CS 631 Database Management Systems Design

City Library Management System

Group 4

Deliverable 3

Group members

Karthik Ganeshula – kg496

Sai Venkata Krishna Kopparapu - sk2924

Vineeth Reddy Maddula - vm477

CITY LIBRARY MANAGEMENT SYSTEM

IMPLEMENTATION

City Library System is an online library management system that manages library resources or literary works such as books, magazines, articles, journals and reports. The users are able to search, read, save and reserve their desired literature through this web application. It can be perceived as a reader's stack of reading materials. The library system consists of an administrator to manage the library and the readers. The administrator keeps track of all the users and uploads the literary works and stores them in the appropriate databases . He is responsible for the functionalities like adding a document copy and checking its status and adding a new reader and branch. The works are classified with respect to their genre. Coming to the SQL part we have the database tables for authors, books, journals, conference proceedings, publisher, admin and branch etc. Administrator is the only person who is responsible to upload his preferred documents and readers have no access to uploading the documents except using the document resources that are available in the library system.

City Library System aims to assist readers in managing their preferred literary works such as books, magazines, journals, newspapers, etc. in order to have them available for the purposes of reading (both online and offline), safekeeping of the literary works and sharing the resources so as to pave the way for the database to grow as the number of users increases .They have the functionalities like the Search a document by ID, title, or publisher name. He can check Document checkout Document return and Document reserve. He can also Compute fine for a document copy borrowed based on the current date. He can Print the list of documents reserved by a reader and their status. He can also Print the document id and document titles of documents published by the publisher This system has been implemented and developed in PHP language and MySQL has been used to create and manage databases. Furthermore, Xampp web server has been employed to execute the code.

This system needs a robust database to store users' preferred books, articles, journals and other literary material. PHP MySQL has been used to create databases to fulfill the requirements of storage. This system contains simple design and a high level of user-friendliness. Each page in this application has a 'Quit' button that takes the user to the homepage and 'settings' button that consists of a checklist of different options such as Profile that contains user information and Sign out to log out of the application. Additionally, this system requires a screen - a monitor or touchscreen- for display and a keyboard or keypad for input. The user may need a printer to print the downloadable files and, optionally, a mouse for pointing and selecting purposes. No additional devices or interfaces are required.

The application operates efficiently on any operating system including Windows, Linux and Mac and on any hardware platform such as desktop, laptop, tablets or smart phones. This product requires minimal external hardware and peripheral devices. The product requires a secured network connection and interrupted internet access.

Problems encountered:

This is our first time to use PHP to develop a website . There are too many things for us to deal with during our development. XAMPP is an AMP stack which lets you install apache Mysql and PHP in your computer together with some useful software .It is designed to make easy connection. We experienced that developing a fancy interface for our website is too difficult.

We also faced a problem regarding writing the codes to Print top 10 most frequent borrowers in a branch and the number of books each has borrowed, print top 10 most borrowed books in a branch, print the 10 most popular books of the year and find the average fine paid per reader and we also find difficulties regarding the guest editor implementation

User Guide :

This document elucidates the requirements, specifications and features of *City Library System* which aims to assist readers in managing their preferred literary works such as books, magazines, journals, newspapers, etc. in order to have them available for the purposes of reading (both online and offline), safekeeping of the literary works and sharing the resources so as to pave the way for the database to grow as the number of users increases.

The file paths of all literary works available on this system are stored in the databases while the actual files are stored in system folders and they are available in pdf format. The library records information about all documents that are available in its system. Each document is identified by a unique number (DocumentId) It also has a title, a publisher, a publication date, and can be one of three different types: book, journal volume, or conference proceedings.

For each document type we stored additional information. For instance, for books we need to record their ISBN, and for conference proceedings we need to record the date and location of the conference, and the proceedings chair. A journal volume has a number and can have several issues (up to 10). A journal issue has a number (1, 2, 3..) that uniquely identifies the journal issues of the same journal volume. A journal has a single chief editor who usually lasts for many years and supervises many consecutive journal volumes. Chief editors are identified by their ID and their name has to be recorded. Journal issues might have one or more guest editors and a scope that need to be recorded. The library system contains several branches, which are identified by a number (LibId). We also need to store the name and the location of each branch. Each branch of

the library holds a number of copies of a particular document. Each copy of the same document kept by the same library branch is numbered from 1 to n. The total number of copies of each document in the library is needed .This system has been implemented and developed in PHP language and MySQL has been used to create and manage databases. Furthermore, Xampp web server has been employed to execute the code.
The system is organized into two realms: user and administrator. The users are required to first register themselves by providing valid details to the administrator and the administrator only has the permission to add a new reader, the administrator logs the user in the system and permits users to maintain and manage their profile. Users are granted access to all the resources entitled to them. They may search for a specific book or search a document by ID, title, or publisher name. They can check the document checkout, document return and document reserve. The reader can compute fine for a document copy borrowed by a reader based on the current date. He can also print the list of documents reserved by a reader and their status. He can also print the document id and document titles of documents published by a publisher.

This product can be installed on a desktop or a laptop with Windows Operating System. It needs a secured and uninterrupted internet access. The users must have a substantial amount of memory or storage space in order to save and download resources.

Following are the installation instructions that must be followed for installation of the product:

1. The user must install Xampp server. The Xampp server contains MySql and Apache, which are used for creating databases and implementing the necessary features of the web pages.

2. The user must create a database named 'library' in MySQL and it must contain tables such as
'books' with attributes name, author, isbn; 'users'with attributes userid, username, password, date of birth, email; 'categories' or genre, 'languages' which specifies the language the book is published in, and 'admin' with attributes admin_id, admin_username and admin_password. Databases and the relationships between entities should be created only after verifying the
Entity-Relationship Diagrams. The user must assign primary keys and/or unique keys to specific fields of the table.

3. The user may download available templates from the internet to enhance the overall design for web pages. Codes can be written in the templates and templates, in turn, should be modified in order to meet the needs of the system.

4. The user should install Notepad++ or any IDE such as PHP Eclipse in order to write, run, test and debug their codes. In the 'xampp' folder in C:/ drive in 'My PC' section of Windows 8.1, the user must store the php files in the 'htdocs' folder. Files must be grouped and classified with respect to their purpose, features, functions and users.

5. The user must have session management and establish connections by having a separate file dedicated for these features. These files can be included in all the other files of the source code.

6. The user must create disparate folders for storing resources, images, and codes pertaining to either the regular user or the system administrator.

7. While testing, codes should run in Xampp server and corrections must be made at specific line numbers in case of error conditions.

To uninstall the product, please follow the following steps:

1. Delete all the tables and databases from MySql.

2. Delete all php files and folders containing the php files in htdocs folder of Xampp server.

3. Delete all the downloaded materials.

3. SQL COMMANDS FOR CREATING TABLES:

```
--
-- Table structure for table `admin`
--
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 DEFAULT '0000-00-00 00:00:00' ON UPDATE CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `authors`
--
CREATE TABLE `authors` (
`AuthorId` int(11) NOT NULL,
`AuthorName` varchar(159) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `publishers`
--
CREATE TABLE `publishers` (
  `PubId` int(11) NOT NULL,
  `PubName` varchar(159) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `editors`
--
CREATE TABLE `editors` (
  `EditorId` int(11) NOT NULL,
  `EditorName` varchar(159) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `guesteditors`
--
CREATE TABLE `guesteditors` (
  `GEditorId` int(11) NOT NULL,
  `GEditorName` varchar(159) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `chiefeditors`
--
CREATE TABLE `chiefeditors` (
  `CEditorId` int(11) NOT NULL,
  `CEditorName` varchar(159) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `branch`
--
CREATE TABLE `branch` (
   `LibId` int(11) NOT NULL,
   `LibName` varchar(159) DEFAULT NULL,
   `Location` varchar(159) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `books`
--
CREATE TABLE `books` (
   `DocId` int(11) NOT NULL,
   `Title` varchar(255) DEFAULT NULL,
   `AuthorId` int(11) DEFAULT NULL,
   `ISBNNumber` int(11) DEFAULT NULL,
   `PubDate` DATE DEFAULT NULL,
   `PublisherId` int(11) DEFAULT NULL,
   `LibId` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Table structure for table `journal`
--
CREATE TABLE `journal` (
  `DocId` int(11) NOT NULL,
  `Title` varchar(255) DEFAULT NULL,
  `EditorId` int(11) DEFAULT NULL,
  `VolNo` int(11) DEFAULT NULL,
  `IssNo` int(11) DEFAULT NULL,
  `GEditorId` int(11) DEFAULT NULL,
  `PubDate` DATE DEFAULT NULL,
  `PublisherId` int(11) DEFAULT NULL,
  `LibId` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `proceedings`
--
CREATE TABLE `proceedings` (
  `DocId` int(11) NOT NULL,
  `Title` varchar(255) DEFAULT NULL,
  `CEditorId` int(11) DEFAULT NULL,
  `CLocation` varchar(255) DEFAULT NULL,
  `PubDate` DATE DEFAULT NULL,
  `PublisherId` int(11) DEFAULT NULL,
  `LibId` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `documents`
--
CREATE TABLE `documents` (
  `DocId` int(11) NOT NULL,
  `Title` varchar(255) DEFAULT NULL,
  `PubDate` DATE DEFAULT NULL,
  `PublisherId` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `copies`
--
CREATE TABLE `copies` (
  `DocId` int(11) NOT NULL,
  `CopyNo` int(11) DEFAULT NULL,
  `LibId` int(11) DEFAULT NULL,
  `Status` int(1) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `borrows`
--
CREATE TABLE `borrows` (
  `BorrowId` int(11) NOT NULL,
  `BorrowDate` DATE NOT NULL,
  `ReturnDate` DATE DEFAULT NULL,
  `ReaderId` int(11) NOT NULL,
  `DocId` int(11) NOT NULL,
  `CopyNo` int(11) NOT NULL,
  `LibId` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `reserve`
--
CREATE TABLE `reserve` (
  `ReserveId` int(11) NOT NULL,
  `ReserveDate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `ReaderId` int(11) NOT NULL,
  `DocId` int(11) NOT NULL,
  `CopyNo` int(11) NOT NULL,
  `LibId` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `reader`
--
CREATE TABLE `reader` (
  `ReaderId` int(11) NOT NULL,
  `ReaderName` varchar(120) DEFAULT NULL,
  `RType` varchar(120) DEFAULT NULL,
  `MobileNumber` char(11) DEFAULT NULL,
  `Address` varchar(120) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The above are the SQL commands for creating the tables and for updating the primary keys the following are the commands

```
--
-- Indexes for table `admin`
--
ALTER TABLE `admin`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `authors`
--
ALTER TABLE `authors`
  ADD PRIMARY KEY (`AuthorId`);

--
-- Indexes for table `editors`
--
ALTER TABLE `editors`
  ADD PRIMARY KEY (`EditorId`);

--
-- Indexes for table `guesteditors`
--
ALTER TABLE `guesteditors`
  ADD PRIMARY KEY (`GEditorId`);

--
-- Indexes for table `chiefeditors`
--
ALTER TABLE `chiefeditors`
  ADD PRIMARY KEY (`CEditorId`);

--
-- Indexes for table `publishers`
--
ALTER TABLE `publishers`
  ADD PRIMARY KEY (`PubId`);


--
-- Indexes for table `branch`
--
ALTER TABLE `branch`
  ADD PRIMARY KEY (`LibId`);
```

```
--
-- Indexes for table `books`
--
ALTER TABLE `books`
  ADD PRIMARY KEY (`DocId`);

--
-- Indexes for table `journal`
--
ALTER TABLE `journal`
  ADD PRIMARY KEY (`DocId`);

--
-- Indexes for table `proceedings`
--
ALTER TABLE `proceedings` ADD PRIMARY
  KEY (`DocId`);

--
-- Indexes for table `documents`
--
ALTER TABLE `documents`
  ADD PRIMARY KEY (`DocId`);

--
-- Indexes for table `borrows`
--
ALTER TABLE `borrows`
  ADD PRIMARY KEY (`BorrowId`);

--
-- Indexes for table `reserve`
--
ALTER TABLE `reserve`
  ADD PRIMARY KEY (`ReserveId`);

--
-- Indexes for table `reader`
--
ALTER TABLE `reader`
  ADD PRIMARY KEY (`ReaderId`);
```

We can auto-increment the values of the each primary key using the following SQL commands. The SQL commands are as the follows

```
--
-- AUTO_INCREMENT for table `admin`
--
ALTER TABLE `admin`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT for table `authors`
--
ALTER TABLE `authors`
  MODIFY `AuthorId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

--
-- AUTO_INCREMENT for table `books`
--
ALTER TABLE `books`
  MODIFY `DocId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;

--
-- AUTO_INCREMENT for table `journal`
--
ALTER TABLE `journal`
  MODIFY `DocId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;

--
-- AUTO_INCREMENT for table `proceedings`
--
ALTER TABLE `proceedings`
  MODIFY `DocId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;

--
-- AUTO_INCREMENT for table `borrows`
--
ALTER TABLE `borrows`
  MODIFY `BorrowId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

--
-- AUTO_INCREMENT for table `reserve`
--
ALTER TABLE `reserve`
```

```
  MODIFY `ReserveId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

--
-- AUTO_INCREMENT for table `reader`
--
ALTER TABLE `reader`
  MODIFY `ReaderId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6; COMMIT;
```

4. SQL COMMANDS FOR INSERTING VALUES INTO TABLES:

```
--

-- Dumping data for table `admin`

--

INSERT INTO `admin` (`id`, `FullName`, `AdminEmail`, `UserName`, `Password`,
`updationDate`) VALUES

(007, 'Meghashyam Senapathi', 'meghashyamsenapathi004@gmail.com', 'megha',
'f925916e2754e5e03f75dd58a5733251', '2017-07-16 18:11:42');



--

-- Dumping data for table `authors`

--

INSERT INTO `authors` (`AuthorId`, `AuthorName`) VALUES (1, 'RL STINE'),

(2, 'J K ROWLING'), (3, 'STAN

LEE'),

(4, 'STEPHEN KING'),

(5, 'GEORGE R R MARTIN'),

(6, 'WILLIAM SHAKESPHERE'), (7, 'VYASA'),

(8, 'VALMIKI'),

(9, 'R D SHARMA'),
(10, 'H C VERMA');


--
-- Dumping data for table `publishers`
--
INSERT INTO `publishers` (`PubId`, `PubName`) VALUES (1, 'SCHOLASTIC'),
(2, 'McGraw-Hill Education'), (3,
'Wiley'),
(4, 'Macmillian'), (5,
'Oxford'),
(6, 'Pearson'),
(7, 'Penguin Random'), (8,
'Harper Collins'),
(9, 'Three Rivers Press'), (10,
'Elsevier');


--
-- Dumping data for table `editors`
--
INSERT INTO `editors` (`EditorId`, `EditorName`) VALUES (1, 'Jason Epstein'),
(2, 'Bill Buford'), (3,
'Jim Baen'), (4,
'Donald A'),
(5, 'Robert B Silvers');
```

```
--

-- Dumping data for table `guesteditors`

--

INSERT INTO `guesteditors` (`GEditorId`, `GEditorName`) VALUES (1, 'Dan Mallory'),

(2, 'Stephen Jones'), (3,

'Martin H'),

(4, 'Lou Aronica'), (5,

'Lester Ray');


--

-- Dumping data for table `chiefeditors`

--

INSERT INTO `coeditors` (`CEditorId`, `CEditorName`) VALUES (1, 'Dan Mallory'),

(2, 'Stephen Jones'), (3,

'Martin H'),

(4, 'Lou Aronica'), (5,

'Lester Ray');


--

-- Dumping data for table `branch`

--

INSERT INTO `branch` (`LibId`, `LibName`, `Location`) VALUES (1, 'Martin Luther King

Library', 'New York'),

(2, 'Van Houten Library', 'Newark'), (3, 'Central

King Library', 'Kearny'), (4, 'JFK Library',

'Edison'),
(5, 'Kingsman Library', 'Jersey City');


--
-- Dumping data for table `books`
--
INSERT INTO `books` (`DocId`, `Title`, `AuthorId`, `ISBNNumber`, `PubDate`, `PublisherId`,
`LibId`) VALUES
(1, 'Goosebumps Part 1', 1, 1256, '2002-07-22', 1, 1),
(2, 'Harry Potter - The Sorcers Stone', 2, 12565, '2001-01-14', 5, 2), (3, 'Stephen King
Biography', 4, 12556, '2005-08-06', 3, 3),
(4, 'Spider Man ', 3, 12576, '2005-08-20', 8, 4),
(5, 'Song Of Ice and Fire', 5, 15256, '2018-07-19', 6, 5), (6, 'Julius
caesar', 6, 12526, '2002-09-22', 5, 1),
(7, 'MahaBharata', 7, 51256, '2018-08-2', 9, 2), (8,
'Ramayana', 8, 61256, '2018-07-12', 9, 3),
(9, 'Integral Mathematics', 9, 91256, '2018-07-22', 4, 4), (10, 'Physics',
10, 81256, '2018-05-18', 2, 5);


--
-- Dumping data for table `journal`
--
INSERT INTO `journal` (`DocId`, `Title`, `EditorId`, `VolNo`, `IssNo`, `GEditorId`, `PubDate`,
`PublisherId`, `LibId`) VALUES
(11, 'Elsevier - Neural Networks', 1, 1, 4, 1, '2002-07-22', 1, 1), (12, 'IEEE - Machine
Learning', 2, 12, 5, 2, '2001-01-14', 5, 2),
(13, 'Oxford University Press - Bioinformatics', 4, 56, 12, 3, '2005-08-06', 3, 3), (14, 'Springer -
Information Technology ', 3, 15, 2, 4, '2018-08-20', 8, 4),
(15, 'Microtome - Machine Learning', 5, 4, 8, 5, '2018-07-19', 6, 5);
```

```
--
-- Dumping data for table `Proceedings`
--
INSERT INTO `proceedings` (`DocId`, `Title`, `CEditorId`, `CLocation`, `PubDate`,
`PublisherId`, `LibId`) VALUES
(16, 'Elsevier', 1, 'Paris', '2018-07-22', 1, 1), (17, 'IEEE', 2,
'Tokyo', '2018-01-14', 5, 2),
(18, 'Oxford University Press', 4, 'Mumbai', '2005-08-06', 3, 3), (19, 'Springer', 3,
'London', '2005-08-20', 8, 4),
(20, 'Microtome', 5, 'Washington', '2018-07-19', 6, 5);


--
-- Dumping data for table `documents`
--
INSERT INTO `documents` (`DocId`, `Title`, `PubDate`, `PublisherId`) VALUES (1, 'Goosebumps Part 1',
'2002-07-22', 1),
(2, 'Harry Potter - The Sorcers 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);


--
-- Dumping data for table `copies`
--
INSERT INTO `copies` (`DocId`, `CopyNo`, `LibId`, `Status`) VALUES (1, 1, 1, 1),
(1, 2, 2, 1), (1, 3,
3, 1), (1, 4, 4, 1),
(1, 5, 5, 1), (2, 1,
1, 1), (2, 2, 2, 1),
(2,  3,  3,  1),  (2,
4, 4, 1), (2, 5, 5,
1), (3, 1, 1, 1), (3,
2, 2, 1), (3, 3, 3,
1),  (3,  4,  4,  1),
(3, 5, 5, 1), (4, 1,
1, 1), (4, 2, 2, 1),
```

(4, 3, 3, 1),

(4, 4, 4, 1), (4, 5, 5, 1), (5, 1, 1, 1), (5, 2, 2, 1), (5, 3, 3, 1), (5, 4, 4, 1),

(5, 5, 5, 1),

(6, 1, 1, 1), (6, 2, 2, 1), (6, 3, 3, 1), (6, 4, 4, 1), (6, 5, 5, 1), (7, 1, 1, 1), (7, 2, 2, 1), (7, 3, 3, 1), (7, 4, 4, 1), (7, 5, 5, 1), (8, 1, 1, 1), (8, 2, 2, 1), (8, 3, 3, 1), (8, 4, 4, 1), (8, 5, 5, 1), (9, 1, 1, 1), (9, 2, 2, 1), (9, 3, 3, 1), (9, 4, 4, 1), (9, 5, 5, 1)

(10, 1, 1, 1),

(10, 2, 2, 1), (10, 3, 3, 1), (10, 4, 4, 1), (10, 5, 5, 1), (11, 1, 1, 1), (11, 2, 2, 1), (11, 3, 3, 1),

(11, 4, 4, 1), (11, 5, 5, 1), (12, 1, 1, 1), (12, 2, 2, 1), (12, 3, 3, 1), (12, 4, 4, 1), (12, 5, 5, 1), (13, 1, 1, 1), (13, 2, 2, 1), (13, 3, 3, 1), (13, 4, 4, 1), (13, 5, 5, 1), (14, 1, 1, 1), (14, 2, 2, 1), (14, 3, 3, 1), (14, 4, 4, 1), (14, 5, 5, 1), (15, 1, 1, 1), (15, 2, 2, 1), (15, 3, 3, 1)

(15, 4, 4, 1),
(15, 5, 5, 1), (16, 1,
1, 1), (16, 2, 2, 1),
(16, 3, 3, 1), (16,
4, 4, 1), (16, 5,, 1),
(17, 1, 1, 1),

(17, 2, 2, 1), (17, 3,
3, 1), (17, 4, 4, 1),
(17, 5, 5, 1), (18, 1,
1, 1), (18, 2, 2, 1),
(18, 3, 3, 1), (18, 4,
4, 1), (18, 5, 5, 1),
(19, 1, 1, 1), (19, 2, 2,
1), (19, 3, 3, 1), (19,
4, 4, 1), (19, 5, 5, 1),
(20, 1, 1, 1), (20, 2,
2, 1), (20, 3, 3, 1),
(20, 4, 4, 1), (20,
5, 5, 1);

```
--

-- Dumping data for table `borrows`

--

INSERT INTO `borrows` (`BorrowId`, `BorrowDate`, `ReturnDate`, `ReaderId`, `DocId`,
`CopyNo`, `LibId`) VALUES

(1, '2019-01-15', '2019-01-30', 1, 1, 1, 1), (2, '2019-01-

15', '2019-01-30', 1, 2, 1, 1),

(3, '2019-01-15', NULL, 1, 9, 1, 1),

(4, '2019-01-15', '2019-01-30', 5, 7, 1, 1), (5, '2019-01-

15', NULL, 5, 6, 1, 1),

(6, '2019-01-15', NULL, 5, 4, 1, 1);


--

-- Dumping data for table `reserve`

--

INSERT INTO `reserve` (`ReserveId`, `ReserveDate`, `ReaderId`, `DocId`, `CopyNo`, `LibId`) VALUES

(1, '2019-01-14 18:35:25', 1, 1, 2, 2), (2, '2019-01-14

18:35:25', 1, 2, 2, 2), (3, '2019-01-14 18:35:25', 1, 9,

2, 2), (4, '2019-01-14 18:35:25', 5, 7, 2, 2), (5,

'2019-01-14 18:35:25', 5, 6, 2, 2), (6, '2019-01-14

18:35:25', 5, 4, 2, 2);


--

-- Dumping data for table `reader`

--

INSERT INTO `reader` (`ReaderId`, `ReaderName`, `RType`, `MobileNumber`, `Address`) VALUES
```

```
(1, ‘Karthik’, ‘Student’, ‘9733420342’, ‘306 Sussex St’)

(2, ‘Christina’, ‘Staff’, ‘85690025', '97 Hudson St’)

(3, 'Vineeth', 'Senior Citizen', '59874527', '309 Sussex St')

(4, ‘Venkat', 'Other', '585856224', '127 Warren St')

(5, 'Likith', 'Student', '672423754', '739 Harrison Av');
```

Source Code:

The Source code for the admin login page is as follows:

```
  <?php
session_start();
error_reporting(0);
include('includes/config.php');
if($_SESSION['alogin']!=''){
$_SESSION['alogin']='';
}
if(isset($_POST['login']))
{
 //code for captacha verification
//if ($_POST["vercode"] != $_SESSION["vercode"] OR $_SESSION["vercode"]=='')  {
//        echo "<script>alert('Incorrect verification code');</script>" ;
//    }
//        else {

$username=$_POST['username'];
$password=md5($_POST['password']);
$sql ="SELECT UserName,Password FROM admin WHERE UserName=:username and Password=:password";
$query= $dbh -> prepare($sql);
$query-> bindParam(':username', $username, PDO::PARAM_STR);
$query-> bindParam(':password', $password, PDO::PARAM_STR);
$query-> execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
$_SESSION['alogin']=$_POST['username'];
echo "<script type='text/javascript'> document.location ='admin/dashboard.php'; </script>";
} else{
echo "<script>alert('Invalid Details');</script>";
}
//}
}
```

Reader login page:

```
<?php
session_start();
error_reporting(0);
include('includes/config.php');
if($_SESSION['login']!=''){
$_SESSION['login']='';
}
if(isset($_POST['login']))
{
 //code for captacha verification
//if ($_POST["vercode"] != $_SESSION["vercode"] OR $_SESSION["vercode"]=='')  {
//        echo "<script>alert('Incorrect verification code');</script>" ;
//    }
//        else {

$username=$_POST['username'];
//$password=md5($_POST['password']);
$sql ="SELECT ReaderId FROM reader WHERE ReaderId=:username";
//$sql ="SELECT ReaderId,MobileNumber FROM reader WHERE ReaderId=:username and MobileNumber=:password";
$query= $dbh -> prepare($sql);
$query-> bindParam(':username', $username, PDO::PARAM_STR);
//$query-> bindParam(':password', $password, PDO::PARAM_STR);
$query-> execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
$_SESSION['login']=$_POST['username'];
echo "<script type='text/javascript'> document.location ='dashboard.php'; </script>";
} else{
echo "<script>alert('Invalid Details');</script>";
}
//}
}
```

Adding a Document:

```
<?php
session_start();

include('includes/config.php');
error_reporting(0);

if(strlen($_SESSION['alogin'])==0)
    {
header('location:index.php');
}
else{

if(isset($_POST['add']))
{
$count_my_page = ("documentid.txt");
$hits = file($count_my_page);
$hits[0] ++;
$fp = fopen($count_my_page , "w");
fputs($fp , "$hits[0]");
fclose($fp);
$DocId= $hits[0];
$title=$_POST['title'];
//$category=$_POST['category'];
$authorid=$_POST['authorid'];
$isbn=$_POST['isbn'];
$pdate=$_POST['pdate'];
$pubid=$_POST['pubid'];
$libid=$_POST['libid'];
$copies=$_POST['copies'];
$price=$_POST['price'];
$sql="INSERT INTO documents(DocId,Title,PubDate,PublisherId) VALUES(:DocId,:title,'2019-05-07',:pubid)";
//$sql="INSERT INTO documents(DocId,Title,AuthorId,ISBNNumber,PubDate,PublisherId,LibId)
VALUES(:DocId,:title,:authorid,:isbn,:pdate,:pubid,:libid)";
$query = $dbh->prepare($sql);
$query->bindParam(':DocId',$DocId,PDO::PARAM_STR);
$query->bindParam(':title',$title,PDO::PARAM_STR);
$query->bindParam(':pubid',$pubid,PDO::PARAM_STR);
/*$query->bindParam(':category',$category,PDO::PARAM_STR);
$query->bindParam(':authorid',$authorid,PDO::PARAM_STR);
$query->bindParam(':isbn',$isbn,PDO::PARAM_STR);
$query->bindParam(':pdate',$pdate,PDO::PARAM_STR);
$query->bindParam(':libid',$libid,PDO::PARAM_STR);
$query->bindParam(':copies',$copies,PDO::PARAM_STR);
*/
$query->execute();
$lastInsertId = $dbh->lastInsertId();
if($lastInsertId)
{
$_SESSION['msg']="Book Listed successfully";
header('location:manage-books.php');
}
else
{
header('location:manage-books.php');
}

}
```

Searching a Document:

```
<?php $sql= "SELECT documents.DocId,documents.Title,documents.PubDate,documents.PublisherId,publishers.PubName
from documents INNER join publishers on documents.PublisherId = publishers.PubId ";
//"SELECT documents.DocId,documents.Title,documents.PubDate,documents.PublisherId from documents join publishers
on publishers.PubId = documents.PublisherId";
//"SELECT
tblbooks.BookName,tblcategory.CategoryName,tblauthors.AuthorName,tblbooks.ISBNNumber,tblbooks.BookPrice,tblbooks.id as bookid from tblbooks join tblcategory on tblcategory.id=tblbooks.CatId join tblauthors on
tblauthors.id=tblbooks.AuthorId";
$query = $dbh -> prepare($sql);
$query->execute();
$rsults=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
```

Adding a new reader:

```
                    <?php
session_start();
include('includes/config.php');
error_reporting(0);
if(isset($_POST['signup']))
{
//code for captach verification
/* if ($_POST["vercode"] != $_SESSION["vercode"] OR $_SESSION["vercode"]=='')  {
      echo "<script>alert('Incorrect verification code');</script>" ;
   }
      else {   */
//Code for student ID
$count_my_page = ("readerid.txt");
$hits = file($count_my_page);
$hits[0] ++;
$fp = fopen($count_my_page , "w");
fputs($fp , "$hits[0]");
fclose($fp);
$StudentId= $hits[0];
$fname=$_POST['fname'];
$type=$_POST['type'];
$mobileno=$_POST['mobileno'];
$address=$_POST['address'];
//$password=md5($_POST['password']);
//$status=1;
$sql="INSERT INTO reader(ReaderId,ReaderName,RType,MobileNumber,Address)
VALUES(:StudentId,:fname,:type,:mobileno,:address)";
$query = $dbh->prepare($sql);
$query->bindParam(':StudentId',$StudentId,PDO::PARAM_STR);
$query->bindParam(':fname',$fname,PDO::PARAM_STR);
$query->bindParam(':type',$type,PDO::PARAM_STR);
$query->bindParam(':mobileno',$mobileno,PDO::PARAM_STR);
$query->bindParam(':address',$address,PDO::PARAM_STR);
//$query->bindParam(':password',$password,PDO::PARAM_STR);
//$query->bindParam(':status',$status,PDO::PARAM_STR);
$query->execute();
$lastInsertId = $dbh->lastInsertId();
if($lastInsertId)
{
echo '<script>alert("Registration successfull and The Reader Id is  "+"'.$StudentId.'")</script>';
}
else
{
echo "<script>alert('Something went wrong. Please try again');</script>";
}
}

?>
```

Printing Branch information:

```
<?php
session_start();
error_reporting(0);
include('includes/config.php');
if(strlen($_SESSION['alogin'])==0)
    {
header('location:index.php');
}
else{
if(isset($_GET['del']))
{
$id=$_GET['del'];
$sql = "delete from branchtbl  WHERE LibId=:id";
$query = $dbh->prepare($sql);
$query -> bindParam(':id',$id, PDO::PARAM_STR);
$query -> execute();
$_SESSION['delmsg']="Branch deleted";
header('location:manage-branch.php');

}
```

Most frequently borrowed books:

```
<?php $sql = "SELECT borrows.*,reader.*, COUNT(BorrowId) AS post_count from borrows LEFT JOIN reader on borrows.ReaderId = reader.ReaderId GROUP BY borrows.ReaderId ORDER BY post_count DESC LIMIT 10";
//$sql = "SELECT * from reader";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{ ?>
```

Top 10 most borrowed books:

```
<?php $sql = $sql = "SELECT borrows.*,documents.*, COUNT(BorrowId) AS post_count from borrows LEFT JOIN documents on borrows.DocId = documents.DocId GROUP BY borrows.DocId ORDER BY post_count DESC LIMIT 10";
// "SELECT  * from documents ORDER BY PubDate DESC  limit 10";
//"SELECT documents.DocId,documents.Title,documents.PubDate,documents.PublisherId from documents join publishers on publishers.PubId = documents.PublisherId";
//"SELECT
tblbooks.BookName,tblcategory.CategoryName,tblauthors.AuthorName,tblbooks.ISBNNumber,tblbooks.BookPrice,tblbooks.id as bookid from  tblbooks join tblcategory on tblcategory.id=tblbooks.CatId join tblauthors on tblauthors.id=tblbooks.AuthorId";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{           ?>
```

10 most popular books of the year:

```
<?php $sql = "SELECT  * from documents ORDER BY PubDate DESC  limit 10";
//"SELECT documents.DocId,documents.Title,documents.PubDate,documents.PublisherId from documents join publishers
on publishers.PubId = documents.PublisherId";
//"SELECT
tblbooks.BookName,tblcategory.CategoryName,tblauthors.AuthorName,tblbooks.ISBNNumber,tblbooks.BookPrice,tblbooks.i
d as bookid from  tblbooks join tblcategory on tblcategory.id=tblbooks.CatId join tblauthors on
tblauthors.id=tblbooks.AuthorId";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{               ?>
```

Searching a document by a user:

```
<?php $sql = "SELECT documents.DocId,documents.Title,documents.PubDate,documents.PublisherId,publishers.PubName
from documents INNER join publishers on documents.PublisherId = publishers.PubId  ";
//"SELECT documents.DocId,documents.Title,documents.PubDate,documents.PublisherId from documents join publishers
on publishers.PubId = documents.PublisherId";
//"SELECT
tblbooks.BookName,tblcategory.CategoryName,tblauthors.AuthorName,tblbooks.ISBNNumber,tblbooks.BookPrice,tblbooks.i
d as bookid from  tblbooks join tblcategory on tblcategory.id=tblbooks.CatId join tblauthors on
tblauthors.id=tblbooks.AuthorId";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{               ?>
```

Document checkout:

```
<?php
$sid=$_SESSION['login'];
$sql1 ="SELECT BorrowId from borrows where ReaderID=:sid";
$query1 = $dbh -> prepare($sql1);
$query1->bindParam(':sid',$sid,PDO::PARAM_STR);
$query1->execute();
$results1=$query1->fetchAll(PDO::FETCH_OBJ);
$issuedbooks=$query1->rowCount();
?>
```

Document Reserve:

```
<?php
session_start();
error_reporting(0);
include('includes/config.php');
if(strlen($_SESSION['login'])==0)
{
header('location:index.php');
}
else{
if(isset($_GET['del']))
{
$docid=$_GET['del'];
$sql1 ="SELECT * from copies where DocId=:docid";
$query1 = $dbh -> prepare($sql1);
$query1->bindParam(':docid',$docid,PDO::PARAM_STR);
$query1->execute();
$results=$query1->fetchAll(PDO::FETCH_OBJ);
$issuedbooks=$query1->rowCount();
$libid = $results->LibId;
$copy = $results->CopyNo;
$rid=$_GET['login'];
$count_my_page = ("reserveid.txt");
$hits = file($count_my_page);
$hits[0] ++;
$fp = fopen($count_my_page , "w");
fputs($fp , "$hits[0]");
fclose($fp);
$id= $hits[0];
$sql="INSERT INTO reserve(ReserveId,ReserveDate,ReaderId,DocId,CopyNo,LibId) VALUES(:id,NULL,:rid,:docid,:copy,:libid)";
$query = $dbh->prepare($sql);
$query -> bindParam(':docid',$docid, PDO::PARAM_STR);
$query -> bindParam(':rid',$rid, PDO::PARAM_STR);
$query -> bindParam(':copy',$copy, PDO::PARAM_STR);
$query -> bindParam(':libid',$libid, PDO::PARAM_STR);
$query -> execute();
$_SESSION['delmsg']="Document Reserved scuccessfully ";
header('location:reserve-book.php');
}
?>
```

<u>Computing fine for a user:</u>

```
<?php
$sid=$_SESSION['login'];
$sql = "SELECT * from borrows where ReaderId=:sid";
//$sql="SELECT
tblbooks.BookName,tblbooks.ISBNNumber,tblissuedbookdetails.IssuesDate,tblissuedbookdetails.ReturnDate,tblissuedbookd
etails.id as rid,tblissuedbookdetails.fine from tblissuedbookdetails join tblstudents on
tblstudents.StudentId=tblissuedbookdetails.StudentId join tblbooks on tblbooks.id=tblissuedbookdetails.BookId where
tblstudents.StudentId=:sid order by tblissuedbookdetails.id desc";
$query = $dbh -> prepare($sql);
$query-> bindParam(':sid', $sid, PDO::PARAM_STR);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{
/*
$fine = 0;
$temp =0;
```

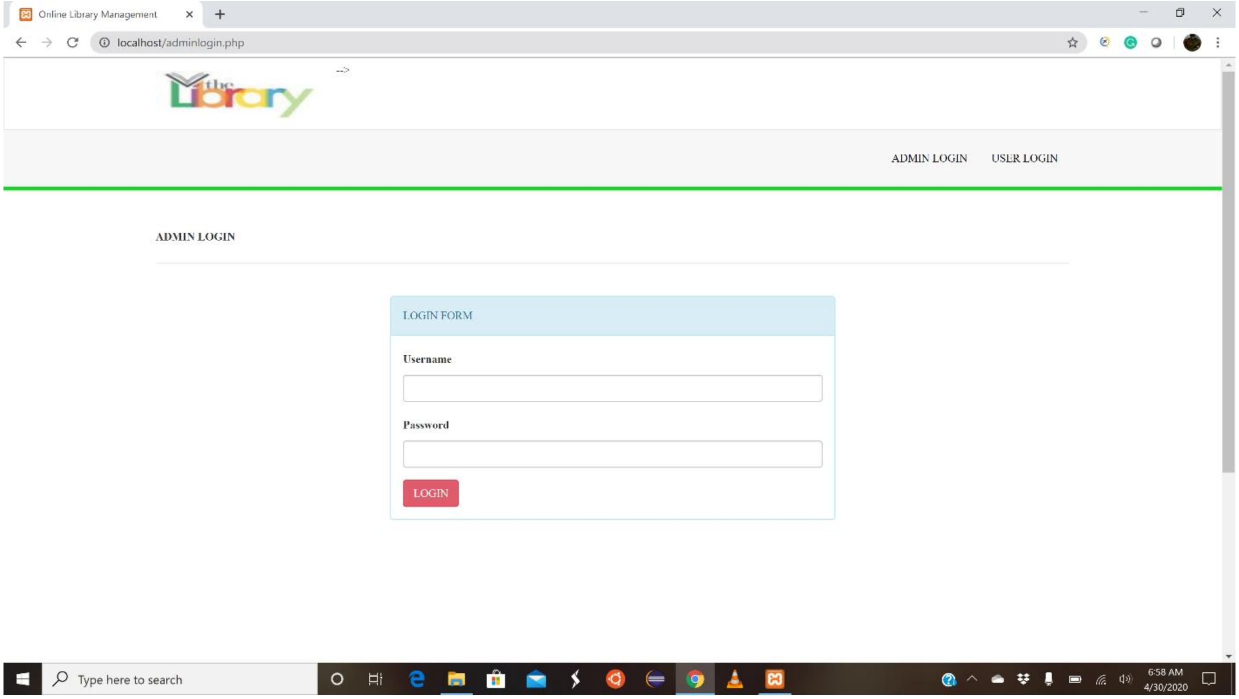
```
$today_date = date("Y-m-d");
$borrow_date = date(strtotime($row['BorrowDate']));
$i=20;
echo $due_by = strtotime(date("Y-m-d", strtotime($borrow_date)) . " +".$i."days");
if($today_date > $due_by){
if (strtotime($row['ReturnDate']) == '0000-00-00'){
$temp = $today_date->diff($due_by);
$fine = $temp * 0.10;
}
else {
$temp = $row['ReturnDate']->diff($due_by);
$fine = $temp * 0.10;
}
}
else {
$fine = 0;
}
```

Searching a document by using publisherid or Publishername:

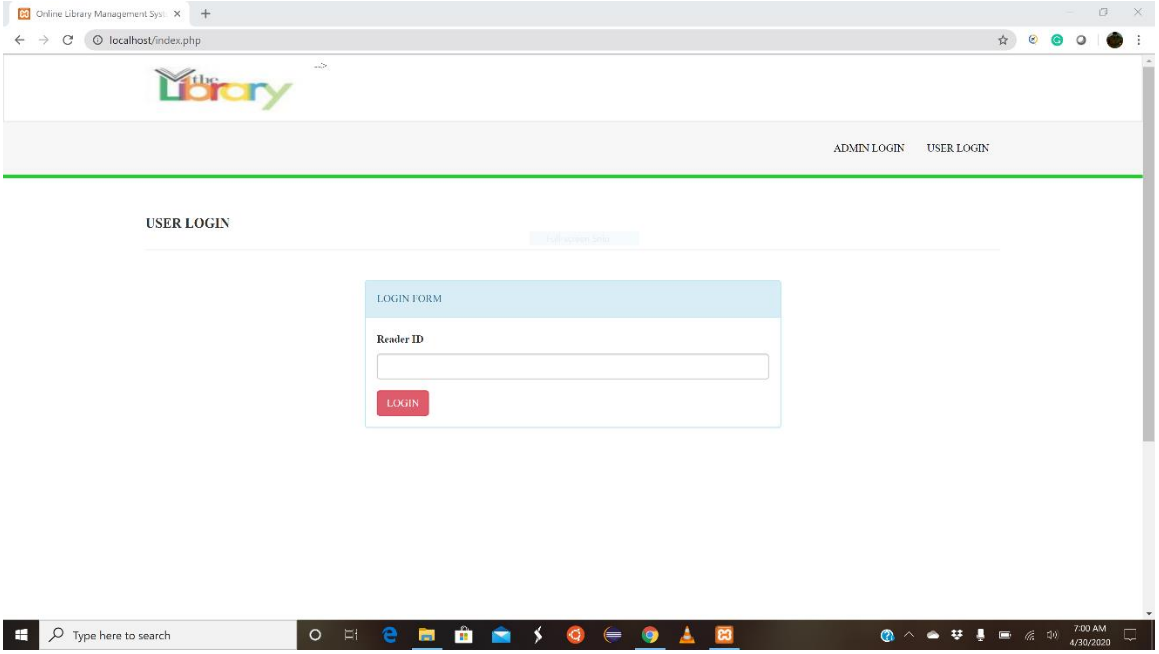
```
<?php $sql = "SELECT documents.DocId,documents.Title,documents.PubDate,documents.PublisherId,publishers.PubName
from documents INNER join publishers on documents.PublisherId = publishers.PubId ";
//"SELECT documents.DocId,documents.Title,documents.PubDate,documents.PublisherId from documents join publishers
on publishers.PubId = documents.PublisherId";
//"SELECT
tblbooks.BookName,tblcategory.CategoryName,tblauthors.AuthorName,tblbooks.ISBNNumber,tblbooks.BookPrice,tblbooks.i
d as bookid from tblbooks join tblcategory on tblcategory.id=tblbooks.CatId join tblauthors on
tblauthors.id=tblbooks.AuthorId";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{ ?>
```

Usage of the application:

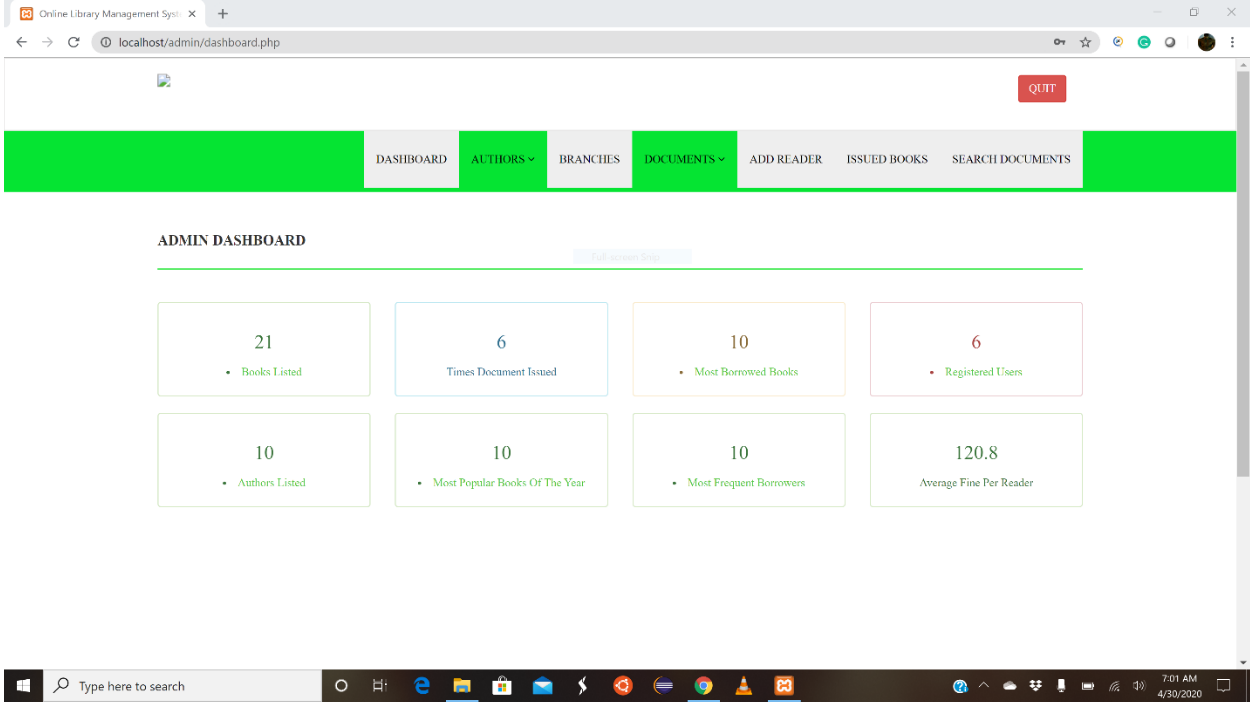
Admin login



Reader/User login



Admin Dashboard



ADMIN DASHBOARD

21
- Books Listed

6
Times Document Issued

10
- Most Borrowed Books

6
- Registered Users

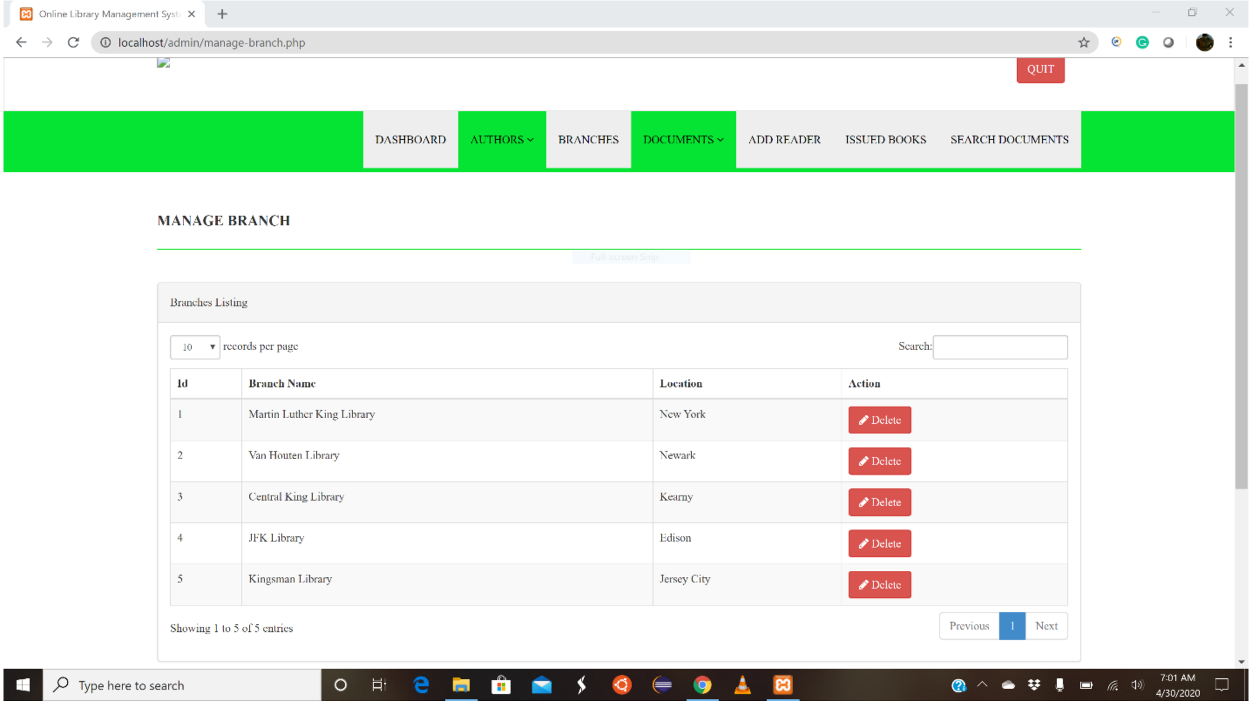
10
- Authors Listed

10
- Most Popular Books Of The Year

10
- Most Frequent Borrowers

120.8
Average Fine Per Reader

Branch Information



MANAGE BRANCH

Branches Listing

10 records per page

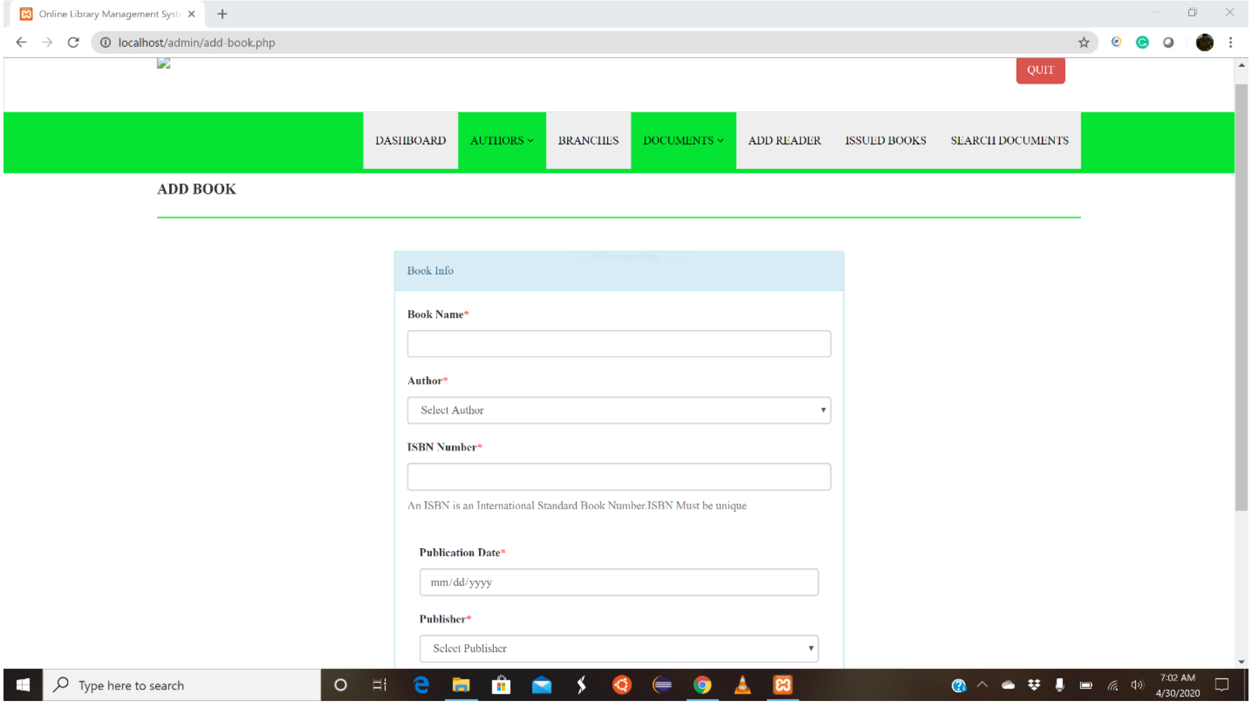
Search:

Id	Branch Name	Location	Action
1	Martin Luther King Library	New York	Delete
2	Van Houten Library	Newark	Delete
3	Central King Library	Kearny	Delete
4	JFK Library	Edison	Delete
5	Kingsman Library	Jersey City	Delete

Showing 1 to 5 of 5 entries

Previous 1 Next

Adding a Document by admin



ADD BOOK

Book Info

Book Name*

Author*

Select Author

ISBN Number*

An ISBN is an International Standard Book Number.ISBN Must be unique

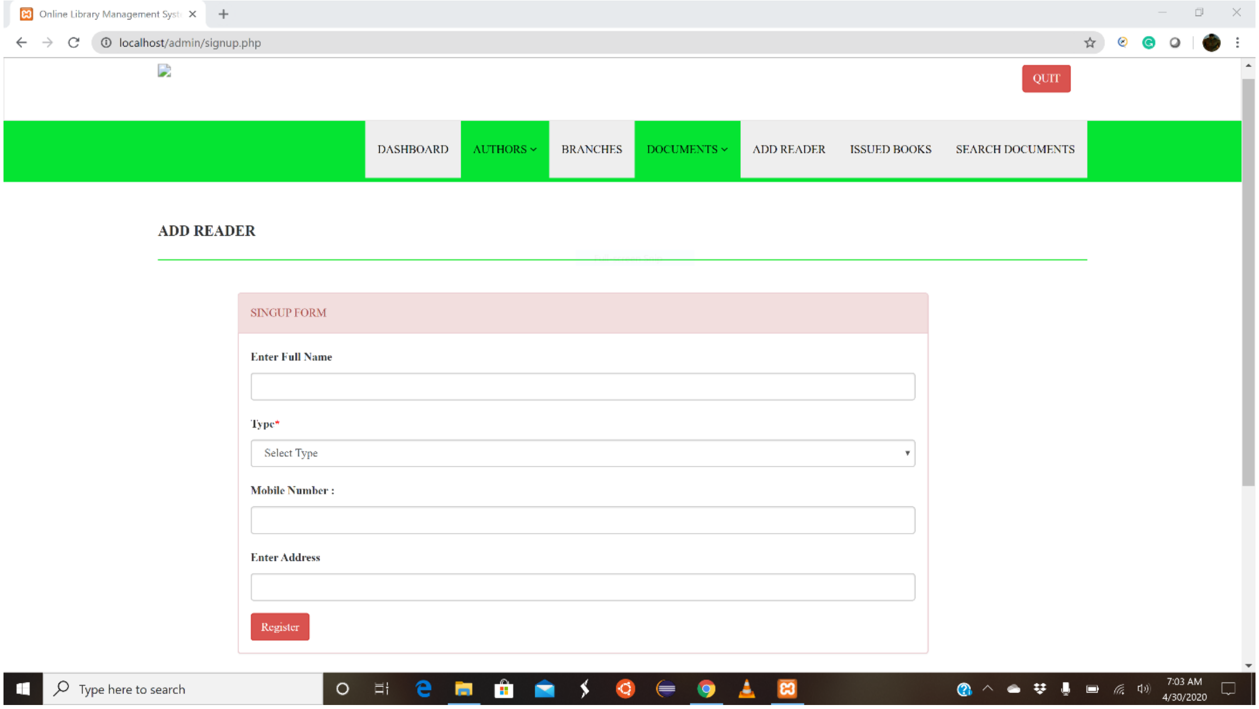
Publication Date*

mm/dd/yyyy

Publisher*

Select Publisher

New User signup



ADD READER

SINGUP FORM

Enter Full Name

Type*

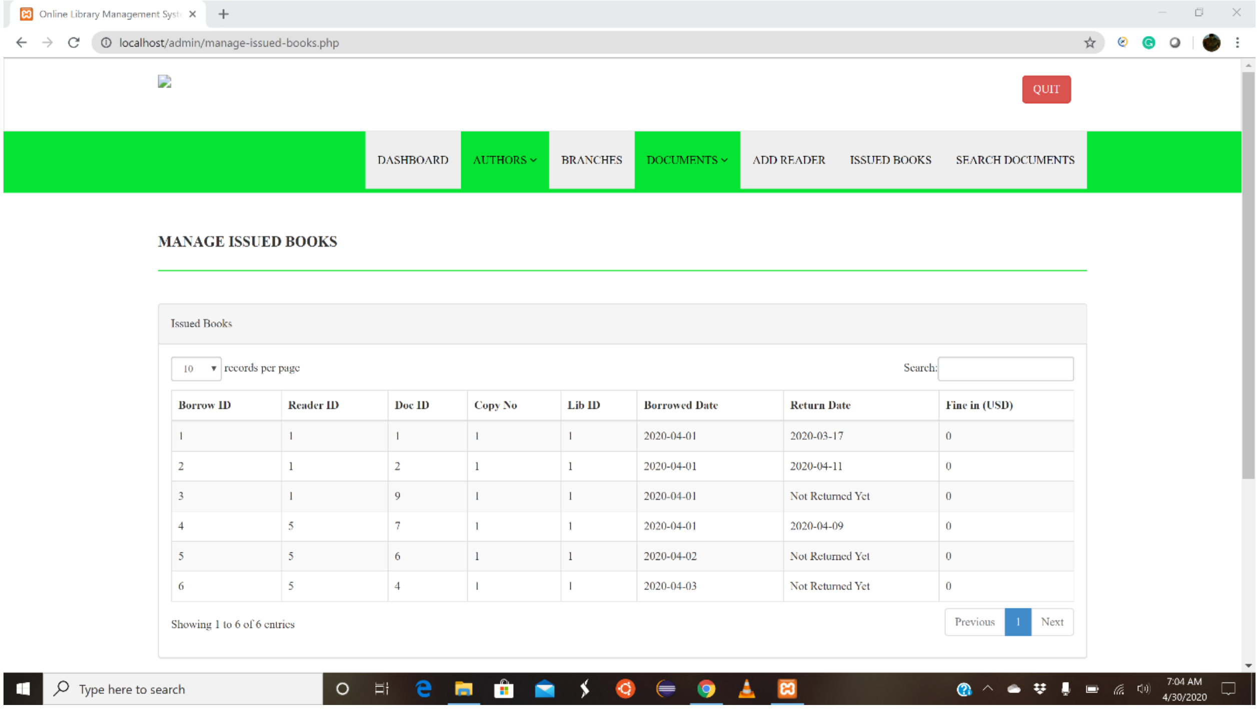
Select Type

Mobile Number :

Enter Address

Register

List of issued books



MANAGE ISSUED BOOKS

Issued Books

10 records per page

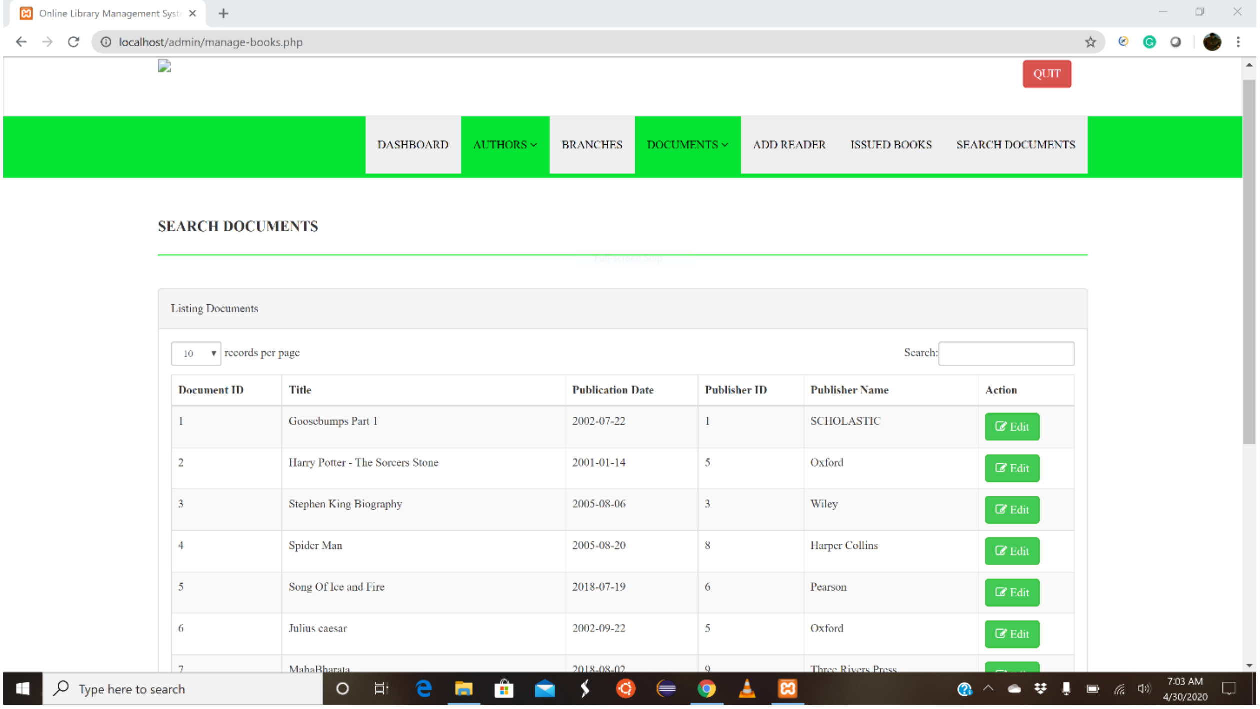
Search:

Borrow ID	Reader ID	Doc ID	Copy No	Lib ID	Borrowed Date	Return Date	Fine in (USD)
1	1	1	1	1	2020-04-01	2020-03-17	0
2	1	2	1	1	2020-04-01	2020-04-11	0
3	1	9	1	1	2020-04-01	Not Returned Yet	0
4	5	7	1	1	2020-04-01	2020-04-09	0
5	5	6	1	1	2020-04-02	Not Returned Yet	0
6	5	4	1	1	2020-04-03	Not Returned Yet	0

Showing 1 to 6 of 6 entries

Previous 1 Next

Searching for Documents



SEARCH DOCUMENTS

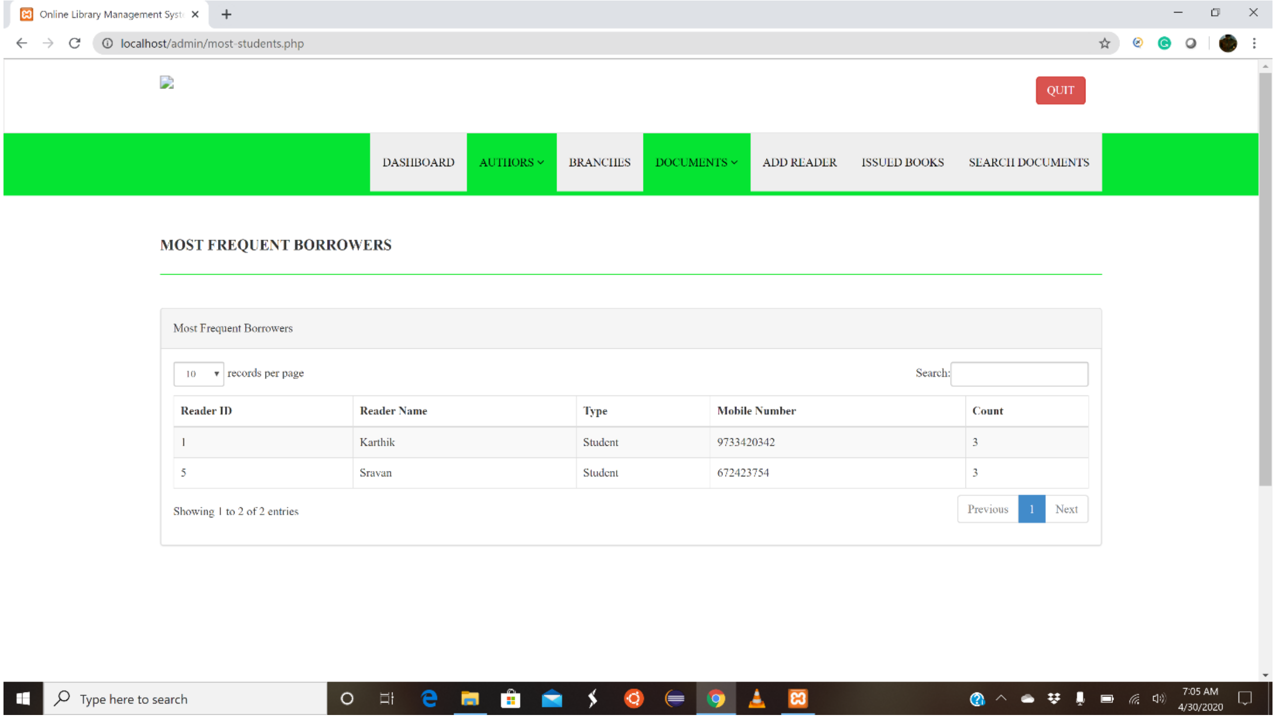
Listing Documents

10 records per page

Search:

Document ID	Title	Publication Date	Publisher ID	Publisher Name	Action
1	Goosebumps Part 1	2002-07-22	1	SCHOLASTIC	Edit
2	Harry Potter - The Sorcers Stone	2001-01-14	5	Oxford	Edit
3	Stephen King Biography	2005-08-06	3	Wiley	Edit
4	Spider Man	2005-08-20	8	Harper Collins	Edit
5	Song Of Ice and Fire	2018-07-19	6	Pearson	Edit
6	Julius caesar	2002-09-22	5	Oxford	Edit
7	MahaBharata	2018-08-02	9	Three Rivers Press	

Most frequent borrowers



MOST FREQUENT BORROWERS

Most Frequent Borrowers

10 records per page

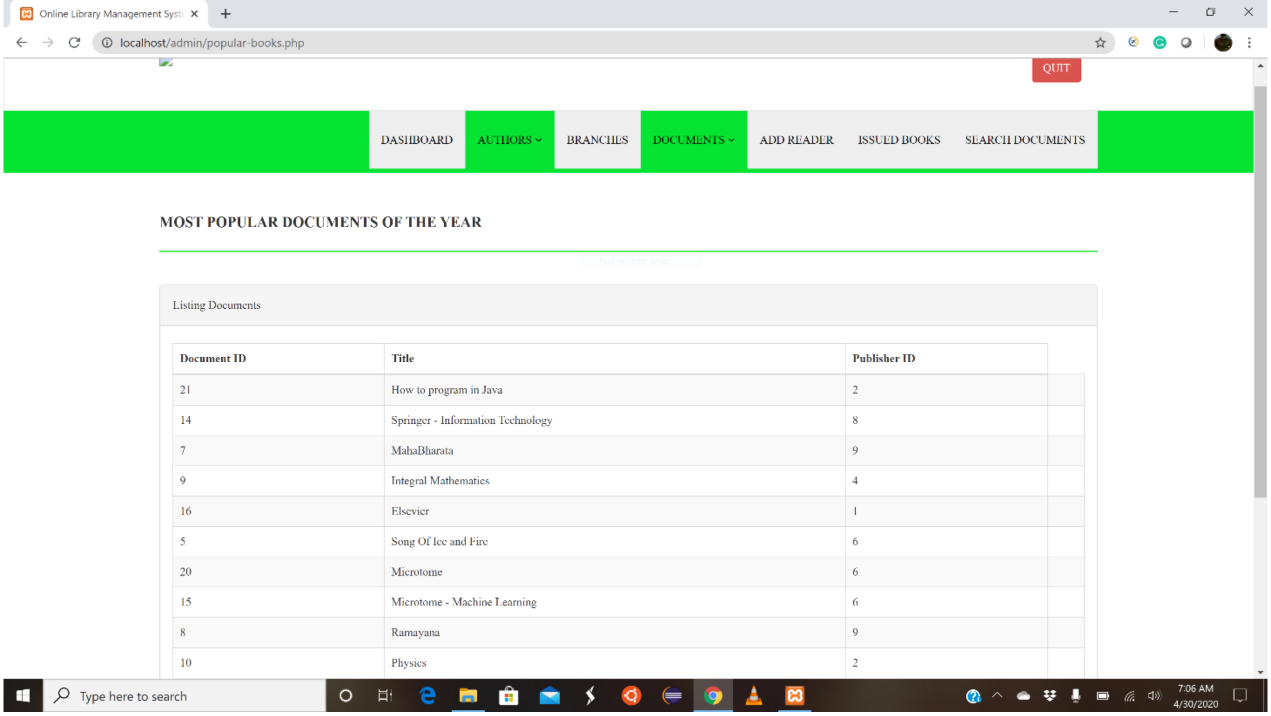
Search:

Reader ID	Reader Name	Type	Mobile Number	Count
1	Karthik	Student	9733420342	3
5	Sravan	Student	672423754	3

Showing 1 to 2 of 2 entries

Previous 1 Next

Most popular books of the year

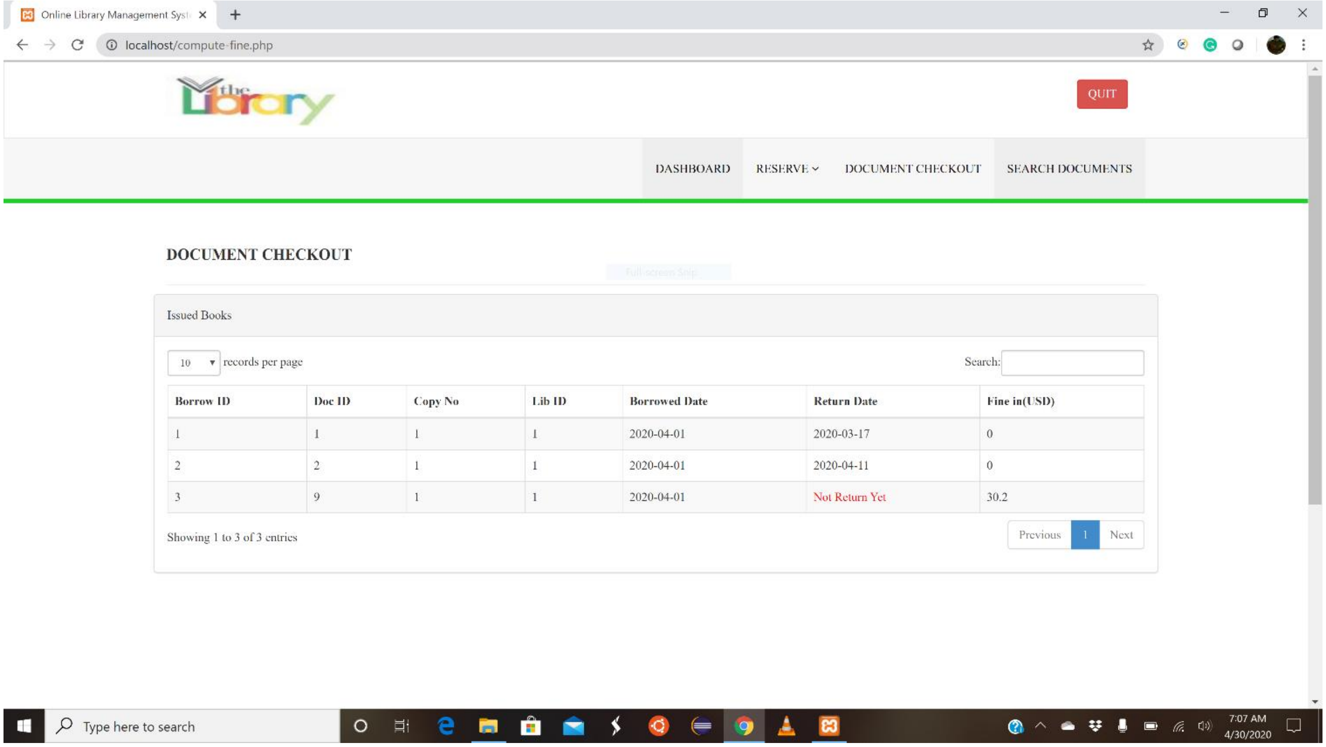


MOST POPULAR DOCUMENTS OF THE YEAR

Listing Documents

Document ID	Title	Publisher ID
21	How to program in Java	2
14	Springer - Information Technology	8
7	MahaBharata	9
9	Integral Mathematics	4
16	Elsevier	1
5	Song Of Ice and Fire	6
20	Microtome	6
15	Microtome - Machine Learning	6
8	Ramayana	9
10	Physics	2

Document checkout by user



DOCUMENT CHECKOUT

Issued Books

10 records per page

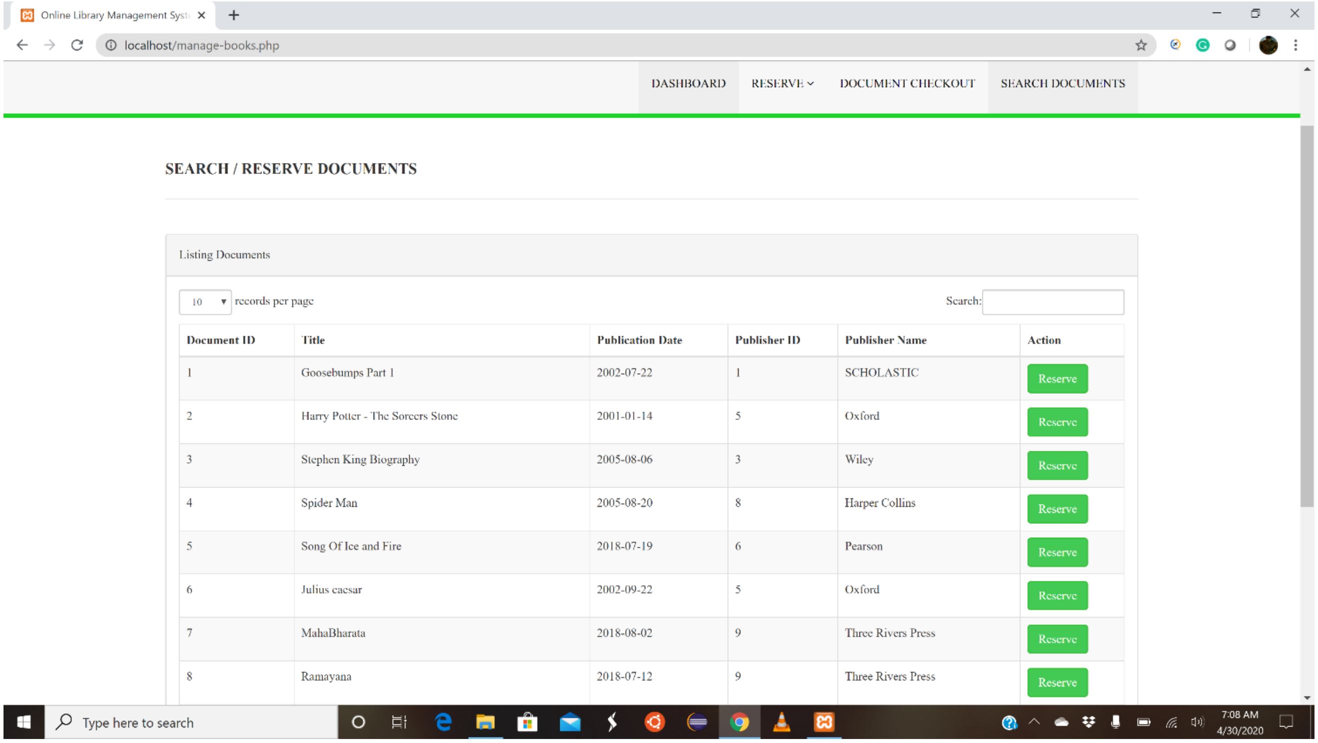
Search:

Borrow ID	Doc ID	Copy No	Lib ID	Borrowed Date	Return Date	Fine in(USD)
1	1	1	1	2020-04-01	2020-03-17	0
2	2	1	1	2020-04-01	2020-04-11	0
3	9	1	1	2020-04-01	Not Return Yet	30.2

Showing 1 to 3 of 3 entries

Previous 1 Next

Document Reserve



SEARCH / RESERVE DOCUMENTS

Listing Documents

10 records per page

Search:

Document ID	Title	Publication Date	Publisher ID	Publisher Name	Action
1	Goosebumps Part 1	2002-07-22	1	SCHOLASTIC	Reserve
2	Harry Potter - The Sorcers Stone	2001-01-14	5	Oxford	Reserve
3	Stephen King Biography	2005-08-06	3	Wiley	Reserve
4	Spider Man	2005-08-20	8	Harper Collins	Reserve
5	Song Of Ice and Fire	2018-07-19	6	Pearson	Reserve
6	Julius caesar	2002-09-22	5	Oxford	Reserve
7	MahaBharata	2018-08-02	9	Three Rivers Press	Reserve
8	Ramayana	2018-07-12	9	Three Rivers Press	Reserve

This concludes the end of this report. Thank you.