SAPTHAGIRI NPS
UNIVERSITY

UNMATCHED EXCELLENCE, UNLIMITED POTENTIAL

Department of Computer Science & Engineering

“WEB TECHNOLOGY LAB”

24BECSE302

LAB MANUAL

(As per the prescribed Syllabus)

Prepared by

Web Technology Course Team

About The University

At Sapthagiri NPS University, we take pride in being a beacon of academic brilliance, seamlessly blending innovation, creativity with enjoyment. Our primary objective is to cultivate a dynamic learning environment that goes beyond traditional education, fostering creativity, and preparing students for real-world success. We educate, empower and enrich academic knowledge and inculcate value education to students. Amidst our state-ofthe-art campus, explore modern classrooms and cutting-edge laboratories that set the stage for holistic learning experience. From lively cultural fests that showcase diverse talents to a spectrum of programs that bring out the best in our students, we create a lively atmosphere where learning becomes an enjoyable journey. Join us on a transformative journey where education meets excitement, knowledge transforms into innovation, creativity and success becomes a way of life. At Sapthagiri NPS University, we are not just shaping leaders; we are crafting an unforgettable experience enriched by cultural celebrations and student-driven programs that showcase their exceptional talents and nourish their ambition of carving a life of their own dreams.

Institute Vision

Offer a transformative impact on society through unique learning experience in engineering technology, Medicine, Applied sciences, Business studies, Management studies and other areas of scholarship to the stakeholders to an unparalleled educational journey to serve the world and betterment of mankind.

To establish Sapthagiri NPS university as a leading institution fostering academic excellence and wisdom for success in a dynamic world.

Institute Mission

To provide a student Centric-learning environment focused on deep disciplinary knowledge, problem solving, leadership, and communication, interpersonal skills through innovative pedagogy and education reforms.

- To generate outstanding leaders in the field of health sciences and to provide optimum human patient-centered health care of the highest quality.
- To create and sustain a community of lifelong learners in an environment that emphasizes literacy, critical and innovative thinking, humanity, scientific inquiry and to promote patriotism and moral values.
- To impact in a transformative way regionally, nationally and globally to face the economic, social and health related challenges for nation building.
- To accomplish quality assurance, enhancement and sustenance in academics and research for a fair and social justice by providing equal opportunity.

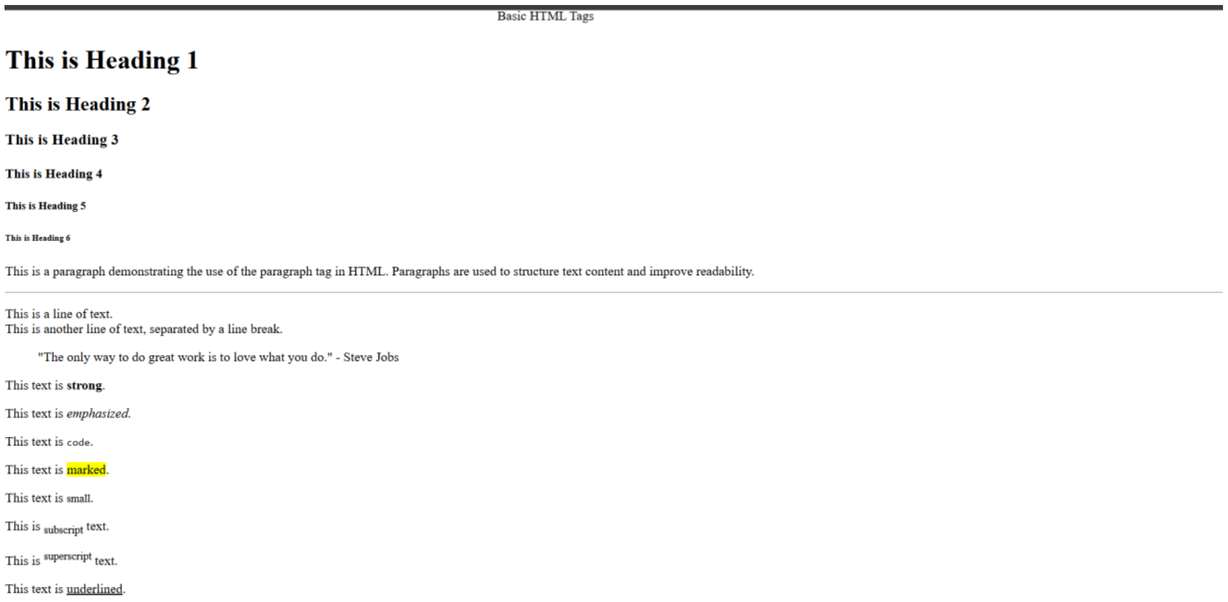
SR. No	**List of Programs**
HTML & CSS:	
1	Develop the HTMLpage named as ”MyFirstwebpage.html”. Add the following with relevant content. a. set the title of the webpage as “MyFirstwebpage” b. moving text=” Basic HTML Tags” c. Different heading tags (h1 to h6) d. Paragraph e. Horizontal line, Line Break, Block Quote **f.** Different logical styles
2	Design and develop a hospital details using html having the following information a. Title of the webpage as Name of Hospital b. Name of the hospital with address as header moving from right to left with hospital logo. c. Write description of hospital which is to be aligned properly d. Create a table to display list of specializations along with doctor names.
3	Design and develop webpage describing INDIA with its important information. The following information must be incorporated in the webpage a. Use appropriate CSS style (internal, external and inline to describe state of India. b. Use Grid model to display population of a state and Ratio of male and female of that state
4	Design and develop online food service webpage with following information a. Create a form to enter first name, last name email id and phone number of the customer. b. Create dropdown list to select food item along with its price c. Use text box to enter quantity of items order d. Create a bill of the customer e. Use media queries to change dimension of webpage to that of mobile and tablet
JavaScript & DOM Manipulation:	
5	Design and develop coffee order webpage with following information a. Create coffee order form with tables having heading Productname , Quantity and Price b. Customer must be able to select product from the dropdown list and price of the product, customer should enter the quantity of a given product c. Write JavaScript to compute total cost by clicking on OK button and display it to customer before submitting the order to the server for processing d. On over the total cost field, it must be disabled so that it is protected for further change
6	Design and develop hospital registration webpage with validation of phone number. The web page should have following information

	a. Create a form to enter first name, last name and phone number of the patient. b. Create radio button to select Gender of patient c. Create checkbox to create list of tests needed. d. Create dropdown for available doctors e. Create list of specializations available and also display table with doctors information heading the department f. The form should provide option to upload patients image with suitable instructions and suitable buttons to upload image Write JavaScript code to validate the phone number entered by user.
7	Design and develop a webpage demonstrating Key Press event in JavaScript with following information a. Title of web page is Keyboard Event example b. Display on web page header as "Press Any Key" c. The web page must display the key press in lowercase as well as in uppercase
8	Design and develop a webpage to create ID card for Nursery Childrens with following with following information a. Create a form to enter details of students such as Name ,parent Name and Age b. Upload the student photo on webpage c. Create Radio button to select Gender d. Create welcome message with student name. If the student is Male change background color to blue and for female change background color of welcome message to Pink e. Create a button Hide me, once you click ,hide the students photo and change the text message on button as show me. f. On clicking on show me button the student photo must be visible
9	Design and develop a simple dashboard /welcome for the tourist company and incorporate following information a. Weather prediction to be displayed on the webpage and top news headlines of the day b. Based on weather prediction display appropriate messages like " Sunny Day. keep a Hat Handy, "Rainy Day ,keep umbrella handy " etc c. Use Weather and News API's
React Application Development	
10	Write React code to create simple calculator.
11	Write React code to design to do list
12	Design a Grocery delivery application in React for online departmental store assuming that grocery items are huge in numbers. The application should list all items in the webpage along with their quantities and prices. Users must be able to sign up and purchase groceries. Use JSON to store product information

1. Develop the HTMLpage named as ”MyFirstwebpage.html”. Add the following with relevant content.

- **a. set the title of the webpage as “MyFirstwebpage”**
- **b. moving text=” Basic HTML Tags”**
- **c. Different heading tags (h1 to h6)**
- **d. Paragraph**
- **e. Horizontal line, Line Break, Block Quote, Different logical styles**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MyFirstwebpage</title>
</head>
<body>
    <!-- Moving text -->
    <marquee>Basic HTML Tags</marquee>
    <!-- Different heading tags (h1 to h6) -->
    <h1>This is Heading 1</h1>
    <h2>This is Heading 2</h2>
    <h3>This is Heading 3</h3>
    <h4>This is Heading 4</h4>
    <h5>This is Heading 5</h5>
    <h6>This is Heading 6</h6>
    <!-- Paragraph -->
    <p>This is a paragraph demonstrating the use of the paragraph tag in
HTML. Paragraphs are used to structure text content and improve
readability.</p>
    <!-- Horizontal line -->
    <hr>
    <!-- Line Break -->
    <p>This is a line of text.<br>This is another line of text, separated by
a line break.</p>
    <!-- Block Quote -->
    <blockquote>
        "The only way to do great work is to love what you do." - Steve Jobs
    </blockquote>
    <!-- Different logical styles -->
    <p>This text is <strong>strong</strong>.</p>
    <p>This text is <em>emphasized</em>.</p>
    <p>This text is <code>code</code>.</p>
    <p>This text is <mark>marked</mark>.</p>
    <p>This text is <small>small</small>.</p>
    <p>This is <sub>subscript</sub> text.</p>
    <p>This is <sup>superscript</sup> text.</p>
    <p>This text is <u>underlined</u>.</p>
    <p>This text is <s>strikethrough</s>.</p>
    <p>This is <abbr title="World Health Organization">WHO</abbr>.</p>
    <p>Press <kbd>Ctrl</kbd> + <kbd>S</kbd> to save.</p>
    <p><var>x</var> + <var>y</var> = 10</p>
  <p>Output: <samp>Hello, World!</samp></p>
</body>
</html>
```



Basic HTML Tags

This is Heading 1

This is Heading 2

This is Heading 3

This is Heading 4

This is Heading 5

This is Heading 6

This is a paragraph demonstrating the use of the paragraph tag in HTML. Paragraphs are used to structure text content and improve readability.

This is a line of text.
This is another line of text, separated by a line break.

> "The only way to do great work is to love what you do." - Steve Jobs

This text is **strong**.

This text is *emphasized*.

This text is `code`.

This text is marked.

This text is small.

This is $_{\text{subscript}}$ text.

This is $^{\text{superscript}}$ text.

This text is underlined.

2.Design and develop a hospital details using html having the following information

a. Title of the webpage as Name of Hospital

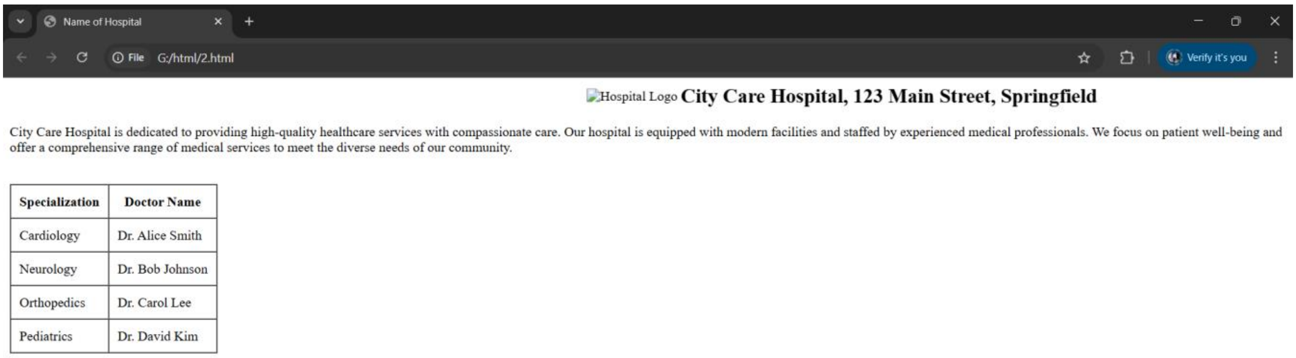
b. Name of the hospital with address as header moving from right to left with hospital logo.

c. Write description of hospital which is to be aligned properly

Create a table to display list of specializations along with doctor names.

```
<!DOCTYPE html>
<html>
<head>
  <title>Name of Hospital</title>
</head>
<body>
  <!-- Header with hospital name, address, and moving text -->
  <marquee behavior="scroll" direction="left" scrollamount="6">
    <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/e/e2/Hospital_icon.svg/120px-
Hospital_icon.svg.png" alt="Hospital Logo" height="40" style="vertical-align:middle;">
    <strong style="font-size: 24px; vertical-align:middle;">City Care Hospital, 123 Main Street,
Springfield</strong>
  </marquee>
  <br>
  <!-- Description -->
  <p>
    City Care Hospital is dedicated to providing high-quality healthcare services with compassionate
care.
    Our hospital is equipped with modern facilities and staffed by experienced medical professionals.
    We focus on patient well-being and offer a comprehensive range of medical services to meet the
diverse needs of our community.
  </p>
  <br>
```

```
    <!-- Table of specializations and doctors -->
    <table border="1" cellpadding="10" cellspacing="0">
        <tr>
            <th>Specialization</th>
            <th>Doctor Name</th>
        </tr>
        <tr>
            <td>Cardiology</td>
            <td>Dr. Alice Smith</td>
        </tr>
        <tr>
            <td>Neurology</td>
            <td>Dr. Bob Johnson</td>
        </tr>
        <tr>
            <td>Orthopedics</td>
            <td>Dr. Carol Lee</td>
        </tr>
        <tr>
            <td>Pediatrics</td>
            <td>Dr. David Kim</td>
        </tr>
    </table>
</body>
</html>
```



Hospital Logo **City Care Hospital, 123 Main Street, Springfield**

City Care Hospital is dedicated to providing high-quality healthcare services with compassionate care. Our hospital is equipped with modern facilities and staffed by experienced medical professionals. We focus on patient well-being and offer a comprehensive range of medical services to meet the diverse needs of our community.

Specialization	Doctor Name
Cardiology	Dr. Alice Smith
Neurology	Dr. Bob Johnson
Orthopedics	Dr. Carol Lee
Pediatrics	Dr. David Kim

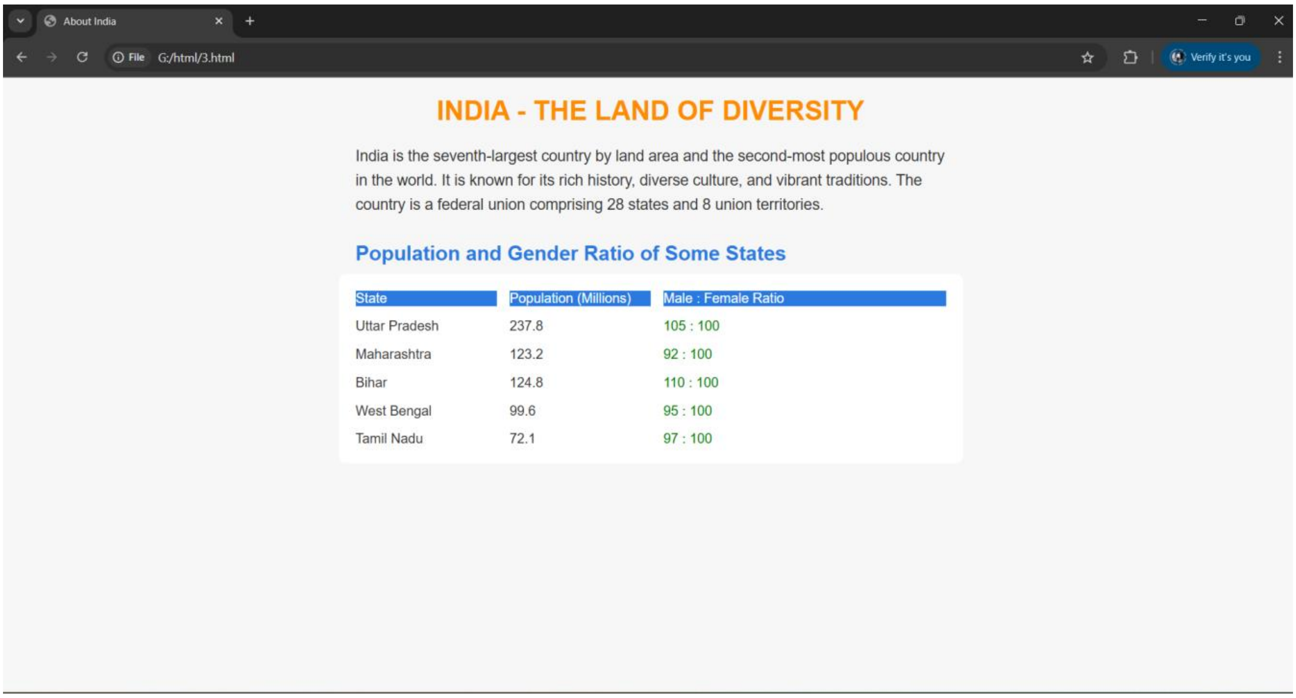
3. Design and develop webpage describing INDIA with its important information. The following information must be incorporated in the webpage

a.Use appropriate CSS style (internal, external and inline to describe state of India.

b.Use Grid model to display population of a state and Ratio of male and female of that state

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>About India</title>
  <style>
    /* Internal CSS */
    body {
      font-family: Arial, sans-serif;
      background-color: #f7f7f7;
      margin: 20px;
      color: #333;
    }
    h1 {
      text-align: center;
      text-transform: uppercase;
      color: darkorange;
    }
    p {
      font-size: 18px;
      line-height: 1.6;
      max-width: 700px;
      margin: 0 auto 30px auto;
    }
    h2 {
      max-width: 700px;
      margin: 0 auto 10px auto;
      color: #2a7ae2;
```

```
        }
        .state-grid {
            display: grid;
            grid-template-columns: 1fr 1fr 2fr;
            gap: 15px;
            background-color: #ffffff;
            padding: 20px;
            border-radius: 8px;
            max-width: 700px;
            margin: 0 auto;
        }
        .header {
            background-color: #2a7ae2;
            color: white;
        }
    </style>
</head>
<body>
    <h1>India - The Land of Diversity</h1>
    <p>
        India is the seventh-largest country by land area and the second-most populous country in the world.
        It is known for its rich history, diverse culture, and vibrant traditions. The country is a federal union comprising 28 states and 8 union territories.
    </p>
    <h2>Population and Gender Ratio of Some States</h2>
    <div class="state-grid">
        <div class="header">State</div>
        <div class="header">Population (Millions)</div>
        <div class="header">Male : Female Ratio</div>

        <div>Uttar Pradesh</div>
        <div>237.8</div>
        <div style="color: green;">105 : 100</div> <!-- inline style for color -->

        <div>Maharashtra</div>
        <div>123.2</div>
        <div style="color: green;">92 : 100</div>

        <div>Bihar</div>
        <div>124.8</div>
        <div style="color: green;">110 : 100</div>

        <div>West Bengal</div>
        <div>99.6</div>
        <div style="color: green;">95 : 100</div>

        <div>Tamil Nadu</div>
        <div>72.1</div>
        <div style="color: green;">97 : 100</div>
    </div>
</body>
</html>
```



INDIA - THE LAND OF DIVERSITY

India is the seventh-largest country by land area and the second-most populous country in the world. It is known for its rich history, diverse culture, and vibrant traditions. The country is a federal union comprising 28 states and 8 union territories.

Population and Gender Ratio of Some States

State	Population (Millions)	Male : Female Ratio
Uttar Pradesh	237.8	105 : 100
Maharashtra	123.2	92 : 100
Bihar	124.8	110 : 100
West Bengal	99.6	95 : 100
Tamil Nadu	72.1	97 : 100

4. Design and develop online food service webpage with following information

a.Create a form to enter first name, last name email id and phone number of the customer.

b.Create dropdown list to select food item along with its price

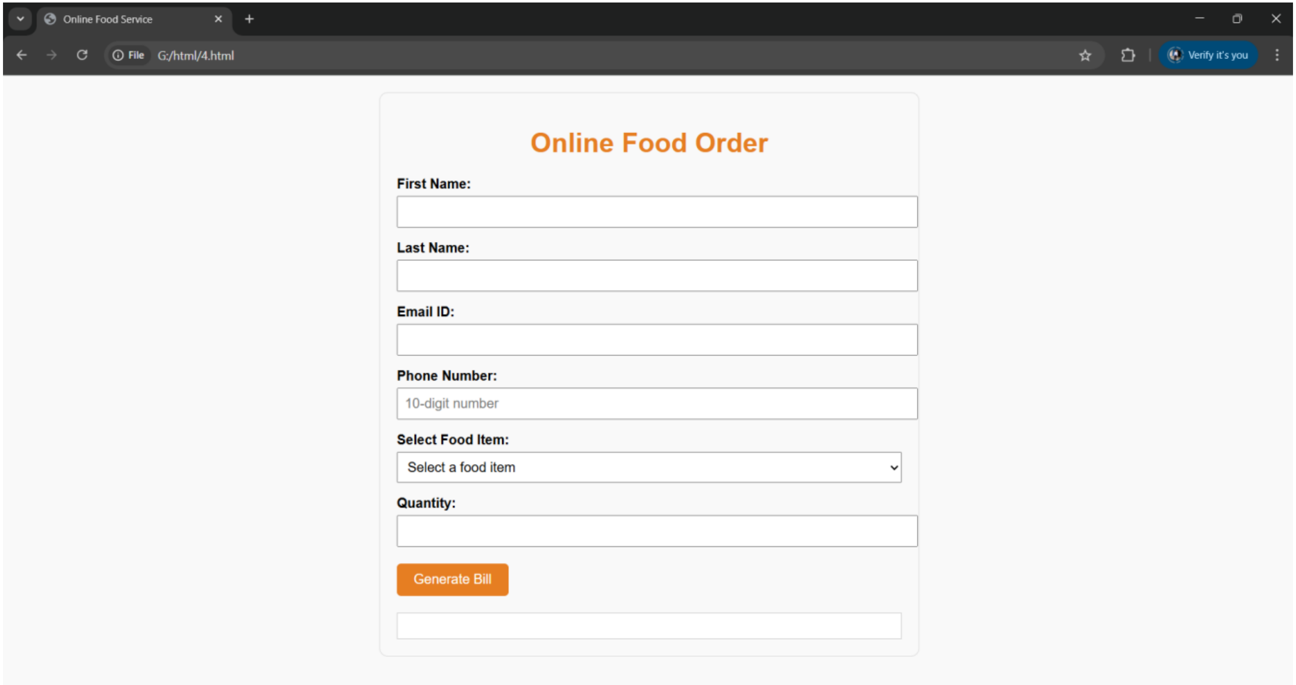
c.Use text box to enter quantity of items order

d.Create a bill of the customer

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Online Food Service</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      max-width: 600px;
      margin: 20px auto;
      padding: 20px;
      background-color: #f9f9f9;
      border-radius: 10px;
      border: 1px solid #ddd;
    }
    h1 {
      text-align: center;
      color: #e67e22;
    }
    form label {
      display: block;
      margin-top: 15px;
      font-weight: bold;
    }
    input, select {
      width: 100%;
      padding: 8px;
      margin-top: 5px;
      font-size: 16px;
    }
    button {
      margin-top: 20px;
      padding: 10px 20px;
      background-color: #e67e22;
      color: white;
      font-size: 16px;
      border: none;
      cursor: pointer;
      border-radius: 5px;
    }
    button:hover {
      background-color: #d35400;
    }
    #bill {
      margin-top: 20px;
      padding: 15px;
      background-color: #fff;
      border: 1px solid #ccc;
      font-family: monospace;
      white-space: pre-wrap;
    }
  </style>
</head>
<body>
  <h1>Online Food Order</h1>
  <form id="foodForm">
    <label for="fname">First Name:</label>
    <input type="text" id="fname" required>
    <label for="lname">Last Name:</label>
    <input type="text" id="lname" required>
    <label for="email">Email ID:</label>
    <input type="email" id="email" required>
    <label for="phone">Phone Number:</label>
    <input type="text" id="phone" pattern="[0-9]{10}" placeholder="10-digit number" required>

    <label for="food">Select Food Item:</label>
    <select id="food" required>
      <option value="" disabled selected>Select a food item</option>
      <option value="Pizza" data-price="12">Pizza - $12</option>
      <option value="Burger" data-price="8">Burger - $8</option>
      <option value="Pasta" data-price="10">Pasta - $10</option>
      <option value="Salad" data-price="6">Salad - $6</option>
      <option value="Sandwich" data-price="5">Sandwich - $5</option>
    </select>
    <label for="quantity">Quantity:</label>
    <input type="text" id="quantity" required>
    <button type="submit">Generate Bill</button>
  </form>
  <div id="bill"></div>
  <script>
    document.getElementById("foodForm").addEventListener("submit", function(event) {
      event.preventDefault();
      const fname = document.getElementById("fname").value.trim();
      const lname = document.getElementById("lname").value.trim();
      const email = document.getElementById("email").value.trim();
      const phone = document.getElementById("phone").value.trim();
      const foodSelect = document.getElementById("food");
      const foodItem = foodSelect.value;
      const foodPrice = parseFloat(foodSelect.options[foodSelect.selectedIndex].getAttribute("data-
price"));
```

```
      const quantity = parseInt(document.getElementById("quantity").value);
      if (!quantity || quantity <= 0) {
        alert("Please enter a valid quantity.");
        return;
      }

      const total = foodPrice * quantity;
      const bill = `
Customer Name : ${fname} ${lname}
Email         : ${email}
Phone         : ${phone}
Food Item     : ${foodItem}
Price per Item: $${foodPrice.toFixed(2)}
Quantity      : ${quantity}
-----------------------------
Total Bill    : $${total.toFixed(2)}
Thank you for ordering with us!
      `;      document.getElementById("bill").textContent = bill;
    });
  </script>
</body>
</html>
```

Output:



Online Food Order

First Name:

Last Name:

Email ID:

Phone Number:

10-digit number

Select Food Item:

Select a food item

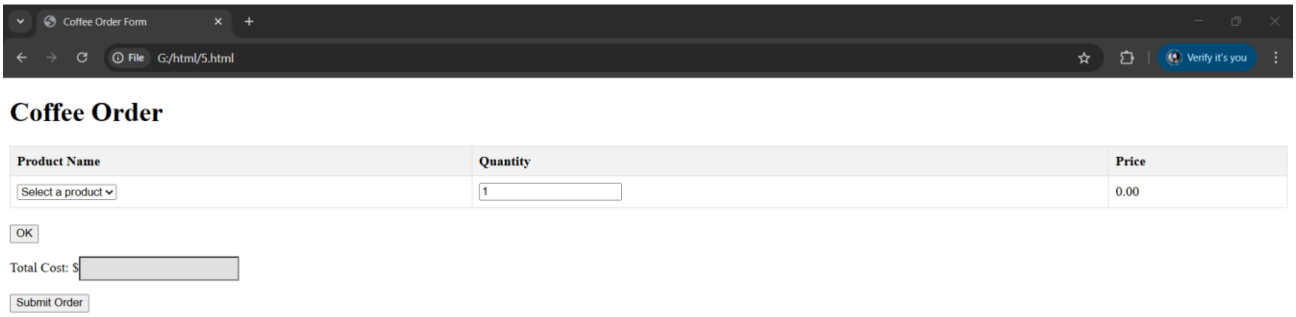
Quantity:

Generate Bill

5. Design and develop coffee order webpage with following information
a.Create coffee order form with tables having heading Productname , Quantity and Price
b.Customer must be able to select product from the dropdown list and price of the product, customer should enter the quantity of a given product
c.Write JavaScript to compute total cost by clicking on OK button and display it to customer before submitting the order to the server for processing
d.On over the total cost field, it must be disabled so that it is protected for further change

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Coffee Order Form</title>
    <style>
        table {
            width: 100%;
            border-collapse: collapse;
            margin-bottom: 20px;
        }
        th, td {
            border: 1px solid #ddd;
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #f2f2f2;
        }
        #totalCost {
            font-weight: bold;
            background-color: #e0e0e0;
            padding: 5px;
        }
    </style>
</head>
<body>
    <h1>Coffee Order</h1>
    <form id="coffeeOrderForm">
        <table>
            <thead>
                <tr>
                    <th>Product Name</th>
                    <th>Quantity</th>
                    <th>Price</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>
                        <select id="productSelect" onchange="updatePrice()">
                            <option value="">Select a product</option>
                            <option value="Espresso" data-price="2.50">Espresso</option>
                            <option value="Latte" data-price="3.75">Latte</option>
                            <option value="Cappuccino" data-price="3.50">Cappuccino</option>
                            <option value="Americano" data-price="3.00">Americano</option>
                        </select>
                    </td>
                    <td><input type="number" id="quantity" min="1" value="1"></td>
                    <td><span id="productPrice">0.00</span></td>
                </tr>
            </tbody>
        </table>
        <button type="button" onclick="calculateTotal()">OK</button>
        <p>Total Cost: $<input type="text" id="totalCost" readonly></p>
        <button type="submit">Submit Order</button>
    </form>

    <script>
        const products = {
            "Espresso": 2.50,
            "Latte": 3.75,
            "Cappuccino": 3.50,
            "Americano": 3.00
        };

        function updatePrice() {
            const productSelect = document.getElementById('productSelect');
            const productPriceSpan = document.getElementById('productPrice');
            const selectedOption = productSelect.options[productSelect.selectedIndex];
            const price = selectedOption.dataset.price || '0.00';
            productPriceSpan.textContent = parseFloat(price).toFixed(2);
        }

        function calculateTotal() {
            const productSelect = document.getElementById('productSelect');
            const quantityInput = document.getElementById('quantity');
            const totalCostInput = document.getElementById('totalCost');

            const selectedOption = productSelect.options[productSelect.selectedIndex];
            const price = parseFloat(selectedOption.dataset.price || '0');
            const quantity = parseInt(quantityInput.value);

            if (selectedOption.value && quantity > 0) {
                const total = price * quantity;
                totalCostInput.value = total.toFixed(2);
            } else {
                totalCostInput.value = '0.00';
            }
        }

        // Disable total cost field on mouseover
        document.getElementById('totalCost').addEventListener('mouseover', function() {
            this.disabled = true;
        });

        // Re-enable total cost field on mouseout (optional, for user convenience)
```

```
        document.getElementById('totalCost').addEventListener('mouseout', function() {
            this.disabled = false;
        });
    </script>
</body>
</html>
```



Coffee Order

Product Name	Quantity	Price
Select a product	1	0.00

Total Cost: $

6.Design and develop hospital registration webpage with validation of phone number. The web page should have following information

**a.	Create a form to enter first name, last name and phone number of the patient.**

**b.	Create radio button to select Gender of patient**

**c.	Create checkbox to create list of tests needed.**

**d.	Create dropdown for available doctors**

**e.	Create list of specializations available and also display table with doctors information heading the department**

**f.	The form should provide option to upload patients image with suitable instructions and suitable buttons to upload image**

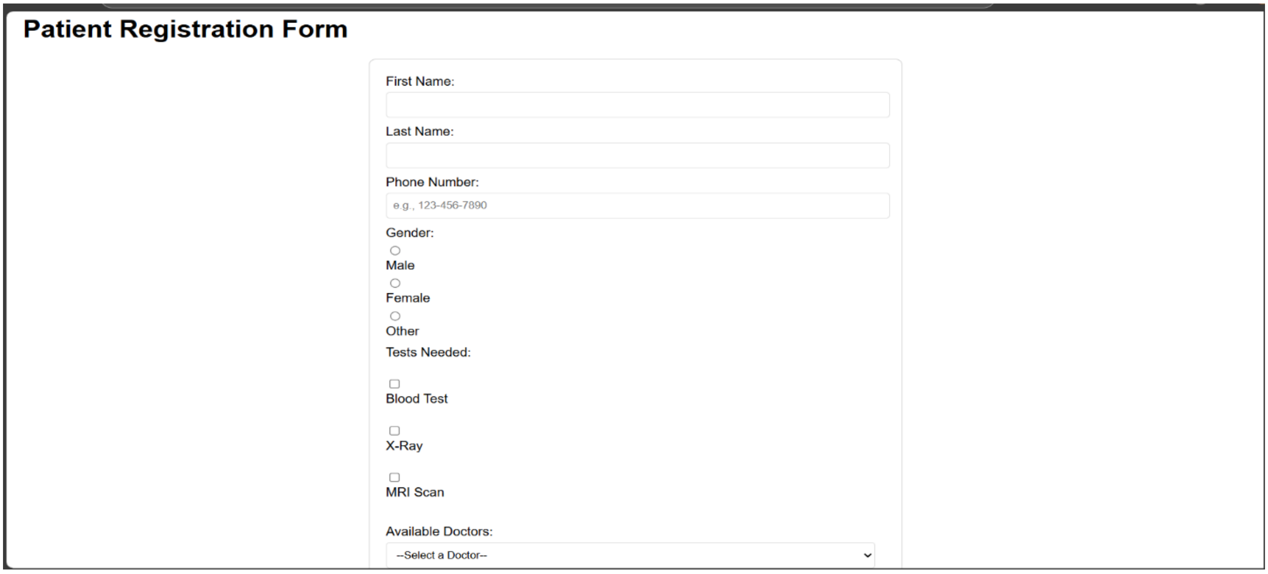
Write JavaScript code to validate the phone number entered by user.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Hospital Patient Registration</title>
   <style>
      body { font-family: Arial, sans-serif; margin: 20px; }
      form { max-width: 600px; margin: auto; padding: 20px; border: 1px solid #ccc; border-radius: 8px; }
      label { display: block; margin-bottom: 5px; }
      input[type="text"], input[type="tel"], select, textarea { width: calc(100% - 12px); padding: 8px; margin-
bottom: 10px; border: 1px solid #ddd; border-radius: 4px; }
      .radio-group, .checkbox-group { margin-bottom: 10px; }
      .error { color: red; font-size: 0.9em; }
      table { width: 100%; border-collapse: collapse; margin-top: 20px; }
      th, td { border: 1px solid #ddd; padding: 8px; text-align: left; }
      th { background-color: #f2f2f2; }
   </style>
</head>
<body>
   <h1>Patient Registration Form</h1>
   <form id="registrationForm">
      <label for="firstName">First Name:</label>
      <input type="text" id="firstName" name="firstName" required>
      <label for="lastName">Last Name:</label>
      <input type="text" id="lastName" name="lastName" required>
      <label for="phoneNumber">Phone Number:</label>
      <input type="tel" id="phoneNumber" name="phoneNumber" placeholder="e.g., 123-456-7890"
required>
      <span id="phoneError" class="error"></span>
      <div class="radio-group">
         <label>Gender:</label>
         <input type="radio" id="male" name="gender" value="male" required>
```

```
        <label for="male">Male</label>
        <input type="radio" id="female" name="gender" value="female">
        <label for="female">Female</label>
        <input type="radio" id="other" name="gender" value="other">
        <label for="other">Other</label>
    </div>
    <div class="checkbox-group">
        <label>Tests Needed:</label><br>
        <input type="checkbox" id="bloodTest" name="tests" value="bloodTest">
        <label for="bloodTest">Blood Test</label><br>
        <input type="checkbox" id="xray" name="tests" value="xray">
        <label for="xray">X-Ray</label><br>
        <input type="checkbox" id="mri" name="tests" value="mri">
        <label for="mri">MRI Scan</label><br>
    </div>
    <label for="doctor">Available Doctors:</label>
    <select id="doctor" name="doctor" required>
        <option value="">--Select a Doctor--</option>
        <option value="drSmith">Dr. John Smith (Cardiology)</option>
        <option value="drJones">Dr. Sarah Jones (Pediatrics)</option>
        <option value="drWilliams">Dr. David Williams (Orthopedics)</option>
    </select>
    <h2>Specializations and Department Heads</h2>
    <p>Below is a list of available specializations and the doctors heading those departments:</p>
    <table>
        <thead>
            <tr>
                <th>Specialization</th>
                <th>Department Head</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Cardiology</td>
                <td>Dr. Emily White</td>
            </tr>
            <tr>
                <td>Pediatrics</td>
                <td>Dr. Robert Green</td>
            </tr>
            <tr>
                <td>Orthopedics</td>
                <td>Dr. Lisa Brown</td>
            </tr>
        </tbody>
    </table>
    <label for="patientImage">Upload Patient Image:</label>
    <input type="file" id="patientImage" name="patientImage" accept="image/*">
    <p>Accepted formats: JPG, PNG. Max file size: 5MB.</p>
    <button type="button" onclick="uploadImage()">Upload Image</button>
    <br><br>
    <button type="submit">Register</button>
</form>
<script>
```

```
        document.getElementById('registrationForm').addEventListener('submit', function(event) {
            if (!validatePhoneNumber()) {
                event.preventDefault(); // Prevent form submission if validation fails
            } else {
                alert('Registration successful!'); // Placeholder for actual submission
            }
        });
        function validatePhoneNumber() {
            const phoneNumberInput = document.getElementById('phoneNumber');
            const phoneError = document.getElementById('phoneError');
            const phoneNumber = phoneNumberInput.value.trim();
            const phonePattern = /^\(?([0-9]{3})\)?[-. ]?([0-9]{3})[-. ]?([0-9]{4})$/;
            if (phoneNumber === "") {
                phoneError.textContent = "Phone number is required.";
                return false;
            } else if (!phonePattern.test(phoneNumber)) {
                phoneError.textContent = "Please enter a valid phone number (e.g., 123-456-7890).";
                return false;
            } else {
                phoneError.textContent = ""; // Clear error message
                return true;
            }
        }
        function uploadImage() {
            const imageInput = document.getElementById('patientImage');
            if (imageInput.files.length > 0) {
                const file = imageInput.files[0];
                // In a real application, you would send this file to a server for storage.
                alert(`Image "${file.name}" selected for upload. (Upload functionality not implemented in this
demo)`);
            } else {
                alert("Please select an image to upload.");
            }
        }
    </script>
</body>
</html>
```

Output:



Patient Registration Form

First Name:

Last Name:

Phone Number:

e.g., 123-456-7890

Gender:

○ Male

○ Female

○ Other

Tests Needed:

□ Blood Test

□ X-Ray

□ MRI Scan

Available Doctors:

--Select a Doctor--

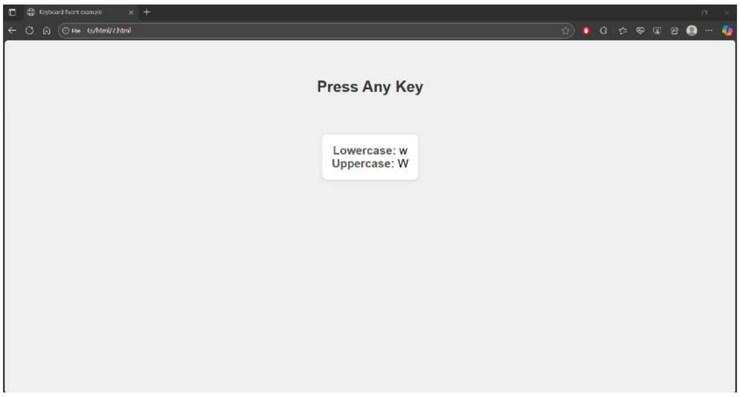
7.Design and develop a webpage demonstrating Key Press event in JavaScript with following information

a.Title of web page is Keyboard Event example

b.Display on web page header as “Press Any Key”

c.The web page must display the key press in lowercase as well as in uppercase

```
!DOCTYPE html>
<html lang="en">
<head>
    <title>Keyboard Event example</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      padding: 50px;
      background-color: #f0f0f0;
    }
    h1 {
      color:#333;
    }
    .output {
      margin-top: 60px;
      font-size: 1.5rem;
      background-color:white;
      padding: 20px;
      border-radius: 10px;
      display: inline-block;
      box-shadow: 0 2px 6px rgba(0,0,0,0.1);
    }
    .label {
      font-weight: bold;
      color: #555;
    }
  </style>
</head>
<body>
  <h1>Press Any Key</h1>
  <div class="output">
    <div><span class="label">Lowercase:</span> <span id="lowercase">-</span></div>
    <div><span class="label">Uppercase:</span> <span id="uppercase">-</span></div>
  </div>
  <script>
    document.addEventListener('keydown', function(event) {
      const key = event.key;
      // Display lowercase and uppercase
      document.getElementById('lowercase').textContent = key.toLowerCase();
      document.getElementById('uppercase').textContent = key.toUpperCase();
    });
  </script>
</body>
</html>
```



Press Any Key

Lowercase: w
Uppercase: W

8.Design and develop a webpage to create ID card for Nursery Childrens with following with following information

a. Create a form to enter details of students such as Name ,parent Name and Age

b. Upload the student photo on webpage

c. Create Radio button to select Gender

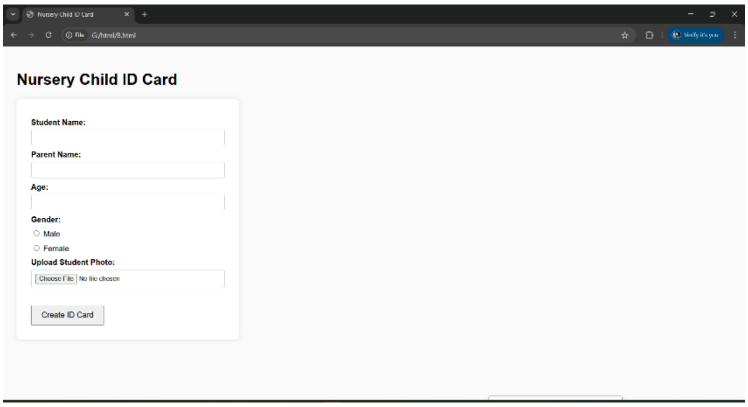
d. Create welcome message with student name. If the student is Male change background color to blue and for female change background color of welcome message to Pink

e. Create a button Hide me, once you click ,hide the students photo and change the text message on button as show me.

f. On clicking on show me button the student photo must be visible

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Nursery Child ID Card</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      padding: 20px;
      background-color: #fafafa;
    }
    form {
      max-width: 400px;
      margin-bottom: 80px;
      background: #fff;
      padding: 30px;
      border-radius: 8px;
      box-shadow: 0 0 8px rgba(0,0,0,0.1);
    }
    label {
      display: block;
      margin-top: 10px;
      font-weight: bold;
    }
    input[type="text"], input[type="number"], input[type="file"] {
      width: 100%;
      padding: 8px;
      margin-top: 5px;
      box-sizing: border-box;
      border-radius: 4px;
      border: 1px solid #ccc;
    }
    .gender-group {
      margin-top: 10px;
    }
    .gender-group label {
      font-weight: normal;
      margin-right: 15px;
    }
    #welcomeMessage {
      font-size: 1.5rem;
      margin-top: 20px;
      padding: 10px 20px;
      border-radius: 8px;
      color: white;
      display: inline-block;
    }
    #photoContainer {
      margin-top: 20px;
      text-align: center;
    }
    #studentPhoto {
      max-width: 200px;
      border-radius: 10px;
      border: 2px solid #ccc;
    }
    #togglePhotoBtn {
      margin-top: 10px;
      padding: 8px 15px;
      font-size: 1rem;
      cursor: pointer;
      border: none;
      border-radius: 5px;
      background-color: #007bff;
      color: white;
      transition: background-color 0.3s ease;
    }
    #togglePhotoBtn:hover {
      background-color: #0056b3;
    }
  </style>
</head>
<body>
  <h1>Nursery Child ID Card</h1>
  <form id="idCardForm">
    <label for="studentName">Student Name:</label>
    <input type="text" id="studentName" name="studentName" />
    <label for="parentName">Parent Name:</label>
    <input type="text" id="parentName" name="parentName"  />
    <label for="age">Age:</label>
    <input type="number" id="age" name="age" min="1" max="10"  />
    <label>Gender:</label>
    <div class="gender-group">
      <label><input type="radio" name="gender" value="male"  /> Male</label>
      <label><input type="radio" name="gender" value="female" /> Female</label>
    </div>
    <label for="photoUpload">Upload Student Photo:</label>
    <input type="file" id="photoUpload" accept="image/*"  />
    <button type="submit" style="margin-top:35px; padding:10px 20px; font-size:1rem;">Create ID Card</button>
  </form>
```

```
<div id="welcomeMessage" style="display:none;"></div>
<div id="photoContainer" style="display:none;">
  <img id="studentPhoto" src="" alt="Student Photo" />
  <br />
  <button id="togglePhotoBtn">Hide me</button>
</div>
<script>
  const form = document.getElementById('idCardForm');
  const welcomeMessage = document.getElementById('welcomeMessage');
  const photoContainer = document.getElementById('photoContainer');
  const studentPhoto = document.getElementById('studentPhoto');
  const togglePhotoBtn = document.getElementById('togglePhotoBtn');
  const photoUpload = document.getElementById('photoUpload');
  let photoVisible = true;
  form.addEventListener('submit', function(event) {
    event.preventDefault();
    // Get values
    const studentName = form.studentName.value.trim();
    const gender = form.gender.value;
    // Show welcome message with color depending on gender
    welcomeMessage.textContent = `Welcome, ${studentName}!`;
    if (gender === 'male') {
      welcomeMessage.style.backgroundColor = 'blue';
    } else if (gender === 'female') {
      welcomeMessage.style.backgroundColor = 'pink';
    }
    welcomeMessage.style.display = 'inline-block';
    // Handle photo upload preview
    const file = photoUpload.files[0];
    if (file) {
      const reader = new FileReader();
      reader.onload = function(e) {
        studentPhoto.src = e.target.result;
        photoContainer.style.display = 'block';
        studentPhoto.style.display = 'inline-block';
        togglePhotoBtn.textContent = 'Hide me';
        photoVisible = true;
      };
      reader.readAsDataURL(file);
    }
  });
  // Toggle photo visibility on button click
  togglePhotoBtn.addEventListener('click', function() {
    if (photoVisible) {
      studentPhoto.style.display = 'none';
      togglePhotoBtn.textContent = 'Show me';
      photoVisible = false;
    } else {
      studentPhoto.style.display = 'inline-block';
      togglePhotoBtn.textContent = 'Hide me';
      photoVisible = true;
    }
  });
</script> </body></html>
```



Nursery Child ID Card

Student Name:

Parent Name:

Age:

Gender:

○ Male

○ Female

Upload Student Photo:

Choose File No file chosen

Create ID Card

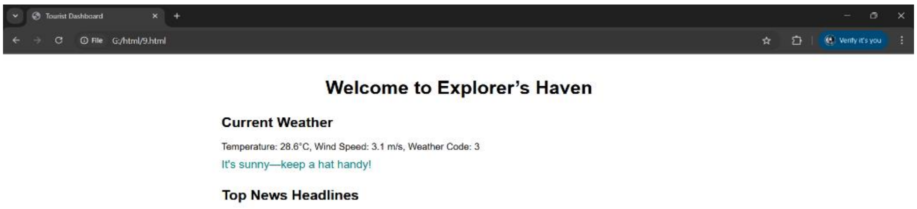
9.Design and develop a simple dashboard /welcome for the tourist company and incorporate following information

a.Weather prediction to be displayed on the webpage and top news headlines of the day

b.Based on weather prediction display appropriate messages like " Sunny Day. keep a Hat Handy, "Rainy Day ,keep umbrella handy " etc

c.Use Weather and News API's

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Tourist Dashboard</title>
  <style>
    body { font-family: Arial, sans-serif; max-width: 800px; margin: auto; padding: 20px; }
    h1 { text-align: center; }
    .section { margin-top: 30px; }
    .advice { font-size: 1.2em; color: teal; margin-top: 10px; }
    ul { list-style-type: disc; margin-left: 20px; }
  </style>
</head>
<body>
<h1>Welcome to Explorer's Haven</h1>
<div class="section" id="weather-section">
  <h2>Current Weather</h2>
  <div id="weather-info">Loading weather...</div>
  <div id="advice" class="advice"></div>
</div>
<div class="section" id="news-section">
  <h2>Top News Headlines</h2>
  <ul id="news-list"><li>Loading news...</li></ul>
</div>
<script>
  async function fetchWeather() {
    // Coordinates for New Delhi, India (change as needed)
    const url = 'https://api.open-meteo.com/v1/forecast?latitude=28.6&longitude=77.2&current_weather=true';
    const res = await fetch(url);
    const data = await res.json();
    const w = data.current_weather;
    const info = `Temperature: ${w.temperature}°C, Wind Speed: ${w.windspeed} m/s, Weather Code: ${w.weathercode}`;
    document.getElementById('weather-info').innerText = info;
    const code = w.weathercode;
    const advice = (code === 0 || code === 1 || code === 2 || code === 3)
      ? "It's sunny—keep a hat handy!"
      : (code >= 51 && code <= 67) || code >= 95
        ? "Rainy or stormy—bring an umbrella!"
        : "Weather is moderate—plan accordingly.";
    document.getElementById('advice').innerText = advice;
  }
  async function fetchNews() {
    const key = 'YOUR_NEWSDATA_API_KEY';
    const url = `https://newsdata.io/api/1/latest?apikey=${key}&country=us`;
    const res = await fetch(url);
    const data = await res.json();
    const list = document.getElementById('news-list');
    list.innerHTML = ";
    (data.results || []).slice(0, 5).forEach(item => {
      const li = document.createElement('li');
      li.innerHTML = `<a href="${item.link}" target="_blank">${item.title}</a>`;
      list.appendChild(li);
    });
  }
  fetchWeather();
  fetchNews();
</script>
</body>
</html>
```



Welcome to Explorer's Haven

Current Weather

Temperature: 28.6°C, Wind Speed: 3.1 m/s, Weather Code: 3

It's sunny—keep a hat handy!

Top News Headlines

10.Write React code to create simple calculator.

File Structure

```
calculator-app/

|── public/

|   └── index.html

|

|── src/

|   ├── App.js

|   ├── App.css

|   ├── Calculator.js

|   └── index.js

|

|── package.json
```

public/index.html

(make sure this exists, otherwise React won't run)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>React Calculator</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

src/App.js

```
import React from 'react';
import Calculator from './Calculator';
import './App.css'; // for styling

function App() {
  return (
    <div className="App">
      <Calculator />
    </div>
  );
}

export default App;
```

src/Calculator.js

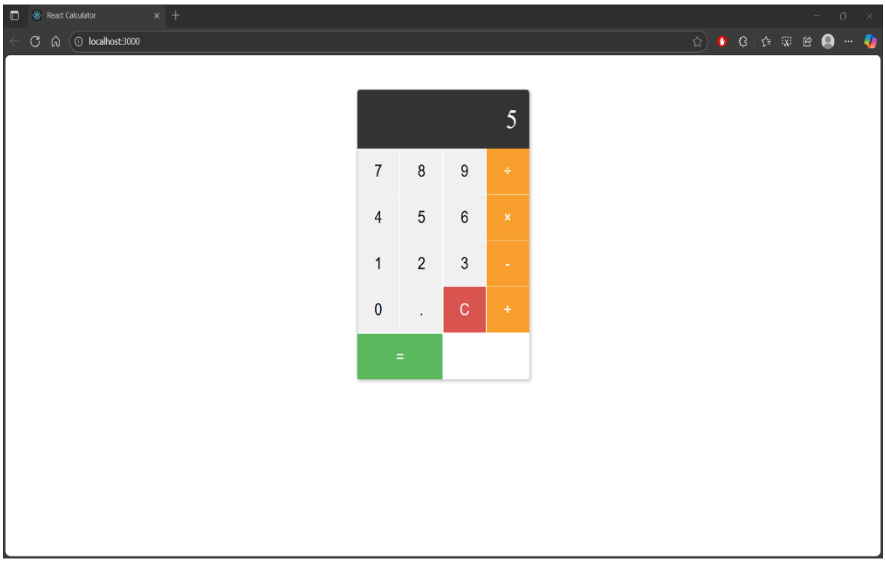
(you already wrote this, keeping exactly the same)

```
import React, { useState } from 'react';

function Calculator() {
  const [displayValue, setDisplayValue] = useState('0');
  const [firstOperand, setFirstOperand] = useState(null);
  const [operator, setOperator] = useState(null);
  const [waitingForSecondOperand, setWaitingForSecondOperand] = useState(false);

  const inputDigit = (digit) => {
    if (waitingForSecondOperand) {
      setDisplayValue(String(digit));
      setWaitingForSecondOperand(false);
    } else {
      setDisplayValue(displayValue === '0' ? String(digit) : displayValue + digit);
    }
  };

  const inputDecimal = () => {
    if (waitingForSecondOperand) {
      setDisplayValue('0.');
      setWaitingForSecondOperand(false);
      return;
    }
    if (!displayValue.includes('.')) {
      setDisplayValue(displayValue + '.');
    }
  };

  const clearDisplay = () => {
    setDisplayValue('0');
    setFirstOperand(null);
    setOperator(null);
    setWaitingForSecondOperand(false);
  };

  const performOperation = (nextOperator) => {
    const inputValue = parseFloat(displayValue);

    if (firstOperand === null && !isNaN(inputValue)) {
      setFirstOperand(inputValue);
    } else if (operator) {
      const result = calculate(firstOperand, inputValue, operator);
      setDisplayValue(String(result));
      setFirstOperand(result);
    }

    setWaitingForSecondOperand(true);
    setOperator(nextOperator);
  };

  const calculate = (firstNum, secondNum, op) => {
    if (op === '+') return firstNum + secondNum;
    if (op === '-') return firstNum - secondNum;
    if (op === '*') return firstNum * secondNum;
    if (op === '/') return firstNum / secondNum;
    return secondNum; // For "="
  };

  return (
    <div className="calculator">
      <div className="calculator-display">{displayValue}</div>
      <div className="calculator-buttons">
        <button onClick={() => inputDigit(7)}>7</button>
        <button onClick={() => inputDigit(8)}>8</button>
        <button onClick={() => inputDigit(9)}>9</button>
        <button className="operator" onClick={() => performOperation('/')}>&divide;</button>

        <button onClick={() => inputDigit(4)}>4</button>
        <button onClick={() => inputDigit(5)}>5</button>
        <button onClick={() => inputDigit(6)}>6</button>
        <button className="operator" onClick={() => performOperation('*')}>&times;</button>

        <button onClick={() => inputDigit(1)}>1</button>
        <button onClick={() => inputDigit(2)}>2</button>
        <button onClick={() => inputDigit(3)}>3</button>
        <button className="operator" onClick={() => performOperation('-')}>-</button>

        <button onClick={() => inputDigit(0)}>0</button>
        <button onClick={inputDecimal}>.</button>
        <button className="clear" onClick={clearDisplay}>C</button>
        <button className="operator" onClick={() => performOperation('+')}>+</button>

        <button className="equal-sign operator" onClick={() => performOperation('=')}>=</button>
      </div>
    </div>
  );
}

export default Calculator;
```

src/App.css

(your styling is great, keeping as-is)

```
.calculator {
  width: 300px;
  margin: 50px auto;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2);
}
.calculator-display {
  background-color: #333;
  color: #fff;
  font-size: 2.5em;
  padding: 20px;
  text-align: right;
  border-top-left-radius: 5px;
  border-top-right-radius: 5px;
}
.calculator-buttons {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 1px;
}
.calculator-buttons button {
  background-color: #f0f0f0;
  border: none;
  padding: 20px;
  font-size: 1.5em;
  cursor: pointer;
  transition: background-color 0.2s;
}
.calculator-buttons button:hover {
  background-color: #e0e0e0;
}
.calculator-buttons .operator {
  background-color: #f79e2c;
  color: #fff;
}
.calculator-buttons .operator:hover {
  background-color: #e68a1c;
}
.calculator-buttons .clear {
  background-color: #d9534f;
  color: #fff;
}
.calculator-buttons .clear:hover {
  background-color: #c9302c;
}
.calculator-buttons .equal-sign {
  grid-column: span 2;
  background-color: #5cb85c;
  color: #fff;
}
.calculator-buttons .equal-sign:hover {
  background-color: #4cae4c;
}
```



11.Write React code to design to do list

npx create-react-app@latest todolist

cd todolist

Features

- Add new tasks
- Mark tasks as complete / incomplete
- Delete tasks
- Tasks saved in **localStorage** (so refreshing won't lose them)
- Clean UI with CSS

☞ The **important files** for you to work with are:

- **src/App.js → React To-Do List logic**
- **src/App.css → Styling**
- **public/index.html → Base HTML**

src/App.js

```
import React, { useState, useEffect } from "react";
import "./App.css";

function App() {
  const [task, setTask] = useState("");
  const [tasks, setTasks] = useState([]);

  // Load from localStorage
  useEffect(() => {
    const savedTasks = JSON.parse(localStorage.getItem("tasks"));
    if (savedTasks) {
      setTasks(savedTasks);
    }
  }, []);

  // Save to localStorage
  useEffect(() => {
    localStorage.setItem("tasks", JSON.stringify(tasks));
  }, [tasks]);

  const addTask = () => {
    if (task.trim() === "") return;
    setTasks([...tasks, { text: task, completed: false }]);
    setTask("");
  };

  const toggleTask = (index) => {
    const newTasks = [...tasks];
    newTasks[index].completed = !newTasks[index].completed;
    setTasks(newTasks);
  };

  const deleteTask = (index) => {
    const newTasks = tasks.filter((_, i) => i !== index);
    setTasks(newTasks);
  };

  return (
    <div className="app-container">
      <h1>📝 To-Do List</h1>
      <div className="input-container">
        <input
          type="text"
          placeholder="Enter a task..."
          value={task}
          onChange={(e) => setTask(e.target.value)}
        />
        <button onClick={addTask}>Add</button>
      </div>
      <ul className="task-list">
        {tasks.map((t, index) => (
          <li key={index} className={t.completed ? "completed" : ""}>
            <span onClick={() => toggleTask(index)}>{t.text}</span>
            <button onClick={() => deleteTask(index)}>❌</button>
          </li>
        ))}
      </ul>
    </div>
  );
}

export default App;
```

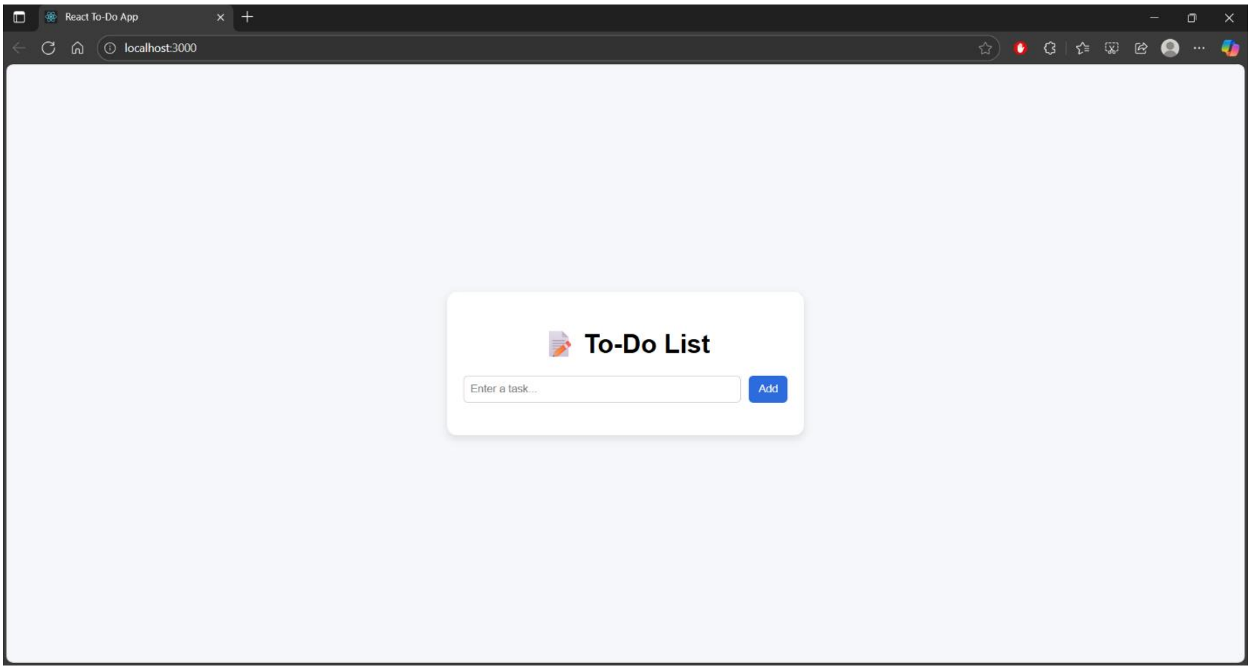
src/App.css

```
body {
  margin: 0;
  font-family: Arial, sans-serif;
  background: #f5f7fa;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.app-container {
  width: 400px;
  background: #fff;
  padding: 20px;
  border-radius: 12px;
  box-shadow: 0 4px 10px rgba(0,0,0,0.1);
  text-align: center;
}

h1 {
  margin-bottom: 20px;
}

.input-container {
  display: flex;
  gap: 10px;
  margin-bottom: 20px;
}

input {
  flex: 1;
  padding: 8px;
  border-radius: 6px;
  border: 1px solid #ccc;
}

button {
  padding: 8px 12px;
  border: none;
  border-radius: 6px;
  background: #2d6cdf;
  color: white;
  cursor: pointer;
  transition: background 0.2s;
}

button:hover {
  background: #1c4cad;
}

.task-list {
  list-style: none;
  padding: 0;
  margin: 0;
}

.task-list li {
  display: flex;
  justify-content: space-between;
  align-items: center;
  background: #f1f3f6;
  padding: 10px;
  margin-bottom: 10px;
  border-radius: 8px;
}

.task-list li.completed span {
  text-decoration: line-through;
  color: gray;
  font-style: italic;
}

.task-list li span {
  flex: 1;
  cursor: pointer;
  text-align: left;
}
```

public/index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <title>React To-Do App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```



12.Design a Grocery delivery application in React for online departmental store assuming that grocery items are huge in numbers. The application should list all items in the webpage along with their quantities and prices. Users must be able to sign up and purchase groceries. Use JSON to store product information.

1. Product Listing – Grocery items displayed from a JSON file with name, quantity, and price.
2. User Signup & Login – Simple local storage–based authentication (no backend).
3. Shopping Cart – Users can add/remove items before checkout.
4. Purchase Flow – Simulated purchase action.
5. Responsive UI – Grid layout for products.

🗁 Project Structure

```
grocery-app/
├── public/
│   └── products.json      # Store grocery items here
├── src/
│   ├── components/
│   │   ├── ProductList.js
│   │   ├── Cart.js
│   │   ├── Signup.js
│   │   └── Login.js
│   ├── App.js
│   ├── App.css
│   └── index.js
├── package.json
```

products.json (inside public/)

```
[
  { "id": 1, "name": "Rice", "quantity": "1kg", "price": 50 },
  { "id": 2, "name": "Wheat Flour", "quantity": "1kg", "price": 40 },
  { "id": 3, "name": "Sugar", "quantity": "1kg", "price": 45 },
  { "id": 4, "name": "Milk", "quantity": "1L", "price": 25 },
  { "id": 5, "name": "Eggs", "quantity": "12 pcs", "price": 60 }
]
```

App.js

```
import React, { useState, useEffect } from "react";
import ProductList from "./components/ProductList";
import Cart from "./components/Cart";
import Signup from "./components/Signup";
import Login from "./components/Login";
import "./App.css";

function App() {
  const [products, setProducts] = useState([]);
  const [cart, setCart] = useState([]);
  const [user, setUser] = useState(null);
  const [view, setView] = useState("login"); // "login", "signup", "shop"

  useEffect(() => {
    fetch("/products.json")
      .then((res) => res.json())
      .then((data) => setProducts(data));
  }, []);
  const addToCart = (product) => {
    setCart([...cart, product]);
  };

  const removeFromCart = (id) => {
    setCart(cart.filter((item, index) => index !== id));
  };

  const handleSignup = (username, password) => {
    localStorage.setItem("user", JSON.stringify({ username, password }));
    alert("Signup successful! Please login.");
    setView("login");
  };

  const handleLogin = (username, password) => {
    const stored = JSON.parse(localStorage.getItem("user"));
    if (stored && stored.username === username && stored.password === password) {
      setUser(username);
      setView("shop");
    } else {
      alert("Invalid credentials!");
    }
  };

  const handleLogout = () => {
    setUser(null);
    setView("login");
  };

  return (
    <div className="App">
      <header>
        <h1>🛒 Grocery Delivery Store</h1>
        {user && <button onClick={handleLogout}>Logout</button>}
      </header>

      {view === "signup" && <Signup onSignup={handleSignup} setView={setView} />}
      {view === "login" && <Login onLogin={handleLogin} setView={setView} />}
      {view === "shop" && (
        <div className="container">
          <ProductList products={products} addToCart={addToCart} />
          <Cart cart={cart} removeFromCart={removeFromCart} />
        </div>
      )}
    </div>
  );
}

export default App;
```

components/ProductList.js

```
import React from "react";
```

```
function ProductList({ products, addToCart }) {
  return (
    <div className="product-grid">
      {products.map((p) => (
        <div className="product-card" key={p.id}>
          <h3>{p.name}</h3>
          <p>Quantity: {p.quantity}</p>
          <p>Price: ₹{p.price}</p>
          <button onClick={() => addToCart(p)}>Add to Cart</button>
        </div>
      ))}
    </div>
  );
}

export default ProductList;
```

components/Cart.js

```
import React from "react";

function Cart({ cart, removeFromCart }) {
  const total = cart.reduce((sum, item) => sum + item.price, 0);

  return (
    <div className="cart">
      <h2>🛍️ Cart</h2>
      {cart.length === 0 && <p>No items in cart.</p>}
      {cart.map((item, index) => (
        <div className="cart-item" key={index}>
          <span>{item.name} - ₹{item.price}</span>
          <button onClick={() => removeFromCart(index)}>Remove</button>
        </div>
      ))}
      <h3>Total: ₹{total}</h3>
      {cart.length > 0 && <button onClick={() => alert("Purchase Successful!")}>Checkout</button>}
    </div>
  );
}

export default Cart;
```

components/Signup.js

```
import React, { useState } from "react";

function Signup({ onSignup, setView }) {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");

  const handleSubmit = (e) => {
    e.preventDefault();
    onSignup(username, password);
  };

  return (
    <div className="auth-form">
      <h2>Signup</h2>
      <form onSubmit={handleSubmit}>
```

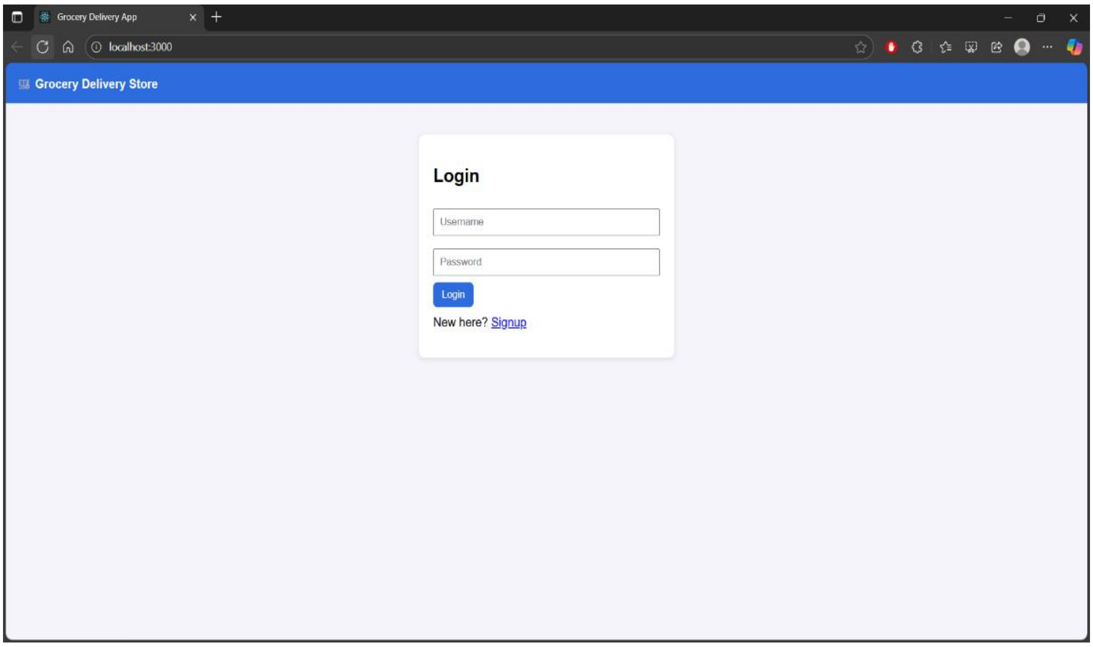
```
        <input type="text" placeholder="Username" value={username}
          onChange={(e) => setUsername(e.target.value)} required />
        <input type="password" placeholder="Password" value={password}
          onChange={(e) => setPassword(e.target.value)} required />
        <button type="submit">Signup</button>
      </form>
      <p>
        Already have an account?{" "}
        <span onClick={() => setView("login")} className="link">Login</span>
      </p>
    </div>
  );
}

export default Signup;
```

components/Login.js

```
import React, { useState } from "react";

function Login({ onLogin, setView }) {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");

  const handleSubmit = (e) => {
    e.preventDefault();
    onLogin(username, password);
  };

  return (
    <div className="auth-form">
      <h2>Login</h2>
      <form onSubmit={handleSubmit}>
        <input type="text" placeholder="Username" value={username}
          onChange={(e) => setUsername(e.target.value)} required />
        <input type="password" placeholder="Password" value={password}
          onChange={(e) => setPassword(e.target.value)} required />
        <button type="submit">Login</button>
      </form>
      <p>
        New here?{" "}
        <span onClick={() => setView("signup")} className="link">Signup</span>
      </p>
    </div>
  );
}

export default Login;
```

🎨 **App.css**

```
body {
  margin: 0;
  font-family: Arial, sans-serif;
  background: #f4f4f9;
}

header {
  background: #2d6cdf;
  color: white;
  padding: 15px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.container {
  display: flex;
  gap: 20px;
  padding: 20px;
}

.product-grid {
  flex: 2;
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
  gap: 15px;
}

.product-card {
  background: white;
  padding: 15px;
  border-radius: 8px;
  box-shadow: 0 2px 6px rgba(0,0,0,0.1);
}
button {
  background: #2d6cdf;
  color: white;
  border: none;
  padding: 8px 12px;
  border-radius: 6px;
  cursor: pointer;
}

button:hover {
  background: #174a9b;
}
.cart {
  flex: 1;
  background: white;
  padding: 15px;
  border-radius: 8px;
  box-shadow: 0 2px 6px rgba(0,0,0,0.1);
}
.auth-form {
  max-width: 300px;
  margin: 40px auto;
  background: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 6px rgba(0,0,0,0.1);
}
.auth-form input {
  width: 100%;
  margin: 8px 0;
  padding: 8px;
}
.link {
  color: blue;
  cursor: pointer;
}
```

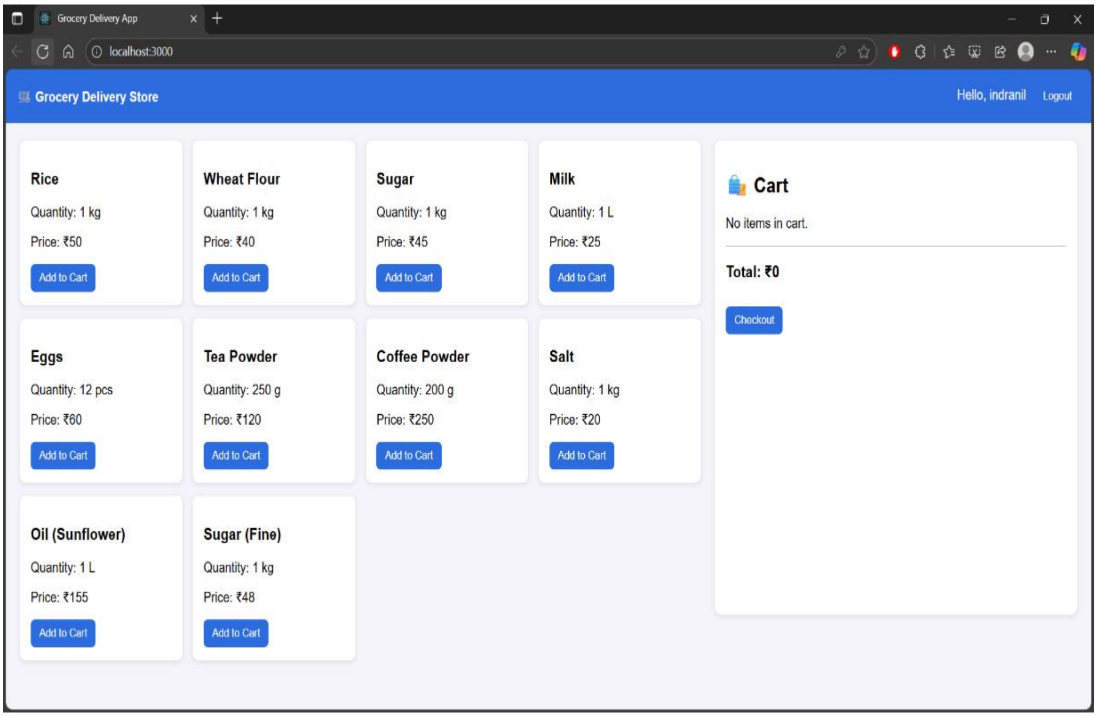


Login

Username

Password

New here? Signup



Rice
Quantity: 1 kg
Price: ₹50

Wheat Flour
Quantity: 1 kg
Price: ₹40

Sugar
Quantity: 1 kg
Price: ₹45

Milk
Quantity: 1 L
Price: ₹25

Eggs
Quantity: 12 pcs
Price: ₹60

Tea Powder
Quantity: 250 g
Price: ₹120

Coffee Powder
Quantity: 200 g
Price: ₹250

Salt
Quantity: 1 kg
Price: ₹20

Oil (Sunflower)
Quantity: 1 L
Price: ₹155

Sugar (Fine)
Quantity: 1 kg
Price: ₹48

Cart

No items in cart.

Total: ₹0