DATABASE MANAGEMENT SYSTEMS

PROJECT REPORT

LIBRARY MANAGEMENT SYSTEM

Contents

ABSTRACT

A college library management is a project that manages and stores books information electronically according to student's 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.

So, in this project we will see that how performing different tasks on database becomes easy when we program it and link it by GUI. Different task like insertions/deletion/update of books and insertions/deletion/update of student details are include. Also issuing and returning operations are implemented in a very practical manner. Like before issuing a book, program will first check whether book is available or not. Also, after every issue/return operation it will update the no. of copies in the database.

This project is implemented in html, CSS and php. Software like visual studio code and Xampp is used.

ABOUT THE SOFTWARES USED

Visual studio code

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, HTML and C++. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Xampp

XAMPP is a software distribution which provides the Apache web server, MySQL database (actually MariaDB), Php and Perl (as command-line executables and Apache modules) all in one package. It is available for Windows, MAC and Linux systems. No configuration is necessary to integrate Php with MySQL.

It is a great fit for this course and provides a relatively painless installation and way to manage the configuration changes. Also provided is PhpMyAdmin which gives a GUI tool for managing your MySQL databases.

In this project, I am using HTML, CSS and PHP only. I have used visual studio code for the coding implementation of the project whereas Xampp is used for php module and creating database.

ABOUT THE LANGUAGES USED

HTML stands for Hypertext Mark-up Language. It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications.

HTML is not a programming language, meaning it doesn't have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word. When working with HTML, we use simple code structures (tags and attributes) to mark up a website page.

HTML documents are files that end with a .html or .htm extension. You can view then using any web browser (such as Google Chrome, Safari, or Mozilla Firefox). The browser reads the HTML file and renders its content so that internet users can view it.

Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

CSS saves time − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

Pages load faster − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

Superior styles to HTML − CSS have a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

Multiple Device Compatibility − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

Global web standards − Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

PHP is a recursive acronym for "PHP: Hypertext Preprocessors". PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynami6c content, databases, session tracking, even build entire e-commerce sites. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

PHP is forgiving: PHP language tries to be as forgiving as possible. Its Syntax is C-Like.

LIBRARY MANAGEMENT SYSTEM

A college library management is a project that manages and stores books information electronically according to student's 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.
Thus this system reduces manual work to a great extent allows smooth flow of library activities by removing chances of errors in the details.

Admin login: Admin is the one who administers the system by adding or removing books/students into and from the system respectively.

User login: Students have to login into the system to access their account info. But to login, admin will have to register them into their system.

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 details of the book.

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

Issue/Return Option: Only admin have the authority to issue/return any book. Also, no books will be issued if student is having any fine or exceeded its max limit (i.e., 5).

Advantages

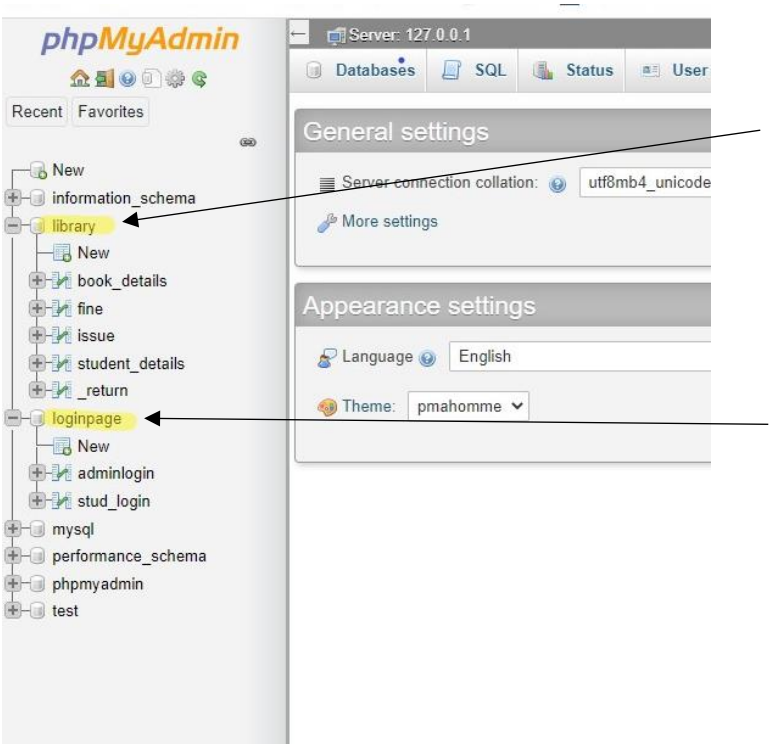
- The system excludes the use of paper work by managing all the book information electronically.
- Admin can keep updating the system by providing the new books arrival in system and their availability thus students need not to go to library for issuing purpose.
- The system has books well organized and systematically arranged in different categories in the system so that user can easily search and find the book.

Thus, it saves human efforts and resources.

Disadvantages

There is no human interaction if users have some enquiry.

DATABASE USED



DATABASE: library

Tables in database: book_details, fine, issue, student_details, _return

DATABASE: loginpage

Tables in database: adminlogin, stud_login

DATABASE: loginpage

This database is used to store records of login details of both **admin** and **students**.

1. Table: adminlogin

adminname	password
admin	admin

Table name: adminlogin

This table contains two attributes, adminname and password. This table is used check the login details of admin at the of login into the system.

2. Table: stud_login

	stud_id	password
Edit Copy Delete	2K19/CO/001	admin
Edit Copy Delete	2K19/CO/002	admin
Edit Copy Delete	2K19/CO/003	admin
Edit Copy Delete	2K19/CO/004	admin
Edit Copy Delete	2K19/CO/005	admin

This table contains two attributes, stud_id and password. This table is used check the login details of student at the of login into the system.

DATABASE: library

This database contains 5 tables- book_details, fine, issue, student_details and _return.

1. ***book_details***

	book_id	book_title	auther	copies	_status
Edit Copy Delete	10001	Renewables — A review of sustainable energy supply...	"Ferry, D K"	0	NAVL
Edit Copy Delete	10002	Semiconductors — Bonds and bands	"Tománek, D"	5	AVL
Edit Copy Delete	10003	Guide Through the Nanocarbon Jungle — Buckyballs	Williams, J H	5	AVL
Edit Copy Delete	10004	Defining and Measuring Nature — The make of all th...	"Fujimoto, M"	5	AVL
Edit Copy Delete	10005	Introduction to the Mathematical Physics of Nonlin...	"Kent, B R"	4	AVL
Edit Copy Delete	10006	3D Scientific Visualization with Blender®	"Zheng, L"	5	AVL
Edit Copy Delete	10007	Advanced Tokamak Stability Theory	"Lakshmanan, S"	4	AVL

This table contains 5 attributes:

- **Book_id**- It contains the book id of each book belonging to same category.
- **Book_title**- It contains name of each book.
- **Auther**- It contains the name of author of book.
- **Copies**- It contains the no. of copies currently present in the library.
- **_status**- it states whether the book is available in library or not.

2. ***student_details***

	stud_id	stud_name	stud_email
Edit Copy Delete			
Edit Copy Delete	2K19/CO/001	AADHEESH SHARMA	aadheeshsharma_2k19co001@dtu.ac.in
Edit Copy Delete	2K19/CO/002	AADI RAWAT	aadirawat_2k19co002@dtu.ac.in
Edit Copy Delete	2K19/CO/003	AADITYA NARAYAN SUBEDY	aadityanarayansubedy_2k19co003@dtu.ac.in
Edit Copy Delete	2K19/CO/004	AAKASH SHARMA	aakashsharma_2k19co004@dtu.ac.in
Edit Copy Delete	2K19/CO/005	AANCHAL SAHU	aanchalsahu_2k19co005@dtu.ac.in
Edit Copy Delete	2K19/CO/006	AARADHYA BERI	aaradhyaberi_2k19co006@dtu.ac.in

This table contains 3 attributes:

- **Stud_id**- it contains the id no. of each student.
- **Stud_name**- it contains names of students.
- **Stud_email**- it contains email id of students.

3. ***Issue***

stud_id	book_id	date	-status
2K19/CO/020	10003	2021-03-22 16:15:13	returned
2K19/CO/020	10005	2021-03-22 16:16:10	returned
2K19/CO/020	10006	2021-03-22 16:17:11	returned
2K19/CO/021	10020	2021-03-22 17:07:58	issued
2K19/CO/020	10022	2021-03-26 21:27:12	issued

This table contains 4 attributes:

- **Stud_id**- it contains the id no. of each student.
- **Book_id-** it contains id of the issued book
- **Date** - it contains date of the time of issued book.
- **Status**- it contains the status of the book that whether its issued or returned.

4. ***Return***

stud_id	book_id	date
2K19/CO/020	10003	2021-03-26 21:25:33
2K19/CO/020	10005	2021-03-26 21:26:34
2K19/CO/020	10006	2021-03-26 21:37:48
2K19/CO/023	10004	2021-03-26 21:38:25
2K19/CO/074	10022	2021-04-01 18:37:13
2K19/CO/021	10060	2021-04-01 18:41:37

This table contains 3 attributes:

- **Stud_id**- it contains the id no. of each student.
- **Book_id-** it contains id of the issued book
- **Date** - it contains date of the time of book return.

5. ***Fine***

	stud_id	fine
Edit Copy Delete	2K19/CO/020	30
Edit Copy Delete	2K19/CO/075	5

This table contains 2 attributes:

- **Stud_id**- it contains the id no. of each student.
- **Fine**- fine on the student.

Connection from database

Since I have used two databases, so I have used two different connections file to connect them.

- Connection to database **loginpage**

```
<?php
$conn = "";
try {
    $servername = "localhost:3306";
    $dbname = "loginpage";
    $username = "root";
    $password = "";

    $conn = new PDO(
        "mysql:host=$servername; dbname=loginpage",
        $username, $password
    );
$conn->setAttribute(PDO::ATTR_ERRMODE,
                    PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>
```

- Connection to database **library**

```
<?php
    $servername = "localhost:3306";
    $dbname = "library";
    $username = "root";
    $password = "";

    $conn = mysqli_connect($servername, $username, $password, $dbname);

    if(!$conn){
        die("connection to this database failed due to".mysqli_connect_error()
);
    }
?>
```

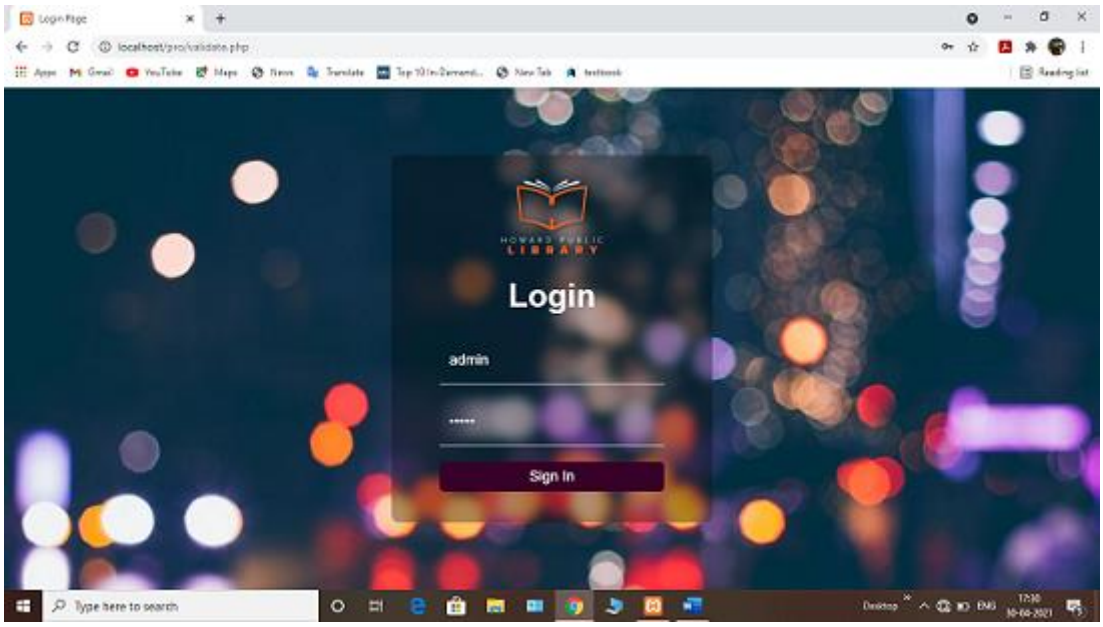
Using the about pieces of code (in php), GUI (webpage) will get connected to the database I have created.

GUI AND WORKING

ADMIN SIDE

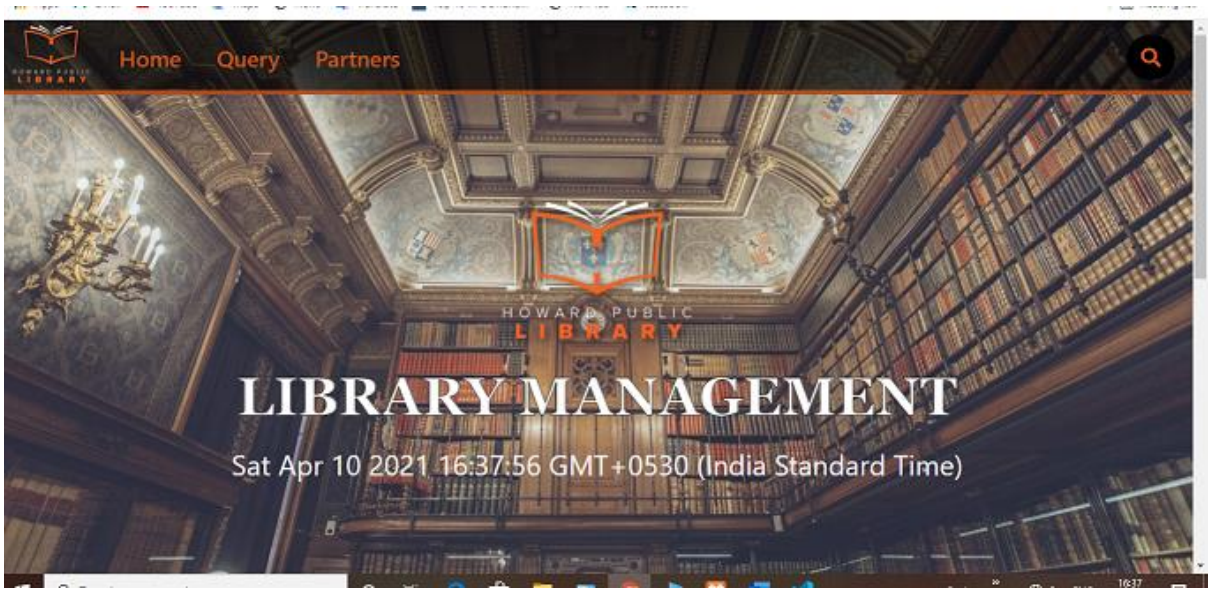
1. Admin login page

   To access the operations of admin, anyone will have to login in to system. To login into the system one will have enter admin user id and password as shown below:



Admin login page

These login details will get verified from table adminlogin which is mentioned in the previous section and if verified successfully, main admin page will be displayed else an error will appear. For the simplicity I have added only one user details.



Main page

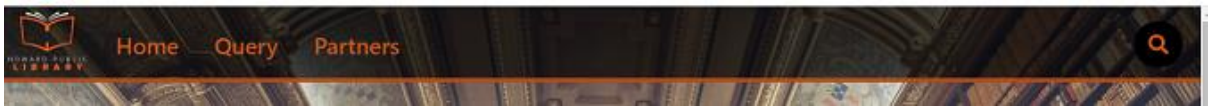
After login into the system, admin will come to the main page of the system we have created.

2. Mainpage

2.1. Navigation bar

First thing we see on the page is the navigation bar. Here, we can see 4 options as mentioned below:

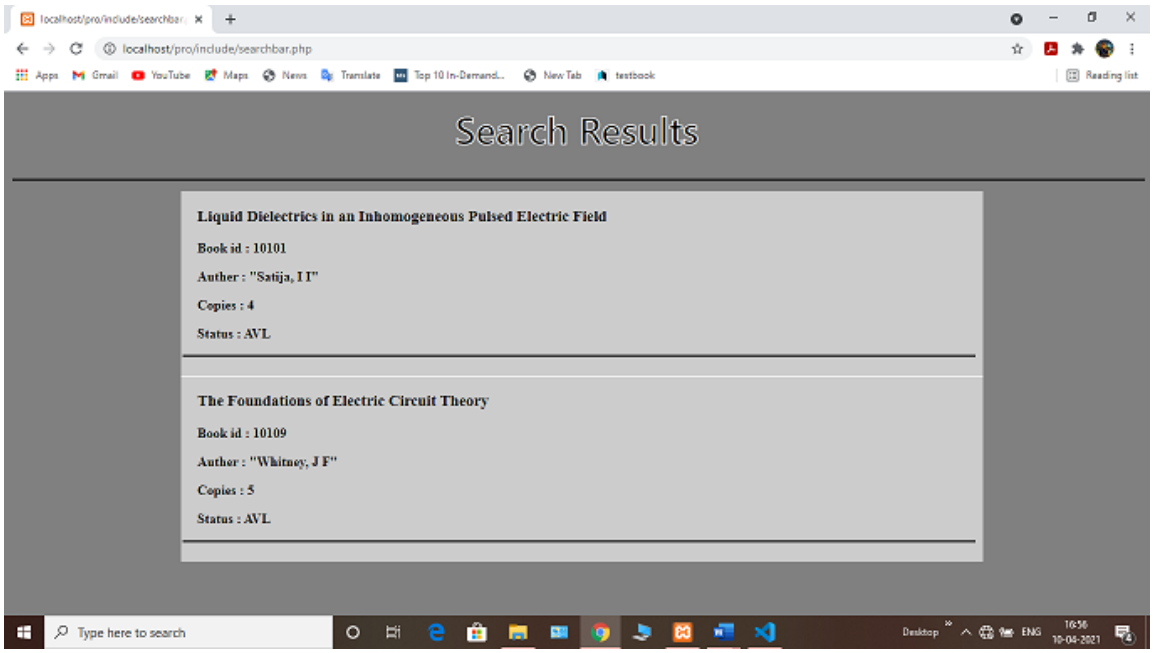
- **Home**- click on it and you will come to the top of the page.
- **Query**- click on it and you will come to operation sections.
- **Partners**- click on it and you will come to the library's partner sections. Its just for designing purpose.
- **Searchbar**- click on it and type something the about the book you want to search.



About the search bar:

In the search bar we can type something related to the book and in result we will see the info about the book/books we want. For example, if we type '*electric*' in the search bar we would be able to see desired results.

Typing result in search bar



Books having 'electric' in their name gets display

Similarly, if we to search any book by the name of its author, title or book id; we can find the results and access any info about the book.

Query statement executed:

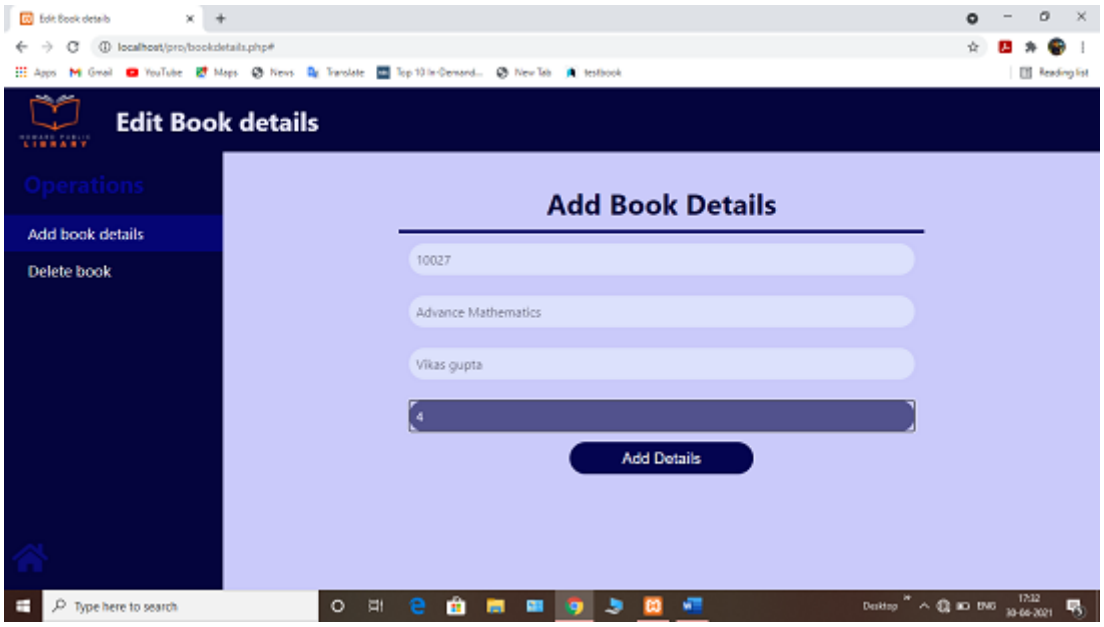
```
$sql = "SELECT * FROM `book_details` WHERE book_title LIKE '%$srch%' OR auther LIKE '%$srch%' OR bo
ok_id='$srch'";
```

2.2. Query section

In this section we would be able to perform different operations like edit book details in which we would be able to add details about the books or even can delete it; edit students details I which we would able to add, update, and delete info about the students; and at last issue/return books.

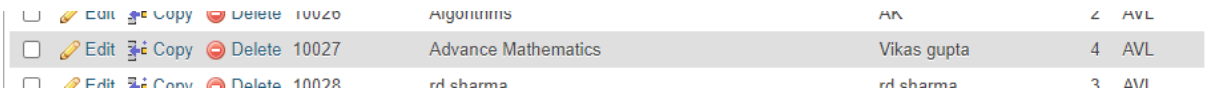
2.2.1. Edit book details

CLICK on this and you will move to **bookdetails** page.



Book with book id '10027' is added

In add book details enter bookid, title, author's name and no. of copies of the book and click on add details. This operation will add that info in table book_details which is shown below.



Query statement executed:

```
$sql = "INSERT INTO `book_details`(`book_id`, `book_title`, `auther`, `copies`
, `_status`) VALUES ('$book_id','$book_title','$auther','$copies','AVL')";
```

Delete book

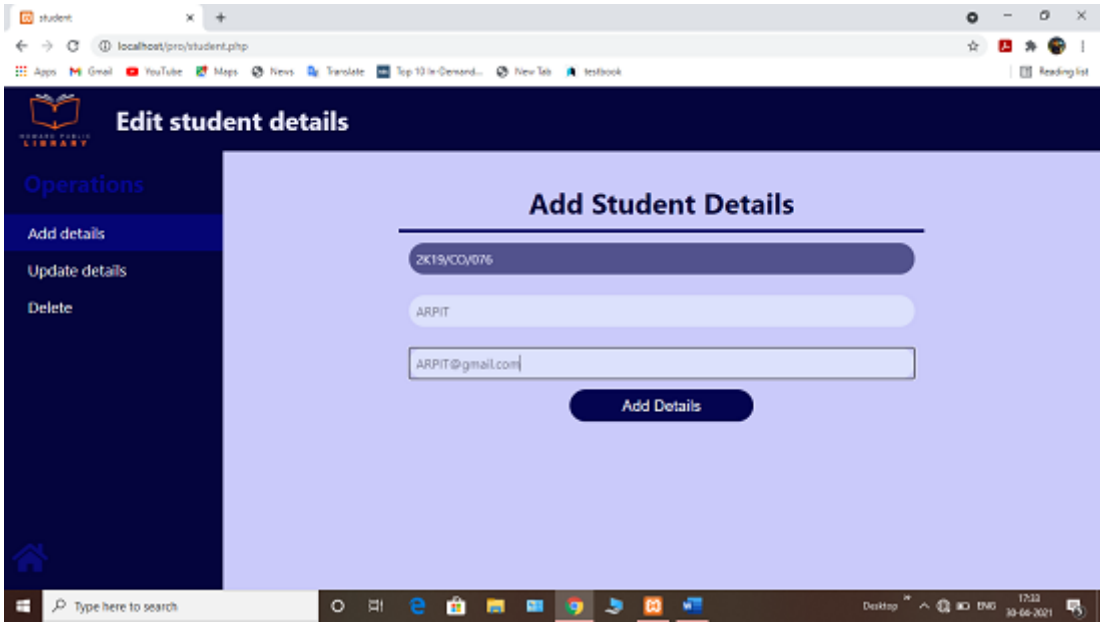
For deleting the book just type the book id and it will delete that book from database.

Query statement executed:

```
$sql = "DELETE FROM `book_details` WHERE book_id='$delete_id'";
```

2.2.2. Edit students' details

CLICK on Edit student Details and we will move to **student** page.



Details are added of student having id 2K19/CO076

Here you will see 3 options ADD, UPDATE and DELETE.

ADD: In this operation admin have to enter details about the student and this info is stored in table student_details. Also, one more entry is added into stud_login table having attributes stud_id and password (= 'admin').

Query statement executed:

1. $sql = "INSERT INTO student_details (stud_id, stud_name, stud_email) VALUES ('$stud_id','$stud_name','$stud_email')";
2. $stmt = $con->prepare("INSERT INTO stud_login VALUES('$stud_id','admin')");

UPDATE: In this operation admin have to enter details about the student and this info (id can't be updated) is updated in table student_details.

Query statement executed: $sql5 = "UPDATE `student_details` SET `stud_id`='$u_stud_id' ,`stud_name`='$u_stud_name' ,`stud_email`='$u_stud_email' WHERE stud_id = '$u_stud_id'";

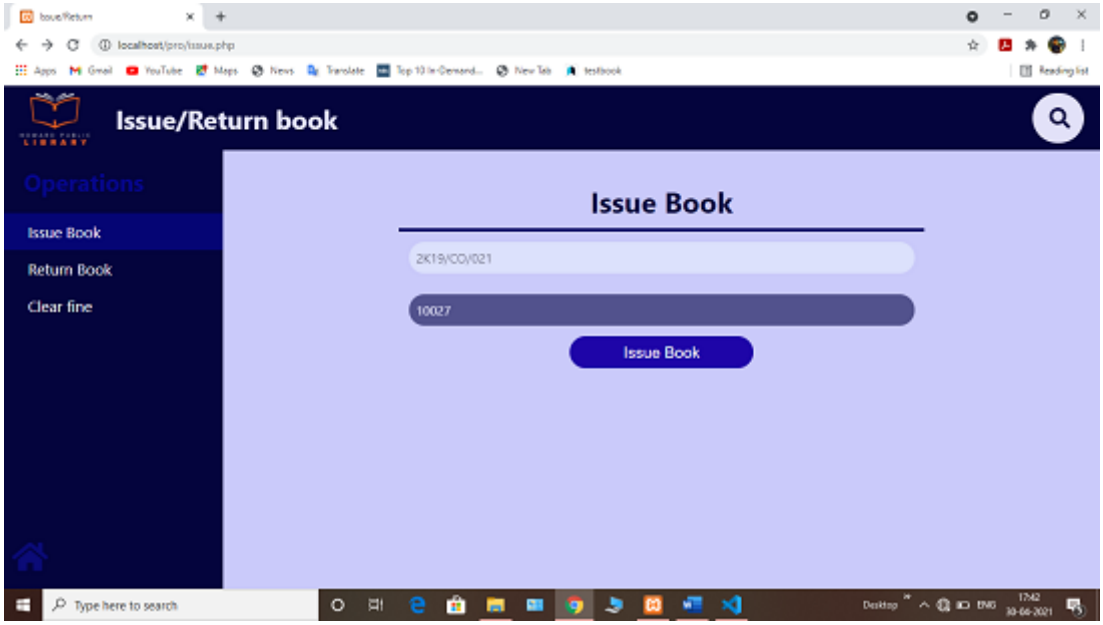
Remove student: In this operation admin have to enter student Id and this info will get deleted from table student_details. Also, one more entry is delete from stud_login table having stud_id = enter info.

Query statement executed:

1. $sql = "DELETE FROM `student_details` WHERE stud_id='$delete_stud_id'";
2. $stmt = $con->prepare("DELETE FROM stud_login WHERE stud_id='$delete_stud_id'");

2.2.3. Issue/return book

CLICK on Issue/return book and you will come to **issue** page.



Here you will see 3 options ISSUE, RETURN and CLEAR FINE.

ISSUE- In this operation we have to enter student id and book id. Doing so will add details in table issue. But for issuing a book, I have imposed certain constraints mentioned below.

Book will not be issued if:

- Fine on the student is not cleared.
- Student have exceeded max limit of books (=5).
- Student is not registered in table student_details.

Query statement executed:

$stmt = "SELECT * FROM `book_details` WHERE book_id = '$book_issue_id'";

$results = mysqli_query($conn,$stmt);

$stmt2 = "SELECT * FROM `issue` WHERE stud_id = '$stud_issue_id' AND `-status`='issued' ;";

```
$r = mysqli_query($conn,$stmt2);
$num = mysqli_num_rows($r);
$copies = "";
foreach($results as $result) {
  if(($result['_status'] == 'AVL') && ($num < 5) && ((int)$result['copies'] > 0)) {
    $check= true;
    $copies = (int)$result['copies'];
  } }
$verify = false;
$stmt3 = "SELECT * FROM `student_details` WHERE stud_id = '$stud_issue_id';";
$r2 = mysqli_query($conn,$stmt3);
foreach($r2 as $st) {
 if(($st['stud_id'] ==$stud_issue_id )) {
   $verify= true;  } }
$check2 = true;
$stmt4= "SELECT * FROM `fine` WHERE stud_id = '$stud_issue_id'";
$r3 = mysqli_query($conn,$stmt4);
$n2 = mysqli_num_rows($r3);
if($n2!=0){
 foreach($r3 as $s) {
   if(($s['fine']!=0)) {
     $check2 = false;   } }}
 if(($check==true)&&($verify == true)&&($check2==true)){
   $copies = $copies - 1;
   if($copies>0){
     $sql  = "UPDATE `book_details` SET `copies` = '$copies' WHERE book_id = '$book_issue_id';";
     mysqli_query($conn, $sql);
     $sql2 = "INSERT INTO `issue` (`stud_id`, `book_id`, `date`, `-status`) VALUES ('$stud_issue_id', '$book_issue_id',
current_timestamp(), 'issued');";
     mysqli_query($conn, $sql2); }
   else{
     $sql  = "UPDATE `book_details` SET `copies` = '$copies' WHERE book_id = '$book_issue_id';";
     mysqli_query($conn, $sql);
     $sql1  = "UPDATE `book_details` SET `_status` = 'NAVL' WHERE book_id = '$book_issue_id';";
     mysqli_query($conn, $sql1);
     $sql2 = "INSERT INTO `issue` (`stud_id`, `book_id`, `date`, `-status`) VALUES ('$stud_issue_id', '$book_issue_id',
current_timestamp(), 'issued');";
     mysqli_query($conn, $sql2);
```

```
  }
```

RETURN- In this operation we have to enter student id and book id. Doing so will update the details in table issue as "-status = 'returned'".

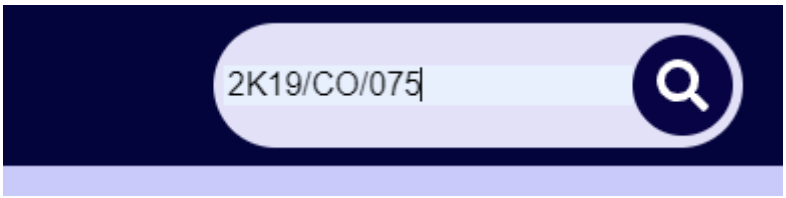
Query statement executed:

```
$stmt = "SELECT * FROM `issue` WHERE stud_id = '$stud_return_id' AND book_id = '$book_return_id';";
 $results = mysqli_query($conn,$stmt);
foreach($results as $result) {
  if(($result['stud_id'] ==  $stud_return_id) && ($result['book_id'] ==  $book_return_id)) {
    $check= true;  } }
 if($check==true){
  $stmt1 = "SELECT * FROM `book_details` WHERE book_id = '$book_return_id'";
  $rt = mysqli_query($conn,$stmt1);
  $n="";
  foreach($rt as $r){
   $n= (int)$r['copies'];}
  if($n == 0){
   $n = $n + 1;
   $sql = "UPDATE `issue` SET `-status`= 'returned' WHERE `book_id` = '$book_return_id' AND `stud_id` =
'$stud_return_id'";
   mysqli_query($conn, $sql);
   $sql1  = "UPDATE `book_details` SET `_status` = 'AVL' WHERE book_id = '$book_return_id';";
   mysqli_query($conn, $sql1);
   $sql2  = "UPDATE `book_details` SET `copies` = '$n' WHERE book_id = '$book_return_id';";
   mysqli_query($conn, $sql2);
   $sql3 = "INSERT INTO `_return`(`stud_id`, `book_id`, `date`) VALUES
('$stud_return_id','$book_return_id',current_timestamp())";
   mysqli_query($conn, $sql3); }
  else {
   $n = $n + 1;
   $sql = "UPDATE `issue` SET `-status`= 'returned' WHERE `book_id` = '$book_return_id' AND `stud_id` =
'$stud_return_id'";
   mysqli_query($conn, $sql);
   $sql2  = "UPDATE `book_details` SET `copies` = '$n' WHERE book_id = '$book_return_id';";
   mysqli_query($conn, $sql2);
   $sql3 = "INSERT INTO `_return`(`stud_id`, `book_id`, `date`) VALUES
('$stud_return_id','$book_return_id',current_timestamp())";
   mysqli_query($conn, $sql3);
  }
```

CLEAR FINE- In this operation we have to enter student. Doing so will delete the details in table fine.

Query statement executed: $sql="DELETE FROM `fine` WHERE `fine`.`stud_id` = '$stud_id_fine';";

SEARCH MENU AT TOP RIGHT:

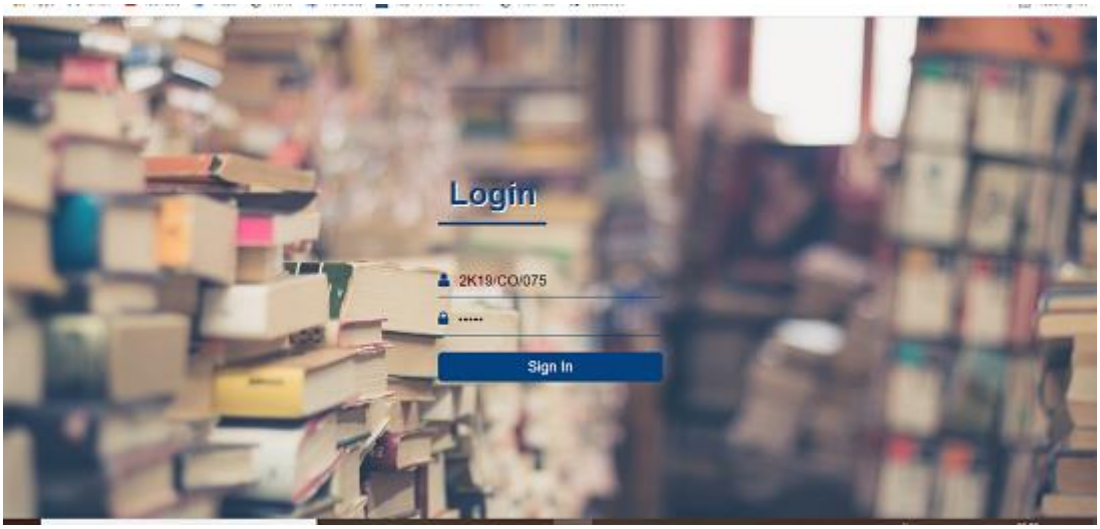


There is a search bar in top right of the page. Here we have to enter the student id and in result we would be able to see the details about students like name, roll no., email, info about the books issued, validity of books and also fine on the students. This feature becomes necessary to calculate fine on the students. How its done, we will see in the user side which is discussed latter in this project.

USER SIDE

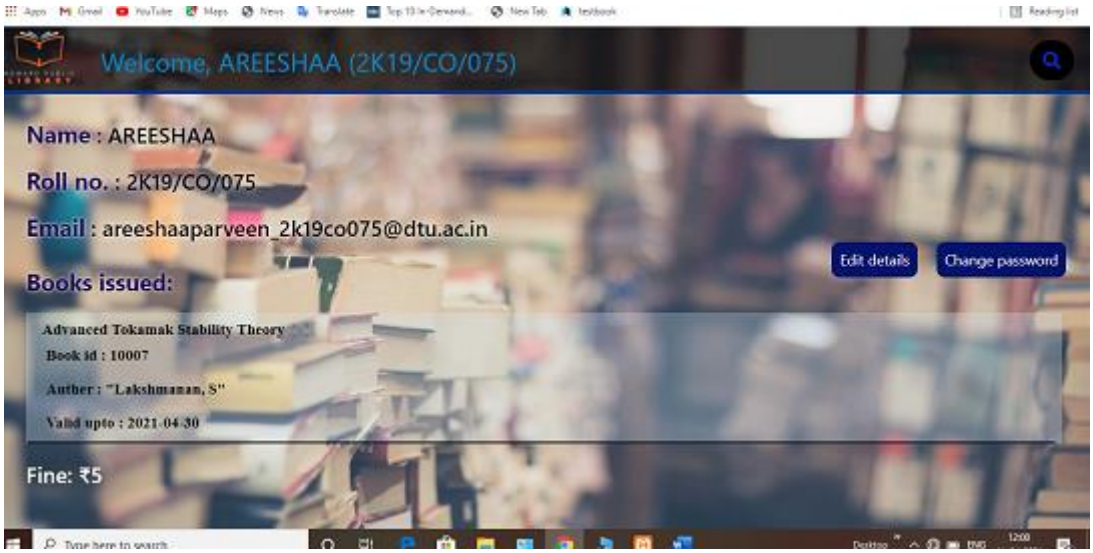
1. Student login page

   To access the user page, anyone will have to login in to system. To login into the system one will have enter user id and password as shown below:



Student login page

These login details will get verified from table stud_login which is mentioned in the previous section and if verified successfully, student page will be displayed else an error will appear.

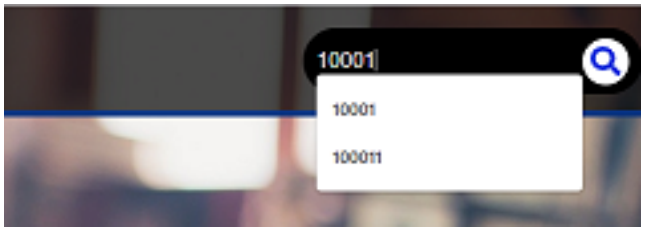


Student page

After login into the system, student will come to the student page of the system we have created.

2. Student page

   After coming to this page, at the top we can see student's name and roll no. and at right end we can see a search bar which works exactly same the admin main page search bar. For example, search for a book with book id '10001' is shown below:



Searching for book id '10001'

Result of search

As we move down in the page, we can see full details about the students (name, roll no., email) and details about the book issued if any. We can also see the validity of book issued which is 30 days from the issued date.

Query statement executed:

1. $sql = "SELECT * FROM `student_details` WHERE stud_id = '$stud_id'";
2. echo " Name : ".$name." ";
   echo " Roll no. : ".$stud_id." ";
   echo " Email : ".$email." ";
3. $stmt = "SELECT * FROM `issue` WHERE `stud_id` = '$stud_id' AND `-status`='issued';";
4. $currdate = date('Y-m-d h:m:s',time());
   $date= strftime("%Y-%m-%d", strtotime("$rlt2 +30 day"));
   if(strtotime($currdate)>strtotime($date)){
   $date = "Expired";}

Calculation of fine

For calculating fine, here are some operations done.

1. Step one will select all the entries from table issue where stud_id ='$stud_id'.
   $sql2 = "SELECT * FROM `issue` WHERE stud_id ='$stud_id'";
   $results = mysqli_query($conn,$sql2);

2. Step two will check each book with status = issued and compare its issued date with the present date. If the difference is more then 30 days, fine will be imposed at Rs.7/week.
   foreach ($results as $result) {

```
if($result['-status']=='issued'){
  $time = (int)strtotime($date) - (int)strtotime($result['date']);
  $days = $time/(86400);
  if($days>30){
    $days = $days - 30;
    $fine += ($days/7)*5; } }}
```

3. In step 3, it will 1st chech that whether the fine calculated = 0 or not. If fine = 0, then it will check from 'fine' table that whether the student is having any fine in present time or not. If yes, then the variable fine will be updated to the fine from the table.

```
if($fine==0){
    $stmt3 = "SELECT * FROM fine WHERE stud_id = '$stud_id';";
    $result1 = mysqli_query($conn, $stmt3);
    $num2 = mysqli_num_rows($result1);
    if($num2!=0){
      foreach ($result1 as $r){
        $fine=$r['fine']; }}}
```

If the calculated fine = 0, then it will check from 'fine' table that whether the student is having any fine in present time or not. If no, then it will simplily insert the values in table fine. But if yes, it will update the attribute 'fine' in table fine with calculated value of fine.

```
if($fine !=0){
    $stmt2 = "SELECT * FROM fine WHERE stud_id = '$stud_id';";
    $result = mysqli_query($conn, $stmt2);
    $num = mysqli_num_rows($result);
    if($num > 0){
    $sql3 = "UPDATE `fine` SET `fine`='$fine' WHERE stud_id = '$stud_id';";
    mysqli_query($conn, $sql3);}
    else{
     $sql3 = "INSERT INTO `fine`(`stud_id`, `fine`) VALUES ('$stud_id','$fine');";
     mysqli_query($conn, $sql3);}
  }
```

Also, this page has some additional features like edit details and change password.

EDIT DETAILS: CLICK on edit details and a page will appear. Enter roll no. (can't update), name and email you want to update. Example is shown below:

Above made changes gets reflect on student page

Query statement executed: $sql = "UPDATE `student_details` SET `stud_name` = '$name',`stud_email`='$email' WHERE `student_details`.`stud_id` = '$stud_id';";

CHANGE PASSWORD: CLICK on change password and a page will appear. Enter roll no., current password and new one and it will get updated in stud_login table. Example is shown below:

Change password page

Query statement executed:

$sql = $conn->prepare("UPDATE `stud_login` SET `password`='$newpassword' WHERE stud_id='$adminname';"); $sql->execute();

Conclusion

In the above discussed report, we have achieved our goal for making a library management system. Different task has been performed successfully keeping all practical situations under consideration. This system is very easy to use and requires no manual work in implementing the queries. Also, changes in the design or addition of some other operations can be on client's demand.

Reference

Project idea:

- https://nevonprojects.com/e-library-project/#:~:text=A%20college%20library%20management%20is,search%20for%20the%20desired%20book.
- https://data-flair.training/blogs/library-management-system-python-project/