CS 631 - DATA MANAGEMENT SYSTEMS DESIGN

(Prof. Dimitrios Theodoratos)

PROJECT - THE CITY LIBRARY

DELIVERABLE 3

Kavitha Kannanunny	**kk46@njit.edu**
Harshitha Sadashiva Murthy	**hs57@njit.edu**
Ankan Dash	**ad892@njit.edu**

PHASE 2 GOAL:

In this phase of the project we implement an online library management system that maintains all the library resources such as books, journals etc. The database management system was created after going through several steps throughout the semester that include creating the ERR diagram (Phase 1) and the Relational database schema (Phase 2). The database system has functionality for both the reader and the database administrator. The administrator is responsible for the database maintenance and he/she can add a reader or documents and search for relevant documents and check their status. The readers can reserve the documents, borrow them using the database system.

DESCRIPTION OF CREATION OF DATABASE SCHEMA:

The database was created using SQL language on the MySQL relational database management system which includes several relevant tables like Authors, Journal issue, Journal volume, Documents, Books, Readers, Publishers etc. to store all the relevant information in the database.

APPLICATION PROGRAM INSTANCE:

We used the PHP language to create the database application system. XAMPP (Apache HTTP) server is required to run and execute the code. The system has a simple yet efficient GUI for ease of use and can be used by a variety of readers. The database application can be used both on Windows, Mac and Linux.

REVISIONS:

We did not make any revisions to the Relational Schema provided to us in Phase 2.

PROBLEMS ENCOUNTERED:

This is the first time we were creating a database application using PHP thus we faced a lot of problems during the application development. Implementing all the functionalities was difficult so we prioritized what to implement so that we have a running application with all the basic functionalities.

Running the website using an XAMPP server can be tricky sometimes and due to some missing dependencies or files the website may fail to load or function properly. It is vital to make sure to use the right version of the packages and libraries to have smooth functioning of the application.

USER GUIDE:

1. The user can download the database application.
2. The user will need to install the XAMPP server and run the APACHE and MySQL server to run the application and create the database. MySQL workbench can also be used for database creation.
3. The user will need to create the database named '`citylibrary`' which contains all the relevant tables as described in the relational schema.
4. The administrator and the reader can login or sign up by providing the relevant information.
5. If the user encounters some errors then she/he can modify the code using some basic text editor like sublime text.

SQL COMMANDS TO CREATE DATABASE, TABLES AND POPULATE DATA:

Admin :

```
DROP DATABASE IF EXISTS citylibrary;
CREATE DATABASE IF NOT EXISTS citylibrary;
USE citylibrary;

CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `FullName` varchar(100) DEFAULT NULL,
  `AdminEmail` varchar(120) DEFAULT NULL,
  `UserName` varchar(100) NOT NULL,
  `Password` varchar(100) NOT NULL,
  `updationDate` timestamp NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


INSERT INTO `admin`(`id`,`FullName`,`AdminEmail`, `UserName`, `Password`,`updationDate`) VALUES (001, 'Kavitha Kannanunny', 'kavita.kannanunny@gmail.com','admin','0192023a7bbd73250516f069df18b5', sysdate());
```

Reader:

```
DROP TABLE IF EXISTS READER;
CREATE TABLE READER (
      RID INT AUTO_INCREMENT PRIMARY KEY,
      RTYPE VARCHAR(50),
      RNAME VARCHAR(50),
      RADDRESS VARCHAR(200),
      PHONE_NO VARCHAR(13)
);

INSERT INTO `READER` (`RID`, `RNAME`, `RTYPE`, `PHONE_NO`, `RADDRESS`)
VALUES
(1, 'Maria', 'Student', '98652555', '615 Pavonia Ave'),
(2, 'Kavitha', 'Staff', '85690025', '97 Hudson St'),
(3, 'Karen', 'Senior Citizen', '59874527', '309 Sussex St'),
(4, 'Harshitha', 'Other', '585856224', '127 Warren St'),
(5, 'Ankan', 'Student', '672423754', '739 Harrison Av'),
(6, 'Akash', 'Student', '672432754', '250 Harrison Av');
```

Reservation:

```
DROP TABLE IF EXISTS RESERVATION;
CREATE TABLE RESERVATION (
      RES_NO INT AUTO_INCREMENT PRIMARY KEY,
      DTIME DATETIME
);


INSERT INTO `RESERVATION` (`RES_NO`,`DTIME`) VALUES
(1,sysdate()),
(2,sysdate()),
(3,sysdate()),
(4,sysdate()),
(5,sysdate()),
(6,sysdate());
```

Borrowing:

```
DROP TABLE IF EXISTS BORROWING;
CREATE TABLE BORROWING (
      BOR_NO INT AUTO_INCREMENT PRIMARY KEY,
      BDTIME DATETIME,
      RDTIME DATETIME
);


INSERT INTO `BORROWING` (`BOR_NO`,`BDTIME`,RDTIME) VALUES
(1, '2020-01-15', '2020-01-30'),
(2, '2020-01-15', '2020-01-30'),
(3, '2020-01-15', NULL),
(4, '2020-01-15', '2020-01-30'),
(5, '2020-01-15', NULL),
(6, '2020-01-15', NULL);
```

Branch:

```
DROP TABLE IF EXISTS BRANCH;
CREATE TABLE BRANCH (
      BID INT AUTO_INCREMENT PRIMARY KEY,
      LNAME VARCHAR(50),
      LOCATION VARCHAR(50)
);


INSERT INTO `branch` (`BID`, `LNAME`, `LOCATION`) VALUES
(1, 'Oklahoma City', 'Ap #671-247 Eget Ave'),
(2, 'Kingussie', 'Ap #115-3145 Magna. Avenue'),
(3, 'Spruce Grove', '9115 Sed Road'),
(4, 'Waiuku', '659-4785 Turpis St.'),
(5, 'Buizingen', 'P.O. Box 354, 6610 Facilisis St.'),
(6, 'Varna/Vahrn', '8949 Eu Street'),
(7, 'Chanco', '3980 Nullam Ave'),
(8, 'Rimbey', '8799 Laoreet Ave'),
(9, 'Quedlinburg', '5163 A, Street'),
(10, 'Abbateggio', '4463 Ut St.'),
(11, 'Turriaco', 'P.O. Box 935, 9142 Amet St.'),
(12, 'Rio Marina', 'P.O. Box 230, 4429 Orci Road'),
(13, 'Carleton', 'Ap #459-2446 Non, Avenue'),
```

```
(14, 'Midlands', 'P.O. Box 152, 8543 Pede. Av.'),
(15, 'San Diego', '3780 At, Av.'),
(16, 'Sargodha', 'P.O. Box 667, 7201 Aliquam Avenue'),
(17, 'Notre-Dame-du-Nord', '155-8151 Vestibulum Rd.'),
(18, 'Irricana', '7069 Habitant Road'),
(19, 'Pangnirtung', 'P.O. Box 940, 6860 Ut Rd.'),
(20, 'Aartrijke', 'Ap #404-322 Etiam Rd.');
```

Publisher:

```
DROP TABLE IF EXISTS PUBLISHER;
CREATE TABLE PUBLISHER (
      PUBLISHERID INT AUTO_INCREMENT PRIMARY KEY,
      PUBNAME VARCHAR(50),
      ADDRESS VARCHAR(100)
);


INSERT INTO `PUBLISHER` (`PUBLISHERID`, `PUBNAME`,ADDRESS) VALUES
(1, 'SCHOLASTIC','610 Pavonia Ave'),
(2, 'McGraw-Hill Education','23 CLIFTON PLACE'),
(3, 'Wiley','23 ST'),
(4, 'Macmillian','65 LEXINGTON'),
(5, 'Oxford','22 BENTLY'),
(6, 'Pearson','123 STREET'),
(7, 'Penguin Random','124 GROVE ST'),
(8, 'Harper Collins','22 Harrison AVE'),
(9, 'Three Rivers Press','12 NEWARK'),
(10, 'Elsevier','22 HOBOKEN');
```

Person:

```
DROP TABLE IF EXISTS PERSON;
CREATE TABLE PERSON (
      PID INT AUTO_INCREMENT PRIMARY KEY,
      PNAME VARCHAR(50)
);


INSERT INTO `PERSON` (`PID`, `PNAME`) VALUES
(1, 'Genevieve Farrell'),
```

```
(2, 'Ashely Thornton'),
(3, 'Rhiannon Rush'),
(4, 'Ivy Campbell'),
(5, 'Tanek Beasley'),
(6, 'Leroy Melton'),
(7, 'Colton Banks'),
(8, 'Kato Valenzuela'),
(9, 'Chaney Brock'),
(10, 'Mira Jensen'),
(11, 'Odessa Rocha'),
(12, 'Dan Mallory'),
(13, 'Stephen Jones'),
(14, 'Lou Aronica'),
(15, 'Lester Ray');
```

Document:

```
DROP TABLE IF EXISTS DOCUMENT;
CREATE TABLE DOCUMENT (
      DOCID INT AUTO_INCREMENT PRIMARY KEY,
      TITLE VARCHAR(50),
      PDATE DATE,
      PUBLISHERID INT,
      FOREIGN KEY (PUBLISHERID)
      REFERENCES PUBLISHER (PUBLISHERID)
      ON DELETE CASCADE
);


INSERT INTO `document` (`DOCID`, `TITLE`, `PDATE`, `PUBLISHERID`)
VALUES
(1, 'Goosebumps Part 1', '2002-07-22', 1),
(2, 'Harry Potter - The Sorcerer's Stone', '2001-01-14', 5),
(3, 'Stephen King Biography', '2005-08-06', 3),
(4, 'Spider Man ', '2005-08-20', 8),
(5, 'Song Of Ice and Fire', '2018-07-19', 6),
(6, 'Julius caesar', '2002-09-22', 5),
(7, 'MahaBharata', '2018-08-2', 9),
(8, 'Ramayana', '2018-07-12', 9),
(9, 'Integral Mathematics', '2018-07-22', 4),
(10, 'Physics', '2018-05-18', 2);
(11, 'Elsevier - Neural Networks', '2002-07-22', 1),
(12, 'IEEE - Machine Learning', '2001-01-14', 5),
```

```
(13, 'Oxford University Press - Bioinformatics', '2005-08-06', 3),
(14, 'Springer - Information Technology ', '2018-08-20', 8),
(15, 'Microtome - Machine Learning', '2018-07-19', 6),
(16, 'Elsevier', '2018-07-22', 1),
(17, 'IEEE', '2018-01-14', 5),
(18, 'Oxford University Press', '2005-08-06', 3),
(19, 'Springer', '2005-08-20', 8),
(20, 'Microtome', '2018-07-19', 6);
```

Copy:

```
DROP TABLE IF EXISTS COPY;
CREATE TABLE COPY (
      DOCID INT,
      COPYNO INT,
      BID INT,
    POSITION VARCHAR(20),
      PRIMARY KEY (DOCID , COPYNO , BID),
      FOREIGN KEY (DOCID)
      REFERENCES DOCUMENT (DOCID)
      ON DELETE CASCADE,
      FOREIGN KEY (COPYNO)
      REFERENCES BRANCH (BID)
      ON DELETE CASCADE
);


INSERT INTO `COPY` (`DOCID`,`COPYNO`, `BID`, `POSITION`) VALUES
(1,  1, 2, '001A03'),
(2, 2, 4, '001B09'),
(3, 3, 3, '001C08'),
(4,  4, 2, '001D05'),
(5,  5, 2, '001E03'),
(6, 6, 2, '001A05');
```

Reserves:

```
DROP TABLE IF EXISTS RESERVES;
CREATE TABLE RESERVES (
      RID INT,
      RESERVATION_NO INT,
```

```
      DOCID INT,
      COPYNO INT,
      BID INT,
      PRIMARY KEY (RESERVATION_NO , DOCID , COPYNO , BID),
      FOREIGN KEY (DOCID , COPYNO , BID)
      REFERENCES COPY (DOCID , COPYNO , BID)
      ON DELETE CASCADE,
      FOREIGN KEY (RESERVATION_NO)
      REFERENCES RESERVATION (RES_NO)
      ON DELETE CASCADE,
      FOREIGN KEY (RID)
      REFERENCES READER (RID)
      ON DELETE CASCADE
);


INSERT INTO `RESERVES` (`RESERVATION_NO`,`RID`, `DocId`, `CopyNo`,
`BID`) VALUES
(1, 1, 1, 1, 2),
(2, 2, 2, 2, 4),
(3, 3, 3, 3, 3),
(4, 4, 4, 4, 2),
(5, 5, 5, 5, 2),
(6, 6, 6, 6, 2);
```

Borrows:

```
DROP TABLE IF EXISTS BORROWS;
CREATE TABLE BORROWS (
      BOR_NO INT,
      DOCID INT,
      COPYNO INT,
      BID INT,
      RID INT,
      PRIMARY KEY (BOR_NO , DOCID , COPYNO , BID),
      FOREIGN KEY (DOCID , COPYNO , BID)
      REFERENCES COPY (DOCID , COPYNO , BID)
      ON DELETE CASCADE,
      FOREIGN KEY (BOR_NO)
      REFERENCES BORROWING (BOR_NO)
      ON DELETE CASCADE,
      FOREIGN KEY (RID)
      REFERENCES READER (RID)
```

```
      ON DELETE CASCADE
);


INSERT  INTO  `borrows`  (`BOR_NO`,`RID`,  `DOCID`,  `COPYNO`,  `BID`)
VALUES
(1, 1, 1, 1, 2),
(2, 2, 2, 2, 4),
(3, 3, 3, 3, 3),
(4, 4, 4, 4, 2),
(5, 5, 5, 5, 2),
(6, 6, 6, 6, 2);
```

Book:

```
DROP TABLE IF EXISTS BOOK;
CREATE TABLE BOOK (
      DOCID INT PRIMARY KEY,
      ISBN VARCHAR(50),
      FOREIGN KEY (DOCID)
      REFERENCES DOCUMENT (DOCID)
      ON DELETE CASCADE
);


INSERT INTO `BOOK` (`DocId`, `ISBN`) VALUES
(1, '0-1234-23456'),
(2, '1-2345-67543'),
(3, '7-4325-98076'),
(4, '3-4325-93465'),
(5, '5-2342-23435'),
(6, '6-7976-98766'),
(7, '6-7976-98763'),
(8, '6-7976-9876'),
(9, '5-2342-23435'),
(10,'5-2342-23435');
```

Authors:

```
DROP TABLE IF EXISTS AUTHORS;
CREATE TABLE AUTHORS (
      PID INT,
```

```
      DOCID INT,
      PRIMARY KEY (PID , DOCID),
      FOREIGN KEY (PID)
      REFERENCES PERSON (PID)
      ON DELETE CASCADE,
      FOREIGN KEY (DOCID)
      REFERENCES BOOK (DOCID)
      ON DELETE CASCADE
);

INSERT INTO `AUTHORS` (`PID`, `DOCID`) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10,10);
```

Journal Volume:

```
DROP TABLE IF EXISTS JOURNAL_VOLUME;
CREATE TABLE JOURNAL_VOLUME (
      DOCID INT PRIMARY KEY,
      VOLUME_NO INT,
      EDITOR INT,
      FOREIGN KEY (DOCID)
      REFERENCES DOCUMENT (DOCID)
      ON DELETE CASCADE,
      FOREIGN KEY (EDITOR)
      REFERENCES PERSON (PID)
      ON DELETE CASCADE
);

INSERT INTO `JOURNAL_VOLUME` (`DOCID`,VOLUME_NO,EDITOR) VALUES
( 8,1,1),
( 9,2,2),
(10,3,3);
```

Journal Issue:

```
DROP TABLE IF EXISTS JOURNAL_ISSUE;
CREATE TABLE JOURNAL_ISSUE (
      DOCID INT,
      ISSUE_NO INT,
      SCOPE VARCHAR(100),
      PRIMARY KEY (DOCID , ISSUE_NO),
      FOREIGN KEY (DOCID)
      REFERENCES JOURNAL_VOLUME (DOCID)
      ON DELETE CASCADE
);


INSERT INTO `JOURNAL_ISSUE` (`DOCID`,ISSUE_NO) VALUES
( 8,1),
( 9,2),
(10,3);
```

Gedits:

```
DROP TABLE IF EXISTS GEDITS;
CREATE TABLE GEDITS (
      PID INT,
      DOCID INT,
      ISSUE_NO INT,
      PRIMARY KEY (DOCID , ISSUE_NO , PID),
      FOREIGN KEY (PID)
         REFERENCES PERSON (PID)
      ON DELETE CASCADE,
      FOREIGN KEY (DOCID , ISSUE_NO)
      REFERENCES JOURNAL_ISSUE (DOCID , ISSUE_NO)
      ON DELETE CASCADE
);


INSERT INTO `GEDITS` (`PID`, `DOCID`,ISSUE_NO) VALUES
(13, 8,1),
(14, 9,2),
(15,10,3);
```

Proceedings:

```
DROP TABLE IF EXISTS PROCEEDINGS;
CREATE TABLE PROCEEDINGS (
      DOCID INT PRIMARY KEY,
      CDATE DATE,
      CLOCATION VARCHAR(50),
      CEDITOR VARCHAR(50),
      FOREIGN KEY (DOCID)
      REFERENCES DOCUMENT (DOCID)
      ON DELETE CASCADE
);


INSERT INTO `PROCEEDINGS` (DOCID,CDATE,CLOCATION,CEDITOR) VALUES
(1,makedate(2020,3),'Harrison','john'),
(2,makedate(2020,4),'Jersey city','Mike'),
(8,makedate(2020,5),'Newark','Tyson'),
(9,makedate(2020,6),'Hoboken','Taylor'),
(10,makedate(2020,3),'Clifton','Mark');
```

Chairs:

```
DROP TABLE IF EXISTS CHAIRS;
CREATE TABLE CHAIRS (
      PID INT,
      DOCID INT,
      PRIMARY KEY (PID , DOCID),
      FOREIGN KEY (PID)
      REFERENCES PERSON (PID)
      ON DELETE CASCADE,
      FOREIGN KEY (DOCID)
      REFERENCES PROCEEDINGS (DOCID)
      ON DELETE CASCADE
);


INSERT INTO `CHAIRS` (`PID`, `DOCID`) VALUES
(1,1),
(2,2),
(8,8),
(9,9),
(10,10);
```

Source code:

We have provided the source code for some to the Reader and Admin functionalities. The complete source code is available on GITHUB.

Github Link : City Library Management

Admin Login:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/adminlogin.php

Reserve book:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/reserve-book.php

Manage books:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/manage-books.php

Checked out documents and fines:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/compute-fine.php

Admin Features

Add author:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/add-author.php

Add book:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/add-book.php

Add journal:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/add-journal.php

Add proceedings:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/add-proceedings.php

Manage Authors:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/manage-authors.php

Manage Books:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/manage-books.php

Manage Branch:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/manage-branch.php

Reg-students:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/reg-students.php

Signup:

https://github.com/kavihat/City-Library-Management/blob/main/library_k/admin/signup.php

Insert Scripts:

Get number N and branch number I as input and print the top N most frequent borrowers (Rid and name) in branch I and the number of books each has borrowed

```
Select COUNT(B.DOCID),R.RID,R.RNAME,COUNT(BOR_NO) AS post_count from borrows B, document D, READER R,BRANCH BR WHERE  B.DocId = D.DocId AND B.RID=R.RID AND BR.BID=B.BID AND B.BID = I GROUP BY B.DocId,R.RID, R.RNAME ORDER BY post_count DESC LIMIT N ;
```

Here values N and I are taken from the user. From the resultset of the query RID,RNAME AND COUNT(B.DOCID) are displayed to the interface.

Get number N as input and print the top N most frequent borrowers (Rid and name) in the library and the number of books each has borrowed.

```
Select COUNT(B.DOCID),R.RID,R.RNAME,COUNT(BOR_NO) AS post_count from borrows B, document D, READER R WHERE  B.DocId = D.DocId AND B.RID=R.RID GROUP BY B.DocId,R.RID, R.RNAME ORDER BY post_count DESC LIMIT N ;
```

Here values Nis taken from the user. From the resultset of the query RID,RNAME AND COUNT(B.DOCID) are displayed to the interface.

Get number N and branch number I as input and print the N most borrowed books in branch I.

```
Select B.DOCID,D.TITLE,R.RID,R.RNAME,COUNT(BOR_NO) AS post_count from borrows B, document D, READER R,BRANCH BR WHERE  B.DocId = D.DocId AND B.RID=R.RID AND BR.BID=B.BID AND B.BID = I GROUP BY B.DocId,R.RID, R.RNAME ORDER BY post_count DESC LIMIT N ;
```

Here values N and I are taken from the user. From the resultset of the query D.TITLE is displayed to the interface

Get number N as input and print the N most borrowed books in the library.

```
Select B.DOCID,D.TITLE,R.RID,R.RNAME,COUNT(BOR_NO) AS post_count from borrows B, document D, READER R WHERE  B.DocId = D.DocId AND B.RID=R.RID GROUP BY B.DocId,R.RID, R.RNAME ORDER BY post_count DESC LIMIT N ;
```

Here values N is taken from the user.

Get a year as input and print the 10 most popular books of that year in the library

```
SELECT COUNT(B.BOR_NO) AS
post_count,COUNT(R.RESERVATION_NO) AS RCOUNT from
borrows B, RESERVES R ,BORROWING G WHERE  B.DocId =
R.DocId AND B.BOR_NO = G.BOR_NO AND
YEAR(SYSDATE())=YEAR(G.BDTIME)  GROUP BY B.DocId LIMIT 10;
```

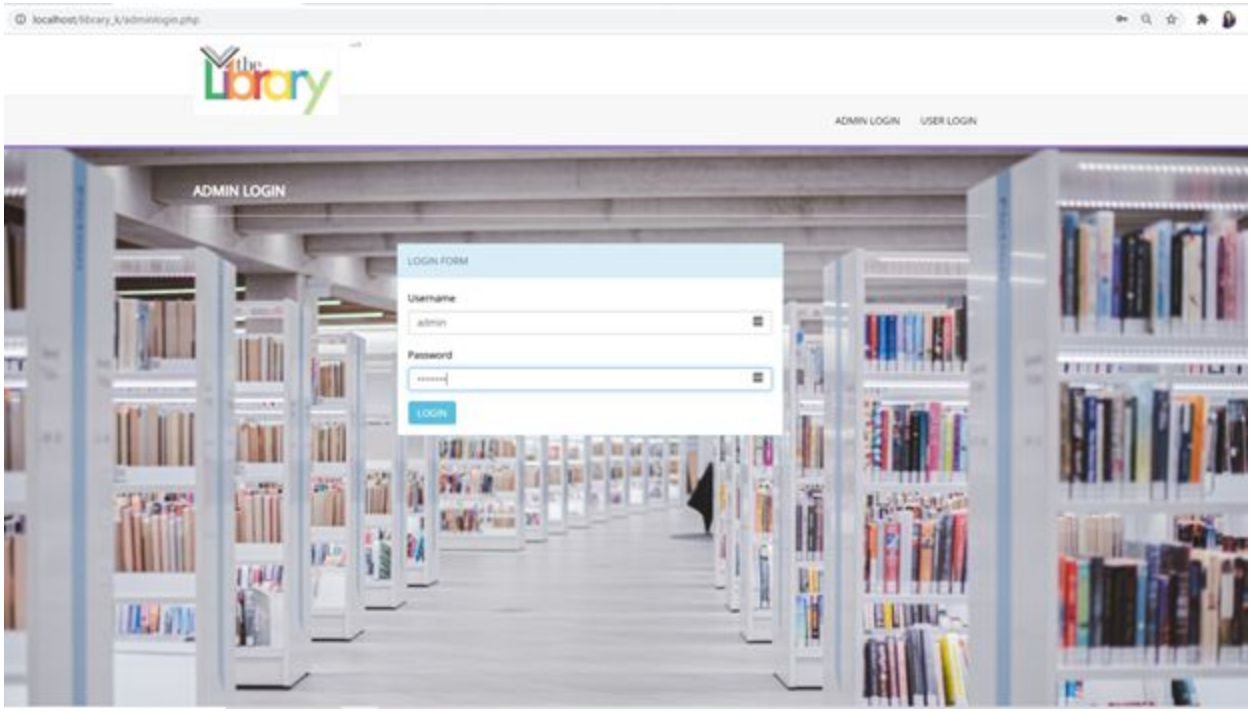
Get a start date S and an end date E as input and print, for each branch, the branch Id and name and the average fine paid by the borrowers for documents borrowed from this branch during the corresponding period of time.

```
SELECT AVG(FINE),BID FROM(SELECT 0.02 *DATEDIFF(SYSDATE(), BDTIME) AS FINE,B.BID FROM BRANCH B, DOCUMENT D,BORROWS BR,BORROWING G WHERE B.BID=BR.BID AND BR.DOCID=D.DOCID AND G.BOR_NO=BR.BOR_NO AND RDTIME IS NULL AND BR.BDTIME>S AND BR.BDTIME<E) AS F GROUP BY BID;
```

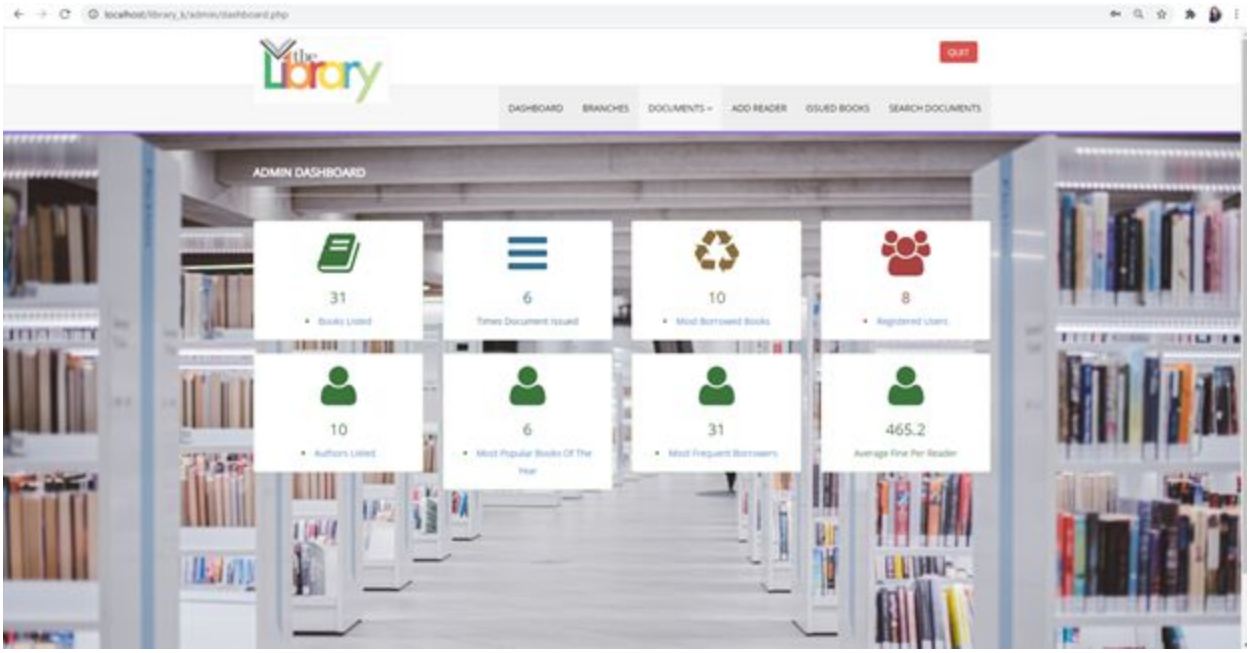
Here values S and E are taken from the user.

APPLICATION SCREENSHOTS:

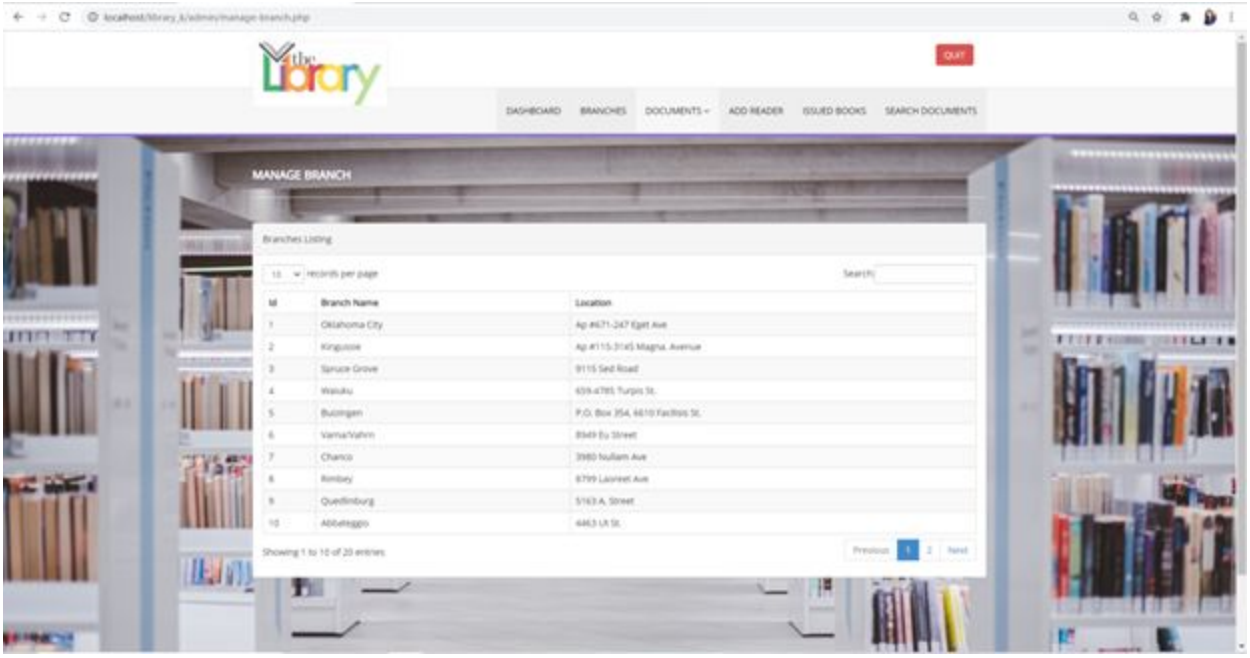
Login form for the Admin:



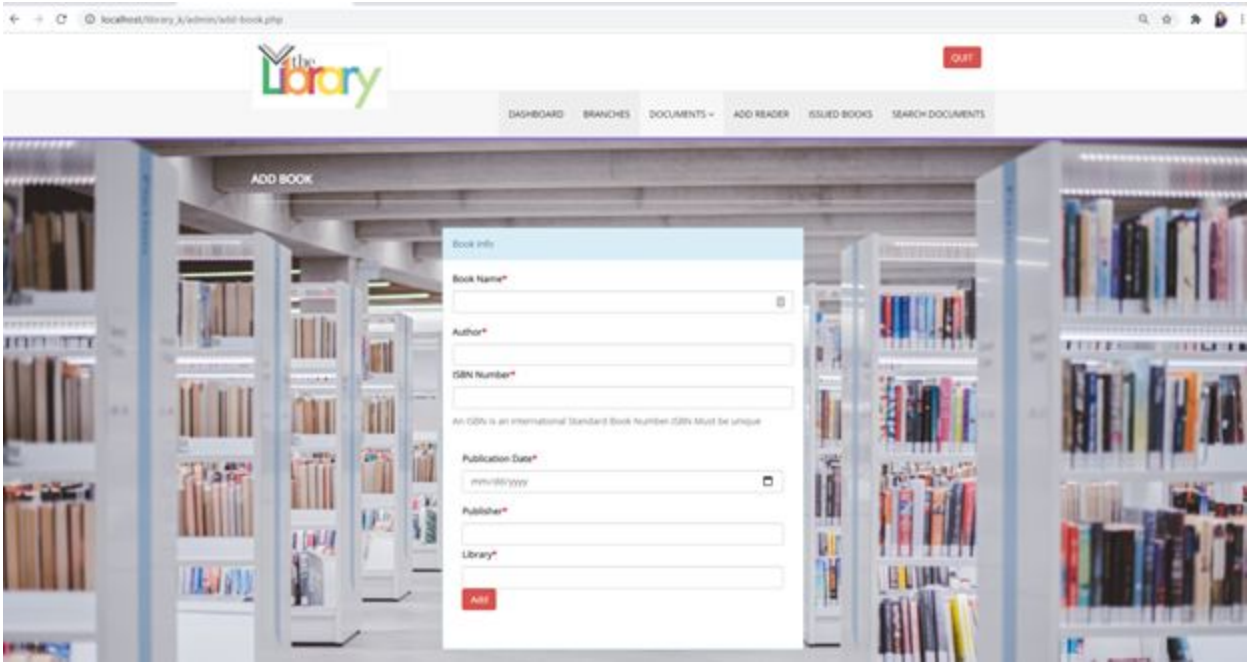
Admin dashboard:



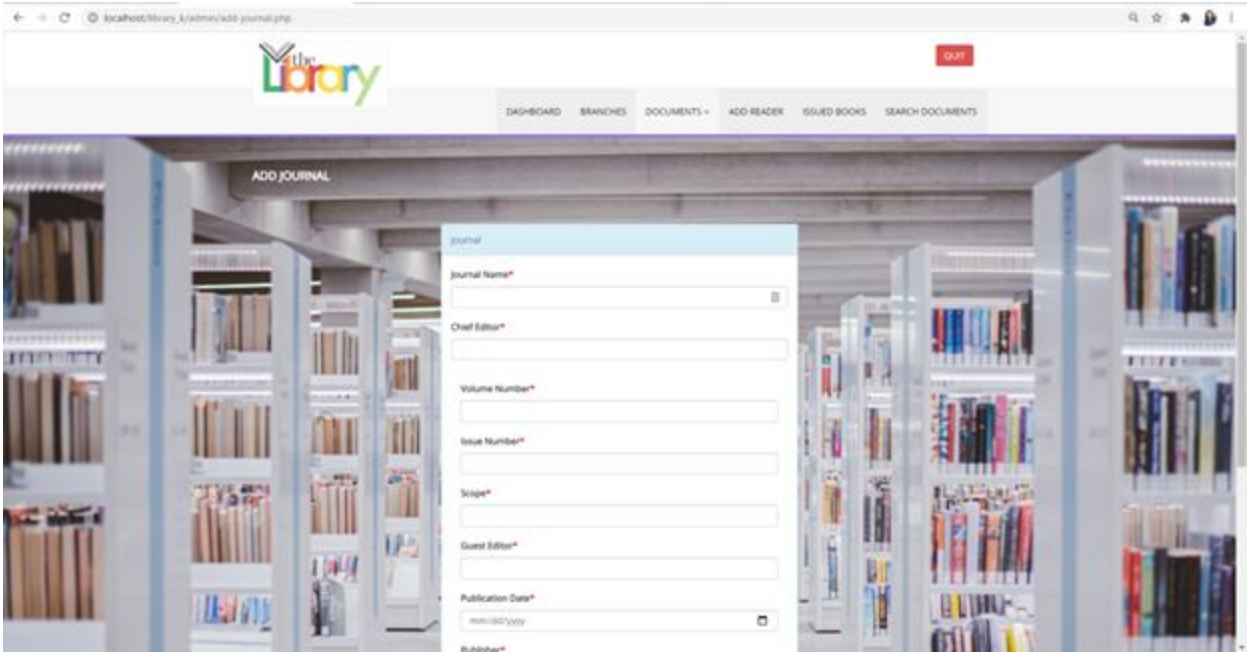
Branch details:



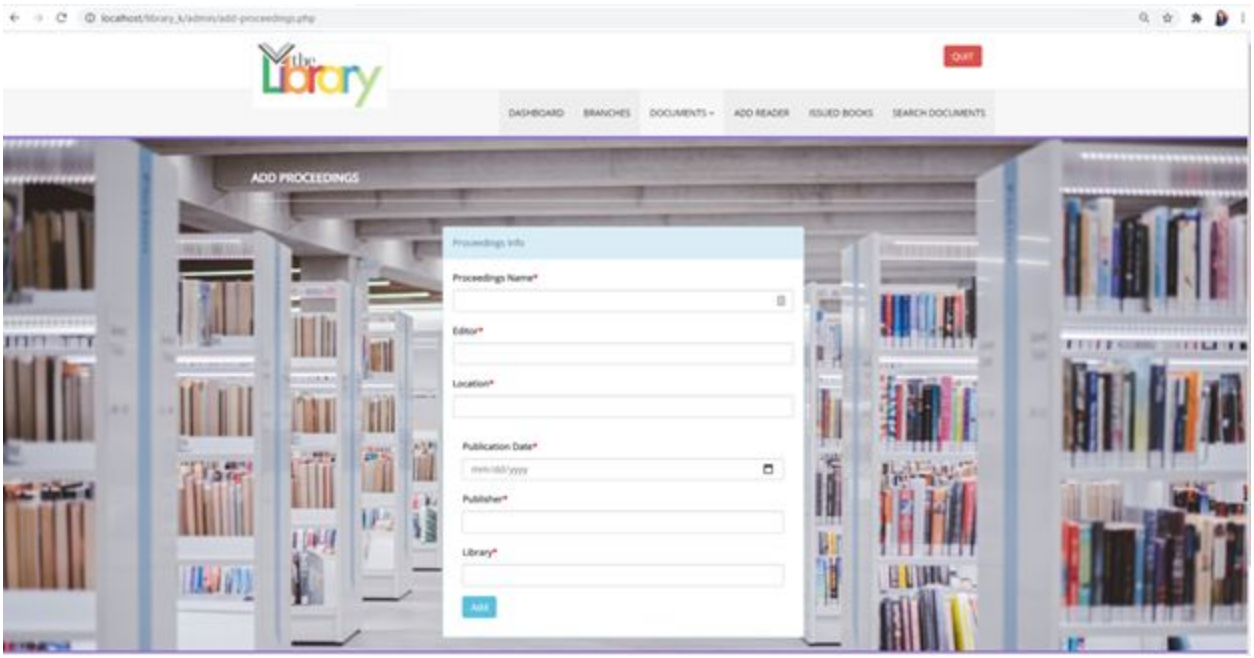
Add a new book:



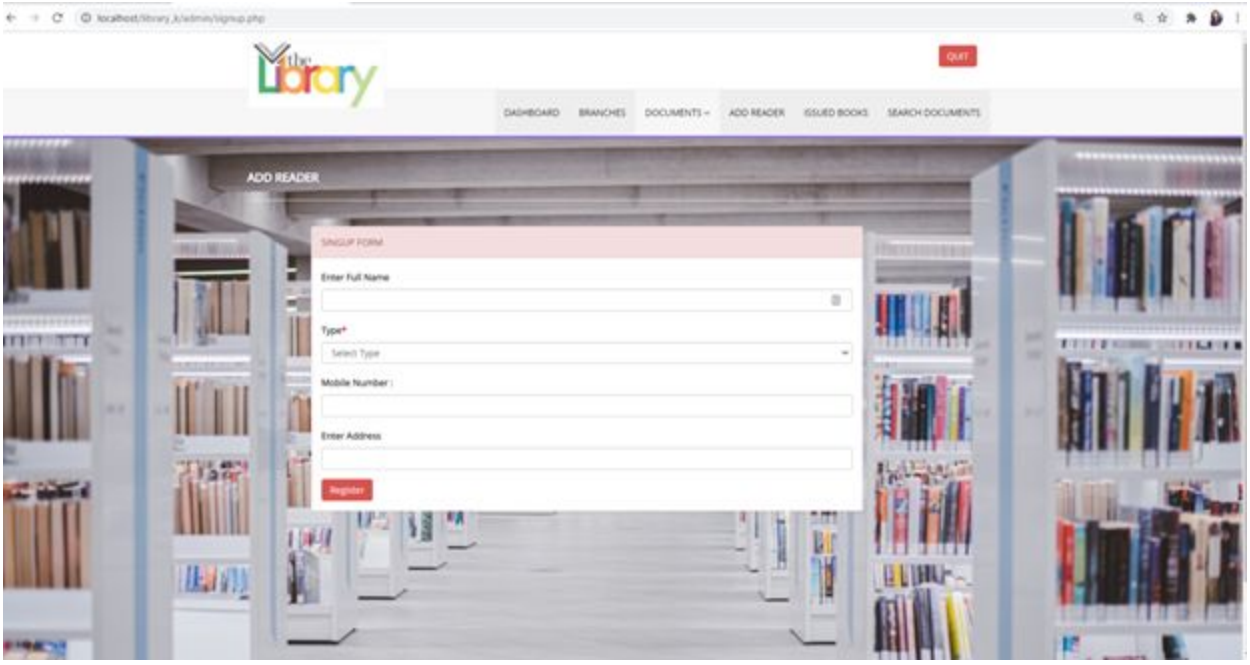
Add a new journal:



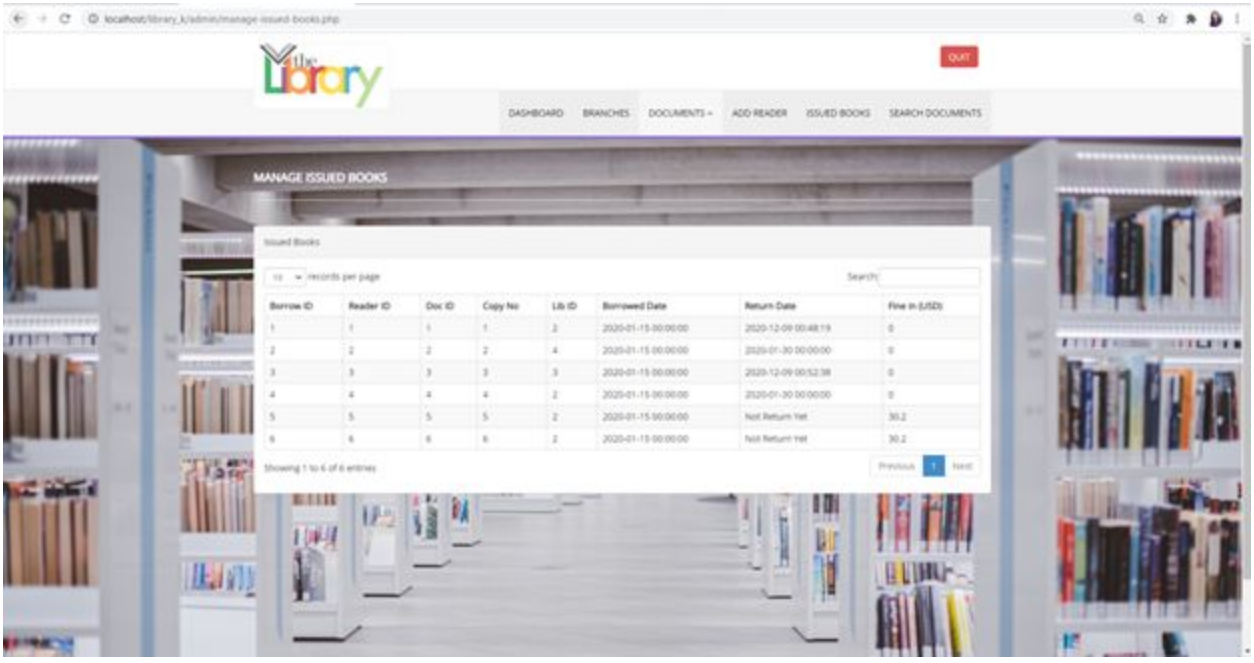
Add new proceedings:



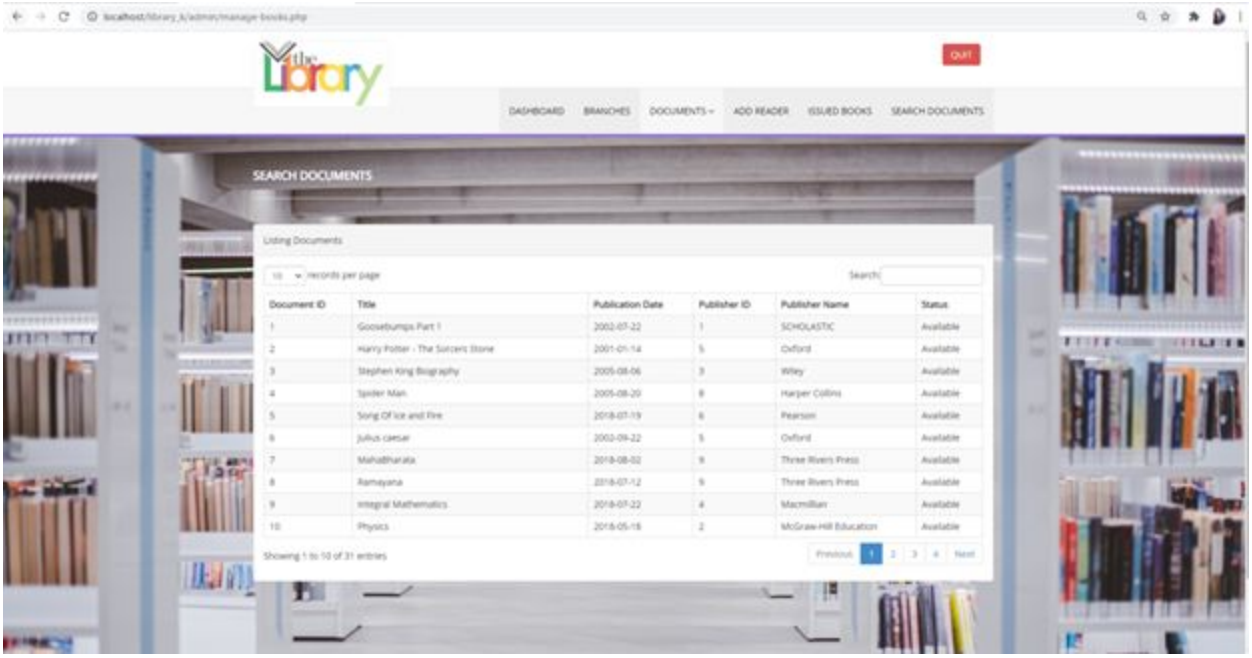
Add new reader:



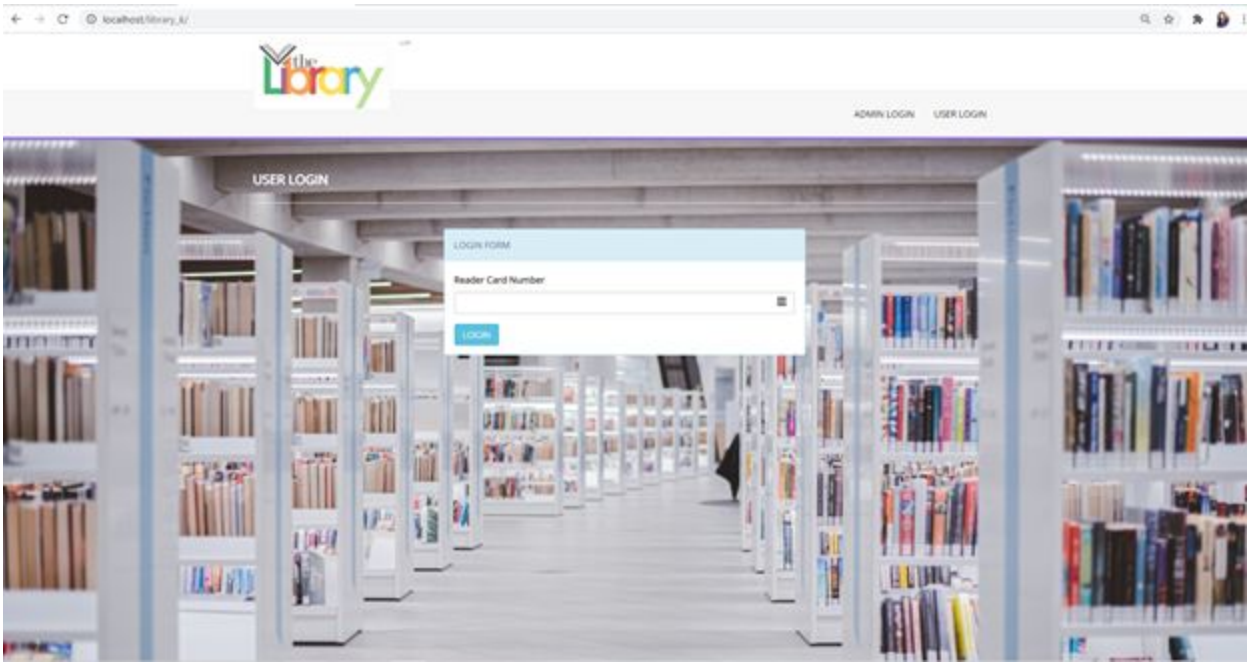
Details of the issued books:



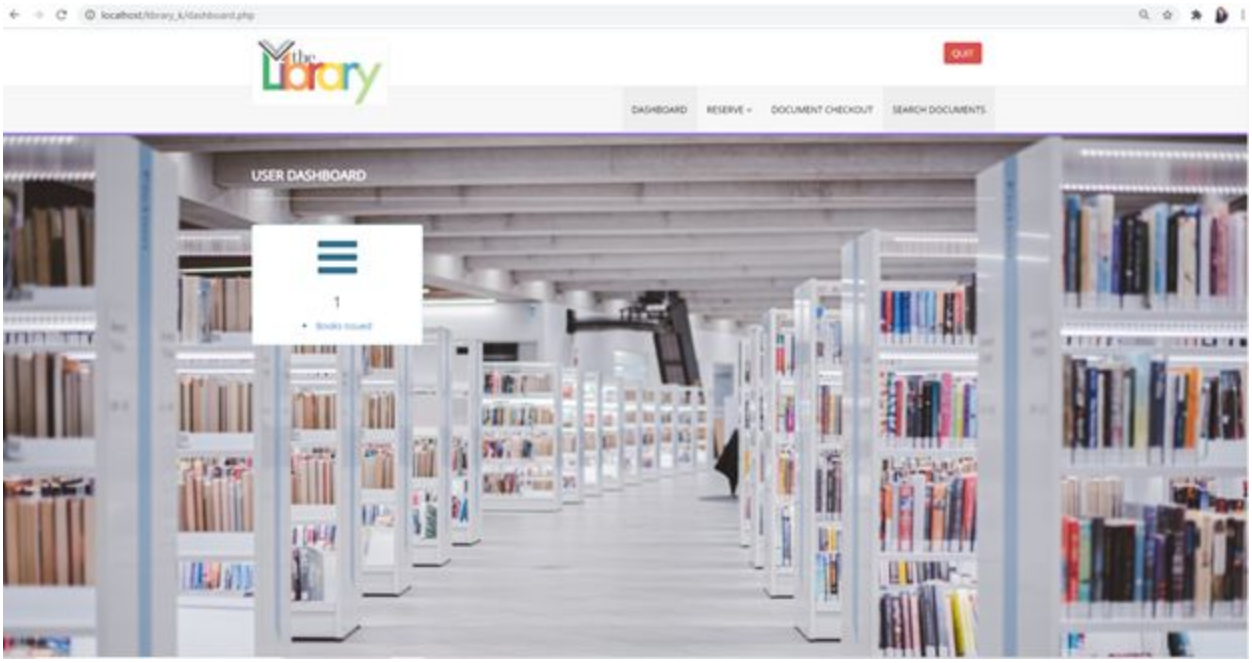
Search Document:



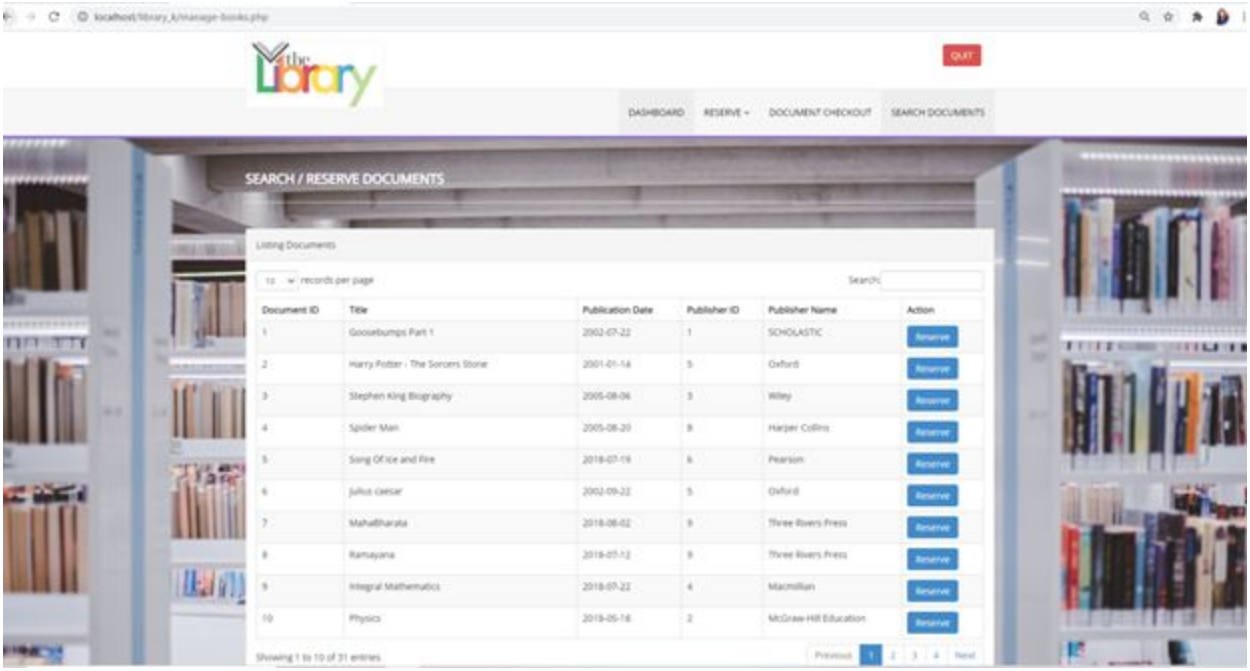
Login form for the reader:



Books issued:



Document search and document reserve:



Document checkout: