Web Architecture and Application Development Laboratory

B.Tech. 7th Semester

Department: Computer Science and Engineering

Faculty of Engineering & Technology
M. S. Ramaiah University of Applied Sciences

Name: K Srikanth Roll number: 17ETCS002124

Ramaiah University of Applied Sciences

Private University Established in Karnataka State by Act No. 15 of 2013

Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Course	Web Architecture and Application Development Laboratory
Year/Semester	2015/7th Semester
Course Code	CSC404A

List of Experiments

1. Software design -1
2. Software design -2
3. Database design
4. PHP form for student registration
5. PHP form for login and dashboard
6. HTML user interface
7. HTML user interface
8. Search implementation
9. Functionality implementation
10. Mock

Consider a scenario where a student enrolls for a course in university. The student also registers in the library available at university by providing his basic details. Consider all the attributes of the student and library to develop the web application design.

Name: K Srikanth Roll number: 17ETCS002124

Laboratory 1

Title of the Laboratory Exercise: Software Design -1

1. Introduction and Purpose of Experiment

   Web application design by formulating functional and non-functional requirements helps developers understand the various needs of the client on a basic level. ER diagrams assists developers in designing the database and understand user interaction as well.

2. Aim and Objectives

   Aim

   - To develop
     1. Functional & Nonfunctional requirements
     2. ER diagram

   Objectives

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

   - Formulate Functional Requirements and Non-functional requirements
   - Model the information required for the given scenario using E-R diagrams

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 requirement:

   User login:

   1. The system must only allow user with valid id and password to enter the system.
   2. If not, system should allow user to register as new user.
   3. The system performs authorization process which decides what user level can access to.
   4. The user must be able to logout after they finished using system.

   Register new user:

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

   Search book:

   7. System should allow the user to search the book by book name.

8. System should allow the user to search books by genre or type.

Purchase book:

9. System should allow the user to add the books to cart and display the amount.

10. System should allow the user to purchase the books from the cart with a buy option.

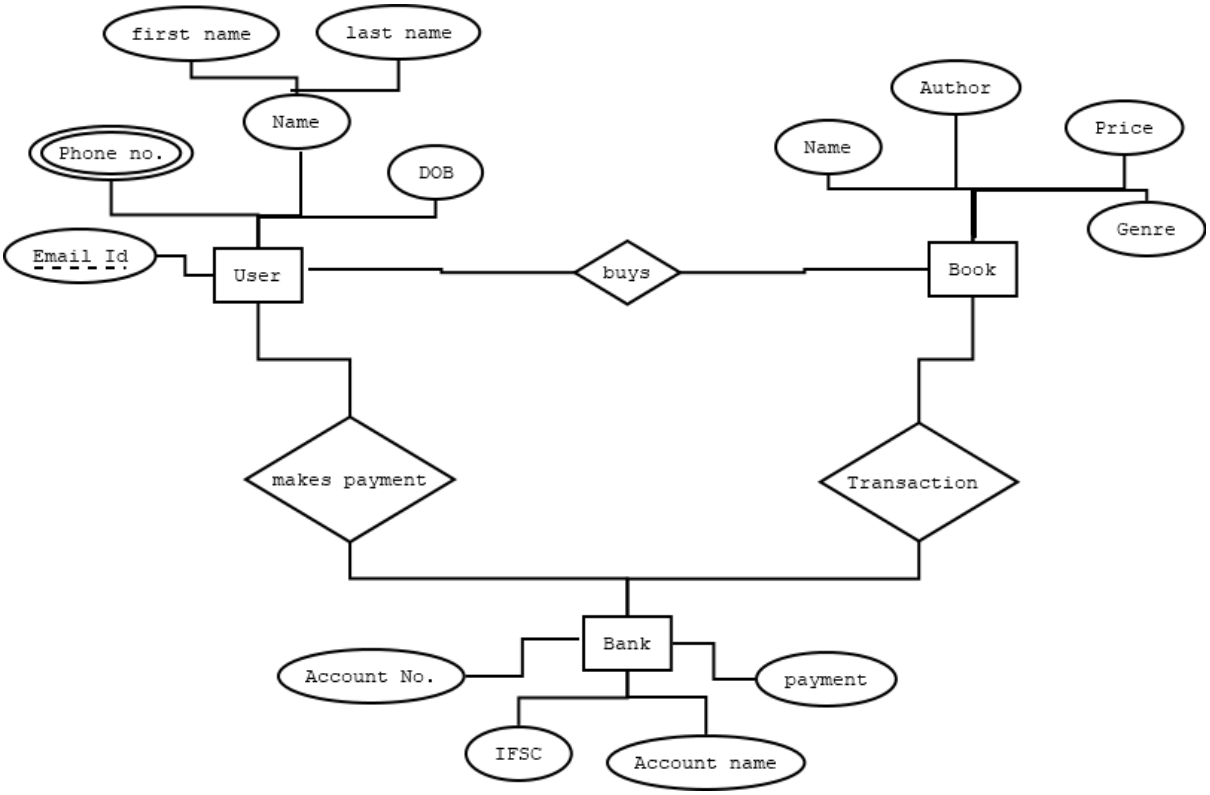
Non-Functional Requirement:

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

2. The system should accurately perform member registration, member validation, report generation, book transaction and search.

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

ER diagram:



5. Analysis and Discussions

In the above, the base for the library management system is shown, where it contains the functional requirement and non-functional requirement and ER diagram for a better understanding.

Here, the LMS allows the user to log in with name and password, if the user isn't registered then the system will allow the user to register as a new user to the system. The user can add the book to the

cart if he wants to purchase it and user can add how much ever book user wants and can buy it in the cart. User is allowed to search the book by the name or the type or genre of the book. And non-functional requirement shows the reliability and accessibility of the website we are trying to develop. The ER diagram shows us the relationship between the entities those are user, book and bank with their attribute.

6. Conclusions

The functional and non-functional requirements are stated in the above for a LMS which we will be coding in the future labs and for a better understanding we have developed ER diagram showing the relationship between the entities.

7. Comments

a. Limitations of Experiments

Not everything can be specified before developing a software.

b. Limitations of Results

Not everything can be specified in ER diagram.

c. Learning happened

As I have mentioned in the above, we have developed ER diagram using functional requirement and we have mentioned non-functional requirement.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: K Srikanth Roll number: 17ETCS002124

Laboratory 2

Title of the Laboratory Exercise: Software Design - 2

1. Introduction and Purpose of Experiment

   Web application design using class diagrams and sequence diagram helps developers understand the scenario from a more technical approach.

2. Aim and Objectives

   Aim:

   - To develop

     1. Class diagram,
     2. Interaction Sequence diagram
     3. Algorithm/flowchart

   Objectives

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

   - Design scenarios using class and sequence diagrams.

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. Algorithms

Step 1: start

Step 2: Create a table with user name and password with user details.

Step 3: create a user log in page and register page if user isn't registered.

Step 4: Upload the registered user to the user table.

Step 5: Store all the book details to the database with its attribute by creating a table.

Step 6: Create a search option for the user to search by name or genre and let the user add books to cart.

Step 7: Create a Cart page to show the books which are added to the card if its available with purchase option.

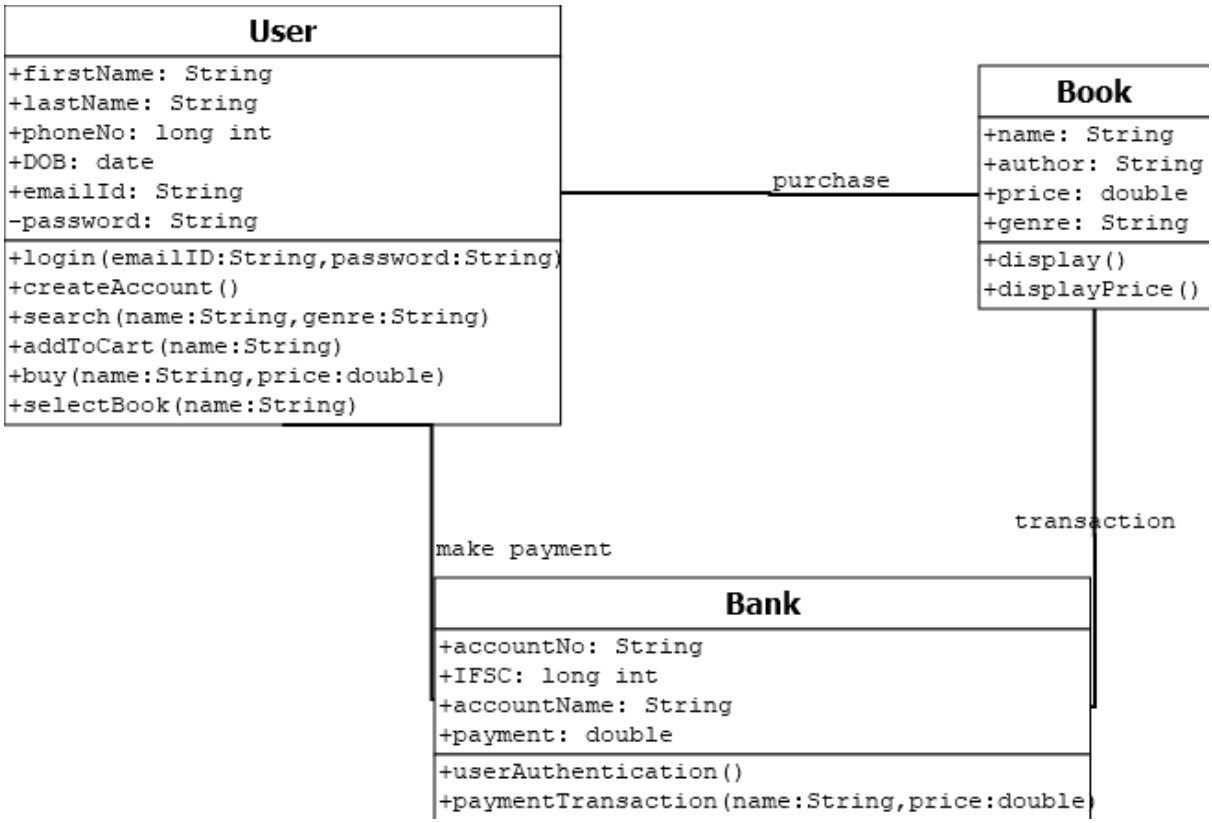
Step 8: Create a purchase option for the user then enter the back details and delete the book count after the books purchased.

Step 9: Create a log out if the user is done with the purchase.

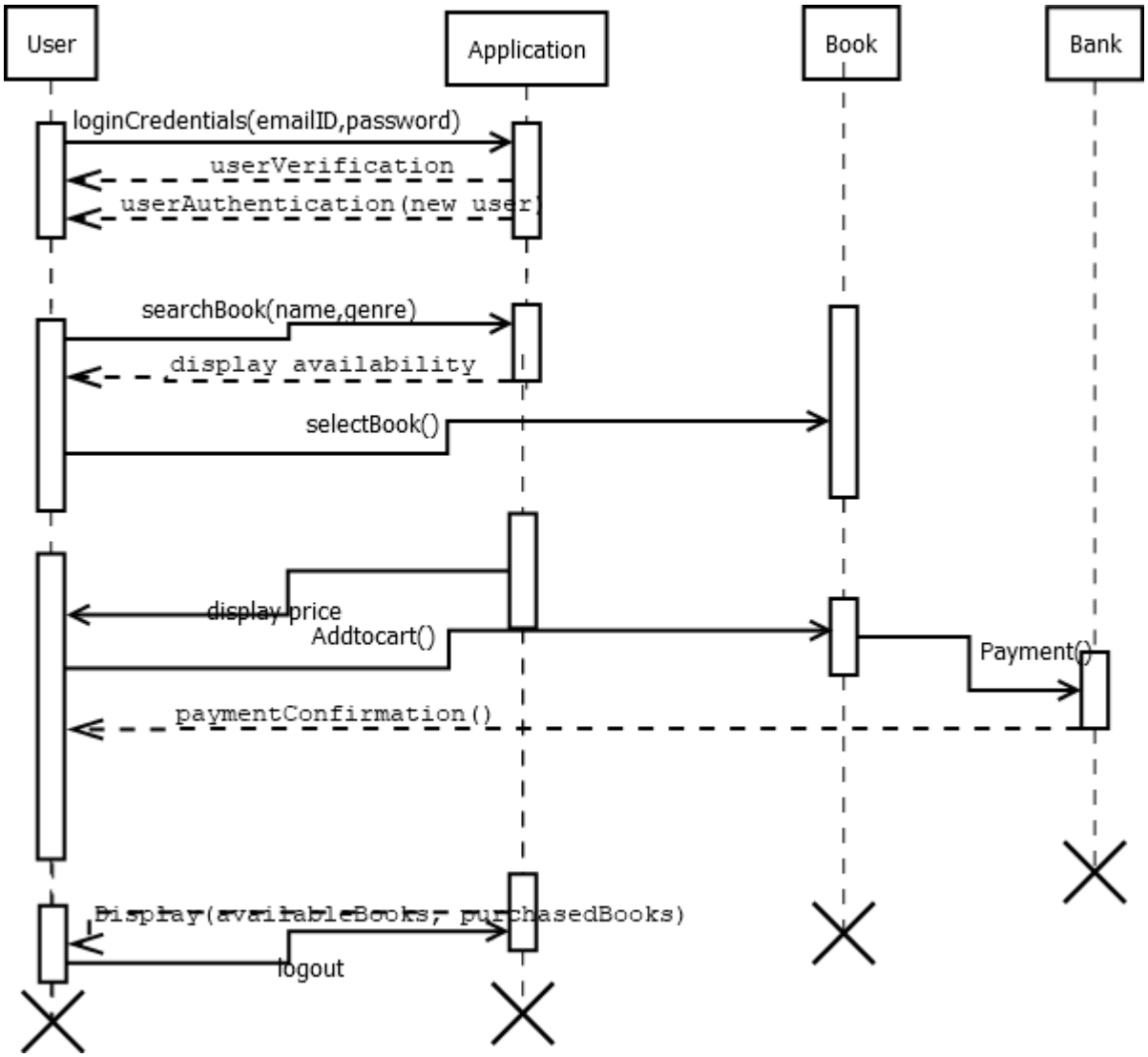
Step 10: End.

5. Presentation of Results

Class diagram:



Sequence Diagram:



6. Analysis and Discussions

The above developed class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Where the class has structural and behavioral features and a class has its name, attribute and operations or methods helps us understand the system easily. A class may be involved in one or more relationships with other classes which are of few different types.

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

7. Conclusions

We have developed sequence and class diagram for the specification we had mentioned in the lab 1 and also developed algorithm elaborating the process of developing the system.

8. Comments

a. Limitations of Experiments

it's slower than ad hoc diagramming. as a model of a complicated system (the code), it's still fairly complicated.

b. Limitations of Results

No such limitations of the results as we are just developing a diagram.

c. Learning happened

the developing the class and sequence diagram for the functional requirement and understanding the attributes and methods involved in developing them and the relationship they determine between the entities.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Laboratory 3

Title of the Laboratory Exercise: Database Design

1. Introduction and Purpose of Experiment

   Students will learn to use HTML and JavaScript in html platform.

2. Aim and Objectives Aim: To design the database

   Objectives:

   1. Create database
   2. Create tables
   3. Data entry for tables

3. Experimental Procedure
4. Presentation of Results

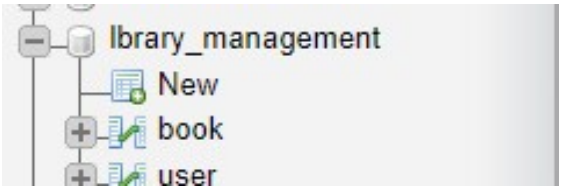


Figure 3.1: Image showing the tables in library database.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_id	varchar(10)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index Spatial More
2	name	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index Spatial More
3	phone_number	int(10)			No	None			Change Drop Primary Unique Index Spatial More
4	email	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index Spatial More
5	password	varchar(10)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index Spatial More

Figure 3.2: Image showing the structure of test table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	book_id	int(11)			No	None			Change Drop More
2	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	author	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	imageUrl	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
5	price	double			No	None			Change Drop More
6	reserved	varchar(1)	utf8mb4_general_ci		No	None			Change Drop More

Figure 3.4: Image showing the structure of books table.

	book_id	name	author	imageUrl	price	reserved
Edit Copy Delete	1	The Last Sister	Kendra Elliot	https://encrypted-tbn2.gstatic.com/images?q=tbn:AN...	450	0
Edit Copy Delete	2	In an Instant	Suzanne Redfearn	https://encrypted-tbn2.gstatic.com/images?q=tbn:AN...	330	0

Figure 3.5: Image showing the content of books table.

5. Analysis and Discussions

In this lab we have created a database using PHPMyAdmin. Here the database name is lbrary_Management. In the database 2 tables are present Book and User. Name is the primary key in test table and book_id is primary key in books table. And the book table contains all the attributes needed to add the book to dashboard.

6. Conclusions

The creation of database for the user and the books to be displayed is successfully deployed and are shown in the above from phpMyAdmin which is from xampp server.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Laboratory 4

Title of the Laboratory Exercise: PHP form for student registration

1. Introduction and Purpose of Experiment

   Students will learn to use css and javascript in html platform.

2. Aim and Objectives

   Aim: Creation of PHP form for student registration.

   Objectives:

   - Create basic HTML UI
   - PHP code for registration and saving user data to database.

3. Experimental Procedure
4. Presentation of Results

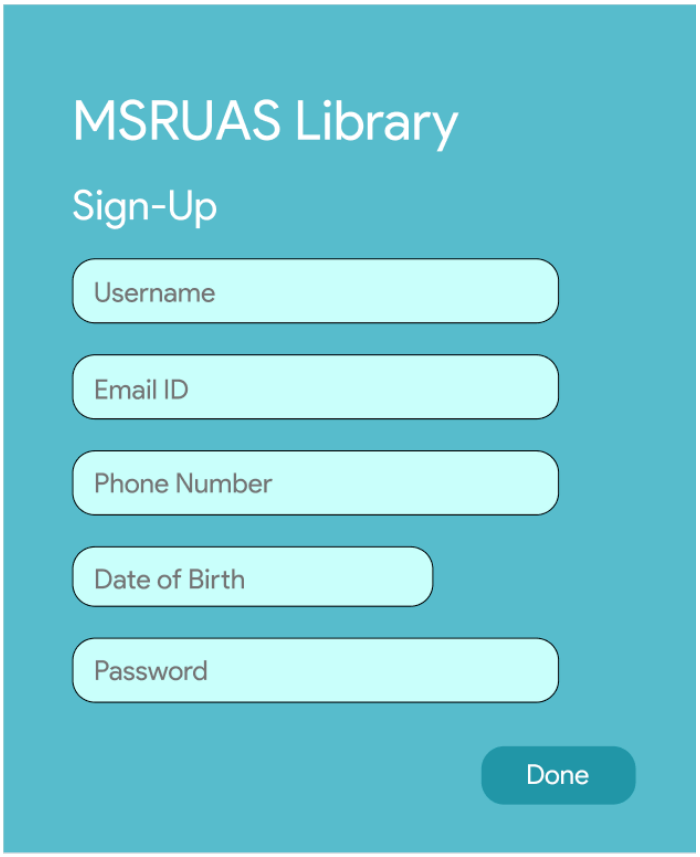


Figure 4.1: index.php for user registration.

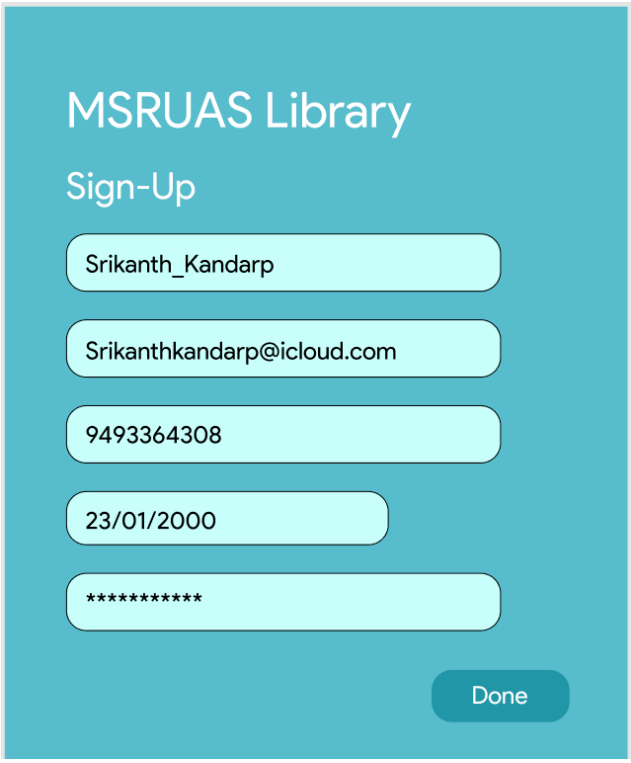


Figure 4.2: Image showing the index.php page with inputs filled.

```
<body>
    <?php

    $Name = $Age = $Contact = $Password = "" ;
    if($_SERVER["REQUEST_METHOD"]=="POST"){
        $Name = $_POST['name'];
        $Age = $_POST['age'];
        $Contact = $_POST['contact'];
        $Password = $_POST['password'];

        $con = mysqli_connect("localhost","root","","library");


                    $query = "INSERT INTO `testtable` (age, contact, name, password) VALUES ('$Age', '$Contact', '$Name', '$Password')";
                    $result = mysqli_query($con,$query);
                    if($result){
                     echo ("Registration Succesful");
                                }
                                else
                                {
                                    echo ("Fail");
                                }
                     }
    ?>


    <form  action="" method="post" >
        <div class="center">
        <h1>Registration </h1>
        <label>NAME</label></br><input type="text"   name="name"><br/>
        <label>AGE</label></br><input type="text"   name="age"><br/>
        <label>CONTACT</label></br><input type="text"   name="contact"><br/>
        <label>PASSWORD</label></br><input type="text"   name="password"><br/></br>
        <button type="submit"  name="submit" value="Register" >Update</button>
        </div>
    </form>

</body>
```

Figure 4.3: The functionality of the registration page is shown here with php and html code.

5. Analysis and Discussions

The data is taken from the input fields or text fields of HTML and then are passed to the php code which is the backend of this website. The php then is connected to the database which is sql using its connector functions which will be updated with the new user and if the user already exists then the website leads the user to log in page.

6. Conclusions

We were successfully able to create a basic HTML interface to register a user by using MySQL as the database for user info storage and used php to handle the data from front and backend.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Laboratory 5

Title of the Laboratory Exercise: PHP form for login and dashboard

1. Introduction and Purpose of Experiment
2. Aim and Objectives

   Aim: Creation of PHP form for login and dashboard

   Objectives

   - Create basic HTML UI
   - PHP code for login and checking user data from database.
   - Routing to dashboard and displaying items.

3. Experimental Procedure
4. Presentation of Results

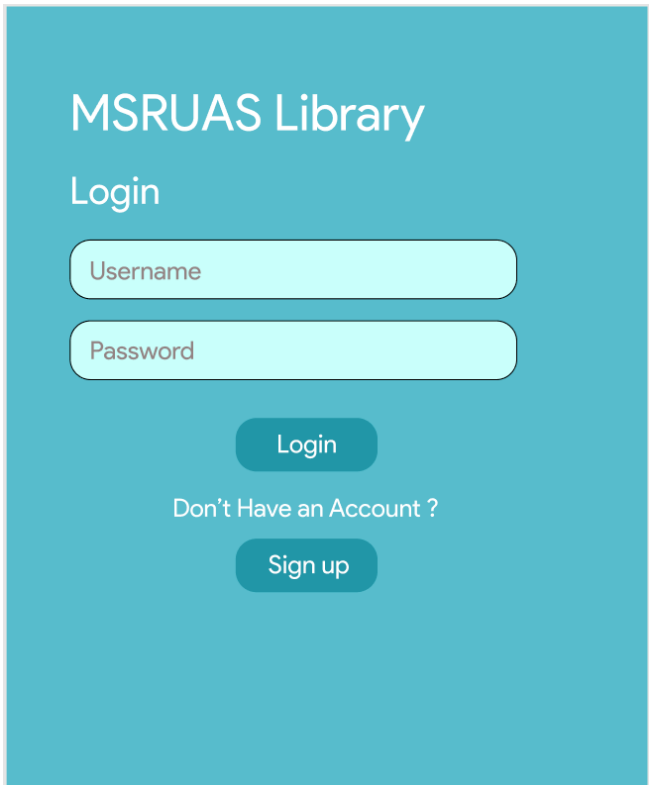


Figure 5.1:Index.php for log in page.

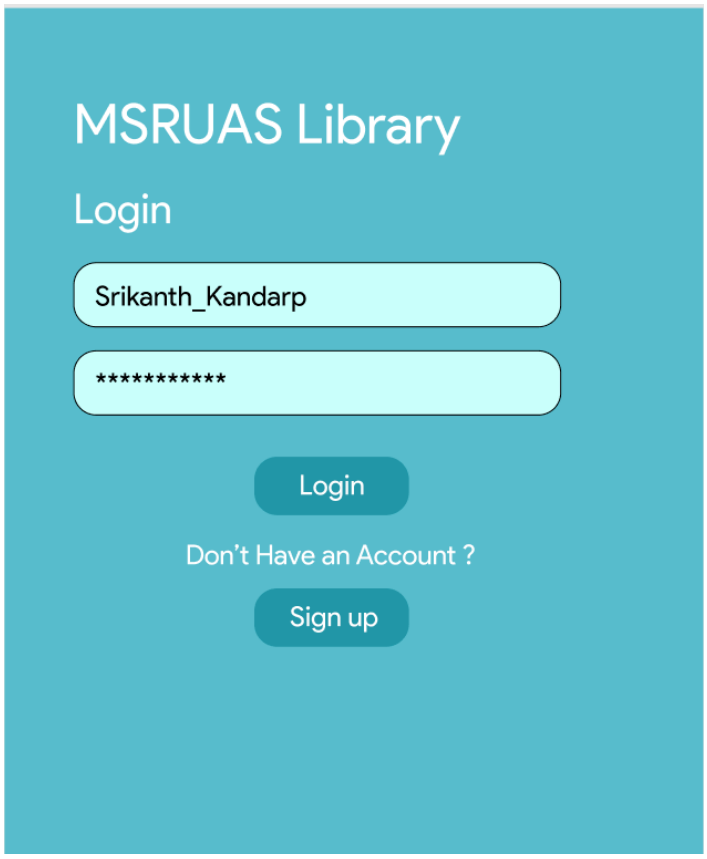


Figure 5.2: The log in page with input variables in the text field.

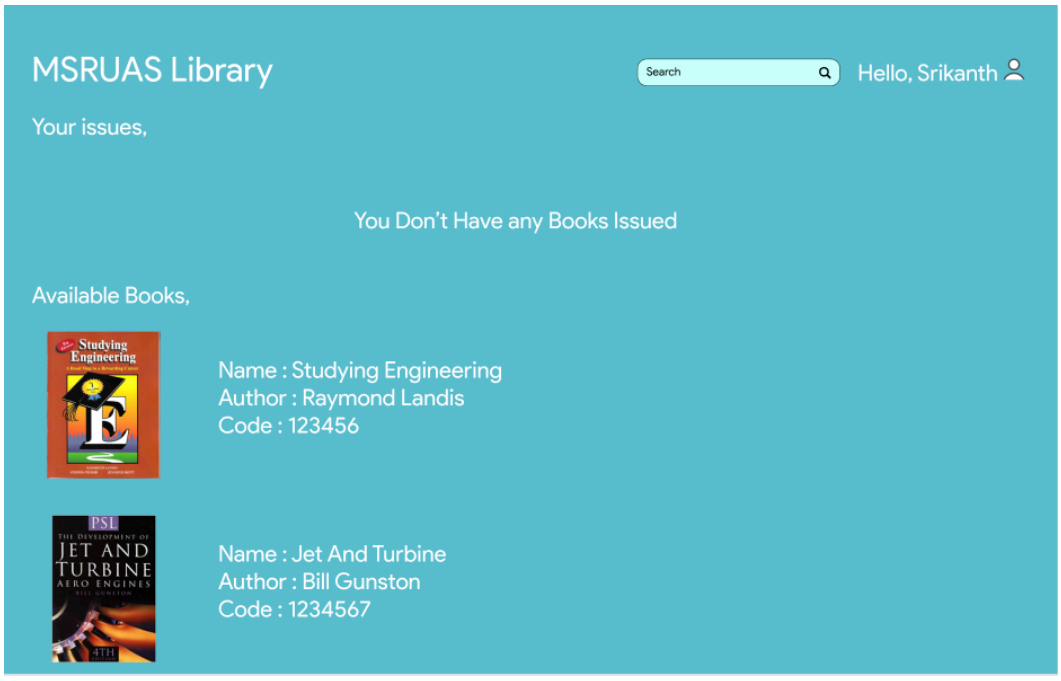


Figure 5.3: The dashboard displayed after the log in is successful.

```php
<?php
if (isset($_POST['username'])) {
    $username = $_POST['username'];

    $password = $_POST['password'];

    $con = mysqli_connect("localhost", "root", "", "library");

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_errno();
    }

    $query = "SELECT * FROM testtable WHERE name='$username' and password='$password'";
    $result = mysqli_query($con, $query);

    if ($result) {
        if (mysqli_num_rows($result) > 0) {
            session_start();
            $_SESSION['username'] = $username;
            header("Location: home.php");
        } else {
            echo("Invalid credentials");
        }
    } else {
        echo("Invalid credentials");
    }
}
?>


<form role ="form" id="templateno-preferences-form" name="login" action="" method="post" >
    <div class="center">
        <h1>Login </h1>
        <label>Username</label>
        </br>
        <input type="text" id="username" placeholder="Enter username" name="username" required>
        </br>
        </br>
        <label>Password</label>
        </br>
        <input type="password" id="password" placeholder="Enter password"   name="password" required>
        <br/></br>
        <button type="submit"  name="submit" value="Register" >Login</button>
    </div>
</form>
```

Figure 4.4: Image showing the PHP code for logging in along with base HTML code.

5. Analysis and Discussions

The html and CSS are used to design the frontend of the log in page linked to database with php, here the input from the input text fields are taken then authenticated in the database with username and the password match and then the log in successful is lead to dash board page where the products are displayed.

6. Conclusions

We were successfully able to create a basic HTML interface to login a user by using MySQL to check if user's password and username match.

Name: K Srikanth Roll number: 17ETCS002124

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Laboratory 6

Title of the Laboratory Exercise: HTML user interface

1. Introduction and Purpose of Experiment
2. Aim and Objectives

   Aim: To update the HTML UI for registration page

   Objectives

   Update the Old HTML page using bootstrap components.

3. Experimental Procedure
4. Presentation of Results

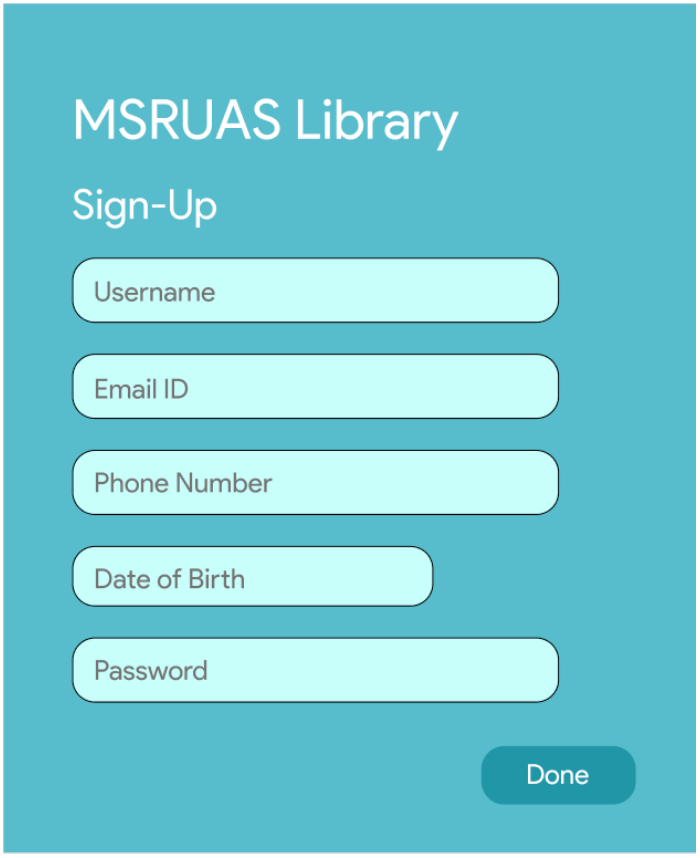


Figure 6.1: User interface for registration page.

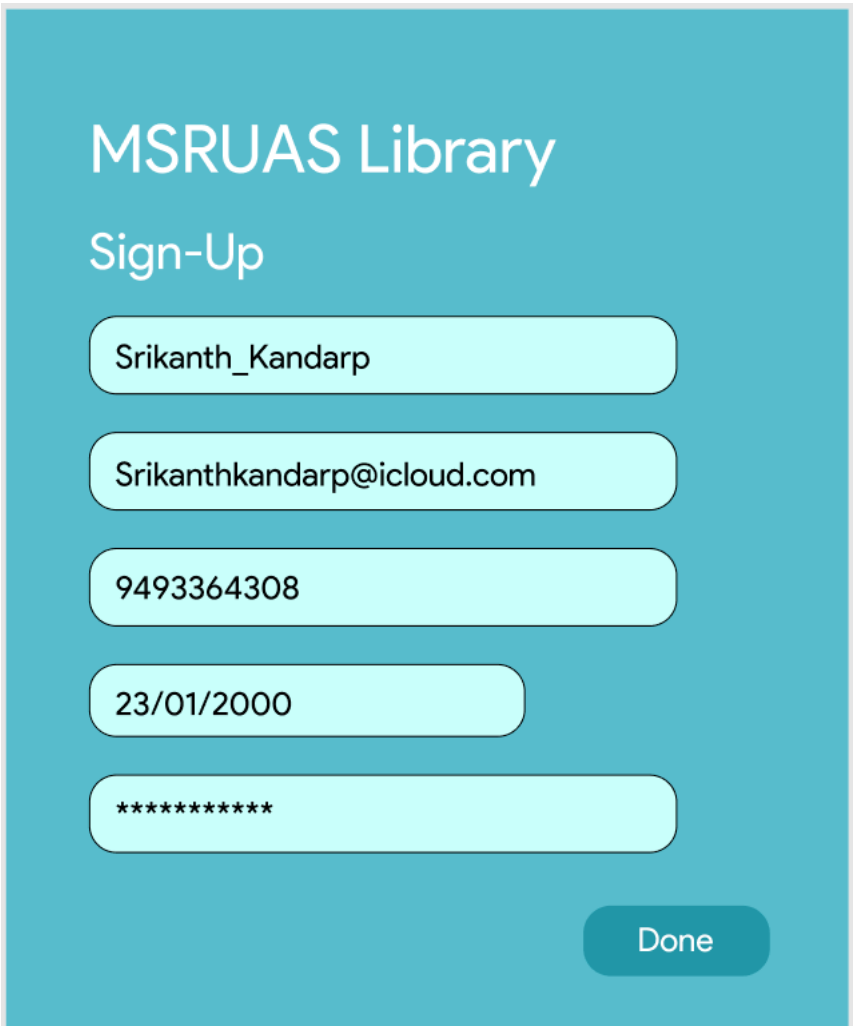


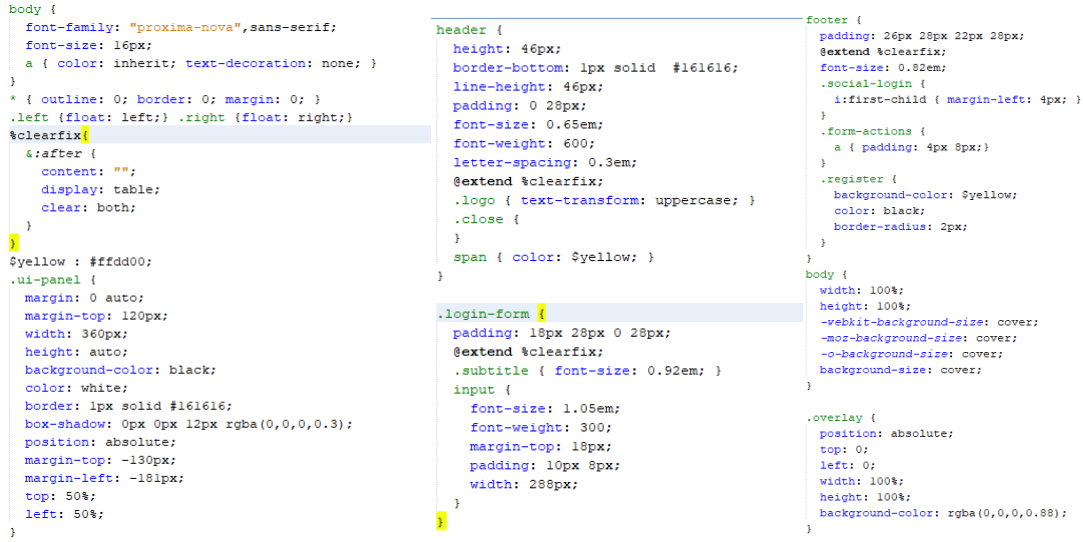
Figure 6.2: User interface for registration page with data in test fields.

```
<body class="text-center">
    <?php
    $Name = $Age = $Contact = $Password = "";
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $Name = $_POST['name'];
        $Age = $_POST['age'];
        $Contact = $_POST['contact'];
        $Password = $_POST['password'];

        $con = mysqli_connect("localhost", "root", "", "library");


        $query = "INSERT INTO `testtable` (age, contact, name, password) VALUES ('$Age', '$Contact', '$Name', '$Password')";
        $result = mysqli_query($con, $query);
        if ($result) {
            echo ("<script type=\"text/javascript\">" .
                "alert('Registration successful.');" .
                "</script>");
        } else {
            echo ("Fail");
        }
    }
    ?>
```

Figure 6.4: Updated PHP code for new UI.



```
body {
  font-family: "proxima-nova",sans-serif;
  font-size: 16px;
  a { color: inherit; text-decoration: none; }
}
* { outline: 0; border: 0; margin: 0; }
.left {float: left;} .right {float: right;}
%clearfix{
  &:after {
    content: "";
    display: table;
    clear: both;
  }
}
$yellow : #ffdd00;
.ui-panel {
  margin: 0 auto;
  margin-top: 120px;
  width: 360px;
  height: auto;
  background-color: black;
  color: white;
  border: 1px solid #161616;
  box-shadow: 0px 0px 12px rgba(0,0,0,0.3);
  position: absolute;
  margin-top: -130px;
  margin-left: -181px;
  top: 50%;
  left: 50%;
}
```

```
header {
  height: 46px;
  border-bottom: 1px solid  #161616;
  line-height: 46px;
  padding: 0 28px;
  font-size: 0.65em;
  font-weight: 600;
  letter-spacing: 0.3em;
  @extend %clearfix;
  .logo { text-transform: uppercase; }
  .close {
  }
  span { color: $yellow; }
}

.login-form {
  padding: 18px 28px 0 28px;
  @extend %clearfix;
  .subtitle { font-size: 0.92em; }
  input {
    font-size: 1.05em;
    font-weight: 300;
    margin-top: 18px;
    padding: 10px 8px;
    width: 288px;
  }
}
```

```
footer {
  padding: 26px 28px 22px 28px;
  @extend %clearfix;
  font-size: 0.82em;
  .social-login {
    i:first-child { margin-left: 4px; }
  }
  .form-actions {
    a { padding: 4px 8px;}
  }
  .register {
    background-color: $yellow;
    color: black;
    border-radius: 2px;
  }
}
body {
  width: 100%;
  height: 100%;
  -webkit-background-size: cover;
  -moz-background-size: cover;
  -o-background-size: cover;
  background-size: cover;
}

.overlay {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0,0,0,0.88);
}
```

Figure 6.5: CSS code for registration and log in page.

5. Analysis and Discussions

The UI part of registration page is shown here, the CSS part of the UI design is also put in figures, the php behind this page is shown which helps in working and feeding data to sql queries for updating.

6. Conclusions

We successfully updated the UI for registration page with the help of bootstrap.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Laboratory 7

Title of the Laboratory Exercise: HTML user interface

1. Introduction and Purpose of Experiment 2.

Aim and Objectives

Aim: To update the HTML UI for login page

Objectives

Update the Old HTML page using bootstrap components.

3. Experimental Procedure
4. Presentation of Results

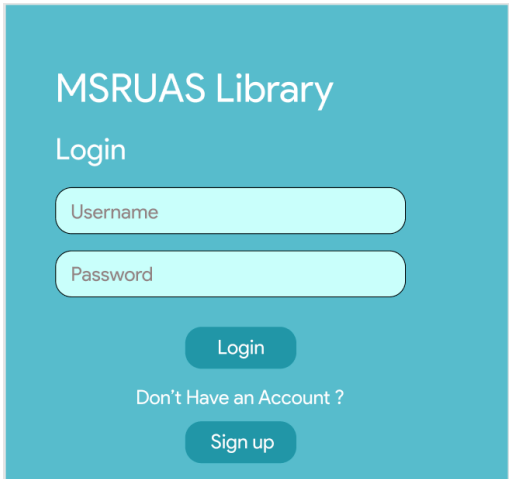


Figure 7.1: Updated UI for Login page

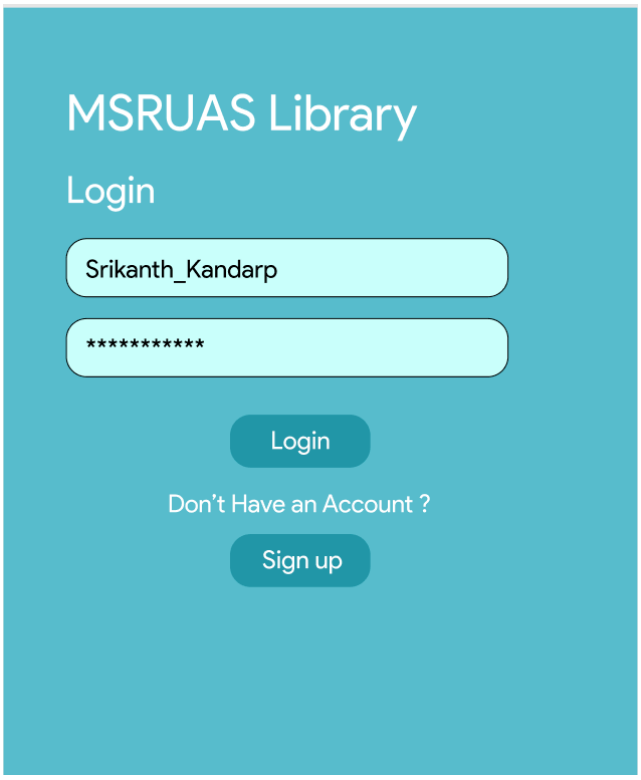


Figure 7.2: Updated UI for Login page with test fields filled.

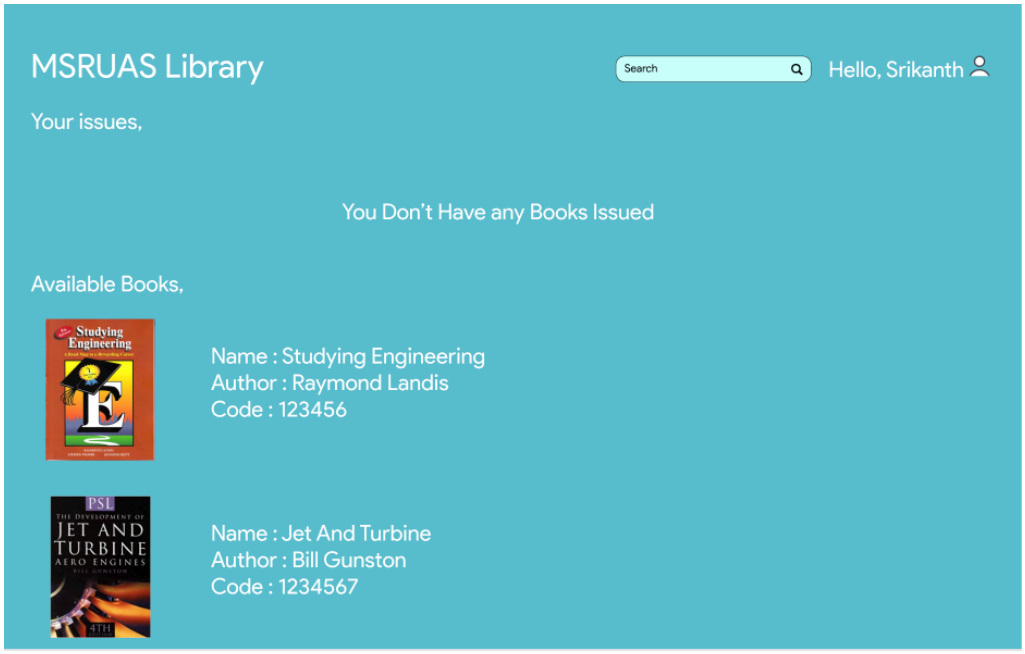


Figure 7.3: Dashboard displayed with the user name.

```php
<?php
$name = $username = "";
if (isset($_POST['username']) || isset($_POST['name'])) {
    $name = $_POST['name'];
    $password = $_POST['password'];
    $con = mysqli_connect("localhost", "root", "", "library");
    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_errno();
    }
    $query = "SELECT * FROM testtable WHERE (name='$name') and password='$password'";
    $result = mysqli_query($con, $query);
    if ($result) {
        if (mysqli_num_rows($result) > 0) {
            session_start();
            $_SESSION['username'] = $username;
            $_SESSION['name'] = $name;

            header("Location: home.php");
        } else {
            echo ("Invalid credentials");
        }
    } else {
        echo ("Invalid credentials");
    }
}
?>
<form role="form" id="templateno-preferences-form" name="login" action="" method="post" class="form-signin">
    <img class="mb-4" src="assets/books-logo.png" alt="" width="72" height="72">
    <h1 class="h3 mb-3 font-weight-normal">Login </h1>
    <label for="inputName" class="sr-only">Name</label>
    <input type="text" name="name" id="inputName" class="form-control" placeholder="Name" required autofocus>
```

Figure 7.4: PHP code for handling things on Login page.

```php
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <a class="navbar-brand" href="home.php"><img src="assets/books-logo.png" alt="" width="40" height="40"> Library</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item">
                <a class="nav-link"><?php
                                session_start();
                                $name = $_SESSION['name'];
                                print(explode("@", "Welcome " . $name)[0]); ?>
                    <span class="sr-only">(current)</span></a>
            </li>

            <li class="nav-item">
                <div class="dropdown">
                    <div class="dropdown-menu">
                        <?php
                        if (isset($_POST['search'])) {
                            // session_start();

                            $_SESSION['search'] = $_POST['search'];
                            header("Location: search.php");
                        }

                        ?>
                    </div>
                </div>
            </li>

        </ul>
        <form class="form-inline my-2 my-lg-0" method="post" role="form" id="templateno-preferences-form">
            <input class="form-control mr-sm-2" type="text" placeholder="Search library" aria-label="Search" name="search" id="search">
            <button class="btn btn-outline-success my-2 my-sm-0" type="submit" name="searchQ" value="Search">Search</button>
        </form></div> </nav>
```

Figure 7.5: PHP and HTML code for Items on Nav Bar.

```
<h1 class="display-4 center topPadding"><img src="assets/books-logo.png" alt="" width="100" height="100"> Library</h1>

<form method="post">
    <div class="container">
        <div class="row">
            <?php
            $con = mysqli_connect("localhost", "root", "", "library");

            if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_errno();
            }

            $query = "select * from books";
            $result = mysqli_query($con, $query);

            if (mysqli_num_rows($result) > 0) {
                while ($row = mysqli_fetch_assoc($result)) {

                    if (isset($_POST[$row["book_id"]])) {

                        $con = mysqli_connect("localhost", "root", "", "library");

                        if (mysqli_connect_errno()) {
                            echo "Failed to connect to MySQL: " . mysqli_connect_errno();
                        }

                        $reserve_query = "update `books` set reserved=1 where book_id=".$row['book_id'];

                        $res = mysqli_query($con, $reserve_query);

                        if ($res) {
                            // print("<script>alert('success');</script>");
                        } else {
                            echo ("Error: " . $res);
                        }

                    }
                    if($row["reserved"]==0){
                                                    echo '<div class="col-lg-4 col-md-6 mb-4">
                    <div class="card h-100">
                        <a href="#"
                        ><img
                            class="card-img-top"
                            src= ' . $row["imageUrl"] . '
                            alt=""
                        />
                        </a>
                        <div class="card-body">
                        <h4 class="card-title">
                            <a href="#">' . strval($row["name"]) . '</a>
                        </h4>
                        <h5>₹' . $row["price"] . '</h5>
                        </div>
                        <div class="card-footer">
                        <input type="submit" class="btn btn-primary btn-sm" name=' . $row["book_id"] . ' value="Reserve"/>
                        </div>
                    </div>
                    </div>';
                    }
                    else{
                                                    echo '<div class="col-lg-4 col-md-6 mb-4">
                    <div class="card h-100">
                        <a href="#"
                        ><img
                            class="card-img-top"
                            src= ' . $row["imageUrl"] . '
                            alt=""
                        />
                        </a>
                        <div class="card-body">
                        <h4 class="card-title">
                            <a href="#">' . strval($row["name"]) . '</a>
                        </h4>
                        <h5>₹' . $row["price"] . '</h5>
                        </div>
                        <div class="card-footer">
                        <input type="submit" disabled class="btn btn-primary btn-sm disabled" name=' . $row["book_id"] . ' value="Reserve"/>
                        </div>
                    </div>
                    </div>';
                    }

                }
            } else {
                echo "error";
            } ?>
        </div>
    </div>
</form>
```

Figure 7.6: PHP and HTML code for displaying items on the dashboard.

5. Analysis and Discussions

The html and php behind user log in and dashboard is shown in here. The link of php and sql is also shown in the codes itself. The dashboard gets updated with the user name and into their data base to update any info.

6. Conclusions

We successfully updated the UI for login page and dashboard with the help of bootstrap.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Laboratory 8

Title of the Laboratory Exercise: Service Development: Search implementation

1. Introduction and Purpose of Experiment

Students learnt to develop restful web services for the given scenario.

2. Aim and Objectives

Aim: To implement search functionality into our website.

3. Objectives

4. Experimental Procedure

6. Presentation of Results

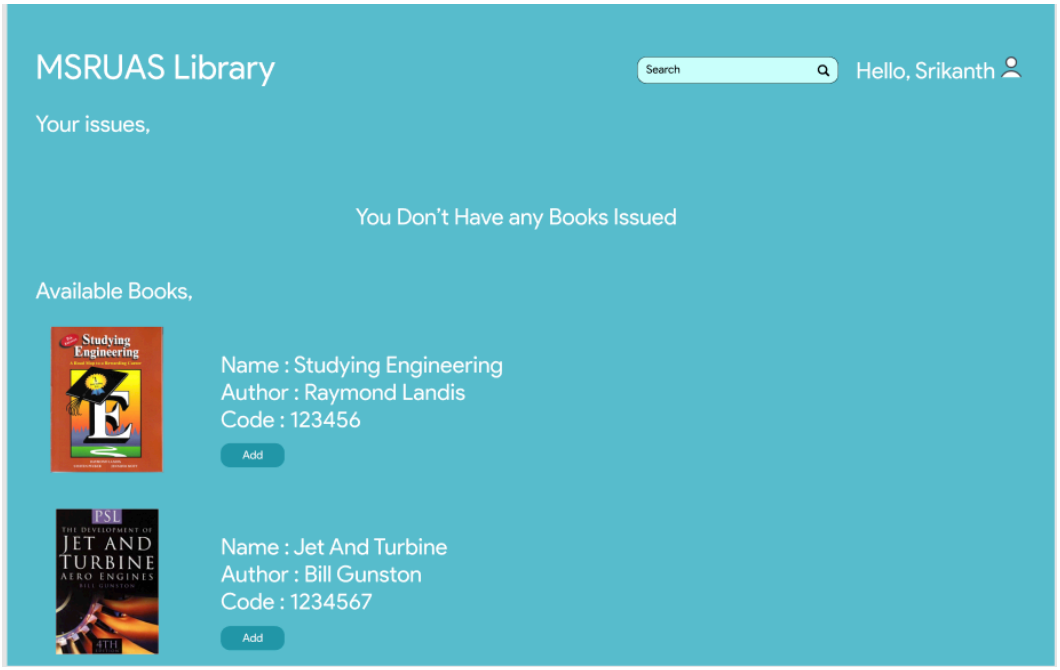


Figure 8.1:

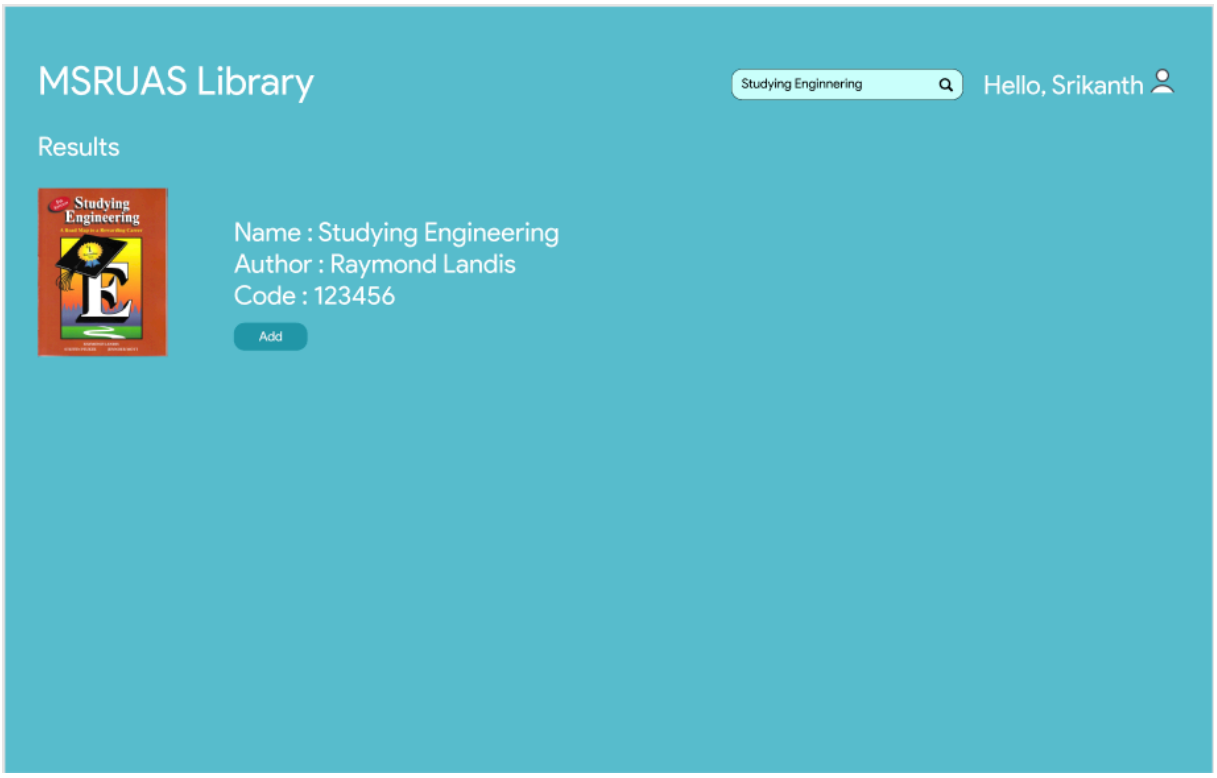


Figure 8.2: when “Studying Engineering” is typed, the above is shown.

```
    <li class="nav-item">
        <div class="dropdown">
            <div class="dropdown-menu">
                <?php
                if (isset($_POST['search'])) {
                    // session_start();

                    $_SESSION['search'] = $_POST['search'];
                    header("Location: search.php");
                }

                ?>
            </div>
        </div>
    </li>

</ul>
<form class="form-inline my-2 my-lg-0" method="post" role="form" id="templateno-preferences-form">
    <input class="form-control mr-sm-2" type="text" placeholder="Search library" aria-label="Search" name="search" id="search">
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit" name="searchQ" value="Search">Search</button>
</form></div> </nav>
```

Figure 8.3: here the search function is implemented in php.

.

```
<form method="post">
    <div class="container">
        <div class="row">
            <?php
            $con = mysqli_connect("localhost", "root", "", "library");

            if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_errno();
            }

            $search = $_SESSION['search'];

            $query = "select * from books  WHERE name LIKE '%$search%'";
            $result = mysqli_query($con, $query);

            if (mysqli_num_rows($result) > 0) {
                while ($row = mysqli_fetch_assoc($result)) {

                    if (isset($_POST[$row["book_id"]])) {

                        $con = mysqli_connect("localhost", "root", "", "library");

                        if (mysqli_connect_errno()) {
                            echo "Failed to connect to MySQL: " . mysqli_connect_errno();
                        }

                        $reserve_query = "update `books` set reserved=1 where book_id=".$row['book_id'];

                        $res = mysqli_query($con, $reserve_query);

                        if ($res) {
                            // print("<script>alert('success');</script>");
                        } else {
                            echo ("Error: " . $res);
                        }
                    }
                    if($row['reserved']==0){
                        echo '<div class="col-lg-4 col-md-6 mb-4">
                    <div class="card h-100">
                        <a href="#"
                        ><img
                            class="card-img-top"
                            src= ' . $row["imageUrl"] . '
                            alt=""
                        /></a>
                        <div class="card-body">
                        <h4 class="card-title">
                            <a href="#">' . $row["name"] . '</a>
                        </h4>
                        <h5>₹' . $row["price"] . '</h5>
                        </div>
                        <div class="card-footer">
                        <input type="submit" class="btn btn-primary btn-sm" name=' . $row["book_id"] . ' value="Reserve"/>
                        </div>
                    </div>
                    </div>';
                    }
                    else{
                        echo '<div class="col-lg-4 col-md-6 mb-4">
                    <div class="card h-100">
                        <a href="#"
                        ><img
                            class="card-img-top"
                            src= ' . $row["imageUrl"] . '
                            alt=""
                        /></a>
                        <div class="card-body">
                        <h4 class="card-title">
                            <a href="#">' . $row["name"] . '</a>
                        </h4>
                        <h5>₹' . $row["price"] . '</h5>
                        </div>
                        <div class="card-footer">
                        <input type="submit" disabled class="btn btn-primary btn-sm disabled" name=' . $row["book_id"] . ' value="Reserve"/>
                        </div>
                    </div>
                    </div>';
                    }
                }
            } else {
                echo "error";
            }

            ?>
        </div>
    </div>
</form>
```

Figure 8.4: PHP code for displaying results for the search query.

7. Analysis and Discussions

The search option on the website is put or implemented on the website built. Here in this example, hush is searched in the search bar and the book “Hush little baby” shows up in the dashboard. The code is shown in the above where the search functionality is applied in the website.

8. Conclusions

We successfully implemented the searching functionality for my msruas library website.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Laboratory 9

Title of the Laboratory Exercise: Functionality implementation

1. Introduction and Purpose of Experiment
2. Aim and Objectives

   Aim: To implement reservation of books functionality into our website.

3. Experimental Procedure
4. Presentation of Results

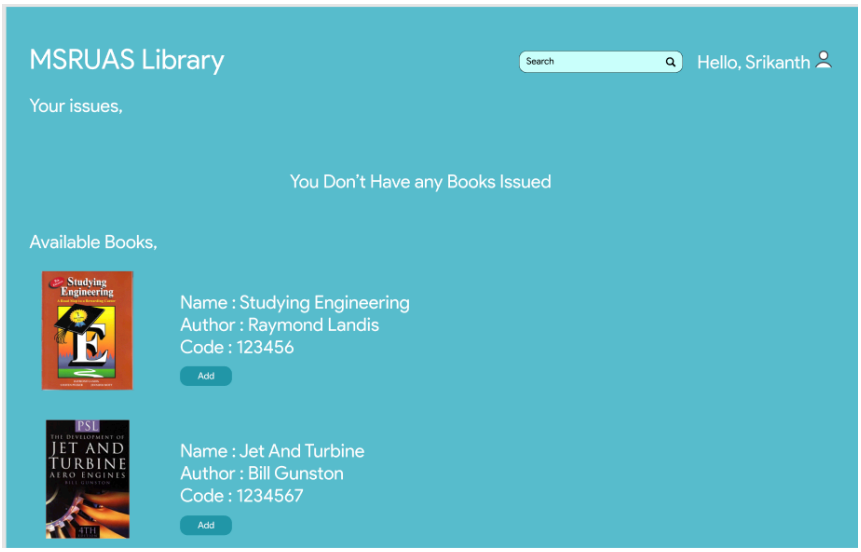


Figure 9.1: The hover on the image gives add to book bag option done in js.

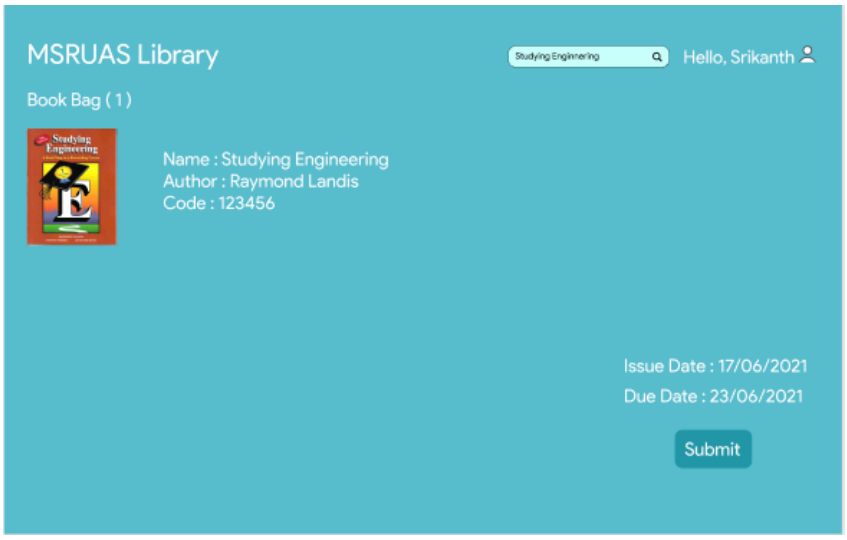


Figure 9.2: Page to reserve the books.

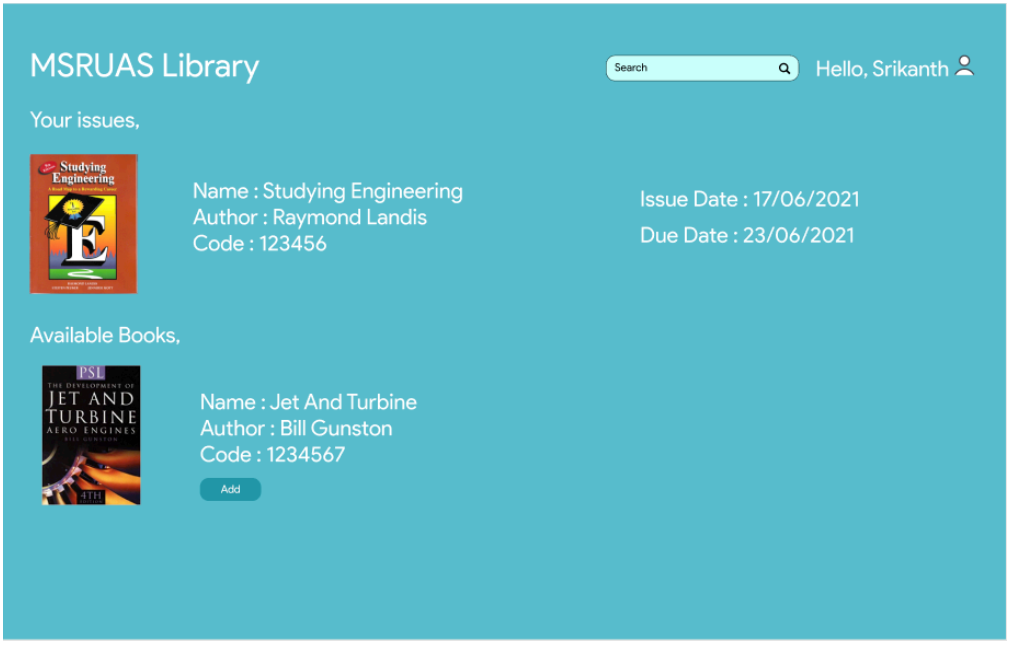


Figure 9.3: Page to reserve the books is issued for the student to collect

```
<form method="post">
    <div class="container">
        <div class="row">
            <?php
            $con = mysqli_connect("localhost", "root", "", "library");

            if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_errno();
            }

            $query = "select * from books";
            $result = mysqli_query($con, $query);

            if (mysqli_num_rows($result) > 0) {
                while ($row = mysqli_fetch_assoc($result)) {

                    if (isset($_POST[$row["book_id"]])) {

                        $con = mysqli_connect("localhost", "root", "", "library");

                        if (mysqli_connect_errno()) {
                            echo "Failed to connect to MySQL: " . mysqli_connect_errno();
                        }

                        $reserve_query = "update `books` set reserved=1 where book_id=".$row['book_id'];

                        $res = mysqli_query($con, $reserve_query);

                        if ($res) {
                            // print("<script>alert('success');</script>");
                        } else {
                            echo ("Error: " . $res);
                        }

                    }
                    if($row["reserved"]==0){
                                                    echo '<div class="col-lg-4 col-md-6 mb-4">
                    <div class="card h-100">
                        <a href="#"
                        ><img
                            class="card-img-top"
                            src= ' . $row["imageUrl"] . '
                            alt=""
                        /></a>
                        <div class="card-body">
                        <h4 class="card-title">
                            <a href="#">' . strval($row["name"]) . '</a>
                        </h4>
                        <h5>₹' . $row["price"] . '</h5>
                        </div>
                        <div class="card-footer">
                        <input type="submit" class="btn btn-primary btn-sm" name=' . $row["book_id"] . ' value="Reserve"/>
                        </div>
                    </div>
                    </div>';
                    }
                    else{
                                                    echo '<div class="col-lg-4 col-md-6 mb-4">
                    <div class="card h-100">
                        <a href="#"
                        ><img
                            class="card-img-top"
                            src= ' . $row["imageUrl"] . '
                            alt=""
                        /></a>
                        <div class="card-body">
                        <h4 class="card-title">
                            <a href="#">' . strval($row["name"]) . '</a>
                        </h4>
                        <h5>₹' . $row["price"] . '</h5>
                        </div>
                        <div class="card-footer">
                        <input type="submit" disabled class="btn btn-primary btn-sm disabled" name=' . $row["book_id"] . ' value="Reserve"/>
                        </div>
                    </div>
                    </div>';
                    }

                }
            } else {
                echo "error";
            } ?>
        </div>
    </div>
</form>
```

Figure 9.4: Backend to handle the reservation of the book which is in the cart.

5. Analysis and Discussions

Here in this functionality, the reservation of the books which are added to the cart by clicking in the button and then is shown in the cart. When checkout is clicked then we are taken to a new page to check out or apply discounts before reservation of the book. The code which links the page with its functionality is in the above.

6. Conclusions

We successfully implemented reservation of books functionality for books.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	