

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI



REPORT ON DBMS MINI PROJECT(21CSL55)

“LIBRARY MANAGEMENT SYSTEM”

Submitted in partial fulfillment for the award of Degree of,

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE &ENGINEERING

By

Student Name : Nikita Patil

USN:2BU21CS075

Student Name : Pranjal B.

USN:2BU21CS086

Under the Guidance of

Dr. S.F. RODD

Professor, CSE Dept



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Accredited by NBA

S.S EDUCATION TRUST

S. G. BALEKUNDRI INSTITUTE OF TECHNOLOGY



S. S. Education Trust's
S.G Balekundri Institute of Technology

Shivabasava Nagar Belagavi, Karnataka, India – 590010



2023– 2024

S. G. BALEKUNDRI INSTITUTE OF TECHNOLOGY

Shivabasavanagar, Belagavi-10, Karnataka.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Accredited by NBA

CERTIFICATE

This is to certify that the DBMS mini project entitled **“Library Management System”** has been successfully completed by

STUDENT NAME : Nikita Patil USN:2BU21CS075

STUDENT NAME : Pranjal B USN:2BU21CS086

the bonafide students of **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,**
S.G. BALEKUNDRI INSTITUTE OF TECHNOLOGY of the **VISVESVARAYA**
TECHNOLOGICAL UNIVERSITY, BELAGAVI during the year 2023–2024. It is certified that
all corrections/suggestions indicated for Internal Assessment have been incorporated in the
report deposited in the departmental library. The Project Work (21CSL55) has been approved
as it satisfies the academic requirements in respect of DBMS mini Project work prescribed for
the Bachelor of Engineering Degree.

Dr. S.F. Rodd
Project Guide

Dr. B. S. Halakarnimath
Head of the Department

Dr.B. R. Patagundi
Principal

TABLE OF CONTENTS

Ch.no	Description	Pg.no
	Acknowledgement.....	i
	Abstract.....	ii
	List of figures.....	iii
	List of tables.....	iv
1.	INTRODUCTION	1
	1.1 Problem statement	1
	1.2 Objectives	1
	1.3 SQL commands	2
2.	METHODOLOGIES	3
	2.1 About backend connection	3
	2.2 Introduction to server	4
3.	SYSTEM REQUIREMENTS AND SPECIFICATIONS	5
	3.1 Software requirements	5
	3.2 Hardware requirements	5
4.	SYSTEM DESIGN	6
	4.1 ER Diagram	6
	4.2 Schema Diagram	7
5.	IMPLEMENTATION	8
	5.1 Technologies used	8
	5.2 Database creation	9
	5.3 Implementation codes	12

6.	RESULTS	15
6.1	Home page	15
6.2	User login page	15
6.3	User signup page	15
6.4	Admin login page	16
6.5	User dashboard page	16
6.6	Issued book page	16
6.7	User change password page	17
6.8	My profile page	17
6.9	Admin dashboard page	17
6.10	Add category page	18
6.11	Manage category page	18
6.12	Add author page	18
6.13	Manage author page	19
6.14	Add book page	19
6.15	Manage book page	19
6.16	Issue new book page	20
6.17	Manage issued book page	20
6.18	Manage reg students page	20
	REFERENCES	21

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, with gratitude we acknowledge all those whose guidance and encouragement served as beacon of light and crowned the effort with success.

We thank our project guide **Dr.S F Rodd** ,Professor in Department of Computer Science & Engineering, who has been our source of inspiration. He has been especially enthusiastic in giving his valuable guidance and critical reviews.

The selection of this project work as well as the timely completion is mainly due to the interest and persuasion of my project coordinator **Mrs. Sheetal Kokatanoor**, Assistant Professor, Department of Computer Science & Engineering. We will remember her contribution for ever.

We sincerely thank, **Dr. B. S. Halakarnimath**, Professor and Head, Department of Computer Science & Engineering who has been the constant driving force behind the completion of the project.

We thank Principal **Dr. B. R. Patagundi**, for his constant help and support throughout.

We are also indebted to **Management of S. G. Balekundri Institute of Technology, Belagavi** for providing an environment which helped us in completing the project.

Also, we thank all the teaching and non-teaching staff of Department of Computer Science & Engineering for the help rendered.

Finally, we would like to thank my parents and friends whose encouragement and support was invaluable.

STUDENT NAME: Nikita Patil USN: 2BU21CS075

STUDENT NAME: Pranjal B. USN: 2BU21CS086

ABSTRACT

The "Student Library Management System" is a comprehensive software solution designed to streamline and automate the management of library resources in educational institutions. This project aims to provide an efficient platform for librarians and students to manage library operations effectively. The system encompasses various modules to handle tasks such as book management, student registration, book issuance, returns, and administrative functions.

Key features of the system include:

Student Registration: Enables the registration of students into the library system, capturing essential details such as student ID, full name, contact information, and login credentials.

Book Management: Facilitates the management of books available in the library, including adding new books, updating existing book information, and categorizing books based on genres or subjects.

Book Search and Availability: Provides a search interface for students to find books based on title, author, category, or ISBN number. The system also displays real-time availability status to inform students about the availability of desired books.

Book Issuance and Returns: Allows students to borrow books from the library by scanning their student ID or entering their credentials. The system records issuance details such as book ID, student ID, issuance date, and expected return date. Additionally, it manages the return process and calculates fines for overdue books.

Admin Dashboard: Offers an administrative dashboard for librarians and administrators to monitor library activities, view statistics, generate reports, and perform configuration tasks.

Notifications and Reminders: Sends automated notifications and reminders to students regarding upcoming due dates, overdue books, and other relevant announcements.

Security and Access Control: Implements robust security measures to safeguard sensitive data and restrict unauthorized access to the system functionalities.

The "Student Library Management System" enhances the efficiency of library operations, improves access to educational resources for students, and provides valuable insights for administrators to optimize resource allocation and decision-making. This project report explores the system architecture, design methodologies, implementation details, user documentation, and future enhancements to support continuous improvement and scalability of the system.

LIST OF FIGURES

Figure 1.1	Configuration code snippet	3
Figure 1.2	ER Diagram	6
Figure 1.3	Schema Diagram	7
Figure 1.4	Table admin	9
Figure 1.5	Table books	9
Figure 1.6	Table authors	10
Figure 1.7	Table category	10
Figure 1.8	Table issued_book_details	11
Figure 1.9	Table student	11
Figure 1.10	Home page	15
Figure 1.11	User login page	15
Figure 1.12	User signup page	15
Figure 1.13	Admin login page	16
Figure 1.14	User dashboard page	16
Figure 1.15	Issued book page	16
Figure 1.16	User change password page	17
Figure 1.17	My profile page	17
Figure 1.18	Admin dashboard page	17
Figure 1.19	Add category page	18
Figure 1.20	Manage category page	18
Figure 1.21	Add author page	18
Figure 1.22	Manage author page	19
Figure 1.23	Add book page	19
Figure 1.24	Manage book page	19

Figure 1.25	Issue new book page	20
Figure 1.26	Manage issued book page	20
Figure 1.27	Manage reg students page	20

LIST OF TABLES

Table 1.1	SQL commands	10
Table 1.2	ER relationships	14

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

In the physical library management system, manual processes, outdated technology, and disjointed systems lead to inefficiencies and errors. Our project aims to develop a modern library management system to automate processes, integrate systems, and enhance user experience. By automating cataloging and record-keeping, integrating systems, and implementing online access, we streamline operations, ensure data consistency, and improve accessibility. Real-time reporting and analytics capabilities empower administrators with insights for informed decision-making. Through this project, we address these challenges, transforming the library management system into a modern, efficient, and user-friendly platform.

1.2 OBJECTIVES:

- **Automate processes:** Develop automated cataloging and record-keeping systems to reduce manual errors and administrative workload.
- **Integrate systems:** Consolidate disparate systems into a unified platform to ensure data consistency and streamline administrative tasks.
- **Enhance user experience:** Implement online access and self-service functionalities to improve accessibility and convenience for users.
- **Enable real-time reporting:** Introduce reporting and analytics capabilities to provide administrators with insights for informed decision-making.
- **Modernize library management:** Transform the physical library management system into a modern, efficient, and user-centric platform to meet the evolving needs of librarians and users.
- **Ensure scalability and adaptability:** Design the system to be scalable and adaptable to accommodate future growth, technological advancements, and evolving user needs, ensuring its long-term effectiveness and relevance in the ever-changing library landscape.

1.3 SQL COMMANDS:

SQL Command	Use	Example
SELECT	Retrieve data from one or more tables	<code>SELECT * FROM tblbooks;</code> <code>SELECT BookName, BookPrice FROM tblbooks WHERE CatId = 5;</code>
INSERT	Insert new records into a table	<code>INSERT INTO tblstudents (StudentId, FullName, EmailId) VALUES ('SID013', 'John Smith', 'john@example.com');</code>
UPDATE	Update existing records in a table	<code>UPDATE tblbooks SET BookPrice = 25.00 WHERE id = 1;</code>
DELETE	Delete records from a table	<code>DELETE FROM tblauthors WHERE id = 3;</code>
JOIN	Combine rows from two or more tables based on a related column	<code>SELECT b.BookName, s.FullName FROM tblissuedbookdetails ibd JOIN tblbooks b ON ibd.BookId = b.id JOIN tblstudents s ON ibd.StudentID = s.StudentId;</code>
GROUP BY	Group rows that have the same values into summary rows	<code>SELECT c.CategoryName, COUNT(*) AS TotalBooks FROM tblbooks b JOIN tblcategory c ON b.CatId = c.id GROUP BY c.CategoryName;</code>
ORDER BY	Sort the result set in ascending or descending order	<code>SELECT * FROM tblbooks ORDER BY BookName ASC;</code>
WHERE	Filter records based on specified conditions	<code>SELECT * FROM tblbooks WHERE CatId = 5;</code>
LIMIT	Limit the number of records returned	<code>SELECT * FROM tblbooks LIMIT 5;</code>

Table 1.1 SQL commands

CHAPTER 2

METHODOLOGY

2.1 ABOUT BACKEND CONNECTION:

Database Layer: This layer consists of the database management system (DBMS) where all data related to the library management system is stored. In this project, the DBMS used is MySQL, indicated by the SQL commands and structure defined for tables such as `tblbooks`, `tblstudents`, etc.

Application Layer: The application layer contains the business logic and application code responsible for processing user requests, interacting with the database, and generating responses. This layer can be implemented using various programming languages and frameworks. We have used PHP for server-side scripting.

Connection Setup:

The connection setup involves configuring the backend to establish a connection with the database management system. This typically includes:

- **Database Configuration:** Setting up the database server, creating the necessary databases and tables, and configuring user permissions and access rights.
- **Backend Configuration:** Configuring the backend application to connect to the database. This involves specifying the database host, port, username, password, and other connection parameters. In PHP applications, this is often done using functions like `mysqli_connect()` or PDO (PHP Data Objects) to establish a connection to the MySQL database.

This is the configuration code used in our project for backend connection:

```
1  <?php
2  // DB credentials.
3  define('DB_HOST','localhost');
4  define('DB_USER','root');
5  define('DB_PASS','');
6  define('DB_NAME','library');
7  // Establish database connection.
8  try
9  {
10 $dbh = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME,DB_USER, DB_PASS,array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES 'utf8'"));
11 }
12 catch (PDOException $e)
13 {
14     exit("Error: " . $e->getMessage());
15 }
16 ?>
```

Figure 1.1 Configuration code snippet

- **DB credentials:** These constants (``DB_HOST``, ``DB_USER``, ``DB_PASS``, ``DB_NAME``) store the database connection details, including the host (localhost), username (root), password (blank in this case), and database name (library).
- **Establish database connection:** The ``try-catch`` block attempts to establish a connection to the MySQL database using PDO. The ``PDO()`` constructor is used to create a new PDO object, passing the database DSN (Data Source Name) as the first argument, which includes the host, database name, username, and password.
- **Error handling:** If an exception (PDOException) occurs during the connection attempt, the ``catch`` block captures the exception, and an error message is displayed, providing details about the error.

This code snippet demonstrates the backend connection setup in a PHP application, where the application connects to the MySQL database using PDO. It's a common approach to establishing database connections in PHP applications, ensuring secure and reliable communication between the application and the database management system.

2.2 INTRODUCTION TO SERVER:

To host a library management system, you need a server configured with the following components:

- **Operating System:** Choose a stable and secure operating system like Linux (e.g., Ubuntu, CentOS) or Windows Server.
- **Web Server:** Install and configure a web server such as Apache or Nginx to handle HTTP requests.
- **Database Server:** Set up a database management system (e.g., MySQL, PostgreSQL) to store and manage library data.
- **Server-Side Scripting:** Use a server-side scripting language like PHP, Python, or Node.js to build the backend logic of the system.
- **Backup and Monitoring:** Set up regular backups of data and monitor server performance to ensure reliability and availability.

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATIONS

3.1 SOFTWARE REQUIREMENTS:

1. Web Server:

Apache HTTP Server: To serve web pages locally during development.

XAMPP: Bundled package including Apache, MySQL, PHP, and phpMyAdmin for local development environment setup.

2. Database Management System:

MySQL: Database system for storing and managing library data.

3. Development Tools:

Visual Studio Code (VSCode): Integrated Development Environment (IDE) for writing and editing code.

Browser: Google Chrome for testing and debugging web pages.

3.1 HARDWARE REQUIREMENTS:

Server Hardware (for Deployment):

Processor: Dual-core or higher for handling server operations efficiently.

RAM: At least 4GB, though 8GB or more would be beneficial for smoother performance.

Storage: 512 SSD storage for faster read/write operations.

Network: Stable internet connection for serving requests to client devices.

CHAPTER 4

SYSTEM DESIGN

4.1 ER DIAGRAM

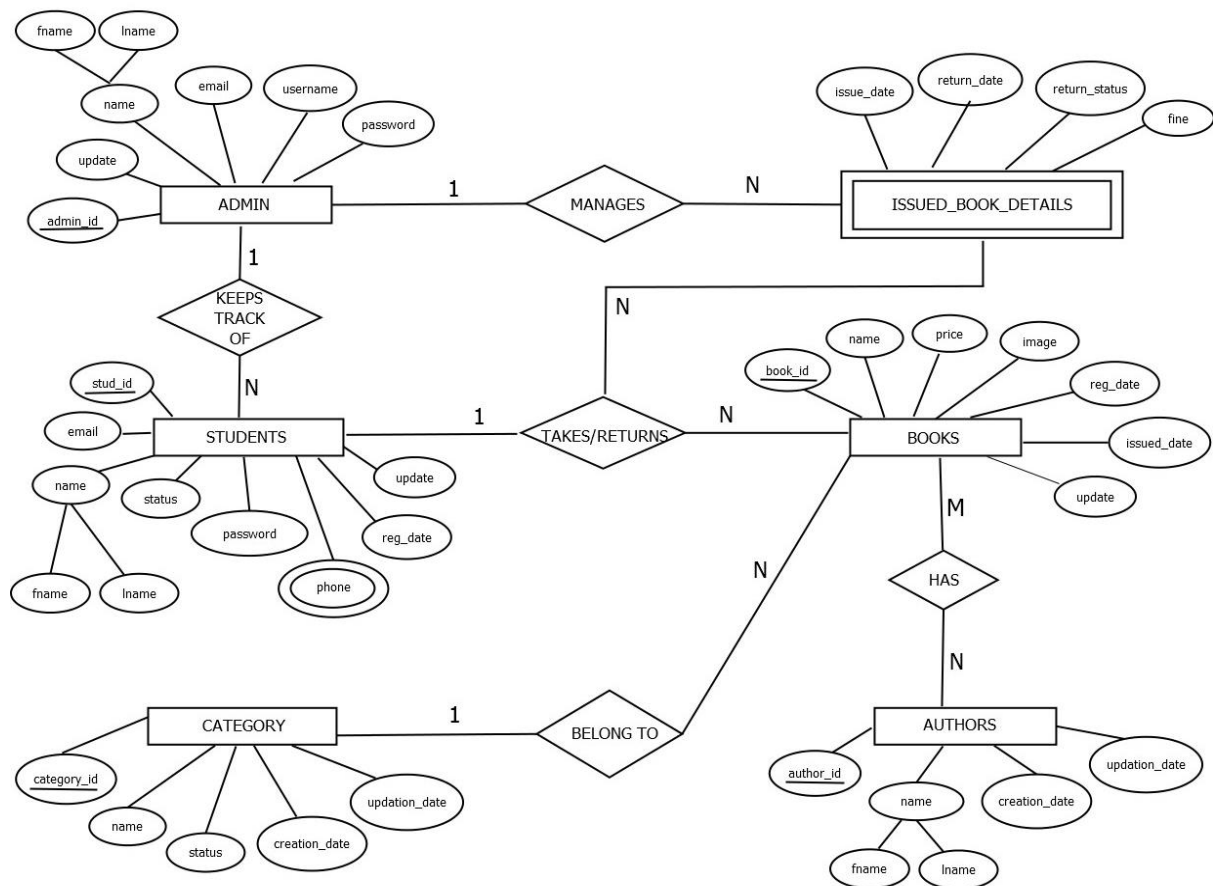


Figure 1.2 ER Diagram

Relationships and Cardinality ratios:

Entity 1	Relationship	Entity 2	Cardinality Ratio
Admin	Manages	Books	1: M
Admin	Manages	IssuedBookDetails	1: M
Admin	Keepstrackof	Students	1: M
Students	Take/Return	Books	1: M
Books	BelongsTo	Category	M :1
Books	Have	Authors	M: N

Table 1.2 ER Relationships

4.2 SCHEMA DIAGRAM:

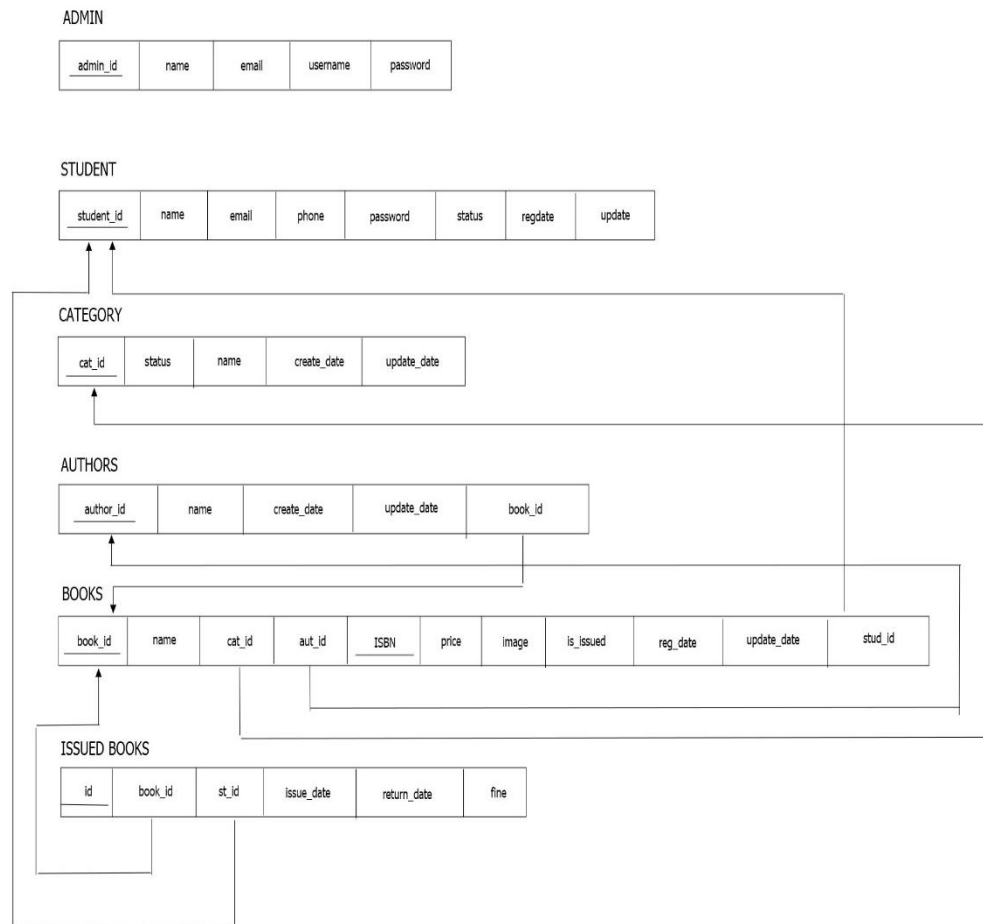


Figure 1.3 Schema Diagram

- **Admin:-** Primary Key: id
- **Books:-** Primary Key: id

Foreign Key: Admin_id (refers to Admin's id)

Foreign Key: Category_id (refers to Category's id)

- **IssuedBookDetails:-** Primary Key: id

Foreign Key: BookId (refers to Books' id)

Foreign Key: StudentID (refers to Students' id)

- **Students:-** Primary Key: id
- **Category:-** Primary Key: id
- **Authors:-** Primary Key: id

CHAPTER 5

IMPLEMENTATION

5.1 TECHNOLOGIES USED:

Development Environment

- The Library Management System was developed using a combination of software tools and technologies.
- The primary development environment consisted of:
 - Visual Studio Code (VSCode) as the Integrated Development Environment (IDE).
 - XAMPP for providing a local server environment including Apache, MySQL, and PHP.
 - MySQL Workbench for database design and management.

Database Design

- The database schema was designed to capture all relevant information for managing the library system effectively.
- Detailed discussions on the tables, columns, relationships, and constraints were carried out to ensure data integrity and efficiency.
- Normalization techniques were applied to reduce redundancy and improve data organization.

Backend Development

- Backend logic was implemented using PHP scripting language to handle database interactions and business logic.
- PHP scripts were responsible for querying the database, processing user requests, and generating dynamic content for the frontend.
- Object-oriented programming (OOP) principles were employed to modularize code and improve maintainability.

Frontend Development

- The user interface was developed using HTML, CSS, and JavaScript.
- Responsive design principles were applied to ensure compatibility across different devices and screen sizes.
- Bootstrap framework was utilized to streamline the frontend development process and enhance UI consistency.

5.2 DATABASE CREATION:

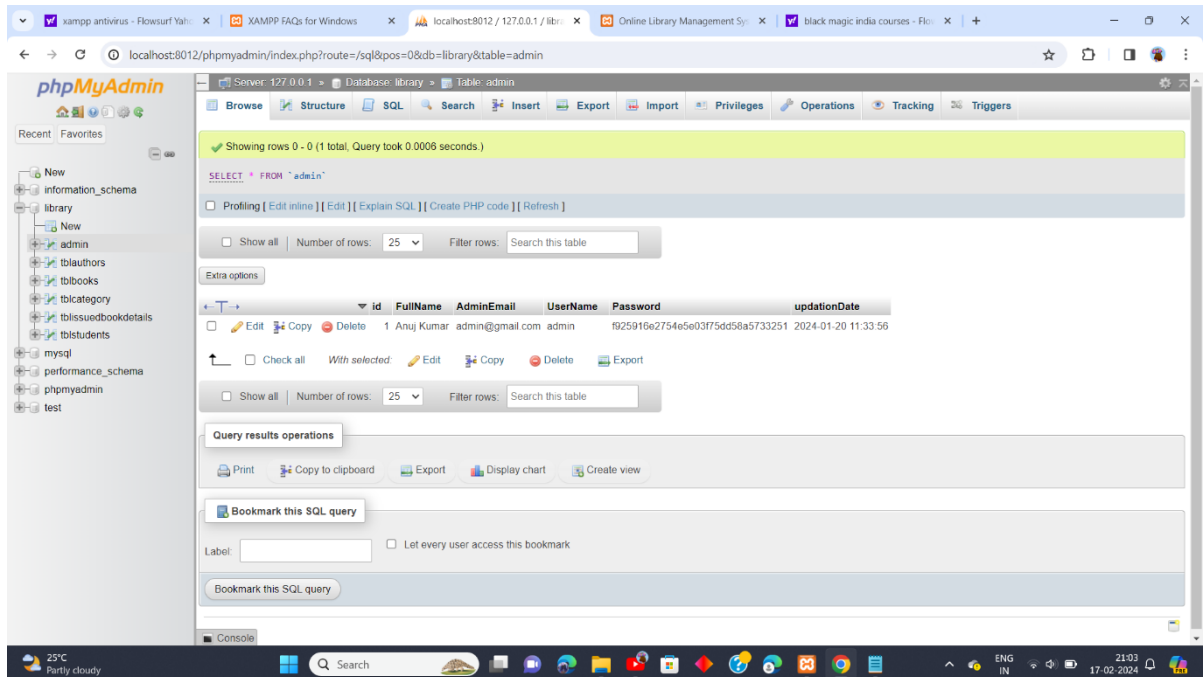


Figure 1.4 Table admin

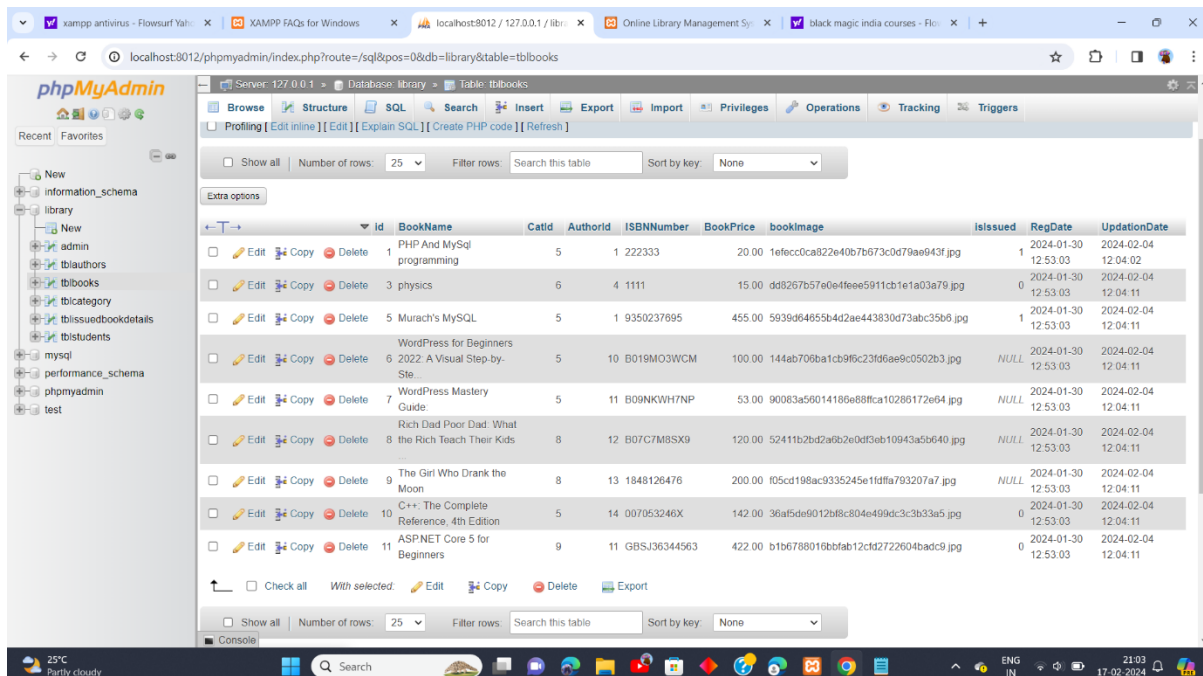


Figure 1.5 Table books

Showing rows 0 - 10 (11 total, Query took 0.0003 seconds)

```
SELECT * FROM `tblauthors`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	AuthorName	creationDate	UpdationDate
<input type="checkbox"/>	1	Anuj kumar	2024-01-25 12:53:03	2024-02-04 12:04:19
<input type="checkbox"/>	2	Chetan Bhagatt	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	3	Anita Desai	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	4	HC Verma	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	5	R.D. Sharma	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	9	fwdfwr	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	10	Dr Andy Williams	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	11	Kyle Hill	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	12	Robert T. Kyosak	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	13	Kelly Barnhill	2024-01-25 12:53:03	2024-02-04 12:04:26
<input type="checkbox"/>	14	Harbert Schildt	2024-01-25 12:53:03	2024-02-04 12:04:26

Figure 1.6 Table authors

Showing rows 0 - 5 (6 total, Query took 0.0004 seconds)

```
SELECT * FROM `tblcategory`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	CategoryName	Status	CreationDate	UpdationDate
<input type="checkbox"/>	4	Romantic	1	2024-01-31 12:53:03	2024-02-04 12:03:43
<input type="checkbox"/>	5	Technology	1	2024-01-31 12:53:03	2024-02-04 12:03:51
<input type="checkbox"/>	6	Science	1	2024-01-31 12:53:03	2024-02-04 12:03:51
<input type="checkbox"/>	7	Management	1	2024-01-31 12:53:03	2024-02-04 12:03:51
<input type="checkbox"/>	8	General	1	2024-01-31 12:53:03	2024-02-04 12:03:51
<input type="checkbox"/>	9	Programming	1	2024-01-31 12:53:03	2024-02-04 12:03:51

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view

Bookmark this SQL query

Figure 1.7 Table category

Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

```
SELECT * FROM `tblissuedbookdetails`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	BookId	StudentID	IssuesDate	ReturnDate	RetrunStatus	fine
<input type="checkbox"/>	7	5	SID011	2024-02-01 11:15:57	NULL	NULL	NULL
<input type="checkbox"/>	8	1	SID002	2024-02-01 11:15:57	2024-02-04 12:03:08	1	0
<input type="checkbox"/>	9	10	SID009	2024-02-01 11:15:57	2024-02-04 12:03:08	1	0
<input type="checkbox"/>	10	11	SID009	2024-02-01 11:15:57	2024-02-04 12:03:08	1	0
<input type="checkbox"/>	11	1	SID012	2024-02-01 11:15:57	NULL	NULL	NULL
<input type="checkbox"/>	12	10	SID012	2024-02-01 11:15:57	2024-02-04 12:03:08	1	5

Figure 1.8 Table issued_book_details

Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

```
SELECT * FROM `tblstudents`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	Studentid	FullName	Emailid	MobileNumber	Password	Status	RegDate	UpdationDate
<input type="checkbox"/>	1	SID002	Anuj kumar	anujk@gmail.com	9865472555	f925916e2754e5e03f75dd58a5733251	1	2024-01-31 12:53:03	2024-02-04 12:02:36
<input type="checkbox"/>	4	SID005	sdfs	csfsd@dfsfs.com	8569710025	92228410fc8b872914e023160cf4ee8f	1	2024-01-31 12:53:03	2024-02-04 12:02:42
<input type="checkbox"/>	8	SID009	test	test@gmail.com	2359874527	f925916e2754e5e03f75dd58a5733251	1	2024-01-31 12:53:03	2024-02-04 12:02:42
<input type="checkbox"/>	9	SID010	Amit	amit@gmail.com	8585856224	f925916e2754e5e03f75dd58a5733251	1	2024-01-31 12:53:03	2024-02-04 12:02:42
<input type="checkbox"/>	10	SID011	Sarita Pandey	sarita@gmail.com	4672423754	f925916e2754e5e03f75dd58a5733251	1	2024-01-31 12:53:03	2024-02-04 12:02:42
<input type="checkbox"/>	11	SID012	John Doe	john@test.com	1234569870	f925916e2754e5e03f75dd58a5733251	1	2024-01-31 12:53:03	2024-02-04 12:02:42

Figure1.9 Table students

5.3 IMPLEMENTATION CODE (INDEX.PHP)

```
<?php
session_start();
error_reporting(0);
include('includes/config.php');
if($_SESSION['login']!=""){
$_SESSION['login']="";
}
if(isset($_POST['login']))
{
$email=$_POST['emailid'];
$password=md5($_POST['password']);
$sql ="SELECT EmailId,Password,StudentId,Status FROM tblstudents WHERE
EmailId=:email and Password=:password";
$query= $dbh -> prepare($sql);
$query-> bindParam(':email', $email, PDO::PARAM_STR);
$query-> bindParam(':password', $password, PDO::PARAM_STR);
$query-> execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
    foreach ($results as $result) {
        $_SESSION['stdid']=$result->StudentId;
        if($result->Status==1)
        {
            $_SESSION['login']=$_POST['emailid'];
            echo "<script type='text/javascript'> document.location ='dashboard.php'; </script>";
        } else {
            echo "<script>alert('Your Account Has been blocked .Please contact admin');</script>";
        }
    }
}
else{
    echo "<script>alert('Invalid Details');</script>";
}
}

?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1" />
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>Online Library Management System | </title>
    <!-- BOOTSTRAP CORE STYLE -->
    <link href="assets/css/bootstrap.css" rel="stylesheet" />
```

```

<!-- FONT AWESOME STYLE -->
<link href="assets/css/font-awesome.css" rel="stylesheet" />
<!-- CUSTOM STYLE -->
<link href="assets/css/style.css" rel="stylesheet" />
<!-- GOOGLE FONT -->
<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet'
type='text/css' />
</head>
<body>
  <!--MENU SECTION START-->
  <?php include('includes/header.php');?>
  <!-- MENU SECTION END-->
  <div class="content-wrapper">
    <div class="container">
      <!--Slider---->
      <div class="row">
        <div class="col-md-10 col-sm-8 col-xs-12 col-md-offset-1">
          <div id="carousel-example" class="carousel slide slide-bdr" data-
ride="carousel" >
            <div class="carousel-inner">
              <div class="item active">
                
              </div>
              <div class="item">
                
              </div>
              <div class="item">
                
              </div>
            </div>

            <!--INDICATORS-->
            <ol class="carousel-indicators">
              <li data-target="#carousel-example" data-slide-to="0" class="active"></li>
              <li data-target="#carousel-example" data-slide-to="1"></li>
              <li data-target="#carousel-example" data-slide-to="2"></li>
            </ol>

            <!--PREVIUS-NEXT BUTTONS-->

            <a class="left carousel-control" href="#carousel-example" data-slide="prev">
              <span class="glyphicon glyphicon-chevron-left"></span>
            </a>
            <a class="right carousel-control" href="#carousel-example" data-slide="next">
              <span class="glyphicon glyphicon-chevron-right"></span>
            </a>

          </div>
        </div>
      </div>
    </div>
  <hr />
  <div class="row pad-botm">

```

```

<div class="col-md-12">
<h4 class="header-line">USER LOGIN FORM</h4>
</div>
</div>
<a name="ulogin"></a>
<!--LOGIN PANEL START-->
<div class="row">
<div class="col-md-6 col-sm-6 col-xs-12 col-md-offset-3">
<div class="panel panel-info">
<div class="panel-heading">

LOGIN FORM
</div>
<div class="panel-body">
<form role="form" method="post">
<div class="form-group">
<label>Enter Email id</label>
<input class="form-control" type="text" name="emailid" required autocomplete="off" />
</div>
<div class="form-group">
<label>Password</label>
<input class="form-control" type="password" name="password" required
autocomplete="off" />
<p class="help-block"><a href="user-forgot-password.php">Forgot Password</a></p>
</div>
<button type="submit" name="login" class="btn btn-info">LOGIN </button> | <a
href="signup.php">Not Register Yet</a>

</form>

</div>
</div>
</div>
</div>

<!--LOGIN PABNEL END-->
</div>
</div>
<!-- CONTENT-WRAPPER SECTION END-->
<?php include('includes/footer.php');?>
<!-- FOOTER SECTION END-->
<script src="assets/js/jquery-1.10.2.js"></script>
<!-- BOOTSTRAP SCRIPTS -->
<script src="assets/js/bootstrap.js"></script>
<!-- CUSTOM SCRIPTS -->
<script src="assets/js/custom.js"></script>
</body>
</html>

```


CHAPTER 6

RESULTS

6.1 HOME PAGE

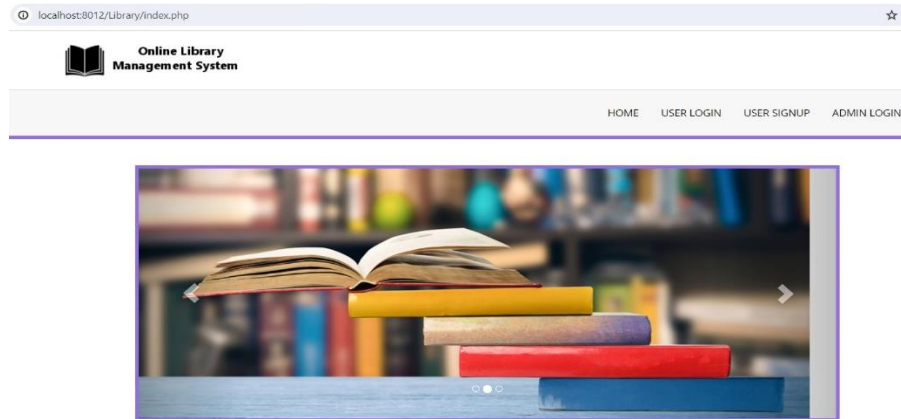


Figure 1.10 Home page

6.2 USER LOGIN PAGE

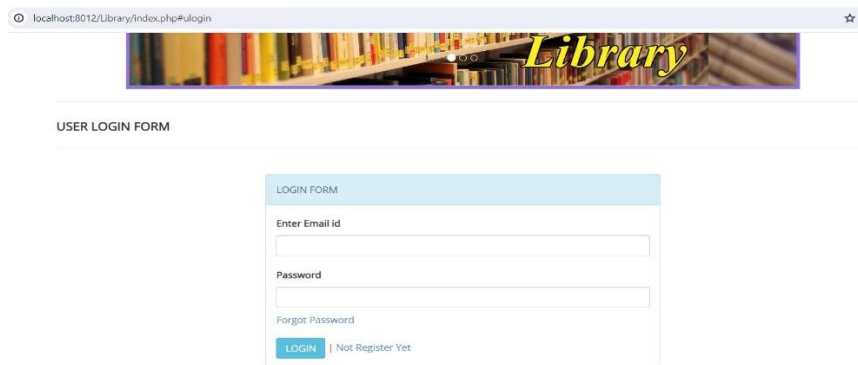


Figure 1.11 User login page

6.3 USER SIGNUP PAGE

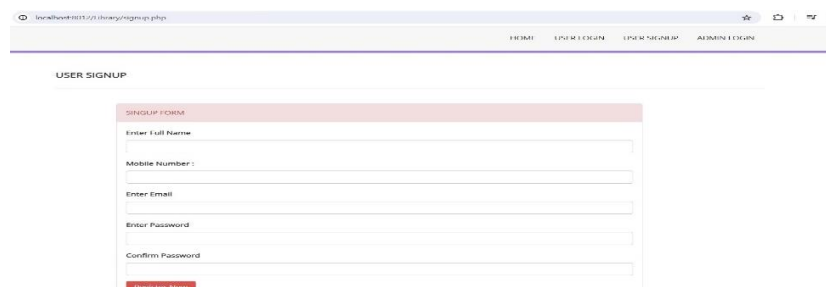


Figure 1.12 User signup page

6.4 ADMIN LOGIN PAGE

The screenshot shows the Admin Login page of the Online Library Management System. The page has a header with the system name and navigation links: HOME, USER LOGIN, USER SIGNUP, and ADMIN LOGIN. The main content area is titled "ADMIN LOGIN FORM" and contains a "LOGIN FORM" box with fields for "Enter Username" and "Password", and a "LOGIN" button.

Figure 1.13 Admin Login page

6.5 USER DASHBOARD PAGE

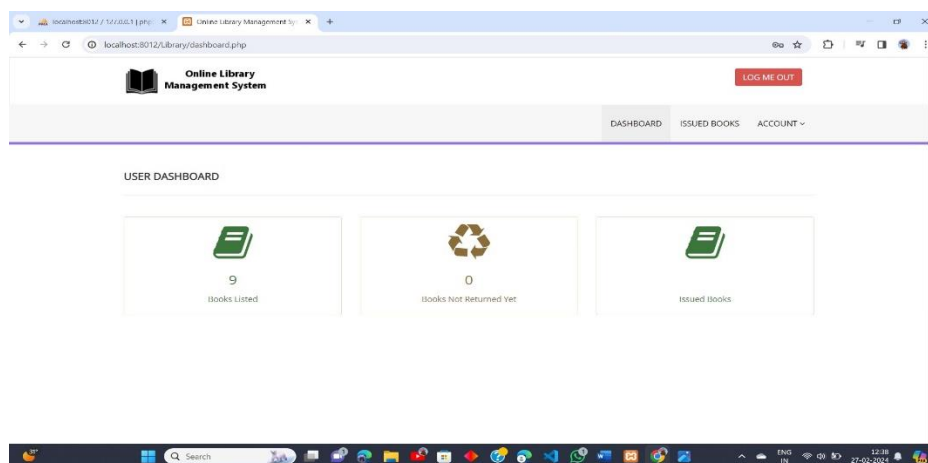


Figure 1.14 User Dashboard page

6.6 ISSUED BOOK PAGE

The screenshot shows the Issued Books page of the Online Library Management System. The page has a header with the system name, a "LOG ME OUT" button, and navigation links: DASHBOARD, ISSUED BOOKS, and ACCOUNT. The main content area is titled "MANAGE ISSUED BOOKS" and contains a table of issued books. The table has columns for Book Name, ISBN, Issued Date, Return Date, and Fine (in USD). There are two entries in the table. Below the table, there is a pagination control showing "Showing 1 to 2 of 2 entries" and "Previous 1 Next".

#	Book Name	ISBN	Issued Date	Return Date	Fine (in USD)
1	ASP.NET Core 5 for Beginners	GB936344563	2024-02-01 11:15:57	2024-02-04 12:03:08	0
2	C++ The Complete Reference, 4th Edition	007053246X	2024-02-01 11:15:57	2024-02-04 12:03:08	0

Figure 1.15 Issued Books page

6.7 USER CHANGE PASSWORD PAGE

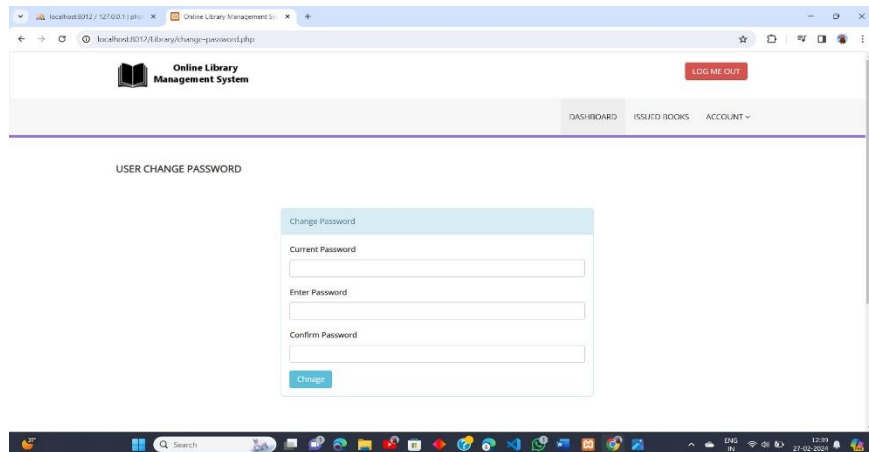


Figure 1.16 User Change password page

6.8 MY PROFILE PAGE

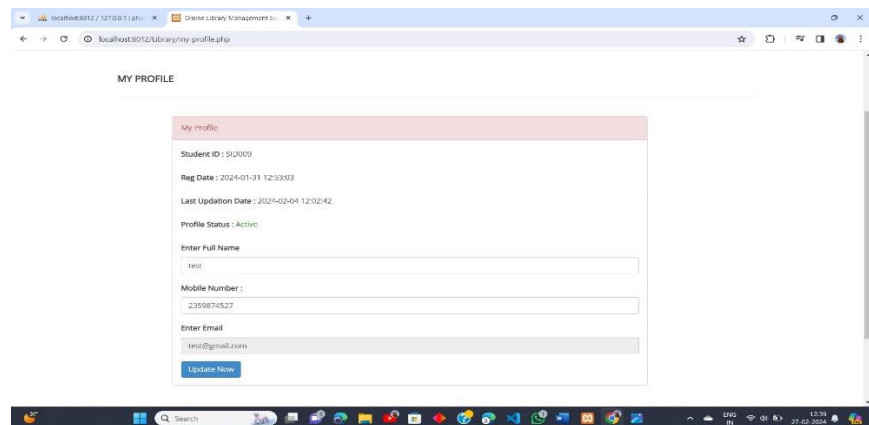


Figure 1.17 My Profile page

6.9 ADMIN DASHBOARD PAGE

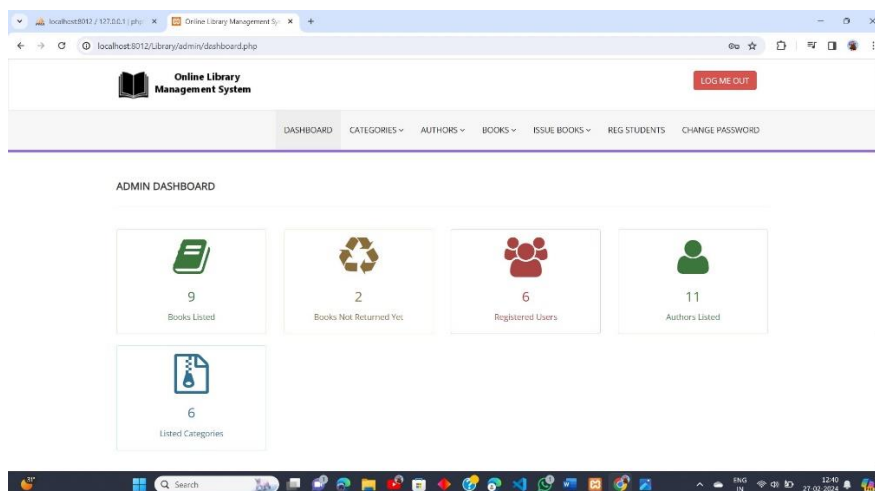


Figure 1.18 Admin Dashboard page

6.10 ADD CATEGORY PAGE

Online Library Management System

DASHBOARD CATEGORIES AUTHORS BOOKS ISSUE BOOKS REG STUDENTS CHANGE PASSWORD

ADD CATEGORY

Category Info

Category Name

Status

☒ Active

☐ Inactive

Create

© 2024 Online Library Management System | Designed by : PIRGURUKUL

Figure 1.19 Add category page

6.11 MANAGE CATEGORIES PAGE

MANAGE CATEGORIES

Categories Listing

to records per page Search

#	Category	Status	Creation Date	Update Date	Action
1	Romanic	Active	2024-01-31 12:53:03	2024-02-04 12:03:43	Edit Delete
2	Technology	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete
3	Science	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete
4	Management	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete
5	General	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete
6	Programming	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete

Showing 1 to 6 of 6 entries

Previous 1 Next

Figure 1.20 Manage category page

6.12 ADD AUTHOR PAGE

Online Library Management System

DASHBOARD CATEGORIES AUTHORS BOOKS ISSUE BOOKS REG STUDENTS CHANGE PASSWORD

ADD AUTHOR

Author Info

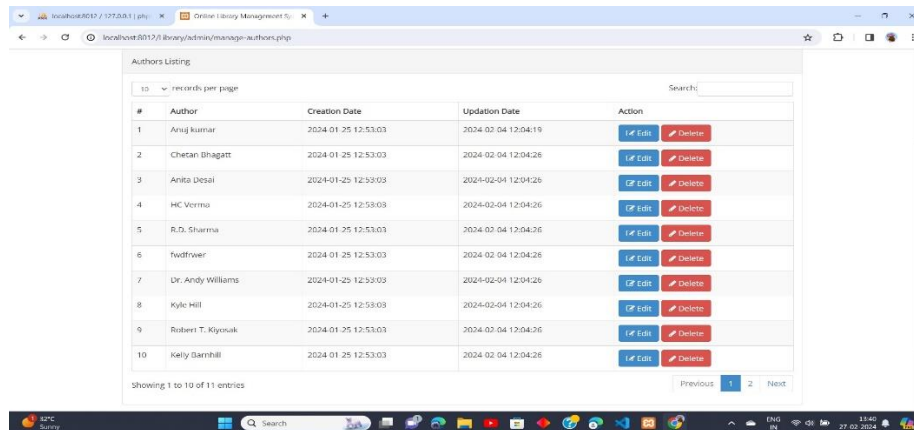
Author Name

Add

© 2024 Online Library Management System | Designed by : PIRGURUKUL

Figure 1.21 Add Author page

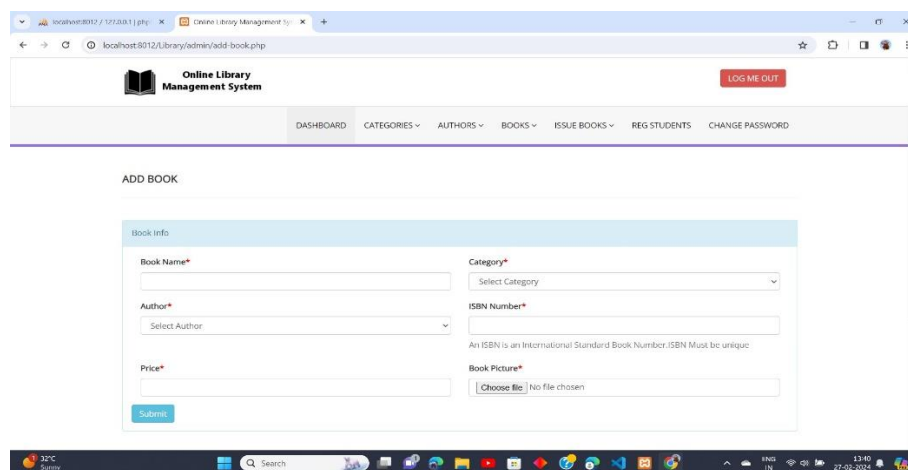
6.13 MANAGE AUTHOR PAGE



#	Author	Creation Date	Update Date	Action
1	Anuj kumar	2024-01-25 12:53:03	2024-02-04 12:04:19	Edit Delete
2	Chetan Bhagatt	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete
3	Anita Desai	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete
4	HC Verma	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete
5	R.D. Sharma	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete
6	Reddfover	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete
7	Dr. Andy Williams	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete
8	Kyle Hill	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete
9	Robert T. Kiyosaki	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete
10	Kelly Garnhill	2024-01-25 12:53:03	2024-02-04 12:04:26	Edit Delete

Figure 1.22 Manage Author page

6.14 ADD BOOK PAGE



Online Library Management System

DASHBOARD CATEGORIES AUTHORS BOOKS ISSUE BOOKS REG STUDENTS CHANGE PASSWORD

ADD BOOK

Book Info

Book Name*

Category*

Author*

ISBN Number*

Price*

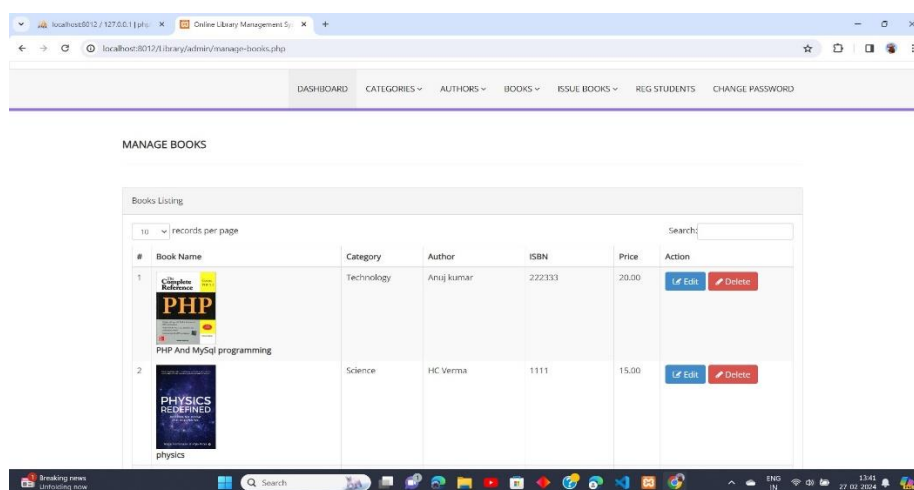
Book Picture*

Submit

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

Figure 1.23 Add Book page

6.15 MANAGE BOOK PAGE





#	Book Name	Category	Author	ISBN	Price	Action
1	 PHP And MySQL programming	Technology	Anuj kumar	222133	20.00	Edit Delete
2	 PHYSICS REDEFINED	Science	HC Verma	1111	15.00	Edit Delete

Figure 1.24 Manage Book page

6.16 ISSUE A NEW BOOK PAGE

Online Library Management System

DASHBOARD CATEGORIES AUTHORS BOOKS ISSUE BOOKS REG STUDENTS CHANGE PASSWORD

ISSUE A NEW BOOK

Issue a New Book

Student id*

ISBN Number or Book Title*

Issue Book

Figure 1.25 Issue a new Book page

6.17 MANAGE ISSUED BOOKS PAGE

Online Library Management System

DASHBOARD CATEGORIES AUTHORS BOOKS ISSUE BOOKS REG STUDENTS CHANGE PASSWORD

MANAGE ISSUED BOOKS

Issued Books

10 records per page Search

#	Student Name	Book Name	ISBN	Issued Date	Return Date	Action
1	John Doe	C++: The Complete Reference, 4th Edition	007053246X	2024-02-01 11:15:57	2024-02-04 12:03:08	Edit
2	John Doe	PHP And MySQL programming	222333	2024-02-01 11:15:57	Not Return Yet	Edit
3	test	ASP.NET Core 5 for Beginners	00536344563	2024-02-01 11:15:57	2024-02-04 12:03:08	Edit
4	test	C++: The Complete Reference, 4th Edition	007053246X	2024-02-01 11:15:57	2024-02-04 12:03:08	Edit
5	Anuj kumar	PHP And MySQL programming	222333	2024-02-01 11:15:57	2024-02-04 12:03:08	Edit
6	Sarita Pandey	Murach's MySQL	9350237695	2024-02-01 11:15:57	Not Return Yet	Edit

Showing 1 to 6 of 6 entries Previous 1 Next

Figure 1.26 Manage Issued Book page

6.18 MANAGE REG STUDENTS PAGE

Online Library Management System

DASHBOARD CATEGORIES AUTHORS BOOKS ISSUE BOOKS REG STUDENTS CHANGE PASSWORD

MANAGE REG STUDENTS

Reg Students

10 records per page Search

#	Student ID	Student Name	Email id	Mobile Number	Reg Date	Status	Action
1	SD002	Anuj kumar	anuj@gmail.com	9865472555	2024-01-31 12:53:03	Active	Inactive Details
2	SD005	sdhd	csfhd@dfhds.com	8560710025	2024-01-31 12:53:03	Active	Inactive Details
3	SD009	test	test@gmail.com	2398874527	2024-01-31 12:53:03	Active	Inactive Details
4	SD010	Amit	amit@gmail.com	8589856224	2024-01-31 12:53:03	Active	Inactive Details
5	SD011	Sarita Pandey	sarita@gmail.com	467243754	2024-01-31 12:53:03	Active	Inactive Details
6	SD012	John Doe	john@tes.com	1234569870	2024-01-31 12:53:03	Active	Inactive Details

Showing 1 to 6 of 6 entries Previous 1 Next

Figure 1.27 Manage reg student page

REFERENCES

- <https://www.w3schools.com/php/>
- <https://w3schools.com/MySQL/>
- <https://chat.openai.com/>
- <https://stackoverflow.com/questions/15930883/connecting-to-mysql-using-apache-and-php>
- <https://goo.gl/K8mszB>