



01IT0701 - Advance Web Technology Student Lab Manual – Index

Student Name					
Enrollment No					
Subject (Subject code)	Semester - Branch - Div.	Lab/ Week	Batch		
			A	B	C
AWT (01IT0701)	7th C.E TC3	1 Lab (2 hours)			

Lab	Lab Title	Lab Required	Date	Grade	Signature
1.	a. Creating a registration Page with HTML 5 Validation. b. Adding a external stylesheet to make registration form look elegant.	1			
2.	a. Implement JavaScript Validations to the registration form using Event Listener and Events.	1			
3.	Design web pages for mobile shopping application using HTML5, CSS3 and Bootstrap 5. a. Home Page: list few products & few menus like category, registration, sign in and about us. b. Categories: display few products per each category and provide show more option for complete list. c. User registration web page d. Login page for User e. Detailed product page f. Admin login g. Management of category & product (insert, update and delete product and categories)	1			
4.	a. Implement OOP concepts with PHP for following program: Use Student, Teacher and Subject Class to define relation between each of them. Display the details of the student studying a particular subject under a particular teacher. Use Constructors and Polymorphism.	1			
5.	Using PHP OOP, implement following using design of lab 3: a. Create a database for mobile shopping. b. Implement registration page and login system with session management c. Store details like password, time of login and url in the cookies.	1			
6.	Using design of lab 3: a. Implement a category and product management page with insert, update and delete operations. b. Display products on home page from DB to all users.	1			



01IT0701 - Advance Web Technology Student Lab Manual – Index

	c. If user is logged in then show “more details” on each product and show detailed product description if user clicks “more details.”				
7.	a. Implementing Web Scrapping using CURL to fetch the relevant data from IMDB Movie Website.	1			
8.	a. Implementing Job Application Registration Form Validation using Regular Expressions.	1			
9.	a. Implementing an application to fetch weather details using Weather APIs. b. Implementing an application to fetch cricket/stock details using relevant APIs.	1			
10.	a. Setup Laravel project for Book Inventory. b. Setup the database in Laravel for Book Inventory. c. Implement Select Operation with GUI to fetch details of books from Book Inventory.	1			
11.	a. Implement Insert Operation with GUI to add new details of book in Book Inventory. b. Implement Update Operation with GUI to modify details of any book in Book Inventory.	1			
12.	a. Implement Delete Operation with GUI to remove a book from Book Inventory. b. Implement Search Operation on Book Name and Author Name with GUI for Book Inventory.	1			
13.	a. Setup Node Js project on Student Information System. b. Setup the database in Node Js for Student Information System. c. Implement Select Operation to fetch details of students from Student Information System.	1			
14.	a. Implement Insert Operation to add new record of student in Student Information System. b. Implement Update Operation to modify any details of student in Student Information System. c. Implement Delete Operation to remove any student from Student Information System.	1			

Note:

1. All Students Will perform Practical as per the Given Lab Manual Only.
2. All Students need to use this Index only.
3. For Lab File: Mandatory Follow AWT Subject Lab Print Format. All the Students need to take Hard Copy of Practical in Given Format only.

Prof. Munindra Lunagaria
Subject In-Charge (AWT)
Computer Engineering,
MU

Prof. Hardik Doshi
Head of Department,
Computer Engineering,
MU

Lab#1(a) Creating a registration Page with HTML 5 Validation

TAGS USED	ATTRIBUTES USED	DESCRIPTION
<HTML>	--	Write the description of HTML tags and attributes
<HEAD>	--	Write the title and meta tags here
<META>	CHARSET,NAME	Attribute of metadata
<TITLE>	BGCOLOR,TEXT, BACKGROUND	Write title of your website
<STYLE >	ALIGN	
<BODY >	FACE,SIZE,COLOR	
<FORM>	ACTION,METHOD	Attributes for user input and data submission
<LABEL>	FOR	Attribute of the corresponding input element
<TD>	COLSPAN,ALIGN	Define cell in html table
<TR>	--	Define row in table

Final Outcome:

First Name:	<input type="text"/>	
Last Name:	<input type="text"/>	
Email	<input type="text"/>	
Phone	<input type="text"/>	
City	<input type="text"/>	
Age	<input type="text"/>	
Profession	<input type="text"/>	
Password	<input type="text"/>	
Confirm Password	<input type="text"/>	
Gender	<input type="radio"/> Male	<input checked="" type="radio"/> Female
<input type="button" value="Submit"/>		

Code Snippet:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registration Form</title>
  <style>
    body {
      background-color: #333;
      color: white;
    }
    table {
      margin: 0 auto;
      border-collapse: collapse;
    }
    td {
      padding: 8px;
    }
    input[type="text"],
    input[type="email"],
    input[type="tel"],
    input[type="password"] {
      width: 98%;
    }
  </style>
</head>
<body>
  <form action="#">
    <table border="1">
      <tr>
        <td>First Name:</td>
        <td><input type="text" name="first_name"></td>
      <td>Last Name:</td>
        <td><input type="text" name="last_name"></td>
      </tr>
      <tr>
        <td>Email</td>
        <td colspan="3"><input type="email" name="email"></td>
      </tr>
      <tr>
        <td>Phone</td>
        <td colspan="3"><input type="tel" name="phone"></td>
      </tr>
      <tr>
        <td>City</td>
```

Lab #1a

```
<td colspan="3"><input type="text" name="city"></td>
</tr>
<tr>
    <td>Age</td>
    <td colspan="3"><input type="text" name="age"></td>
</tr>
<tr>
    <td>Profession</td>
    <td colspan="3"><input type="text" name="profession"></td>
</tr>
<tr>
    <td>Password</td>
    <td colspan="3"><input type="password" name="password"></td>
</tr>
<tr>
    <td>Confirm Password</td>
    <td colspan="3"><input type="password" name="confirm_password"></td>
</tr>
<tr>
    <td>Gender</td>
    <td><input type="radio" name="gender" value="male"> Male</td>
    <td colspan="2"><input type="radio" name="gender" value="female"> Female</td>
</tr>
<tr>
    <td align="center" colspan="4"><input type="submit" value="Submit"></td>
</tr>
</table>
</form>
</body>
</html>
```

Lab #1b

Lab#1(b) Adding a external stylesheet to make registration form look elegant.

TAGS USED	ATTRIBUTES USED	DESCRIPTION
<HTML>	--	Write the description of HTML tags and attributes
<HEAD>	--	Write the title and meta tags here
<META>	CHARSET,NAME	its attributes for metadata.
<TITLE>	--	Write the title your website
<STYLE>	--	
<BODY>	--	
<FORM>	ACTION,METHOD	attributes for user input and data submission.
<LABEL>	FOR	attribute of the corresponding input element
<INPUT>	TYPE,TEXT,EMAIL,TEL,PASSWORD,RESET,SUBMIT	

Final Outcome:

First Name:	<input type="text"/>	Last Name:	<input type="text"/>
Email	<input type="text"/>		
Phone	<input type="text"/>		
City	<input type="text"/>		
Age	<input type="text"/>		
Profession	<input type="text"/>		
Password	<input type="text"/>		
Confirm Password	<input type="text"/>		
Gender	<input type="radio"/> Male	<input type="radio"/> Female	
<input type="button" value="Submit"/>			

Lab #1b

Code Snippet:

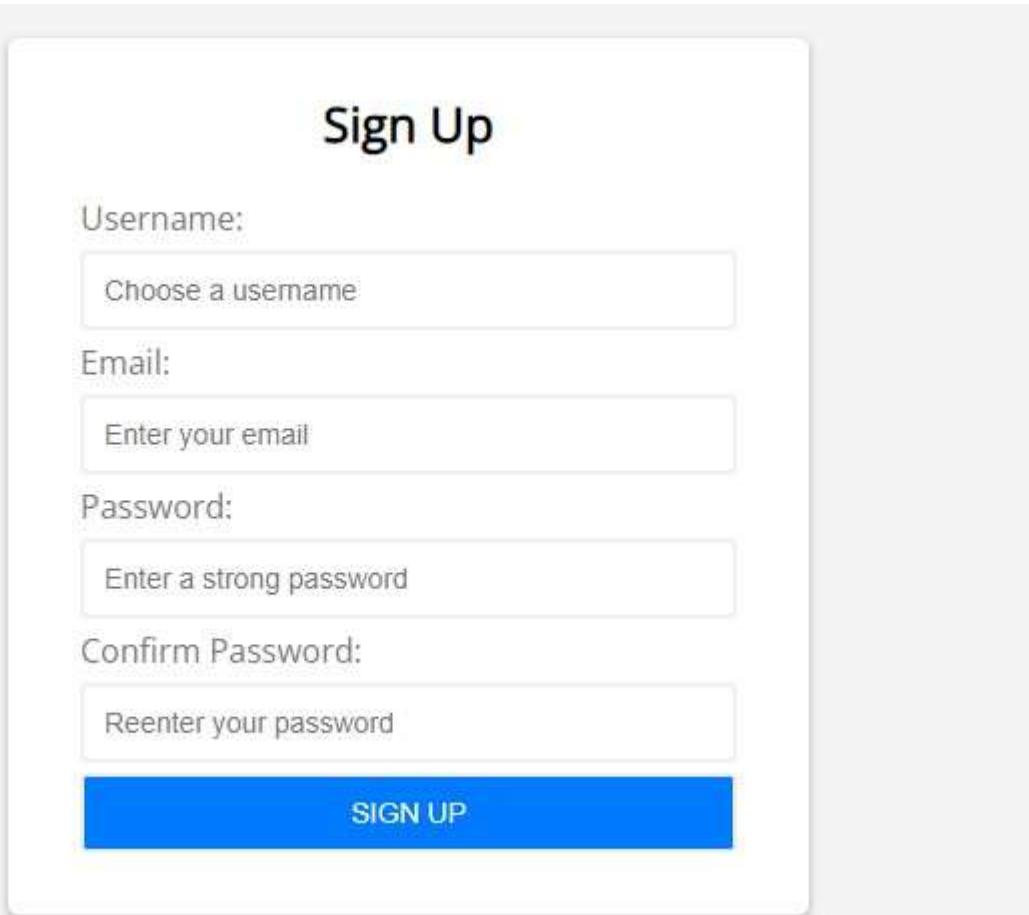
```
body, h1, h2, p {  
    margin: 0;  
    padding: 0;  
}  
  
body {  
    background: linear-gradient(45deg, #4CAF50, #2196F3);  
    font-family: Arial, sans-serif;  
}  
  
table {  
    margin: 0 auto;  
}  
  
td {  
    padding: 8px;  
    border-radius: 8px;  
}  
  
input[type="text"],  
input[type="email"],  
input[type="tel"],  
input[type="password"] {  
    width: 98%;  
    padding: 10px;  
    border: none;  
    border-radius: 4px;  
    margin-bottom: 10px;  
}  
  
input[type="submit"] {  
    background-color: #4285f4;  
    color: white;  
    border: none;  
    border-radius: 4px;  
    padding: 10px 20px;  
    cursor: pointer;  
}  
  
input[type="submit"]:hover {  
    background-color: #3578e5;  
}
```

Lab #2(a)

Lab#2(a) Implement JavaScript Validations to the registration form using Event Listener and Events.

TAGS USED	ATTRIBUTES USED	DESCRIPTION
<SCRIPT>		Denotes html code
<FUNCTION>		Initialize constant variable
<CONST>		To select input fields
<METHOD>	Username, email, password, confirm password and sign up	Check the field value
Regular expression		Check the field value

Final Outcome:



Sign Up

Username:

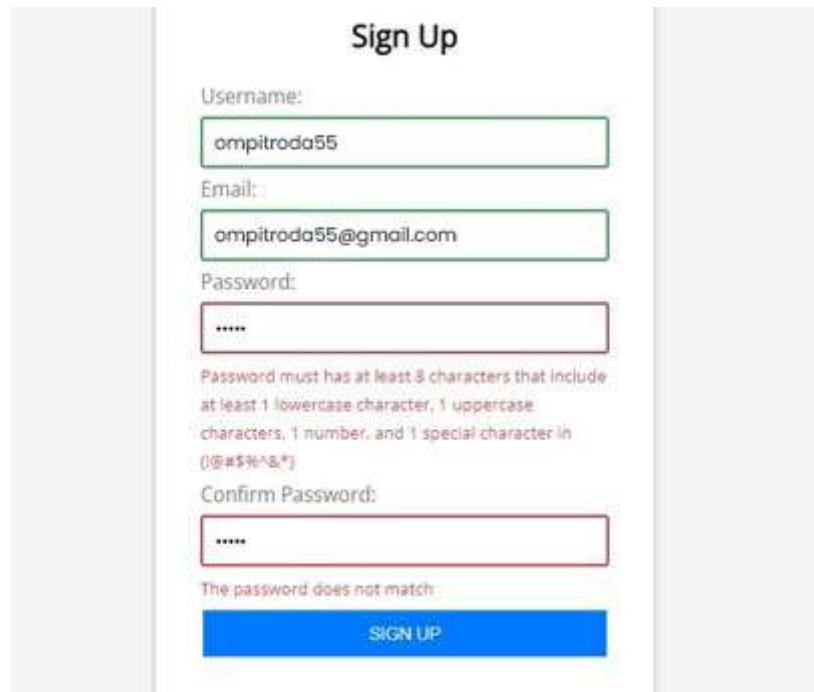
Email:

Password:

Confirm Password:

SIGN UP

Lab #2(a)



Sign Up

Username:
ompitroda55

Email:
ompitroda55@gmail.com

Password:
.....

Confirm Password:
.....

The password does not match.

SIGN UP

Code Snippet:

```

<script>
const usernameEl = document.querySelector('#username'); const emailEl =
document.querySelector('#email'); const passwordEl = document.querySelector('#password');
const confirmPasswordEl = document.querySelector('#confirm-password');

const form = document.querySelector('#signup');

const checkUsername = () => {

    let valid = false;

    const min = 3,      max = 25;

    const username = usernameEl.value.trim();

    if (!isRequired(username)) { showError(usernameEl, 'Username cannot be blank.'); } else if
    (!isBetween(username.length, min, max)) {
        showError(usernameEl, `Username must be between ${min} and ${max} characters.`)
    } else {
        showSuccess(usernameEl);
        valid = true;
    }
    return valid;
}

```

Lab #2(a)

};

```
const checkEmail = () => {    let valid = false;
  const email = emailEl.value.trim();    if (!isRequired(email)) {
    showError(emailEl, 'Email cannot be blank.');
  } else if (!isValidEmail(email)) {
    showError(emailEl, 'Email is not valid.')
  } else {
    showSuccess(emailEl);      valid = true;
  }
  return valid;
};
```

let valid = false;

```
const password = passwordEl.value.trim();

if (!isRequired(password)) {
  showError(passwordEl, 'Password cannot be blank.');
} else if (!isPasswordSecure(password)) {
  showError(passwordEl, 'Password must has at least 8 characters that include at least 1 lowercase character, 1 uppercase characters, 1 number, and 1 special character in (!@#$%^&*)');
} else {
  showSuccess(passwordEl);
  valid = true;
}
```

```
return valid;
```

};

```
const checkConfirmPassword = () => {
  let valid = false;

  const confirmPassword = confirmPasswordEl.value.trim();  const password =
passwordEl.value.trim();
```

```
if (!isRequired(confirmPassword)) {
  showError(confirmPasswordEl, 'Please enter the password again');
} else if (password !== confirmPassword) {
  showError(confirmPasswordEl, 'The password does not match');
} else {
  showSuccess(confirmPasswordEl);      valid = true;
}
```

return valid;

Lab #2(a)

```

};

const isEmailValid = (email) => {
    const re = /^[^<>0\[\]\.,;\s@"]+(\[^<>0\[\]\.,;\s@"]+)*|(.+))@((\[[0-9]{1,3}\].[0-
9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-\-0-9]+\.)+[a-zA-Z]{2,}))$/;  return re.test(email);
};

const isPasswordSecure = (password) => {
    const re = new RegExp("^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[@#\$\%^\&\*])(?=.{8,})");
    return re.test(password);
};

constisRequired = value => value === "" ? false : true;
const isBetween = (length, min, max) => length < min || length > max ? false : true;

const showError = (input, message) => {
    const formField = input.parentElement;
    formField.classList.remove('success');
    formField.classList.add('error');

    const error = formField.querySelector('small');  error.textContent = message;
};

const showSuccess = (input) => {
    const formField = input.parentElement;
    formField.classList.remove('error');  formField.classList.add('success');

    const error = formField.querySelector('small');
    error.textContent = "";
}

form.addEventListener('submit', function (e) {
    e.preventDefault();

    let isUsernameValid = checkUsername(),      isEmailValid = checkEmail(),      isPasswordValid =
    checkPassword(),
    confirmPasswordValid = checkConfirmPassword();
}

```

Lab #2(a)

```

let isFormValid = isUsernameValid &&
    isEmailValid &&      isPasswordValid &&      isConfirmPasswordValid;

if (isFormValid) {
}

});

const debounce = (fn, delay = 500) => {
let timeoutId;
return (...args) => {

    if (timeoutId) {
        clearTimeout(timeoutId);
    }

    timeoutId = setTimeout(() => {      fn.apply(null, args)
    }, delay);
};

};

form.addEventListener('input', debounce(function (e) {
    switch (e.target.id) {      case 'username':      checkUsername();      break;      case 'email':
checkEmail();      break;      case 'password':      checkPassword();      break;      case
'confirm-password':      checkConfirmPassword();
        break;
    }
}));
</script>

```

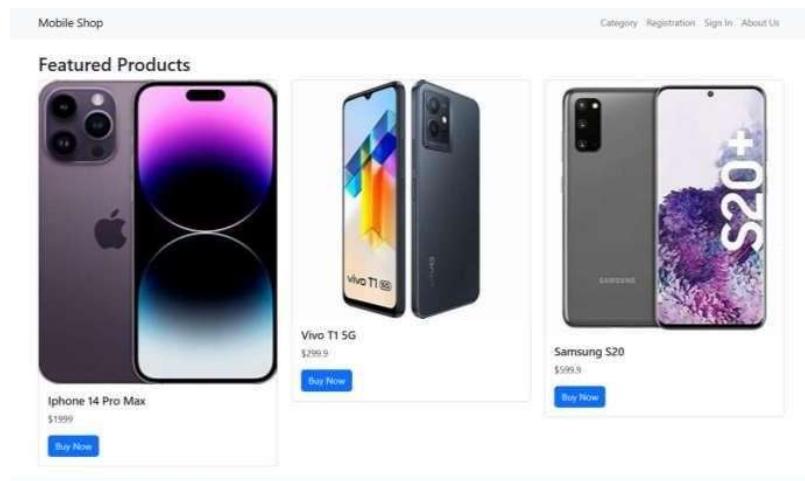
Lab #3

Lab#3(a) Home Page: list few products & few menus like category, registration, sign in and about us.

TAGS USED	ATTRIBUTES USED	DESCRIPTION
<HTML>	lang="en"	Write the description of HTML tags and attributes
<HEAD>	-	Contains meta-information about the document, such as title and CSS scripts.
<TITLE>	-	Defines the title of the document that appears in the browser's title bar.
<LINK>	rel, href	Links external resources, like CSS stylesheets, to the HTML document.
<BODY>	-	Contains the visible content of the document.
<NAV>	-	Defines a container for navigation links.
<DIV>	class	A generic container element for styling and layout purposes.
<A>	class, href	Creates hyperlinks and defines the URL they link to.
<BUTTON>	class, type, onclick	Defines a clickable button.
	class	Represents an unordered list.
	-	Represents an item in a list (or).
<H2>	-	Defines a level 2 heading.
<H5>	-	Defines a level 5 heading.
<P>	-	Defines a paragraph of text.
	src, alt, class	Embeds an image into the document with a specified source and alternate text.

Lab #3

Final Outcome:



Lab #3

Code Snippet:

```

<html>
<head>
<title>Mobile Shopping App - Home</title>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
<!-- Custom CSS -->
<link rel="stylesheet" href="styles.css">
</head>
<body>
<!-- Navbar -->
<nav class="navbar navbar-expand-md navbar-light bg-light">
<div class="container">
<!-- Your logo/image -->
<a class="navbar-brand" href="#">Mobile Shop</a>

<!-- Navbar toggler for mobile -->
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
<ul class="navbar-nav ms-auto">
<!-- Menu options -->
<li class="nav-item">
<a class="nav-link" href="P1B.html">Category</a>
</li>
<li class="nav-item">
<a class="nav-link" href="P1C.html">Registration</a>
</li>
<li class="nav-item">
<a class="nav-link" href="P1D.html ">Sign In</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">About Us</a>
</li>
</ul>
</div>
</div>
</nav>
<!-- Main content -->
<div class="container mt-4">
```

Lab #3

```

<h2>Featured Products</h2>
<div class="row"> <!-- Product 1 -->
<div class="col-md-4">
<div class="card">

<div class="card-body">
<h5 class="card-title">Iphone 14 Pro Max</h5>
<p class="card-text">$1999</p>
<a href="#" class="btn btn-primary">Buy Now</a>
</div>
</div>
</div>
<!-- Product 2 -->
<div class="col-md-4">
<div class="card">

<div class="card-body">
<h5 class="card-title">Vivo T1 5G</h5>
<p class="card-text">$299.9</p>
<a href="#" class="btn btn-primary">Buy Now</a>
</div>
</div>
</div>
<!-- Product 3 -->
<div class="col-md-4">
<div class="card">

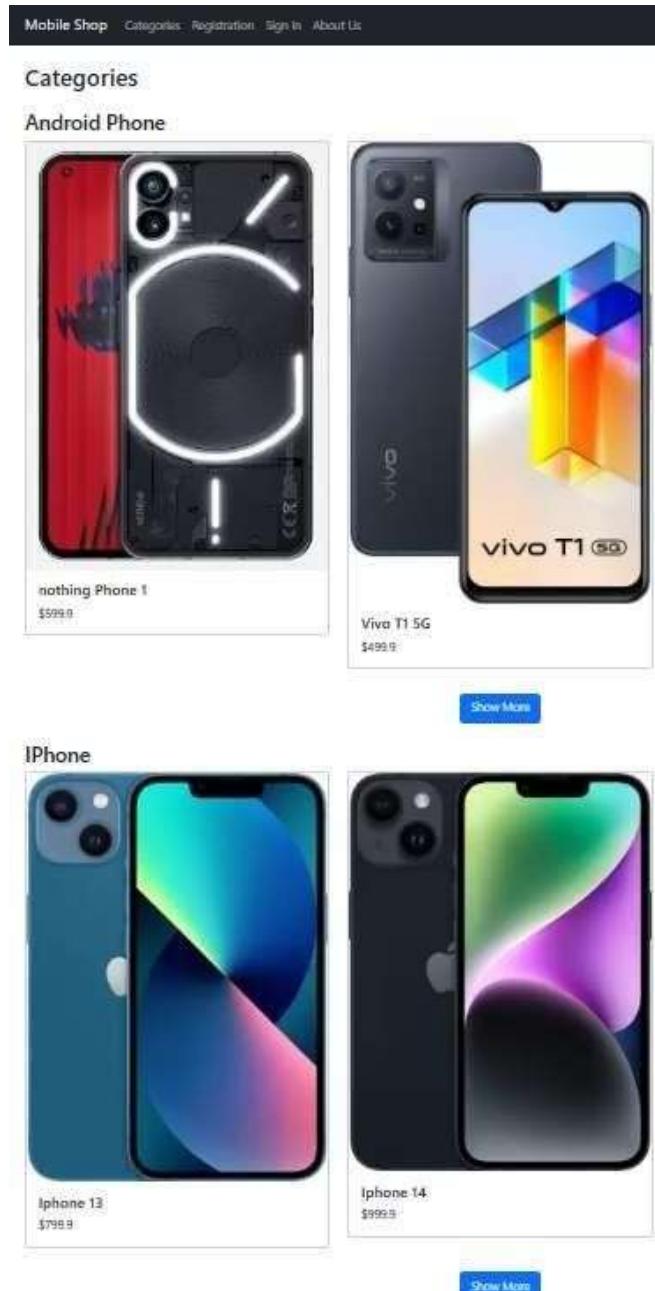
<div class="card-body">
<h5 class="card-title">Samsung S20</h5>
<p class="card-text">$599.9</p>
<a href="#" class="btn btn-primary">Buy Now</a>
</div>
</div>
</div>
</div> </div>
<!-- Include Bootstrap JS at the end of the body -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

Lab #3

Lab#3(b) Categories: display few products per each category and provide show more option for complete list.

Final Outcome:



Code Snippet:

```
<html>
```

```
<head>
```

Lab #3

```

<title>Mobile Shopping App - Categories</title>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
> <!-- Custom CSS -->

<style>
  /* Add any custom styles here */
</style>

</head>

<body>
  <!-- Navbar -->
  <nav class="navbar navbar-expand-md navbar-dark bg-dark">
    <div class="container">
      <a class="navbar-brand" href="#">Mobile Shop</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <a class="nav-link" href="#categories">Categories</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#registration">Registration</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#signin">Sign In</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

```

Lab #3

```
</li>
<li class="nav-item">
<a class="nav-link" href="#aboutus">About Us</a>
</li>
</ul>
</div>
</div>
</nav>
<!-- Main content --&gt;
&lt;div class="container mt-4" id="categories"&gt;
&lt;h2 class="mb-4"&gt;Categories&lt;/h2&gt;
&lt;!-- Category 1 --&gt;
&lt;div class="category"&gt;
&lt;h3&gt;Android Phone&lt;/h3&gt;
&lt;div class="row"&gt;
&lt;div class="col-md-4 mb-4"&gt;
&lt;div class="card"&gt;
&lt;img src="category1_product1.jpg" class="card-img-top" alt="Category 1 - Product 1"&gt;
&lt;div class="card-body"&gt;
&lt;h5 class="card-title"&gt;nothing Phone 1&lt;/h5&gt;
&lt;p class="card-text"&gt;$599.9&lt;/p&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;div class="col-md-4 mb-4"&gt;
&lt;div class="card"&gt;
&lt;img src="category1_product2.jpg" class="card-img-top" alt="Category 1 - Product 2"&gt;
&lt;div class="card-body"&gt;</pre>
```

Lab #3

```
<h5 class="card-title">Vivo T1 5G</h5>
<p class="card-text">$499.9</p>
</div>
</div>
</div>
<!-- Add more product cards for Category 1 here -->
</div>
<div class="text-center mt-2">
  <button class="btn btn-primary">Show More</button>
  <!-- Implement JavaScript to load more products for Category 1 -->
</div>
</div>
<!-- Category 2 -->
<div class="category mt-4">
  <h3>IPhone</h3>
  <div class="row">
    <div class="col-md-4 mb-4">
      <div class="card">
        
        <div class="card-body">
          <h5 class="card-title">Iphone 13</h5>
          <p class="card-text">$799.9</p>
        </div>
      </div>
    </div>
    <div class="col-md-4 mb-4">
      <div class="card">
        
```

Lab #3

```
<div class="card-body">  
    <h5 class="card-title">Iphone 14</h5>  
    <p class="card-text">$999.9</p>  
</div>  
</div>  
</div>  
<!-- Add more product cards for Category 2 here -->  
</div>  
<div class="text-center mt-2">  
    <button class="btn btn-primary">Show More</button>  
    <!-- Implement JavaScript to load more products for Category 2 -->  
</div>  
</div>  
<!-- Add more categories with "Show More" buttons as needed -->  
</div>  
<!-- Include Bootstrap JS at the end of the body -->  
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>  
</body>  
</html>
```

Lab #3

Lab#3(C) Implement Design web pages for mobile shopping application using HTML5, CSS3 and Bootstrap 5. c. User registration web page Final Outcome:



User Registration

Full Name

Email address

Password

Confirm Password

I agree to the [Terms and Conditions](#)

Register

Code Snippet:

```

<html>
<head>
    <title>Mobile Shopping App - User Registration</title>
    <!-- Include Bootstrap CSS -->
    <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
    <!-- Custom CSS -->
    <style> body
        {
            font-family: Arial, sans-serif;
        }
        form { max-width:
            400px; margin: 0
            auto;
        }
    </style>

```

Lab #3

```
label { display: block;  
margin-bottom:  
8px;  
}  
  
input[type="text"],  
input[type="email"],  
input[type="password"] {  
width: 100%; padding:  
10px; margin-bottom:  
20px;  
}  
  
input[type="submit"] {  
background-color: #4CAF50;  
color: white;  
padding: 10px 20px;  
border: none;  
cursor:  
pointer;  
}  
  
</style>  
</head>  
<body>  
<!-- Navbar --&gt;<br/><nav class="navbar navbar-expand-md navbar-dark bg-dark">  
 <div class="container">  
   <a class="navbar-brand" href="#">Mobile Shop</a>
```

Lab #3

```

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation"> <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item">
      <a class="nav-link" href="#categories">Categories</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#registration">Registration</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#signin">Sign In</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#aboutus">About Us</a>
    </li>
  </ul>
</div>
</div>
</nav>
<!-- Main content -->
<div class="container mt-4">
  <h2>User Registration</h2>
  <form>
    <div class="form-group">
      <label for="fullname">Full Name</label>

```

Lab #3

```

<input type="text" class="form-control" id="fullname" placeholder="Enter your full name"
required>

</div>

<div class="form-group">
    <label for="email">Email address</label>
    <input type="email" class="form-control" id="email" placeholder="Enter your email
address" required>
</div>

<div class="form-group">
    <label for="password">Password</label>
    <input type="password" class="form-control" id="password" placeholder="Enter your
password" required>
</div>

<div class="form-group">
    <label for="confirmPassword">Confirm Password</label>
    <input type="password" class="form-control" id="confirmPassword"
placeholder="Confirm your password"
required>
</div>

<div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="termsCheckbox" required>
    <label class="form-check-label" for="termsCheckbox">I agree to the <a href="#">Terms
and Conditions</a></label>
</div>

<button type="submit" class="btn btn-primary">Register</button>
</form>

<!-- Include Bootstrap JS at the end of the body -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>

```

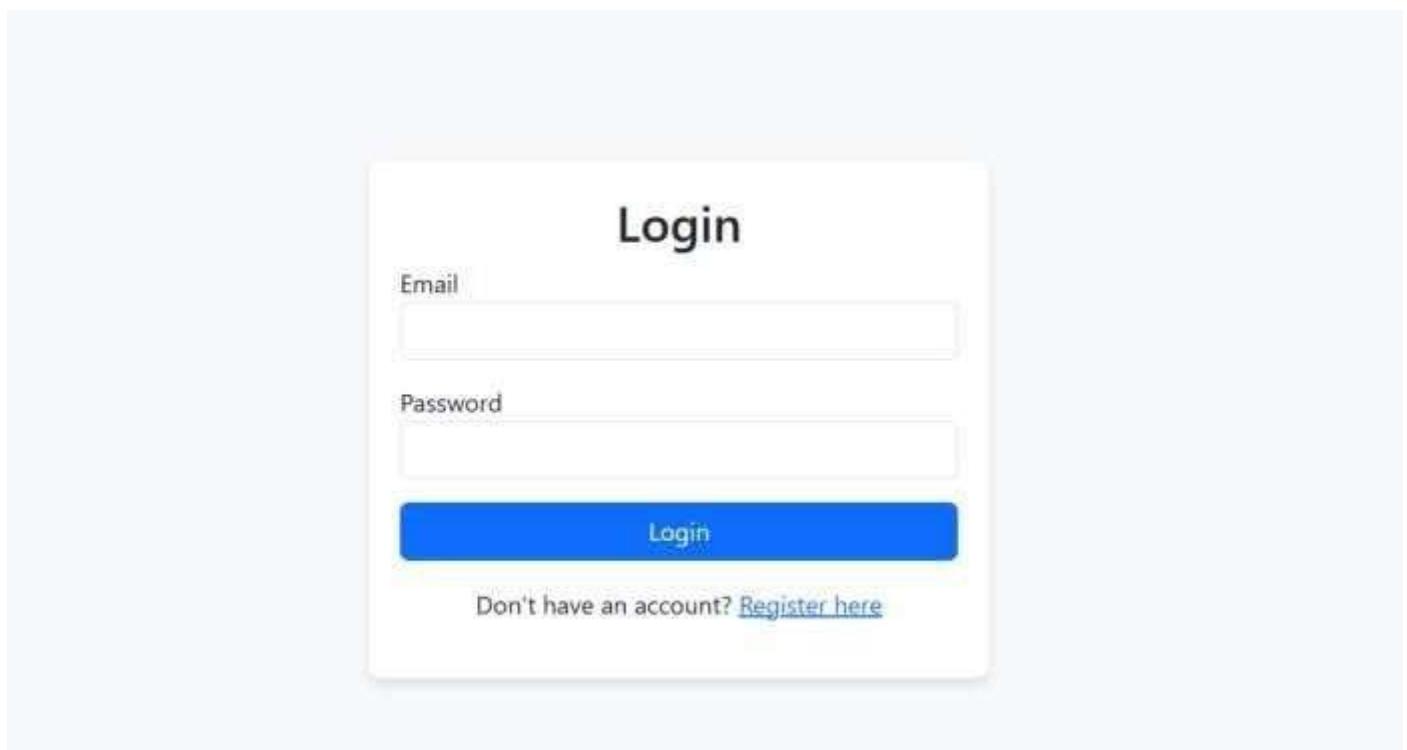
Lab #3

</html>

Lab#3(D) Implement Design web pages for mobile shopping application using HTML5, CSS3 and Bootstrap 5.

d. Login page for User

Final Outcome:



Code Snippet:

```
<html>
<head>
<title>Mobile Shopping App - Login</title>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"> <!--
- Custom CSS -->
<style>
```

Lab #3

```
/* Add any custom styles here */ body
{
    background-color: #f8f9fa;
}

.login-container { max-width: 400px;
    margin: 0 auto; padding: 20px;
    background-color: #fff; box-shadow: 0
    4px 8px rgba(0, 0, 0, 0.1); border-radius:
    8px; margin-top: 100px;
}

.login-container h2 { text-align:
    center;
}

.form-group { margin-bottom:
    15px;
}

.btn-login { width:
    100%;
}

</style>

</head>

<body>

<!-- Main content -->

<div class="container">
    <div class="login-container">
        <h2>Login</h2>
        <form action="#" method="POST">
```

Lab #3

```
<div class="form-group">
    <label for="email">Email</label>
    <input type="email" id="email" name="email" class="form-control" required>
</div>
<div class="form-group">
    <label for="password">Password</label>
    <input type="password" id="password" name="password" class="form-control" required>
</div>
<button type="submit" class="btn btn-primary btn-login">Login</button>
</form>
<p class="mt-3 text-center">Don't have an account? <a href="#">Register here</a></p>
</div>
</div>
<!-- Include Bootstrap JS at the end of the body -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Lab#3(E) Implement Design web pages for mobile shopping application using HTML5, CSS3 and Bootstrap 5.

e. Detailed product page Final

Outcome:

Lab #3



Iphone 14 Pro Max

The iPhone 14 Pro Max: Unparalleled performance, stunning design. Experience the pinnacle of mobile technology.

\$1999

[Add to Cart](#)

Material: Cotton

Color: Blue

Size: Medium

Code Snippet:

```
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Mobile Shopping App - Product Page</title>
<!-- Bootstrap 5 CSS -->
```

Lab #3

```

<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
<!-- Custom CSS -->
<link rel="stylesheet" href="styles.css">
</head>

<body>
<!-- Product Page Content -->
<div class="container">
  <div class="row">
    <div class="col-md-6 offset-md-3">
      
      <h2>Iphone 14 Pro Max</h2>
      <p class="text-muted">The iPhone 14 Pro Max: Unparalleled performance, stunning design. Experience the pinnacle of mobile technology.</p>
      <h3>$1999</h3>
      <div class="rating">
        <!-- You can use a rating system here (e.g., stars or thumbs up/down) -->
      </div>
      <button class="btn btn-primary">Add to Cart</button>
      <hr>
      <!-- Additional Product Information -->
      <ul>
        <li>Material: Cotton</li>
        <li>Color: Blue</li>
        <li>Size: Medium</li>
        <!-- Add more details here as needed -->
      </ul>
    </div>
  </div>
</body>

```

Lab #3

</div>

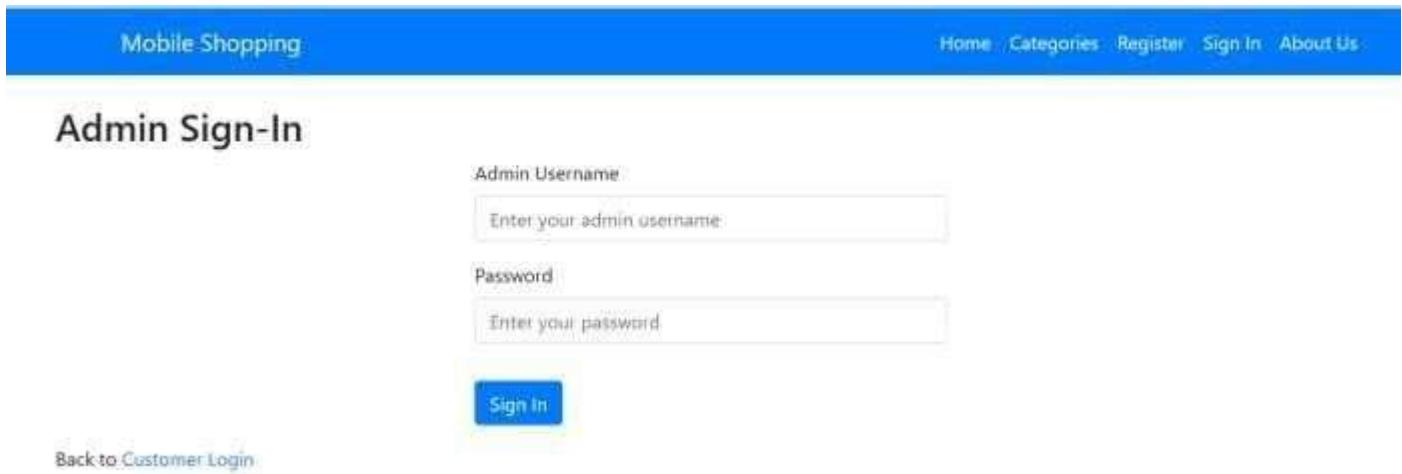
<!-- Bootstrap 5 JS and dependencies -->

```
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js"></script>
</body>
</html>
```

Lab#3(F) Implement Design web pages for mobile shopping application using HTML5, CSS3 and Bootstrap 5.

f. Admin login

Final Outcome:



The screenshot shows a web page titled "Mobile Shopping". At the top right, there is a navigation bar with links: Home, Categories, Register, Sign In, and About Us. Below the navigation bar, the title "Admin Sign-In" is displayed. The form consists of two input fields: "Admin Username" and "Password", each with a placeholder text ("Enter your admin username" and "Enter your password" respectively). Below the password field is a blue "Sign In" button. At the bottom left of the form area, there is a link "Back to Customer Login".

Code Snippet:

```
<html lang="en">
<head>
```

Lab #3

```
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Mobile Shopping App</title>
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
/>
<script src="https://kit.fontawesome.com/463f82e63e.js" crossorigin="anonymous" ></script>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
/>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<style> form {
    max-width: 400px; margin: 0 auto;
}
.navbar-brand { color: #fff !important;
}
.nav-link { color: #fff !important;
}
form {
    max-width: 400px; margin: 0 auto;
}
.redirect{ color: #fff;
}
```

Lab #3

```

</style>
</head>
<body>
<!-- Navbar -->
<nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #007bff
!important" >
<div class="container">

<a class="navbar-brand" href="home.html" style="color: #fff">
<img width="50" /> Mobile Shopping
</a> <button class="navbar-toggler"
type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-
expanded="false"
aria-label="Toggle navigation"
>
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse redirect" id="navbarNav">
<ul class="navbar-nav ms-auto">
<li class="nav-item">
<a class="nav-link" href="home.html">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="categories.html">Categories</a>
</li>
<li class="nav-item">

```

Lab #3

```
<a class="nav-link" href="registration.html">Register</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Sign In</a>
</li>
<li class="nav-item">
<a class="nav-link" href="about_us.html">About Us</a>
</li>
</ul>
</div> </div>
</nav>
<!-- Admin Sign-In Section -->
<section class="container mt-4">
<h2>Admin Sign-In</h2>
<form>
<div class="form-group">
<label for="adminUsername">Admin Username</label>
<input type="text"
class="form-control" id="adminUsername" placeholder="Enter
your admin username" required
/>
</div>
<div class="form-group">
<label for="adminPassword">Password</label>
<input type="password" class="form-control" id="adminPassword" placeholder="Enter
your password" required
/>
</div>
```

Lab #3

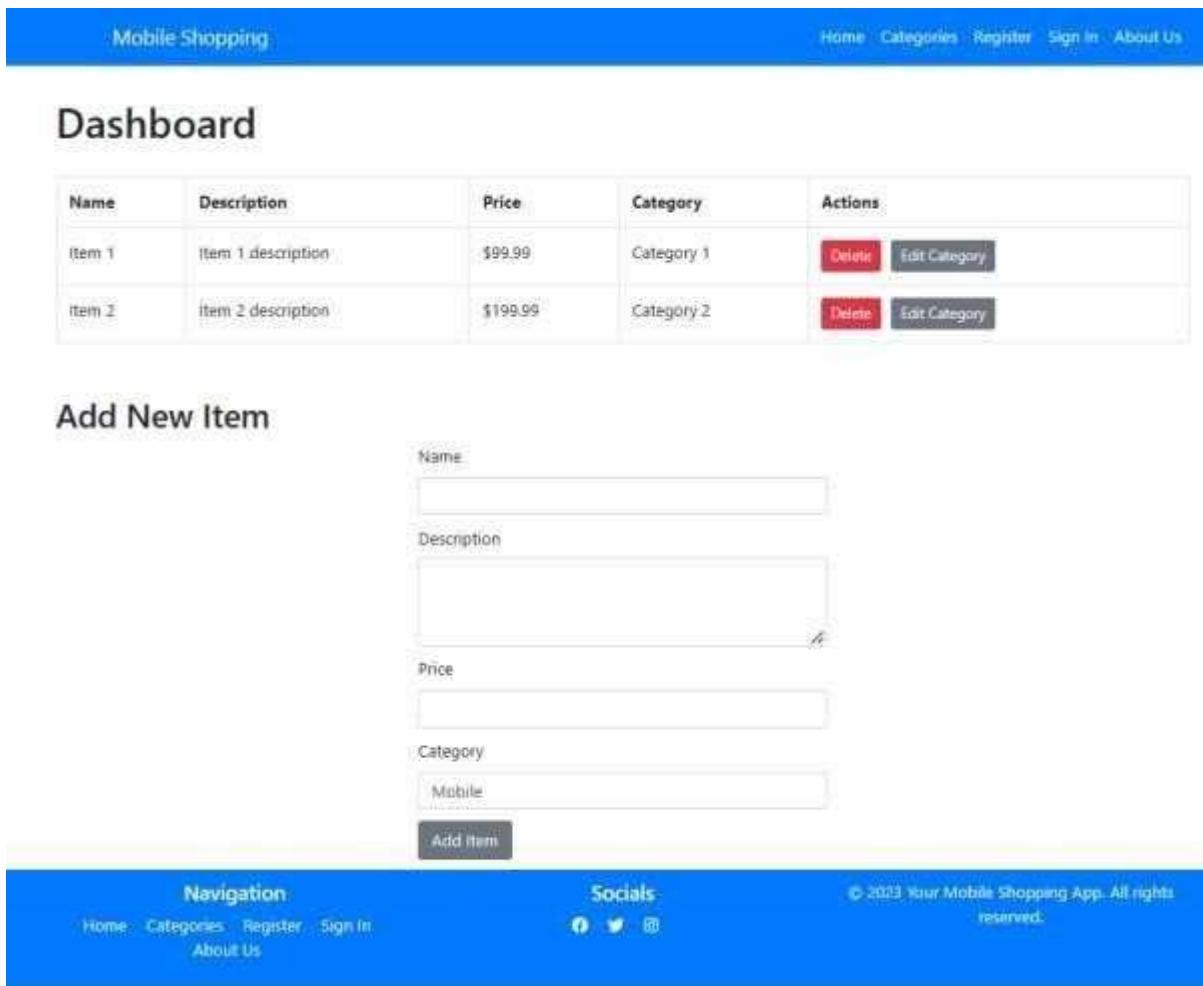
```
<button type="submit" class="btn btn-primary mt-3">
<a class="redirect" href="management_page.html">Sign In</a>
</button>
</form>
<p class="mt-3">Back to <a href="login.html">Customer Login</a></p>
</section>
</body>
</html>
```

Lab #3

Lab#3(G) Implement Design web pages for mobile shopping application using HTML5, CSS3 and Bootstrap 5.

g. Management of category & product (insert, update and delete product and categories)

Final Outcome:



The screenshot displays a mobile shopping application interface. At the top, there is a blue header bar with the text "Mobile Shopping" on the left and navigation links "Home", "Categories", "Register", "Sign In", and "About Us" on the right. Below the header is a section titled "Dashboard" containing a table with two rows of item data. The table columns are "Name", "Description", "Price", "Category", and "Actions". The first row shows "Item 1" with "Item 1 description", "\$99.99", "Category 1", and buttons for "Delete" and "Edit Category". The second row shows "Item 2" with "Item 2 description", "\$199.99", "Category 2", and similar buttons. Below the dashboard is a form titled "Add New Item" with fields for "Name", "Description", "Price", and "Category", each with an input field and a dropdown menu. A "Mobile" option is selected in the category dropdown. At the bottom of the form is a "Add Item" button. The footer of the page is a blue bar with sections for "Navigation" (links to Home, Categories, Register, Sign In, and About Us), "Socials" (links to Facebook, Twitter, and Instagram), and a copyright notice: "© 2023 Your Mobile Shopping App. All rights reserved."

Code Snippet:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

Lab #3

```
<head>
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Mobile Shopping App</title>
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" />
<script src="https://kit.fontawesome.com/463f82e63e.js" crossorigin="anonymous"></script>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" />
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<style> form {
    max-width: 400px; margin: 0 auto;
}
.navbar-brand {
    color: #fff !important;
}
.nav-link {
    color: #fff !important;
} a
{
    text-decoration: none;
```

}

.dashboard-container { margin-top: 30px;

} h1 {

margin-bottom: 30px;

} h2 {

margin-bottom: 10px; margin-top: 50px;

}

.table th {

font-weight: bold;

}

.table-actions { display: flex;

}

.action-button { margin-right: 10px;

}

.add-item-form div { margin: 10px

auto 10px auto;

}

.navigation li { display: inline-block; margin-

right: 15px;

}

.social-icons li { display: inline-block; margin-right: 15px;

}

.footer {

background-color: #007bff; color: #fff; padding: 10px 0;

margin-top: 10px;

}

.footer a { color: #fff;

}

```

</style>
</head>
<body>
<!-- Navbar -->
<nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #007bff
!important">
<div class="container">
<a class="navbar-brand" href="home.html" style="color: #fff">
<img width="50" /> Mobile Shopping
</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation"> <span
class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse redirect" id="navbarNav">
<ul class="navbar-nav ms-auto">
<li class="nav-item">
<a class="nav-link" href="home.html">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="categories.html">Categories</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Register</a>
</li>
<li class="nav-item">
<a class="nav-link" href="login.html">Sign In</a>
</li>

```

```
<li class="nav-item">
<a class="nav-link" href="about_us.html">About Us</a>
</li>
</ul>
</div> </div>
</nav>
<!-- Admin Dashboard -->
<div class="container dashboard-container">
<h1>Dashboard</h1>
<!-- Table to display all items -->
<table class="table table-bordered">
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Price</th>
<th>Category</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<!-- Replace the data placeholders with actual data from the backend -->
<tr>
<td>Item 1</td>
<td>Item 1 description</td>
<td>$99.99</td>
<td>Category 1</td>
<td>
<div class="table-actions">
```

Lab #3

```
<button class="btn btn-sm btn-danger action-button"> Delete
</button>

<button class="btn btn-sm btn-secondary action-button"> Edit Category
</button>
</div>
</td>
</tr>
<tr>
<td>Item 2</td>
<td>Item 2 description</td>
<td>$199.99</td>
<td>Category 2</td>
<td>
<div class="table-actions">
<button class="btn btn-sm btn-danger action-button"> Delete
</button>
<button class="btn btn-sm btn-secondary action-button"> Edit Category
</button>
</div>
</td>
</tr>
<!-- Add more table rows for each item --&gt;
&lt;/tbody&gt;
&lt;/table&gt;
<!-- Buttons to add new items --&gt;
&lt;h2&gt;Add New Item&lt;/h2&gt;
&lt;form class="add-item-form"&gt;
&lt;div class="form-group"&gt;</pre>
```

Lab #3

```
<label for="item-name">Name</label>
<input type="text" class="form-control" id="item-name" name="item-name" required /> </div>
<div class="form-group">
<label for="item-description">Description</label>
<textarea class="form-control" id="item-description" name="item-description" rows="3" required></textarea>
</div>
<div class="form-group">
<label for="item-price">Price</label>
<input type="number" class="form-control" id="item-price" name="item-price" required />
</div>
<div class="form-group">
<label for="item-category">Category</label>
<select id="item-category" name="item-category" class="form-control" required name="">
<option value="">Laptop</option>
<option value="">Mobile</option>
<option value="">Camera</option>
</select>
</div>
<button type="submit" class="btn btn-secondary">Add Item</button>
</form>
</div>
<!-- Footer -->
<footer class="footer text-center">
<div class="container">
<div class="row">
<div class="col-md-4">
<h5>Navigation</h5>
<ul class="list-unstyled navigation">
```

Lab #3

```

<li><a href="home.html">Home</a></li>
<li><a href="categories.html">Categories</a></li>
<li><a href="registration.html">Register</a></li>
<li><a href="login.html">Sign In</a></li>
<li><a href="about_us.html">About Us</a></li>
</ul>
</div>
<div class="col-md-4">
<h5>Socials</h5>
<ul class="list-unstyled social-icons">
<li>
<a href="#"><i class="fa-brands fa-facebook"></i> </a>
</li>
<li>
<a href="#"><i class="fa-brands fa-twitter"></i> </a>
</li>
<li>
<a href="#"><i class="fa-brands fa-instagram"></i> </a>
</li>
</ul>
</div>
<div class="col-md-4">
<p>&copy; 2023 Your Mobile Shopping App. All rights reserved.</p>
<!-- Add your copyright line -->
</div> </div>
</div>
</footer>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
integrity="sha384-

```

geWF76RCwLtnZ8qwWowPQNgul3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
crossorigin="anonymous"></script>

</body>

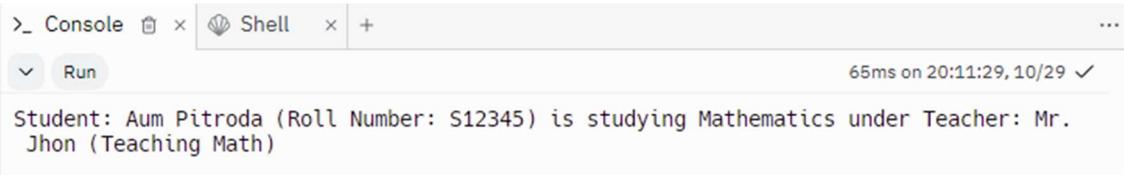
</html>

Lab #4

- Lab#4(a) Implement OOP concepts with PHP for following program:**
Use Student, Teacher and Subject Class to define relation between each of them. Display the details of the student studying a particular subject under a particular teacher. Use Constructors and Polymorphism.

TAGS USED	ATTRIBUTES USED	DESCRIPTION
<?php ?>	--	The opening and closing tags to denote PHP code blocks. All PHP code goes between these tags.
spl_autoload_register	--	Registers a function as an autoloader, which is called when a class is accessed and not yet defined. It is used to automatically load classes as needed.
_DIR	--	A magic constant that returns the directory of the current file.
file_exists	--	Checks if a file or directory exists.
require_once	--	Includes and evaluates a specified file only once during a script's execution. It is used to load class files.
echo	--	Outputs one or more strings. It is used to print content to the output
foreach	--	Iterates over arrays and objects and executes a block of code for each element.
if	--	Executes a block of code if a specified condition is true.
public, private, protected	--	Access modifiers used to define the visibility of class properties and methods.
return	--	Exits a function and returns a value to the caller.

Final Outcome:



```
>_ Console  x  Shell  x  +
▼ Run
65ms on 20:11:29, 10/29 ✓
Student: Aum Pitroda (Roll Number: S12345) is studying Mathematics under Teacher: Mr. Jhon (Teaching Math)
```

Code Snippet:

```
<?php
class
Student {
    private $name; private
    $subjects = [];

    public function construct($name) { $this->name
        = $name;
    }

    public function getName() {
        return $this->name;
    }

    public function addSubject(Subject $subject) { $this->subjects[]
        = $subject;
    }

    public function getSubjects() {
        return $this->subjects;
    }
}
?>
```

Teacher.php:

```
<?php
class
Teacher {
    private $name; private
    $subjects = [];

    public function construct($name) { $this->name
        = $name;
    }

    public function getName() {
```

```

        return $this->name;
    }

    public function addSubject(Subject $subject) { $this->subjects[]
        = $subject;
    }
    public function getSubjects() {
        return $this->subjects;
    }
}
?>
```

Subject.php:

```

<?php
class Subject {
    private $name; private
    $teacher;

    public function __construct($name, Teacher $teacher) {
        $this->name = $name;
        $this->teacher = $teacher;
    }

    public function getName() {
        return $this->name;
    }

    public function getTeacherName() {
        return $this->teacher->getName();
    }
}
?>
```

Index.php:

```

<?php
spl_autoload_register(function ($class) {
    $classPath = _DIR_ . '/classes/' . $class . '.php'; if
    (file_exists($classPath)) {
        require_once $classPath;
    }
});
```

```

$teacher1 = new Teacher("Jiya Manek");
$teacher2 = new Teacher("Kaushal Nathnani");
$teacher3 = new Teacher("Alis Khachar");

$mathSubject = new Subject("Math", $teacher1);
$scienceSubject = new Subject("Science", $teacher2);
$historySubject = new Subject("History", $teacher3); $student1 =
new Student("Ameesha Parmar ");
$student2 = new Student("Jency Patel ");
$student3 = new Student("Ridhi Keraliya ");

// Add subjects to students
$student1->addSubject($mathSubject);
$student1->addSubject($scienceSubject);
$student2->addSubject($scienceSubject);
$student3->addSubject($historySubject);

function displayStudentSubjectDetails(Student $student, $subjectName) {
    $subjects = $student->getSubjects(); foreach
    ($subjects as $subject) {
        if ($subject->getName() === $subjectName) {
            $studentName = $student->getName(); $teacherName =
            $subject->getTeacherName(); echo "Student Name:
            {$studentName}<br>"; echo "Subject Name:
            {$subjectName}<br>"; echo "Teacher Name:
            {$teacherName}<br><br>"; return;
        }
    }
}

displayStudentSubjectDetails($student1, "Math");
displayStudentSubjectDetails($student2, "Science");
displayStudentSubjectDetails($student3, "History");

?>

```

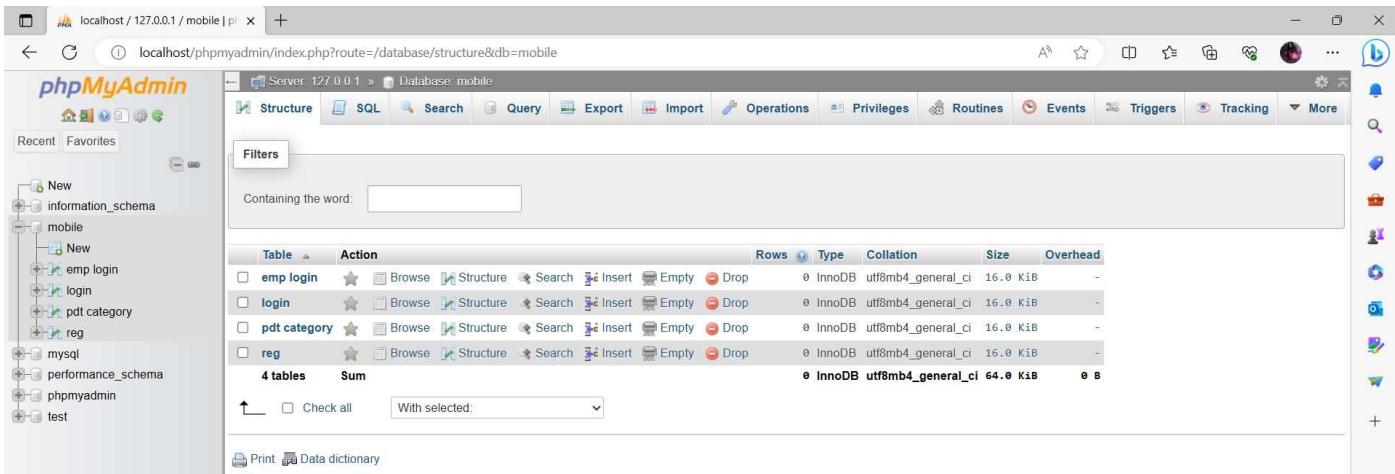
Lab #5(a)
Lab#5(a) Using PHP OOP, implement following using design of lab 3
a. Create a database for mobile shopping.

TAGS USED	ATTRIBUTES USED	DESCRIPTION
DATABASE	mobile	A SQL database is a collection of tables that stores a specific set of structured data.
TABLE	Emp login, login , pdt category, reg	A table in SQL is a structure that stores data in rows and columns
VALUES()	--	VALUES is a table value constructor that specifies a set of row values to be inserted into a table.
PRIMARY KEY	--	A primary key in SQL is a constraint that uniquely identifies each row in a table.

Final Outcome: [Font Cambria, size 14 bold]

Final Outcome:

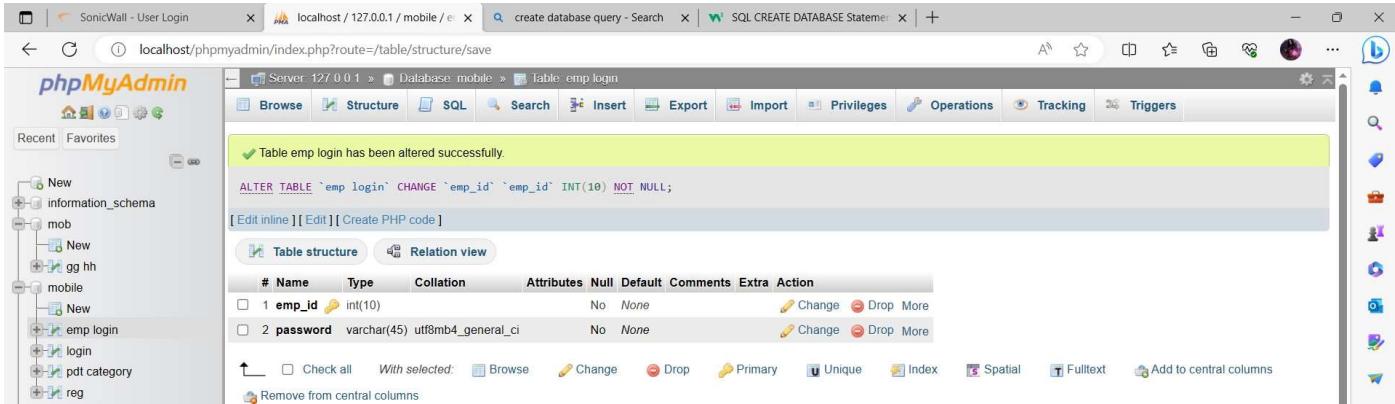
Mobile application Database



Action	Table	Rows	Type	Collation	Size	Overhead
Browse Structure Search Insert Empty Drop	emp login	0	InnoDB	utf8mb4_general_ci	16.0 Kib	-
Browse Structure Search Insert Empty Drop	login	0	InnoDB	utf8mb4_general_ci	16.0 Kib	-
Browse Structure Search Insert Empty Drop	pdt category	0	InnoDB	utf8mb4_general_ci	16.0 Kib	-
Browse Structure Search Insert Empty Drop	reg	0	InnoDB	utf8mb4_general_ci	64.0 Kib	0 B

Lab #5(a)

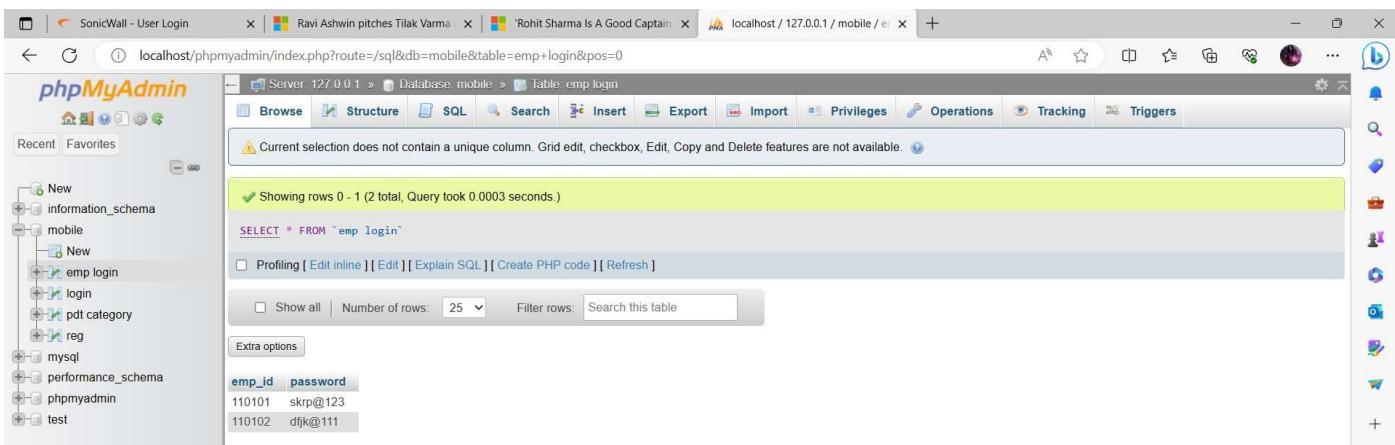
Employee Login structure



The screenshot shows the 'Structure' tab for the 'emp_login' table in the 'mobile' database. The table has two columns: 'emp_id' (int(10)) and 'password' (varchar(45)). The 'emp_id' column is marked as 'NOT NULL'. A message at the top indicates that the table was altered successfully.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	emp_id	int(10)	utf8mb4_general_ci		No	None			Change Drop More
2	password	varchar(45)	utf8mb4_general_ci		No	None			Change Drop More

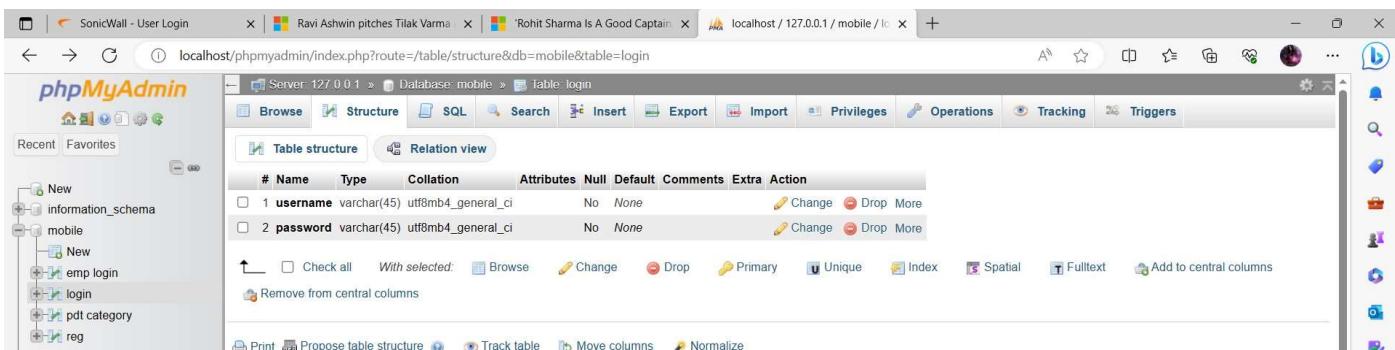
Values in Employee Login



The screenshot shows the 'Data' tab for the 'emp_login' table. It displays two rows of data:

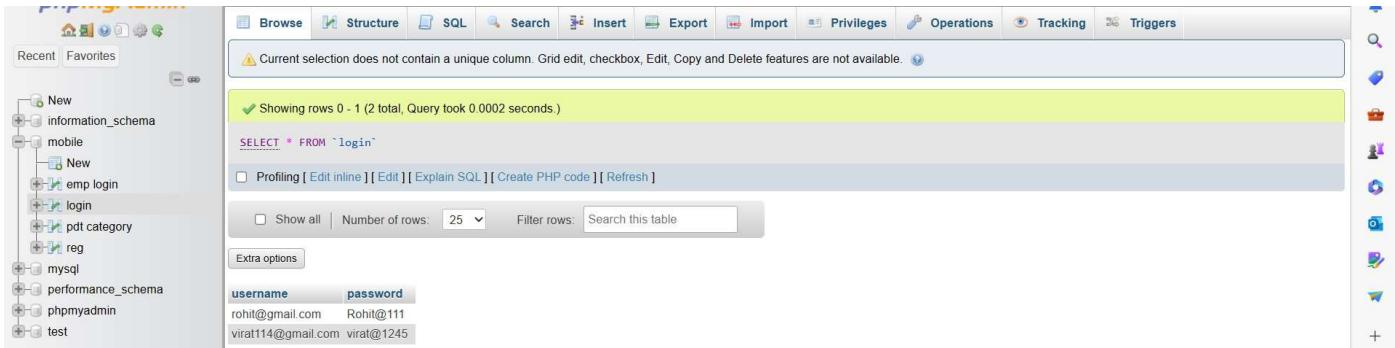
emp_id	password
110101	skrp@123
110102	dfjk@111

Login Structure



The screenshot shows the 'Structure' tab for the 'login' table in the 'mobile' database. The table has two columns: 'username' (varchar(45)) and 'password' (varchar(45)). Both columns have a collation of 'utf8mb4_general_ci'.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	username	varchar(45)	utf8mb4_general_ci		No	None			Change Drop More
2	password	varchar(45)	utf8mb4_general_ci		No	None			Change Drop More

Lab #5(a)
Values in Login


Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
SELECT * FROM `login`
```

Profile [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

username	password
rohit@gmail.com	Rohit@111
virat114@gmail.com	virat@1245

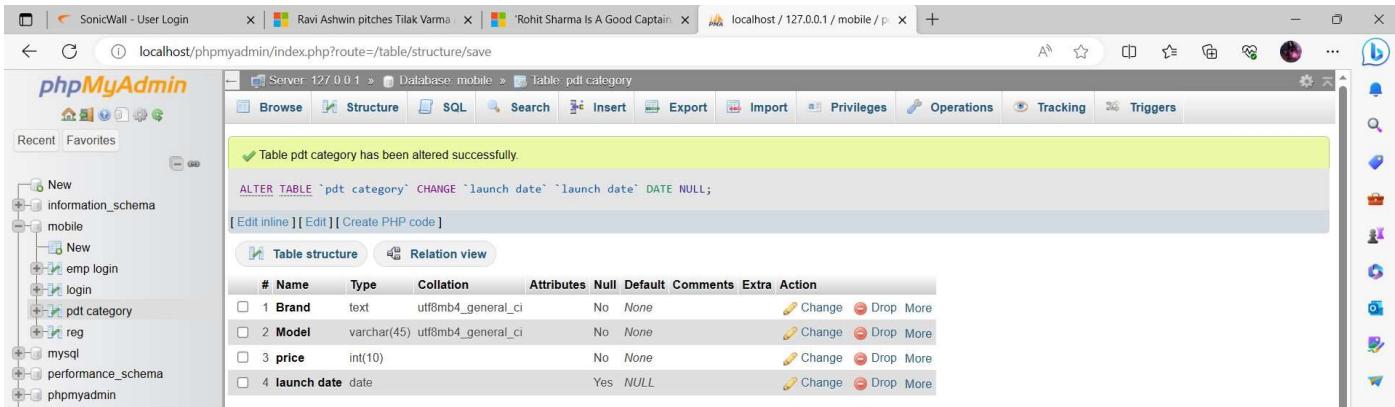
Product Category Structure


Table pdt category has been altered successfully.

```
ALTER TABLE `pdt category` CHANGE `launch date` `launch date` DATE NULL;
```

Table structure [Relation view]

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Brand	text	utf8mb4_general_ci		No	None			Change Drop More
2	Model	varchar(45)	utf8mb4_general_ci		No	None			Change Drop More
3	price	int(10)			No	None			Change Drop More
4	launch date	date			Yes	NULL			Change Drop More

Values in product Category


Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM `pdt category`
```

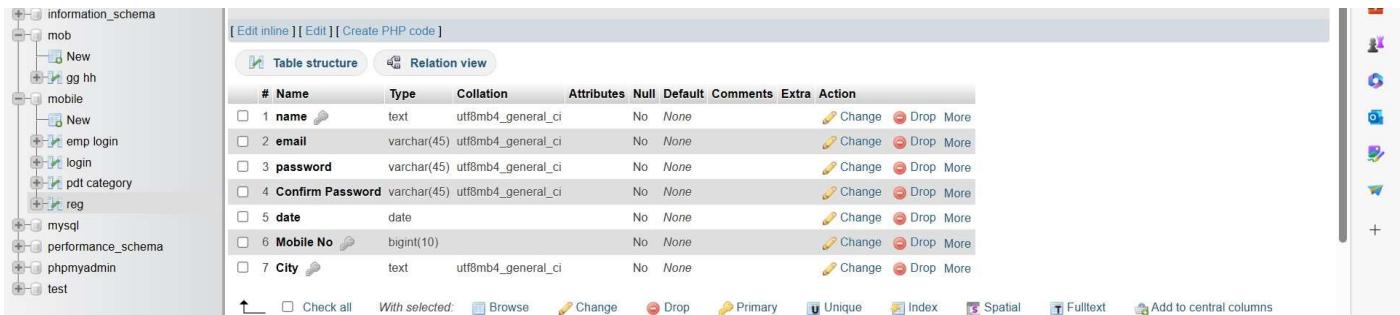
Profile [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Brand	Model	price	launch date
Samsung	Samsung S22	82000	2022-02-25
Nothing	Nothing 2.0	45000	NULL
Nothing	Nothing 1.0	29000	NULL

Lab #5(a)

Registration Structure

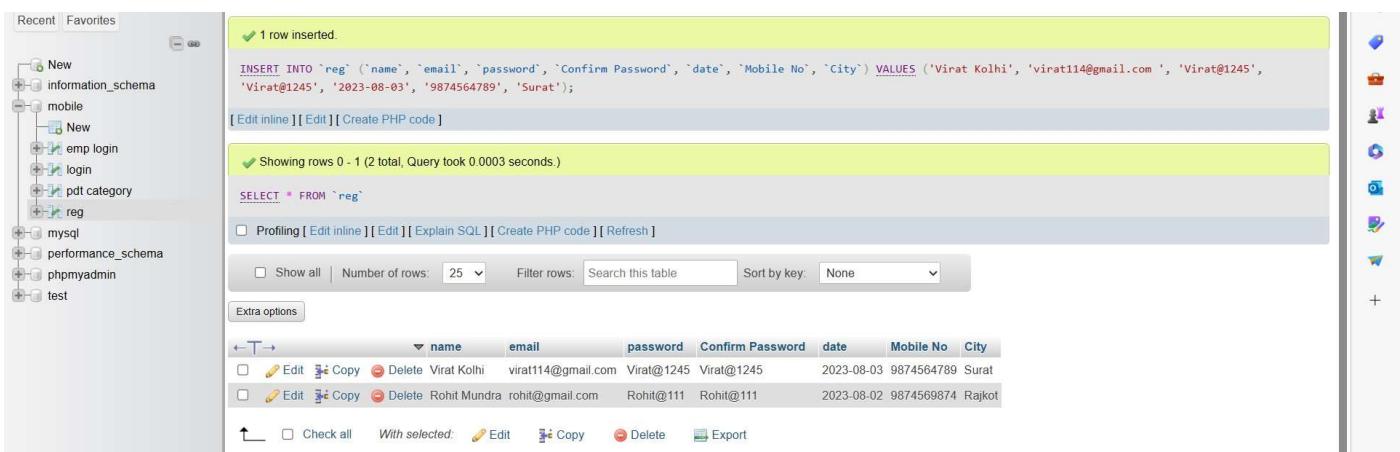


The screenshot shows the MySQL Workbench interface. On the left, the database schema is visible with the 'reg' table selected. The main pane displays the 'Table structure' of the 'reg' table. The columns are:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	name	text	utf8mb4_general_ci		No	None			Change Drop More
2	email	varchar(45)	utf8mb4_general_ci		No	None			Change Drop More
3	password	varchar(45)	utf8mb4_general_ci		No	None			Change Drop More
4	Confirm Password	varchar(45)	utf8mb4_general_ci		No	None			Change Drop More
5	date	date			No	None			Change Drop More
6	Mobile No	bigint(10)			No	None			Change Drop More
7	City	text	utf8mb4_general_ci		No	None			Change Drop More

Below the table structure, there are buttons for 'Check all', 'With selected:', and various table operations like 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', and 'Add to central columns'.

Values in Registration



The screenshot shows the MySQL Workbench interface with the 'reg' table selected. The top message bar indicates '1 row inserted.' Below it, the SQL query used is:

```
INSERT INTO `reg` (`name`, `email`, `password`, `Confirm Password`, `date`, `Mobile No`, `City`) VALUES ('Virat Kolhi', 'virat114@gmail.com', 'Virat@1245', 'Virat@1245', '2023-08-03', '9874564789', 'Surat');
```

The results pane shows the inserted data:

	name	email	password	Confirm Password	date	Mobile No	City
1	Virat Kolhi	virat114@gmail.com	Virat@1245	Virat@1245	2023-08-03	9874564789	Surat
2	Rohit Mundra	rohit@gmail.com	Rohit@111	Rohit@111	2023-08-02	9874569874	Rajkot

Below the table, there are buttons for 'Check all', 'With selected:', and various table operations like 'Edit', 'Copy', 'Delete', and 'Export'.

Lab #5(b)
Lab#5(b) Adding a external stylesheet to make registration form look elegant.

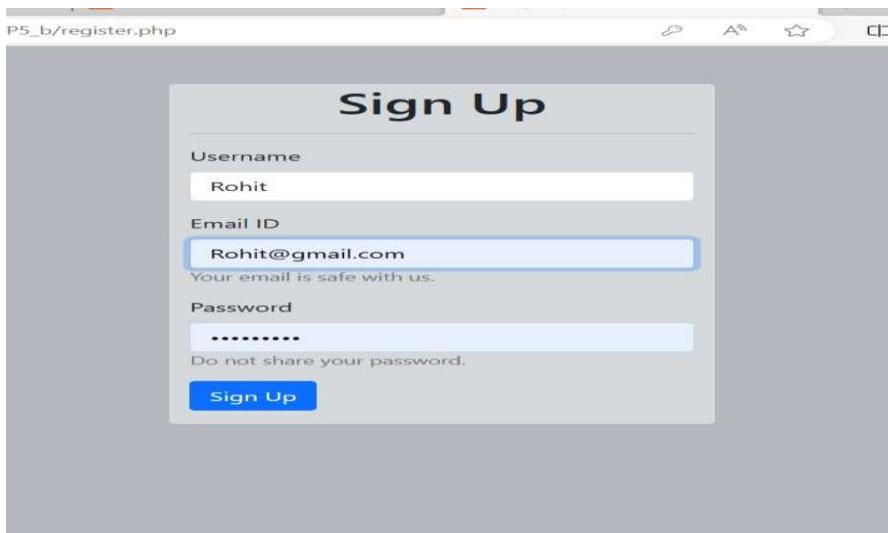
PROPERTY USED	ATTRIBUTES USED	DESCRIPTION
<?php	--	a server side scripting language that is embedded in HTML
\$_POST[]	name, email, password	PHP \$_POST is a PHP super global variable which is used to collect form data after submitting an HTML form
INSERT	--	used to insert values inside a particular table.
die()	\$conn	used to immediately terminate the execution of a script and display a specified message or error.
\$conn	mysqli, connect_error	\$CONN represents a database connection object. It is used when establishing a connection to a database server.
isset()		checks whether a variable is set, which means that it has to be declared and is not NULL
mysqli_query()	\$conn,\$query	accepts a string value representing a query as one of the parameters and, executes/performs the given query on the database.
mysqli_error	\$conn	used to return the error in the most recent MySQL function call that failed
mysqli_fetch_array()		used to fetch rows from the database and store them as an array
\$_SESSION[]	id,name,email	used to get or set the session id for the current session

Lab #5(b)

session_start()	creates a session or resumes the current one based on a session identifier passed via a GET or POST request.
-----------------	--------------------------------------------------------------------------------------------------------------

Final Outcome:

Registration Page:



P5_b/register.php

Sign Up

Username
Rohit

Email ID
Rohit@gmail.com
Your email is safe with us.

Password
.....
Do not share your password.

Sign Up

Lab #5(b)

Users table:



Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

SELECT * FROM `users`

ID	username	email	password
1	Rohit	Rohit@gmail.com	Rohit@111

Login page:



Dashboard:



Welcome, This your dashboard!

ID	Username	Email
1	Rohit	Rohit@gmail.com

[Sign Out](#)

Code Snippet:

database.php

```
<?php
$host= 'localhost';
$user= 'root';
$password = "";
$database = 'mydb2';
```

Lab #5(b)

```
$conn = mysqli_connect($host,$user,$password,$database);
if (!$conn)
{
die("Connection failed: " . mysqli_connect_error());
}
```

?>

```
register.php <?php if
(isset($_POST['signup'])) { $name =
$_POST['name']; $email =
$_POST['email'];
$password = $_POST['password'];

$query = "INSERT INTO users(username,email,password)
VALUES('{$name}', '{$email}', '{$password}')";
$addUser = mysqli_query($conn, $query);
```

```
if (!$addUser) { echo "Something went wrong" .
mysqli_error($conn);
} else { header('location: login.php');
}
?>
```

login.php

```
<?php session_start(); ?>
<?php

if (isset($_POST['signin'])) {
$email = $_POST['email'];
$password = $_POST['password'];

$query = "SELECT * from users WHERE email = '$email' AND password = '$password'";
$user = mysqli_query($conn, $query);

if (!$user) { die('query Failed' . mysqli_error($conn));
}
```

Lab #5(b)

```

while ($row = mysqli_fetch_array($user)) {

    $user_id = $row['ID'];
    $user_name = $row['username'];
    $user_email = $row['email'];
    $user_password = $row['password'];

}

if ($user_email == $email && $user_password == $password) {

    $_SESSION['id'] = $user_id; // Storing the value in session
    $_SESSION['name'] = $user_name; // Storing the value in session $_SESSION['email'] =
$user_email; // Storing the value in session header('location: dashboard.php?user_id=' .
$user_id);
} else { header('location: login.php');
}
}

?> dashboard.php
<?php session_start(); // Start the session ?>

<?php
if (!isset($_SESSION['id'])) { // condition Check: if session is not set. header('location:
login.php'); // if not set the user is sendback to login page.
}
?>

```

Lab #5(c)

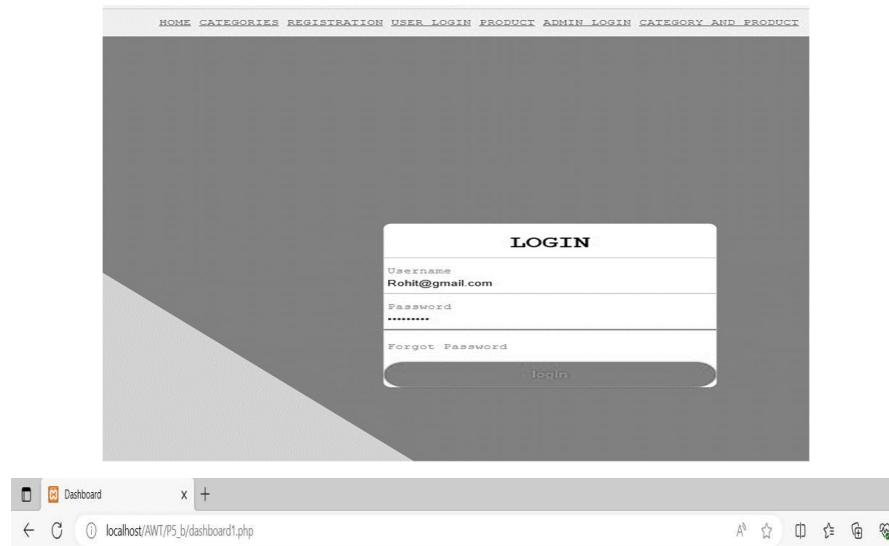
Lab#5(c) Store details like password, time of login and url in the cookies

PROPERTY USED	ATTRIBUTES USED	DESCRIPTION
<?php	--	a server side scripting language that is embedded in HTML
\$_POST[]	name, email, password	PHP \$_POST is a PHP super global variable which is used to collect form data after submitting an HTML form
INSERT	--	used to insert values inside a particular table.
die()	\$conn	used to immediately terminate the execution of a script and display a specified message or error.
isset()		checks whether a variable is set, which means that it has to be declared and is not NULL
mysqli_query()	\$conn,\$query	accepts a string value representing a query as one of the parameters and, executes/performs the given query on the database.
mysqli_error	\$conn	used to return the error in the most recent MySQL function call that failed
mysqli_fetch_array()		used to fetch rows from the database and store them as an array
\$_SESSION[]	id,name,email	used to get or set the session id for the current session
setcookie()		function defines a cookie to be sent along with the rest of the HTTP headers.

Lab #5(c)

time()		returns the current time in the number of seconds since the Unix Epoch
--------	--	------------------------------------------------------------------------

Final Outcome:



Welcome to the Dashboard

Last login time: 2023-08-10 08:25:57
Last visited URL: https://example.com/dashboard

Code Snippet:

login.php

```
<?php if ($_SERVER['REQUEST_METHOD'] === 'POST') {
if (isset($_POST['signin'])) {
$email = $_POST['email'];
$password = $_POST['password'];
```

Lab #5(c)

```
$query = "SELECT * from users WHERE email = '$email' AND password = '$password"'; $user =  
mysqli_query($conn, $query);  
  
if (!$user) { die('query Failed' . mysqli_error($conn));}  
  
}  
  
  
while ($row = mysqli_fetch_array($user)) {  
  
$user_id = $row['ID'];  
$user_name = $row['username'];  
$user_email = $row['email'];  
$user_password = $row['password'];  
  
}  
  
if ($user_email == $email && $user_password == $password) {  
$loginTime = time();  
$url = 'https://example.com/dashboard';  
// Set cookies setcookie('login_time', $loginTime, time() + (86400 * 30),  
'/'); setcookie('last_visited_url', $url, time() + (86400 * 30), '/');  
header('Location: dashboard1.php');  
exit;  
}  
?>
```

dashboard1.php

```

<?php

// Check if the user is logged in if
(!isset($_COOKIE['login_time'])) {

header('Location: login.php');

exit;

}

$loginTime = $_COOKIE['login_time'];

$lastVisitedUrl = isset($_COOKIE['last_visited_url']) ? $_COOKIE['last_visited_url'] : '';
$currentUrl = 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'];
setcookie('last_visited_url',
$currentUrl, time() + (86400 * 30), '/');

?>

<!DOCTYPE html>

<html>

<head>

<title>Dashboard</title>

</head>

<body>

<h1>Welcome to the Dashboard</h1>

<p>Last login time: <?php echo date('Y-m-d H:i:s', $loginTime); ?></p>

<p>Last visited URL: <?php echo $lastVisitedUrl; ?></p>

</body>

</html>

```

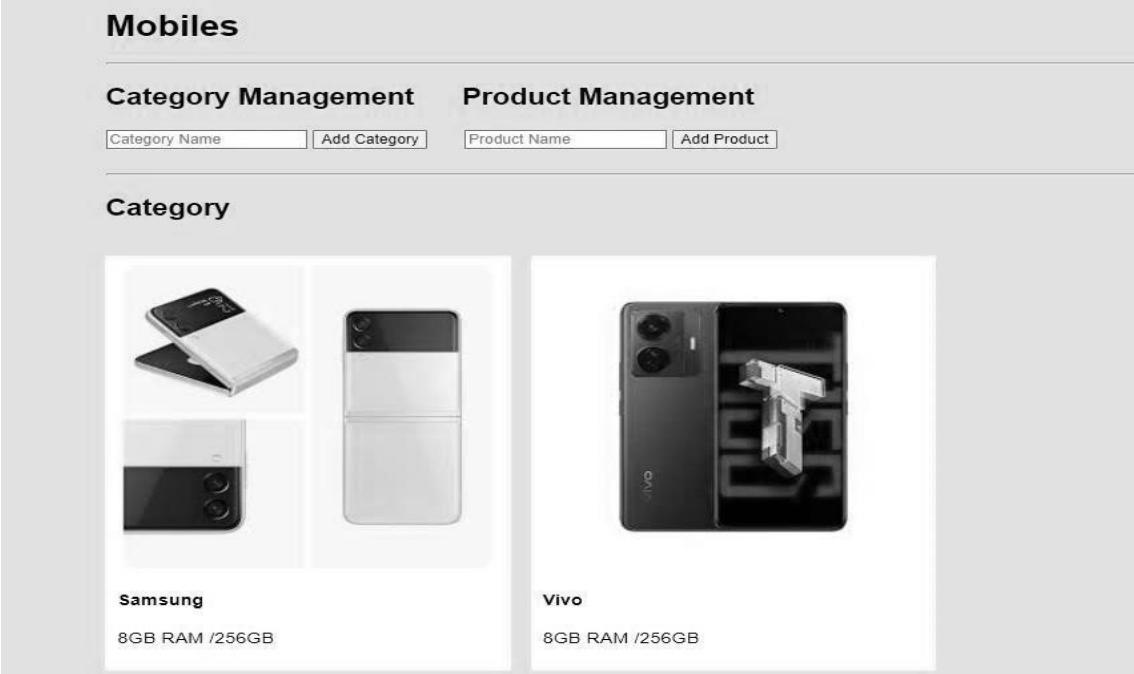
Lab #6(a)
Lab#6(a) Using design of lab 3:

- a. Implement a category and product management page with insert, update and delete operations.**

TAGS USED	ATTRIBUTES USED	DESCRIPTION
<HTML>	--	Define that how web browser will format and display the content.
<HEAD>	--	It is used to contain information of the document.
<TITLE>	--	Display the title of webpage.
<LINK>	Rel, href	Link External CSS
<BODY>	Background	Display the content of webpage
<H3>	Align	Define the Heading according H3 style
<Form>	--	It is use to create html for user input.
<Div>	class	It is for display the content section wise.
<input>	Type	It is use to declare input controls.
	Src, class	Display the particular image.
 (Break Line)	--	
<Button>	Class, Type	It is use to clickable function.

Lab #6(a)

Final Outcome:



The screenshot displays a user interface for managing mobile products. At the top, there are sections for 'Category Management' and 'Product Management' with input fields for 'Category Name' and 'Product Name' respectively, along with 'Add Category' and 'Add Product' buttons. Below these, a 'Category' section is shown with two items: 'Samsung' and 'Vivo'. Each item has a thumbnail image of a smartphone and its specifications listed below it.

Category	Product	RAM	Storage
Samsung	Samsung (Image)	8GB RAM	256GB
Vivo	Vivo (Image)	8GB RAM	256GB

Code Snippet:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
{
  box-sizing: border-box;
}
body {
  background-color: bisque;
  padding: 20px;
  font-family: Arial;
}
.main {
  max-width: 1000px;
  margin: auto;
}
h1 {
  font-size: 30px;
  word-break: break-
```

Lab #6(a)

```
all;  
}  
.row {  
margin: 10px -16px;  
}  
.row,  
.row > .column {  
padding: 8px;  
}  
.column {  
float: left;  
width: 33.33%;  
display: none;  
}  
.row:after { content: ""; display: table;  
clear: both;  
}  
.content {  
background-color: white;  
padding: 10px;  
}  
.show { display: block;  
}  
/style>  
</head>  
<body>  
<div class="main">  
<h1>Mobiles</h1>  
<hr>
```

Lab #6(a)

```

<h2>Category Management &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Product Management</h2> <form
id="category-form">

<input type="text" id="category-name" placeholder="Category Name" required>
<button type="submit">Add Category</button>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type="text" id="product-name" placeholder="Product Name" required>
<button type="submit">Add Product</button>
</form> &nbsp; <hr>
<h2>Category</h2>
<div class="row">
<div class="column nature">
<div class="content">

<h4>Samsung</h4>
<p>8GB RAM /256GB </p>
</div>
</div>
<div class="column nature">
<div class="content">

<h4>Vivo</h4>
<p>8GB RAM /256GB</p>
</div>
</div>
</div>
</div>
</body>
</html>

```

Lab #6(b)

Lab#6(b) Display products on home page from DB to all users.

Type Used	Attribute Used	DESCRIPTION
<?php	If, else, die	It is use for opening php.
\$_SERVER ['REQUEST_METHOD']	--	It is use for request the current script.
Die	\$conn	It is use for immediate terminate the execution of the script.
\$conn	mysqli, connect_error	It is use for connection purpose and establish the connection.
post	--	It is use for sending the data to the server.
\$stmt	Close()	It is use for representing the prepared statement object.
Insert	--	It is use for inserting the value into the particular table.
Prepare	--	It is use for creating the prepared statement.
Bind_param	--	It is use for binding the variables to the placeholders in prepared statement.
Echo	--	It is use for display the content.
?>	--	It is use for php ending tag.

Final Outcome:

Mobile Shopping Application Home Category Registration Sign In About Us

Product



Samsung Galaxy Z Flip3 5G

[More Details](#)



Vivo T1 5G

[More Details](#)

Code Snippet:

```
const { new } = require('nw'); const new
= new new({
name: "root", host:
"localhost", database:
"products", password: "",
});
module.exports = new; const express =
require('express'); const app = express();
const new = require('./db'); const PORT =
process.env.PORT || 3000;
app.get('/api/products', async (req, res) => {
try {
  const { rows } = await new.query('SELECT * FROM products');
  res.json(rows); } catch (error) { console.error(error);
  res.status(500).json({ message: 'Error fetching products' });
}
});
app.listen(PORT, () => { console.log(`Server is
running ${PORT}`);});
```

Lab #6(b)

```
async function fetchProducts() {  const response =  
await fetch('/api/products');  const products = await  
response.json();  
  const productsDiv = document.getElementById('products');  
products.forEach(product => {  const productElement =  
document.createElement('div');  productElement.innerHTML = `  
<h2>${product.name}</h2>  <p>Price:  
$ ${product.price}</p>`;  productsDiv.appendChild(productElement);  
});  
}  
fetchProducts();
```

Lab #6(c)

Lab#6(c) If user is logged in then show “more details” on each product and show detailed product description if user clicks “more details.”

TAGS USED	ATTRIBUTES USED	DESCRIPTION
<HTML>	--	Define that how web browser will format and display the content.
<HEAD>	--	It is used to contain information of the document.
<TITLE>	--	Display the title of webpage.
<LINK>	Rel, href	Link External CSS
<BODY>	Background	Display the content of webpage
<H3>	Align	Define the Heading according H3 style
<Form>	--	It is use to create html for user input.
<Div>	class	It is for display the content section wise.
<input>	Type	It is use to declare input controls.
	Src, class	Display the particular image.
 (Break Line)	--	
<Button>	Class, Type	It is use to clickable function.

Final Outcome:

Mobile Shopping Application Home Category Registration Sign In About Us

Product



Samsung Galaxy Z Flip3 5G

[More Details](#)



Vivo T1 5G

[More Details](#)

Code Snippet:

```
<!DOCTYPE html>
<html>
<head>
    <title>Mobile Shopping Application</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <style>
        body {
            font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
            background-color: #f0f0f0;
        }
        div {
            background-color: #e0e0e0;
            border: 1px solid #ccc;
            padding: 10px;
        }
        .button {
            background-color: #007bff;
            color: white;
            border: none;
            padding: 5px 10px;
            text-decoration: none;
            font-size: 1em;
        }
        .button:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <div>
        <h1>Mobile Shopping Application</h1>
        <ul>
            <li>Home</li>
            <li>Category</li>
            <li>Registration</li>
            <li>Sign In</li>
            <li>About Us</li>
        </ul>
        <h2>Product</h2>
        <div>
            <img alt="Samsung Galaxy Z Flip3 5G" data-bbox="165 265 525 495"/>
            <p>Samsung Galaxy Z Flip3 5G</p>
            <button>More Details</button>
        </div>
        <div>
            <img alt="Vivo T1 5G" data-bbox="625 285 855 495"/>
            <p>Vivo T1 5G</p>
            <button>More Details</button>
        </div>
    </div>
</body>

```

Lab #6(c)

```

</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Mobile Shopping Application</a>    <button
        class="navbar-toggler" type="button" data-bs-toggle="collapse" data- bs
        target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
        navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="index1.html">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="categories.html">Category</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="user_reg.html">Registration</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="login.html">Sign In</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">About Us</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
  <div class="container mt-4">
    <h1>Product</h1>
    <div class="row">
      <div class="col-md-4 mb-4">
        <div class="card">
          
          <div class="card-body">
            <h5 class="card-title">Samsung Galaxy Z Flip3 5G</h5>
            <p class="card-text">Samsung Galaxy Z Flip3 5G ,Cream,8GB,<br>256GB Storage</p>
            <div>Price: ₹89,000 &nbsp;&nbsp; <a href="#" class="btn btn-primary">Buy Now</a> </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

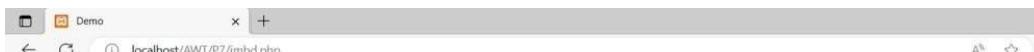
Lab #6(c)

```
</div>
<div class="col-md-4 mb-4">
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">Vivo T1 5G</h5>
      <p class="card-text">Vivo T1 5G,Dark Blue,8GB,<br>256 GB Storage</p>
      <div>Price: ₹17,000 &nbsp;&nbsp; <a href="#" class="btn btn-primary">Buy Now</a> </div>
    </div>
  </div>
</div>
</div>
</body>
</html>
```

Lab #7(a)
Lab#7 Using PHP Curl,

(a) Implementing Web Scrapping using CURL to fetch the relevant data from IMDB Movie Website.

PROPERTY USED	ATTRIBUTES USED	DESCRIPTION
<?php	--	a server side scripting language that is embedded in HTML
curl_init()		Initialize a curl session
curl_setopt()	\$curl,CURLOPT_URL,\$url	Set various options for the session
curl_exec()	\$curl	Execute and fetch/send data from/to server
errno()	\$curl	Returns the error number or 0 (zero) if no error occurred
curl_close()	\$curl	Close the session
preg_match_all	pattern,input,matches ,flags,offset	Returns the number of times the pattern was found in the string, which also may be 0

Final Outcome:


IMDB Site

Total Headings = 32

Movie Title : Harry Potter and the Sorcerer's Stone
 Movie Title : Harry Potter and the Chamber of Secrets
 Movie Title : Harry Potter and the Prisoner of Azkaban
 Movie Title : Harry Potter and the Goblet of Fire
 Movie Title : Harry Potter és a bőlcsek köve
 Movie Title : Harry Potter and the Order of the Phoenix
 Movie Title : Harry Potter and the Half-Blood Prince
 Movie Title : A Very Potter Musical
 Movie Title : A Very Potter Sequel
 Movie Title : Harry Potter and the Deathly Hallows: Part 1
 Movie Title : The Battle of Hogwarts
 Movie Title : Harry Potter and the Deathly Hallows: Part 2
 Movie Title : In Search of Hogwarts
 Movie Title : Harry Potter and the Ten Years Later
 Movie Title : A Very Potter Senior Year
 Movie Title : The Greater Good
 Movie Title : The Tale of the Three Brothers

Code Snippet:

```

<!DOCTYPE html>
<html>
<head>
<title>Demo</title>
<head>
<body>
<h1>IMDB Site</h1>
<?php

$curl=curl_init(); if(!$curl){
die("Cannot init curl.");
}

curl_setopt($curl,CURLOPT_URL,"https://www.imdb.com/list/ls072049366/");
curl_setopt($curl,CURLOPT_RETURNTRANSFER,true);
curl_setopt($curl,CURLOPT_SSL_VERIFYHOST,false);
curl_setopt($curl,CURLOPT_SSL_VERIFYPeer,false);

$result = curl_exec($curl); if(curl_errno($curl)){ echo "Error:
".curl_error($curl); }
else {
$regex = '/<h3 class="lister-item-header">.*?<a href=".*?(&.*?)</a>.*?</h3>/s';
$match_count=preg_match_all($regex,$result,$arr);
echo "<h2>Total Headings = ".$match_count."<br/><hr/></h2>";
$mn = 1;
foreach ($arr[1] as $title) { echo "<div class='movie-title'>" . $mn . ". Movie Title: " . $title .
"</div><hr>"; $mn++;
}
}
curl_close($curl);
?>
</body>
</html>

```

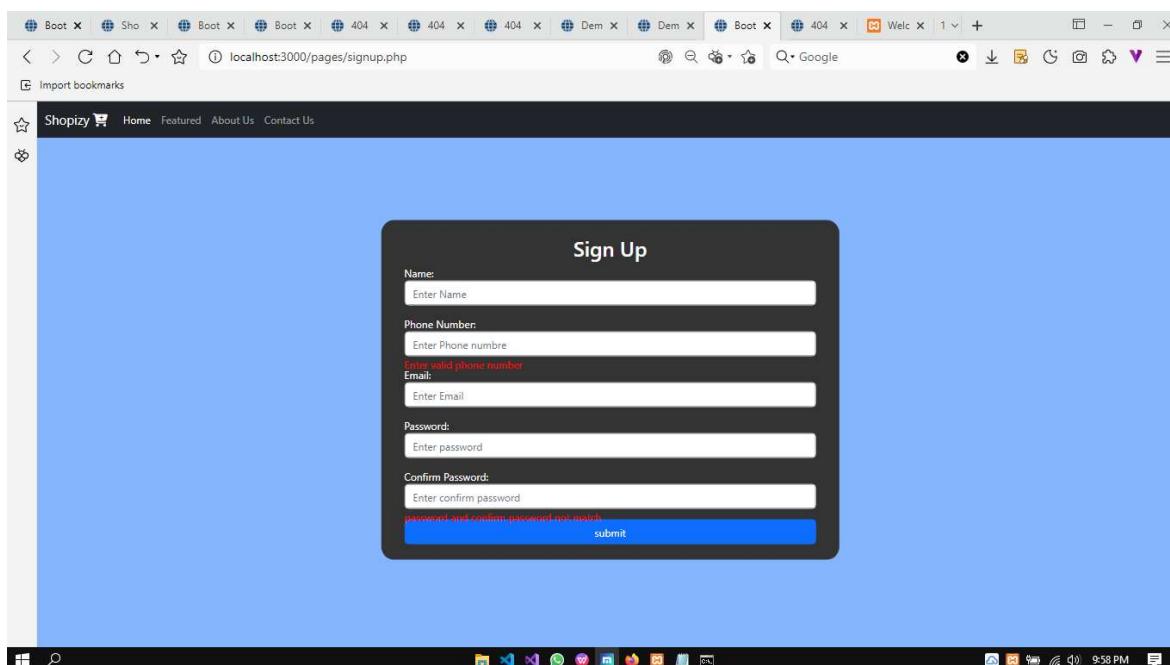
Lab #8(a)

Lab#8 Using PHP Regex,

a. Implementing Registration Form Validation using Regular Expressions.

PROPERTY USED	ATTRIBUTES USED	DESCRIPTION
<?php	--	a server side scripting language that is embedded in HTML
curl_init()		Initialize a curl session
curl_setopt()	\$curl,CURLOPT_URL,\$url	Set various options for the session
curl_exec()	\$curl	Execute and fetch/send data from/to server
errno()	\$curl	Returns the error number or 0 (zero) if no error occurred
curl_close()	\$curl	Close the session
preg_match_all	pattern,input,matches ,flags,offset	Returns the number of times the pattern was found in the string, which also may be 0

Final Outcome:



Code Snippet:

```

<!DOCTYPE html>
<html lang="en">

<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.3.0/css/all.min.css">
<style>

.form-box-signup{
background-color: #333;
border-radius: 20px;
width: 40%;
position: absolute;
top:50%;
left:50%;
transform: translate(-50%,-50%);

}

.user button i {
color: #fff;

border-radius: 50%;
}

.card-body a {
text-decoration: none;
color: #0aff37;
}

.login-media #loginlink,
#cartlink {
color: #fff;
display: none;
}

.user button {
border: 2px solid #fff;
transform: scale(1.13);

```

Lab #8(a)

```

margin-left: 15px;
border-radius: 50%;
background-color: transparent;
}
.cart i {
  color: #fff;
  border: 2px solid #fff;
  margin-left: 10px;
  padding: 7px;
  border-radius: 50%;
}
.slider img {
  height: 70%;
  width: 100%;
}

.row img {
  margin-top: 10px;
  margin-left: 70px;
  height: 47%;
  width: 65%;
}
.star {
  background-color: rgb(34, 0, 128);
  color: #fff;
  padding: 0px 4px 0 4px;
  border-radius: 4px;
}
.sell {
  margin-left: 15px;
  transform: translateY(-2px);
  display: inline-block;
}
.office-cat {
  margin-top: 25px;
}
.error{
  color: red;
  position: absolute;
}
</style>
</head>

<body class=" bg-opacity-50 bg-primary ">

<?php

```

Lab #8(a)

```

session_start();

#define empty variable;
$name_err=$mob_err=$email_err=$pass_err=$cpass_err=$dob_err="";
if($_SERVER["REQUEST_METHOD"]=="POST")
{
//NAME VALIDATION
if(empty($_POST['name']))
{
    $name_err="Enter Name";
}
else if(!preg_match("/^[a-zA-Z]*$/,$_POST['name'])) 
{
    $name_err="only letters and white space allowed";
}

//MOBILE NUMBER VALIDATION

if(empty($_POST['phone']))
{
    $mob_err="enter mobile number";
}
else if(!preg_match("/^([0-9]{10})$/,$_POST['phone'])) 
{
    $mob_err="Enter valid phone number";
}

//EMAIL VALIDATION
if(empty($_POST['email']))
{
    $email_err="enter email";
}
else if(!filter_var($_POST['email'],FILTER_VALIDATE_EMAIL)) 
{
    $email_err="Enter Valid email";
}

//PASSWORD VALIDATION

if(empty($_POST['pswd']))
{
    $pass_err="enter password";
}
else if(strlen($_POST['pswd'])<8))

```

Lab #8(a)

```
{
    $pass_err="password must be atleast 8 characters";
}

//CONFIRM PASSWORD ERR

if(empty($_POST['cpswd']))
{
    $cpass_err="Enter Confirm password";
}
else if($_POST['pswd']!=$_POST['cpswd'])
{
    $cpass_err="password and confirm password not match";
}

if(empty($name_err)&&empty($mob_err)&&empty($email_err)&&empty($pass_err)&&empty($cpas
s_err))
{
    $_SESSION['uname']=$_POST['name'];
    $_SESSION['email']=$_POST['email'];
    $_SESSION['password']=$_POST['pswd'];

    header("location:home.php");
}

?>
<!--navbar start-->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark text-white">
<div class="container-fluid">
    <a class="navbar-brand" href="#">Shopizy<i class="fa-sharp fa-solid fa-cart-
plus"></i></a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarScroll" aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarScroll">
        <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll" style="--bs-scroll-height:
100px;">
            <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="#">Home</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" aria-current="page" href="featured.html">Featured</a>
            </li>
        </ul>
    </div>
</div>

```

Lab #8(a)

```

</li>
<li class="nav-item">
    <a class="nav-link" area-curren="page" href="about.html">About Us</a>
</li>
<li class="nav-item">
    <a class="nav-link" area-curren="page" href="contact.html">Contact Us</a>
</li>
</ul>
</div>
</div>
</nav>
<!--navbar over-->
<!--Form starts here-->
<div class="form-box-signup mt-4" id="demo" >
    <div class="container mt-4 text-white">
        <h2 class="text-center">Sign Up</h2>
        <form action="" method="post" onsubmit="return handleSubmit()">
            <div class="mb-3 mt-1 mx-4">
                <label for="email">Name:</label>
                <input type="text" class="form-control" id="name" placeholder="Enter Name" name="name">
                <span class="error"><?php echo $name_err;?> </span>
            </div>
            <div class="mb-3 mt-3 mx-4">
                <label for="phone">Phone Number:</label>
                <input type="text" class="form-control" id="phnum" placeholder="Enter Phone number" name="phone">
                <span class="error"><?php echo $mob_err;?> </span>
            </div>
            <div class="mb-3 mt-3 mx-4">
                <label for="email">Email:</label>
                <input type="email" class="form-control" id="email" placeholder="Enter Email" name="email">
                <span class="error"><?php echo $email_err;?> </span>
            </div>
            <div class="mb-3 mx-4">
                <label for="pwd">Password:</label>
                <input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pswd">
                <span class="error"><?php echo $pass_err;?> </span>
            </div>
            <div class="mb-3 mx-4">
                <label for="cpwd">Confirm Password:</label>
                <input type="password" class="form-control" id="cpwd" placeholder="Enter confirm password" name="cpswd">
                <span class="error"><?php echo $cpass_err;?> </span>
            </div>
        </form>
    </div>
</div>

```

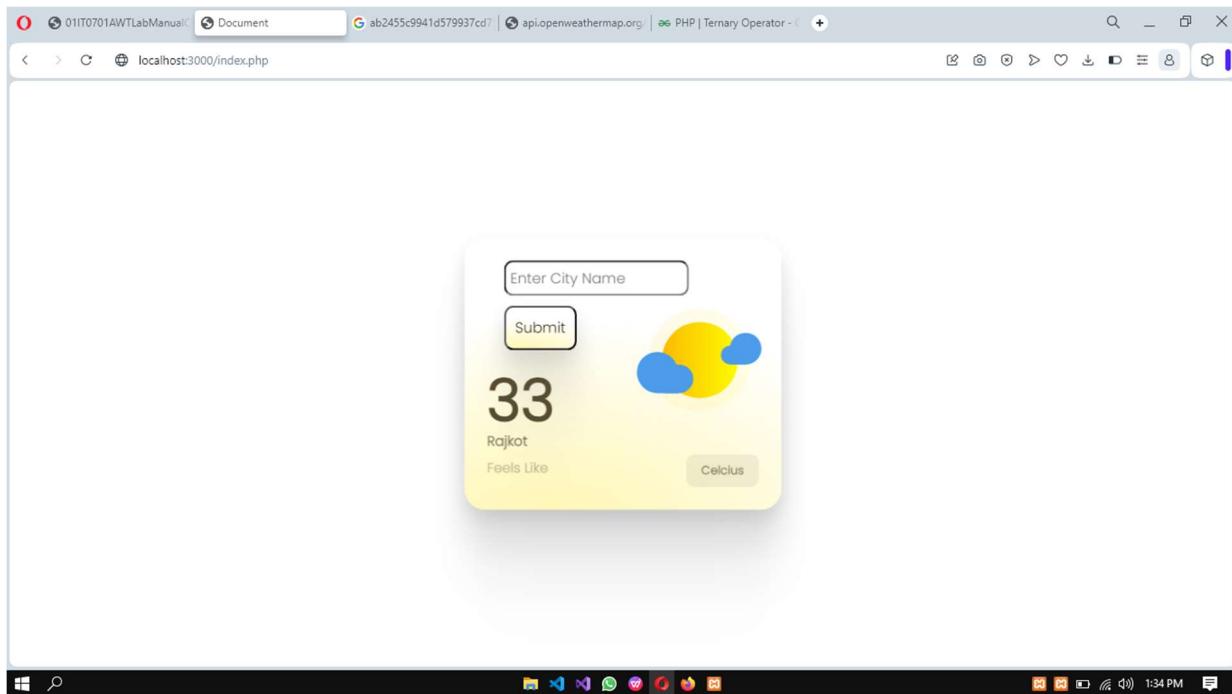
Lab #8(a)

```
<div class="row mb-4 mx-4 mt-2">
<input type="submit" class="btn btn-primary" value="submit">
</div>
</form>
</div>
</div>
</body>
</html>
```

Lab #9(a)
Lab#9 Using PHP Regex,
a. Implementing an application to fetch weather details using Weather

PROPERTY USED	ATTRIBUTES USED	DESCRIPTION
<?php	--	a server side scripting language that is embedded in HTML
Isset()	--	Check for variable is initialised or not
Json_decode()	--	Deflate Json data from api's
File_get_contents()	--	Fetch content from file
errno()	--	Returns the error number or 0 (zero) if no error occurred
<HTML>	lang="en"	Write the description of HTML tags and attributes
<HEAD>	-	Contains meta-information about the document, such as title and CSS scripts.
<TITLE>	-	Defines the title of the document that appears in the browser's title bar.
<LINK>	rel, href	Links external resources, like CSS stylesheets, to the HTML document.
<BODY>	-	Contains the visible content of the document.
<NAV>	-	Defines a container for navigation links.
<DIV>	class	A generic container element for styling and layout purposes.
<A>	class, href	Creates hyperlinks and defines the URL they link to.

Final Outcome:



Code Snippet:

request.php

```

<?php

$ApiKey='ab2455c9941d579937cd7504e03449b6';

$city='Rajkot';

$URL="https://api.openweathermap.org/data/2.5/weather?q=".$city."&appid=".$ApiKey;

$json_data=file_get_contents($URL);

$res_data=json_decode($json_data);

$tempInKelvin=$res_data->main->temp;

$tempInCelcius=ceil($tempInKelvin-273.15);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="style.css">

<script src="index.js" defer></script>

</head>

<body>

<div class="card">

<input id="cityData" placeholder="Enter City Name" type="text">

<button id="btn_search" >Search</button>

<div class="container">

<div class="cloud front">

```

Lab #9(a)

```

<span class="left-front"></span>
<span class="right-front"></span>
</div>

<span class="sun sunshine"></span>
<span class="sun"></span>

<div class="cloud back">
    <span class="left-back"></span>
    <span class="right-back"></span>
</div>

</div>

<div class="card-header">
    <span id="city_name" >Messadine, Susah</span>
    <span >Feels Like <span id="feels-like" ></span> </span>
</div>

<span class="temp"><?php echo $tempInCelcius; ?></span>

<div class="temp-scale">
    <span>Celcius</span>
</div>
</div>

<script>

let city=document.getElementById("cityData").value;
</script>
</body>
</html>

```

```

<?php
$tempInCelcius=null;
if(isset($_POST['submit']))
{
$ApiKey='ab2455c9941d579937cd7504e03449b6';
$city=$_POST['citydata'];
$URL="https://api.openweathermap.org/data/2.5/weather?q=".$city."&appid=".$ApiKey;
$json_data=file_get_contents($URL);
$res_data=json_decode($json_data);
$tempInKelvin=$res_data->main->temp;

$tempInCelcius=ceil($tempInKelvin-273.15);

}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>

<link rel="stylesheet" href="style.css">
<!-- <script src="index.js" defer></script> -->
</head>
<body>

```

Lab #9(a)

```

<div class="card">
  <form action="" method="post">
    <input id="cityData" name="citydata" placeholder="Enter City Name" type="text"/>
    <input type="submit" name="submit" id="btn_search" />
  </form>

  <div class="container">
    <div class="cloud front">
      <span class="left-front"></span>
      <span class="right-front"></span>
    </div>

    <span class="sun sunshine"></span>
    <span class="sun"></span>

    <div class="cloud back">
      <span class="left-back"></span>
      <span class="right-back"></span>
    </div>
  </div>

```

```

<div class="card-header">
  <span id="city_name" ><?php if(isset($_POST['citydata'])) echo $_POST['citydata']; else echo "";;
?></span>
  <span >Feels Like <span id="feels-like" ></span> </span>
</div>

```

```

<span class="temp"><?php echo $tempInCelcius; ?></span>
<div class="temp-scale">
  <span>Celcius</span>
</div>
</div>

```

Lab #9(a)

```
</body>
</html>
```

Style.css

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300&display=swap');
```

```
*{
  font-family: 'Poppins', sans-serif;
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}
```

```
body{
  min-height: 100vh;
  width: 100vw;
  display: grid;
  justify-content: center;
  align-items: center;
}
```

```
.card {
  width: 350px;
  height: 300px;
  position: relative;
  padding: 25px;
  background: radial-gradient(178.94% 106.41% at 26.42% 106.41%, #FFF7B1 0%, rgba(255, 255, 255, 0) 71.88%) /* warning: gradient uses a rotation that is not supported by CSS and may not behave as expected */, #FFFFFF;
  box-shadow: 0px 155px 62px rgba(0, 0, 0, 0.01), 0px 87px 52px rgba(0, 0, 0, 0.05), 0px 39px 39px rgba(0, 0, 0, 0.09), 0px 10px 21px rgba(0, 0, 0, 0.1), 0px 0px 0px rgba(0, 0, 0, 0.1);
```

Lab #9(a)

```
border-radius: 23px;  
transition: all 0.8s cubic-bezier(0.15, 0.83, 0.66, 1);  
cursor: pointer;  
}  
  
input{  
position: absolute;  
display: inline-block;  
padding: 5px;  
font-size: 1em;  
left: 30%;  
transform: translateX(-30%);  
border-radius: 10px;  
outline: none;  
}  
  
button{  
position: absolute;  
right: 30px;  
padding: 5px;  
font-size: 1em;  
border-radius: 10px;  
background-color: #91ff00;  
color: #fff;  
outline: none;  
font-weight: 600;  
cursor: pointer;  
}  
.card:hover {  
transform: scale(1.05);
```

Lab #9(a)

}

.container {

margin-top: 60px;

width: 250px;

height: 250px;

position: absolute;

right: -35px;

top: -50px;

display: flex;

align-items: center;

justify-content: center;

transform: scale(0.7);

}

.cloud {

width: 250px;

}

.front {

padding-top: 45px;

margin-left: 25px;

display: inline;

position: absolute;

z-index: 11;

animation: clouds 8s infinite;

animation-timing-function: ease-in-out;

}

.back {

margin-top: -30px;

margin-left: 150px;

Lab #9(a)

```

z-index: 12;
animation: clouds 12s infinite;
animation-timing-function: ease-in-out;
}

.right-front {
width: 45px;
height: 45px;
border-radius: 50% 50% 50% 0%;
background-color: #4c9beb;
display: inline-block;
margin-left: -25px;
z-index: 5;
}

#btn_search{
margin-top: 50px;
margin-left: -47px;
position: relative;
padding: 10px;
background: radial-gradient(178.94% 106.41% at 26.42% 106.41%, #FFF7B1 0%, rgba(255, 255, 255, 0) 71.88%) /* warning: gradient uses a rotation that is not supported by CSS and may not behave as expected */, #FFFFFF;
box-shadow: 0px 155px 62px rgba(0, 0, 0, 0.01), 0px 87px 52px rgba(0, 0, 0, 0.05), 0px 39px 39px rgba(0, 0, 0, 0.09), 0px 10px 21px rgba(0, 0, 0, 0.1), 0px 0px 0px rgba(0, 0, 0, 0.1);
border-radius: 10px;
transition: all 0.8s cubic-bezier(0.15, 0.83, 0.66, 1);
cursor: pointer;
}

.temp{

```

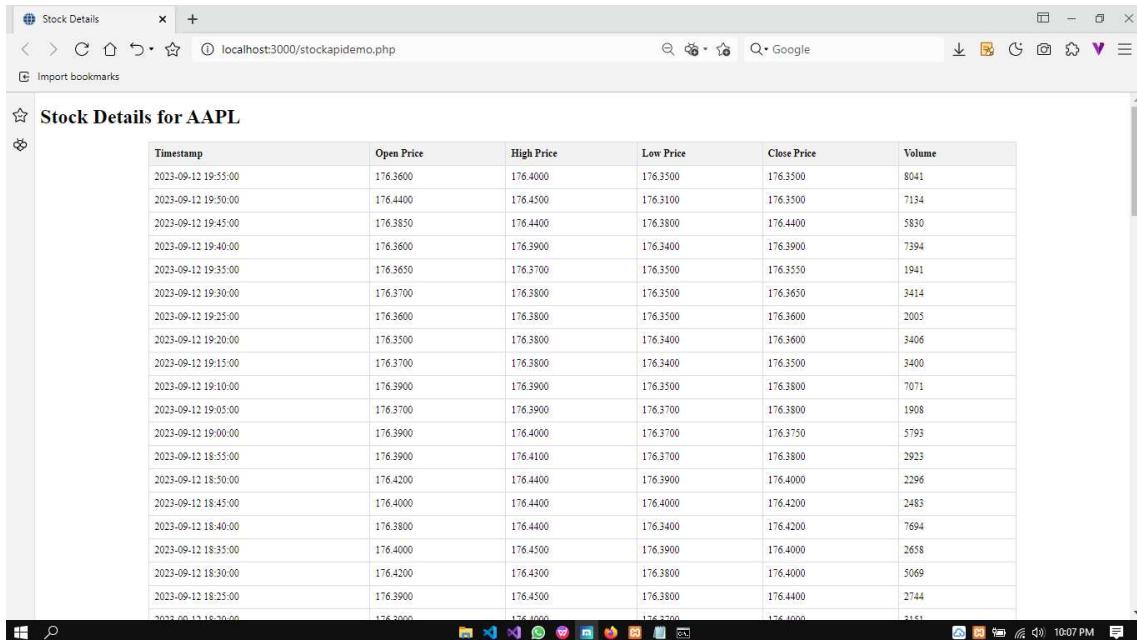
Lab #9(a)

```
margin-bottom: 70px;  
}  
  
#btn_search:hover{  
background-color: rgb(149, 255, 0);  
}
```

Lab#9 Using PHP Regex,

b. Implementing an application to fetch cricket/stock details using relevant

Final Outcome:



Timestamp	Open Price	High Price	Low Price	Close Price	Volume
2023-09-12 19:55:00	176.3600	176.4000	176.3500	176.3500	8041
2023-09-12 19:50:00	176.4400	176.4500	176.3100	176.3500	7134
2023-09-12 19:45:00	176.3850	176.4400	176.3800	176.4400	5830
2023-09-12 19:40:00	176.3600	176.3900	176.3400	176.3900	7394
2023-09-12 19:35:00	176.3650	176.3700	176.3500	176.3550	1941
2023-09-12 19:30:00	176.3700	176.3800	176.3500	176.3650	3414
2023-09-12 19:25:00	176.3600	176.3800	176.3500	176.3600	2005
2023-09-12 19:20:00	176.3500	176.3800	176.3400	176.3600	3406
2023-09-12 19:15:00	176.3700	176.3800	176.3400	176.3500	3400
2023-09-12 19:10:00	176.3900	176.3900	176.3500	176.3800	7071
2023-09-12 19:05:00	176.3700	176.3900	176.3700	176.3800	1908
2023-09-12 19:00:00	176.3900	176.4000	176.3700	176.3750	5793
2023-09-12 18:55:00	176.3900	176.4100	176.3700	176.3800	2923
2023-09-12 18:50:00	176.4200	176.4400	176.3900	176.4000	2296
2023-09-12 18:45:00	176.4000	176.4400	176.4000	176.4200	2483
2023-09-12 18:40:00	176.3800	176.4400	176.3400	176.4200	7694
2023-09-12 18:35:00	176.4000	176.4500	176.3900	176.4000	2658
2023-09-12 18:30:00	176.4200	176.4300	176.3800	176.4000	5069
2023-09-12 18:25:00	176.3900	176.4500	176.3800	176.4400	2744
2023-09-12 18:20:00	176.3600	176.4000	176.3700	176.4000	2151

Code Snippet:

```

<?php
$alphaVantageApiKey = 'YOUR_ALPHAVANTAGE_KEY';
$symbol = 'AAPL'; // Replace with the stock symbol you want to fetch

// Fetch stock details
$stockApiUrl =
"https://www.alphavantage.co/query?function=TIME_SERIES_INTRADAY&symbol=$symbol&interval=
5min&apikey=$alphaVantageApiKey";
$stockData = file_get_contents($stockApiUrl);

// Process stock data
$stockdataArray = json_decode($stockData, true);
$latestData = $stockdataArray['Time Series (5min)'];

// Create an array of data for the table
$tableData = [];
foreach ($latestData as $timestamp => $data) {
  $tableData[] = [
    'Timestamp' => $timestamp,
    'Open Price' => $data['1. open'],
    'High Price' => $data['2. high'],
    'Low Price' => $data['3. low'],
    'Close Price' => $data['4. close'],
    'Volume' => $data['5. volume']
  ];
}

```

Lab #9(b)

```

'Open Price' => $data['1. open'],
'High Price' => $data['2. high'],
'Low Price' => $data['3. low'],
'Close Price' => $data['4. close'],
'Volume' => $data['5. volume'],
];
}

?>

<!DOCTYPE html>
<html>
<head>
<title>Stock Details</title>
<style>
  table {
    border-collapse: collapse;
    width: 80%;
    margin: 20px auto;
  }
  th, td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;
  }
  th {
    background-color: #f2f2f2;
  }
</style>
</head>
<body>
<h1>Stock Details for <?php echo $symbol; ?></h1>
<table>
  <tr>
    <th>Timestamp</th>
    <th>Open Price</th>
    <th>High Price</th>
    <th>Low Price</th>
    <th>Close Price</th>
    <th>Volume</th>
  </tr>
<?php foreach ($tableData as $row): ?>
  <tr>
    <td><?php echo $row['Timestamp']; ?></td>
    <td><?php echo $row['Open Price']; ?></td>
    <td><?php echo $row['High Price']; ?></td>
    <td><?php echo $row['Low Price']; ?></td>

```

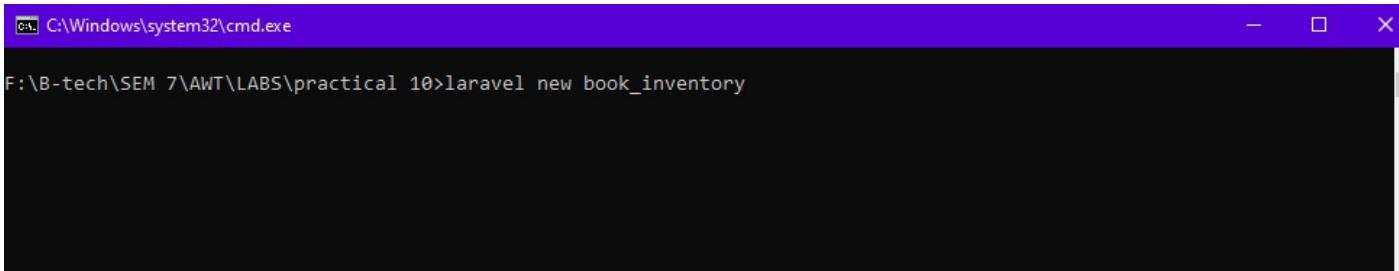
Lab #9(b)

```
<td><?php echo $row['Close Price']; ?></td>
<td><?php echo $row['Volume']; ?></td>
</tr>
<?php endforeach; ?>
</table>
</body>
</html>
```

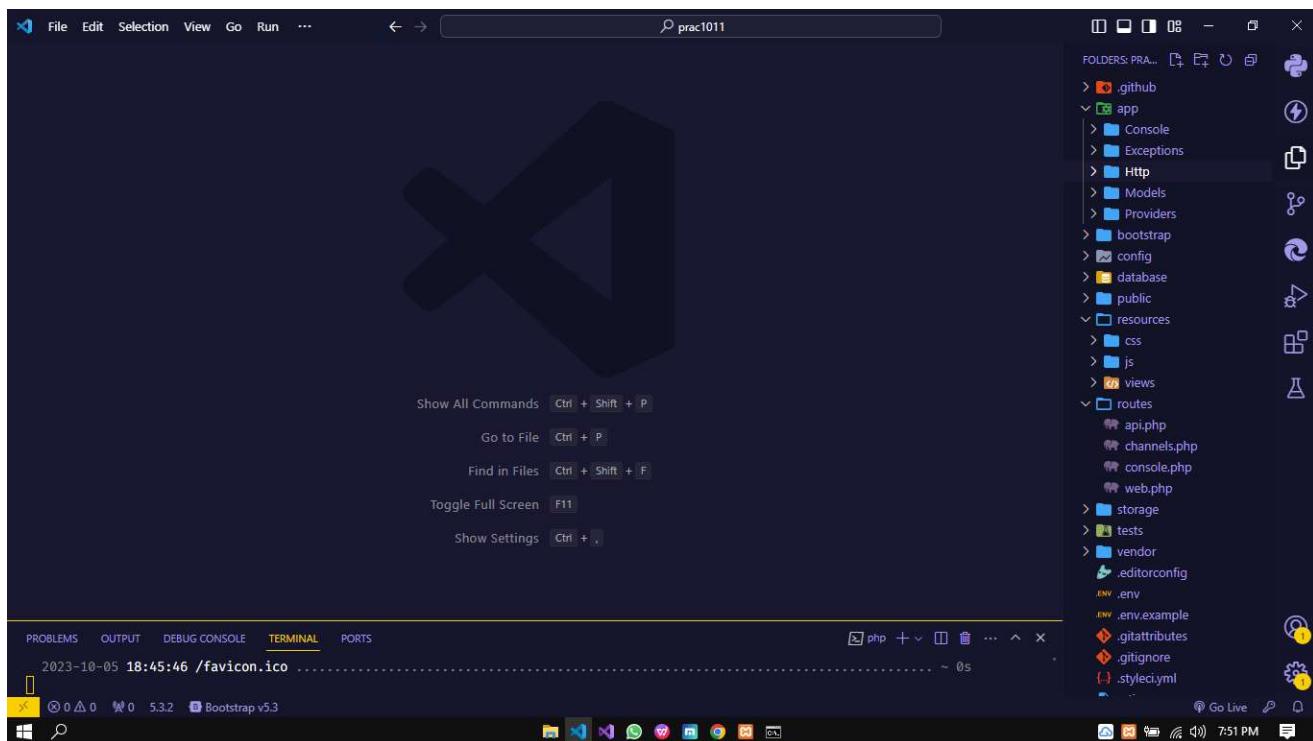
Lab #10(a)
Lab#10(a) Setup a Laravel project for book inventory

COMMAND USED	ATTRIBUTES USED	DESCRIPTION
Artisan	make:controller, make:model, make:migration, serve	In Laravel, Artisan is a command-line tool that provides a wide range of commands for various tasks related to the development and management of Laravel applications. Artisan simplifies common development tasks by automating them through command-line commands.

For creating a Laravel Project for book inventory management create a folder for the project and open terminal and type command “laravel new book_inventory”.

Final Outcome:


```
C:\Windows\system32\cmd.exe
F:\B-tech\SEM 7\AWT\LABS\practical 10>laravel new book_inventory
```



Lab #10(a)

Code Snippet:

For creating a Laravel Project for book inventory management create a folder for the project and open terminal and type command “larave new book_inventory”.

Lab #10(b)
b. Setup the database in Laravel.

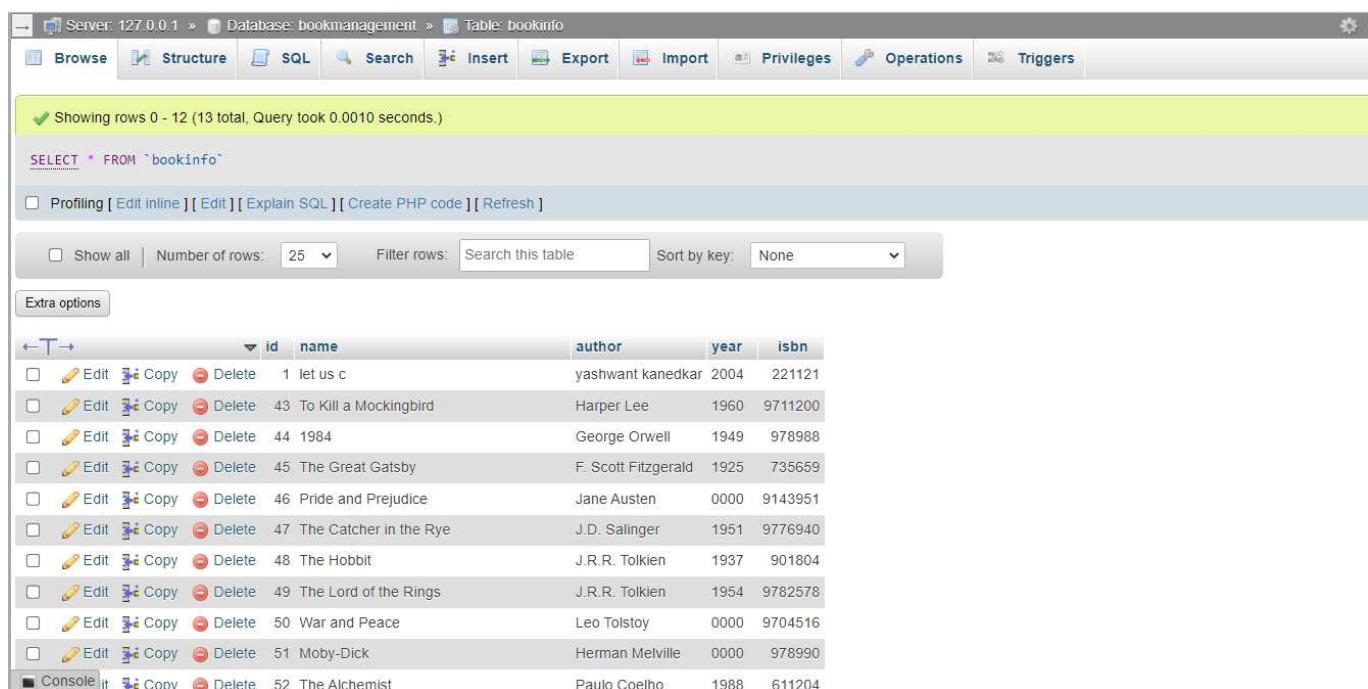
Change this values in .env file

Final Outcome:

```

LOG_CHANNEL=stack
LOG_DEPRECATED_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=bookmanagement
DB_USERNAME=root
DB_PASSWORD=
  
```



The screenshot shows the MySQL Workbench interface with the following details:

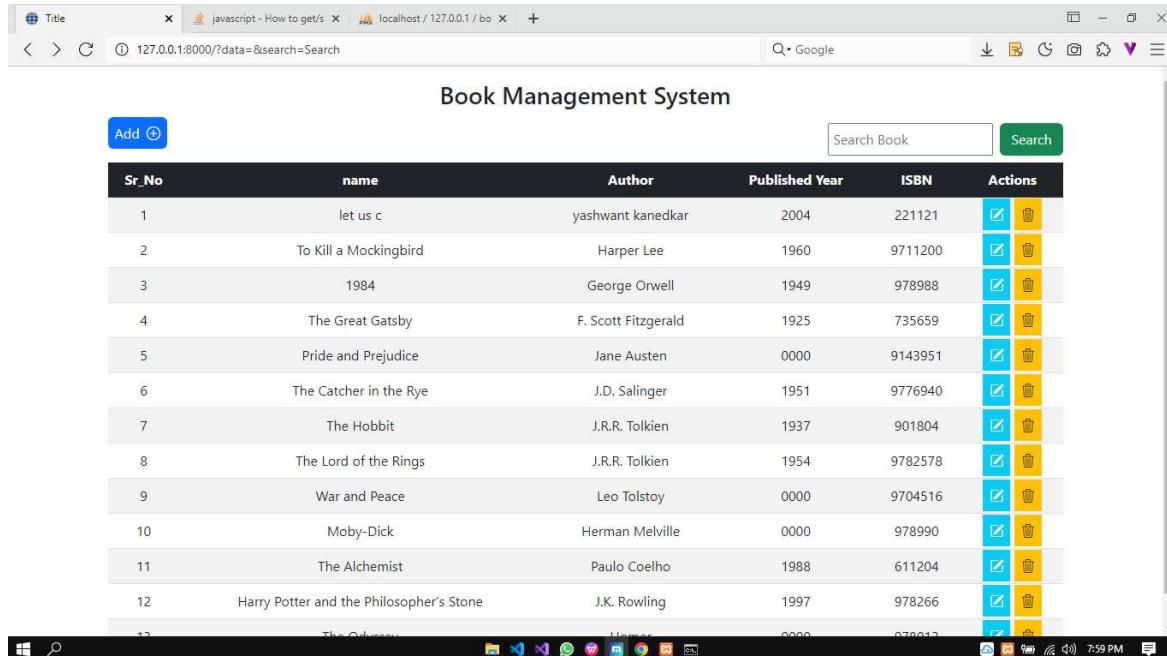
- Server:** 127.0.0.1
- Database:** bookmanagement
- Table:** bookinfo
- Rows:** 0 - 12 (13 total)
- Query Result:** SELECT * FROM `bookinfo`
- Table Headers:** id, name, author, year, isbn
- Data Rows:** 52 rows listed, starting with:

1	let us c	yashwant kanedkar	2004	221121
43	To Kill a Mockingbird	Harper Lee	1960	9711200
44	1984	George Orwell	1949	978988
45	The Great Gatsby	F. Scott Fitzgerald	1925	735659
46	Pride and Prejudice	Jane Austen	0000	9143951
47	The Catcher in the Rye	J.D. Salinger	1951	9776940
48	The Hobbit	J.R.R. Tolkien	1937	901804
49	The Lord of the Rings	J.R.R. Tolkien	1954	9782578
50	War and Peace	Leo Tolstoy	0000	9704516
51	Moby-Dick	Herman Melville	0000	978990
52	The Alchemist	Paulo Coelho	1988	611204

Lab #10(c)

c. Implement Select Operation in Laravel with GUI

Final Outcome:



Sr_No	name	Author	Published Year	ISBN	Actions
1	let us c	yashwant kanedkar	2004	221121	 
2	To Kill a Mockingbird	Harper Lee	1960	9711200	 
3	1984	George Orwell	1949	978988	 
4	The Great Gatsby	F. Scott Fitzgerald	1925	735659	 
5	Pride and Prejudice	Jane Austen	0000	9143951	 
6	The Catcher in the Rye	J.D. Salinger	1951	9776940	 
7	The Hobbit	J.R.R. Tolkien	1937	901804	 
8	The Lord of the Rings	J.R.R. Tolkien	1954	9782578	 
9	War and Peace	Leo Tolstoy	0000	9704516	 
10	Moby-Dick	Herman Melville	0000	978990	 
11	The Alchemist	Paulo Coelho	1998	611204	 
12	Harry Potter and the Philosopher's Stone	J.K. Rowling	1997	978266	 
13	The Odyssey	Homer	0000	978012	 

Code snippet:

```

<!doctype html>
<html lang="en">
<head>
  <title>Title</title>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
  icons/1.11.1/font/bootstrap-icons.min.css" integrity="sha512-
  oAvZuuYVzkcTc2dH5z1ZJup50mSQ000qlfRvuTTiyTBjwX1faoyearj8KdMq0LgsBTHMrRuMek7s+CxF
  8yE+w==" crossorigin="anonymous" referrerPolicy="no-referrer" />
  <!-- Bootstrap CSS v5.2.1 -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
  rel="stylesheet"
  integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6ssoy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
  crossorigin="anonymous"

```

Lab #10(c)

```

</head>
<body>
<header>
    <!-- place navbar here -->
</header>
<main>
</main>
<footer>
    <!-- place footer here -->
</footer>
<!-- Bootstrap JavaScript Libraries -->
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
    integrity="sha384-
0BqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
    crossorigin="anonymous">
</script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"
    integrity="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz"
    crossorigin="anonymous">
</script>
<h1 class="fs-3 text-center my-3">Book Management System</h1>
<div class="container">
    <div class="content d-flex mb-2 justify-content-between">
        <div class="addbtn">
            <a class="bg-primary rounded text-white p-2 my-1 text-decoration-none " href="/add">Add<i
                class="ms-2 bi bi-plus-circle"></i></a>
        </div>
        <div class="searchbox">
            <form method="get" class="d-flex justify-content-center gap-2" action="">
                <input type="text" class="p-1" name="data" placeholder="Search Book">
            </form>
        </div>
    </div>
</div>

```

Lab #10(c)

```

<input class="btn btn-success" name="search" type="submit" value="Search">
</form>
</div>
</div>

<table class="table table-striped caption-top table-hover table-responsive ">
<thead class="text-center table-dark">
<tr>
    <th scope="col">Sr_No</th>
    <th scope="col">
        name
    </th>
    <th scope="col">
        Author
    </th>
    <th scope="col">
        Published Year
    </th>
    <th scope="col">
        ISBN
    </th>
    <th scope="col">Actions</th>
</tr>
</thead>
<tbody class="text-center">
@php
    $count=1;
@endphp
@foreach ($r as $book)
<tr scope="row">

```

Lab #10(c)

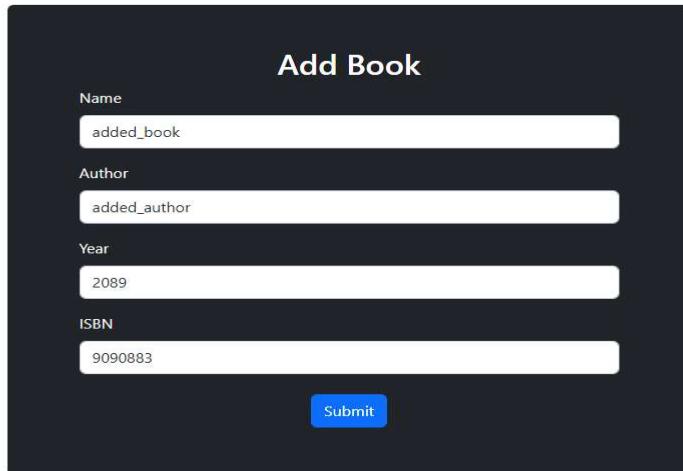
```

<td>{{$count}}</td>
<td>{{$book->name}}</td>
<td>{{$book->author}}</td>
<td>{{$book->year}}</td>
<td>{{$book->isbn}}</td>
@php
$count++;
@endphp
<td>
<a id="" class="text-white bg-info p-2" href="{{route('book.edit',['id'=>$book->id])}}"><i
class="bi bi-pencil-square"></i></a>
<a id="delbtn" href="{{route('book.delete',['id'=>$book->id])}}" class="text-dark bg-
warning p-2" > <i class="bi bi-trash3"></i></a>
</td>
</tr>
@endforeach
</tbody>
</table>
</div>
</body>
</html>

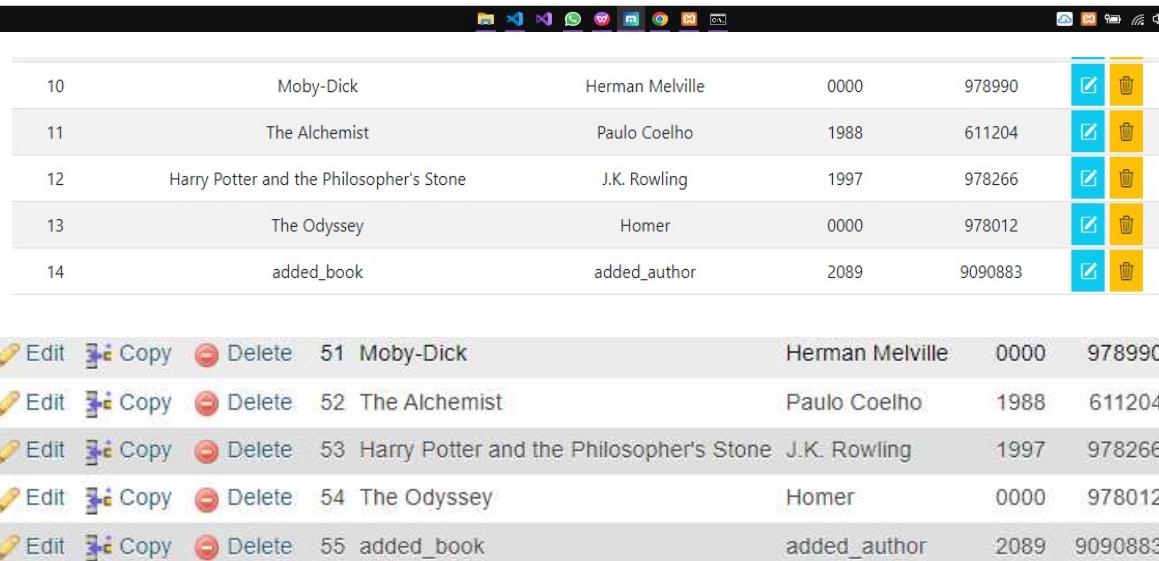
```

Lab #11(a)
Lab#11(a) Implement Insert Operation with GUI to add new details of book in book inventory

COMMAND USED	ATTRIBUTES USED	DESCRIPTION
Artisan	make:controller, make:model, make:migration, serve	In Laravel, Artisan is a command-line tool that provides a wide range of commands for various tasks related to the development and management of Laravel applications. Artisan simplifies common development tasks by automating them through command-line commands.

Final Outcome:


The screenshot shows a web browser window with the URL `127.0.0.1:8000/add`. The page title is "Add Book". The form contains four input fields: "Name" with value "added_book", "Author" with value "added_author", "Year" with value "2089", and "ISBN" with value "9090883". A blue "Submit" button is located at the bottom right of the form.



The screenshot shows a Windows desktop with a taskbar at the bottom. Below the taskbar is a table listing books. The columns are: ID, Title, Author, Year, ISBN, and Actions (Edit, Copy, Delete). The table contains the following data:

ID	Title	Author	Year	ISBN	Action
10	Moby-Dick	Herman Melville	0000	978990	
11	The Alchemist	Paulo Coelho	1988	611204	
12	Harry Potter and the Philosopher's Stone	J.K. Rowling	1997	978266	
13	The Odyssey	Homer	0000	978012	
14	added_book	added_author	2089	9090883	

ID	Title	Author	Year	ISBN	Action
51	Moby-Dick	Herman Melville	0000	978990	
52	The Alchemist	Paulo Coelho	1988	611204	
53	Harry Potter and the Philosopher's Stone	J.K. Rowling	1997	978266	
54	The Odyssey	Homer	0000	978012	
55	added_book	added_author	2089	9090883	

Code snippet:

addbook.blade.php

```
<!doctype html>
<html lang="en">
<head>
<title>Title</title>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- Bootstrap CSS v5.2.1 -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
      integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/Ujpbcx/TYkiZhlZB6+fzT"
      crossorigin="anonymous">
</head>
<body>
<header>
<!-- place navbar here -->
</header>
<main>
</main>
<footer>
<!-- place footer here -->
</footer>
<!-- Bootstrap JavaScript Libraries -->
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
      integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
      crossorigin="anonymous">
</script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"
      integrity="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz"
      crossorigin="anonymous">
```

Lab #11(a)

```

</script>

<div class="container mx-auto w-50 rounded m-5 p-5 bg-dark text-white ">
    <h2 class="text-center">Add Book</h2>
    <form method="POST" action="{{ $url }}" class="px-4">
        @csrf
        <div class="mb-3">
            <label for="exampleInputEmail1" class="form-label">Name</label>
            <input type="text" value="{{ isset($edit) ? $r[0]->name : '' }}" name="name" autofocus class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
            <span>
            </span>
        </div>
        <div class="mb-3">
            <label for="exampleInputPassword1" class="form-label">Author</label>
            <input type="text" name="author" class="form-control" id="exampleInputPassword1">
        </div>
        <div class="mb-3">
            <label for="exampleInputPassword1" class="form-label">Year</label>
            <input type="text" name="year" class="form-control" id="exampleInputPassword1">
        </div>
        <div class="mb-3">
            <label for="exampleInputPassword1" class="form-label">ISBN</label>
            <input type="text" name="isbn" class="form-control" id="exampleInputPassword1">
        </div>
        <div class="btn d-flex justify-content-center ">
            <button type="submit" class="btn btn-primary">Submit</button>
        </div>
    </div></form>
</div>
</body>

```

</html>

DBCONTROLLER.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Support\Facades\DB;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
class dbcontroller extends Controller
{
    public function dbselect(Request $req)
    {
        $search=$req['data']??"";
        if($search!="")
        {
            $search=$search.'%';
            $r=DB::select('select * from bookinfo where name LIKE ?',[ $search]);
        }
        else{
            $r=DB::select('select * from bookinfo');
        }
        $data=compact('r','req');
        return view('view')->with($data);
    }
    public function delete($id)
    {
        $r=DB::delete('delete from bookinfo where id=?',[ $id]);
        return redirect('/');
    }
    public function edit($id,Request $req)
    {
```

Lab #11(a)

```
$row_affected=DB::update('update bookinfo set name=?,author=?,year=?,isbn=? where
id=?',[${req['name']},${req['author']},${req['year']},${req['isbn']}],$id]);

return redirect('/');

}

public function fetchdata($id)
{
    $r=DB::select('select * from bookinfo where id=?',[${id}]);

    $url=url('/edit').'/'.$id;

    $edit=true;

    $data=compact('r','edit','url');

    return view('add')->with($data);

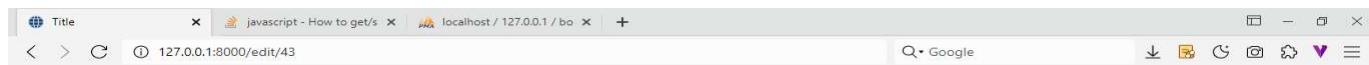
}
}
```

Lab #11(b)

Lab 11(b): Implement Update Operation in Laravel with GUI.

Final outcome:

1	let us c	yashwant kanedkar	2004	221121		
2	To Kill a Mockingbird	Harper Lee	1960	9711200		
3	1984	George Orwell	1949	978988		



Add Book

Name

Author

Year

ISBN



Add Book

Name
To Kill a Mockingbird

Author
Harper Lee

Year
1999 Changing year to 1999

ISBN
9711200

Submit

Add + Search Book Search

Sr_No	name	Author	Published Year	ISBN	Actions
1	let us c	yashwant kanedkar	2004	221121	Edit Delete
2	To Kill a Mockingbird	Harper Lee	1999	9711200	Edit Delete

← ↑ → id name author year isbn

	<input type="checkbox"/>	Edit	Copy	Delete	1 let us c	yashwant kanedkar	2004	221121
	<input type="checkbox"/>	Edit	Copy	Delete	43 To Kill a Mockingbird	Harper Lee	1999	9711200
	<input type="checkbox"/>	Edit	Copy	Delete	44 1984	George Orwell	1949	978988

Code snippet:

```
public function edit($id,Request $req)
{
    $row_affected=DB::update('update bookinfo set name=?,author=?,year=?,isbn=? where id=?',[ $req['name'],$req['author'],$req['year'],$req['isbn'],$id ]);
    return redirect('/');
}

public function fetchdata($id)
{
    $r=DB::select('select * from bookinfo where id=?',[ $id ]);
    $url=url('/edit').'/'. $id;
    $edit=true;
    $data=compact('r','edit','url');
    return view('add')->with($data);
}
```

Web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\dbcontroller;

Route::get('/', [dbcontroller::class,'dbselect']);
Route::get('/add', function()
{
    $url=url('/add');
```

Lab #11(b)

```
$data=compact('url');
return view('add')->with($data);
}
);

Route::get('/delete/{id}',[dbcontroller::class,'delete'])->name('book.delete');
Route::get('/edit/{id}',[dbcontroller::class,'fetchdata']);
Route::post('/edit/{id}',[dbcontroller::class,'edit'])->name('book.edit');
Route::post('/add',[dbcontroller::class,'add']);
```

Lab #12(a)
Lab#12(a) Implement Delete Operation in Laravel with GUI.

COMMAND USED	ATTRIBUTES USED	DESCRIPTION
Artisan	make:controller, make:model, make:migration, serve	In Laravel, Artisan is a command-line tool that provides a wide range of commands for various tasks related to the development and management of Laravel applications. Artisan simplifies common development tasks by automating them through command-line commands.

Final Outcome:

Sr_No	name	Author	Published Year	ISBN	Actions
1	To Kill a Mockingbird	Harper Lee	1999	9711200	 
2	1984	George Orwell	1949	978988	 
3	The Great Gatsby	F. Scott Fitzgerald	1925	735659	 

Sr_No	name	Author	Published Year	ISBN	Actions
1	1984	George Orwell	1949	978988	 
2	The Great Gatsby	F. Scott Fitzgerald	1925	735659	 
3	Pride and Prejudice	Jane Austen	0000	9143951	 

←	→	▼	id	name	author	year	isbn
<input type="checkbox"/>	 Edit	 Copy	 Delete	43 To Kill a Mockingbird	Harper Lee	1999	9711200
<input type="checkbox"/>	 Edit	 Copy	 Delete	44 1984	George Orwell	1949	978988
<input type="checkbox"/>	 Edit	 Copy	 Delete	45 The Great Gatsby	F. Scott Fitzgerald	1925	735659

←	→	▼	id	name	author	year	isbn
<input type="checkbox"/>	 Edit	 Copy	 Delete	44 1984	George Orwell	1949	978988
<input type="checkbox"/>	 Edit	 Copy	 Delete	45 The Great Gatsby	F. Scott Fitzgerald	1925	735659
<input type="checkbox"/>	 Edit	 Copy	 Delete	46 Pride and Prejudice	Jane Austen	0000	9143951

Code Snippet:
DBCONTROLLER.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Support\Facades\DB;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class dbcontroller extends Controller
{
    public function dbselect(Request $req)
    {
        $search=$req['data']??"";
        if($search!="")
        {
            $search=$search.'%';
            $r=DB::select('select * from bookinfo where name LIKE ?',[ $search]);
        }
        else{
            $r=DB::select('select * from bookinfo');
        }
        $data=compact('r','req');
        return view('view')->with($data);
    }

    public function delete($id)
    {
        $r=DB::delete('delete from bookinfo where id=?',[ $id]);
        return redirect('/');
    }

    public function edit($id,Request $req)
    {
```

Lab #12(a)

```

$row_affected=DB::update('update bookinfo set name=?,author=?,year=?,isbn=? where
id=?',[${req['name']},${req['author']},${req['year']},${req['isbn']}],$id]);
return redirect('/');
}

public function fetchdata($id)
{
    $r=DB::select('select * from bookinfo where id=?',[${id}]);
    $url=url('/edit').'/'.$id;
    $edit=true;
    $data=compact('r','edit','url');
    return view('add')->with($data);
}

```

web.php

```

<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\dbcontroller;
Route::get('/', [dbcontroller::class,'dbselect']);
Route::get('/add', function()
{
    $url=url('/add');
    $data=compact('url');
    return view('add')->with($data);
});
Route::get('/delete/{id}',[dbcontroller::class,'delete'])->name('book.delete');
Route::get('/edit/{id}',[dbcontroller::class,'fetchdata']);
Route::post('/edit/{id}',[dbcontroller::class,'edit'])->name('book.edit');
Route::post('/add',[dbcontroller::class,'add']);

```

Lab #12(b)
Lab#12(b) Search Operation in Laravel with GUI.
Final outcome:


Sr. No	name	Author	Published Year	ISBN	Actions
1	1984	ret	2000	43563	


Code snippet:
Web.php

```
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\dbcontroller;

Route::get('/', [dbcontroller::class,'dbselect']);

Route::get('/add', function()
{
  $url=url('/add');
```

Lab #12(b)

```
$data=compact('url');
return view('add')->with($data);
}
);
Route::get('/delete/{id}',[dbcontroller::class,'delete'])->name('book.delete');
Route::get('/edit/{id}',[dbcontroller::class,'fetchdata']);
Route::post('/edit/{id}',[dbcontroller::class,'edit'])->name('book.edit');
Route::post('/add',[dbcontroller::class,'add']);
```

View.blade.php:

```
public function dbselect(Request $req)
{
    $search=$req['data']??"";
    if($search!="")
    {
        $search=$search.'%';
        $r=DB::select('select * from bookinfo where name LIKE ?',[ $search]);
    }
    else{
        $r=DB::select('select * from bookinfo');
    }
    $data=compact('r','req');
    return view('view')->with($data);
}
```

Lab #13(a)

Lab#13(a) Setup a Node.js project on Student Information System

Final Outcome:

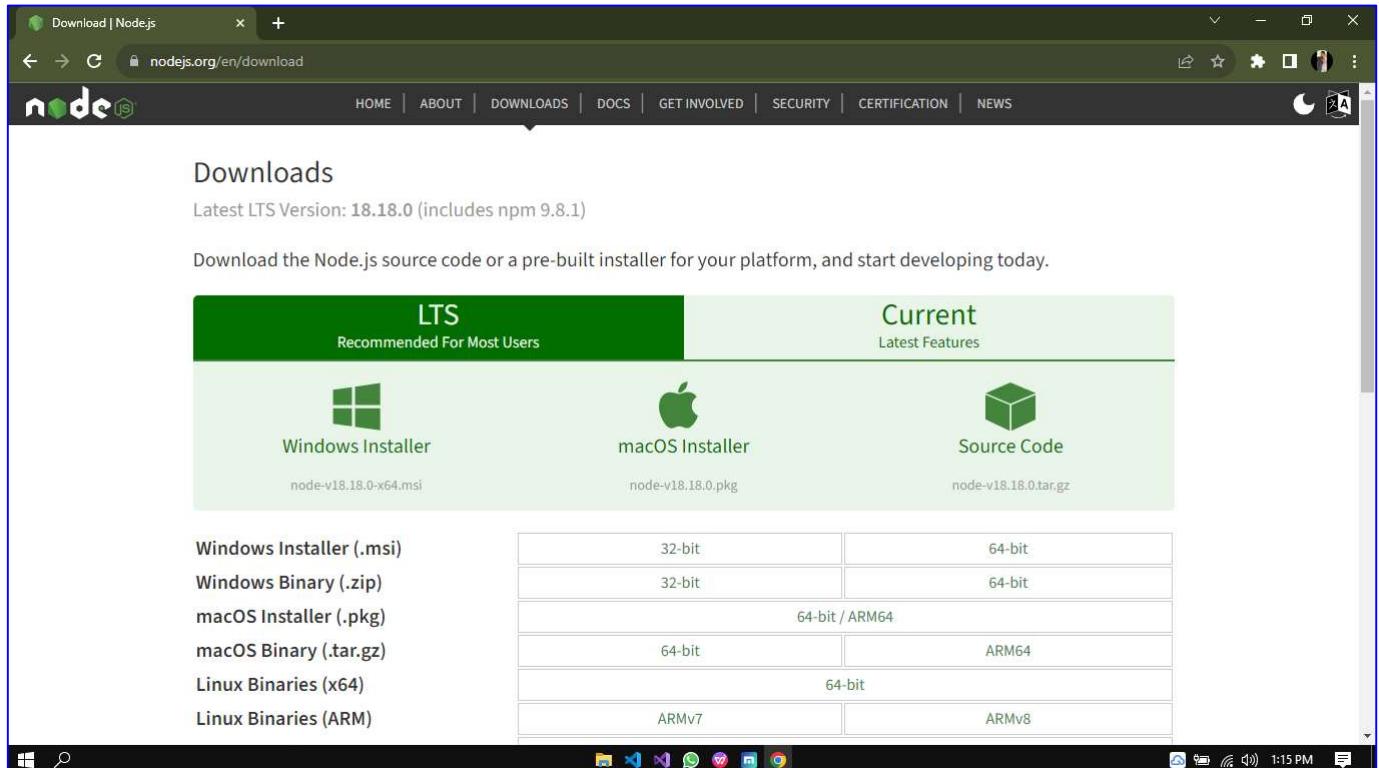
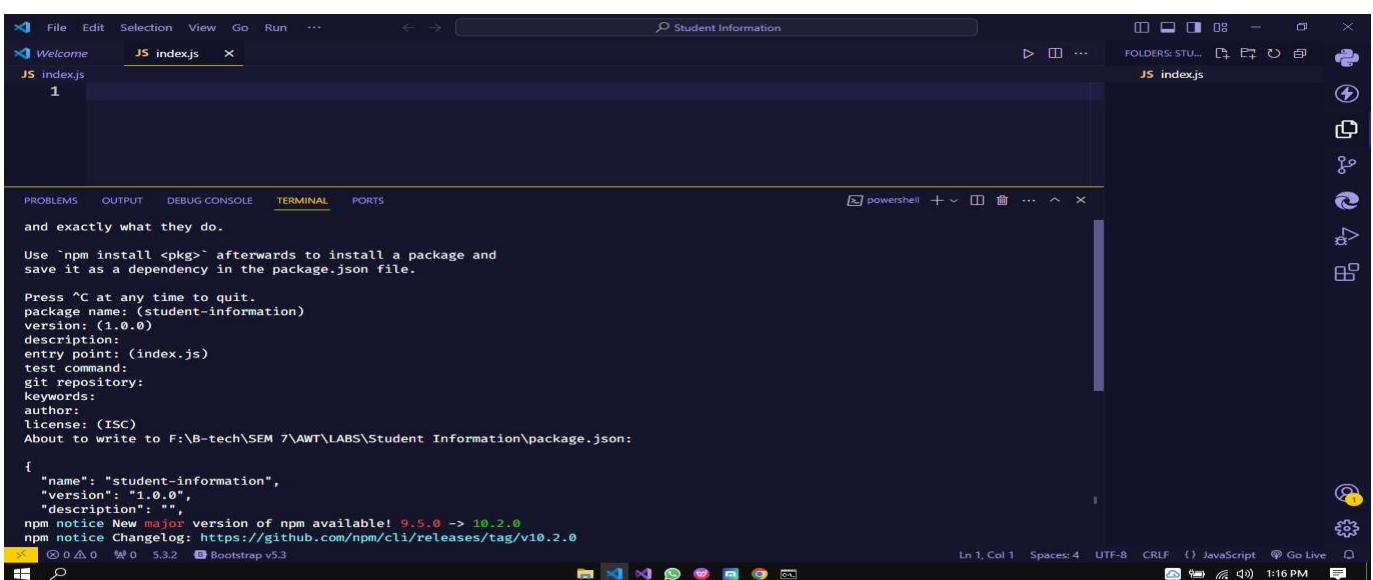


Figure 1 - Downloading Node.js



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. A terminal window is open at the bottom, displaying the following command sequence:

```

File Edit Selection View Go Run Run ... ← → ⌘P Student Information
Welcome JS index.js
JS index.js 1
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + ... ^ x
and exactly what they do.
Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.
Press ^C at any time to quit.
package name: (student-information)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to F:\B-tech\SEM 7\AWT\LABS\Student Information\package.json:
{
  "name": "student-information",
  "version": "1.0.0",
  "description": "",
  npm notice New major version of npm available! 9.5.0 -> 10.2.0
  npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.0
  Bootstrap v5.3.2

```

The terminal output indicates that a new package.json file is being created in the 'student-information' directory, listing the project's name, version, and other basic metadata. The status bar at the bottom right shows the current date and time as 1:16 PM.

Figure 2 - Creating a folder for index.js file

Steps:

1. Download Nodejs from **nodejs.org/eng/download** and install it
2. Create a folder name as “**Student Information**” and create “**Index.js**” file in it.
3. Execute command “**npm init**” to initialize node.js project.
4. Install **mysql** using npm install mysql.
5. Install **express** using npm install express.

Lab #13(b)
Lab#13(b) Setup the database in Node.js in Student Information System.
Final Outcome:

Table: students									
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	roll_no	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	course	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	semester	int(11)			No	None			Change Drop More
5	city	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

Check all With selected:
 Browse
 Change
 Drop
 Primary
 Unique
 Index
 Spatial
 Fulltext

Code Snippet:

```

const mysql = require('mysql')

const express = require('express')

const app = express()

app.use(express.json())

const conn = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'student_info'
})
  
```

Lab#13(c) Implement Select Operation to fetch details of students from Student Information System.

Final Outcome:



The screenshot shows a web-based student management system. At the top, there is a header bar with a logo of a butterfly, the text "STUDENT MANAGEMENT SYSTEM", "STUDENTS", and "ADD STUDENT". Below the header, the title "List of Students" is displayed. A table lists two students with columns for ID, First Name, Last Name, Gender, Date of Birth, Address, City, State, Zipcode, Email, Phone Number, Guardian Name, Guardian Phone, and Action (Edit and Delete buttons). The data in the table is as follows:

ID	First Name	Last Name	Gender	Date of Birth	Address	City	State	Zipcode	Email	Phone Number	Guardian Name	Guardian Phone	Action
2	abc	xyz	male	2002-09-08	xyxxy	abc	abxczcz	00000	abc@gmail.com	09876547	qabc	0987654333	<button>Edit</button> <button>Delete</button>
3	abc	xyz	male	2023-10-13	palce	city	udsdudu	09083	abc@gmail.com	7890456789	ababndbs	34567890	<button>Edit</button> <button>Delete</button>

Quick Links

- Home
- About Us
- Contact Us
- Register
- Login

Our Contact

Address: 360003, India, Rajkot, Gujarat
 Phone: (123) 456-7890
[Email: send me email](#)

Connect With Us



Code snippet:

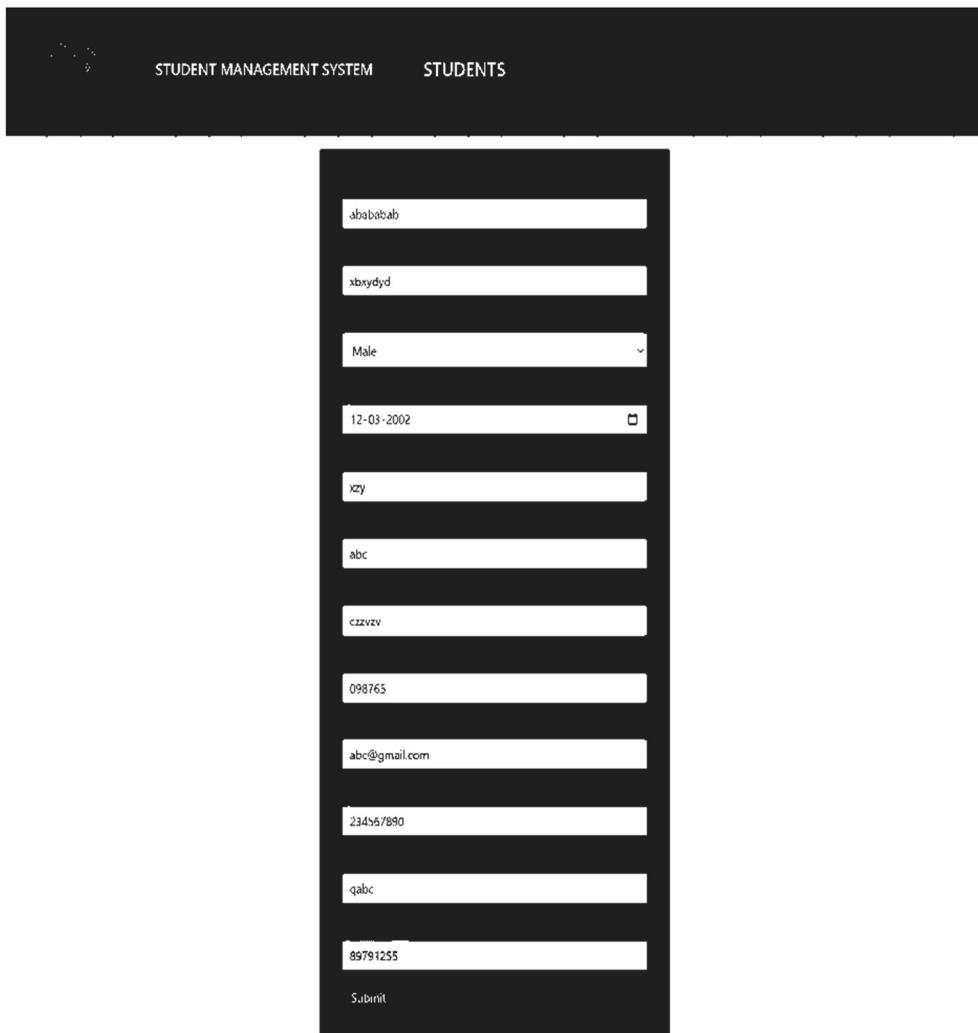
```
// Route handler to list all students
exports.listStudents = (req, res) => {
  db.all("SELECT * FROM students", (err, students) =>
    { if (err) {
      console.error(err);
      res.status(500).send("Internal Server Error");
      return;
    }
  }
}
```

Lab #13(c)

```
if (students.length === 0) {  
    // Handle the case where no students are  
    found res.status(404).send("No Students  
    Found"); return;  
}
```

Lab#14(a) Implement Insert Operation to add new record of student in Student Information System

Final Outcome:



The screenshot shows a web-based application titled "STUDENT MANAGEMENT SYSTEM" with a sub-section titled "STUDENTS". The form consists of several input fields:

- Text input: abababab
- Text input: xbxydyd
- Select dropdown: Male
- Date input: 12-03-2002
- Text input: xyz
- Text input: abc
- Text input: czzvzv
- Text input: 098765
- Email input: abc@gmail.com
- Text input: 234567890
- Text input: qabc
- Text input: 89781255

At the bottom right of the form is a "Submit" button.

Lab #14(a)

Code Snippet:

```
// Route handler to create a new student
exports.createStudent = (req, res) => {
  // Extract student data from req.body and insert it into the SQLite database
  const {
    first_name,
    last_name,
    gender,
    date_of_birth,
    address,
    city,
    state,
    zipcode
  ,
    email,
    phone_number,
    guardian_name,
    guardian_phone,
  } = req.body;

  const stmt = db.prepare(
    "INSERT INTO students (first_name, last_name, gender, date_of_birth, address, city, state, zipcode, email, phone_number, guardian_name, guardian_phone) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
  );

  stmt.run([
    first_name,
    last_name,
    gender,
    date_of_birth,
    address,
    city, state,
    zipcode, email,
    phone_numbe
    r,
    guardian_nam
    e,
    guardian_ph
    ne,
  ],
  (err) => {
    if (err) {
```

Lab #14(a)

```
console.error(err);
res.status(500).send("Internal Server Error");
return;

res.redirect("/sims/all");
}

);

stmt.finalize();
};

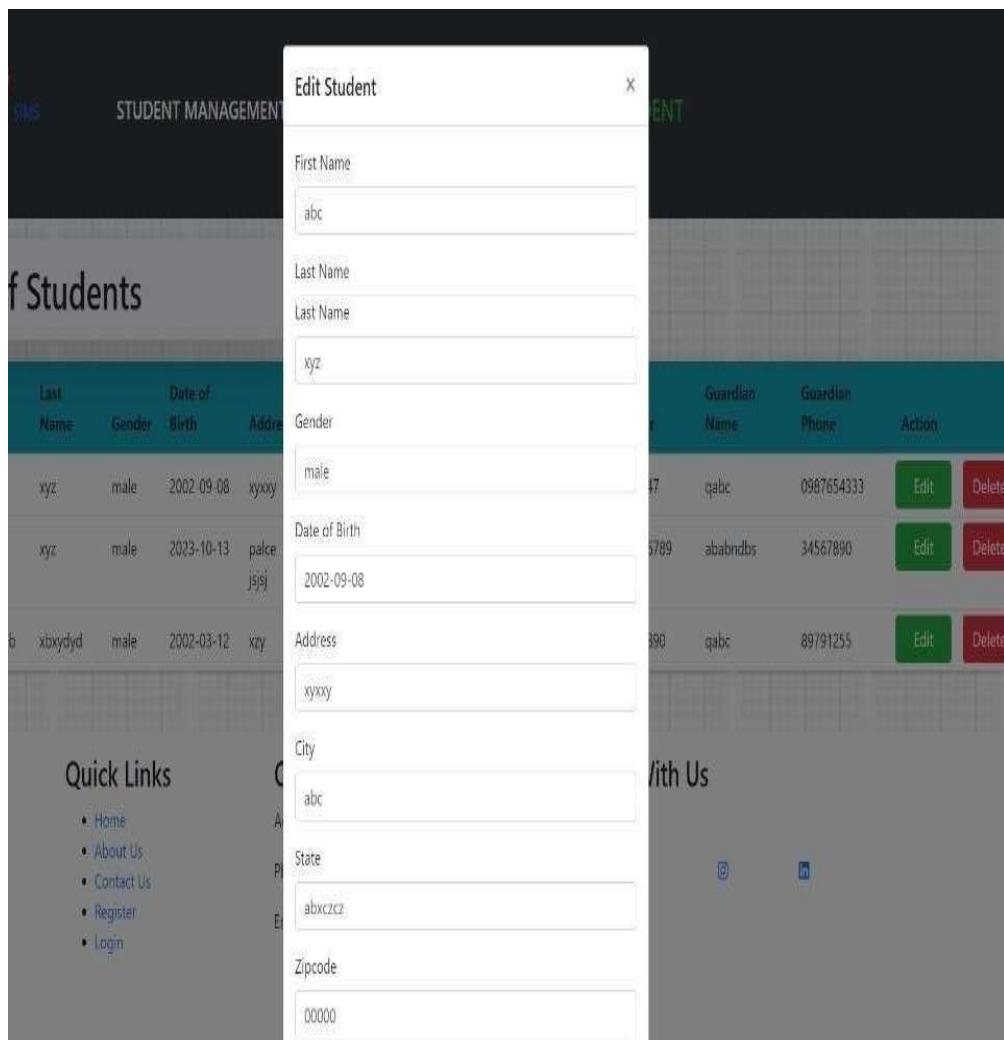
//delete student from database exports.deleteStudent = (req, res) => { const id = req.params.id; const
stmt = db.prepare("DELETE FROM students WHERE id = ?");
stmt.run([id], (err) => {
if (err) {
console.error(err);
res.status(500).send("Internal Server Error"); return;
}

res.redirect("/sims/all");
});

stmt.finalize();
};
```

Lab#14(b) Implement Update Operation to modify any record of student in Student Information System

Final Outcome:



Code Snippet:

```
// Route handler to update a student by ID
exports.updateStudent = (req, res) => {
  const studentId = req.params.id;

  const {
```

Lab #14(b)

```

first_name,
last_name,
gender,
date_of_birth,
address,
city,
state,
zipcode,
email,
phone_number,
guardian_name,
guardian_phone,
} = req.body;

// Begin a transaction db.run("BEGIN
TRANSACTION", function (beginErr) { if (beginErr)
{
  console.error(beginErr);
  res.status(500).send("Internal Server Error");
  return;
}

const stmt = db.prepare(
  "UPDATE Students SET first_name = ?, last_name = ?, gender = ?, date_of_birth = ?, address = ?, city
= ?, state = ?, zipcode = ?, email = ?, phone_number = ?, guardian_name = ?, guardian_phone = ? WHERE
id = ?"
);
stmt.run(
  first_name,
  last_name,
  gender,
  date_of_birth,
  address,
  city,
  state,
  zipcode
,
  email,
  phone_number,
  guardian_name,
  guardian_phone,
  studentId,
  (err) => { if (err) {

```

Lab #14(b)

```
console.error(err);
// Rollback the
transaction in case
of an error
db.run("ROLLBACK",
", (rollbackErr) => {
if (rollbackErr) {
console.error(rollba
ckErr);
}
res.status(500).sen
d("Internal Server
Error");
});
} else {
// Commit the
transaction
db.run("COMMIT",
(commitErr) => {
if (commitErr) {
console.error(com
mitErr);
res.status(500).sen
d("Internal Server
Error");
} else {
console.log("Studen
t updated
successfully.");
res.redirect("/sims
/all");
}
});
}
}
});
```

stmt.finalize();
});

Lab#14(c). Implement Delete Operation to remove any student from Student Information System.

Final Outcome:



ID	First Name	Last Name	Gender	Date of Birth	Address	City	State	Zipcode	Email	Phone Number	Guardian Name	Guardian Phone	Action
2	abc	xyz	male	2002-09-08	xyxx	abc	abxczcz	00000	abc@gmail.com	09876547	qabc	0987654333	<button>Edit</button> <button>Delete</button>
3	abc	xyz	male	2023-10-13	palce jsjsj	citry 26262	udscludu	09083	abc@gmail.com	7890456789	ababndbs	34567890	<button>Edit</button> <button>Delete</button>

Quick Links

- Home
- About Us
- Contact Us
- Register
- Login

Our Contact

Address: 360003, India, Rajkot, Gujarat
 Phone: (123) 456-7890
 Email: [send me email](#)

Connect With Us



© 2021 Niyi Innocent. All Rights Reserved.

Code Snippet:

```
//delete student from database
exports.deleteStudent = (req, res) => {
  const id = req.params.id;
  const stmt = db.prepare("DELETE FROM students WHERE id = ?");

  stmt.run([id], (err) => {
    if (err) {
      console.error(err);
      res.status(500).send("Internal Server Error");
      return;
    }

    res.redirect("/sims/all");
  });
}
```

```
stmt.finalize();  
};
```