**Practical Assignment-2**

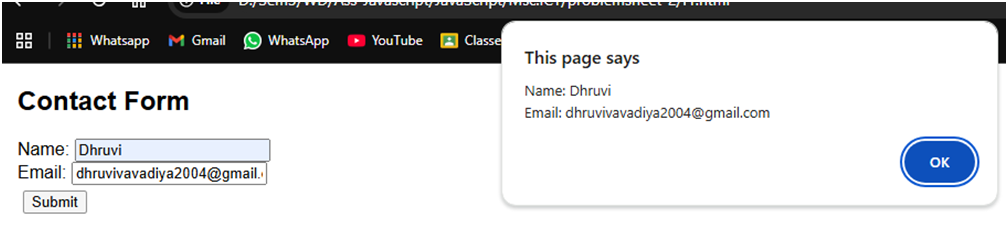
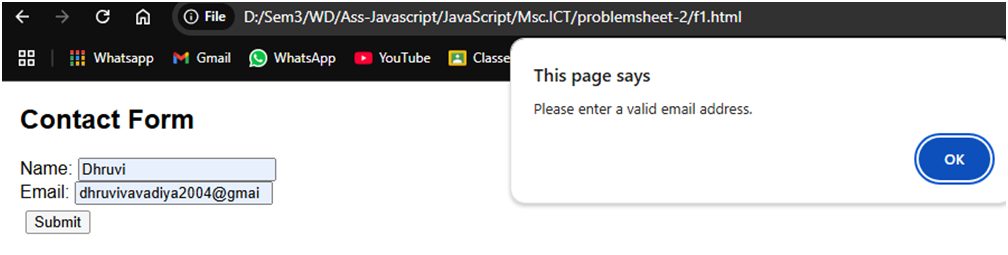
|  |  |
| --- | --- |
| **Name** | **Vavadiya Dhruvi Maheshbhai** |
| **Cource** | **Msc.ICT(sem-1)** |
| **Roll No :** | **73** |
| **Subject** | **101(Frontend Development using ReactJs)** |

**jQuery Practical Questions:**

**1. Create a simple HTML form with fields for name, email, and a submit button. Field name**

**and email should have relevant validation. Use jQuery to prevent the form from submitting**

**when the submit button is clicked and instead display an alert with the form data.**

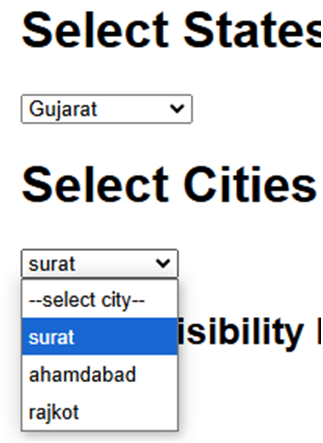


**2. Create an HTML page that includes two dropdown menus (combos): one for selecting states**

**and another for displaying cities. When the user selects a state from the primary combo, the**

**secondary combo (cities) should dynamically update to display cities relevant to the selected**

**state.**



**3. Write jQuery code to toggle the visibility of a div element when a button is clicked. The div**

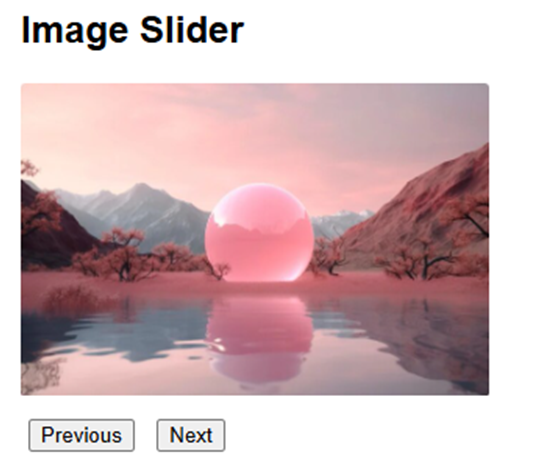
**should start as hidden, and clicking the button should alternate between showing and hiding**

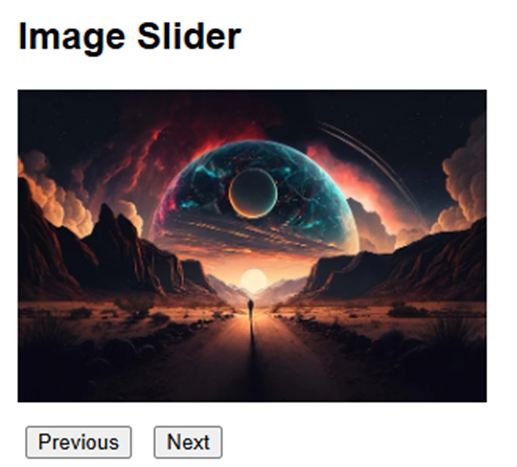
**it.**



**4. Create a basic image slider using jQuery. Users should be able to click &quot;Next&quot; and &quot;Previous&quot;**

**buttons to navigate between a series of images displayed on a webpage.**

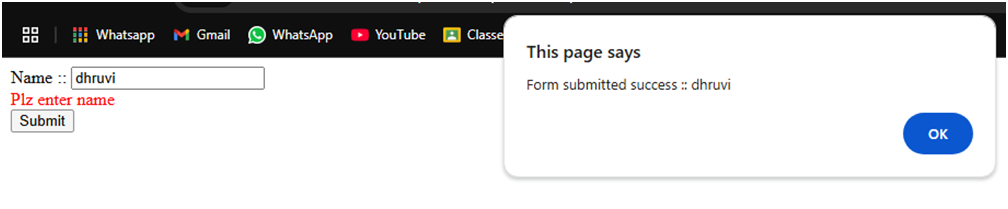




**5. Implement a simple client-side validation using jQuery. Check if a text input field is empty**

**when a form is submitted. If it&#39;s empty, display an error message next to the input field.**



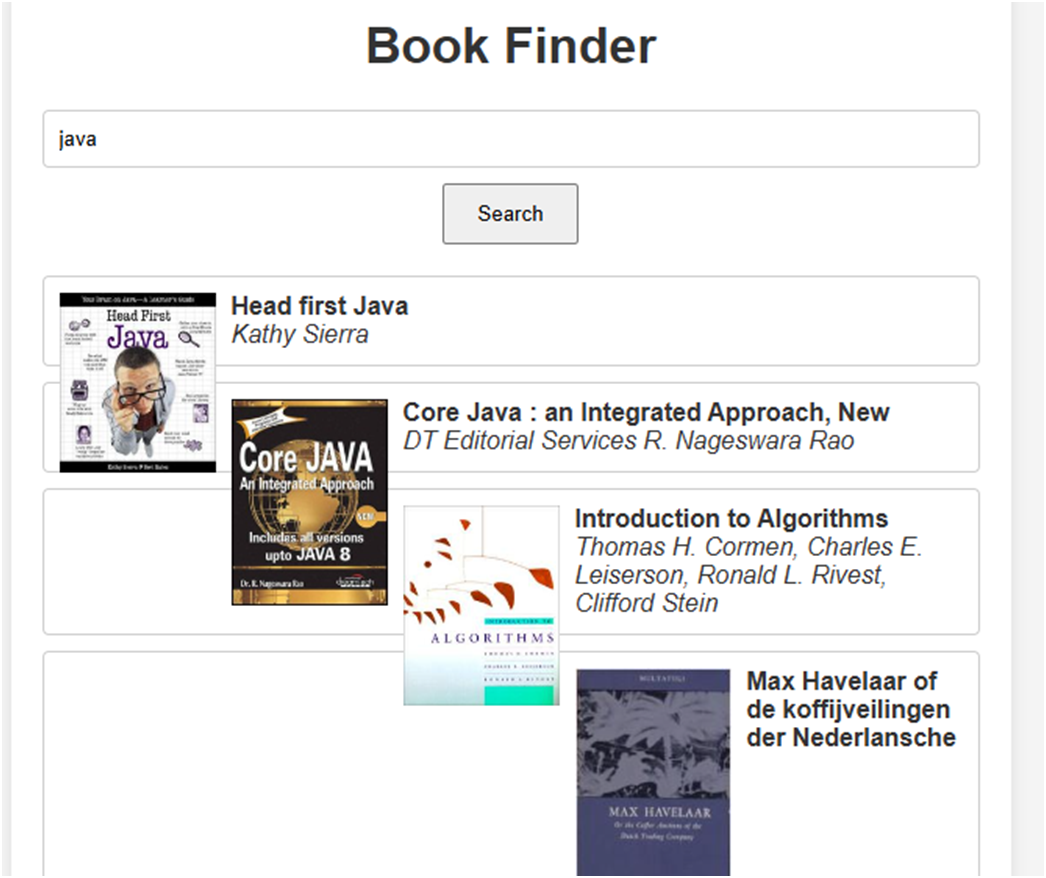


**AJAX Practical Questions:**

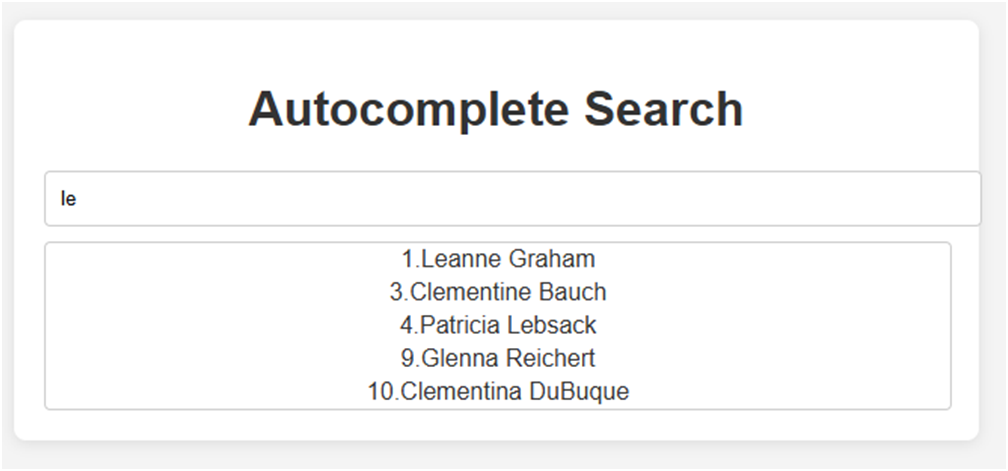
**6. Develop a webpage that retrieves and displays data from a public JSON API (e.g., a list of**

**books, movies, or weather information) using AJAX. Display this data in a visually appealing**

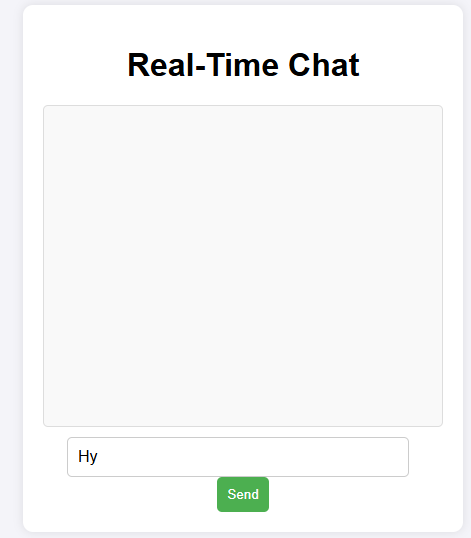
**manner on the page.**



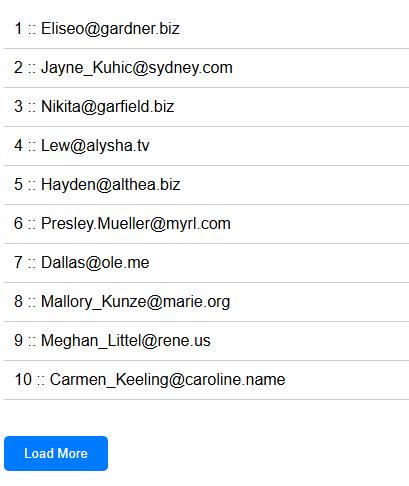
**7. Create an autocomplete search feature using AJAX and jQuery. When users start typing a query into a search input field, dynamically fetch and display matching results from an API as dropdown suggestions.**



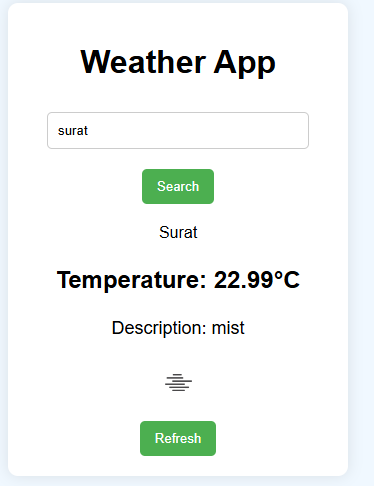
**8. Build a real-time chat application using AJAX to send and receive messages without refreshing the page. Messages should be displayed on the page as they are sent and received.**



**9. Design a &quot;Load More&quot; button that, when clicked, fetches additional content from an API and appends it to an existing list on the webpage. Implement pagination using AJAX.**



**10. Develop a weather app that allows users to enter a city name and retrieve weather data from a weather API using AJAX.  Display the temperature, weather description, and an icon representing the weather condition.**



* **CODE ::-**

**jQuery Practical Questions Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>First Question</title>

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

    <style>

        #toggleDiv {

            display: none;

            padding: 20px;

            border: 1px solid #ccc;

            margin-top: 10px;

        }

        body {

            font-family: Arial, sans-serif;

            margin: 20px;

        }

        #slider {

            width: 300px;

            height: 200px;

            overflow: hidden;

            position: relative;

            margin-bottom: 10px;

        }

        #slider img {

            width: 100%;

            height: auto;

            display: none;

            position: absolute;

            top: 0;

            left: 0;

        }

        #slider img.active {

            display: block;

        }

        button {

            margin: 5px;

        }

    </style>

</head>

<body>

    <div class="form-container">

        <h2>Contact Form</h2>

        <form id="contact-form">

            <label for="name">Name:</label>

            <input type="text" id="name" name="name">

            <br>

            <label for="email">Email:</label>

            <input type="email" id="email" name="email">

            <br>

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

        </form>

    </div>

    <h1>Select States</h1>

    <select name="state" id="state">

        <option value="">--select state--</option>

        <option value="Gujarat">Gujarat</option>

        <option value="Maharashtra">Maharashtra</option>

        <option value="Rajasthan">Rajasthan</option>

    </select>

    <h1>Select Cities</h1>

    <select name="city" id="city" disabled>

        <option value="">--select city--</option>

    </select>

    <h2>Toggle Visibility Example</h2>

    <button id="toggleButton">Toggle Div</button>

    <div id="toggleDiv">

        This is the content of the div that can be toggled.

    </div>

    <h2>Image Slider</h2>

    <div id="slider">

        <img src="https://gratisography.com/wp-content/uploads/2024/03/gratisography-funflower-800x525.jpg"  alt="Image 1">

        <img src="https://fps.cdnpk.net/images/home/subhome-ai.webp?w=649&h=649" alt="Image 2">

        <img src="https://cdn.pixabay.com/photo/2023/03/15/20/46/background-7855413\_640.jpg" alt="Image 3">

    </div>

    <button id="prevButton">Previous</button>

    <button id="nextButton">Next</button>

    <form action="" id="fomr">

        <label for="">Name ::</label>

        <input type="text" name="name" id="name" placeholder="Enter name"><br>

        <div id="error" style="display: none;">Plz enter name</div>

        <input type="submit" value="Submit" id="submit">

    </form>

    <div class="container">

        <h1>Book Finder</h1>

        <input type="text" id="search" placeholder="Search for books...">

        <button id="search-btn">Search</button>

        <div id="results" class="results"></div>

    </div>

    <script>

        $(document).ready(function () {

            $('#contact-form').on('submit', function (event) {

                event.preventDefault();

                const name = $('#name').val().trim();

                const email = $('#email').val().trim();

                alert(email);

                if (name === "") {

                    alert("Please enter your name.");

                    return;

                }

                const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

                if (!emailPattern.test(email)) {

                    alert("Please enter a valid email address.");

                    return;

                }

                alert(`Name: ${name}\nEmail: ${email}`);

            });

        });

        $(document).ready(function () {

            const cities = {

                Gujarat: ['surat', 'ahamdabad', 'rajkot'],

                Maharashtra: ['mumbai', 'pune', 'nagpur'],

                Rajasthan: ['jaipur', 'udaipur', 'jodhpur']

            }

            $("#state").on("change", function () {

                const seleState = $(this).val();

                const city = $("#city");

                city.empty().append(`<option value="">--select city--</option>`);

                if (seleState) {

                    city.prop("disabled", false);

                    cities[seleState].forEach(element => {

                        city.append(`<option value="${element}">${element}</option>`);

                    });

                } else {

                    city.prop("disabled", true);

                }

            });

        });

        $(document).ready(function () {

            $('#toggleButton').on('click', function () {

                $('#toggleDiv').toggle();

            });

        });

        $(document).ready(function() {

            let currentIndex = 0;

            const images = $('#slider img');

            const totalImages = images.length;

            images.eq(currentIndex).addClass('active');

            $('#nextButton').on('click', function() {

                images.eq(currentIndex).removeClass('active');

                currentIndex = (currentIndex + 1) % totalImages;

                images.eq(currentIndex).addClass('active');

            });

            $('#prevButton').on('click', function() {

                images.eq(currentIndex).removeClass('active');

                console.log("without count :: "+currentIndex);

                currentIndex = (currentIndex - 1 + totalImages) % totalImages;

                images.eq(currentIndex).addClass('active');

            });

        });

        document.addEventListener('contextmenu', event => event.preventDefault());

        $(document).ready(function(){

            $("#fomr").on("submit",function(e){

                e.preventDefault();

                const nm=$("#name").val().trim();

                console.log(nm);

                if(nm == ''){

                    alert("don't need to empty this field");

                    $("#error").css("display","block");

                    $("#error").css("color","red");

                }else{

                    alert("Form submitted success :: "+nm);

                }

            });

        });

        $(document).ready(function () {

    $('#search-btn').click(function () {

        const query = $('#search').val();

        if (query) {

            fetchBooks(query);

        } else {

            alert('Please enter a search term.');

        }

    });

    function fetchBooks(query) {

        const apiUrl = `https://openlibrary.org/search.json?q=${encodeURIComponent(query)}`;

        $.ajax({

            url: apiUrl,

            method: 'GET',

            success: function (data) {

                displayResults(data.docs);

            },

            error: function () {

                $('#results').html('<p>Error retrieving data.</p>');

            }

        });

    }

    function displayResults(books) {

        $('#results').empty();

        if (books.length > 0) {

            books.forEach(book => {

                const title = book.title || 'No title available';

                const author = book.author\_name ? book.author\_name.join(', ') : 'Unknown author';

                const cover = book.cover\_i ? `https://covers.openlibrary.org/b/id/${book.cover\_i}-M.jpg` : 'https://via.placeholder.com/100x150?text=No+Image';

                $('#results').append(`

                    <div class="book">

                        <img src="${cover}" alt="Cover">

                        <div>

                            <strong>${title}</strong><br>

                            <em>${author}</em>

                        </div>

                    </div>

                `);

            });

        } else {

            $('#results').html('<p>No books found.</p>');

        }

    }

});

    </script>

</body>

</html>

**AJAX Practical Questions Code:**

**6.**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Book Finder</title>

     <style>

        body {

    font-family: Arial, sans-serif;

    background-color: #f4f4f4;

    color: #333;

    margin: 0;

    padding: 0;

}

.container {

    width: 80%;

    max-width: 600px;

    margin: 50px auto;

    padding: 20px;

    background: white;

    border-radius: 8px;

    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

    text-align: center;

}

input {

    padding: 10px;

    width: calc(100% - 22px);

    border: 1px solid #ccc;

    border-radius: 4px;

    margin-bottom: 10px;

}

button {

    padding: 10px 20px;

    cursor: pointer;

}

.results {

    margin-top: 20px;

}

.book {

    border: 1px solid #ccc;

    border-radius: 4px;

    padding: 10px;

    margin: 10px 0;

    text-align: left;

}

.book img {

    max-width: 100px;

    margin-right: 10px;

    float: left;

}

     </style>

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

</head>

<body>

    <div class="container">

        <h1>Book Finder</h1>

        <input type="text" id="search" placeholder="Search for books...">

        <button id="search-btn">Search</button>

        <div id="results" class="results"></div>

    </div>

     <script>

        $(document).ready(function () {

    $('#search-btn').click(function () {

        const query = $('#search').val();

        if (query) {

            fetchBooks(query);

        } else {

            alert('Please enter a search term.');

        }

    });

    function fetchBooks(query) {

        const apiUrl = `https://openlibrary.org/search.json?q=${encodeURIComponent(query)}`;

        $.ajax({

            url: apiUrl,

            method: 'GET',

            success: function (data) {

                displayResults(data.docs);

            },

            error: function () {

                $('#results').html('<p>Error retrieving data.</p>');

            }

        });

    }

    function displayResults(books) {

        $('#results').empty();

        if (books.length > 0) {

            books.forEach(book => {

                const title = book.title || 'No title available';

                const author = book.author\_name ? book.author\_name.join(', ') : 'Unknown author';

                const cover = book.cover\_i ? `https://covers.openlibrary.org/b/id/${book.cover\_i}-M.jpg` : 'https://via.placeholder.com/100x150?text=No+Image';

                $('#results').append(`

                    <div class="book">

                        <img src="${cover}" alt="Cover">

                        <div>

                            <strong>${title}</strong><br>

                            <em>${author}</em>

                        </div>

                    </div>

                `);

            });

        } else {

            $('#results').html('<p>No books found.</p>');

        }

    }

});

     </script>

</body>

</html>

**7.**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Autocomplete Search</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f4f4;

            color: #333;

        }

        .container {

            width: 80%;

            max-width: 600px;

            margin: 50px auto;

            padding: 20px;

            background: white;

            border-radius: 8px;

            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

            text-align: center;

        }

        input {

            padding: 10px;

            width: 100%;

            border: 1px solid #ccc;

            border-radius: 4px;

            margin-bottom: 10px;

        }

        .suggestions {

            border: 1px solid #ccc;

            border-radius: 4px;

            background: white;

            z-index: 1000;

            width: 100%;

            max-height: 200px;

            overflow-y: auto;

            display: none;

        }

        .suggestion-item {

            padding: 2px;

            cursor: pointer;

        }

        .suggestion-item:hover {

            background-color: #e0e0e0;

        }

    </style>

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

</head>

<body>

    <div class="container">

        <h1>Autocomplete Search</h1>

        <input type="text" id="search" placeholder="Start typing..." value="le">

        <div id="suggestions" class="suggestions"></div>

    </div>

    <script>

        $(document).ready(function () {

            $('#search').on('input', function () {

                const query = $(this).val();

                if (query.length > 1) {

                    fetchSuggestions(query);

                } else {

                    $('#suggestions').empty().hide();

                }

            });

            function fetchSuggestions(query) {

                const apiUrl = `https://jsonplaceholder.typicode.com/users`;

                $.ajax({

                    url: apiUrl,

                    method: 'GET',

                    success: function (data) {

                        $('#suggestions').empty();

                        const filteredData = data.filter(user => user.name.toLowerCase().includes(query.toLowerCase()));

                        if (filteredData.length > 0) {

                            filteredData.forEach(user => {

                                $('#suggestions').append(`<div class="suggestion-item">${user.id}.${user.name}</div>`); // Use username for suggestions

                            });

                            $('#suggestions').show();

                        } else {

                            $('#suggestions').empty().hide();

                        }

                    },

                    error: function () {

                        $('#suggestions').empty().hide();

                    }

                });

            }

            $(document).on('click', '.suggestion-item', function () {

                $('#search').val($(this).text());

                $('#suggestions').empty().hide();

            });

            $(document).on('click', function (e) {

                if (!$(e.target).closest('.container').length) {

                    $('#suggestions').empty().hide();

                }

            });

        });

    </script>

</body>

</html>

**8.**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Real-Time Chat</title>

 <style>

body {

    font-family: Arial, sans-serif;

    background-color: #f4f4f9;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

    margin: 0;

  }

  .chat-container {

    background-color: white;

    padding: 20px;

    border-radius: 10px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    width: 400px;

    text-align: center;

  }

  .chat-box {

    height: 300px;

    overflow-y: auto;

    background-color: #f9f9f9;

    border: 1px solid #ddd;

    padding: 10px;

    margin-bottom: 10px;

    border-radius: 5px;

  }

  #message-input {

    padding: 10px;

    width: 80%;

    font-size: 16px;

    margin-right: 10px;

    border-radius: 5px;

    border: 1px solid #ccc;

  }

  #send-btn {

    padding: 10px;

    background-color: #4CAF50;

    color: white;

    border: none;

    border-radius: 5px;

    cursor: pointer;

  }

  #send-btn:hover {

    background-color: #45a049;

  }

  .message {

    padding: 5px;

    margin: 5px 0;

    background-color: #e0f7fa;

    border-radius: 5px;

  }

 </style>

</head>

<body>

  <div class="chat-container">

    <h1>Real-Time Chat</h1>

    <div id="chat-box" class="chat-box"></div>

    <input type="text" id="message-input" placeholder="Type a message..." />

    <button id="send-btn">Send</button>

  </div>

 <script>

document.addEventListener("DOMContentLoaded", () => {

    const chatBox = document.getElementById('chat-box');

    const messageInput = document.getElementById('message-input');

    const sendButton = document.getElementById('send-btn');

    function displayMessage(message) {

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

      messageElement.classList.add('message');

      messageElement.textContent = message;

      chatBox.appendChild(messageElement);

      chatBox.scrollTop = chatBox.scrollHeight;

    }

    function sendMessage() {

      const message = messageInput.value.trim();

      if (message) {

        const xhr = new XMLHttpRequest();

        xhr.open('POST', '/send', true);

        xhr.setRequestHeader('Content-Type', 'application/json');

        xhr.onload = function () {

          if (xhr.status === 200) {

            displayMessage(message);

            messageInput.value = '';

          }

        };

        xhr.send(JSON.stringify({ message: message }));

      }

    }

    sendButton.addEventListener('click', sendMessage);

    function loadMessages() {

      const xhr = new XMLHttpRequest();

      xhr.open('GET', '/messages', true);

      xhr.onload = function () {

        if (xhr.status === 200) {

          const messages = JSON.parse(xhr.responseText);

          chatBox.innerHTML = '';

          messages.forEach(message => {

            displayMessage(message.message);

          });

        }

      };

      xhr.send();

    }

    loadMessages();

    setInterval(loadMessages, 2000);

  });

 </script>

</body>

</html>

const express = require('express');

const bodyParser = require('body-parser');

const app = express();

const PORT = 3000;

let messages = [];

app.use(bodyParser.json());

app.use(express.static('public'));

app.get('/messages', (req, res) => {

  res.json(messages);

});

app.post('/send', (req, res) => {

  const message = req.body.message;

  if (message) {

    messages.push({ message: message, time: new Date() });

    res.status(200).json({ status: 'success', message });

  } else {

    res.status(400).json({ status: 'error', message: 'No message content' });

  }

});

app.listen(PORT, () => {

  console.log(`Server running at http://localhost:${PORT}`);

});

**9.**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Load More Example</title>

    <style>

        body {

            font-family: Arial, sans-serif;

        }

        #item-list {

            list-style-type: none;

            padding: 0;

        }

        #item-list li {

            padding: 10px;

            border-bottom: 1px solid #ccc;

        }

        #loadbtn {

            padding: 10px 20px;

            margin: 20px 0;

            cursor: pointer;

            background-color: #007bff;

            color: white;

            border: none;

            border-radius: 5px;

        }

        #loadbtn:hover {

            background-color: #0056b3;

        }

    </style>

</head>

<body>

    <div id="content">

        <ul id="item-list">

        </ul>

    </div>

    <button id="loadbtn">Load More</button>

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

    <script>

        $(document).ready(function () {

            let cPage=1;

            function fetchItems(cPage){

                $.ajax({

                    url: `https://jsonplaceholder.typicode.com/comments?\_limit=5&\_page=${cPage}`,

                    method :'GET',

                    dataType:'json',

                    success : function (data){

                        if(data.length > 0){

                            data.forEach(ele => {

                                $("#item-list").append(`<li>${ele.id} :: ${ele.email}</li>`);

                            });

                        }else{

                            $("#loadbtn").hide();

                        }

                    },

                    error : function (){

                        alert("FEtching data is not responding");

                    }

                })

            }

            fetchItems(cPage);

            $("#loadbtn").click(function(){

                cPage++;

                fetchItems(cPage);

            })

        });

    </script>

</body>

</html>

**10.**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Weather App</title>

 <style>

    /\* style.css \*/

body {

    font-family: Arial, sans-serif;

    background-color: #f0f8ff;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

    margin: 0;

}

.weather-container {

    text-align: center;

    background-color: #fff;

    padding: 20px;

    border-radius: 10px;

    box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);

    width: 300px;

}

#city-input {

    padding: 10px;

    width: 80%;

    margin: 10px 0;

    border: 1px solid #ccc;

    border-radius: 5px;

}

#search-btn, #refresh-btn {

    padding: 10px 15px;

    margin-top: 10px;

    background-color: #4CAF50;

    color: white;

    border: none;

    border-radius: 5px;

    cursor: pointer;

}

#search-btn:hover, #refresh-btn:hover {

    background-color: #45a049;

}

.weather-info {

    margin-top: 20px;

}

#temperature {

    font-size: 24px;

    font-weight: bold;

}

#description {

    font-size: 18px;

}

#weather-icon {

    width: 50px;

    height: 50px;

}

 </style>

</head>

<body>

    <div class="weather-container">

        <h1>Weather App</h1>

        <input type="text" id="city-input" placeholder="Enter city name" />

        <button id="search-btn">Search</button>

        <div id="weather-info" class="weather-info">

            <p id="city-name"></p>

            <p id="temperature"></p>

            <p id="description"></p>

            <img id="weather-icon" src="" alt="" />

        </div>

        <button id="refresh-btn">Refresh</button>

    </div>

    <script>

document.addEventListener("DOMContentLoaded", () => {

    const apiKey = '601698c3c85962be7ee2dbfb4bc01c4e';

    const cityInput = document.getElementById('city-input');

    const searchButton = document.getElementById('search-btn');

    const refreshButton = document.getElementById('refresh-btn');

    const weatherInfo = document.getElementById('weather-info');

    const cityNameElement = document.getElementById('city-name');

    const temperatureElement = document.getElementById('temperature');

    const descriptionElement = document.getElementById('description');

    const weatherIconElement = document.getElementById('weather-icon');

    function fetchWeather(city) {

        const url = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric`;

        const xhr = new XMLHttpRequest();

        xhr.open('GET', url, true);

        xhr.onload = function () {

            if (xhr.status === 200) {

                const weatherData = JSON.parse(xhr.responseText);

                updateWeatherDisplay(weatherData);

            } else {

                alert('City not found. Please try again.');

            }

        };

        xhr.send();

    }

    function updateWeatherDisplay(data) {

        const { name, main, weather } = data;

        cityNameElement.textContent = `${name}`;

        temperatureElement.textContent = `Temperature: ${main.temp}°C`;

        descriptionElement.textContent = `Description: ${weather[0].description}`;

        const iconUrl = `https://openweathermap.org/img/wn/${weather[0].icon}.png`;

        weatherIconElement.src = iconUrl;

        weatherIconElement.alt = weather[0].description;

    }

    searchButton.addEventListener('click', () => {

        const city = cityInput.value.trim();

        if (city) {

            fetchWeather(city);

        } else {

            alert('Please enter a city name.');

        }

    });

    refreshButton.addEventListener('click', () => {

        const city = cityInput.value.trim();

        if (city) {

            fetchWeather(city);

        }

    });

    fetchWeather('London');

});

    </script>

</body>

</html>

* **GitHub Link :-**

https://github.com/Dhruvi-Vavadiya/ReactJS/blob/main/Practical\_Assignment\_2.docx