

EXP NO: 1	WRITE A HTML PROGRAM FOR CREATION OF WEBSITE USING FRAME, FORM, LINK AND TABLE
DATE: 1.2.25	

AIM:

To write a html program for creation of forms links and tables

ALGORITHM:

Step 1: Start the HTML document using `<!DOCTYPE html>` and open `<html>` and `<head>` tags.

Step 2: Set the title of the webpage using the `<title>` tag inside `<head>`.

Step 3: Open the `<body>` tag to begin adding visible content.

Step 4: Create a form using the `<form>` tag with action and method attributes.

Step 5: Add input fields such as `<input type="text">` and `<input type="email">` inside the form.

Step 6: Include a submit button using `<input type="submit">`.

Step 7: Create hyperlinks using `Link Text`.

Step 8: Design a table using `<table>`, and add rows with `<tr>`, headers with `<th>`, and data with `<td>`.

Step 9: Close all opened tags properly: `</form>`, `</table>`, `</body>`, and `</html>`.

SOURCE CODE:

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML Forms, Links, and Tables Example</title>

</head>

<body {
  font-family: Arial, sans-serif;
  margin: 20px;
}

table {
  border-collapse: collapse;
  width: 60%;
  margin-top: 20px;
}

table, th, td {
  border: 1px solid #444;
```

```

    }
    th, td {
        padding: 10px;
        text-align: left;
    }
    form {
        margin-bottom: 20px;
    }
</style>
</head>
<body>

<h2>Registration Form</h2>
<form action="#" method="post">
    <label for="name">Name:</label><br>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">Email:</label><br>
    <input type="email" id="email" name="email" required><br><br>

    <input type="submit" value="Register">
</form>

<h2>Useful Links</h2>
<ul>
    <li><a href="https://www.w3schools.com" target="_blank">Visit W3Schools</a></li>
    <li><a href="https://www.mozilla.org" target="_blank">Visit Mozilla</a></li>
</ul>

<h2>Participant Table</h2>
<table>
    <tr>

```

```

        <th>S.No</th>

        <th>Name</th>

        <th>Email</th>

    </tr>

    <tr>

        <td>1</td>

        <td>Alice Johnson</td>

        <td>alice@example.com</td>

    </tr>

    <tr>

        <td>2</td>

        <td>Bob Smith</td>

        <td>bob@example.com</td>

    </tr>

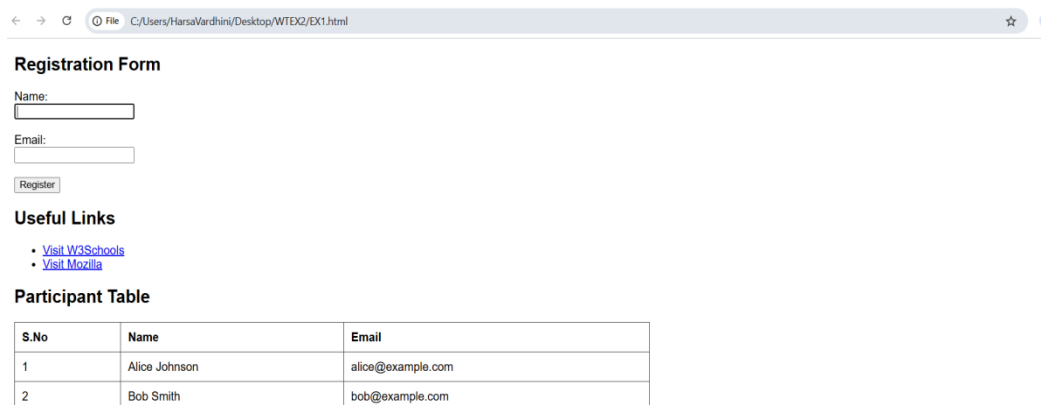
</table>

</body>

</html>

```

OUTPUT:



Registration Form

Name:

Email:

Useful Links

- [Visit W3Schools](#)
- [Visit Mozilla](#)

Participant Table

S.No	Name	Email
1	Alice Johnson	alice@example.com
2	Bob Smith	bob@example.com

RESULT:

Thus, the HTML webpage containing hyperlinks, forms, and tables is successfully created. The functionalities were implemented and verified with proper structure and formatting using basic HTML tags.

EXP NO: 2	DESIGN A WEBSITE USING HTML WITH BASIC TEXT FORMATTING AND IMAGES
DATE: 8.2.25	

AIM:

To create a website using html to create a basic textformatting and images

ALGORITHM:

Step 1: Start the HTML document using `<!DOCTYPE html>` and open `<html>`, `<head>`, and `<body>` tags.

Step 2: Set the character encoding and viewport settings using `<meta>` tags inside `<head>`.

Step 3: Add the title of the webpage using the `<title>` tag.

Step 4: Create a main heading using the `<h1>` tag.

Step 5: Add multiple paragraphs using `<p>`, and apply formatting tags like ``, `<i>`, `<u>`, `<mark>`, ``, and `<small>`.

Step 6: Insert a subheading using `<h2>` and display an image using the `` tag with `src` and `alt` attributes.

Step 7: Create another subheading and add a hyperlink using the `<a>` tag with `href` and `target="_blank"`.

Step 8: Add a bulleted list using the `` tag with items inside `` tags.

Step 9: Close all opened tags properly, including `</body>` and `</html>`.

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Welcome to My Creative Webpage</title>

</head>

<body>

  <!-- Heading -->

  <h1>Discover the Beauty of Simplicity</h1>

  <!-- Paragraphs with text formatting -->
```

<p>Welcome to a space where creativity meets simplicity. This webpage showcases how <i>beautiful design</i> can be achieved with <u>basic HTML elements</u>.</p>

<p>Every great journey begins with a single step. Just like <mark>every well-crafted webpage</mark> starts with a structured layout and a creative touch.</p>

<p><i>Innovation</i> is not about complexity; it's about making things <small>meaningful</small> and impactful. Overthinking is unnecessary—just start building!</p>

<!-- Image -->

<h2>A Glimpse of Elegance</h2>

<!-- Hyperlink -->

<h2>Stay Inspired</h2>

<p>Explore more amazing content at Example Website and keep learning!</p>

<!-- List -->

<h2>Keys to a Stunning Webpage</h2>

Minimalism - Less is more.

Typography - Choose fonts wisely.

Visual Balance - Keep it clean and structured.

</body>

</html>

OUTPUT:

Discover the Beauty of Simplicity

Welcome to a space where creativity meets simplicity. This webpage showcases how *beautiful design* can be achieved with basic HTML elements.

Every great journey begins with a single step. Just like **every well-crafted webpage** starts with a structured layout and a creative touch.

Innovation is not about complexity; it's about making things meaningful and **impactful**. ~~Overthinking~~ is unnecessary—just start building!

A Glimpse of Elegance



Stay Inspired

Explore more amazing content at [Example Website](#) and keep learning!

Keys to a Stunning Webpage

- Minimalism - Less is more.
- Typography - Choose fonts wisely.
- Visual Balance - Keep it clean and structured.

RESULT:

Thus, the HTML webpage title is successfully created. It demonstrates the use of text formatting tags, image embedding, hyperlinks, and unordered lists to build a clean and creative webpage layout using basic HTML elements.

EXP NO: 3	CREATE A WEBPAGE WITH IMAGE MAP AND HOTSPOTS TO SHOW
DATE: 13.2.25	

i)To embed an image in a webpage

ii)To fix the hotspot

iii)Show all the related information when the hotspot is clicked

AIM:

To create a webpage using HTML5 that embeds an image with interactive hotspots and displays related information upon clicking them.

ALGORITHM:

Step 1: Start the HTML document with `<!DOCTYPE html>` and open `<html>` and `<head>` tags.

Step 2: Set the character encoding and viewport using `<meta>` tags inside the `<head>` tag.

Step 3: Define the title of the webpage with `<title>` tag.

Step 4: Inside the `<body>`, use the `` tag to embed the image and define the `usemap` attribute linking it to the image map.

Step 5: Define an image map using the `<map>` tag with a unique name and add `<area>` tags inside it.

Step 6: Set the coordinates of each hotspot using the `coords` attribute of the `<area>` tag and specify the `href` attribute to link to the related information.

Step 7: Ensure the image map's defined regions are clickable and properly configured to display or redirect information when clicked.

Step 8: Close all the tags properly with `</body>` and `</html>`.

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Image Map</title>

</head>

<body>
```

<h1>Interactive World Map</h1>

<p>Click on different continents to learn more.</p>

<!-- Image with an image map -->

<!-- Image Map Definition -->

<map name="worldmap">

<!-- Europe -->

<area shape="rect" coords="350,80,450,180" href="https://en.wikipedia.org/wiki/Europe" target="_blank" alt="Europe">

<!-- Asia -->

<area shape="rect" coords="460,80,660,280" href="https://en.wikipedia.org/wiki/Asia" target="_blank" alt="Asia">

<!-- Africa -->

<area shape="rect" coords="370,200,500,380" href="https://en.wikipedia.org/wiki/Africa" target="_blank" alt="Africa">

<!-- North America -->

<area shape="rect" coords="50,50,250,250" href="https://en.wikipedia.org/wiki/North_America" target="_blank" alt="North America">

<!-- South America -->

<area shape="rect" coords="180,280,300,450" href="https://en.wikipedia.org/wiki/South_America" target="_blank" alt="South America">

<!-- Australia -->

<area shape="rect" coords="650,320,780,450" href="https://en.wikipedia.org/wiki/Australia" target="_blank" alt="Australia">

</map>

<p>Clicking on any hotspot will take you to Wikipedia for more details about that continent.</p>

</body>

</html>

OUTPUT:



Interactive World Map

Click on different continents to learn more.



Clicking on any hotspot will take you to Wikipedia for more details about that continent.

EXP NO: 4	CREATE A WEBPAGE WITH ALL TYPES OF CSS
DATE: 15.2.25	

AIM:

To design a professional webpage using **HTML5** and **CSS** (internal, external, and inline) that showcases various CSS features including styling, layout, and responsiveness.

ALGORITHM:

Step 1: Start the HTML document with `<!DOCTYPE html>` and open `<html>`, `<head>`, and `<body>` tags.

Step 2: Set metadata such as character encoding and viewport using `<meta>` tags inside the `<head>`.

Step 3: Define the title of the webpage using the `<title>` tag.

Step 4: Link an external CSS file using the `<link rel="stylesheet">` tag for global styles.

Step 5: Add internal CSS inside `<style>` tags to customize specific elements like headings and spans.

Step 6: Create a header section using `<header>` with a `<h1>` and `<p>` for the main title and subtitle.

Step 7: Embed an image in the hero section using `` and apply inline CSS for responsive sizing.

Step 8: Overlay text on the hero image using a `<div>` with styled `<h2>` and `<p>` elements.

Step 9: Add a content section with `<h2>` and `<p>` tags, using `` to emphasize text.

Step 10: Build a features section with multiple `<div class="feature-box">` blocks describing key CSS topics.

Step 11: Create a footer using `<footer>` and include a copyright notice.

Step 12: Close all open tags (`</body>`, `</html>`) to complete the webpage structure.

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Professional CSS Webpage</title>

  <!-- External CSS -->

  <link rel="stylesheet" href="stylesex4.css">
```

```
<!-- Internal CSS -->
<style>
  h2 {
    color: #2a2a2b;
    text-align: center;
    margin-top: 30px;
    font-size: 28px;
  }
  .highlight {
    background-color: yellow;
    padding: 5px;
    font-weight: bold;
  }
</style>
</head>
<body>

<!-- Header Section -->
<header>
  <h1>Welcome to the World of CSS</h1>
  <p>Mastering CSS for Modern Web Development</p>
</header>

<!-- Hero Section with Inline CSS -->
<section class="hero">
  
  <div class="hero-text">
    <h2 style="color: white; font-size: 35px;">CSS: The Heart of Web Styling</h2>
    <p style="color: white;">Discover how CSS enhances design, responsiveness, and animations.</p>
  </div>
</section>
```

```
<a href="#" class="btn">Explore More</a>

</div>

</section>

<!-- Content Section -->

<section class="content">

  <h2>Why Learn CSS?</h2>

  <p>CSS (Cascading Style Sheets) allows you to design professional web pages by
<span class="highlight">adding styles, animations, and responsiveness</span>.</p>

  <p>With CSS, you can create visually appealing layouts and improve user
experience.</p>

</section>

<!-- Features Section -->

<section class="features">

  <div class="feature-box">

    <h3>CSS Selectors</h3>

    <p>Target elements with different selectors for precise styling.</p>

  </div>

  <div class="feature-box">

    <h3>CSS Grid & Flexbox</h3>

    <p>Create dynamic layouts with ease.</p>

  </div>

  <div class="feature-box">

    <h3>CSS Animations</h3>

    <p>Enhance UI with animations and transitions.</p>

  </div>

</section>

<!-- Footer -->

<footer>
```

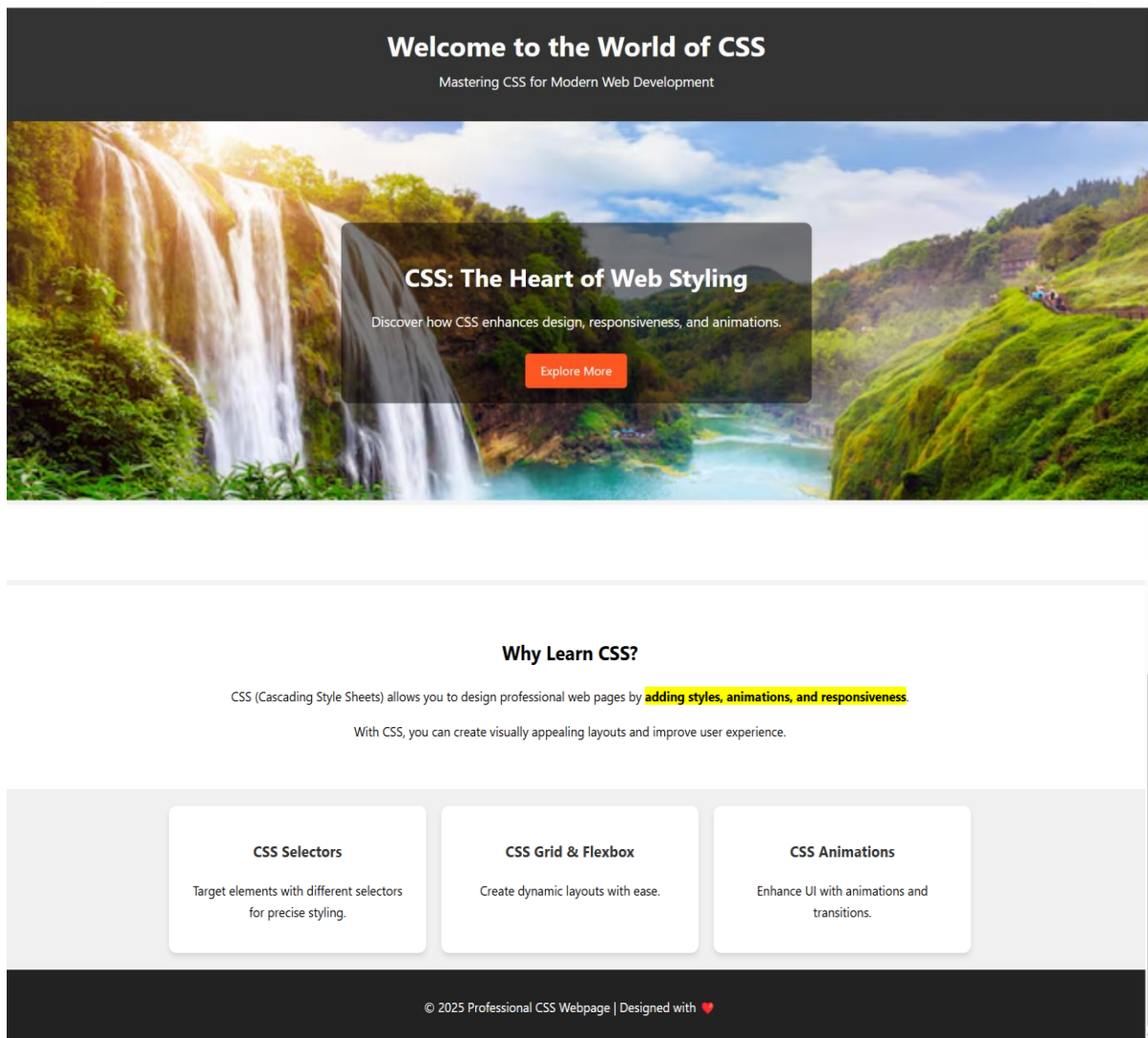
<p>© 2025 Professional CSS Webpage </p>

</footer>

</body>

</html>

OUTPUT:



RESULT:

Thus, the professional HTML5 webpage integrated with internal, external, and inline CSS is successfully created. The webpage demonstrates structured layout, custom styling, and responsiveness, showcasing core concepts of CSS effectively.

EXP NO: 5	DESIGN A SCIENTIFIC CALCULATOR USING HTML, CSS, AND JAVASCRIPT
DATE: 22.2.25	

AIM:

To design a Scientific Calculator using HTML, CSS, and JavaScript.

ALGORITHM:

- Step 1:** Create the HTML structure with a display and calculator buttons.
- Step 2:** Style the calculator layout and buttons using CSS for better user interface.
- Step 3:** Implement appendToDisplay(value) to add clicked button values to the input field.
- Step 4:** Implement clearDisplay() to reset the input display when needed.
- Step 5:** Implement calculateResult() to evaluate the mathematical expression.
- Step 6:** Use try-catch block in calculateResult() to handle invalid expressions.
- Step 7:** Add scientific functions like sqrt, pow, sin, cos, tan, log, exp, and pi.
- Step 8:** Assign onclick events to all calculator buttons to trigger JavaScript functions.

SOURCE CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    body {
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-color: #f4f4f4;
    }
    .calculator {
```



```
width: 300px;
background: #fff;
padding: 20px;
border-radius: 10px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
text-align: center;
}
input {
width: 100%;
height: 50px;
text-align: right;
font-size: 1.5em;
margin-bottom: 10px;
}
.buttons {
display: grid;
grid-template-columns: repeat(4, 1fr);
gap: 5px;
}
button {
height: 50px;
font-size: 1.2em;
border: none;
cursor: pointer;
background: #eee;
border-radius: 5px;
}
button:active {
background: #ddd;
}
```

```

.equal {
    background: #28a745;
    color: white;
}
.clear {
    background: #dc3545;
    color: white;
}
</style>
</head>
<body>
<div class="calculator">
    <input type="text" id="display" disabled>
    <div class="buttons">
        <button onclick="clearDisplay()" class="clear">C</button>
        <button onclick="appendToDisplay('(')">(</button>
        <button onclick="appendToDisplay(')')">)</button>
        <button onclick="appendToDisplay('/')">/</button>
        <button onclick="appendToDisplay('7')">7</button>
        <button onclick="appendToDisplay('8')">8</button>
        <button onclick="appendToDisplay('9')">9</button>
        <button onclick="appendToDisplay('*')">*</button>
        <button onclick="appendToDisplay('4')">4</button>
        <button onclick="appendToDisplay('5')">5</button>
        <button onclick="appendToDisplay('6')">6</button>
        <button onclick="appendToDisplay('-')">-</button>
        <button onclick="appendToDisplay('1')">1</button>
        <button onclick="appendToDisplay('2')">2</button>
        <button onclick="appendToDisplay('3')">3</button>
        <button onclick="appendToDisplay('+')">+</button>
    </div>
    </div>
</body>
</html>

```

```

    <button onclick="appendToDisplay('0')">0</button>
    <button onclick="appendToDisplay('.')">.</button>
    <button onclick="calculateResult()" class="equal">=</button>
    <button onclick="appendToDisplay('Math.sqrt()')"> $\sqrt{\phantom{x}}$ </button>
    <button onclick="appendToDisplay('Math.pow()')"> $x^y$ </button>
    <button onclick="appendToDisplay('Math.sin()')">sin</button>
    <button onclick="appendToDisplay('Math.cos()')">cos</button>
    <button onclick="appendToDisplay('Math.tan()')">tan</button>
    <button onclick="appendToDisplay('Math.log()')">log</button>
    <button onclick="appendToDisplay('Math.exp()')"> $e^x$ </button>
    <button onclick="appendToDisplay('Math.PI')"> $\pi$ </button>
</div>
</div>
</body>
<script>
    function appendToDisplay(value){
        document.getElementById("display").value+=value;
    }
    function clearDisplay(){
        document.getElementById("display").value="";
    }
    function calculateResult(){
        try{
            document.getElementById("display").value=
eval(document.getElementById("display").value);
        }
        catch(e){
            alert("Invalid Expression");
            clearDisplay();
        }
    }
}

```

</script>

</html>

OUTPUT:



RESULT:

Thus, a scientific calculator is designed using JavaScript successfully and verified.

EXP NO: 6	DESIGN A REGISTRATION FORM USING HTML, CSS, AND JAVASCRIPT VALIDATION(REGULAR EXPRESSION)
DATE: 1.3.25	

AIM:

To design a Registration Form using HTML, CSS (Bootstrap), and JavaScript validation.

ALGORITHM:

Step 1: Create the HTML structure with input fields for name, email, mobile, password, and confirm password.

Step 2: Use Bootstrap and custom CSS to style the form and improve user interface.

Step 3: Extract Head Size as X (independent variable) and Brain Weight as y (dependent variable).

Step 4: Use regular expressions to validate name, email, mobile number, and password formats.

Step 5: Check if password and confirm password fields match.

Step 6: Display corresponding error messages for invalid inputs dynamically.

Step 7: Prevent form submission if any validation fails and allow it if all inputs are valid.

SOURCE CODE:

```
<!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>

  <link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">

  <style>

    body {

      background-color: #f4f4f4;

      font-family: Arial, sans-serif;

    }

    .container {

      max-width: 450px;

      background: #fff;
```

```

padding: 20px;
border-radius: 8px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
margin-top: 50px;
}
.error {
color: red;
font-size: 14px;
}
</style>
</head>
<body>

<div class="container">
  <h3 class="text-center">Registration Form</h3>
  <form id="regForm" onsubmit="return validateForm()">

    <div class="mb-3">
      <label class="form-label">Full Name</label>
      <input type="text" class="form-control" id="name">
      <span class="error" id="nameError"></span>
    </div>

    <div class="mb-3">
      <label class="form-label">Email</label>
      <input type="email" class="form-control" id="email">
      <span class="error" id="emailError"></span>
    </div>

    <div class="mb-3">

```

```

        <label class="form-label">Mobile Number</label>
        <input type="text" class="form-control" id="mobile">
        <span class="error" id="mobileError"></span>
    </div>

    <div class="mb-3">
        <label class="form-label">Password</label>
        <input type="password" class="form-control" id="password">
        <span class="error" id="passwordError"></span>
    </div>

    <div class="mb-3">
        <label class="form-label">Confirm Password</label>
        <input type="password" class="form-control" id="confirmPassword">
        <span class="error" id="confirmPasswordError"></span>
    </div>

    <button type="submit" class="btn btn-primary w-100">Register</button>
</form>
</div>

<script>
function validateForm() {
    let valid = true;

    let name = document.getElementById("name").value.trim();
    let email = document.getElementById("email").value.trim();
    let mobile = document.getElementById("mobile").value.trim();
    let password = document.getElementById("password").value;
    let confirmPassword = document.getElementById("confirmPassword").value;

```

```

let nameRegex = /^[A-Za-z\s]{3,}$/;

let emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

let mobileRegex = /^[6-9]\d{9}$/;

let passwordRegex = /^(?=.*[A-Za-z])(?=.*\d).{6,}$/;


document.getElementById("nameError").innerText = nameRegex.test(name) ? "" :
"Name must be at least 3 letters";

document.getElementById("emailError").innerText = emailRegex.test(email) ? "" :
"Invalid email format";

document.getElementById("mobileError").innerText = mobileRegex.test(mobile) ? ""
: "Enter a valid 10-digit mobile number";

document.getElementById("passwordError").innerText =
passwordRegex.test(password) ? "" : "Min 6 chars with at least one letter & number";

document.getElementById("confirmPasswordError").innerText = password ===
confirmPassword ? "" : "Passwords do not match";


return nameRegex.test(name) && emailRegex.test(email) &&
mobileRegex.test(mobile) && passwordRegex.test(password) && password ===
confirmPassword;

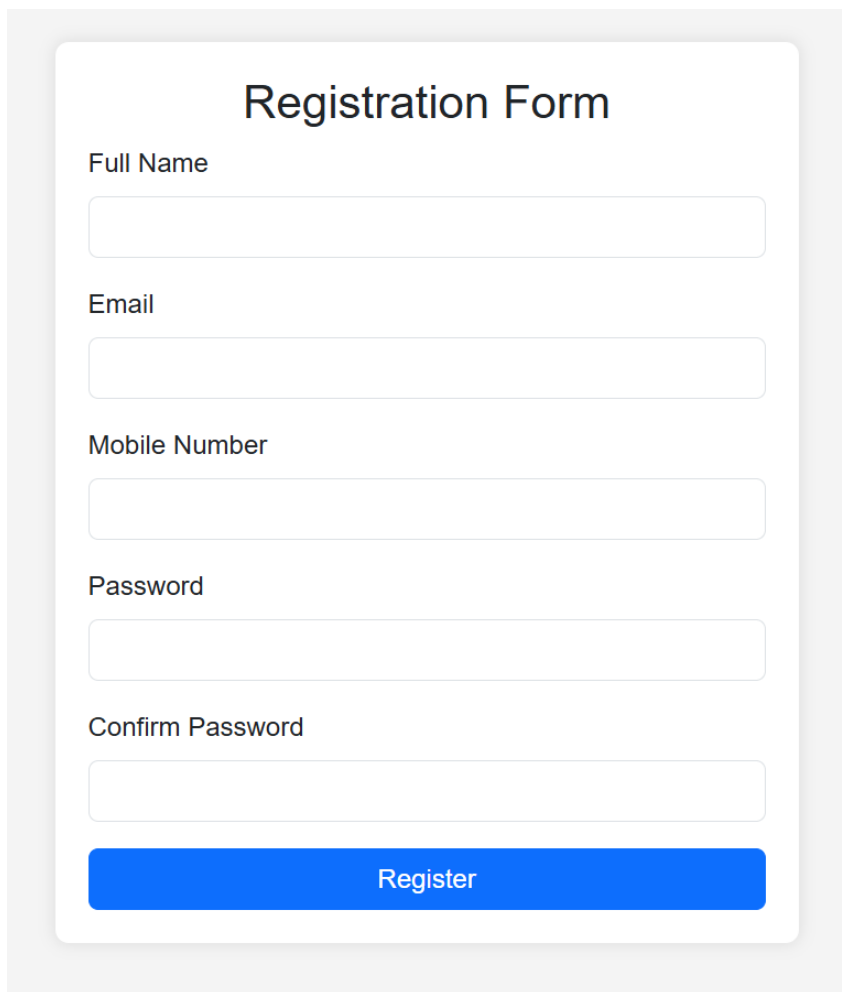
}

</script>


<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```


OUTPUT:



The image shows a registration form titled "Registration Form". It contains five input fields: "Full Name", "Email", "Mobile Number", "Password", and "Confirm Password". Each field is a white rectangle with a thin grey border. Below the "Confirm Password" field is a blue button with the text "Register" in white. The form is set against a light grey background.

RESULT:

Thus, a registration form is designed using HTML,CSS and JavaScript successfully and verified.

EXP NO: 7	A RESPONSIVE WEBPAGE USING BOOTSTRAP
DATE: 8.3.25	

AIM:

To design a fully responsive and modern web page using Bootstrap 5 and Font Awesome with smooth animations and an elegant layout.

ALGORITHM:

- Step 1:** Start with the HTML5 boilerplate structure
- Step 2:** Link Bootstrap CSS and Font Awesome icons from CDN
- Step 3:** Design a navigation bar using Bootstrap's navbar component.
- Step 4:** Create a hero section with a background image and animated welcome text.
- Step 5:** Build a features section using Bootstrap cards to showcase highlights.
- Step 6:** Create an image gallery using a Bootstrap grid layout with hover effects.
- Step 7:** Add a contact section with a call-to-action button linked to an email.
- Step 8:** Design a footer with social media icons and copyright information.
- Step 9:** Use Bootstrap's utility classes and custom CSS for styling, hover effects, and responsiveness.
- Step 10:** Link Bootstrap JavaScript at the bottom to enable collapsible navbar and other components.

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

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

  <title>Enhanced Responsive Web Page</title>

  <link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">

  <link      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
rel="stylesheet">

  <style>

    body {

      font-family: 'Segoe UI', sans-serif;
```

```

    scroll-behavior: smooth;

    background-color: #f8f9fa;
}

.navbar {
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

.hero {
    background: url('https://images.unsplash.com/photo-1519389950473-47ba0277781c') no-repeat center center/cover;
    color: white;
    padding: 120px 20px;
    text-align: center;
    animation: fadeIn 2s ease-in-out;
}

@keyframes fadeIn {
    from { opacity: 0; transform: translateY(20px); }
    to { opacity: 1; transform: translateY(0); }
}

.card:hover {
    transform: translateY(-5px);
    transition: transform 0.3s ease;
}

.feature-icon {
    font-size: 2.5rem;
    color: #0d6efd;
}

```

```
.gallery img {  
  width: 100%;  
  height: 250px;  
  object-fit: cover;  
  border-radius: 8px;  
  transition: transform 0.3s ease;  
}
```

```
.gallery img:hover {  
  transform: scale(1.05);  
}
```

```
.footer {  
  background-color: #343a40;  
  color: #fff;  
  padding: 30px 0;  
}
```

```
.social-icons i {  
  font-size: 1.5rem;  
  margin: 0 10px;  
  color: white;  
  transition: color 0.3s;  
}
```

```
.social-icons i:hover {  
  color: #0d6efd;  
}
```

```
</style>
```

```

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg navbar-dark bg-primary sticky-top">
  <div class="container">
    <a class="navbar-brand" href="#">MySite</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item"><a class="nav-link active" href="#home">Home</a></li>
        <li class="nav-item"><a class="nav-link" href="#features">Features</a></li>
        <li class="nav-item"><a class="nav-link" href="#gallery">Gallery</a></li>
        <li class="nav-item"><a class="nav-link" href="#contact">Contact</a></li>
      </ul>
    </div>
  </div>
</nav>

<!-- Hero Section -->

<section id="home" class="hero text-white">
  <div class="container">
    <h1 class="display-4 fw-bold">Welcome to My Beautiful Website</h1>
    <p class="lead">Fully responsive. Smooth, modern design with animations and rich
visuals.</p>
    <a href="#features" class="btn btn-light mt-3">Explore Features</a>
  </div>
</section>

```

```

<!-- Features Section -->
<section id="features" class="py-5">
  <div class="container">
    <h2 class="text-center mb-5">Amazing Features</h2>
    <div class="row g-4">
      <div class="col-md-4">
        <div class="card text-center p-3 shadow-sm h-100">
          <div class="card-body">
            <i class="fas fa-laptop-code feature-icon mb-3"></i>
            <h5 class="card-title">Responsive Design</h5>
            <p class="card-text">Adapts beautifully to mobile, tablet, and desktop screens.</p>
          </div>
        </div>
      </div>
      <div class="col-md-4">
        <div class="card text-center p-3 shadow-sm h-100">
          <div class="card-body">
            <i class="fas fa-magic feature-icon mb-3"></i>
            <h5 class="card-title">Modern Animations</h5>
            <p class="card-text">Smooth, eye-catching transitions for a sleek experience.</p>
          </div>
        </div>
      </div>
      <div class="col-md-4">
        <div class="card text-center p-3 shadow-sm h-100">
          <div class="card-body">
            <i class="fas fa-bolt feature-icon mb-3"></i>
            <h5 class="card-title">Fast & Lightweight</h5>
            <p class="card-text">Minimal and optimized code for faster performance.</p>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```
        </div>
    </div>
</div>
</div>
</div>
</section>
```

```
<!-- Gallery Section -->
<section id="gallery" class="py-5 bg-light">
    <div class="container">
        <h2 class="text-center mb-5">Image Gallery</h2>
        <div class="row g-4 gallery">
            <div class="col-md-4"></div>
            <div class="col-md-4"></div>
            <div class="col-md-4"></div>
        </div>
    </div>
</section>
```

```
<!-- Contact Section -->
<section id="contact" class="py-5">
    <div class="container text-center">
        <h2 class="mb-4">Get in Touch</h2>
        <p>Have questions or want to work together? Email me at <a href="mailto:hansi1122012@gmail.com">hansi1122012@gmail.com</a></p>
        <a href="mailto:hansi1122012@gmail.com" class="btn btn-primary mt-2">Contact Now</a>
    </div>
</section>
```

```
<!-- Footer -->

<footer class="footer text-center">

  <div class="container">

    <div class="social-icons mb-3">

      <a href="#"><i class="fab fa-facebook-f"></i></a>

      <a href="#"><i class="fab fa-twitter"></i></a>

      <a href="#"><i class="fab fa-instagram"></i></a>

      <a href="#"><i class="fab fa-github"></i></a>

    </div>

    <p class="mb-0">&copy; 2025 MySite. All rights reserved.</p>

  </div>

</footer>
```

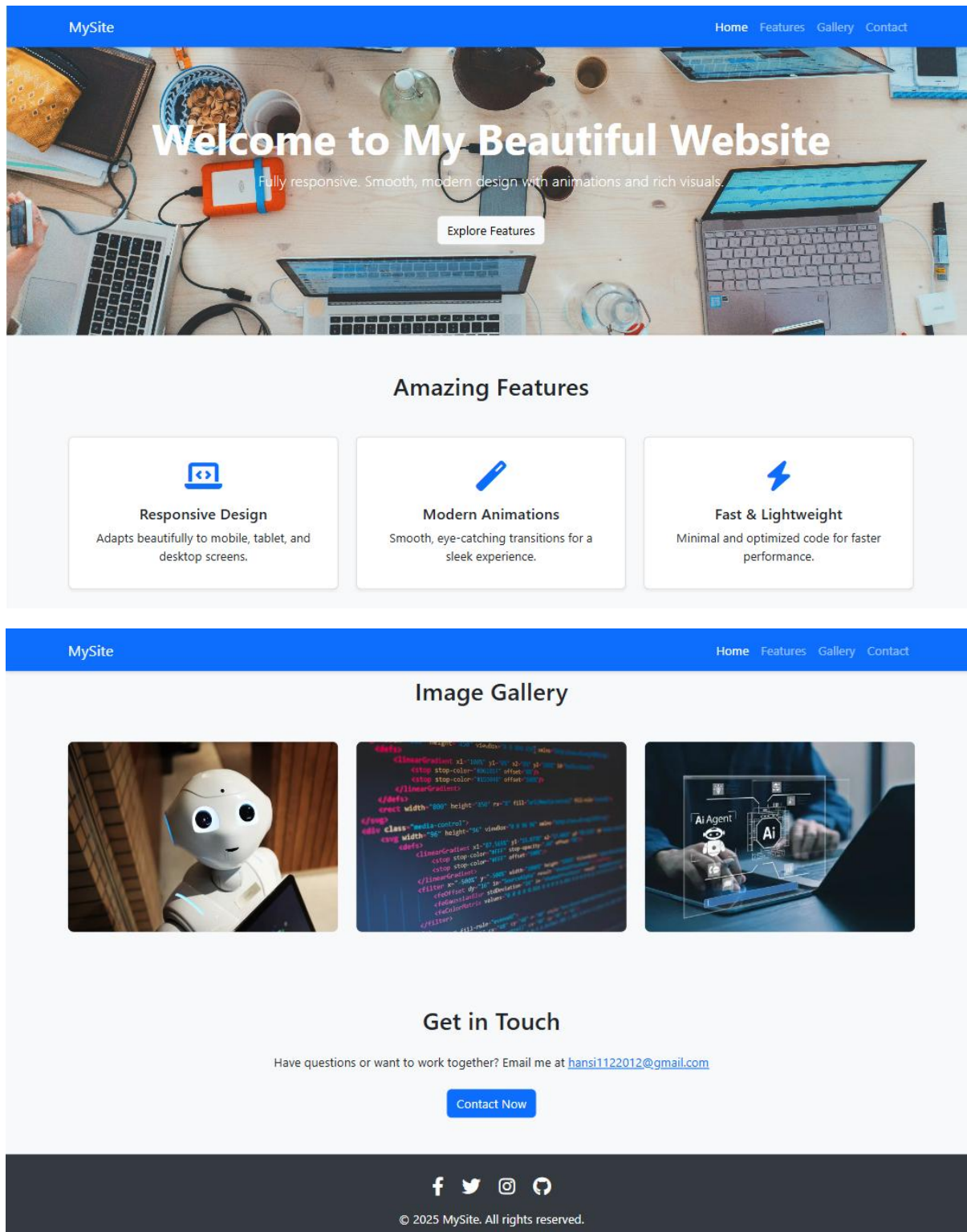
```
<!-- Bootstrap + Font Awesome -->

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>
```


OUTPUT:



RESULT:

Thus, a fully responsive and visually appealing web page was successfully designed using Bootstrap 5 and verified.

EXP NO: 8	A RESPONSIVE WEB PAGE USING BOOTSTRAP'S GRID SYSTEM
DATE: 22.3.25	

AIM:

To design a responsive web page using Bootstrap's grid system for adaptive layout.

ALGORITHM:

Step 1: Set up basic HTML structure with meta tags for responsiveness.

Step 2: Include Bootstrap CSS and JS libraries.

Step 3: Create a header with title and description.

Step 4: Define a container for grid-based content.

Step 5: Add a row with two columns for an image-text section.

Step 6: Add a row with three equal-width columns for cards.

Step 7: Add a row with four equal-width columns for smaller blocks.

Step 8: Include footer with copyright information.

Step 9: Test responsiveness across different screen sizes.

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

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

  <title>Bootstrap Grid Page</title>

  <link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">

  <style>

    body {

      font-family: 'Segoe UI', sans-serif;

    }

    .header {

      background-color: #007bff;

      color: white;

      padding: 40px 0;
```

```

        text-align: center;
    }
    .img-fluid {
        border-radius: 8px;
    }
    .card:hover {
        transform: translateY(-5px);
        transition: 0.3s ease;
    }
</style>
</head>
<body>

<!-- Header -->
<div class="header">
    <h1>Bootstrap Grid System</h1>
    <p>Responsive layout using rows and columns</p>
</div>

<!-- Grid Section -->
<div class="container py-5">
    <!-- Row 1 -->
    <div class="row mb-4">
        <div class="col-md-6">
            
        </div>
        <div class="col-md-6 d-flex align-items-center">
            <div>
                <h3>Responsive Columns</h3>
                <p>This is a 2-column layout using Bootstrap's grid. On smaller screens, it stacks vertically.</p>
            </div>
        </div>
    </div>

```

```
</div>

</div>

</div>

<!-- Row 2 (3-column cards) -->
<div class="row text-center">
  <div class="col-md-4 mb-4">
    <div class="card shadow-sm h-100">
      <div class="card-body">
        <h5 class="card-title">Column One</h5>
        <p class="card-text">This column spans 4/12 of the row on medium+ screens.</p>
      </div>
    </div>
  </div>
  <div class="col-md-4 mb-4">
    <div class="card shadow-sm h-100">
      <div class="card-body">
        <h5 class="card-title">Column Two</h5>
        <p class="card-text">Bootstrap handles spacing and responsiveness beautifully.</p>
      </div>
    </div>
  </div>
  <div class="col-md-4 mb-4">
    <div class="card shadow-sm h-100">
      <div class="card-body">
        <h5 class="card-title">Column Three</h5>
        <p class="card-text">Cards stay side by side or stack depending on screen width.</p>
      </div>
    </div>
  </div>
</div>
```

</div>

<!-- Row 3 (4 columns) -->

<div class="row text-center">

<div class="col-sm-6 col-lg-3 mb-4">

<div class="p-3 bg-light border rounded">1/4 Width</div>

</div>

<div class="col-sm-6 col-lg-3 mb-4">

<div class="p-3 bg-light border rounded">1/4 Width</div>

</div>

<div class="col-sm-6 col-lg-3 mb-4">

<div class="p-3 bg-light border rounded">1/4 Width</div>

</div>

<div class="col-sm-6 col-lg-3 mb-4">

<div class="p-3 bg-light border rounded">1/4 Width</div>

</div>

</div>

</div>

<!-- Footer -->

<footer class="text-center py-4 bg-dark text-white">

© 2025 Bootstrap Grid Demo

</footer>

<!-- Bootstrap Script -->

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

OUTPUT:

Bootstrap Grid System

Responsive layout using rows and columns



Responsive Columns

This is a 2-column layout using Bootstrap's grid. On smaller screens, it stacks vertically.

Column One

This column spans 4/12 of the row on medium+ screens.

Column Two

Bootstrap handles spacing and responsiveness beautifully.

Column Three

Cards stay side by side or stack depending on screen width.

1/4 Width

1/4 Width

1/4 Width

1/4 Width

© 2025 Bootstrap Grid Demo

Bootstrap Grid System

Responsive layout using rows and columns



Responsive Columns

This is a 2-column layout using Bootstrap's grid. On smaller screens, it stacks vertically.

Column One

This column spans 4/12 of the row on medium+ screens.

Column Two

Bootstrap handles spacing and responsiveness beautifully.

Column Three

Cards stay side by side or stack depending on screen width.

1/4 Width

1/4 Width

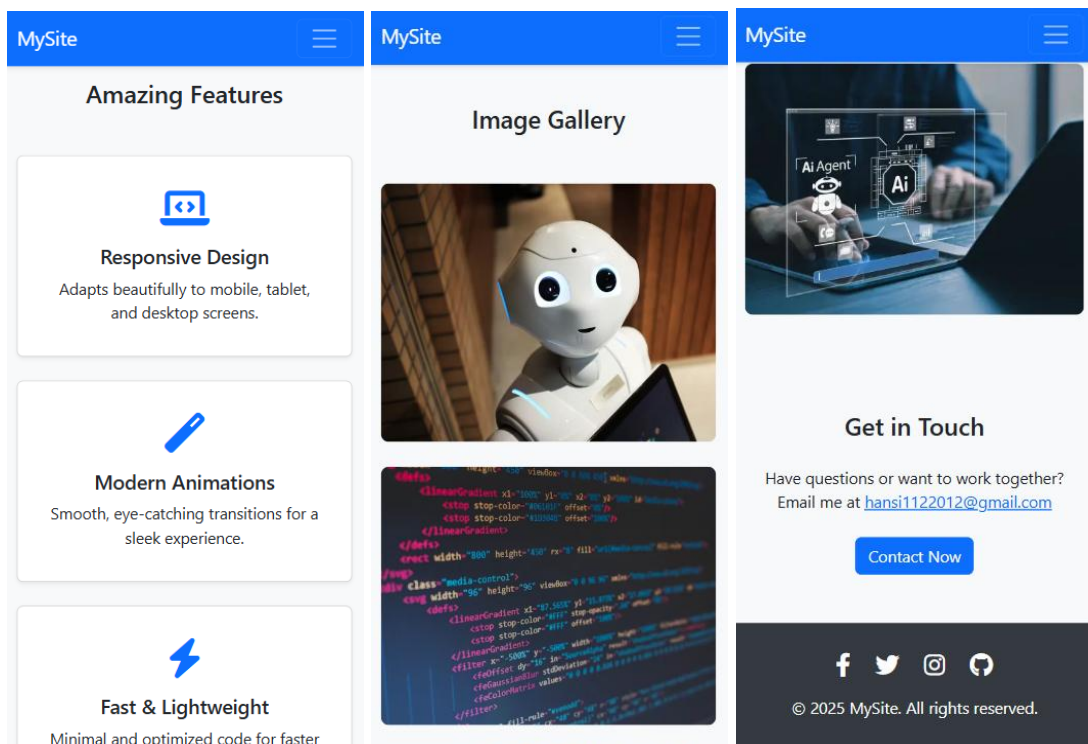
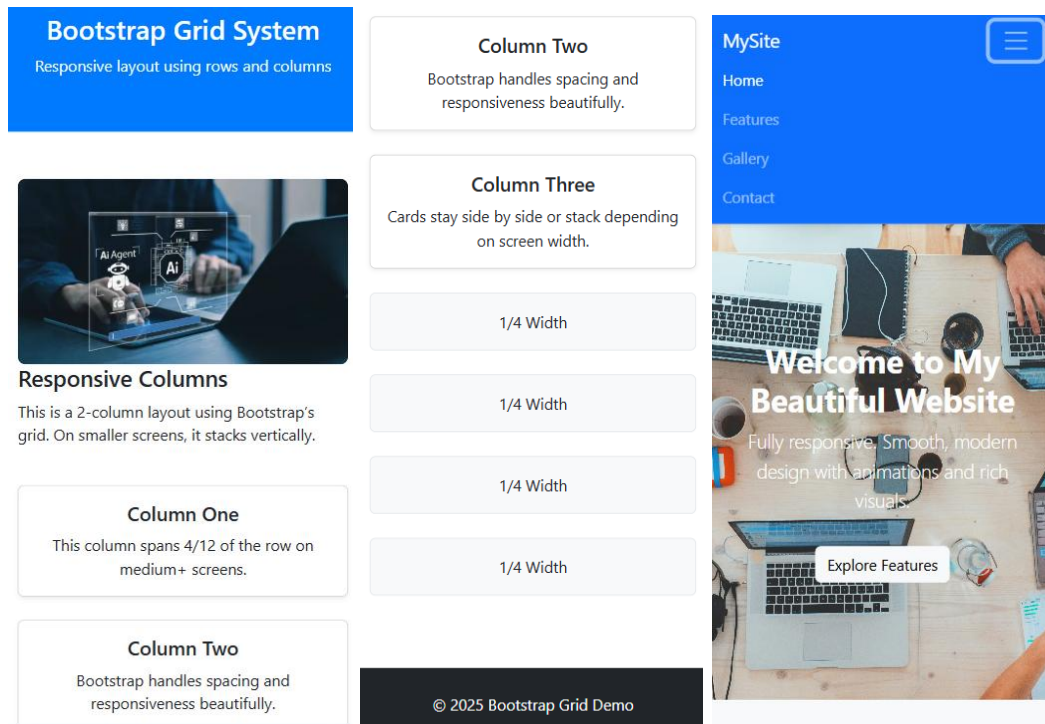
1/4 Width

1/4 Width

© 2025 Bootstrap Grid Demo

ROLL NO:2116221801003

PAGE NO:38



RESULT:

Thus, a responsive web page using bootstrap's grid system is implemented successfully.

EXP NO: 9	DESIGN A WEBPAGE WITH DROPDOWN, NAVIGATION BAR AND PAGINATION
DATE: 29.3.25	

AIM:

To design a webpage with Dropdown, Navigation bar and Pagination.

ALGORITHM:

- Step 1:** Create a responsive HTML structure using Bootstrap 4 layout.
- Step 2:** Add a dark-themed Bootstrap navbar with brand name and toggler.
- Step 3:** Insert navbar links including Home, About, and a dropdown for Services.
- Step 4:** Define dropdown items under Services using Bootstrap dropdown classes.
- Step 5:** Add a container with welcome heading and paragraph content.
- Step 6:** Insert Bootstrap pagination component with Previous, numbered pages, and Next.
- Step 7:** Include Bootstrap and jQuery CDN links for styling and interactivity.
- Step 8:** Add JavaScript to dynamically switch active pagination and update content.
- Step 9:** Test navbar toggle, dropdown, and pagination functionality on various screen sizes.
- Step 10:** Style and organize the layout using Bootstrap utility classes for clean design.

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Perfect Webpage</title>

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

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

  <style>

    body {

      background-color: #f4f6f8;

      font-family: 'Segoe UI', sans-serif;

    }

    .navbar {
```



```

        box-shadow: 0 2px 6px rgba(0,0,0,0.1);
    }
    .item-card {
        border: 1px solid #dee2e6;
        border-radius: 10px;
        padding: 20px;
        background: white;
        margin-bottom: 20px;
        transition: 0.3s;
    }
    .item-card:hover {
        box-shadow: 0 4px 12px rgba(0,0,0,0.1);
    }
    .pagination {
        justify-content: center;
    }
    footer {
        background: #343a40;
        color: white;
        padding: 20px 0;
        text-align: center;
        margin-top: 50px;
    }
</style>
</head>
<body>

<!-- Navigation Bar -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <a class="navbar-brand" href="#">MySite</a>

```

```

<button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav">
  <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item active"><a class="nav-link" href="#">Home</a></li>
    <li class="nav-item"><a class="nav-link" href="#">About</a></li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="servicesDropdown" role="button"
data-toggle="dropdown">
        Services
      </a>
      <div class="dropdown-menu">
        <a class="dropdown-item" href="#">Design</a>
        <a class="dropdown-item" href="#">Development</a>
        <a class="dropdown-item" href="#">SEO</a>
      </div>
    </li>
    <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>
  </ul>
</div>
</nav>

```

```

<div class="container mt-5">
  <h3 class="mb-4 text-center">Our Portfolio (Paginated Items)</h3>
  <div id="item-list" class="row">
  </div>
  <nav>
    <ul class="pagination" id="pagination">
    </ul>
  </nav>

```

```

</nav>

</div>

<footer>

  <div class="container">

    <p>© 2025 MySite. All rights reserved.</p>

  </div>

</footer>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.bundle.min.js"></script>

<script>

  const items = Array.from({ length: 24 }, (_, i) => `Item ${i + 1}`);

  const itemsPerPage = 6;

  let currentPage = 1

  function renderItems() {

    const start = (currentPage - 1) * itemsPerPage;

    const end = start + itemsPerPage;

    const currentItems = items.slice(start, end);

    const itemList = document.getElementById('item-list');

    itemList.innerHTML = "";

    currentItems.forEach(item => {

      const col = document.createElement('div');

      col.className = 'col-md-4';

      col.innerHTML = `

        <div class="item-card">

          <h5>${item}</h5>

          <p>This is a short description for ${item}. Explore more about our awesome work.</p>

        </div>`;

      itemList.appendChild(col);

    });

  }

```

```

function renderPagination() {
    const totalPages = Math.ceil(items.length / itemsPerPage);
    const pagination = document.getElementById('pagination');
    pagination.innerHTML = "";

    // Previous Button
    pagination.innerHTML += `
        <li class="page-item ${currentPage === 1 ? 'disabled' : ""}>
            <a class="page-link" href="#" onclick="changePage(${currentPage - 1})">Previous</a>
        </li>`;

    // Page Numbers
    for (let i = 1; i <= totalPages; i++) {
        pagination.innerHTML += `
            <li class="page-item ${i === currentPage ? 'active' : ""}>
                <a class="page-link" href="#" onclick="changePage(${i})">${i}</a>
            </li>`;
    }

    // Next Button
    pagination.innerHTML += `
        <li class="page-item ${currentPage === totalPages ? 'disabled' : ""}>
            <a class="page-link" href="#" onclick="changePage(${currentPage + 1})">Next</a>
        </li>`;
    }

    function changePage(page) {
        const totalPages = Math.ceil(items.length / itemsPerPage);
        if (page >= 1 && page <= totalPages) {
            currentPage = page;
        }
    }
}

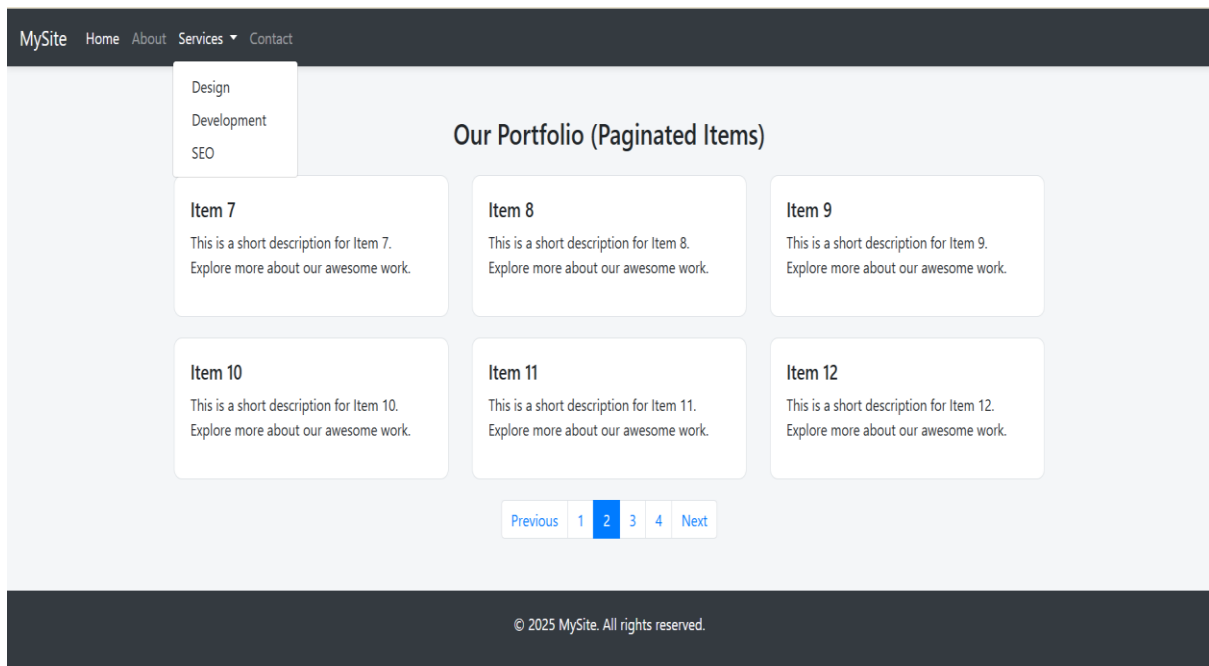
```

```
renderItems();  
renderPagination();  
}  
}
```

```
// Initial Load
```

```
renderItems();  
renderPagination();  
</script>  
</body>  
</html>
```

OUTPUT:



RESULT:

Thus, a webpage with Dropdown, Navigation bar and Pagination is designed successfully and verified.

EXP NO: 10	DESIGN WEBPAGE USING JQUERY SELECTOR
DATE: 5.4.25	

AIM:

To design a web page using jQuery selector.

ALGORITHM:

Step 1: Create a structured HTML layout with headings, paragraphs, and div elements having various classes and attributes.

Step 2: Include jQuery library and Google Fonts via CDN in the <head>.

Step 3: Define CSS styles for layout, typography, buttons, and highlight effects.

Step 4: Add multiple buttons with unique IDs to trigger different selector actions.

Step 5: Wrap all content inside a styled container for better presentation.

Step 6: Use \$(document).ready() to ensure jQuery runs after the DOM loads.

Step 7: Use jQuery element selector to toggle highlights on all paragraphs.

Step 8: Use class, attribute, child, and pseudo selectors (e.g., .note, [data-custom], :nth-child, :not) to target specific elements.

Step 9: Assign click event handlers to each button to apply corresponding visual effects.

SOURCE CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Advanced jQuery Selectors Demo</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap"
rel="stylesheet">
  <style>
    body {
      font-family: 'Roboto', sans-serif;
```

```
background-color: #f8f9fa;
padding: 40px;
}
.container {
max-width: 800px;
margin: auto;
background-color: #ffffff;
padding: 30px;
border-radius: 10px;
box-shadow: 0 0 15px rgba(0,0,0,0.1);
}
h2, h3 {
color: #343a40;
margin-bottom: 20px;
}
p, div {
font-size: 17px;
margin-bottom: 10px;
}
.note {
color: #6c757d;
font-style: italic;
}
.highlight {
background-color: yellow;
font-weight: bold;
}
.custom {
color: darkgreen;
font-weight: bold;
```



```
}  
.special {  
    color: red;  
    font-weight: bold;  
}  
button {  
    padding: 10px 15px;  
    margin: 10px 5px;  
    background-color: #007bff;  
    color: white;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
}  
button:hover {  
    background-color: #0056b3;  
}  
</style>  
</head>  
<body>  
  
<div class="container">  
    <h2>Advanced jQuery Selector Demonstration</h2>  
  
    <p>This is the first paragraph.</p>  
    <p class="note">This is a note paragraph.</p>  
    <div>This is a general div.</div>  
    <div data-custom="true">This div has a custom data attribute.</div>  
    <h3 class="note">This is a heading with class "note".</h3>  
    <div class="note">This is another note div.</div>
```

```
<button id="highlightParagraphs">Highlight Paragraphs</button>
<button id="highlightNotes">Highlight Notes</button>
<button id="highlightCustom">Highlight Data Attribute</button>
<button id="highlightNth">Highlight Every 2nd Paragraph</button>
<button id="highlightNotNote">Highlight Non-Note Paragraphs</button>
<button id="highlightChild">Highlight First Child Div</button>
</div>
```

```
<script>
$(document).ready(function(){
    // Element selector
    $("#highlightParagraphs").click(function(){
        $("p").toggleClass("highlight");
    });
    // Class selector
    $("#highlightNotes").click(function(){
        $(".note").toggleClass("highlight");
    });
    // Attribute selector
    $("#highlightCustom").click(function(){
        $('[data-custom]').toggleClass("custom");
    });
    // nth-child selector
    $("#highlightNth").click(function(){
        $("p:nth-child(2)").toggleClass("special");
    });
    // not selector
    $("#highlightNotNote").click(function(){
        $("p:not(.note)").toggleClass("highlight");
    });
});
</script>
```

```

    });
    // child selector
    $("#highlightChild").click(function(){
        $("div:first-child").toggleClass("special");
    });
});
</script>
</body>
</html>

```

OUTPUT:

Advanced jQuery Selector Demonstration

This is the first paragraph.

This is a note paragraph.

This is a general div.

This div has a custom data attribute.

This is a heading with class "note".

This is another note div.

Highlight Paragraphs

Highlight Notes

Highlight Data Attribute

Highlight Every 2nd Paragraph

Highlight Non-Note Paragraphs

Highlight First Child Div

RESULT:

Thus, a web page using jQuery selector is designed successfully and verified.

EXP NO: 11	CREATE A SIMPLE WEB PAGE USING JQUERY EFFECTS
DATE: 7.4.25	

AIM:

To create a simple web page using jQuery Effects.

ALGORITHM:

Step 1: Start with a basic HTML structure including <head> and <body>.

Step 2: Link jQuery and Google Fonts in the <head> section.

Step 3: Style the layout using CSS for body, container, buttons, and effect box.

Step 4: Create a centered container with a heading and multiple buttons for effects.

Step 5: Add a <div> element (#effectBox) to show visual changes from jQuery.

Step 6: Use \$(document).ready() to initialize jQuery when the page is loaded.

Step 7: Attach click() events to each button to trigger a specific jQuery effect like fadeToggle, slideUp, hide, show, etc.

Step 8: Use animate() to apply combined width, height, and opacity transitions.

Step 9: Use toggleClass() to switch styles dynamically for highlighting.

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Advanced jQuery Effects</title>

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

  <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap"
rel="stylesheet">

  <style>

    body { font-family: 'Roboto', sans-serif;

      background: #f4f7fa;

      padding: 40px;

    }

    .container {

      max-width: 750px;
```

```
margin: auto;
background: #fff;
padding: 30px;
border-radius: 10px;
box-shadow: 0 4px 15px rgba(0,0,0,0.1);
text-align: center;
}
h2 { color: #333;
margin-bottom: 25px;
}
button {
margin: 10px;
padding: 12px 20px;
border: none;
background-color: #007bff;
color: #fff;
border-radius: 6px;
cursor: pointer;
font-size: 15px;
transition: 0.3s;
}
button:hover {
background-color: #0056b3;
}
#effectBox {
width: 100%;
max-width: 500px;
height: 150px;
background: #d1ecf1;
margin: 20px auto;
```

```

padding: 20px;
font-size: 18px;
line-height: 1.5;
border-radius: 8px;
box-shadow: 0 4px 10px rgba(0,0,0,0.1);
transition: all 0.4s ease-in-out;
}
.highlighted {
background-color: #ffc107 !important;
color: #000;
transform: scale(1.05);
}
</style>
</head>
<body>
<div class="container">
  <h2>Advanced jQuery Effects</h2>
  <button id="fadeToggle">Fade Toggle</button>
  <button id="fadeIn">Fade In</button>
  <button id="fadeOut">Fade Out</button>
  <button id="slideToggle">Slide Toggle</button>
  <button id="slideUp">Slide Up</button>
  <button id="slideDown">Slide Down</button>
  <button id="hide">Hide</button>
  <button id="show">Show</button>
  <button id="toggle">Toggle</button>
  <button id="animateBtn">Animate</button>
  <button id="highlightBtn">Highlight Toggle</button>
  <div id="effectBox">

```

This is a dynamic content box. Click any button above to try different effects!

</div>

</div>

<script>

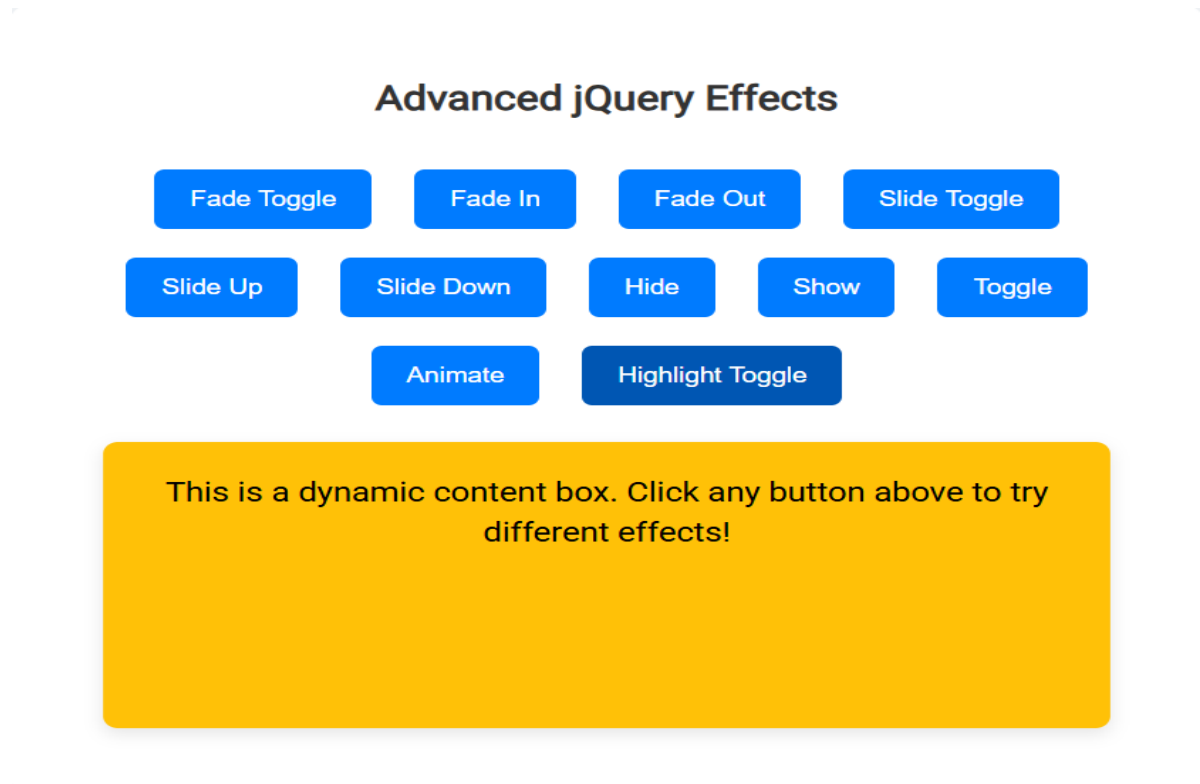
```
$(document).ready(function(){
    $("#fadeToggle").click(function(){
        $("#effectBox").fadeToggle("slow");
    });
    $("#fadeIn").click(function(){
        $("#effectBox").fadeIn("slow");
    });
    $("#fadeOut").click(function(){
        $("#effectBox").fadeOut("slow");
    });
    $("#slideToggle").click(function(){
        $("#effectBox").slideToggle("slow");
    });
    $("#slideUp").click(function(){
        $("#effectBox").slideUp("slow");
    });
    $("#slideDown").click(function(){
        $("#effectBox").slideDown("slow");
    });
    $("#hide").click(function(){
        $("#effectBox").hide("slow");
    });
    $("#show").click(function(){
        $("#effectBox").show("slow");
    });
    $("#toggle").click(function(){
        $("#effectBox").toggle("slow");
    });
});
```

```

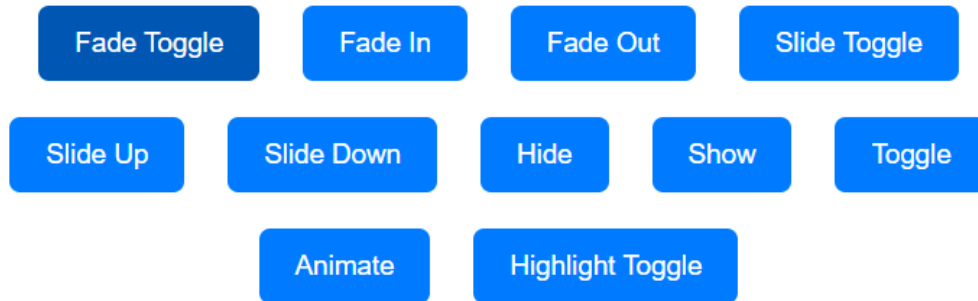
});
$("#animateBtn").click(function(){
    $("#effectBox").animate({
        width: "toggle",
        height: "toggle",
        opacity: "toggle"
    }, 1000);
});
$("#highlightBtn").click(function(){
    $("#effectBox").toggleClass("highlighted");
});
});
</script>
</body>
</html>

```

OUTPUT:



Advanced jQuery Effects



RESULT:

Thus, a simple web page was successfully designed using jQuery effects and verified.

| | |
|----------------------|---|
| EXP NO: 12 | DESIGN A WEB PAGE TO CALCULATE FACTORIAL OF A NUMBER USING PHP |
| DATE: 12.4.25 | |

AIM:

To design a web page to calculate factorial of a number using PHP.

ALGORITHM:

Step 1: Create an HTML form to accept a number as input from the user.

Step 2: Set the form's method to POST and action to the same PHP file.

Step 3: Check if the form is submitted using `$_SERVER["REQUEST_METHOD"] == "POST"`.

Step 4: Retrieve the input number using `$_POST["num"]`.

Step 5: Initialize a variable factorial to 1.

Step 6: If the input number is negative, display an error message.

Step 7: Otherwise, use a for loop to multiply numbers from 1 to the input number.

Step 8: After the loop ends, display the calculated factorial result.

Step 9: Embed the PHP code below the HTML form to process and display the result on the same page.

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Factorial Calculator</title>

  <style>

    body {

      font-family: 'Segoe UI', sans-serif;

      background-color: #f0f8ff;

      display: flex;

      justify-content: center;

      align-items: center;

      height: 100vh;

    }

    .container {
```

```
background-color: white;
padding: 30px 40px;
border-radius: 10px;
box-shadow: 0 0 15px rgba(0,0,0,0.2);
text-align: center;
width: 350px;
}
h2 {
color: #007bff;
margin-bottom: 20px;
}
input[type="number"] {
width: 80%;
padding: 10px;
margin-bottom: 15px;
border-radius: 5px;
border: 1px solid #ccc;
font-size: 16px;
}
input[type="submit"] {
background-color: #007bff;
color: white;
border: none;
padding: 10px 20px;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
}
input[type="submit"]:hover {
background-color: #0056b3;
```

```

    }
    .result {
        margin-top: 20px;
        font-size: 18px;
        font-weight: bold;
        color: green;
    }
    .error {
        color: red;
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Factorial Calculator</h2>
        <form method="post" action="">
            <input type="number" name="num" placeholder="Enter a number" required>
            <br>
            <input type="submit" value="Calculate">
        </form>

        <?php
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $num = $_POST["num"];
            if (!is_numeric($num)) {
                echo "<div class='error'>Please enter a valid number.</div>";
            } elseif ($num < 0) {
                echo "<div class='error'>Factorial is not defined for negative numbers.</div>";
            } else {
                $factorial = 1;

```

```
for ($i = 1; $i <= $num; $i++) {  
    $factorial *= $i;  
}  
echo "<div class='result'>Factorial of $num is $factorial</div>";  
}  
}  
?>  
</div>  
</body>  
</html>
```

OUTPUT:

Factorial Calculator

Calculate

Factorial of 6 is 720

Factorial Calculator



Calculate

Factorial of 3 is 6

RESULT:

Thus, designing a web page to calculate factorial of a number using PHP has been executed successfully.

EXP NO: 13	CREATE A WEB PAGE TO PERFORM ARITHMETIC OPERATIONS USING PHP
DATE: 19.4.25	

AIM:

To create a webpage to perform arithmetic operations using PHP.

ALGORITHM:

Step 1: Start

Step 2: Display a form to input two numbers and select an operation (Add, Subtract, Multiply, Divide).

Step 3: Wait for user to submit the form using the submit button.

Step 4: Retrieve input values num1, num2, and operation from the form

Step 5: Validate inputs to ensure both numbers are numeric

Step 6: Use switch-case to perform the selected arithmetic operation

Step 7: Handle division by zero if the operation is division

Step 8: Display the result on the same page

Step 9: End

SOURCE CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Arithmetic Calculator</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f6f9;

            padding: 50px;

        }

        .calculator {

            background: #fff;

            padding: 30px;

            max-width: 400px;
```

```
margin: auto;
border-radius: 10px;
box-shadow: 0 0 15px rgba(0,0,0,0.2);
}
h2 {
text-align: center;
margin-bottom: 20px;
color: #333;
}
input[type="number"], select {
width: 100%;
padding: 10px;
margin: 10px 0;
border: 1px solid #ccc;
border-radius: 5px;
}
input[type="submit"] {
background-color: #28a745;
color: white;
padding: 12px;
border: none;
width: 100%;
border-radius: 5px;
cursor: pointer;
}
input[type="submit"]:hover {
background-color: #218838;
}
.result {
margin-top: 20px;
```



```

        text-align: center;
        font-weight: bold;
        color: #007bff;
    }
    .error {
        color: red;
        font-weight: bold;
        text-align: center;
    }
</style>
</head>
<body>

<div class="calculator">
    <h2>Arithmetic Calculator</h2>
    <form method="post" action="">
        Number 1:
        <input type="number" name="num1" step="any" required>

        Number 2:
        <input type="number" name="num2" step="any" required>

        Operation:
        <select name="operation" required>
            <option value="">--Select Operation--</option>
            <option value="add">Addition (+)</option>
            <option value="sub">Subtraction (-)</option>
            <option value="mul">Multiplication (×)</option>
            <option value="div">Division (÷)</option>
        </select>

```

```
<input type="submit" value="Calculate">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $num1 = $_POST["num1"];
    $num2 = $_POST["num2"];
    $op = $_POST["operation"];

    if (!is_numeric($num1) || !is_numeric($num2)) {
        echo "<div class='error'>Please enter valid numbers.</div>";
    } else {
        switch($op) {
            case "add": $res = $num1 + $num2; break;
            case "sub": $res = $num1 - $num2; break;
            case "mul": $res = $num1 * $num2; break;
            case "div":
                if ($num2 != 0) {
                    $res = $num1 / $num2;
                } else {
                    $res = "Cannot divide by zero";
                }
                break;
            default: $res = "Invalid operation";
        }
        echo "<div class='result'>Result: $res</div>";
    }
}
?>
```

</div>

</body>

</html>

OUTPUT:

Arithmetic Calculator

Number 1:

13

Number 2:

45

Operation:

Subtraction (-)



Calculate

Result: -32

Arithmetic Calculator

Number 1:

Number 2:

Operation:

Calculate

Result: 58

Arithmetic Calculator

Number 1:

Number 2:

Operation:

Calculate

Result: 585

Arithmetic Calculator

Number 1:

Number 2:

Operation:

Result: 0.28888888888889

RESULT:

Thus, a webpage to perform arithmetic operations using PHP has been executed successfully.

EXP NO: 14	PHP PROGRAM USING REGULAR EXPRESSIONS
DATE: 26.4.25	

AIM:

To design and implement a PHP-based web form that validates user inputs using regular expressions and logic checks.

ALGORITHM:

Step 1: Start the HTML form and accept input fields for username, DOB, mobile, Aadhar, password, PIN code, and PAN number.

Step 2: On form submission, collect form data using the \$_POST method in PHP.

Step 3: Initialize an empty array to store validation error messages.

Step 4: Validate the username using a regular expression to allow 4–15 characters (letters, digits, underscores).

Step 5: Calculate the user's age from DOB and check if it is 18 or above.

Step 6: Validate the mobile number to ensure it starts with 6–9 and is 10 digits long.

Step 7: Validate the Aadhar number to ensure it has exactly 12 digits.

Step 8: If no validation errors, display a success message; otherwise, display all error messages.

SOURCE CODE:

```
<!DOCTYPE html>

<html>

<head>

    <title>Enhanced Form Validation</title>

    <style>

        body {

            font-family: Arial;

            padding: 20px;

        }

        .error {

            color: red;

        }

    </style>

</head>

<body>
```

```

        .success {
            color: green;
        }
    </style>
</head>
<body>

<h2>User Registration Form</h2>

<form method="post">
    Username: <input type="text" name="username"><br><br>
    Date of Birth: <input type="date" name="dob"><br><br>
    Mobile Number: <input type="text" name="mobile"><br><br>
    Aadhar Number: <input type="text" name="aadhar"><br><br>
    Password: <input type="password" name="password"><br><br>
    PIN Code: <input type="text" name="pincode"><br><br>
    PAN Number: <input type="text" name="pan"><br><br>

    <input type="submit" name="submit" value="Submit">
</form>

<?php
if (isset($_POST['submit'])) {
    $username = $_POST['username'];
    $dob = $_POST['dob'];
    $mobile = $_POST['mobile'];
    $aadhar = $_POST['aadhar'];
    $password = $_POST['password'];
    $pincode = $_POST['pincode'];
    $pan = $_POST['pan'];

```

```
$errors = [];
```

```
// Username (4-15 characters, letters, digits, _)
```

```
if (!preg_match('/^[a-zA-Z0-9_]{4,15}$/', $username)) {  
    $errors[] = "Invalid Username";  
}
```

```
// DOB (age must be 18+)
```

```
$today = new DateTime();  
$birthDate = new DateTime($dob);  
$age = $today->diff($birthDate)->y;  
if ($age < 18) {  
    $errors[] = "You must be at least 18 years old.";  
}
```

```
// Mobile
```

```
if (!preg_match('/^[6-9]\d{9}$/', $mobile)) {  
    $errors[] = "Invalid Mobile Number";  
}
```

```
// Aadhar
```

```
if (!preg_match('/^\d{12}$/', $aadhar)) {  
    $errors[] = "Invalid Aadhar Number";  
}
```

```
// Password (min 6 chars, at least 1 letter and 1 number)
```

```
if (!preg_match('/^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{6,}$/', $password)) {  
    $errors[] = "Password must be at least 6 characters with letters and numbers";  
}
```



```

// PIN Code
if (!preg_match('/^[1-9][0-9]{5}$/', $pincode)) {
    $errors[] = "Invalid PIN Code";
}

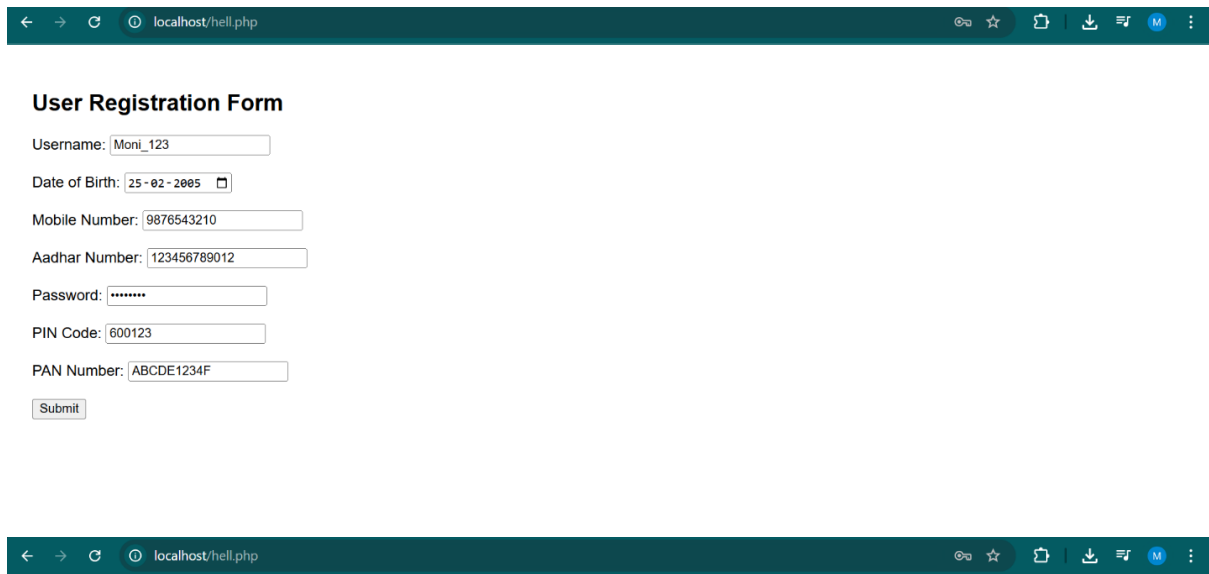
// PAN Number
if (!preg_match('/^[A-Z]{5}[0-9]{4}[A-Z]{1}$/', $pan)) {
    $errors[] = "Invalid PAN Number";
}

// Output
if (empty($errors)) {
    echo "<p class='success'>All inputs are valid!</p>";
} else {
    foreach ($errors as $error) {
        echo "<p class='error'>$error</p>";
    }
}
?>

</body>
</html>

```

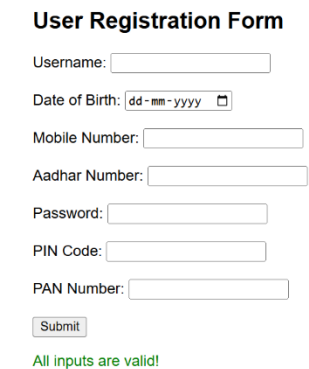
OUTPUT:



A screenshot of a web browser displaying a 'User Registration Form'. The browser's address bar shows 'localhost/hell.php'. The form contains the following fields and values:

- Username: Moni_123
- Date of Birth: 25-02-2005
- Mobile Number: 9876543210
- Aadhar Number: 123456789012
- Password: *****
- PIN Code: 600123
- PAN Number: ABCDE1234F

A 'Submit' button is located at the bottom of the form.



A screenshot of a web browser displaying the same 'User Registration Form'. The browser's address bar shows 'localhost/hell.php'. The form fields are now empty:

- Username:
- Date of Birth: dd-mm-yyyy
- Mobile Number:
- Aadhar Number:
- Password:
- PIN Code:
- PAN Number:

A 'Submit' button is at the bottom. Below the form, a green message reads: 'All inputs are valid!'.

RESULT:

The PHP script successfully validates all user inputs and provides appropriate error or success messages based on the entered data.

EXP NO: 15	DEVELOP ANY APPLICATION USING PHP WITH MYSQL
DATE: 26.4.25	

AIM:

To develop organic grocery store application using PHP with MYSQL

ALGORITHM:

Step 1: Start

1. Initialize the application.
2. Load the **Home Page** interface.
3. Display navigation options:
 - Login
 - Register
 - Browse Books
 - Contact/Help (optional)

Step 2: User Registration/Login

1. **If the user is new:**
 - Prompt for registration details (name, email, password, role).
 - Validate input (e.g., email format, password strength).
 - Store user information in the database.
 - Redirect to respective **Dashboard** based on role.
2. **If the user already exists:**
 - Prompt for email and password.
 - Authenticate credentials from the database.
 - On success, redirect to:
 - **Admin Dashboard** if role = Admin
 - **Reader/Reviewer Dashboard** otherwise

Step 3: Admin Module Execution

1. Display options:
 - Manage Books (Add / Edit / Delete)
 - Manage Authors (Add / Edit / Delete)
 - Moderate Reviews
 - Generate Reports
2. Allow admin to:
 - Add new books and associate authors.
 - Update or remove book entries.
 - Review submitted ratings/comments and flag or delete inappropriate ones.
 - Generate activity logs, user reports, and top-rated books list.

Step 4: Reader/Reviewer Module Execution

1. Display browsing options:
 - Search bar and filters (title, genre, author).
2. On book selection:
 - Show book details (title, author, description, genre).
 - List all existing reviews and ratings.
3. If user is a reviewer:
 - Allow them to submit a new review or edit their existing one.

Step 5: Review Management

1. Accept input: star rating (1–5), textual review.
2. Validate:
 - Non-empty review
 - Rating within valid range
3. Save review to the database.
4. Recalculate and update the **average book rating** based on all approved reviews.

Step 6: Feedback and Moderation

1. Admin receives flagged or newly submitted reviews.
2. Check for:

- Inappropriate content
 - Spam or offensive language
3. Allow admin to:
 - Approve, edit, or delete reviews
 - Warn or block abusive users (optional)

Step 7: End

1. On logout:
 - Destroy user session
 - Clear sensitive data from memory
2. Close any open database connections.
3. Display logout confirmation and redirect to the home page.

SOURCE CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Organic Grocery Store</title>
  <style>
    * {
      box-sizing: border-box;
    }

    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f3f4f6;
      color: #333;
```

```
}
```

```
.container {  
    max-width: 1200px;  
    margin: 0 auto;  
    padding: 20px;  
    display: flex;  
    justify-content: space-between;  
}
```

```
header {  
    font-style: italic;  
    background: linear-gradient(to right, #281063, #25b105);  
    color: #ffffff;  
    padding: 25px 0;  
    text-align: center;  
    margin-bottom: 20px;  
    width: 100%;  
    border-radius: 10px;  
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
}
```

```
header h1 {  
    font-size: 2.5rem;  
    margin: 0;  
}
```

```
header nav ul {  
    list-style-type: none;  
    padding: 15px;
```

```
margin: 10px;
display: flex;
justify-content: center;
gap: 20px;
}
```

```
header nav ul li {
  display: grid;
}
```

```
header nav ul li a {
  color: #ffffff;
  text-decoration: dotted;
  font-size: 1.2rem;
  transition: color 0.3s ease;
}
```

```
header nav ul li a:hover {
  color: #f3f4f6;
}
```

```
#products {
  margin-bottom: 20px;
  flex: 1;
}
```

```
.product {
  background-color: #ffffff;
  border-radius: 10px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}
```

```
padding: 20px;
margin-bottom: 20px;
display: flex;
flex-direction: column;
align-items: center;
text-align: center;
transition: transform 0.3s ease;
}
```

```
.product:hover {
  transform: translateY(-5px);
}
```

```
.product img {
  width: 100%;
  max-width: 200px;
  height: auto;
  margin-bottom: 1rem;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}
```

```
.product h3 {
  font-size: 1.5rem;
  font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
  margin: 0;
}
```

```
.product p {
  color: #666666;
```



```
margin-bottom: 0.5rem;
}

.product button {
  background: linear-gradient(to right, #11998e, #38ef7d);
  color: #ffffff;
  border: none;
  padding: 8px 16px;
  border-radius: 4px;
  cursor: pointer;
  transition: background-color 0.3s ease;
  outline: none;
}

.product button:hover {
  background: linear-gradient(to right, #0c8976, #30c270);
}

#cart {
  background-color: #ffffff;
  border-radius: 10px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  padding: 20px;
  width: 300px;
}

#cart h2 {
  font-size: 2rem;
  margin-bottom: 1rem;
  color: #333;
```

```
}
```

```
#cart-items {  
  list-style-type: none;  
  padding: 0;  
  margin: 0;  
}
```

```
.cart-item {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  margin-bottom: 0.5rem;  
}
```

```
.cart-item button {  
  background: linear-gradient(to right, #ff4d4f, #ff6382);  
  color: #ffffff;  
  border: none;  
  padding: 4px 8px;  
  border-radius: 4px;  
  cursor: pointer;  
  transition: background-color 0.3s ease;  
  outline: none;  
}
```

```
.cart-item button:hover {  
  background: linear-gradient(to right, #e63c3f, #f34d6f);  
}
```

```
.quantity {  
    display: flex;  
    align-items: center;  
}
```

```
.quantity button {  
    font-style: italic;  
    background: linear-gradient(to right, #11998e, #38ef7d);  
    color: #ffffff;  
    border: none;  
    padding: 4px 8px;  
    border-radius: 4px;  
    cursor: pointer;  
    transition: background-color 0.3s ease;  
    outline: none;  
}
```

```
.quantity button:hover {  
    background: linear-gradient(to right, #0c8976, #30c270);  
}
```

```
.quantity input {  
    width: 40px;  
    text-align: center;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
    margin: 0 0.5rem;  
    padding: 4px;  
    outline: none;
```

```
}
```

```
.modal {  
  display: none;  
  position: fixed;  
  z-index: 1;  
  left: 0;  
  top: 0;  
  width: 100%;  
  height: 100%;  
  overflow: auto;  
  background-color: rgb(0, 0, 0);  
  background-color: rgba(0, 0, 0, 0.4);  
  padding-top: 60px;  
}
```

```
.modal-content {  
  background-color: #fefefe;  
  margin: 5% auto;  
  padding: 20px;  
  border: 1px solid #888;  
  width: 80%;  
  max-width: 500px;  
  border-radius: 10px;  
}
```

```
.close {  
  color: #aaa;  
  float: right;  
  font-size: 28px;
```

```
    font-weight: bold;
}
```

```
.close:hover,
.close:focus {
    color: black;
    text-decoration: none;
    cursor: pointer;
}
```

```
.modal button {
    background: linear-gradient(to right, #11998e, #38ef7d);
    color: #ffffff;
    border: none;
    padding: 8px 16px;
    border-radius: 4px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    outline: none;
}
```

```
.modal button:hover {
    background: linear-gradient(to right, #0c8976, #30c270);
}
```

```
.modal input[type="text"],
.modal input[type="tel"] {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
```

```
border: 1px solid #ccc;
border-radius: 4px;
outline: none;
}

.modal select {
width: 100%;
padding: 10px;
margin: 10px 0;
border: 1px solid #ccc;
border-radius: 4px;
outline: none;
}
</style>
</head>

<body>
<header>
<div class="container">
<h1>Organic Grocery Store</h1>
<nav>
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Products</a></li>
<li><a href="#">Cart</a></li>
<li><a href="#">Contact</a></li>
</ul>
</nav>
</div>
</header>
```

```
<div class="container">
  <section id="products">
    <h2>Available Products</h2>
    <div class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
      <div class="product">
        <img src=
"https://wallpaperaccess.com/full/1463517.jpg"
        alt="rice">
        <h3>Rice 1Kg</h3>
        <p>65.00</p>
        <button onclick="addToCart('Rice', 80.00)">
          Add to Cart
        </button>
      </div>
      <div class="product">
        <img src=
"https://www.pillsbury.in/wp-content/uploads/2021/09/Organic-Atta_LS.jpg"
        alt="Wheat">
        <h3>Wheat 1Kg</h3>
        <p>35.00</p>
        <button onclick="addToCart('Wheat', 80.00)">
          Add to Cart
        </button>
      </div>
      <div class="product">
        <img
          src=
"https://media.geeksforgeeks.org/wp-content/uploads/20240308174326/d12.jpg"
        alt="Salt">
        <h3>Salt</h3>
```

```

    <p>20.00</p>
    <button onclick="addToCart('Salt', 20.00)">
        Add to Cart
    </button>
</div>
<div class="product">
    <img
    src=
    "https://media.geeksforgeeks.org/wp-content/uploads/20240308174500/f12.jpg"
    alt="Apple">
    <h3>Apple</h3>
    <p>50.00</p>
    <button onclick="addToCart('Apple', 50.00)">
        Add to Cart
    </button>
</div>
<div class="product">
    <img src=
    "https://media.geeksforgeeks.org/wp-content/uploads/20240308174616/o12.jpg"
    alt="Orange">
    <h3>Orange</h3>
    <p>40.00</p>
    <button onclick="addToCart('Orange', 40.00)">
        Add to Cart
    </button>
</div>
<div class="product">
    <img src=
    "https://th.bing.com/th/id/R.09992bd09dcb791b0c69f1276591a970?rik=4mOqWrGhQXIJgQ
    &riu=http%3a%2f%2fwww.theayurveda.org%2fwp-
    content%2fuploads%2f2015%2f09%2fCarrot-

```


fruits.jpg&ehk=VxTGCN5rKfbNmckesr1TgZwTJvbHrtJb%2byGuy1QI7gc%3d&risl=&pid=ImgRaw&r=0"

alt="Carrot">

<h3>Carrot 1Kg</h3>

<p>50.00</p>

<button onclick="addToCart('Carrot', 50.00)">

Add to Cart

</button>

</div>

<div class="product">

<img src=

"https://th.bing.com/th/id/OIP.EupJjY9C7hB7tHQodU1zhgHaE8?rs=1&pid=ImgDetMain"

alt="Beetroot">

<h3>Beetroot 1Kg</h3>

<p>55.00</p>

<button onclick="addToCart('Oil', 55.00)">

Add to Cart

</button>

</div>

<div class="product">

<img src=

"https://th.bing.com/th/id/OIP.HqNNeb007vdUsocitEE0jAHaE8?rs=1&pid=ImgDetMain"

alt="Onion">

<h3>Onion 1Kg</h3>

<p>45.00</p>

<button onclick="addToCart('Onion', 45.00)">

Add to Cart

</button>

</div>

<div class="product">

<img src=

"https://syncwithnature.in/wp-content/uploads/2020/10/toor-dal.png"

alt="ToorDal">

<h3>ToorDal 1Kg</h3>

<p>75.00</p>

<button onclick="addToCart('Oil', 75.00)">

Add to Cart

</button>

</div>

<div class="product">

<img src=

"https://th.bing.com/th/id/OIP.ijp1vFRHDSwHyzZ3hXmuJwHaE6?rs=1&pid=ImgDetMain"

alt="MasourDal">

<h3>MasourDal 1Kg</h3>

<p>35.00</p>

<button onclick="addToCart('MasourDal', 35.00)">

Add to Cart

</button>

</div>

<div class="product">

<img src=

"https://th.bing.com/th/id/OIP.WfvDRGOey8mJobjZzqLMsAHaHa?rs=1&pid=ImgDetMain"

alt="ChanaDal">

<h3>ChanaDal 1Kg</h3>

<p>55.00</p>

<button onclick="addToCart('ChanaDal',55.00)">

Add to Cart

</button>

</div>

<div class="product">

<img src=

"https://th-
i.thgim.com/public/migration_catalog/article14194713.ece/alternates/LANDSCAPE_1200/A
avin"

alt="Milk">

<h3>Aavin Milk 1Lt</h3>

<p>32.00</p>

<button onclick="addToCart('Milk', 32.00)">

Add to Cart

</button>

</div>

<div class="product">

<img src=

"https://th.bing.com/th/id/OIP.sh_zkPURbmE2RPPjsxczZgHaLH?w=900&h=1350&rs=1&pid=ImgDetMain"

alt="Cookies">

<h3>Cookies</h3>

<p>20.00</p>

<button onclick="addToCart('Cookies', 10.00)">

Add to Cart

</button>

</div>

<div class="product">

<img src=

"https://th.bing.com/th/id/OIP.W65I9bxaqneAw0AhQSMNnAHaFh?rs=1&pid=ImgDetMain"

alt="Chips">

<h3>Chips 50gr</h3>

<p>25.00</p>

<button onclick="addToCart('Chips', 25.00)">

Add to Cart

</button>

</div>

```
        </div>
    </section>
    <aside id="cart">
        <h2>Shopping Cart</h2>
        <ul id="cart-items"></ul>
        <p id="total-price">Total: 0.00</p>
        <button onclick="checkout()">Buy</button>
    </aside>
</div>
```

```
<!-- The Modal -->
<div id="checkoutModal" class="modal">
    <div class="modal-content">
        <span class="close">&times;</span>
        <h2>Checkout Details</h2>
        <form id="checkoutForm">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>
            <label for="address">Address:</label>
            <input type="text" id="address" name="address" required>
            <label for="phone">Phone Number:</label>
            <input type="tel" id="phone" name="phone" required>
            <label for="delivery">Delivery Option:</label>
            <select id="delivery" name="delivery">
                <option value="home">Home Delivery</option>
                <option value="pickup">Pickup</option>
            </select>
            <button type="button" onclick="submitOrder()">Submit</button>
        </form>
    </div>
</div>
```

</div>

<script>

```
let cart = [];
```

```
const products = {
```

```
  'Rice': 65.00,
```

```
  'Wheat': 35.00,
```

```
  'Salt': 20.00,
```

```
  'Apple': 50.00,
```

```
  'Orange': 40.00,
```

```
  'Carrot': 50.00,
```

```
  'Beetroot': 55.00,
```

```
  'Onion': 45.00,
```

```
  'ToorDal': 75.00,
```

```
  'MasourDal': 55.00,
```

```
  'ChanaDal': 55.00,
```

```
  'Aavin Milk': 32.00,
```

```
  'Cookies': 20.00,
```

```
  'Chips': 25.00,
```

```
};
```

```
function addToCart(product, price) {
```

```
  const existingProduct = cart.find(item => item.product === product);
```

```
  if (existingProduct) {
```

```
    existingProduct.quantity++;
```

```
  } else {
```

```
    cart.push({ product, price, quantity: 1 });
```

```
  }
```

```
  displayCart();
```

```
}
```

```
function removeFromCart(product) {
    cart = cart.filter(item => item.product !== product);
    displayCart();
}
```

```
function updateQuantity(product, quantity) {
    const cartItem = cart.find(item => item.product === product);
    if (cartItem) {
        cartItem.quantity = quantity;
        if (cartItem.quantity <= 0) {
            removeFromCart(product);
        }
    }
    displayCart();
}
```

```
function displayCart() {
    const cartItemsElement = document.getElementById('cart-items');
    cartItemsElement.innerHTML = "";
    let total = 0;
    cart.forEach(item => {
        const li = document.createElement('li');
        li.className = 'cart-item';
        li.innerHTML = `
            <span>${item.product} : ${item.price.toFixed(2)}</span>
            <div class="quantity">
                <button onclick="updateQuantity('${item.product}', ${item.quantity - 1})">-
</button>
                <input type="text" value="${item.quantity}" readonly>
                <button onclick="updateQuantity('${item.product}', ${item.quantity +
1})">+</button>
```

```
        </div>

        <button onclick="removeFromCart('${item.product}')">Remove</button>
    `;

    cartItemsElement.appendChild(li);

    total += item.price * item.quantity;

});

document.getElementById('total-price').innerText = `Total: ${total.toFixed(2)} `;
}
```

```
// Get the modal
```

```
const modal = document.getElementById('checkoutModal');
const span = document.getElementsByClassName('close')[0];
```

```
function checkout() {
    modal.style.display = 'block';
}
```

```
span.onclick = function () {
    modal.style.display = 'none';
}
```

```
window.onclick = function (event) {
    if (event.target == modal) {
        modal.style.display = 'none';
    }
}
```

```
function submitOrder() {
const name = document.getElementById('name').value;
const address = document.getElementById('address').value;
```

```
const phone = document.getElementById('phone').value;
const delivery = document.getElementById('delivery').value;
const orderDetails = {
  name,
  address,
  phone,
  delivery,
  cart
};

fetch('http://localhost:5000/submitOrder', { // Adjust URL if backend is hosted elsewhere
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(orderDetails)
})
.then(response => response.json())
.then(data => {
  console.log(data); // Handle response data
  alert('Order submitted!');
  cart = []; // Clear cart
  displayCart(); // Update cart display
  document.getElementById('checkoutModal').style.display = 'none'; // Close modal
})
.catch(error => {
  console.error('Error:', error);
});
}
```



```
</script>
</body>

</html>

Owener.Html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Owner Dashboard</title>
  <style>
    /* Add styles similar to the customer page, but adjusted for the owner page layout */
  </style>
</head>
<body>
  <header>
    <h1>Owner Dashboard</h1>
  </header>
  <div class="container">
    <section id="product-management">
      <h2>Manage Products</h2>
      <form id="productForm">
        <label for="category">Category:</label>
        <select id="category">
          <option value="">Select Category</option>
          <option value="Fruits">Fruits</option>
          <option value="Vegetables">Vegetables</option>
          <option value="Grains">Grains</option>
          <option value="Dairy">Dairy</option>
```

```

        <!-- Add more categories as needed -->
    </select>

    <label for="productName">Product Name:</label>
    <input type="text" id="productName" name="productName" required>

    <label for="productImage">Product Image URL:</label>
    <input type="text" id="productImage" name="productImage" required>

    <label for="productPrice">Price:</label>
    <input type="number" id="productPrice" name="productPrice" step="0.01"
required>

    <button type="button" onclick="addProduct()">Add Product</button>
</form>

<h3>Product List</h3>
<ul id="productList"></ul>
</section>
</div>

<script>
    const productForm = document.getElementById('productForm');
    const productList = document.getElementById('productList');

    async function addProduct() {
        const category = document.getElementById('category').value;
        const productName = document.getElementById('productName').value;
        const productImage = document.getElementById('productImage').value;
        const productPrice = document.getElementById('productPrice').value;

```

```

const product = {
  category,
  name: productName,
  image: productImage,
  price: parseFloat(productPrice)
};

// Send POST request to the backend to add the product
try {
  await fetch('http://localhost:5000/products', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify(product)
  });
  loadProducts();
  productForm.reset();
} catch (error) {
  console.error('Error adding product:', error);
}
}

async function loadProducts() {
  try {
    const response = await fetch('http://localhost:5000/products');
    const products = await response.json();
    productList.innerHTML = products.map(product => `
      <li>
        
        <h4>${product.name}</h4>
        <p>Category: ${product.category}</p>
    `);
  } catch (error) {
    console.error('Error loading products:', error);
  }
}

```

```

        <p>Price: ${product.price.toFixed(2)}</p>
        <button onclick="removeProduct('${product._id}')">Remove</button>
    </li>
    `).join("");
    } catch (error) {
        console.error('Error loading products:', error);
    }
}

async function removeProduct(id) {
    try {
        await fetch(`http://localhost:5000/products/${id}`, { method: 'DELETE' });
        loadProducts();
    } catch (error) {
        console.error('Error removing product:', error);
    }
}

// Load products on page load
loadProducts();
</script>
</body>
</html>

```

Cart.php

```

<?php
session_start();

// Initialize the cart session if it doesn't exist
if (!isset($_SESSION['cart'])) {
    $_SESSION['cart'] = [];
}

```

```

}

// Add product to cart if data is sent via POST
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['name']) &&
isset($_POST['price'])) {
    $name = $_POST['name'];
    $price = floatval($_POST['price']);

    // If product already exists in cart, increase quantity
    if (isset($_SESSION['cart'][$name])) {
        $_SESSION['cart'][$name]['quantity'] += 1;
    } else {
        $_SESSION['cart'][$name] = [
            'price' => $price,
            'quantity' => 1
        ];
    }
    echo json_encode(['status' => 'success', 'message' => "$name added to cart."]);
    exit;
}

// View cart (GET request)
if ($_SERVER['REQUEST_METHOD'] === 'GET') {
    header('Content-Type: application/json');
    echo json_encode($_SESSION['cart']);
    exit;
}
?>

```

OUTPUT:

Available Products



Rice 1 Kg

65.00

Add to Cart



Salt

20.00

Add to Cart



Apple

50.00

Add to Cart

Shopping Cart

Total: 0.00

Buy

Shopping Cart

Rice : 80.00 - 2 + Remove

Total: 160.00

Buy

×

Checkout Details

Name:

Address:

Phone Number:

Delivery Option:

Home Delivery

Submit

Owner Dashboard

Manage Products

Category: Vegetables Product Name: Product Image URL: Price:

Product List

RESULT:

Thus a organic grocery store application is implemented using php with mySQL.