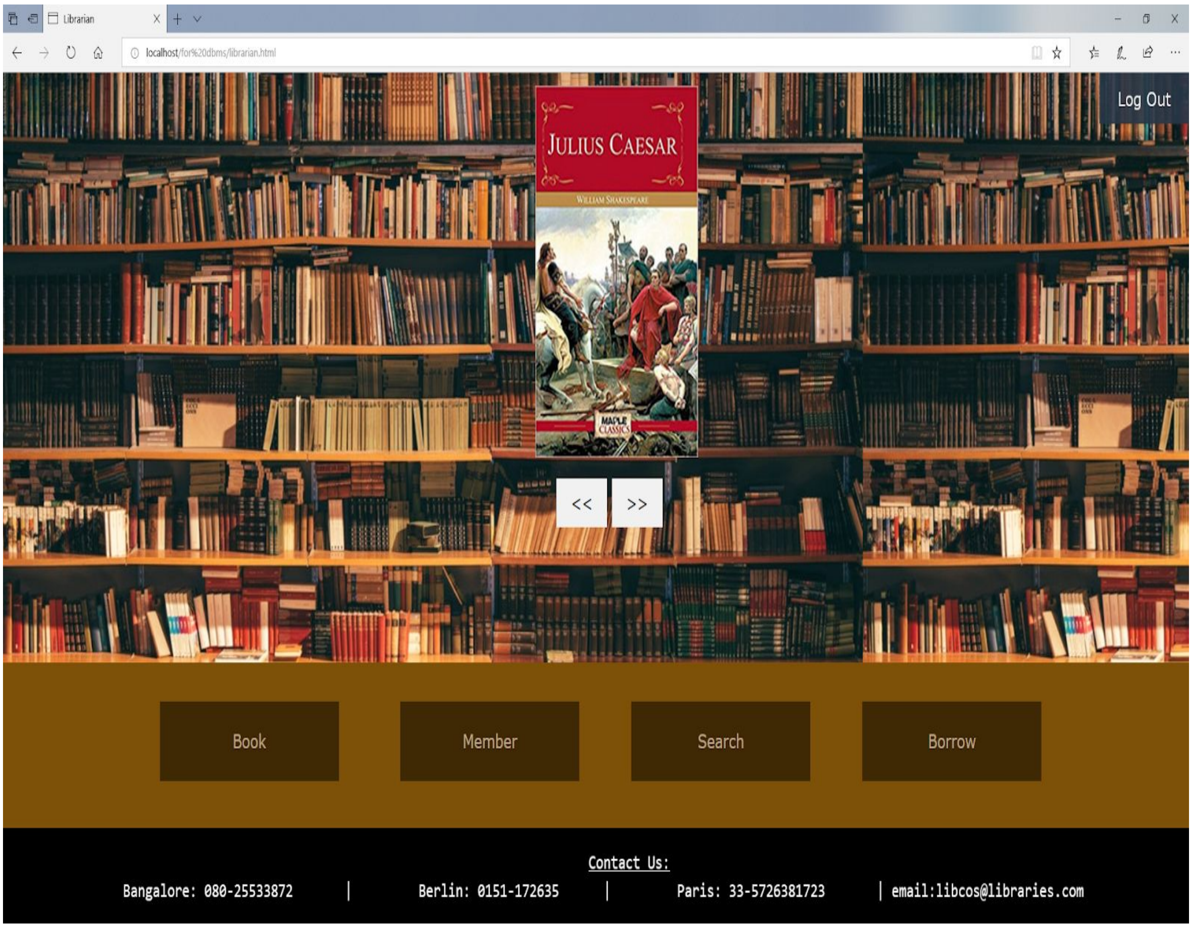
Enter name to search





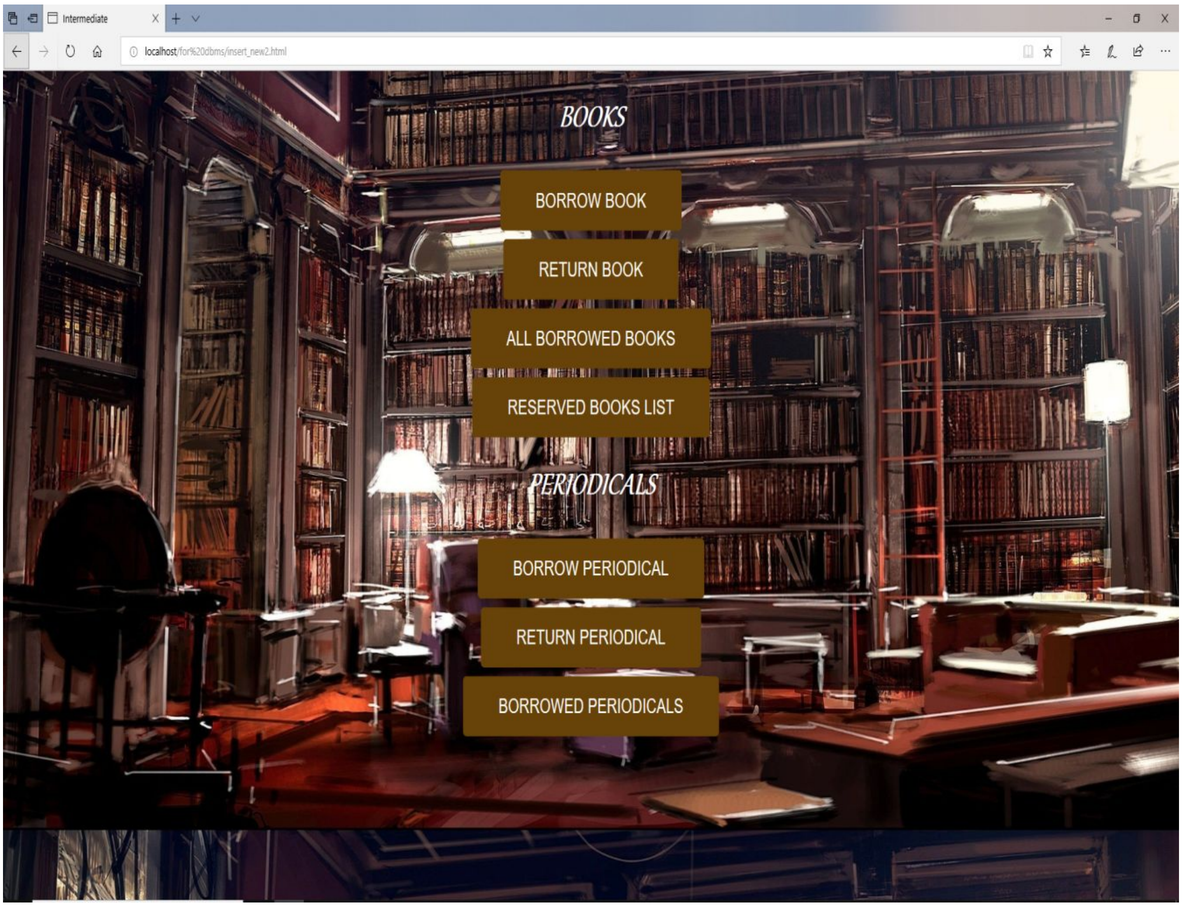
Connection Established

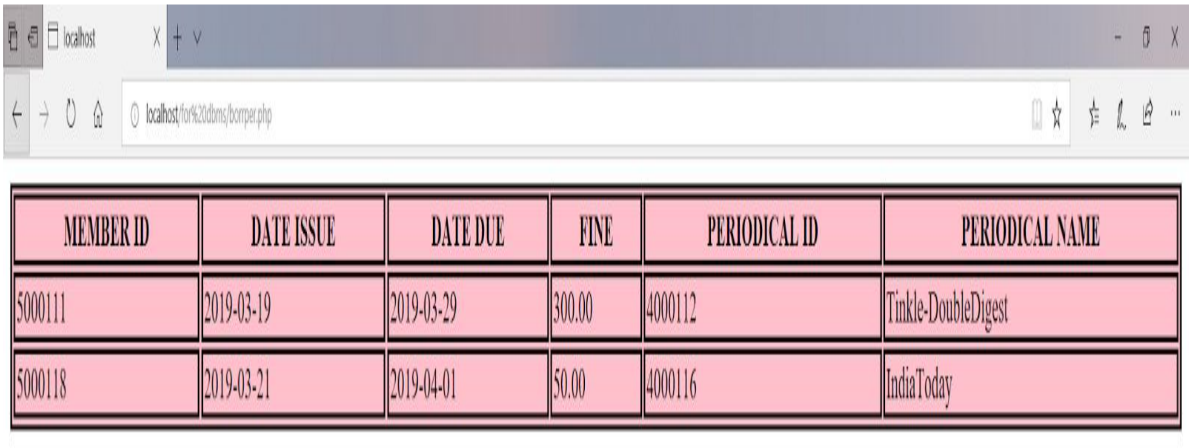
BOOK ID	ISBN	BOOK NAME	BOOK LANGUAGE	BOOK AVAILABILITY	BOOK RESERVE	AUTHOR ID	PUBLISHER ID	AuthorName	PublisherName
1	12	Malegalli Madumagalu	Kannada	Y	Y	1234	1	Kuvempu	PrakashPublications
3000116	999812420	Silmarillion	German	Y	y	2000111	1000118	J.R.R Tolkein	Simon & schuster
3000229	999815263	Gregor	English	N	n	2000222	1000227	Suzanne Collins	Trinity Publishers



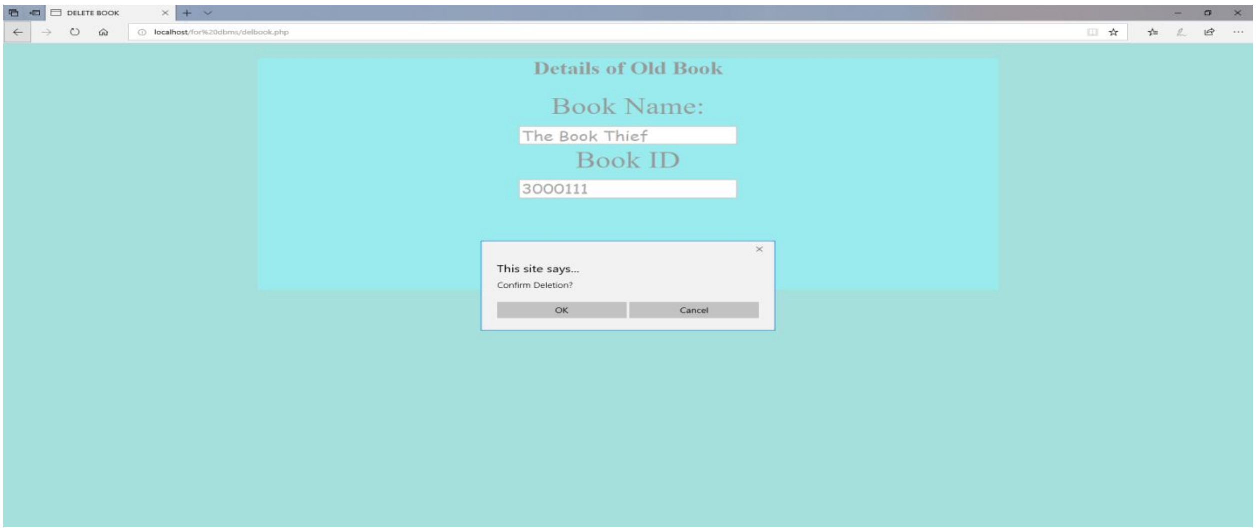
Contact Us:

Bangalore: 080-25533872 | Berlin: 0151-172635 | Paris: 33-5726381723 | email:libcos@libraries.com





MEMBER ID	DATE ISSUE	DATE DUE	FINE	PERIODICAL ID	PERIODICAL NAME
5000111	2019-03-19	2019-03-29	300.00	4000112	Tinkle-DoubleDigest
5000118	2019-03-21	2019-04-01	50.00	4000116	IndiaToday



Details of Old Book

Book Name:

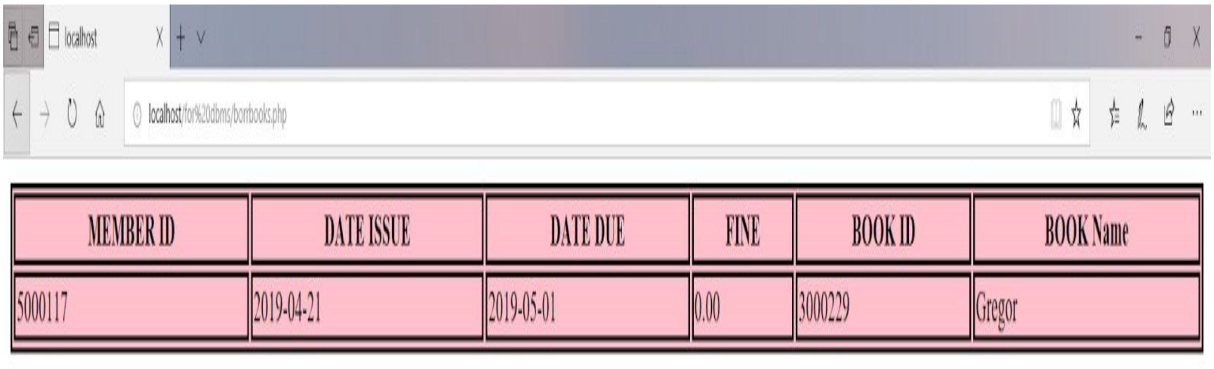
The Book Thief

Book ID

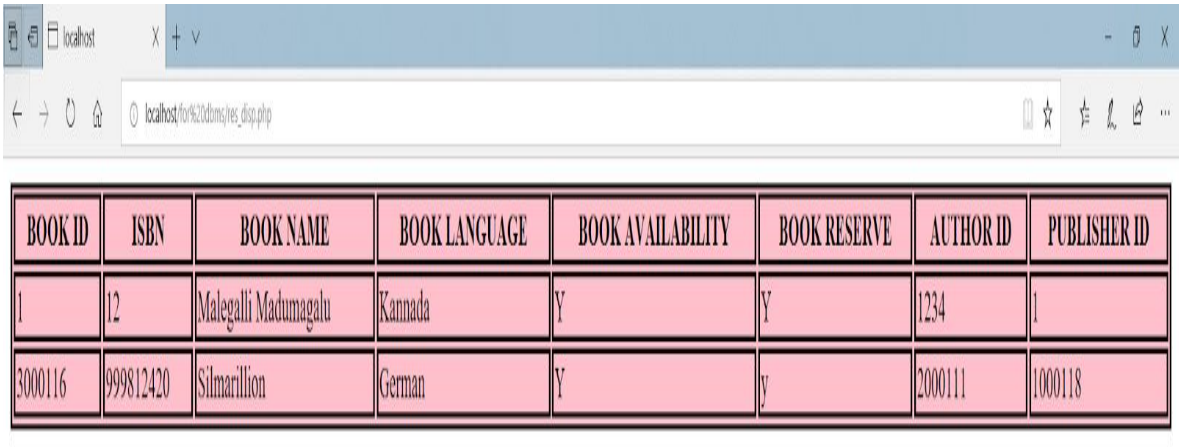
3000111

This site says...

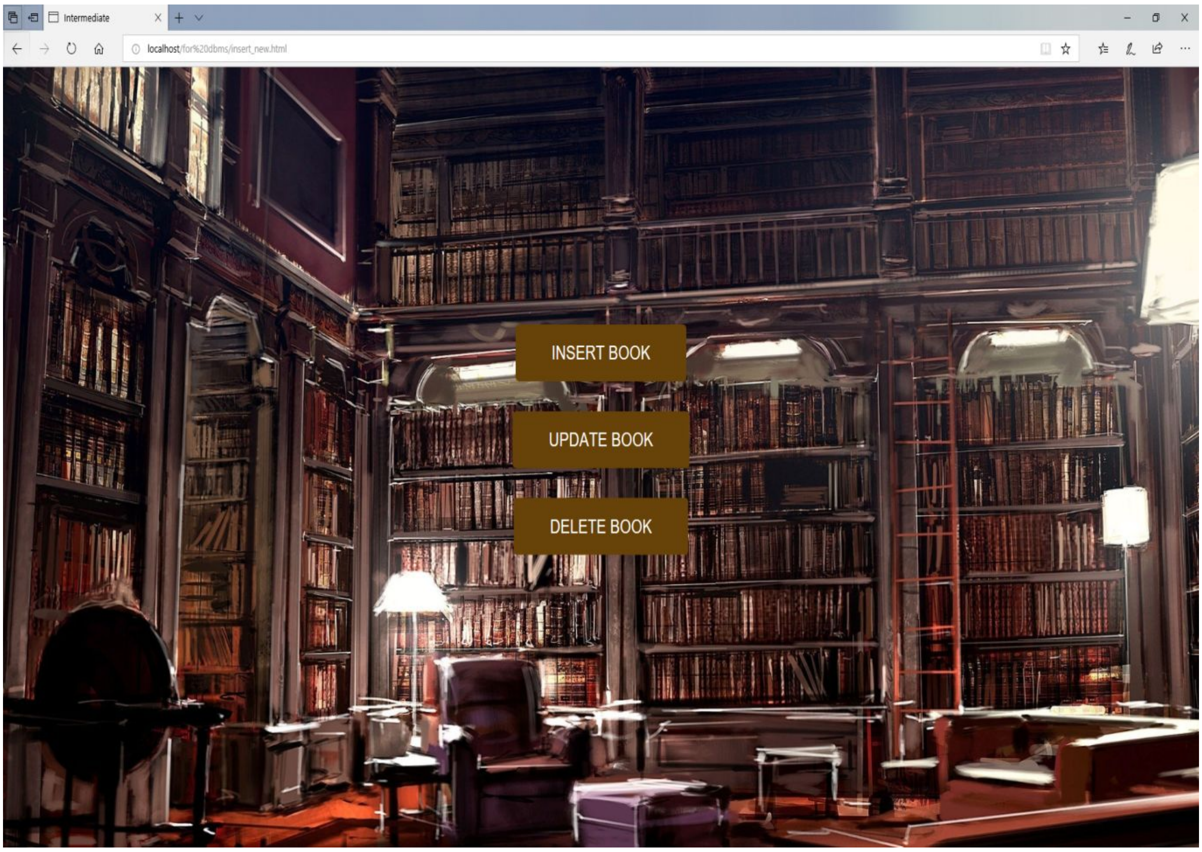
Confirm Deletion?

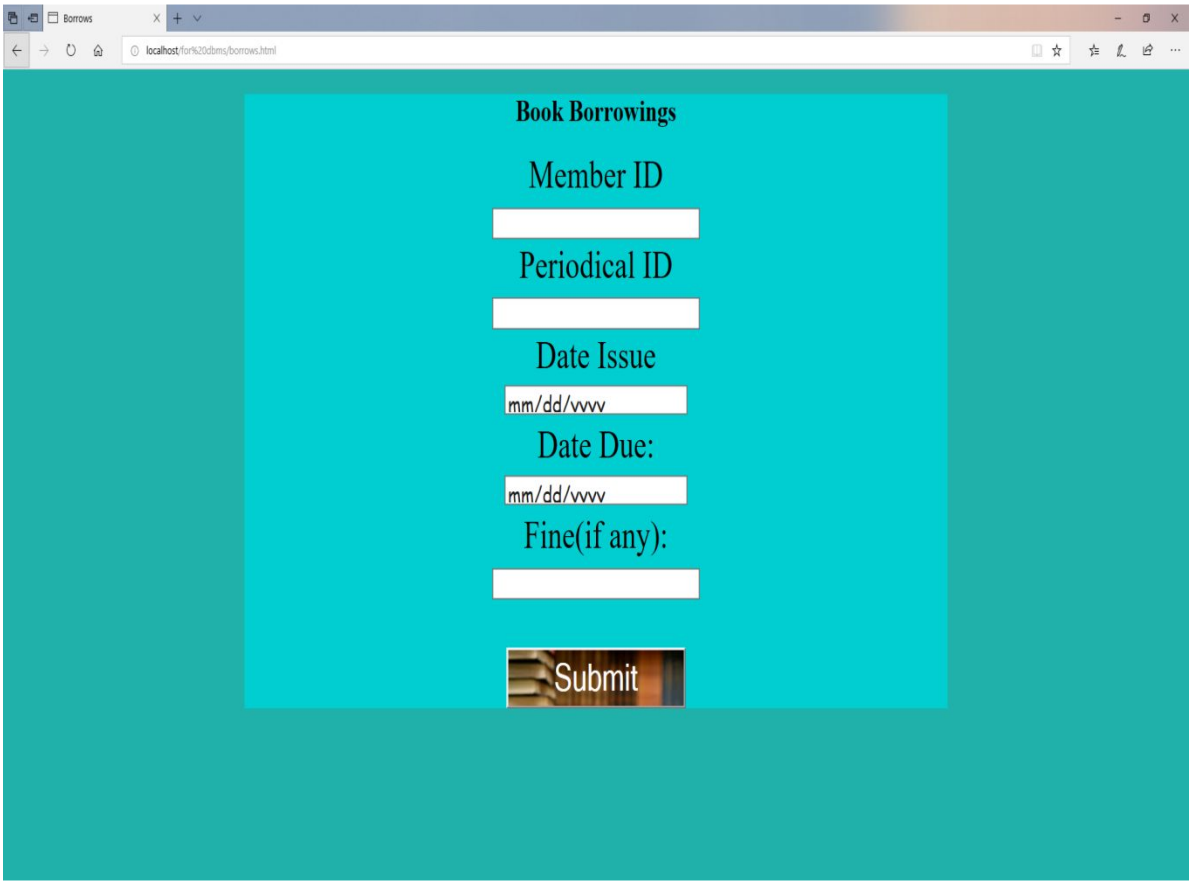


MEMBER ID	DATE ISSUE	DATE DUE	FINE	BOOK ID	BOOK Name
5000117	2019-04-21	2019-05-01	0.00	3000229	Gregor



BOOK ID	ISBN	BOOK NAME	BOOK LANGUAGE	BOOK AVAILABILITY	BOOK RESERVE	AUTHOR ID	PUBLISHER ID
1	12	Malegalli Madumagalu	Kannada	Y	Y	1234	1
3000116	999812420	Silmarillion	German	Y	y	2000111	1000118





Book Borrowings

Member ID

Periodical ID

Date Issue

mm/dd/yyyy

Date Due:

mm/dd/yyyy

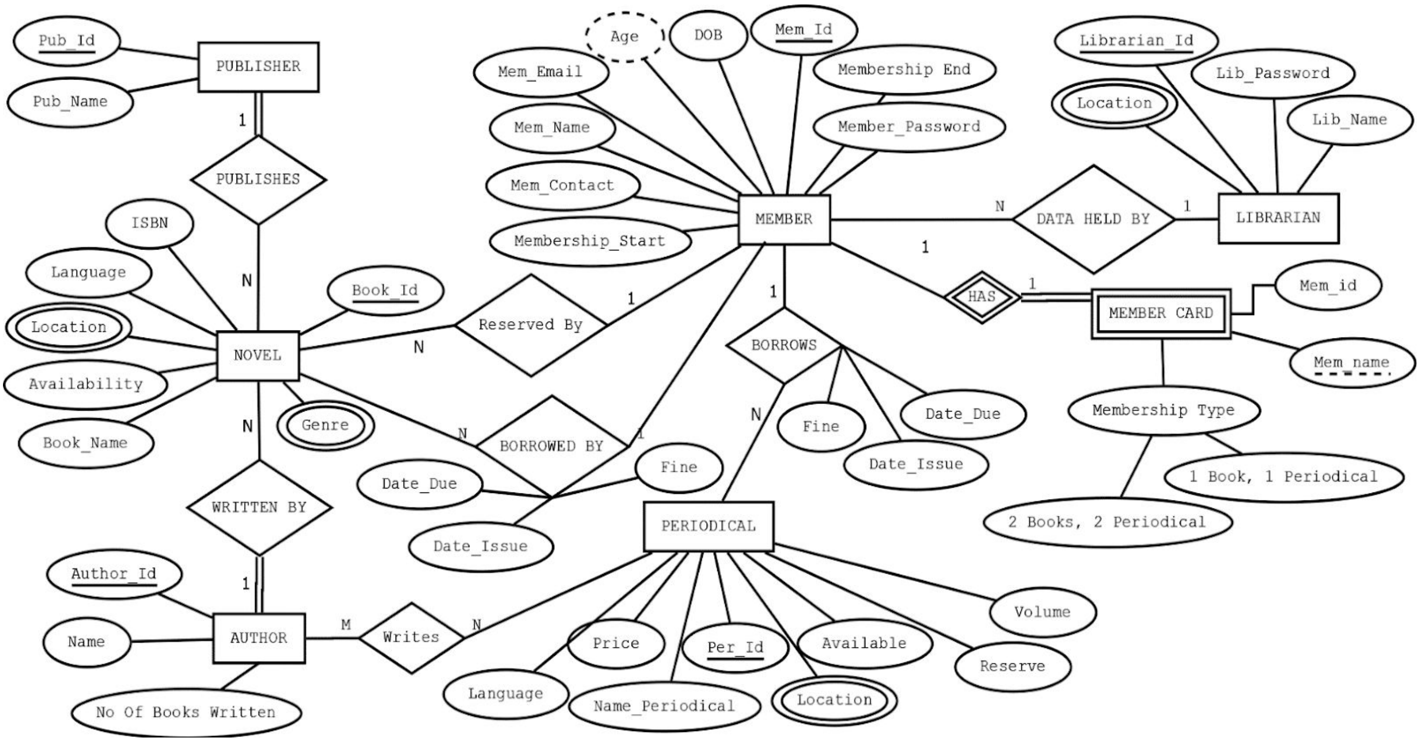
Fine(if any):

Create Table Statements

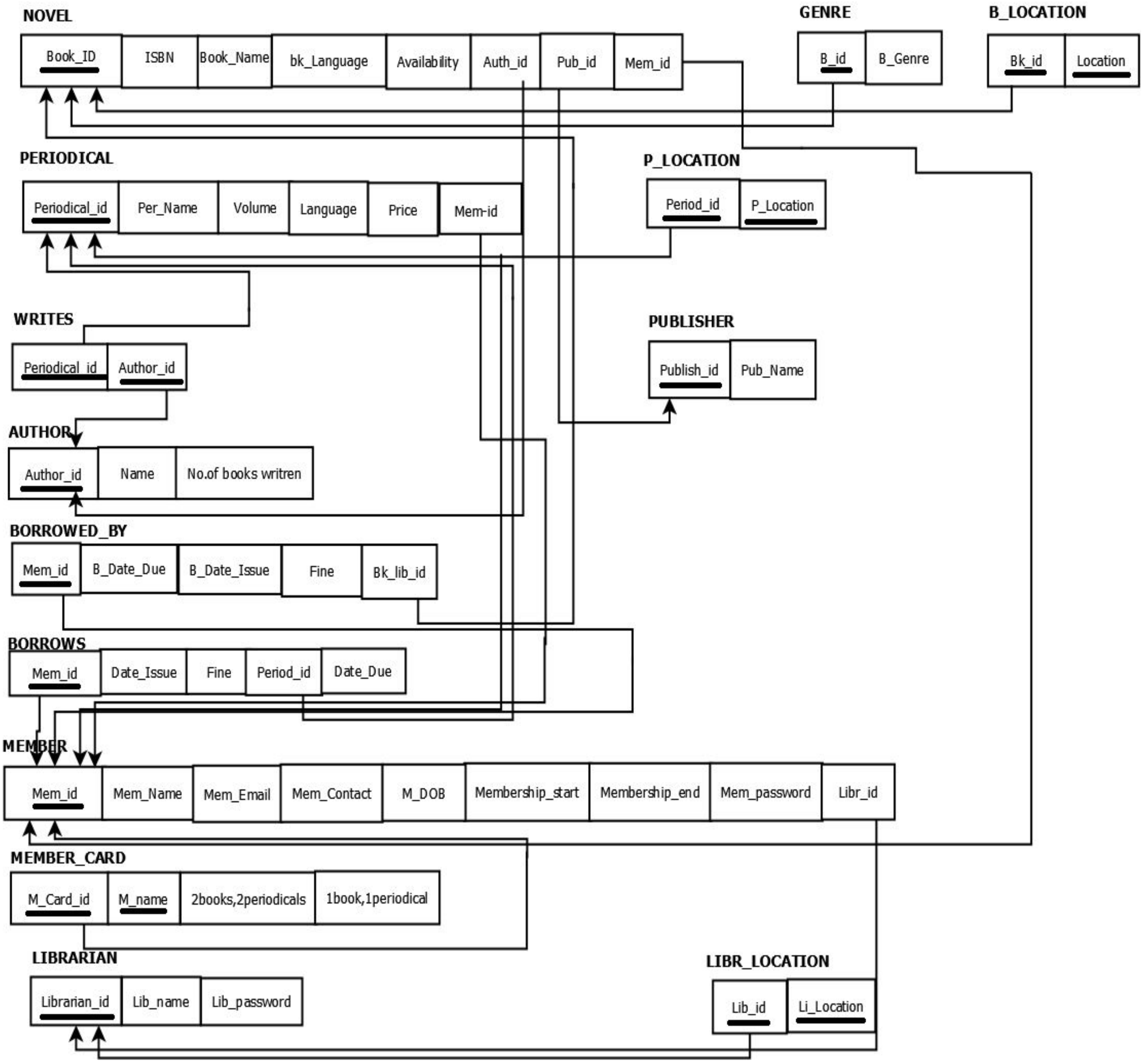
```
CREATE/ ALTER TABLE SCRIPTS AND CONSTRAINTS

CREATE TABLE PUBLISHER(
Pub_Id INT  PRIMARY KEY,
Pub_Name VARCHAR(20) NOT NULL);


CREATE TABLE AUTHOR(
Author_Id INT PRIMARY KEY,
Author_Name VARCHAR(20) UNIQUE,
Author_books INT);


CREATE TABLE NOVEL(
Book_Id INT PRIMARY KEY,
ISBN INT UNIQUE,
Book_Name VARCHAR(50) NOT NULL,
Book_Language VARCHAR(12) ,
Book_Availability CHAR(1) NOT NULL,
Book_Reserve CHAR(1),
Auth_Id INT REFERENCES AUTHOR(Author_Id) ON DELETE SET NULL ON UPDATE
CASCADE,

Publisher_Id INT REFERENCES PUBLISHER(Pub_Id) ON DELETE SET NULL ON
UPDATE CASCADE);


CREATE TABLE PERIODICAL(
Periodical_Id INT PRIMARY KEY,
Per_Name VARCHAR(20) NOT NULL,
Volume INT,
 Price INT);


CREATE TABLE P_LOCATION(
Per_Id INT NOT NULL,
P_Location CHAR(3) NOT NULL,
PRIMARY KEY(Per_Id,P_Location),
FOREIGN KEY(Per_Id) REFERENCES PERIODICAL(Periodical_Id) ON DELETE
CASCADE ON UPDATE CASCADE);


CREATE TABLE WRITES(
P_Id INT NOT NULL,
A_Id INT  NOT NULL,
PRIMARY KEY(P_Id,A_Id),
FOREIGN KEY(P_Id) REFERENCES PERIODICAL(Periodical_Id) ON DELETE
CASCADE ON UPDATE CASCADE,
FOREIGN KEY(A_Id) REFERENCES AUTHOR(Author_Id) ON DELETE SET NULL ON
UPDATE CASCADE);


CREATE TABLE GENRE(
B_Id INT PRIMARY KEY,
B_Genre VARCHAR(15) NOT NULL,
FOREIGN KEY(B_Id) REFERENCES NOVEL(Book_Id) ON DELETE CASCADE ON UPDATE
CASCADE);


CREATE TABLE MEMBER(
Mem_Id INT PRIMARY KEY,
Mem_Name VARCHAR(20) NOT NULL,
Mem_Email VARCHAR(50) ,
Mem_Contact INT NOT NULL,
M_DOB DATE NOT NULL,
Membership_start DATE NOT NULL,
Membership_End DATE NOT NULL,
Mem_psswd VARCHAR(12) NOT NULL,
Libr_Id INT NOT NULL,
FOREIGN KEY(Libr_Id) REFERENCES LIBRARIAN(Lib_Id) ON DELETE CASCADE ON
UPDATE CASCADE);


CREATE  TABLE BORROWED_BY(
Me_Id INT PRIMARY KEY,
B_Date_Issue DATE NOT NULL,
B_Date_Due DATE NOT NULL,
B_Fine DECIMAL(5,2),
Bk_Id INT NOT NULL,
FOREIGN KEY(Bk_Id) REFERENCES NOVEL(Book_Id) ON DELETE CASCADE ON
UPDATE CASCADE,
FOREIGN KEY(Me_Id) REFERENCES MEMBER(Mem_Id) ON DELETE CASCADE ON
UPDATE CASCADE);



CREATE  TABLE BORROWS
(Memb_Id INT PRIMARY KEY,
P_Date_Issue DATE NOT NULL,
P_Date_Due DATE NOT NULL,
P_Fine DECIMAL(5,2),
Pr_Id INT NOT NULL,
FOREIGN KEY(Memb_Id) REFERENCES MEMBER(Mem_Id) ON DELETE CASCADE ON
UPDATE CASCADE,
FOREIGN KEY(Pr_Id) REFERENCES PERIODICAL(Periodical_Id) ON DELETE
CASCADE ON UPDATE CASCADE);


CREATE TABLE MEMBERCARD
(Mbr_Id INT PRIMARY KEY,
Mbr_Name varchar(20) NOT NULL,
B_2_P_2 CHAR(1),B_1_P_1 CHAR(1),
FOREIGN KEY(Mbr_Id) REFERENCES MEMBER(Mem_Id) ON DELETE CASCADE ON
UPDATE CASCADE);


CREATE TABLE LIBRARIAN(
Lib_Id INT PRIMARY KEY,
Lib_Name VARCHAR(20) NOT NULL,
Lib_psswd VARCHAR(12) NOT NULL);
```

```
CREATE TABLE LIBR_LOCATION(
Li_Id INT NOT NULL,
Li_Loc CHAR(3) NOT NULL,
PRIMARY KEY(Li_Id,Li_Loc),
FOREIGN KEY(Li_Id) REFERENCES LIBRARIAN(Lib_Id) ON DELETE CASCADE ON
UPDATE CASCADE);


CREATE TABLE B_LOCATION(
Boo_Id INT NOT NULL,
B_Location CHAR(3) NOT NULL,
PRIMARY KEY(Boo_Id,B_Location),
FOREIGN KEY(Boo_Id) REFERENCES NOVEL(Book_Id) ON DELETE CASCADE ON
UPDATE CASCADE
);
```

ER diagram


Pub_Id
Pub_Name
PUBLISHER
1
PUBLISHES
N
ISBN
Language
Location
Availability
Book_Name
NOVEL
Book_Id
Genre
N
WRITTEN BY
1
Author_Id
Name
AUTHOR
No Of Books Written
M
Writes
N
Age
DOB
Mem_Id
Mem_Email
Membership End
Mem_Name
Member_Password
Mem_Contact
Membership_Start
MEMBER
Reserved By
1
N
BORROWED BY
N
1
Date_Due
Fine
Date_Issue
1
BORROWS
N
Fine
Date_Due
Date_Issue
PERIODICAL
Language
Price
Name_Periodical
Per_Id
Location
Available
Volume
Reserve
N
DATA HELD BY
1
LIBRARIAN
Librarian_Id
Lib_Password
Location
Lib_Name
1
HAS
1
MEMBER CARD
Mem_id
Mem_name
Membership Type
1 Book, 1 Periodical
2 Books, 2 Periodical

Relational Schema



Complex Queries

```
Select author_name

from from libmans.author,libmans.novel

Where libmans.author.author_id=libmans.novel.auth_id;
```

```
Select pub_name

From libmans.publisher,libmans.novel

Where libmans.publisher.pub_id=libmans.novel.publisher_id;
```

```
Select mem_name,book_name

from libmans.borrowed_by,libmans.novel,libmans.member

Where libmans.borrowed_by.me_id=libmans.member.mem_id and
libmans.borrowed_by.bk_id=libmans.novel.book_id;
```

```
Select mem_name, b_date_issue,b_date_due

from libmans.member left outer join libmans.borrowed_by on
libmans.member.mem_id=libmans.borrowed_by.me_id;
```

```
Select mem_name, book_name,b_date_issue,b_date_due

from libmans.borrowed_by left outer join libmans.member
on(member.mem_id=borrowed_by.me_id)left outer join
libmans.novel on (novel.book_id=borrowed_by.bk_id);
```

```
Select author_name

from libmans.author natural join libmans.novel where
libmans.author.author_id=libmans.novel.auth_id;
```

```
Select book_name,book_language,pub_name

from libmans.novel inner join libmans.publisher on
libmans.novel.publisher_id=libmans.publisher.pub_id;
```

CONCLUSION

This website provides a computerized version of library management system which will benefit the students as well as the staff of the library. It makes entire process online where student can search books, staff can generate reports and do book transactions. It also has a facility for student login where student can login and can see status of books issued as well request for book or give some suggestions. It has a facility of teacher's login where teachers can add lectures notes and also give necessary suggestion to library and also add info about workshops or events happening in our college or nearby college in the online notice board.

There is a future scope of this facility that many more features such as online lectures video tutorials can be added by teachers as well as online assignments submission facility, a feature Of group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfils each users need in the best way possible