

Laboratory 1

Title of the Laboratory Exercise: Software Design

1. Introduction and Purpose of Experiment

Web application design.

2. Aim and Objectives

Aim

- To develop Functional and Non-functional Requirements, ER diagram.

Objectives

At the end of this lab, the student will be able to

- Model the information required for the given scenario using E-R diagrams
- Develop ER diagram, class diagram, interaction sequence diagram and algorithm/flowchart

3. Experimental Procedure

Students are given a set of instructions to be executed on the computer. The instructions should be edited and executed and documented by the student in the lab manual. They are expected to answer questions posed in section 5 based on their experiment.

4. Presentation of Results

Functional Requirements:

Actor: Student

FR1: The application should allow the student to register using University ID and basic details

FR2: The application should display the available books to student

FR3: The application should allow the user to reserve a book from available books.

FR4: The application should allow the user to borrow the book

FR5: The application should allow the user to return the book

FR6: The application should allow the user to search the book

FR7: The application should allow the user to logout.

Non-Functional Requirements:

- NFR1:** The system should be user-friendly. The UI should be simple enough for everyone to understand and get the relevant information without any special training
- NFR2:** The system should be accurate, to display only the books that are not reserved
- NFR3:** The system should be maintainable, adding new feature should be simple
- NFR4:** The system should be available for the duration when the library operates.

E-R Diagram

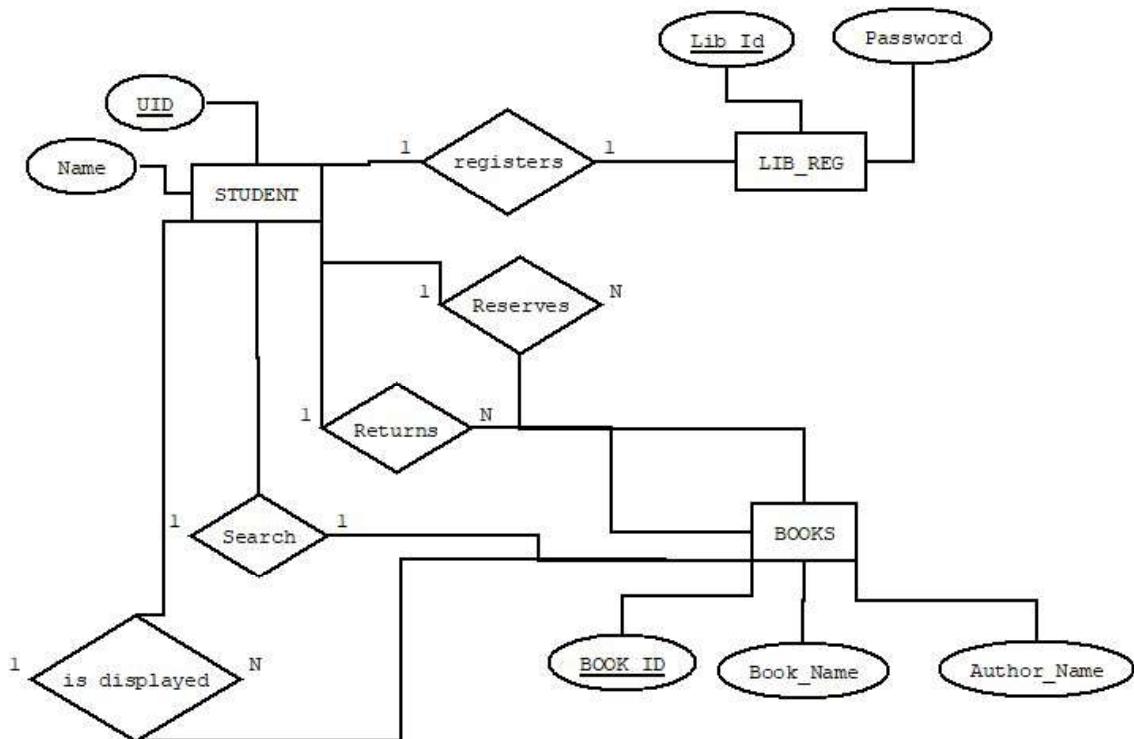


Figure 1: library Management System E-R diagram

E-R Diagram Analysis

- The E-R diagram identifies 3 entities in the system {STUDENT, LIB_REG, BOOKS}
- The STUDENT entity is characterized by two attributes UID and Name, UID is the primary key which refers to the university id of the student
- The BOOK entity is characterized by the attributes Book_ID, Book_Name, Author_Name where Book_ID is the primary key
- Any student in the university can register for library and each student can make only one registration hence the cardinality 1:1

- Any student who is registered with the library have access to the books in the library, where he can perform four actions, which is represented by the relationship in the ER diagram. They are display all books available, search, return, reserve.
- 1 students can reserve N books hence 1:N is the cardinality 1:N
- The LIB_REG entity characterizes the two attributes Lib_Id (Username) which is nothing but the primary key of the table, and Password, these two attributes are used to login by the user

5. Analysis and Discussions

Analysis

The Library Management System consists of the basic functionalities of the Library including the display of the available books, search of the books, reserving the books, returning the books. The application is from user end.

The student has registered with university and has USN number which is unique. And when the student wants to register library, then he uses the username, password and USN number, then for login, he uses only the username and password.

The functionalities of the book are carried out based on the book_Id provided for each book. Only the available books will be displayed in the display section.

The front-end is developed with HTML and CSS where as the back-end is developed using MySql ,Xampp, PhpMyAdmin where the connection for front-end and back-end is developed using PHP.

Discussion

The software has some drawbacks like:

1. The system prints the “Wrong ID or Password” if any one of the login ID credentials are wrong, without specifically prompting which among the two is wrong.
2. The system allows multiple trials for login for System admin, that leads to lower the security of a system.
3. Since, the system flows sequentially, it is only after the user (librarian) enters the book ID , the other actions come into picture

6. Conclusions

1. The requirements should be basic, complete, concrete, clear.
2. The functional requirement should be written in a sequence.
3. The functional requirements is written for each user.
4. Non functional requirements does not have any dependencies.
5. Non functional requirements is written for a software as a whole.

Laboratory 2

Database creation in PHPMyadmin

SQL Statements

```
CREATE TABLE `books` (
  `Book_Id` varchar(20) NOT NULL,
  `Book_Name` varchar(20) NOT NULL,
  `Author_Name` varchar(20) NOT NULL,
  `USN` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Creating table Books (which stores the book information and availability status and the USN of a person for whom the book is reserved to)

```
ALTER TABLE `books`
  ADD PRIMARY KEY (`Book_Id`),
  ADD KEY `USN` (`USN`);
```

Adding Primary to table Books

```
ALTER TABLE `books`
  ADD CONSTRAINT `USN` FOREIGN KEY (`USN`) REFERENCES `student` (`Stu_Id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Adding Foreign key to table Books 'USN'

```
CREATE TABLE `registration` (
  `Reg_Id` varchar(20) NOT NULL,
  `Password` varchar(20) NOT NULL,
  `USN_Id` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Creating table Registration (which stores login information)

```
ALTER TABLE `registration`
  ADD PRIMARY KEY (`Reg_Id`),
  ADD KEY `Reg_id_student` (`USN_Id`);
```

Adding Primary key to the Registration table (Reg_Id= Username)

```
ALTER TABLE `registration`
  ADD CONSTRAINT `Reg_id_student` FOREIGN KEY (`USN_Id`) REFERENCES `student` (`Stu_Id`) ON DELETE CASCADE ON UPDATE CASCADE;
  COMMIT;
```

Adding foreign key for the table registration ('USN_Id')

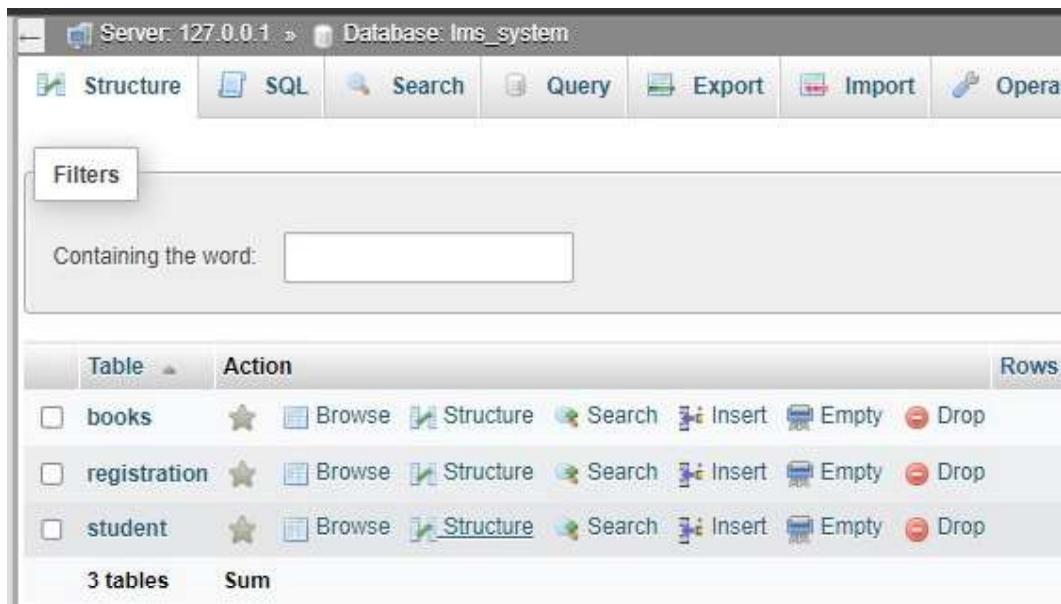
```
CREATE TABLE `student` (
  `Stu_Id` varchar(20) NOT NULL,
  `Stu_Name` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Creating table Student (To keep track of university students only and make sure that only university students register for the library)

```
ALTER TABLE `student`
ADD PRIMARY KEY (`Stu_Id`);
```

Making Stu_Id/USN_Id of the student as the primary key of the student table.

PHPMyadmin database



The screenshot shows the PHPMyAdmin interface with the following details:

- Server:** 127.0.0.1
- Database:** Ims_system
- Toolbar:** Structure, SQL, Search, Query, Export, Import, Opera
- Filters:** A search bar labeled "Containing the word:"
- Table List:** A table showing three tables: books, registration, and student. Each table has a checkbox, a star icon, and links for Browse, Structure, Search, Insert, Empty, and Drop.
- Summary:** 3 tables

Table	Action	Rows
books	Browse Structure Search Insert Empty Drop	
registration	Browse Structure Search Insert Empty Drop	
student	Browse Structure Search Insert Empty Drop	

Tables in the database "Ims_system"

Server: 127.0.0.1 » Database: lms_system » Table: books

Browse Structure SQL Search Insert Export Import

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	Book_Id	varchar(20)	utf8mb4_general_ci		No	None	
2	Book_Name	varchar(20)	utf8mb4_general_ci		No	None	
3	Author_Name	varchar(20)	utf8mb4_general_ci		No	None	
4	USN	varchar(20)	utf8mb4_general_ci		No	None	

Books table structure

Server: 127.0.0.1 » Database: lms_system » Table: registration

Browse Structure SQL Search Insert Export Import

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	Reg_Id	varchar(20)	utf8mb4_general_ci		No	None	
2	Password	varchar(20)	utf8mb4_general_ci		No	None	
3	USN_Id	varchar(20)	utf8mb4_general_ci		No	None	

Registration table structure

Server: 127.0.0.1 » Database: lms_system » Table: student

Browse Structure SQL Search Insert Export Import

Table structure Relation view

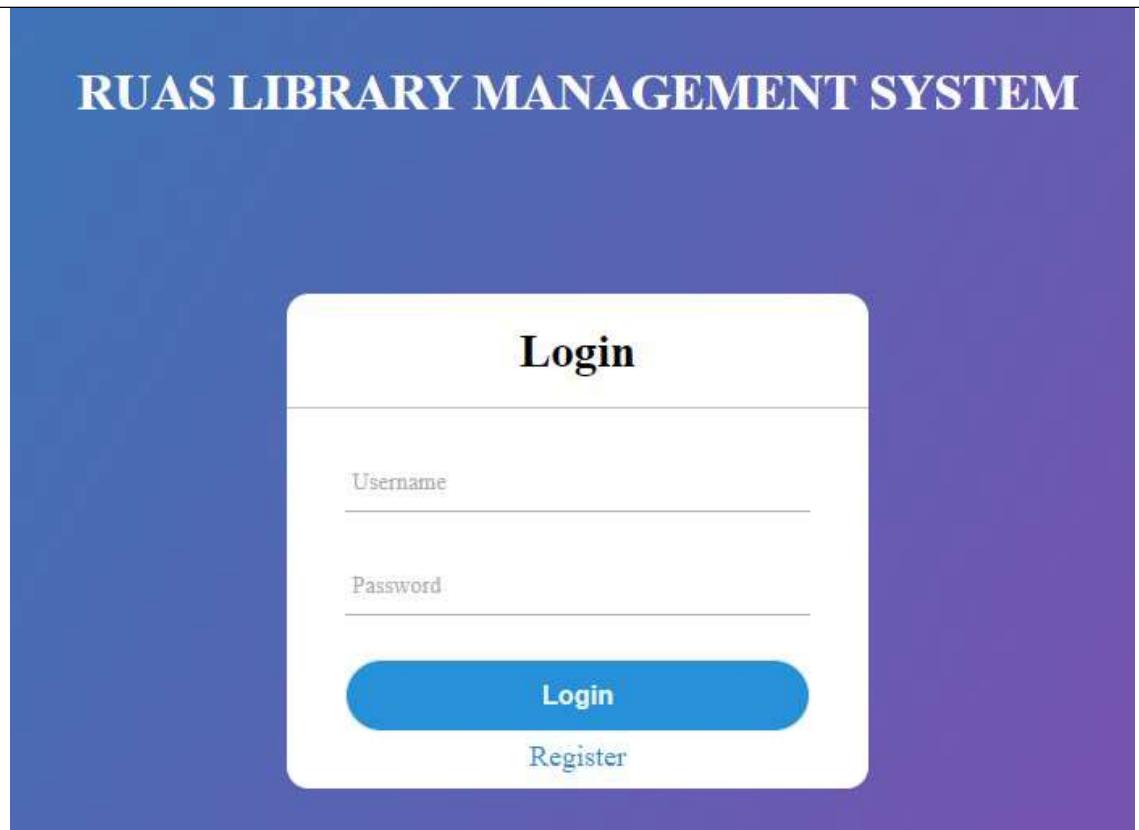
#	Name	Type	Collation	Attributes	Null	Default	Comments
1	Stu_Id	varchar(20)	utf8mb4_general_ci		No	None	
2	Stu_Name	varchar(20)	utf8mb4_general_ci		No	None	

Student Table Structure

Laboratory 3

Creating Library Management System with HTML, CSS and PHP

Login Page



HTML code

```
<div class="title" >
  <h1> RUAS LIBRARY MANAGEMENT SYSTEM </h1>
</div>
<div class="center"> <!-- it is a section of html which is the
  <h1> Login </h1>
  <form method="post">
    <div class="txt_field">
      <input type="text" required name='Username'> <!--
        <label> Username </label>
      </div>

      <div class="txt_field">
        <input type="password" required name='password'> <!--
          <span></span>
        <label> Password </label>
      </div>
      <input type="submit" value='Login' name='submit'> <!--
        <div class='sign_up'>
          <a href= 'signup.php'> Register </a>
        </div>
      </form>
</div>
```

The login page html code starts with the heading declaration of the class "title"

It defines the login component in the class center with the heading to prompt the user

There is one input of the type text which takes the username, and the other of the type password, to take the user password, and these two belongs to the same class "txt_field"

The other input of the type submit, prompt the

	<p>user to login and if login successful it loads home page</p> <p>It also has link for registration page</p>
CSS code	<pre> body{ margin:0; padding:0; font-family: montserrat; background: linear-gradient(120deg, #2980b9,#8e44ad); height: 100vh; /* checkout */ overflow: hidden; /* checkout */ } .center{ /*this is for entire login component */ position: absolute; top: 55%; left: 50%; transform: translate(-50%, -50%); width:400px; height: 340px; background: white; border-radius: 15px; } .title h1{ position: relative; padding: 50px 0; text-align: center; margin: 30px; color: white; font-size: 35px; } .center form{ padding: 0 40px; box-sizing: border-box; } form .txt_field{ position: relative; border-bottom: 2px solid #adadad; margin: 30px 0; } .txt_field input{ width: 100%; padding: 0 5px; height: 40px; font-size: 16px; border: none; background: none; outline: none; } .txt_field label{ position: absolute; top: 50%; left: 5px; color: #adadad; transform: translateY(-50%); font-size: 16px; pointer-events:none; } </pre> <p>The CSS code defines the entire body element with font-family: montserrat and background of blue with linear gradient 120deg. Any overflow in the content in the body is hidden.</p> <p>The “center” class defines the login component which has background white and its size is set using width and height properties, it is made center using top and left attributes, the corners are rounded using the property border-radius.</p> <p>The title heading is made relatively bigger with the font-size and the text is aligned center with the property text-align, the margin, space to the next element is 30px</p> <p>The form element which defines the input fields</p> <p>The inputs “txt and password” has no background defined to it. The height 40px and width is 100% to its parent class.</p> <p>The label for these inputs is styled with the color grey and made the position to appear above the input fields using the</p>

```

input[type='submit']{
  width:100%;
  height: 50px;
  border: 1px solid;
  background: #2691d9;
  border-radius: 25px;
  font-size: 18px;
  color: #e9f4fb;
  font-weight:700;
  cursor: pointer;
  outline: none;
}

input[type="submit"]:hover{
  border-color: #2691d9;
  transition:.5s;
}

.sign_up{
  margin: 5px 0;
  bottom: 5px;
  text-align: center;
  font-size: 20px;
  color: #666666;
}

.sign_up a{
  color: #2691d9;
  text-decoration:none;
}

```

properties top and left.

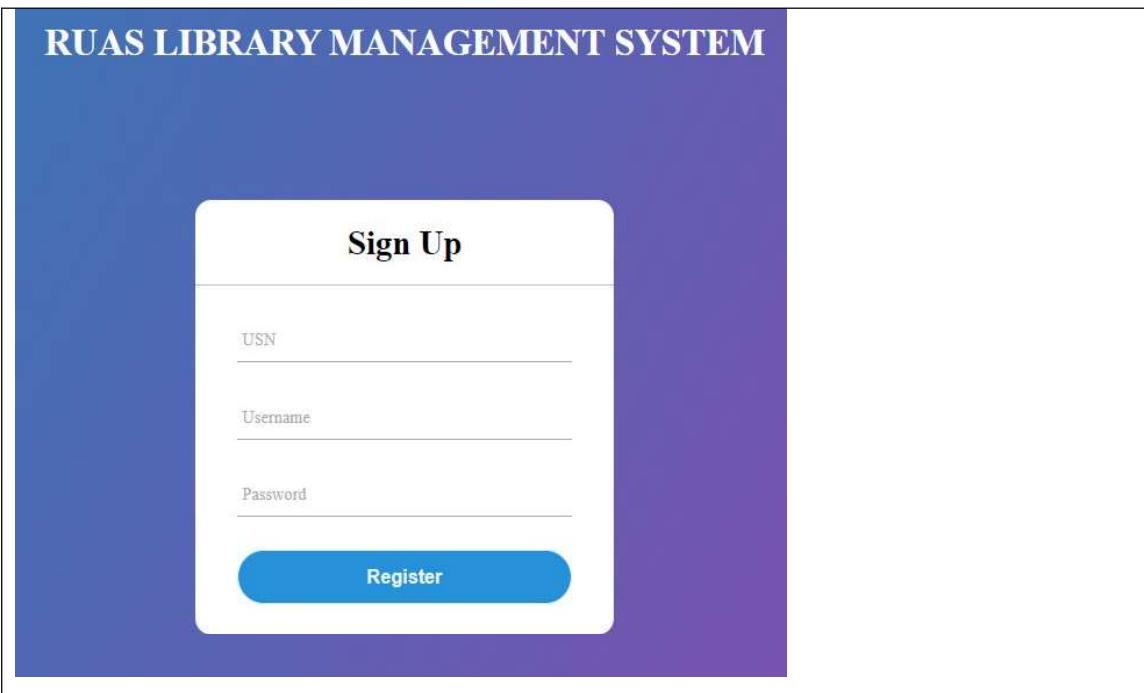
The type submit has a 1px solid border defined with the background dark blue, The border-radius defines the rounded corners. Font-weight defines the boldness of the text

The submit button on hover enlarges with the transition speed of 0.5seconds

The sign up link, has the same color to the theme and aligned center with text-align, margin which separates it from other components is defined to be 5px

<p>PHP code</p> <pre> if (isset(\$_POST['submit'])) { \$Reg_Id = \$_POST['Username']; \$password = \$_POST['password']; //get data from input and store it in local variable \$con = mysqli_connect("localhost", "root", "", "lms_system"); //where the application is running, now it is running in local server and //its domain name is called localhost if (mysqli_connect_errno()) { //if connection failed echo "Failed to connect to MySQL: " . mysqli_connect_error(); } \$query = "SELECT Reg_Id, Password FROM registration WHERE Reg_Id='\$Reg_Id' and Password='\$password'"; \$result = mysqli_query(\$con, \$query); //if there is a row with matching query if (\$result) { if (mysqli_num_rows(\$result) > 0) { //make sure there is atleast one row in a database //if atleast one entry present start session session_start(); \$_SESSION['Username'] = \$Reg_Id; header("Location: home.php"); } else { echo '<script>alert("Invalid Credentials");</script>'; } } else { echo '<script>alert("Invalid Credentials");</script>'; } } ?> </pre>	<p>The PHP code executes if the login/submit is clicked</p> <p>It takes the input of the username and password given by the user and store it in the php variable</p> <p>The code establishes the connection between the database with the required parameters</p> <p>Once the connection is established, the sql query select the username and password form the table registration and if it matches the username and password given as input by the user, it directs to the home page, else throws an error "invalid credentials"</p>
---	--

Registration Page



RUAS LIBRARY MANAGEMENT SYSTEM

Sign Up

USN

Username

Password

Register

HTML code <pre> <div class="center"> <!-- it is a section of html which is then used to start the registration page --> <h1> Sign Up </h1> <form method="post"> <div class="txt_field"> <input type="text" required name='USN'> <!-- it is a boolean as USN is a unique identifier --> <label> USN </label> </div> <div class="txt_field"> <input type="text" required name='Username'> <!-- it is a boolean as Username is a unique identifier --> <label> Username </label> </div> <div class="txt_field"> <input type="password" required name='password'> <!-- it is a boolean as password is a sensitive information --> <label> Password </label> </div> <input type="submit" value='Register'> <!-- login button --> </form> </div> </pre>	The HTML code for registration is same as that of login page, but the another txt_field is added to take the USN of the student, this is to make sure that all the students of the university only register for the library system.
CSS code	Is same as that of login page
PHP code <pre> \$Reg_Id = \$Password = ""; if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$Reg_Id = \$_POST['Username']; \$Password = \$_POST['password']; \$USN = \$_POST['USN']; \$con = mysqli_connect("localhost", "root", "", "lms_system"); //username: root, password: Yashaswini \$query1 = "SELECT Stu_Id FROM 'student' WHERE Stu_Id='USN'"; \$result1 = mysqli_query(\$con, \$query1); if (\$result1) { echo \$USN; \$query = "INSERT INTO 'registration' (Reg_Id, Password, USN_Id) VALUES ('\$Reg_Id', '\$Password', '\$USN')"; \$result = mysqli_query(\$con, \$query); if (\$result) { echo '<script>alert("Registration successful");</script>'; } else { echo '<script>alert("Registration failed");</script>'; } } } ?> </pre>	<p>If the submit button is clicked or, if the post request is made then the php code executes</p> <p>It stores the input of the user to the php variables and establishes the connection to the database</p> <p>Once the database connection is established, the sql query inserts the input into the registration table, which creates the new user with username, password.</p> <p>If the registration not successful, the error is thrown</p>

Home Page

RUAS LIBRARY MANAGEMENT SYSTEM

Reserve Book

Return book

Display Book

Search

HTML code

```
<div class="title" text-align="center">
  <h1> RUAS LIBRARY MANAGEMENT SYSTEM </h1>
  <br><br>
</div>

<div class="logout">
  <form align="right" name="form1" method="post" action="log_out.php">
    <label>
      <input name="submit2" type="submit" id="submit2" value="log out">
    </label>
  </form>
</div>

<div class="submit-container">
  <form action="reserve.php">
    <input type="submit" value="Reserve Book" class="submit-button">
  </form>
  <form action="return.php">
    <input type="submit" value="Return book" class="submit-button">
  </form>
  <form action="display.php">
    <input type="submit" value="Display Book" class="submit-button">
  </form>
  <form action="search.php">
    <input type="submit" name="searching" value="Search" class="submit-button">
  </form>
</div>
```

The home page, displays the heading. It also houses the logout button of the type submit and aligned right in the home page, the entire logout button is enclosed on logout button, when this button is clicked the action is going to load log_out.php which ends the session

The home page contains four submits for reserve, return, display and search and on each button clicked

	lands the corresponding page
CSS code	<pre> body{ background-image: url('dog.jpg'); margin:0; padding:0; font-family: montserrat; background: linear-gradient(120deg, #2980b9,#8e44ad); height: 100vh; /* checkout */ overflow: hidden; /* checkout */ } .title h1{ left: 15%; position: absolute; padding: 50px 0; text-align: center; margin: 30px; color: white; font-size: 40px; } .logout{ background-color: skyblue ; color: Black; /* set text color to white */ border: none; /* remove border */ padding: 15px 25px; /* add padding to button */ text-align: center; /* center text horizontally */ text-decoration: none; /* remove underline */ display: inline-block; /* make button display as font-size: 18px; /* set font size */ cursor: pointer; /* add cursor pointer on hover */ border-radius: 10px; /* add rounded corners */ float:top; float: right; } .submit-container { display: flex; flex-direction: column; align-items: center; justify-content: center; height: 100vh; /* set height to fill viewport */ } .submit-button { padding: 20px 40px; font-size: 40px; background-color: White; color: black; border: none; border-radius: 25px; margin: 15px 0; cursor: pointer; } </pre>

Reserve Book

<h1>RUAS LIBRARY MANAGEMENT SYSTEM</h1>	
<p>Enter Book ID</p> <p>Reserve</p>	
<p>HTML code</p> <pre><div class="form"> <form role="form" id="templatemo-preferences-form" name="reservation" action="" method="post"> <div class="center_container"> <input type="text" id="reserve" placeholder="Enter Book ID" name="bookid" required class="center" align="center" >

 <button type="submit" name="Reserve" value="Reserve" class="center" align="center" >Reserve</button> </div> </form> </div></pre>	<p>The HTML code for Reserving a Book has all the elements enclosed in the class “form” and form element defines the method post</p> <p>There are two inputs one of the type text to enter the bookid and the the is the type of submit input to submit the book id</p>
<p>CSS code</p>	<p>The entire body has the same theme to match with the website</p> <p>The heading is aligned center</p> <p>The form element which contains the input is aligned center</p> <p>The inputs are padded 2% to make</p>

```

body{
    font-family: montserrat;
    background: linear-gradient(120deg, #2980b9, #8e44ad);
    height: 100vh; /* checkout */
    overflow: hidden; /* checkout */
}

h1{
    text-align:center;
}

.form{
    clear:both;
    text-align:center;
}

.form form input{
    padding:2%;
    width:250px;
    border-radius: 25px;
    font-size:30px;
}

.form form button{
    width:100px;
    border-radius:25px;
    font-size:20px;
}

```

the inputs size bigger and are rounded with border-radius of 25px.

PHP code

```

if (isset($_POST['Reserve'])) {
    $bookid = $_POST['bookid'];
    $con = mysqli_connect("localhost", "root", "", "lms_system");
    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
    session_start();
    $user_id = $_SESSION['Username'];
    $sql1 = "select USN_Id from registration where Reg_Id='$user_id'";
    $result1 = mysqli_query($con, $sql1);
    if ($row = mysqli_fetch_assoc($result1)) {
        $column_value = $row['USN_Id'];
    }
    $sql2 = "select USN from books where book_id='$bookid'";
    $result2 = mysqli_query($con, $sql2);
    if ($row = mysqli_fetch_assoc($result2)) {
        $USN_Value = $row['USN'];
    }
    if ($USN_Value == '0') {
        $sql = "UPDATE books SET USN='$column_value' Where book_id = '$bookid' ";
        if (mysqli_query($con, $sql)) {
            // Record was updated successfully
            echo '<script>alert("Your book is reserved");</script>';
        } else {
            // Error updating the record
            echo '<script>alert("Error Updating Record");</script>';
        }
    } else {
        echo '<script>alert("This book is already reserved");</script>';
    }
}

```

When the reserve button is clicked, The code saves the book_id entered by the user to php variable.

The code establishes the connection to the database and the sql query checks whether the registered student is the university student and if so then the other query selects the book with book id entered by the user if it has the USN value as 0, the usn of the student will be entered in the books table, and if the book_id entered does not have usn=0 it displays the book is already reserved.

Return Book

RUAS LIBRARY MANAGEMENT SYSTEM

Enter Book ID

Return

Books you own are

book_id
BK101
BK102

HTML code

```

<div class="form">
  <form role="form" id="templatemo-preferences-form" name="reservation" action="" method="post">
    <div class="center">
      <input type="text" id="reserve" placeholder="Enter Book ID" name="bookid" required align="center"><br/>
      <br/>
      <button type="submit" name="Return" value="Return" align="center" >Return</button>
    </div>
  </form>
</div>

```

The HTML code to return the book houses all the elements in the class "form" and has two inputs one of the type text to input book_id and the other of type submit

CSS code

```

body{
  font-family: montserrat;
  background: linear-gradient(120deg, #2980b9, #8e44ad);
  height: 100vh; /* checkout */
  overflow: hidden; /* checkout */
}
h1{
  text-align:center;
}
.form{
  clear:both;
  text-align:center;
}
.form form input{
  padding:2%;
  width:250px;
  border-radius: 25px;
  font-size:30px;
}
.form form button{
  width:100px;
  border-radius:25px;
  font-size:20px;
}

```

The CSS code to return. The entire body has the same theme to match with the website

The heading is aligned center

The form element which contains the input is aligned center

The inputs are padded 2% to make the inputs size bigger and are rounded with border-radius of 25px.

PHP code

The php before hand initializes the

```

$con = mysqli_connect("localhost","root","","lms_system");
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
session_start();
$user_id=$_SESSION['Username'];
$sql1="select USN_Id from registration where Reg_Id='$user_id'";
$result1=mysqli_query($con,$sql1);
if ($row = mysqli_fetch_assoc($result1)) {
    $column_value = $row['USN_Id'];
}
$sql01="select Book_Id from books where USN='$column_value'";
$result01=mysqli_query($con,$sql01);
echo "<table style='width:50%' border='1'>
<caption> Books you own are </caption>
<tr>
<th>book_id</th>
</tr>";
while($row1=mysqli_fetch_assoc($result01)){
    echo "<tr>";
    echo "<td><center>".$row1["Book_Id"]."</center></td>";
    echo "</tr>";
    continue;
}

if (isset($_POST['Return'])) {
    $bookid = $_POST['bookid'];
    $con = mysqli_connect("localhost","root","","lms_system");
    if (mysqli_connect_errno())
    {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
    $user_id=$_SESSION['Username'];
    $sql1="select USN_Id from registration where Reg_Id='$user_id'";
    $result1=mysqli_query($con,$sql1);
    if ($row = mysqli_fetch_assoc($result1)) {
        $column_value = $row['USN_Id'];
    }
    $sql2="select USN from books where book_id= '$bookid'";
    $result2=mysqli_query($con,$sql2);
    if ($row = mysqli_fetch_assoc($result2)) {
        $value = $row['USN'];
    }
    if($value !='0'){
        $sql="UPDATE books SET USN='0' Where book_id= '$bookid' ";
        if (mysqli_query($con, $sql)) {
            // Record was updated successfully
            echo '<script>alert("Record Updated Successfully");</script>';
        } else {
            // Error updating the record
            echo '<script>alert("Error updating record");</script>';
        }
    }else{
        echo '<script>alert("You have not reserved this book to return");</script>';
    }
}
}

```

connection to the database and executes a query which retrieves the USN no of the user, using username of the session.

Note: Username is in registration table and USN is also in registration table.

Then the code selects the all the books from the book table where the USN no matches to the current user and display it as a table.

On clicking the submit button, the code retrieves the book id entered by the user to return and store it in a php variable and establishes the connection between the database, again it retrieves the USN no of a student using the username of the session and select the book with the entered book id, if it is not equal to 0, that is if the book is reserved, set it back to zero.

Else it will display that, the user didn't reserve the book to return if the USN id and book chosen didn't match

Display Available books

RUAS LIBRARY MANAGEMENT SYSTEM

Welcome Yashaswini63

List of Books in library

book_id	title	author
BK103	WebArchitecture	Sneha
BK105	ISP	Joe
BK106	DBMS	Niyati
BK107	DBMS	Niyati
BK108	Graphics	Nishal
BK109	GTO	Sahitha
BK110	GTO	Sahitha

PHP code

```
session_start();
if($_SESSION["Username"]){
?>
Welcome <?php echo $_SESSION["Username"];
}
$cnn = mysqli_connect("localhost","root","","lms_system");
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
$sql="select * from books where USN = 0 ";
$result=mysqli_query($cnn,$sql);
echo "<table style='width:50%' border='1'>
    <caption> List of Books in library </caption>
    <tr>
        <th>book_id</th>
        <th>title</th>
        <th>author</th>
    </tr>";
if(mysqli_num_rows($result)>0){
    while($row=mysqli_fetch_assoc($result)){
        echo "<tr>";
        echo "<td><center>".$row["Book_Id"]."</center></td>";
        echo "<td><center>".$row["Book_Name"]."</center></td>";
        echo "<td><center>".$row["Author_Name"]."</center></td>";
        echo "</tr>";
        continue;
    }
}
else{
    echo '<script>alert("error");</script>';
}
```

The php code starts the session and establishes the connection to the database.

The sql query selects all the books from the table book where the USN number is set to 0 and displays it as a table

Search Book

RUAS LIBRARY MANAGEMENT SYSTEM

Enter the title

Search

HTML code

```
<div class="form">
<form role="form" id="templatemo-preferences-form" name="Search" action="" method="post" >

    <input type="text" id="search" placeholder="Enter the title"
           class="submit-button" name="searching" required><br/>

    <button type="submit" name="search"
           value="Search" class="submit-button">Search</button>

</form>
<div>
```

The HTML code for Search the book option has the elements housed in class “form” and has two inputs one of type text to enter book Id and the other of type submit

CSS code

```
body{
    font-family: montserrat;
    background: linear-gradient(120deg, #2980b9, #8e44ad);
    height: 100vh; /* checkout */
    overflow: hidden; /* checkout */
}
h1{
    text-align:center;
}
.form{
    clear:both;
    text-align:center;
}
.form form input{
    padding:2%;
    width:250px;
    border-radius: 25px;
    font-size:30px;
}
.form form button{
    width:100px;
    border-radius:25px;
    font-size:20px;
}
```

The CSS code for search the book option. Has The entire body has the same theme to match with the website. The heading is aligned center

The form element which contains the input is aligned center. The inputs are padded 2% to make the inputs size bigger and are rounded with border-radius of 25px.

PHP code

```
if (isset($_POST['search'])) {
    $name = $_POST['searching'];
    $con = mysqli_connect("localhost", "root", "", "lms_system");
    if (mysqli_connect_errno())
    {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
    $sql="select * from books where Book_Name LIKE '%$name%' and USN='0' ";
    $result=mysqli_query($con,$sql);
    echo "<table style='width:50%' border='1'>
        <caption>List of available books</caption>
        <tr>
            <th>book_id</th>
            <th>title</th>
            <th>author</th>
        </tr>";
    if(mysqli_num_rows($result)>0){
        while($row=mysqli_fetch_assoc($result)){
            echo "<tr>";
            echo "<td><center>". $row["Book_Id"] . "</center></td>";
            echo "<td><center>". $row["Book_Name"] . "</center></td>";
            echo "<td><center>". $row["Author_Name"] . "</center></td>";
            echo "</tr>";
        }
    }
    else{
        echo "Book not found";
    }
}
```

The PHP code executes on clicking the submit button that has the value search

It stores the bookname entered by the user to search in the php variable

Then it establishes the connection with the database

The sql query selects all the books which is similar to having the Book_name similar to what entered by the user and the USN_no where it is 0 and displays the book in the form of table

Hence, this will display all the books that are available.

Results

RUAS LIBRARY MANAGEMENT SYSTEM

Login

Username
Yashaswini63

Password
.....

Login

[Register](#)

Login with username and password

RUAS LIBRARY MANAGEMENT SYSTEM

Sign Up

USN
2160

Username
Samuel81

Password
.....|

Register

Sign up with the USN No of the student, Username and Password

localhost says
Registration successful

OK

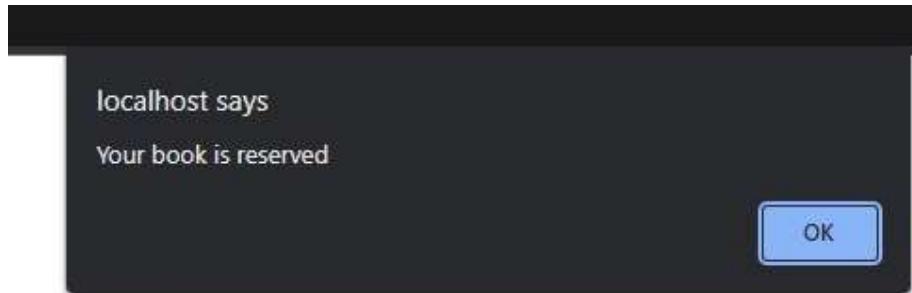
Successful registration prompt



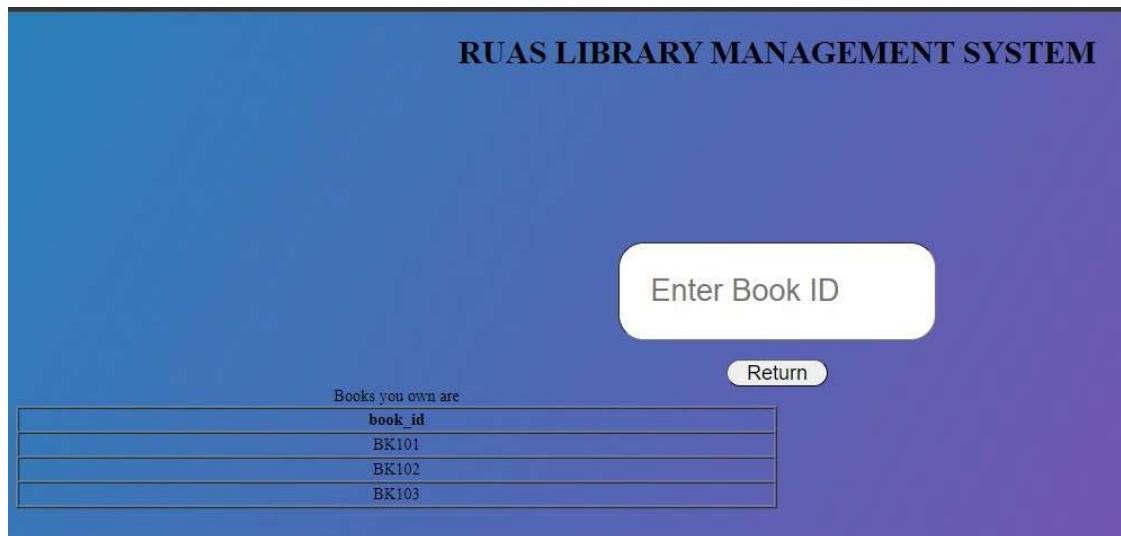
Home Page with Four different functionalities the user can perform and a log out button



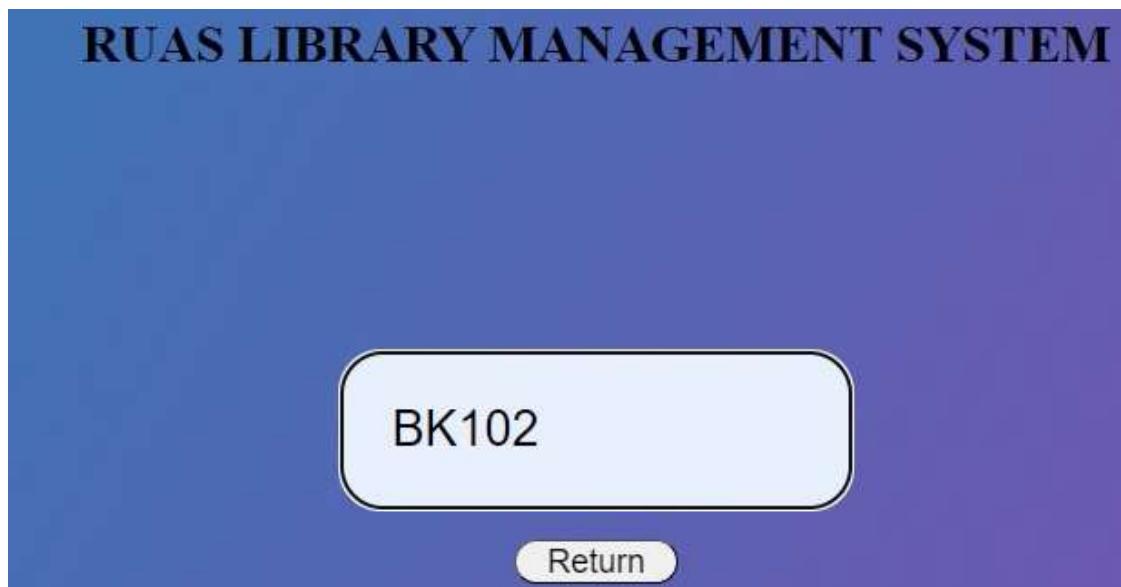
Reserving Book of Book_Id "BK103"



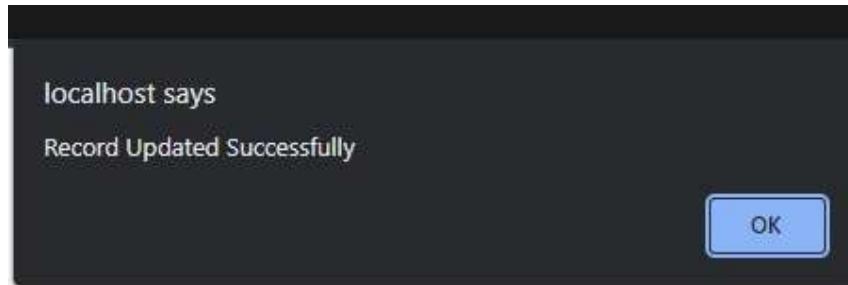
On clicking the Reserve button, the user gets the prompt



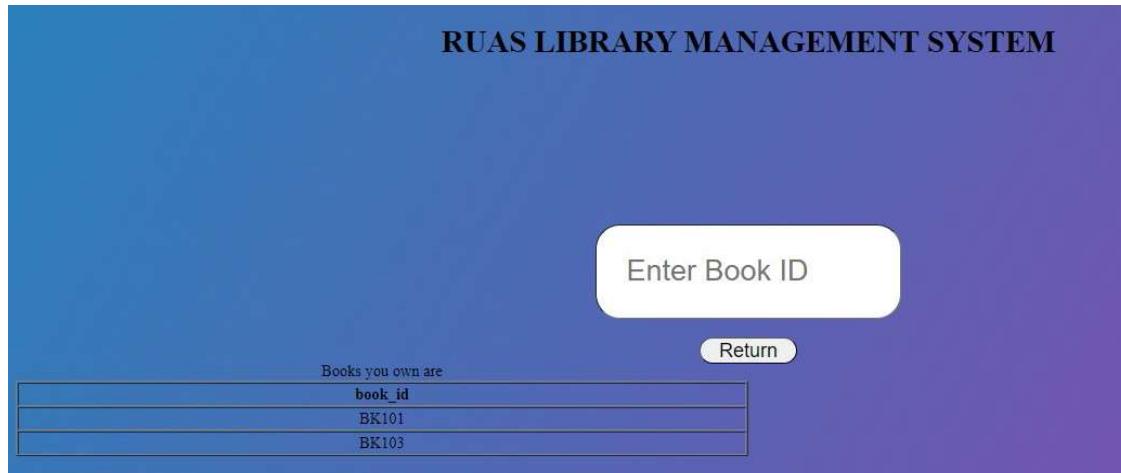
The Return Book Option : It displays all the books the user holds



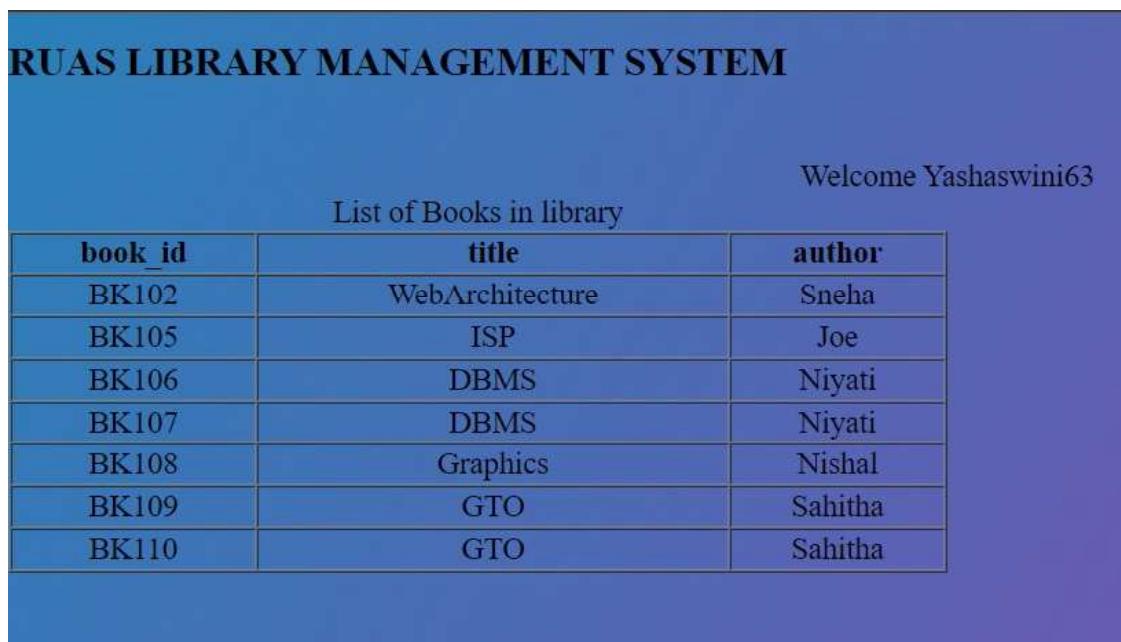
Entering the BookId of the Book that needs to be written



The record of return is updated successfully



Once the record is updated , it removes from the table which displays the Book the user own



The Display from the home page, will display all the books that are available

RUAS LIBRARY MANAGEMENT SYSTEM

Web

Search

The Search Option from the home page, taking the input of book_name

List of available books		
book_id	title	author
BK102	WebArchitecture	Sneha

It displays only the available books in the list of books

Note : the list of books in the database

Book_Id	Book_Name	Author_Name	USN
BK101	WebArchitecture	Sneha	2159
BK102	WebArchitecture	Sneha	0
BK103	WebArchitecture	Sneha	2159
BK104	ISP	Joe	2160
BK105	ISP	Joe	0