AWL PRACTICAL LIST

1.Create a web page for Hostel application using different HTML tags and apply Css to it. create a form and add validation to it using JavaScript

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Hostel Application</title>
   <link rel="stylesheet" href="styles.css">
</head>
<body>
   <h1>Hostel Application Form</h1>
   <form id="hostelForm">
      <label for="name">Full Name:</label>
      <input type="text" id="name" name="name" required><br><br>

      <label for="age">Age:</label>
      <input type="number" id="age" name="age" required><br><br>

      <label for="gender">Gender:</label>
      <select id="gender" name="gender" required>
         <option value="">Select...</option>
         <option value="male">Male</option>
         <option value="female">Female</option>
         <option value="other">Other</option>
      </select><br><br>

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

      <label for="phone">Phone Number:</label>
      <input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required><br><br>

      <button type="submit">Submit</button>
   </form>
   <script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f8ff;
    margin: 0;
    padding: 20px;
}

h1 {
    text-align: center;
    color: #333;
}

form {
    max-width: 600px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

input, select {
    width: 100%;
    padding: 8px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
```

```
}

button:hover {
    background-color: #45a049;
}
```

script.js

```
document.getElementById('hostelForm').addEventListener('submit', function(event) {
    // Form validation
    let isValid = true;
    let form = event.target;

    // Name validation
    let name = form.name.value.trim();
    if (name === '') {
        isValid = false;
        alert('Please enter your full name.');
    }

    // Age validation
    let age = form.age.value;
    if (age === '' || isNaN(age) || age <= 0) {
        isValid = false;
        alert('Please enter a valid age.');
    }

    // Gender validation
    let gender = form.gender.value;
    if (gender === '') {
        isValid = false;
        alert('Please select your gender.');
    }

    // Email validation
    let email = form.email.value;
    let emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailPattern.test(email)) {
        isValid = false;
        alert('Please enter a valid email address.');
    }

    // Phone validation
    let phone = form.phone.value;
    let phonePattern = /^[0-9]{10}$/;
```

```
    if (!phonePattern.test(phone)) {
        isValid = false;
        alert('Please enter a valid 10-digit phone number.');
    }

    if (!isValid) {
        event.preventDefault();
    }
});
```

2.Create a web page for Hospital management using different HTML. tags and apply CSS to it, create a form and add validation to it using JavaScript

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hospital Management</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h1>Hospital Management Form</h1>
    <form id="hospitalForm">
        <label for="patientName">Patient Name:</label>
        <input type="text" id="patientName" name="patientName" required><br><br>

        <label for="patientAge">Patient Age:</label>
        <input type="number" id="patientAge" name="patientAge" required><br><br>

        <label for="patientGender">Patient Gender:</label>
        <select id="patientGender" name="patientGender" required>
            <option value="">Select...</option>
            <option value="male">Male</option>
            <option value="female">Female</option>
            <option value="other">Other</option>
        </select><br><br>

        <label for="disease">Disease:</label>
        <input type="text" id="disease" name="disease" required><br><br>

        <label for="admissionDate">Admission Date:</label>
        <input type="date" id="admissionDate" name="admissionDate" required><br><br>
```

```
        <button type="submit">Submit</button>
    </form>
    <script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #e0f7fa;
    margin: 0;
    padding: 20px;
}

h1 {
    text-align: center;
    color: #00796b;
}

form {
    max-width: 600px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #00796b;
    border-radius: 5px;
    background-color: #ffffff;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

input, select {
    width: 100%;
    padding: 8px;
    margin-bottom: 10px;
    border: 1px solid #00796b;
    border-radius: 3px;
}

button {
```

```
    width: 100%;
    padding: 10px;
    background-color: #00796b;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #004d40;
}
```

script.css

```
document.getElementById('hospitalForm').addEventListener('submit', function(event) {
    // Form validation
    let isValid = true;
    let form = event.target;

    // Patient Name validation
    let patientName = form.patientName.value.trim();
    if (patientName === '') {
        isValid = false;
        alert('Please enter the patient name.');
    }

    // Patient Age validation
    let patientAge = form.patientAge.value;
    if (patientAge === '' || isNaN(patientAge) || patientAge <= 0) {
        isValid = false;
        alert('Please enter a valid age.');
    }

    // Patient Gender validation
    let patientGender = form.patientGender.value;
    if (patientGender === '') {
        isValid = false;
        alert('Please select the patient gender.');
    }

    // Disease validation
    let disease = form.disease.value.trim();
    if (disease === '') {
        isValid = false;
```

```
        alert('Please enter the disease.');
    }

    // Admission Date validation
    let admissionDate = form.admissionDate.value;
    if (admissionDate === '') {
        isValid = false;
        alert('Please enter the admission date.');
    }

    if (!isValid) {
        event.preventDefault();
    }
});
```

3.Create a web page for Student Registration System for an institute using different HTML tags and apply CSS to it, create a form and nod validation to it using JavaScript

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Registration</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h1>Student Registration Form</h1>
    <form id="registrationForm">
        <label for="studentName">Student Name:</label>
        <input type="text" id="studentName" name="studentName" required><br><br>

        <label for="studentAge">Student Age:</label>
        <input type="number" id="studentAge" name="studentAge" required><br><br>

        <label for="studentGender">Student Gender:</label>
        <select id="studentGender" name="studentGender" required>
            <option value="">Select...</option>
            <option value="male">Male</option>
            <option value="female">Female</option>
            <option value="other">Other</option>
        </select><br><br>
```

```
        <label for="course">Course:</label>
        <input type="text" id="course" name="course" required><br><br>

        <label for="registrationDate">Registration Date:</label>
        <input type="date" id="registrationDate" name="registrationDate" required><br><br>

        <button type="submit">Submit</button>
    </form>
    <script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f3e5f5;
    margin: 0;
    padding: 20px;
}

h1 {
    text-align: center;
    color: #6a1b9a;
}

form {
    max-width: 600px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #6a1b9a;
    border-radius: 5px;
    background-color: #ffffff;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

input, select {
    width: 100%;
    padding: 8px;
    margin-bottom: 10px;
```

```
    border: 1px solid #6a1b9a;
    border-radius: 3px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #6a1b9a;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #4a148c;
}
```

script.css

```
document.getElementById('registrationForm').addEventListener('submit', function(event) {
    // Form validation
    let isValid = true;
    let form = event.target;

    // Student Name validation
    let studentName = form.studentName.value.trim();
    if (studentName === '') {
        isValid = false;
        alert('Please enter the student name.');
    }

    // Student Age validation
    let studentAge = form.studentAge.value;
    if (studentAge === '' || isNaN(studentAge) || studentAge <= 0) {
        isValid = false;
        alert('Please enter a valid age.');
    }

    // Student Gender validation
    let studentGender = form.studentGender.value;
    if (studentGender === '') {
        isValid = false;
        alert('Please select the student gender.');
    }
```

```
    // Course validation
    let course = form.course.value.trim();
    if (course === '') {
        isValid = false;
        alert('Please enter the course.');
    }

    // Registration Date validation
    let registrationDate = form.registrationDate.value;
    if (registrationDate === '') {
        isValid = false;
        alert('Please enter the registration date.');
    }

    if (!isValid) {
        event.preventDefault();
    }
});
```

4.Create an HTML Script for a Car Service Center utilizing HTML tags and CSS Features: Enhance content with numerous car workshop photographs and car accessories, as well its details on various car servicing packages. Perform JavaScript Validation

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Car Service Center</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Welcome to Our Car Service Center</h1>
        <nav>
            <ul>
                <li><a href="#services">Services</a></li>
                <li><a href="#packages">Packages</a></li>
                <li><a href="#contact">Contact Us</a></li>
            </ul>
        </nav>
    </header>
```

```
<section id="services">
  <h2>Our Services</h2>
  <p>We offer a variety of car services including:</p>
  <ul>
    <li>Engine Repair</li>
    <li>Oil Change</li>
    <li>Tire Replacement</li>
    <li>Battery Replacement</li>
    <li>Car Wash</li>
  </ul>
  <img src="car-workshop.jpg" alt="Car Workshop">
  <img src="car-accessories.jpg" alt="Car Accessories">
</section>

<section id="packages">
  <h2>Service Packages</h2>
  <div class="package">
    <h3>Basic Package</h3>
    <p>Includes: Oil Change, Tire Rotation, Car Wash</p>
    <p>Price: $50</p>
  </div>
  <div class="package">
    <h3>Standard Package</h3>
    <p>Includes: Oil Change, Tire Rotation, Car Wash, Battery Check</p>
    <p>Price: $80</p>
  </div>
  <div class="package">
    <h3>Premium Package</h3>
    <p>Includes: Oil Change, Tire Rotation, Car Wash, Battery Check, Full Inspection</p>
    <p>Price: $120</p>
  </div>
</section>

<section id="contact">
  <h2>Contact Us</h2>
  <form id="contactForm">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required><br><br>

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

    <label for="phone">Phone Number:</label>
```

```
        <input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required><br><br>

        <label for="package">Choose a package:</label>
        <select id="package" name="package" required>
            <option value="">Select...</option>
            <option value="basic">Basic</option>
            <option value="standard">Standard</option>
            <option value="premium">Premium</option>
        </select><br><br>

        <button type="submit">Submit</button>
    </form>
</section>

<script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f9f9f9;
    margin: 0;
    padding: 20px;
}

header {
    background-color: #333;
    color: white;
    padding: 10px 0;
    text-align: center;
}

nav ul {
    list-style-type: none;
    padding: 0;
}

nav ul li {
    display: inline;
    margin: 0 10px;
}

nav ul li a {
```

```
    color: white;
    text-decoration: none;
}

h1, h2 {
    color: #333;
}

section {
    margin-bottom: 30px;
}

.package {
    border: 1px solid #ccc;
    padding: 10px;
    margin-bottom: 10px;
}

img {
    max-width: 100%;
    height: auto;
    margin: 10px 0;
}

form {
    max-width: 600px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

input, select {
    width: 100%;
    padding: 8px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
```

```
    border-radius: 3px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}
```

script.js

```
document.getElementById('contactForm').addEventListener('submit', function(event) {
    // Form validation
    let isValid = true;
    let form = event.target;

    // Name validation
    let name = form.name.value.trim();
    if (name === '') {
        isValid = false;
        alert('Please enter your name.');
    }

    // Email validation
    let email = form.email.value;
    let emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailPattern.test(email)) {
        isValid = false;
        alert('Please enter a valid email address.');
    }

    // Phone validation
    let phone = form.phone.value;
    let phonePattern = /^[0-9]{10}$/;
    if (!phonePattern.test(phone)) {
        isValid = false;
        alert('Please enter a valid 10-digit phone number.');
```

```
    }

    // Package validation
    let packageSelected = form.package.value;
    if (packageSelected === '') {
        isValid = false;
        alert('Please select a service package.');
    }

    if (!isValid) {
        event.preventDefault();
    }
});
```

5.Create an HTML. Script for a Flower Shop utilizing HTML, fros and CSS features: Enhance content and output with flowers/bouquet photographs etc. as well as details on various car bouquet & combos. Perform JavaScript Validation.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flower Shop</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Welcome to Our Flower Shop</h1>
        <nav>
            <ul>
                <li><a href="#bouquets">Bouquets</a></li>
                <li><a href="#combos">Combos</a></li>
                <li><a href="#contact">Contact Us</a></li>
            </ul>
        </nav>
    </header>

    <section id="bouquets">
        <h2>Our Bouquets</h2>
        <div class="product">
            <img src="bouquet1.jpg" alt="Bouquet 1">
            <h3>Classic Bouquet</h3>
```

```
      <p>Price: $30</p>
    </div>
    <div class="product">
      <img src="bouquet2.jpg" alt="Bouquet 2">
      <h3>Rose Delight</h3>
      <p>Price: $40</p>
    </div>
    <div class="product">
      <img src="bouquet3.jpg" alt="Bouquet 3">
      <h3>Spring Mix</h3>
      <p>Price: $35</p>
    </div>
  </section>

  <section id="combos">
    <h2>Our Combos</h2>
    <div class="product">
      <img src="combo1.jpg" alt="Combo 1">
      <h3>Flower & Chocolate Combo</h3>
      <p>Price: $50</p>
    </div>
    <div class="product">
      <img src="combo2.jpg" alt="Combo 2">
      <h3>Flower & Teddy Combo</h3>
      <p>Price: $55</p>
    </div>
    <div class="product">
      <img src="combo3.jpg" alt="Combo 3">
      <h3>Flower & Wine Combo</h3>
      <p>Price: $60</p>
    </div>
  </section>

  <section id="contact">
    <h2>Contact Us</h2>
    <form id="contactForm">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name" required><br><br>

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

      <label for="phone">Phone Number:</label>
      <input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required><br><br>
```

```
        <label for="product">Choose a product:</label>
        <select id="product" name="product" required>
          <option value="">Select...</option>
          <option value="classic-bouquet">Classic Bouquet</option>
          <option value="rose-delight">Rose Delight</option>
          <option value="spring-mix">Spring Mix</option>
          <option value="flower-chocolate-combo">Flower & Chocolate Combo</option>
          <option value="flower-teddy-combo">Flower & Teddy Combo</option>
          <option value="flower-wine-combo">Flower & Wine Combo</option>
        </select><br><br>

        <button type="submit">Submit</button>
      </form>
  </section>

  <script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f8ff;
  margin: 0;
  padding: 20px;
}

header {
  background-color: #228b22;
  color: white;
  padding: 10px 0;
  text-align: center;
}

nav ul {
  list-style-type: none;
  padding: 0;
}

nav ul li {
  display: inline;
  margin: 0 10px;
}
```

```
nav ul li a {
    color: white;
    text-decoration: none;
}

h1, h2 {
    color: #228b22;
}

section {
    margin-bottom: 30px;
}

.product {
    border: 1px solid #ccc;
    padding: 10px;
    margin-bottom: 10px;
    text-align: center;
}

.product img {
    max-width: 100%;
    height: auto;
    margin: 10px 0;
}

form {
    max-width: 600px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

input, select {
    width: 100%;
```

```
    padding: 8px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #228b22;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #196619;
}
```

script.js

```
document.getElementById('contactForm').addEventListener('submit', function(event) {
    // Form validation
    let isValid = true;
    let form = event.target;

    // Name validation
    let name = form.name.value.trim();
    if (name === '') {
        isValid = false;
        alert('Please enter your name.');
    }

    // Email validation
    let email = form.email.value;
    let emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailPattern.test(email)) {
        isValid = false;
        alert('Please enter a valid email address.');
    }

    // Phone validation
    let phone = form.phone.value;
    let phonePattern = /^[0-9]{10}$/;
```

```
    if (!phonePattern.test(phone)) {
        isValid = false;
        alert('Please enter a valid 10-digit phone number.');
    }

    // Product validation
    let productSelected = form.product.value;
    if (productSelected === '') {
        isValid = false;
        alert('Please select a product.');
    }

    if (!isValid) {
        event.preventDefault();
    }
});
```

6. Design an e-commerce product page with HTML and CSS, create validation to it using Javascript

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>E-commerce Product Page</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Product Page</h1>
        <nav>
            <ul>
                <li><a href="#products">Products</a></li>
                <li><a href="#cart">Cart</a></li>
                <li><a href="#contact">Contact</a></li>
            </ul>
        </nav>
    </header>

    <section id="products">
        <h2>Our Products</h2>
        <div class="product">
```

```
        <img src="product1.jpg" alt="Product 1">
        <h3>Product 1</h3>
        <p>Price: $50</p>
        <button onclick="addToCart('Product 1', 50)">Add to Cart</button>
      </div>
      <div class="product">
        <img src="product2.jpg" alt="Product 2">
        <h3>Product 2</h3>
        <p>Price: $70</p>
        <button onclick="addToCart('Product 2', 70)">Add to Cart</button>
      </div>
      <div class="product">
        <img src="product3.jpg" alt="Product 3">
        <h3>Product 3</h3>
        <p>Price: $100</p>
        <button onclick="addToCart('Product 3', 100)">Add to Cart</button>
      </div>
    </section>

    <section id="cart">
      <h2>Your Cart</h2>
      <form id="cartForm">
        <div id="cartItems"></div>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required><br><br>

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

        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required><br><br>

        <button type="submit">Checkout</button>
      </form>
    </section>

    <script src="script.js"></script>
  </body>
  </html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f9f9f9;
    margin: 0;
    padding: 20px;
}

header {
    background-color: #333;
    color: white;
    padding: 10px 0;
    text-align: center;
}

nav ul {
    list-style-type: none;
    padding: 0;
}

nav ul li {
    display: inline;
    margin: 0 10px;
}

nav ul li a {
    color: white;
    text-decoration: none;
}

h1, h2 {
    color: #333;
}

section {
    margin-bottom: 30px;
}

.product {
    border: 1px solid #ccc;
    padding: 10px;
    margin-bottom: 10px;
    text-align: center;
}
```

```
.product img {
    max-width: 100%;
    height: auto;
    margin: 10px 0;
}

form {
    max-width: 600px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

input, select {
    width: 100%;
    padding: 8px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}
```

script.js

```
let cart = [];

function addToCart(productName, productPrice) {
    cart.push({ name: productName, price: productPrice });
    updateCart();
}

function updateCart() {
    let cartItemsDiv = document.getElementById('cartItems');
    cartItemsDiv.innerHTML = '';

    let total = 0;
    cart.forEach((item, index) => {
        total += item.price;
        cartItemsDiv.innerHTML += `
            <div class="cart-item">
                <p>${item.name} - $${item.price}</p>
                <button onclick="removeFromCart(${index})">Remove</button>
            </div>
        `;
    });

    cartItemsDiv.innerHTML += `<p>Total: $${total}</p>`;
}

function removeFromCart(index) {
    cart.splice(index, 1);
    updateCart();
}

document.getElementById('cartForm').addEventListener('submit', function(event) {
    let isValid = true;
    let form = event.target;

    // Name validation
    let name = form.name.value.trim();
    if (name === '') {
        isValid = false;
        alert('Please enter your name.');
    }

    // Email validation
    let email = form.email.value;
```

```
    let emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailPattern.test(email)) {
        isValid = false;
        alert('Please enter a valid email address.');
    }

    // Address validation
    let address = form.address.value.trim();
    if (address === '') {
        isValid = false;
        alert('Please enter your address.');
    }

    if (!isValid) {
        event.preventDefault();
    } else {
        alert('Thank you for your purchase!');
        cart = [];
        updateCart();
    }
});
```

7.Create an interactive quiz application using HTML. CSS and JavaScript. The quiz should have the following features:

- **Display a set of multiple-choice questions with options**
- **Allow users to select an option for each question.**
- **Display the total number of correct answers after submitting the quiz.**
- **Include a "Submit" button to check the answers**

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Quiz Application</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Quiz Application</h1>
    </header>
```

```
<section id="quiz">
  <form id="quizForm">
    <div class="question">
      <h2>1. What is the capital of France?</h2>
      <label><input type="radio" name="q1" value="A"> A. Berlin</label><br>
      <label><input type="radio" name="q1" value="B"> B. Madrid</label><br>
      <label><input type="radio" name="q1" value="C"> C. Paris</label><br>
      <label><input type="radio" name="q1" value="D"> D. Rome</label>
    </div>
    <div class="question">
      <h2>2. Who wrote "To Kill a Mockingbird"?</h2>
      <label><input type="radio" name="q2" value="A"> A. Harper Lee</label><br>
      <label><input type="radio" name="q2" value="B"> B. Mark Twain</label><br>
      <label><input type="radio" name="q2" value="C"> C. J.K. Rowling</label><br>
      <label><input type="radio" name="q2" value="D"> D. Ernest Hemingway</label>
    </div>
    <div class="question">
      <h2>3. What is the largest planet in our solar system?</h2>
      <label><input type="radio" name="q3" value="A"> A. Earth</label><br>
      <label><input type="radio" name="q3" value="B"> B. Jupiter</label><br>
      <label><input type="radio" name="q3" value="C"> C. Saturn</label><br>
      <label><input type="radio" name="q3" value="D"> D. Mars</label>
    </div>
    <div class="question">
      <h2>4. What is the chemical symbol for gold?</h2>
      <label><input type="radio" name="q4" value="A"> A. Au</label><br>
      <label><input type="radio" name="q4" value="B"> B. Ag</label><br>
      <label><input type="radio" name="q4" value="C"> C. Pb</label><br>
      <label><input type="radio" name="q4" value="D"> D. Fe</label>
    </div>
    <div class="question">
      <h2>5. Who painted the Mona Lisa?</h2>
      <label><input type="radio" name="q5" value="A"> A. Vincent van Gogh</label><br>
      <label><input type="radio" name="q5" value="B"> B. Pablo Picasso</label><br>
      <label><input type="radio" name="q5" value="C"> C. Leonardo da Vinci</label><br>
      <label><input type="radio" name="q5" value="D"> D. Claude Monet</label>
    </div>
    <button type="button" onclick="checkAnswers()">Submit</button>
  </form>
</section>

<section id="result">
  <h2>Your Score: <span id="score"></span></h2>
</section>
```

```
    <script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f8ff;
    margin: 0;
    padding: 20px;
}

header {
    background-color: #4CAF50;
    color: white;
    padding: 10px 0;
    text-align: center;
}

h1, h2 {
    color: #333;
}

section {
    margin-bottom: 30px;
}

.question {
    margin-bottom: 20px;
    padding: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
```

```
}

button:hover {
  background-color: #45a049;
}

#result {
  display: none;
  text-align: center;
}
```

script.js

```
function checkAnswers() {
  const correctAnswers = {
    q1: 'C',
    q2: 'A',
    q3: 'B',
    q4: 'A',
    q5: 'C'
  };

  let score = 0;
  let totalQuestions = Object.keys(correctAnswers).length;

  for (let question in correctAnswers) {
    let selectedOption = document.querySelector(`input[name="${question}"]:checked`);
    if (selectedOption && selectedOption.value === correctAnswers[question]) {
      score++;
    }
  }

  document.getElementById('score').textContent = `${score} out of ${totalQuestions}`;
  document.getElementById('result').style.display = 'block';
}
```

8.Create a Survey For using MIMI. CSS and JAvaScript. The form should have the following features:

- **Display a series of questions with multiple-choice, rating, and open-ended responses**
- **Allow users to select or input their responses**
- **Include a "Submit" button to submit the survey**
- **Display a thank you message and optionally summarize the responses**

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Survey Form</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Survey Form</h1>
    </header>

    <section id="survey">
        <form id="surveyForm">
            <div class="question">
                <h2>1. How satisfied are you with our service?</h2>
                <label><input type="radio" name="q1" value="Very Satisfied"> Very Satisfied</label><br>
                <label><input type="radio" name="q1" value="Satisfied"> Satisfied</label><br>
                <label><input type="radio" name="q1" value="Neutral"> Neutral</label><br>
                <label><input type="radio" name="q1" value="Dissatisfied"> Dissatisfied</label><br>
                <label><input type="radio" name="q1" value="Very Dissatisfied"> Very Dissatisfied</label>
            </div>
            <div class="question">
                <h2>2. Please rate our customer support:</h2>
                <label for="q2">Rating:</label>
                <input type="number" id="q2" name="q2" min="1" max="5">
            </div>
            <div class="question">
                <h2>3. What did you like most about our service?</h2>
                <textarea id="q3" name="q3" rows="4" cols="50"></textarea>
            </div>
            <div class="question">
                <h2>4. Any suggestions for improvement?</h2>
                <textarea id="q4" name="q4" rows="4" cols="50"></textarea>
            </div>
            <button type="button" onclick="submitSurvey()">Submit</button>
        </form>
    </section>

    <section id="thankYou" style="display: none;">
        <h2>Thank you for your feedback!</h2>
        <p>Your responses have been submitted successfully.</p>
```

```
        <div id="summary"></div>
    </section>

    <script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f8ff;
    margin: 0;
    padding: 20px;
}

header {
    background-color: #4CAF50;
    color: white;
    padding: 10px 0;
    text-align: center;
}

h1, h2 {
    color: #333;
}

section {
    margin-bottom: 30px;
}

.question {
    margin-bottom: 20px;
    padding: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
}

textarea {
    width: 100%;
    padding: 8px;
    margin-top: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
```

```
}

button {
    width: 100%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

#thankYou {
    text-align: center;
}
```

script.js

```
function submitSurvey() {
    const surveyForm = document.getElementById('surveyForm');
    const responses = {
        q1: surveyForm.q1.value,
        q2: surveyForm.q2.value,
        q3: surveyForm.q3.value,
        q4: surveyForm.q4.value
    };

    // Display thank you message
    document.getElementById('survey').style.display = 'none';
    document.getElementById('thankYou').style.display = 'block';

    // Optionally display summary of responses
    const summaryDiv = document.getElementById('summary');
    summaryDiv.innerHTML = `
        <h3>Summary of your responses:</h3>
        <p><strong>Question 1:</strong> ${responses.q1}</p>
        <p><strong>Question 2:</strong> ${responses.q2}</p>
        <p><strong>Question 3:</strong> ${responses.q3}</p>
        <p><strong>Question 4:</strong> ${responses.q4}</p>
    `;
}
```

9. Design a True or False Quiz using HTML. CSS, and JavaScript. The quiz should have the following features:

Display a series of true or false questions
Allow users to select either true or false for each question.
Include a Submit button to check the answers.
Display the total number of correct answers after submitting the quiz.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>True or False Quiz</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>True or False Quiz</h1>
  </header>

  <section id="quiz">
    <form id="quizForm">
      <div class="question">
        <h2>1. The sky is blue.</h2>
        <label><input type="radio" name="q1" value="true"> True</label><br>
        <label><input type="radio" name="q1" value="false"> False</label>
      </div>
      <div class="question">
        <h2>2. The capital of Australia is Sydney.</h2>
        <label><input type="radio" name="q2" value="true"> True</label><br>
        <label><input type="radio" name="q2" value="false"> False</label>
      </div>
      <div class="question">
        <h2>3. Water boils at 100 degrees Celsius.</h2>
        <label><input type="radio" name="q3" value="true"> True</label><br>
        <label><input type="radio" name="q3" value="false"> False</label>
      </div>
      <div class="question">
        <h2>4. The human body has 206 bones.</h2>
        <label><input type="radio" name="q4" value="true"> True</label><br>
        <label><input type="radio" name="q4" value="false"> False</label>
      </div>
      <div class="question">
        <h2>5. The Great Wall of China is visible from space.</h2>
```

```
            <label><input type="radio" name="q5" value="true"> True</label><br>
            <label><input type="radio" name="q5" value="false"> False</label>
        </div>
        <button type="button" onclick="checkAnswers()">Submit</button>
    </form>
</section>

<section id="result" style="display: none;">
    <h2>Your Score: <span id="score"></span></h2>
</section>

<script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f8ff;
    margin: 0;
    padding: 20px;
}

header {
    background-color: #4CAF50;
    color: white;
    padding: 10px 0;
    text-align: center;
}

h1, h2 {
    color: #333;
}

section {
    margin-bottom: 30px;
}

.question {
    margin-bottom: 20px;
    padding: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
```

```
}

button {
    width: 100%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

#result {
    text-align: center;
}
```

script.js

```
function checkAnswers() {
    const correctAnswers = {
        q1: 'true',
        q2: 'false',
        q3: 'true',
        q4: 'true',
        q5: 'false'
    };

    let score = 0;
    let totalQuestions = Object.keys(correctAnswers).length;

    for (let question in correctAnswers) {
        let selectedOption = document.querySelector(`input[name="${question}"]:checked`);
        if (selectedOption && selectedOption.value === correctAnswers[question]) {
            score++;
        }
    }

    document.getElementById('score').textContent = `${score} out of ${totalQuestions}`;
    document.getElementById('result').style.display = 'block';
}
```

10. Create a web page for Grocery application using different HTML tags and apply CSS to it, create a form and add validation to it using AngularJS.

index.html

```
<!DOCTYPE html>
<html lang="en" ng-app="groceryApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Grocery Application</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body ng-controller="GroceryController as groceryCtrl">
  <header>
    <h1>Grocery Application</h1>
  </header>

  <section id="groceryForm">
    <form name="groceryForm" ng-submit="groceryCtrl.submitForm()" novalidate>
      <div class="form-group">
        <label>Name:</label>
        <input type="text" name="name" ng-model="groceryCtrl.item.name" required>
        <div ng-show="groceryForm.name.$dirty && groceryForm.name.$error.required"
class="error">Name is required.</div>
      </div>
      <div class="form-group">
        <label>Quantity:</label>
        <input type="number" name="quantity" ng-model="groceryCtrl.item.quantity" required>
        <div ng-show="groceryForm.quantity.$dirty && groceryForm.quantity.$error.required"
class="error">Quantity is required.</div>
      </div>
      <div class="form-group">
        <label>Price:</label>
        <input type="number" name="price" ng-model="groceryCtrl.item.price" required>
        <div ng-show="groceryForm.price.$dirty && groceryForm.price.$error.required"
class="error">Price is required.</div>
      </div>
      <button type="submit" ng-disabled="groceryForm.$invalid">Submit</button>
    </form>
  </section>

  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script src="app.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f8ff;
    margin: 0;
    padding: 20px;
}

header {
    background-color: #4CAF50;
    color: white;
    padding: 10px 0;
    text-align: center;
}

form {
    max-width: 400px;
    margin: 0 auto;
}

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

label {
    display: block;
    font-weight: bold;
}

input[type="text"],
input[type="number"] {
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
```

```
}

button:hover {
  background-color: #45a049;
}

.error {
  color: red;
  font-size: 12px;
  margin-top: 5px;
}
```

app.js

```
angular.module('groceryApp', [])
  .controller('GroceryController', function() {
    var vm = this;
    vm.item = {};

    vm.submitForm = function() {
      // Process form submission here
      console.log('Form submitted:', vm.item);
      // Clear form after submission
      vm.item = {};
      vm.groceryForm.$setPristine();
    };
  });
```

11.Create a web page for Tourism Application using different HTML tags and apply CSS to it, create a form and add validation to it using AngularIS.

index.html

```
<!DOCTYPE html>
<html lang="en" ng-app="tourismApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tourism Application</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body ng-controller="TourismController as tourismCtrl">
  <header>
    <h1>Tourism Application</h1>
  </header>
```

```
  <section id="tourismForm">
    <form name="tourismForm" ng-submit="tourismCtrl.submitForm()" novalidate>
      <div class="form-group">
        <label>Name:</label>
        <input type="text" name="name" ng-model="tourismCtrl.booking.name" required>
        <div ng-show="tourismForm.name.$dirty && tourismForm.name.$error.required"
class="error">Name is required.</div>
      </div>
      <div class="form-group">
        <label>Email:</label>
        <input type="email" name="email" ng-model="tourismCtrl.booking.email" required>
        <div ng-show="tourismForm.email.$dirty && tourismForm.email.$error.required"
class="error">Email is required.</div>
        <div ng-show="tourismForm.email.$dirty && tourismForm.email.$error.email"
class="error">Invalid email format.</div>
      </div>
      <div class="form-group">
        <label>Destination:</label>
        <input type="text" name="destination" ng-model="tourismCtrl.booking.destination" required>
        <div ng-show="tourismForm.destination.$dirty && tourismForm.destination.$error.required"
class="error">Destination is required.</div>
      </div>
      <button type="submit" ng-disabled="tourismForm.$invalid">Submit</button>
    </form>
  </section>

  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script src="app.js"></script>
</body>
</html>
```

styles.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f8ff;
  margin: 0;
  padding: 20px;
}

header {
  background-color: #4CAF50;
  color: white;
  padding: 10px 0;
  text-align: center;
```

```
}

form {
  max-width: 400px;
  margin: 0 auto;
}

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

label {
  display: block;
  font-weight: bold;
}

input[type="text"],
input[type="email"] {
  width: 100%;
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 3px;
}

button {
  padding: 10px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 3px;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}

.error {
  color: red;
  font-size: 12px;
  margin-top: 5px;
}
```

app.js

```
angular.module('tourismApp', [])
  .controller('TourismController', function() {
    var vm = this;
    vm.booking = {};

    vm.submitForm = function() {
      // Process form submission here
      console.log('Form submitted:', vm.booking);
      // Clear form after submission
      vm.booking = {};
      vm.tourismForm.$setPristine();
    };
  });
```

12. Create a web page for a Complaint Management Application using different HTML lags and apply CSS to it, create a form and add validation to it using AngularIS.

index.html

```
<!DOCTYPE html>
<html lang="en" ng-app="complaintApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Complaint Management Application</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body ng-controller="ComplaintController as complaintCtrl">
  <header>
    <h1>Complaint Management Application</h1>
  </header>

  <section id="complaintForm">
    <form name="complaintForm" ng-submit="complaintCtrl.submitForm()" novalidate>
      <div class="form-group">
        <label>Name:</label>
        <input type="text" name="name" ng-model="complaintCtrl.complaint.name" required>
        <div ng-show="complaintForm.name.$dirty && complaintForm.name.$error.required"
class="error">Name is required.</div>
      </div>
      <div class="form-group">
        <label>Email:</label>
        <input type="email" name="email" ng-model="complaintCtrl.complaint.email" required>
```

```
            <div ng-show="complaintForm.email.$dirty && complaintForm.email.$error.required"
class="error">Email is required.</div>
            <div ng-show="complaintForm.email.$dirty && complaintForm.email.$error.email"
class="error">Invalid email format.</div>
          </div>
          <div class="form-group">
            <label>Complaint:</label>
            <textarea name="complaint" ng-model="complaintCtrl.complaint.complaint"
required></textarea>
            <div ng-show="complaintForm.complaint.$dirty &&
complaintForm.complaint.$error.required" class="error">Complaint is required.</div>
          </div>
          <button type="submit" ng-disabled="complaintForm.$invalid">Submit</button>
        </form>
    </section>

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script src="app.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f8ff;
    margin: 0;
    padding: 20px;
}

header {
    background-color: #4CAF50;
    color: white;
    padding: 10px 0;
    text-align: center;
}

form {
    max-width: 400px;
    margin: 0 auto;
}

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

```
label {
    display: block;
    font-weight: bold;
}

input[type="text"],
input[type="email"],
textarea {
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

textarea {
    height: 100px;
}

button {
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

.error {
    color: red;
    font-size: 12px;
    margin-top: 5px;
}
```

app.js

```
angular.module('complaintApp', [])
    .controller('ComplaintController', function() {
        var vm = this;
        vm.complaint = {};
```

```
    vm.submitForm = function() {
      // Process form submission here
      console.log('Form submitted:', vm.complaint);
      // Clear form after submission
      vm.complaint = {};
      vm.complaintForm.$setPristine();
    };
  });
```

13. Create a web page for Cake Shop using different HTML tags and apply CSS to it, create a form and add validation to it using AngularJS.

index.html

```
<!DOCTYPE html>
<html lang="en" ng-app="cakeShopApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cake Shop</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body ng-controller="CakeController as cakeCtrl">
  <header>
    <h1>Cake Shop</h1>
  </header>

  <section id="cakeForm">
    <form name="cakeForm" ng-submit="cakeCtrl.submitForm()" novalidate>
      <div class="form-group">
        <label>Name:</label>
        <input type="text" name="name" ng-model="cakeCtrl.order.name" required>
        <div ng-show="cakeForm.name.$dirty && cakeForm.name.$error.required"
class="error">Name is required.</div>
      </div>
      <div class="form-group">
        <label>Email:</label>
        <input type="email" name="email" ng-model="cakeCtrl.order.email" required>
        <div ng-show="cakeForm.email.$dirty && cakeForm.email.$error.required"
class="error">Email is required.</div>
        <div ng-show="cakeForm.email.$dirty && cakeForm.email.$error.email" class="error">Invalid
email format.</div>
      </div>
      <div class="form-group">
        <label>Cake Type:</label>
```

```
            <select name="cakeType" ng-model="cakeCtrl.order.cakeType" required>
               <option value="">Select</option>
               <option value="Chocolate">Chocolate</option>
               <option value="Vanilla">Vanilla</option>
               <option value="Strawberry">Strawberry</option>
            </select>
            <div ng-show="cakeForm.cakeType.$dirty && cakeForm.cakeType.$error.required"
class="error">Cake Type is required.</div>
         </div>
         <button type="submit" ng-disabled="cakeForm.$invalid">Submit</button>
      </form>
   </section>

   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
   <script src="app.js"></script>
</body>
</html>
```

styles.css

```
body {
   font-family: Arial, sans-serif;
   background-color: #f0f8ff;
   margin: 0;
   padding: 20px;
}

header {
   background-color: #4CAF50;
   color: white;
   padding: 10px 0;
   text-align: center;
}

form {
   max-width: 400px;
   margin: 0 auto;
}

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

label {
   display: block;
```

```
    font-weight: bold;
}

input[type="text"],
input[type="email"],
select {
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

.error {
    color: red;
    font-size: 12px;
    margin-top: 5px;
}
```

app.js

```
angular.module('cakeShopApp', [])
    .controller('CakeController', function() {
        var vm = this;
        vm.order = {};

        vm.submitForm = function() {
            // Process form submission here
            console.log('Form submitted:', vm.order);
            // Clear form after submission
            vm.order = {};
            vm.cakeForm.$setPristine();
        };
```

```
});
```

14. Create a React application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages.

15. Create a React application for the employee management system having registration, login, contact, about pages and implement routing to navigate through these pages.

16. Create a React application for the hotel management system having registration, login, contact, about pages and implement routing to navigate through these pages.

17. Create a React application for the Bookshop having registration, login, contact, about pages and implement routing to navigate through these pages.

18. Create a react application for the Online Clothes Shopping system having registration, login, contacl, about pages and implement routing to navigate through these pages.

- 19. Create a To-do Application using React and deploy it. Create add and delete features.

20. Create a To-do Application using AngularJS. Create add and delete features.

index.html

```
<!DOCTYPE html>
<html lang="en" ng-app="todoApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>To-do Application</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="TodoController as todoCtrl">
  <h1>To-do Application</h1>
  <ul>
    <li ng-repeat="task in todoCtrl.tasks track by $index">
      {{ task }}
      <button ng-click="todoCtrl.deleteTask($index)">Delete</button>
    </li>
  </ul>
```

```
    <input type="text" ng-model="todoCtrl.newTask" placeholder="Add a new task">
    <button ng-click="todoCtrl.addTask()">Add</button>

    <script src="app.js"></script>
</body>
</html>
```

styles.css

app.js

```
angular.module('todoApp', [])
    .controller('TodoController', function() {
        var vm = this;
        vm.tasks = [];
        vm.newTask = '';

        vm.addTask = function() {
            if (vm.newTask.trim() !== '') {
                vm.tasks.push(vm.newTask);
                vm.newTask = '';
            }
        };

        vm.deleteTask = function(index) {
            vm.tasks.splice(index, 1);
        };
    });
```

21. Create a To-do Application using JavaScript. Create add and delete features.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>To-do Application</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<div class="container">
<h1>To-do Application</h1>
<input type="text" id="taskInput" placeholder="Enter a new task">
<button onclick="addTask()">Add</button>
```

```
<ul id="taskList"></ul>
</div>
<script src="app.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
}

.container {
    max-width: 600px;
    margin: 50px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

h1 {
    text-align: center;
    margin-bottom: 20px;
}

input[type="text"] {
    width: 70%;
    padding: 8px;
    margin-right: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    padding: 8px 15px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}
```

```
button:hover {
    background-color: #45a049;
}

ul {
    list-style-type: none;
    padding: 0;
}

li {
    padding: 10px;
    margin-bottom: 5px;
    background-color: #f9f9f9;
    border-radius: 3px;
}

.delete {
    float: right;
    color: red;
    cursor: pointer;
}
```

app.js

```
let tasks = [];

function addTask() {
    const taskInput = document.getElementById('taskInput');
    const taskList = document.getElementById('taskList');

    if (taskInput.value.trim() !== '') {
        tasks.push(taskInput.value.trim());
        displayTasks();
        taskInput.value = '';
    }
}

function deleteTask(index) {
    tasks.splice(index, 1);
    displayTasks();
}

function displayTasks() {
    const taskList = document.getElementById('taskList');
```

```
    taskList.innerHTML = '';

    tasks.forEach((task, index) => {
        const li = document.createElement('li');
        li.textContent = task;

        const deleteButton = document.createElement('span');
        deleteButton.textContent = 'x';
        deleteButton.className = 'delete';
        deleteButton.onclick = () => deleteTask(index);

        li.appendChild(deleteButton);
        taskList.appendChild(li);
    });
}

window.onload = function () {
    displayTasks();
};
```

22. Create a custom server using the http module and explore the other modules of Node IS like OS, path, event

http

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello, World!\n');
});

const PORT = 3000;
server.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}/`);
});
```

os

```
const os = require('os');

console.log('OS Platform:', os.platform());
console.log('OS Architecture:', os.arch());
console.log('CPU Cores:', os.cpus().length);
console.log('Total Memory:', os.totalmem() / 1024 / 1024, 'MB');
```

```
console.log('Free Memory:', os.freemem() / 1024 / 1024, 'MB');
```

path

```
const path = require('path');

const filePath = '/path/to/file.txt';
console.log('File Name:', path.basename(filePath));
console.log('Directory Name:', path.dirname(filePath));
console.log('File Extension:', path.extname(filePath));
```

events

```
const EventEmitter = require('events');

class MyEmitter extends EventEmitter {}

const myEmitter = new MyEmitter();

myEmitter.on('greet', () => {
  console.log('Hello, World!');
});

myEmitter.emit('greet');
```

23. Create a course management system using MySQL connectivity with Node.S to perform CRUD operations using MySQL Connectivity with Node JS

Setting up the project

```
mkdir course-management-system
cd course-management-system
npm init -y
npm install express mysql body-parser
```

Creating database

```
CREATE DATABASE course_management;

USE course_management;

CREATE TABLE courses (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    duration INT
);
```

Setting up the node.js server(server.js)

```
const express = require('express');
const mysql = require('mysql');
const bodyParser = require('body-parser');

const app = express();
const port = 3000;

app.use(bodyParser.json());

// MySQL connection
const db = mysql.createConnection({
  host: 'localhost',
  user: 'your-username',
  password: 'your-password',
  database: 'course_management'
});

db.connect(err => {
  if (err) throw err;
  console.log('MySQL connected...');
});

// CRUD operations

// Create a new course
app.post('/courses', (req, res) => {
  const newCourse = req.body;
  const sql = 'INSERT INTO courses SET ?';
  db.query(sql, newCourse, (err, result) => {
    if (err) throw err;
    res.send('Course added...');
  });
});

// Read all courses
app.get('/courses', (req, res) => {
  const sql = 'SELECT * FROM courses';
  db.query(sql, (err, results) => {
    if (err) throw err;
    res.send(results);
  });
});
```

```
// Read a single course by ID
app.get('/courses/:id', (req, res) => {
  const sql = 'SELECT * FROM courses WHERE id = ?';
  db.query(sql, [req.params.id], (err, result) => {
    if (err) throw err;
    res.send(result);
  });
});

// Update a course by ID
app.put('/courses/:id', (req, res) => {
  const updatedCourse = req.body;
  const sql = 'UPDATE courses SET ? WHERE id = ?';
  db.query(sql, [updatedCourse, req.params.id], (err, result) => {
    if (err) throw err;
    res.send('Course updated...');
  });
});

// Delete a course by ID
app.delete('/courses/:id', (req, res) => {
  const sql = 'DELETE FROM courses WHERE id = ?';
  db.query(sql, [req.params.id], (err, result) => {
    if (err) throw err;
    res.send('Course deleted...');
  });
});

app.listen(port, () => {
  console.log(`Server running on port ${port}`);
});
```

24. Create a web page for movie ticket booking application using different HTML tags and apply CSS to it, create a form and add validation to it using JavaScript

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Movie Ticket Booking</title>
<link rel="stylesheet" href="styles.css">
</head>
```

```
<body>
<div class="container">
<h1>Movie Ticket Booking</h1>
<form id="bookingForm" onsubmit="return validateForm()">
<label for="name">Name:</label>
<input type="text" id="name" name="name" required>
<label for="email">Email:</label>
<input type="email" id="email" name="email" required>
<label for="movie">Select Movie:</label>
<select id="movie" name="movie" required>
<option value="">Select</option>
<option value="movie1">Movie 1</option>
<option value="movie2">Movie 2</option>
<option value="movie3">Movie 3</option>
</select>
<label for="tickets">Number of Tickets:</label>
<input type="number" id="tickets" name="tickets" min="1" max="10" required>
<button type="submit">Book Tickets</button>
</form>
<div id="message"></div>
</div>
<script src="script.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
}

.container {
    max-width: 600px;
    margin: 50px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

h1 {
    text-align: center;
```

```
    margin-bottom: 20px;
}

form {
    display: flex;
    flex-direction: column;
}

label {
    margin-bottom: 5px;
}

input,
select,
button {
    margin-bottom: 10px;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    background-color: #4CAF50;
    color: white;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

#message {
    margin-top: 20px;
    text-align: center;
    font-weight: bold;
}
```

script.js

```
function validateForm() {
    var name = document.getElementById('name').value;
    var email = document.getElementById('email').value;
    var movie = document.getElementById('movie').value;
    var tickets = document.getElementById('tickets').value;
```

```
    if (name.trim() === '' || email.trim() === '' || movie.trim() === '' || tickets.trim() === '') {
        showMessage('Please fill in all fields');
        return false;
    }

    if (!validateEmail(email)) {
        showMessage('Invalid email address');
        return false;
    }

    showMessage('Tickets booked successfully');
    return true;
}

function validateEmail(email) {
    var re = /\S+@\S+\.\S+/;
    return re.test(email);
}

function showMessage(message) {
    var messageElement = document.getElementById('message');
    messageElement.textContent = message;
}
```

25. Create a web page for a Doctor's Appointment Application using different HTML tags and apply CSS to it, create a form and add validation to it using AngularJS.

index.html

```
<!DOCTYPE html>
<html lang="en" ng-app="appointmentApp">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Doctor's Appointment Application</title>
<link rel="stylesheet" href="styles.css">
</head>
<body ng-controller="AppointmentController as appCtrl">
<div class="container">
<h1>Doctor's Appointment Application</h1>
<form name="appointmentForm" novalidate>
<label for="name">Name:</label>
<input type="text" id="name" name="name" ng-model="appCtrl.appointment.name" required>
<label for="email">Email:</label>
```

```
<input type="email" id="email" name="email" ng-model="appCtrl.appointment.email" required>
<label for="date">Date:</label>
<input type="date" id="date" name="date" ng-model="appCtrl.appointment.date" required>
<label for="time">Time:</label>
<input type="time" id="time" name="time" ng-model="appCtrl.appointment.time" required>
<button type="submit" ng-disabled="appointmentForm.$invalid"
ng-click="appCtrl.bookAppointment()">Book Appointment</button>
</form>
<div ng-if="appCtrl.successMessage" class="success-message">{{ appCtrl.successMessage }}</div>
</div>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<script src="app.js"></script>
</body>
</html>
```

styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
}

.container {
    max-width: 600px;
    margin: 50px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

h1 {
    text-align: center;
    margin-bottom: 20px;
}

form {
    display: flex;
    flex-direction: column;
}

label {
    margin-bottom: 5px;
```

```
}

input,
button {
    margin-bottom: 10px;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    background-color: #4CAF50;
    color: white;
    border: none;
    cursor: pointer;
}

button:disabled {
    background-color: #ccc;
    color: #888;
    cursor: not-allowed;
}

.success-message {
    margin-top: 20px;
    text-align: center;
    font-weight: bold;
    color: #4CAF50;
}
```

app.js

```
angular.module('appointmentApp', [])
    .controller('AppointmentController', function() {
        var vm = this;
        vm.appointment = {};

        vm.bookAppointment = function() {
            // Simulating booking logic
            vm.successMessage = 'Appointment booked successfully!';
        };
    });
```