

# **MINI PROJECT ON LIBRARY MANAGEMENT SYSTEM**

**A Project Report Submitted in Partial  
Fulfilment of the Requirements for the  
Degree of**

**BACHELOR OF TECHNOLOGY  
in  
Computer Science & Engineering  
by**

**UTKARSH SHUKLA (190013135096)**

**AJAY KUMAR (190013135007)**

**NIKHIL SONI (190013135054)**

**VISHAL AWASTHI (190013135099)**

**Under the Guidance of  
Dr. JASWANT KUMAR**



**FACULTY OF ENGINEERING AND TECHNOLOGY  
UNIVERSITY OF LUCKNOW, LUCKNOW.**

**2022 – 23**

## **DECLARATION**

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person or material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher education, except where due acknowledgement has been made in the text.

**Signature**

**Student Name: Utkarsh Shukla**

**Univ. Roll. No.: 190013135096**

**Date:**

**Signature**

**Student Name: Ajay Kumar**

**Univ. Roll. No.: 190013135007**

**Date:**

**Signature**

**Student Name: Nikhil Soni**

**Univ. Roll. No.: 190013135054**

**Date:**

**Signature**

**Student Name: Vishal Awasthi**

**Univ. Roll. No.: 190013135099**

**Date:**

## CERTIFICATE

Certified that **Utkarsh Shukla** (190013135096) has carried out the mini project work presented in this project report entitled “**Library Management System**” for the award of **Bachelor of Technology** (Computer Science & Technology) from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my guidance. The project report embodies results of original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University Of Lucknow /Faculty of Engineering and Technology.

Signature

Name of Guide: **Dr. Jaswant Kumar**

Designation: **Assistant Professor, FOET, University of Lucknow**

Place: Lucknow

## CERTIFICATE

Certified that **Ajay Kumar** (190013135007) has carried out the mini project work presented in this project report entitled “**Library Management System**” for the award of **Bachelor of Technology** (Computer Science & Technology) from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my guidance. The project report embodies results of original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University Of Lucknow /Faculty of Engineering and Technology.

Signature

Name of Guide: **Dr. Jaswant Kumar**

Designation: **Assistant Professor, FOET, University of Lucknow**

Place: Lucknow

## CERTIFICATE

Certified that **Nikhil Soni** (190013135054) has carried out the mini project work presented in this project report entitled “**Library Management System**” for the award of **Bachelor of Technology** (Computer Science & Technology) from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my guidance. The project report embodies results of original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University Of Lucknow /Faculty of Engineering and Technology.

Signature

Name of Guide: **Dr. Jaswant Kumar**

Designation: **Assistant Professor, FOET, University of Lucknow**

Place: Lucknow

## CERTIFICATE

Certified that **Vishal Awasthi** (190013135099) has carried out the mini project work presented in this project report entitled “**Library Management System**” for the award of **Bachelor of Technology** (Computer Science & Technology) from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my guidance. The project report embodies results of original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University Of Lucknow /Faculty of Engineering and Technology.

Signature

Name of Guide: **Dr. Jaswant Kumar**

Designation: **Assistant Professor, FOET, University of Lucknow**

Place: Lucknow

## ACKNOWLEDGEMENT

First of all, I am indebted to the GOD ALMIGHTY for giving me an opportunity to excel in my efforts to complete this seminar on time.

I am extremely grateful to **Prof. Alok Kumar Rai, Prof. A.K. Singh**, University of Lucknow and **Er. Zeeshan Ali Siddiqui**, Head of Department, Department of Computer Science and Engineering, for providing all the required resources for the successful completion of my MINI PROJECT.

My heartfelt gratitude to my mini project guide **Dr. Jaswant Kumar**, Computer Science and Engineering, for his valuable suggestions and guidance in the preparation of the mini project report.

I will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred to in this mini project.

Last but not the least; I am very much thankful to my parents who guided me in every step which I took.

**Student Name: Utkarsh Shukla**  
**Univ. Roll. No.: 190013135096**

**Student Name: Ajay Kumar**  
**Univ. Roll. No.: 190013135007**

**Student Name: Nikhil Soni**  
**Univ. Roll. No.: 190013135054**

**Student Name: Vishal Awasthi**  
**Univ. Roll. No.: 190013135099**

## Abstract

A college library management is a project that manages and stores books information electronically according to students needs. The system helps both students and library manager to keep a constant track of all the books available in the library. It allows both the admin and the student to search for the desired book. It becomes necessary for colleges to keep a continuous check on the books issued and returned and even calculate fine. This task if carried out manually will be tedious and includes chances of mistakes. These errors are avoided by allowing the system to keep track of information such as issue date, last date to return the book and even fine information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes.

### Modules:

**Admin login:** Admin is the one who administers the system by adding or removing e-books into and from the system respectively.

**User login:** Students have to register themselves into the system to create an account. After registering successfully, they can then login into the system by entering 10 digit mobile number and their email id.

**Add and Update Books:** The admin can add books to the system by entering the details of the books and can even update the details.

**Search option:** Admin and Students can even search for books by entering the name of the book.

**View Order-**The admin can view order for the books.

**Calculate Fine-** The student can view the issue and expiry date for the book issued and can even calculate fine.

### Basic implementation tools:

- HTML
- CSS
- Java Script
- Bootstrap
- VS Code



# Table of Contents

|                                   |              |
|-----------------------------------|--------------|
| Page No.                          |              |
| Declaration                       | ii           |
| Certificate                       | iii          |
| Acknowledgement                   | iv           |
| Abstract                          | v            |
| <b>CHAPTER 1: INTRODUCTION</b>    | <b>1-2</b>   |
| 1.1 OBJECTIVE                     | 1            |
| 1.2 BACKGROUND OF PROJECT         | 1            |
| 1.3 OPREATION ENVIRONMENT         | 2            |
| <b>CHAPTER 2: SYSTEM ANALYSIS</b> | <b>3-11</b>  |
| 2.1 FUNCTIONAL REQUIREMENTS       | 3            |
| 2.2 NON-FUNCTIONAL REQUIREMENTS   | 4            |
| 2.3 SOFTWARE REQUIREMENTS         | 6            |
| 2.4 HARDWARE REQUIREMENTS         | 7            |
| 2.5 TOOLS                         | 9            |
| <b>CHAPTER 3: SYSTEM DESIGN</b>   | <b>12-29</b> |
| 3.1 SYSTEM DIAGRAM                | 12           |
| 3.2 TABLE NAME AND ATTRIBUTES     | 13           |
| 3.3 IMPLEMENTATION                | 14           |
| <b>CHAPTER 4: PROJECT IMAGES</b>  | <b>30-32</b> |
| 4.1 SYSTEM LOGIN                  | 30           |
| 4.2 HOME PAGE                     | 30           |
| 4.3 REQUEST BOOK                  | 31           |
| 4.4 ISSUE BOOK                    | 32           |
| <b>CHAPTER 5: CONCLUSION</b>      | <b>33</b>    |
| <b>CHAPTER 6: REFERENCES</b>      | <b>34</b>    |

# **CHAPTER 1**

## **INTRODUCTION**

This chapter gives an overview about the aim, objectives, background and operation environment of the system.

### **1.1 PROJECT AIMS AND OBJECTIVES**

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

- Online book issue
- Request column for librarian for providing new books
- A separate column for digital library
- Student login page where student can find books issued by him/her and date of return.
- A search column to search availability of books
- A teacher login page where teacher can add any events being organized in the college and important suggestions regarding books.
- Online notice board about the workshop.

### **1.2 BACKGROUND OF PROJECT**

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students etc.

Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non-computerized system is used.

In addition, report module is also included in Library Management System. If user's position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports.

All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

### **1.3 OPERATION ENVIRONMENT**

|                  |  |
|------------------|--|
| PROCESSOR        | INTEL CORE PROCESSOR OR BETTER PERFORMANCE |
| OPERATING SYSTEM | WINDOWS VISTA ,WINDOWS7, UBUNTU            |
| MEMORY           | 1GB RAM OR MORE                            |
| HARD DISK SPACE  | MINIMUM 3 GB FOR DATABASE USAGE FOR FUTURE |
| DATABASE         | MY SQL                                     |

## CHAPTER 2

### SYSTEM ANALYSIS

In this chapter, we will discuss and analyze about the developing process of Library Management System including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

### 2.1 SOFTWARE REQUIREMENT SPECIFICATION

#### 2.1.1 GENERAL DESCRIPTION

##### **PRODUCT DESCRIPTION:**

Library Management System is a computerized system which helps users (librarian) to manage the library daily activity in electronic format. It reduces the risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively and time saving.

##### **PROBLEM STATEMENT:**

The problem occurred before having computerized includes:

- **File lost:** When computerized system is not implemented file is always lost because of human environment. Sometimes due to some human error there may be a loss of records.
- **File damaged:** When a computerized system is not their file is always lost due to some accident like spilling of water by some member on file accidentally. Besides some natural disaster like floods or fires may also damage the files.
- **Difficult to search record:** When there is no computerized system there is always a difficulty in searching of records if the records are large in number.
- **Space consuming:** After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.
- **Cost consuming:** as there is no computerized system the to add each record paper will be needed which will increase the cost for the management of library.

## 2.1.2 SYSTEM OBJECTIVES

- **Improvement in control and performance:** The system is developed to cope up with the current issues and problems of library. The system can add user, validate user and is also bug free.
- **Save cost** After computerized system is implemented less human force will be required to maintain the library thus reducing the overall cost.
- **Save time** Librarian is able to search record by using few clicks of mouse and few search keywords thus saving his valuable time.
- **Option of online Notice board** Librarian will be able to provide a detailed description of workshops going in the college as well as in nearby colleges
- **Lecture Notes** Teacher have a facility to upload lectures notes in a pdf file having size not more than 10mb.

## 2.1.3 SYSTEM REQUIREMENTS

### 2.1.3.1 NON-FUNCTIONAL REQUIREMENTS

#### SYSTEM REQUIREMENT

**EFFICIENCY REQUIREMENT:** When a library management system will be implemented librarian and user will easily access library as searching and book transaction will be very faster.

**RELIABILITY REQUIREMENT:** The system should accurately perform member registration, member validation, report generation, book transaction and search.

**USABILITY REQUIREMENT:** The system is designed for a user-friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.

#### ORGANIZATIONAL REQUIREMENT

**IMPLEMENTATION REQUIREMENTS:** In implementing whole system it uses html in front end with PHP as server-side scripting language which will be used for database connectivity and the backend i.e., the database part is developed using MySQL.

**DELIVERY REQUIREMENTS:** The whole system is expected to be delivered in six months of time with a weekly evaluation by the project guide.

### **2.1.3.2 FUNCTIONAL REQUIREMENTS**

#### **1. NORMAL USER**

##### **1.1 USER LOGIN**

This feature used by the user to login into system. They are required to enter user id and password before they are allowed to enter the system. The user id and password will be verified and if invalid id is their user is allowed to not enter the system.

- User id is provided when they register
- The system must only allow user with valid id and password to enter the system
- The system performs authorization process which decides what user level can access too.
- The user must be able to logout after they finished using system.

##### **1.2 REGISTER NEW USER**

This feature can be performed by all users to register new user to create account.

- System must be able to verify information.
- System must be able to delete information if information is wrong.

##### **1.3 REGISTER NEW BOOK**

This feature allows to add new books to the library.

- System must be able to verify information
- System must be able to enter number of copies into table.
- System must be able to not allow two books having same book id.
-

## **1.4 SEARCH BOOK**

This feature is found in book maintenance part. we can search book based on book id, book name, publication or by author name.

- System must be able to search the database based on select search type
- System must be able to filter book based on keyword entered
- System must be able to show the filtered book in table view.

## **1.5 ISSUE BOOKS AND RETURN BOOKS**

This feature allows to issue and return books and also view reports of book issued.

- System must be able to enter issue information in database.
- System must be able to update number of books.
- System must be able to search if book is available or not before issuing books.
- System should be able to enter issue and return date information.

## **1.6 EVENT ADDITION**

This feature allows teacher and student to add information about various workshops being conducted in college and colleges nearby.

- System should be able to add detailed information about events.
- System should be able to display information on notice board available in the homepage of site.

## **2.2 SOFTWARE AND HARDWARE REQUIREMENTS**

This section describes the software and hardware requirements of the system.

### **2.2.1 SOFTWARE REQUIREMENTS**

- Operating system- Windows 7 is used as the operating system as it is stable and supports more features and is more user friendly.
- Database MYSQL-MYSQL is used as database as it easy to maintain and retrieve

records by simple queries which are in English language which are easy to understand and easy to write.

- Development tools and Programming language- HTML is used to write the whole code and develop webpages with CSS, Java Script for styling work and PHP for sever side scripting.

### **2.2.2 HARDWARE REQUIREMENTS**

- Intel core i5 2<sup>nd</sup> generation is used as a processor because it is fast than other processors an provide reliable and stable and we can run our pc for long time. By using this processor, we can keep on developing our project without any worries.
- Ram 1 GB is used as it will provide fast reading and writing capabilities and will in turn support in processing.

### **2.3 EXISTING VS PROPOSED SYSTEM**

- Existing system does not have any facility of teacher's login or student login whereas proposed system will have a facility of student login as well as teacher's login.
- Existing system does not have a facility of online reservation of books whereas proposed system has a facility of online reservation of books.
- Existing system does not have any facility of online notice board were description of workshops happening in our college as well as nearby colleges is being provided.
- Existing system does not have any option of lectures notes uploaded by teachers whereas proposed system will have this facility.
- Existing system does not have any facility to generate student reports as well book issue reports whereas proposed system provides librarian with a tool to generate reports.



- Existing system does not have any facility for book request and suggestions where as in proposed system after logging in to their accounts student can request books as well as provide suggestions to improve library.

## 2.4 SOFTWARE TOOLS USED

The whole Project is divided in two parts the front end and the back end.

### 2.4.1 FRONTEND

The front end is designed using of HTML, CSS, Java Script.

#### HTML (HYPER TEXT MARKUP LANGUAGE)

language for creating web pages and other information that can be displayed in a web browser. HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like `<html>`), within the web page content. HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent *empty elements* and so are unpaired, for example `<img>`. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). In between these tags web designers can add text, further tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

#### CSS (CASCADING STYLE SHEETS)

is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including

plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied. CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called *cascade*, priorities or *weights* are calculated and assigned to rules, so that the results are predictable.

## **JAVA SCRIPT (JS)**

is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics.

The key design principles within JavaScript are taken from the Self and Scheme programming languages. The application of JavaScript to use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications.

## **PHP**

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Pre-processor*, a recursive backronym. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications. PHP is free software released under the PHP License. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

### **2.4.2 BACKEND**

The back end is designed using MySQL which is used to design the databases.

## **MySQL**

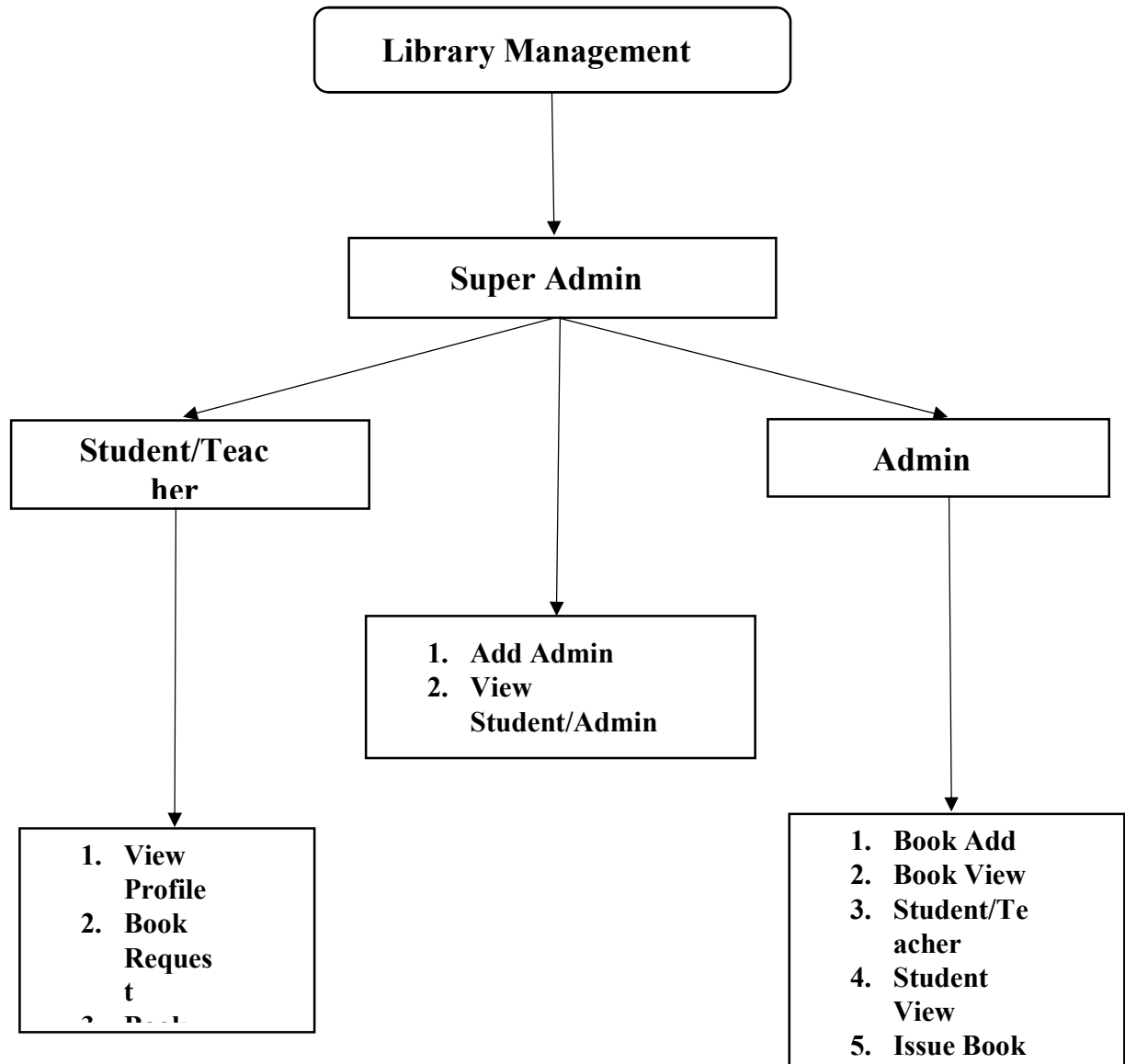
("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular

choice of database for use in web applications, and is a central component of the widely used LAMP open-source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open-source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality.

## CHAPTER 3

### SYSTEM DESIGN

#### 3.1 System Diagram:



## 3.2 Table Name and Their Attributes:

### 3.2.1 Admin

ID, Email, Pass, Type

### 3.2.2 Book:

ID, Book Pic, Issue name, Book Details, Book Author, Branch, Book price, Book quantity, Book rent

### 3.2.3 Issue Book:

ID, User ID, Issue Name, Issue Type, Issue Day, Issue date, Issue return, Fine

### 3.2.4 Request Book:

ID, User ID, Book ID, User name, User type, Book name, Issue Day

### 3.2.5 User data:

ID, Name, Email, Pass, Type

## 3.3 Implementation:

### 3.3.1 Connections with database:

```
db.php > ...
1  <?php
2  class db{
3  protected $connection;
4
5  function setconnection(){
6      try{
7          $this->connection=new PDO("mysql:host=localhost; dbname=library_managment_system","root","");
8          // echo "Done";
9      }catch(PDOException $e){
10         echo "Error";
11         //die();
12     }
13 }
14 }
15
16 }
17
```

### 3.3.2 Login:

login\_server\_page.php > ...

```
1  <?php
2
3  include("data_class.php");
4
5  $login_email=$_GET['login_email'];
6  $login_pasword=$_GET['login_pasword'];
7
8
9  if($login_email==null||$login_pasword==null){
10     $emailmsg="";
11     $pasdmsg="";
12
13     if($login_email==null){
14         $emailmsg="Email Empty";
15     }
16     if($login_pasword==null){
17         $pasdmsg="Pasword Empty";
18     }
19
20     header("Location: index.php?emailmsg=$emailmsg&pasdmsg=$pasdmsg");
21 }
22
23 elseif($login_email!=null&&$login_pasword!=null){
24     $obj=new data();
25     $obj->setconnection();
26     $obj->userLogin($login_email,$login_pasword);
27 }
28
29
```

loginadmin\_server\_page.php > ...

```
1  <?php
2
3  include("data_class.php");
4
5  $login_email=$_GET['login_email'];
6  $login_pasword=$_GET['login_pasword'];
7
8
9  if($login_email==null||$login_pasword==null){
10     $emailmsg="";
11     $pasdmsg="";
12
13     if($login_email==null){
14         $emailmsg="Email Empty";
15     }
16     if($login_pasword==null){
17         $pasdmsg="Pasword Empty";
18     }
19
20     header("Location: index.php?ademailmsg=$emailmsg&adpasdmsg=$pasdmsg");
21 }
22
23 elseif($login_email!=null&&$login_pasword!=null){
24     $obj=new data();
25     $obj->setconnection();
26     $obj->adminLogin($login_email,$login_pasword);
27 }
```

```

21
22
23
24
25     function __construct() {
26         // echo " constructor ";
27         echo "</br></br>";
28     }
29
30
31     function addnewuser($name,$password,$email,$type){
32         $this->name=$name;
33         $this->password=$password;
34         $this->email=$email;
35         $this->type=$type;
36
37
38         $q="INSERT INTO userdata(id, name, email, pass,type)VALUES('','$name','$email','$password','$type')";
39
40         if($this->connection->exec($q)) {
41             header("Location:admin_service_dashboard.php?msg=New Add done");
42         }
43
44         else {
45             header("Location:admin_service_dashboard.php?msg=Register Fail");
46         }

```

### 3.3.3 Some Important Functions:

```

50     }
51     function userLogin($t1, $t2) {
52         $q="SELECT * FROM userdata where email='$t1' and pass='$t2'";
53         $recordSet=$this->connection->query($q);
54         $result=$recordSet->rowCount();
55         if ($result > 0) {
56
57             foreach($recordSet->fetchAll() as $row) {
58                 $logid=$row['id'];
59                 header("location: otheruser_dashboard.php?userlogid=$logid");
60             }
61         }
62         else {
63             header("location: index.php?msg=Invalid Credentials");
64         }
65     }
66     function adminLogin($t1, $t2) {
67
68         $q="SELECT * FROM admin where email='$t1' and pass='$t2'";
69         $recordSet=$this->connection->query($q);
70         $result=$recordSet->rowCount();

```



```

66     function adminLogin($t1, $t2) {
67
68         $q="SELECT * FROM admin where email='$t1' and pass='$t2'";
69         $recordSet=$this->connection->query($q);
70         $result=$recordSet->rowCount();
71
72         if ($result > 0) {
73
74             foreach($recordSet->fetchAll() as $row) {
75                 $logid=$row['id'];
76                 header("location: admin_service_dashboard.php?logid=$logid");
77             }
78         }
79
80         else {
81             header("location: index.php?msg=Invalid Credentials");
82         }
83
84     }
85 }
86
87 function addbook($bookpic, $bookname, $bookdetail, $bookaudor, $bookpub, $branch, $bookprice, $bookquantity) {
88     $this->$bookpic=$bookpic;
89     $this->$bookname=$bookname;
90     $this->$bookdetail=$bookdetail;
91     $this->$bookaudor=$bookaudor;
92     $this->$bookpub=$bookpub;
93     $this->$branch=$branch;
94     $this->$bookprice=$bookprice;
95     $this->$bookquantity=$bookquantity;
96
97     $q="INSERT INTO book (id,bookpic,bookname, bookdetail, bookaudor, bookpub, branch, bookprice,bookquantity,bookava,bookrent)VALUES('
98
99     if($this->connection->exec($q)) {
100         header("Location:admin_service_dashboard.php?msg=done");
101     }
102
103     else {
104         header("Location:admin_service_dashboard.php?msg=fail");
105     }
106
107 }
108
109 }

```

```

115 function getissuebook($userloginid) {
116
117     $newfine="";
118     $issuereturn="";
119
120     $q="SELECT * FROM issuebook where userid='$userloginid'";
121     $recordSetss=$this->connection->query($q);
122
123
124     foreach($recordSetss->fetchAll() as $row) {
125         $issuereturn=$row['issuereturn'];
126         $fine=$row['fine'];
127         $newfine= $fine;
128
129
130         // $newbookrent=$row['bookrent']+1;
131     }
132
133
134     $getdate= date("d/m/Y");
135     if($issuereturn<$getdate){
136         $q="UPDATE issuebook SET fine='$newfine' where userid='$userloginid'";
137
138
139     $getdate= date("d/m/Y");
140     if($issuereturn<$getdate){
141         $q="UPDATE issuebook SET fine='$newfine' where userid='$userloginid'";
142
143         if($this->connection->exec($q)) {
144             $q="SELECT * FROM issuebook where userid='$userloginid' ";
145             $data=$this->connection->query($q);
146             return $data;
147         }
148         else{
149             $q="SELECT * FROM issuebook where userid='$userloginid' ";
150             $data=$this->connection->query($q);
151             return $data;
152         }
153     }
154     else{
155         $q="SELECT * FROM issuebook where userid='$userloginid'";
156         $data=$this->connection->query($q);
157         return $data;
158     }
159 }

```

data\_class.php > data > userLogin

```
163
164     function getbook() {
165         $q="SELECT * FROM book ";
166         $data=$this->connection->query($q);
167         return $data;
168     }
169     function getbookissue(){
170         $q="SELECT * FROM book where bookava !=0 ";
171         $data=$this->connection->query($q);
172         return $data;
173     }
174
175     function userdata() {
176         $q="SELECT * FROM userdata ";
177         $data=$this->connection->query($q);
178         return $data;
179     }
180
181
182     function getbookdetail($id){
183         $q="SELECT * FROM book where id ='$id'";
184         $data=$this->connection->query($q);
185         return $data;
186     }
187
188     function userdetail($id){
189         $q="SELECT * FROM userdata where id ='$id'";
190         $data=$this->connection->query($q);
191         return $data;
```

data\_class.php > data > requestbook

```
193
194     function requestbook($userid,$bookid){
195         $q="SELECT * FROM book where id='$bookid'";
196         $recordSetss=$this->connection->query($q);
197         $q="SELECT * FROM userdata where id='$userid'";
198         $recordSet=$this->connection->query($q);
199         foreach($recordSet->fetchAll() as $row) {
200             $username=$row['name'];
201             $usertype=$row['type'];
202         }
203         foreach($recordSetss->fetchAll() as $row) {
204             $bookname=$row['bookname'];
205         }
206         if($usertype=="student"){
207             $days=7;
208         }
209         if($usertype=="teacher"){
210             $days=21;
211         }
212         $q="INSERT INTO requestbook (id,userid,bookid,username,usertype,bookname,issuedays)VALUES('','$userid', '$bookid', '$username', '$usert
213
214         if($this->connection->exec($q)) {
215             header("Location:otheruser_dashboard.php?userid=$userid");
216         }
217         else {
218             header("Location:otheruser_dashboard.php?msg=fail");
219         }
220     }
221
```

```

data_class.php > data > requestbook
225 $bookava="";
226 $issuebook="";
227 $bookrentel="";
228
229 $q="SELECT * FROM issuebook where id='$id'";
230 $recordSet=$this->connection->query($q);
231
232 foreach($recordSet->fetchAll() as $row) {
233     $userid=$row['userid'];
234     $issuebook=$row['issuebook'];
235     $fine=$row['fine'];
236 }
237
238 if($fine==0){
239
240     $q="SELECT * FROM book where bookname='$issuebook'";
241     $recordSet=$this->connection->query($q);
242
243     foreach($recordSet->fetchAll() as $row) {
244         $bookava=$row['bookava']+1;
245         $bookrentel=$row['bookrent']-1;
246     }
247     $q="UPDATE book SET bookava='$bookava', bookrent='$bookrentel' where bookname='$issuebook'";
248     $this->connection->exec($q);
249
250     $q="DELETE from issuebook where id=$id and issuebook='$issuebook' and fine='0' ";
251     if($this->connection->exec($q)){
252
253         header("Location:otheruser_dashboard.php?userlogid=$userid");
254     }
}

data_class.php > data > issuebookapprove
328 $bookid=$row['id'];
329 $bookname=$row['bookname'];
330
331     $newbookava=$row['bookava']-1;
332     $newbookrent=$row['bookrent']+1;
333 }
334
335
336 $q="UPDATE book SET bookava='$newbookava', bookrent='$newbookrent' where id='$bookid'";
337 if($this->connection->exec($q)){
338
339     $q="INSERT INTO issuebook (userid,issuename,issuebook,issuetype,issuedays,issuedate,issuereturn,fine)
340     VALUES('$issueid','$userselect','$book','$issuetype','$days','$getdate','$returnDate','0')";
341
342     if($this->connection->exec($q)) {
343
344         $q="DELETE from requestbook where id='$redid'";
345         $this->connection->exec($q);
346         header("Location:admin_service_dashboard.php?msg=done");
347     }
348
349     else {
350         header("Location:admin_service_dashboard.php?msg=fail");
351     }
352 }
353 else{
354     header("Location:admin_service_dashboard.php?msg=fail");
355 }

```

```

401
402     $q="UPDATE book SET bookava='$newbookava', bookrent='$newbookrent' where id='$bookid'";
403     if($this->connection->exec($q)){
404
405         $q="INSERT INTO issuebook (userid,issuename,issuebook,issuetype,issuedays,issuedate,issuereturn,fine)
406         VALUES('$issueid','$userselect','$book','$issuetype','$days','$getdate','$returnDate','0')";
407
408         if($this->connection->exec($q)) {
409             header("Location:admin_service_dashboard.php?msg=done");
410         }
411
412         else {
413             header("Location:admin_service_dashboard.php?msg=fail");
414         }
415     }
416     else{
417         header("Location:admin_service_dashboard.php?msg=fail");
418     }
419
420
421 }
422
423 else {
424     header("location: index.php?msg=Invalid Credentials");
425 }

```

```

24
25     function __construct() {
26         // echo " constructor ";
27         echo "</br></br>";
28     }
29
30
31     function addnewuser($name,$password,$email,$type){
32         $this->name=$name;
33         $this->password=$password;
34         $this->email=$email;
35         $this->type=$type;
36
37
38         $q="INSERT INTO userdata(id, name, email, pass,type)VALUES('','$name','$email','$password','$type')";
39
40         if($this->connection->exec($q)) {
41             header("Location:admin_service_dashboard.php?msg=New Add done");
42         }
43
44         else {
45             header("Location:admin_service_dashboard.php?msg=Register Fail");
46         }

```

## CHAPTER 4

### PROJECT IMAGES

#### 4.1 System Login:


The image displays two login interfaces side-by-side. The left interface, titled 'Student Login', has a light gray background and contains two input fields for 'Your Email \*' and 'Your Password \*', a blue 'Login' button, and a 'Forgot Password?' link. The right interface, titled 'Admin Login', has a blue background and contains identical input fields, a white 'Login' button, and a 'Forgot Password?' link. Both forms have small red asterisks indicating required fields.




#### 4.2 Home Page:

The image shows the 'Digital Library' home page. At the top is a banner with the text 'Digital Library' in a stylized font, accompanied by icons of books and a person reading. Below the banner is a sidebar on the left with a vertical list of green buttons: ADMIN, ADD BOOK, BOOK REPORT, BOOK REQUESTS, ADD STUDENT, STUDENT REPORT, ISSUE BOOK, ISSUE REPORT, and LOGOUT. The main content area on the right has a green header 'ADD NEW BOOK'. Below this header is a form with the following fields: 'Book Name:', 'Detail:', 'Autor:', 'Publication', 'Branch:' (with checkboxes for 'It', 'Civil', 'ec', and 'electrical'), 'Price:', 'Quantity:', and 'Book Photo' (with a 'Choose File' button and the text 'No file chosen'). A 'SUBMIT' button is located at the bottom of the form.



#### 4.3 Request Book



|              |   |                   |                     |            |       |                              |
|--------------|---|-------------------|---------------------|------------|-------|------------------------------|
| Welcome      | Request Book  |                   |                     |            |       |                              |
| My Account   | Image   | Book Name         | Book Authour        | branch     | price | Request Book                 |
| Request Book |  | Scott Gallagher   | no idea             | it         | 20    | <a href="#">Request Book</a> |
| Book Report  |  | Ferris McLaughlin | Est voluptates offi | electrical | 157   | Request Book                 |
| LOGOUT       |  | harry             | Ut dolore culpa ex  | it         | 3     | Request Book                 |


#### 4.4 Issue Book



|                |              |                      |
|----------------|--------------|----------------------|
| ADMIN          | ISSUE BOOK   |                      |
| ADD BOOK       | Choose Book: | Scott Gallagher      |
| BOOK REPORT    | Days         | salman               |
| BOOK REQUESTS  |              | <input type="text"/> |
| ADD STUDENT    |              | <input type="text"/> |
| STUDENT REPORT |              | <input type="text"/> |
| ISSUE BOOK     |              | <input type="text"/> |
| ISSUE REPORT   |              | <input type="text"/> |
| LOGOUT         |              | <input type="text"/> |


Days:  SUBMIT

#### 4.5 Student Record



| ADMIN          | Student RECORD    |                        |         |        |
|----------------|-------------------|------------------------|---------|--------|
|                | Name              | Email                  | Type    | Delete |
| ADD BOOK       | salman            | idno22382@gmail.com    | student | Delete |
| BOOK REPORT    | Randall Burch     | voqo@mailinator.com    | teacher | Delete |
| BOOK REQUESTS  | Gabriel Daugherty | bipacer@mailinator.com | teacher | Delete |
| ADD STUDENT    | salmannew         | 1234@gmail.com         | teacher | Delete |
| STUDENT REPORT |                   |                        |         |        |
| ISSUE BOOK     |                   |                        |         |        |
| ISSUE REPORT   |                   |                        |         |        |
| LOGOUT         |                   |                        |         |        |

#### 4.6 Add Person



| ADMIN          | ADD Person  |                                       |
|----------------|---|---------------------------------------|
| ADD BOOK       | Name: <input type="text"/>                              |                                       |
| BOOK REPORT    | Pasword: <input type="text"/>                           |                                       |
| BOOK REQUESTS  | Email: <input type="text"/>                             |                                       |
| ADD STUDENT    | Choose type: <div>student<br/>student<br/>teacher</div> | <input type="button" value="SUBMIT"/> |
| STUDENT REPORT |   |                                       |
| ISSUE BOOK     |   |                                       |
| ISSUE REPORT   |   |                                       |
| LOGOUT         |   |                                       |



## **CHAPTER 5**

### **CONCLUSION**

A Library Management System is a project that tries to create an automated and computerised version for a library so that the daily work of a library can be managed and monitored easily and efficiently. Earlier, the librarian used to manage the whole work in manual mode in the form of files and record books. Also, the process of adding new books, new students, issuing and returning books had to be managed in a manual manner which is very slow and inefficient. The library management system resolves this problem and provide a better solution to this. It provides a user-friendly interface application to the librarian where he can do all the operations of a library very easily. The application mainly consists of three modules which are admin module, librarian module and student module. The admin module will be managed by the system administrator. He manages the overall functioning of the application. The librarian module will be accessed by the librarian. He can perform various operations inside the application such as add new students, new books to the database, issuing and returning of books, updating student's details, book's details, generating weekly/monthly reports etc. The student module can be accessed by the registered students only. The operations that can be performed by the student includes: view all books available in the library, search the availability of a particular book, number of books he has issued, overall fine he has to pay etc. These three modules are interconnected with each other and also with the database. The application is built using Java technology and Sql database.

## REFERENCES

- Sharma, C. K.; Singh, Kiran (2005), Library Management, ISBN 978-81-269-0452-5
- McClure, C. R. (1980). "LIBRARY MANAGERS: Can They Manage? Will They Lead?" Library Journal: 2391.
- Wellish, Hans H. "Dewey Decimal Classification, Universal Decimal Classification, and the Broad System of Ordering: The Evolution of Universal Ordering Systems". College of Library and Information Sciences, University of Maryland.
- Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites – by Robin Nixon
- Programming PHP – by Kevin Tatroe, Peter MacIntyre & Rasmus Lerdorf  
“Foreword By: Michael Bourque”
- A beginner’s guide to HTML, CSS, Javascript, and Web Graphics, by Jennifer Niederst Robbins
- Fullstack Vue: The Complete Guide to Vue.js by Hassan Djirdeh, Nate Murray, and Ari Lerner
- Oracle MySQL online documentation. MySQL Reference Manual. MySQL Reference Manual 8.0 (includes NDB Cluster 8.0 (GA))
- Introduction of Database Management System by C Vairavel, Department of Computer Science and Engineering Galgotias University, Yamuna Expressway Greater Noida, Uttar Pradesh.
- 2019 JETIR March 2019, Volume 6, Issue 3
- Ivan Bayross, “HTML, DHTML, Javascript, Pearl & CGI”, BPB Publication.
- Ramesh Bangia, “Internet and Web Design”, New Age International.
- Jackson, “Web Technologies” Pearson Education.