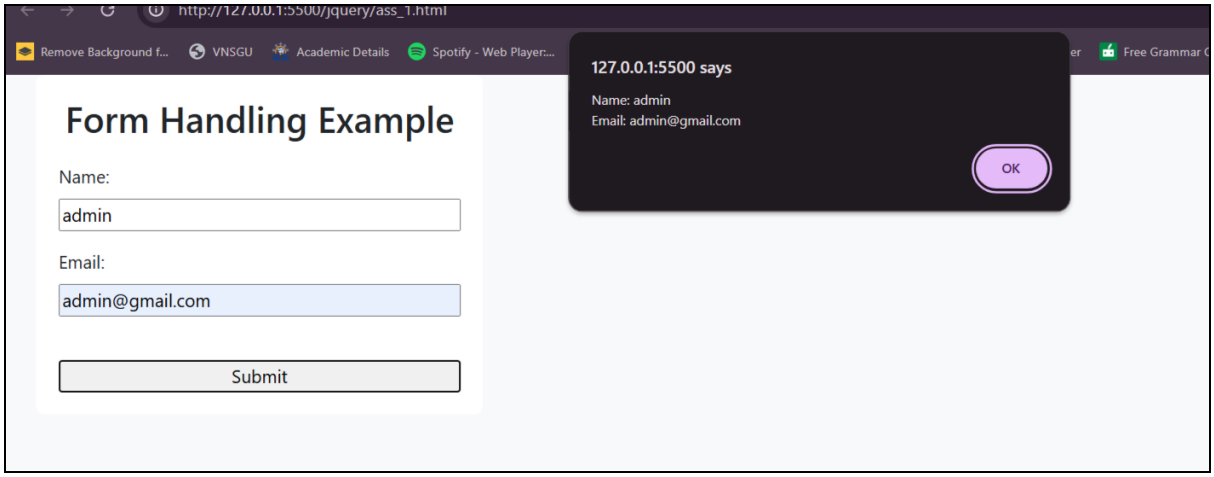
Assignment-2

JQUERY

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.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form Handling Example</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
    body {
      background-color: #f8f9fa;
      display: flex;
      padding-left: 30px  ;
    }
    form {
      background-color: white;
      padding: 20px;
      border-radius: 8px;
      width: 100%;
      max-width: 400px;
    }
    h2 {
      text-align: center;
      margin-bottom: 20px;
    }
    .inp    {
      width: 100%;
      margin-bottom: 15px;
```

```
        }
        #btn {
            width: 100%;
        }
    </style>
</head>
<body>
    <form id="frm">
        <h2>Form Handling Example</h2>
        <label for="name">Name:</label>
        <input type="text" id="name" class="inp" required><br>
        <label for="email">Email:</label>
        <input type="email" id="email" class="inp" required><br><br>
        <div style="align-content: center;">
            <button type="submit" id="btn">Submit</button>
        </div>
    </form>
    <script src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-/JqT3SQfawRcv/BIHPThkBvs0OEvtFFmqPF/lYI/Cxo=" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>
    <script>
        $(document).ready(function () {
            $('#frm').submit(function (e) {
                e.preventDefault();
                var name = $('#name').val();
                var email = $('#email').val();
                alert("Name: " + name + "\nEmail: " + email);
            });
        });
    </script>
</body>
</html>
```

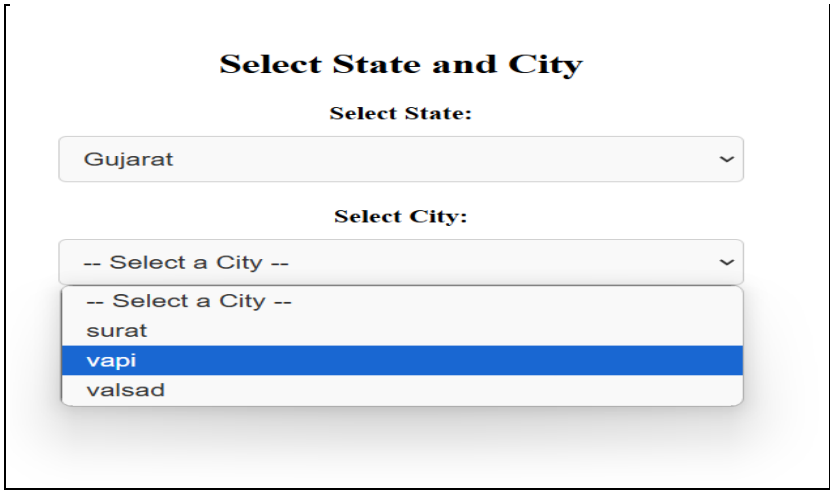


2. Create an HTML page that includes two dropdown menus (combos): one for selecting states and another for displaying cities. When the user selects a state from the primary combo, the secondary combo (cities) should dynamically update to display cities relevant to the selected state.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dynamic State and City Dropdown</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    .dropdown-container {
      background-color: #fff;
      padding: 20px;
      border-radius: 10px;
      max-width: 400px;
      width: 100%;
      text-align: center;
    }
    label {
      display: block;
      font-weight: bold;
```

```
        margin-bottom: 10px;
      }
    select {
        width: 100%;
        padding: 10px;
        margin-bottom: 20px;
        border-radius: 5px;
        border: 1px solid #ccc;
        background-color: #f9f9f9;
        font-size: 16px;
        color: #333;
        outline: none;
    }
  </style>
</head>
<body>
  <div class="dropdown-container">
    <h2>Select State and City</h2>
    <label for="state">Select State:</label>
    <select id="state">
      <option value="">-- Select a State --</option>
      <option value="Gujarat">Gujarat</option>
      <option value="Maharastra">Maharastra</option>
      <option value="Florida">Florida</option>
    </select>
    <label for="city">Select City:</label>
    <select id="city">
      <option value="">-- Select a City --</option>
    </select>
  </div>
```

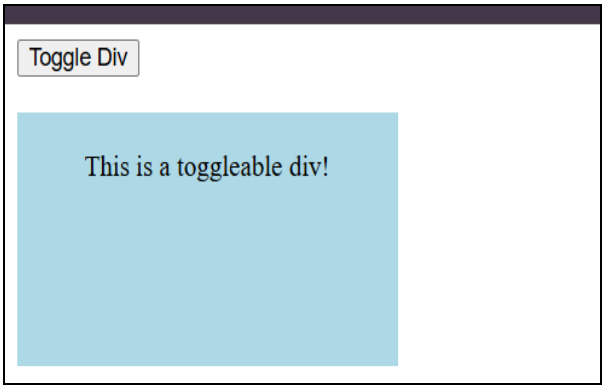
```
  <script>
    $(document).ready(function() {
      var citiesByState = {
        "Gujarat": ["surat", "vapi", "valsad"],
        "Maharastra": ["Mumbai", "pune", "Navi mumbai"],
        "Florida": ["Miami", "Orlando", "Tampa"]
      };
        $('#state').on('change', function() {
        var state = $(this).val();
        var cities = citiesByState[state] || [];
        $('#city').empty();
        $('#city').append('<option value="">-- Select a City --</option>');
        $.each(cities, function(index, city) {
          $('#city').append('<option value="' + city + '">' + city + '</option>');
        });
      });
    });
  </script>
</body>
</html>
```



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.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Toggle Div Visibility</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    #myDiv {
      width: 200px;
      height: 100px;
      background-color: lightblue;
      display: none;
      margin-top: 20px;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <button id="toggleButton">Toggle Div</button>
    <div id="myDiv">
    This is a toggleable div!
  </div>
  <script>
    $(document).ready(function() {
      $('#toggleButton').on('click', function() {
        $('#myDiv').toggle();
      });
    });
  </script>
```

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



4. Create a basic image slider using jQuery. Users should be able to click “Next”; and “Previous”; buttons to navigate between a series of images displayed on a webpage.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Basic jQuery Image Slider</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-color: #f4f4f4;
    }
    .slider {
      width: 400px;
      position: relative;
      text-align: center;
    }
```

```
        .slider img {
            width: 100%;
            display: none;
            border-radius: 10px;
        }
        .slider img.active {
            display: block;
        }
        .slider .controls {
            margin-top: 10px;
        }
        .slider button {
            padding: 10px 20px;
            font-size: 16px;
            margin: 5px;
            cursor: pointer;
            background-color: #007bff;
            color: white;
            border: none;
            border-radius: 5px;
        }
        .slider button:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <div class="slider">
        <img src="image/333333.png" class="active" alt="Image 1">
        <img src="image/333333 (1).png" alt="Image 2">
        <img src="image/333333 (2).png" alt="Image 3">
        <div class="controls">
```

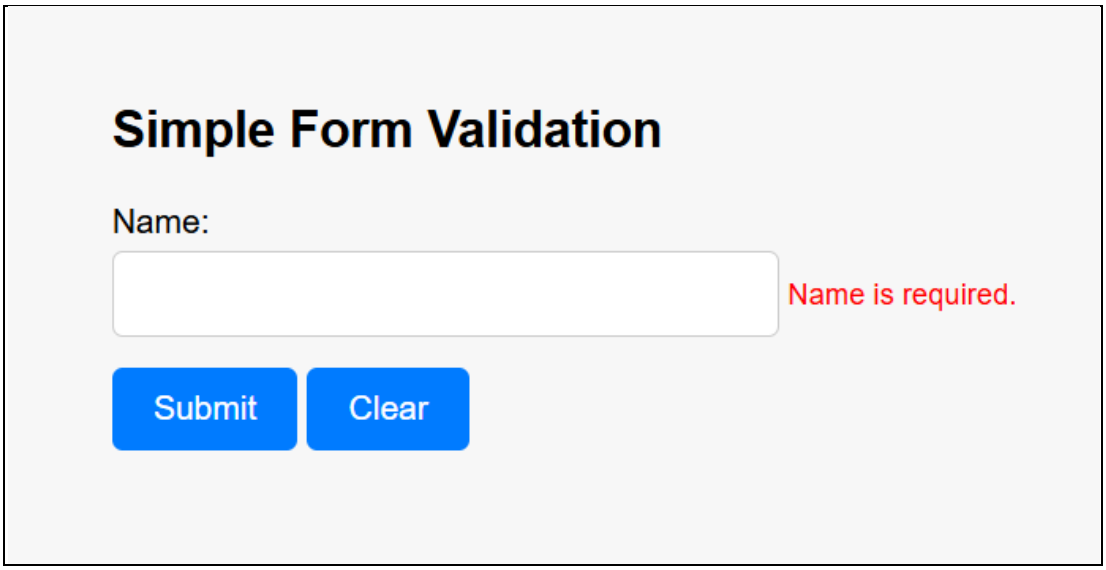
```
        <button id="prev">Previous</button>
        <button id="next">Next</button>
      </div>
    </div>
    <script>
      $(document).ready(function() {
        var currentIndex = 0;
        var images = $('.slider img');
        var totalImages = images.length;
        function showImage(index) {
          images.removeClass('active');
          images.eq(index).addClass('active');
        }
        $('#next').on('click', function() {
          currentIndex = (currentIndex + 1) % totalImages;
          showImage(currentIndex);
        });
        $('#prev').on('click', function() {
          currentIndex = (currentIndex - 1 + totalImages) % totalImages;
          showImage(currentIndex);
        });
      });
    </script>
</body>
</html>
```

5. Implement a simple client-side validation using jQuery. Check if a text input field is empty when a form is submitted. If it’s empty, display an error message next to the input field.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple Form Validation</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 50px;
      background-color: #f7f7f7;
    }
    .form-group {
      margin-bottom: 15px;
    }
    label {
      display: block;
      margin-bottom: 5px;
    }
    input[type="text"] {
      padding: 10px;
      width: 300px;
      font-size: 16px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    .error {
      color: red;
      font-size: 14px;
      display: none; }
```

```
        button {
            padding: 10px 20px;
            background-color: #007bff;
            color: white;
            border: none;
            border-radius: 5px;
            font-size: 16px;
            cursor: pointer;
        }
        button:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <h2>Simple Form Validation</h2>
    <form id="myForm">
        <div class="form-group">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name">
            <span class="error" id="nameError">Name is required.</span>
        </div>
        <button type="submit">Submit</button>
        <button type="reset">Clear</button>
    </form>
    <script>
        $(document).ready(function () {
            $('#myForm').on('submit', function (event) {
                event.preventDefault();
                var name = $('#name').val();
                if (name === "") {
                    $('#nameError').show();
```

```
        } else if (name !== "") {
            $("#nameError").hide();
        }
        else {
            $('#nameError').hide();
            alert('Form submitted successfully!');
        }
      });
    });
  </script>
</body>
</html>
```



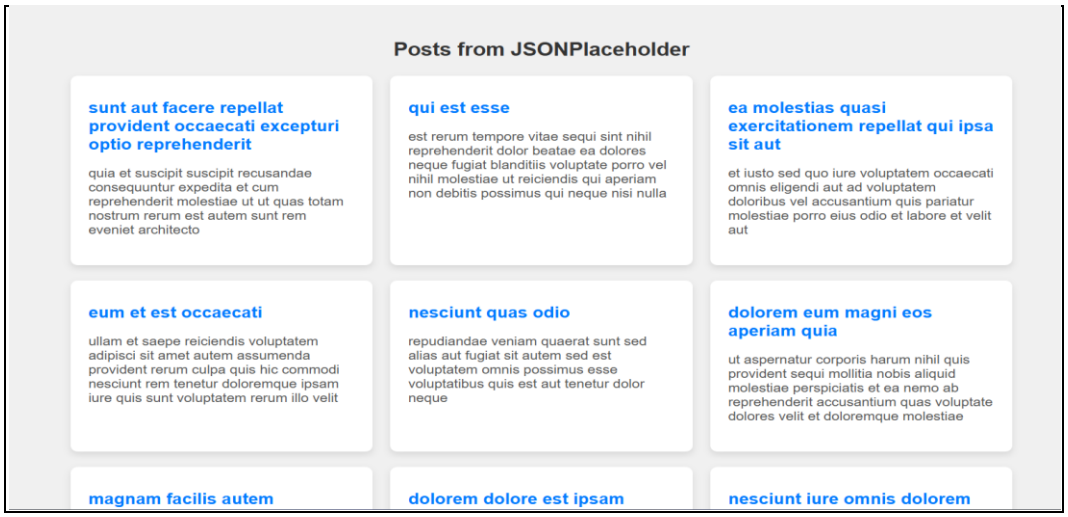
AJAX

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.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AJAX Data Retrieval Example</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      padding: 20px;
    }
    h2 {
      text-align: center;
      color: #333;
    }
    .post-list {
      display: flex;
      flex-wrap: wrap;
      justify-content: center;
      gap: 20px;
    }
    .post-card {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
      width: 300px;
```

```
            text-align: left;
            transition: transform 0.2s;
        }
        .post-card:hover {
            transform: scale(1.02);
        }
        .post-card h3 {
            font-size: 20px;
            margin: 10px 0;
            color: #007bff;
        }
        .post-card p {
            font-size: 16px;
            color: #555;
        }
        .loading {
            text-align: center;
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <h2>Posts from JSONPlaceholder</h2>
        <div class="post-list" id="postList"></div>
    <div class="loading" id="loadingMessage">Loading posts...</div>
    <script>
        $(document).ready(function() {
            $.ajax({
                url: "https://jsonplaceholder.typicode.com/posts",
                method: "GET",
                success: function(data) {
                    console.log("API Response:", data);
```

```
            $('#loadingMessage').hide();
            data.forEach(function(post) {
                var postCard = `
                    <div class="post-card">
                        <h3>${post.title}</h3>
                        <p>${post.body}</p>
                    </div>
                `;
                $('#postList').append(postCard);
            });
        },
        error: function(xhr, status, error) {
            console.error("AJAX Error:", status, error);
            $('#loadingMessage').text("Failed to load posts. Please try again.");
        }
    });
});
</script>
</body>
</html>
```

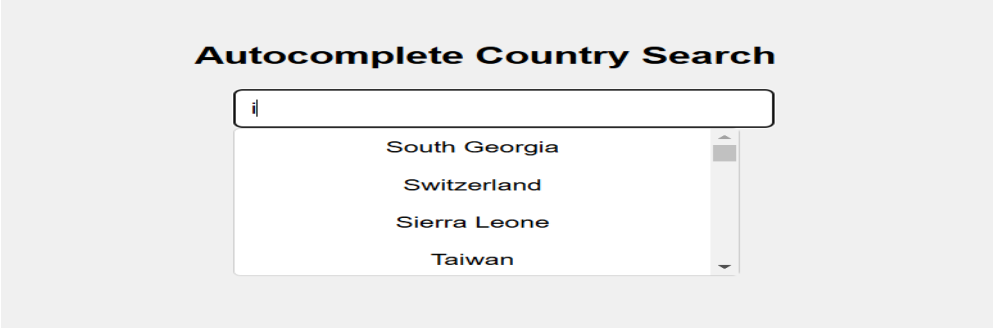


7. Create an autocomplete search feature using AJAX and jQuery. When users start typing a query into a search input field, dynamically fetch and display matching results from an API as dropdown suggestions.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Autocomplete Search Feature</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      padding: 20px;
      text-align: center;
    }
    .search-container {
      position: relative;
      margin-bottom: 20px;
      width: 300px;
      margin: auto;
    }
    input[type="text"] {
      width: 100%;
      padding: 10px;
      font-size: 16px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    .suggestions {
      position: absolute;
      background: white;
```

```
            border: 1px solid #ccc;
            border-radius: 5px;
            width: 100%;
            z-index: 1000;
            max-height: 150px;
            overflow-y: auto;
            display: none;
        }
        .suggestion-item {
            padding: 10px;
            cursor: pointer;
        }
        .suggestion-item:hover {
            background-color: #f0f0f0;
        }
    </style>
</head>
<body>
    <h2>Autocomplete Country Search</h2>
        <div class="search-container">
        <input type="text" id="countrySearch" placeholder="Start typing a country...">
        <div id="suggestions" class="suggestions"></div>
    </div>
    <script>
        $(document).ready(function() {
            $('#countrySearch').on('input', function() {
                var query = $(this).val().trim();
                $('#suggestions').empty();
                if (query === "") {
                    $('#suggestions').hide();
                    return;
                }
```

```
            $.ajax({
                url: `https://restcountries.com/v3.1/name/${encodeURIComponent(query)}`,
                method: "GET",
                success: function(data) {
                    $('#suggestions').show();
                    data.forEach(function(country) {
                        $('#suggestions').append(`
                            <div class="suggestion-item">${country.name.common}</div>
                        `);
                    });
                    $('.suggestion-item').on('click', function() {
                        $('#countrySearch').val($(this).text());
                        $('#suggestions').hide();
                    });
                },
                error: function() {
                    $('#suggestions').hide();                }
            });
        });
        $(document).click(function(event) {
            if (!$(event.target).closest('.search-container').length) {
                $('#suggestions').hide();
            }
        });
    });
  </script>
</body>
</html>
```



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.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AJAX Chat App</title>
  <style>
    body { font-family: Arial, sans-serif; }
    #chat-box {
        border: 1px solid #ccc;
         width: 100%;
        height: 300px;
        overflow-y: scroll;
        padding: 10px;
        margin-bottom: 10px;
    }
    #message { width: 80%; }
    #send-btn { width: 18%; }
  </style>
</head>
<body>
  <div id="chat-box">Loading messages...</div>
  <input type="text" id="message" placeholder="Type your message here...">
  <button id="send-btn">Send</button>
```

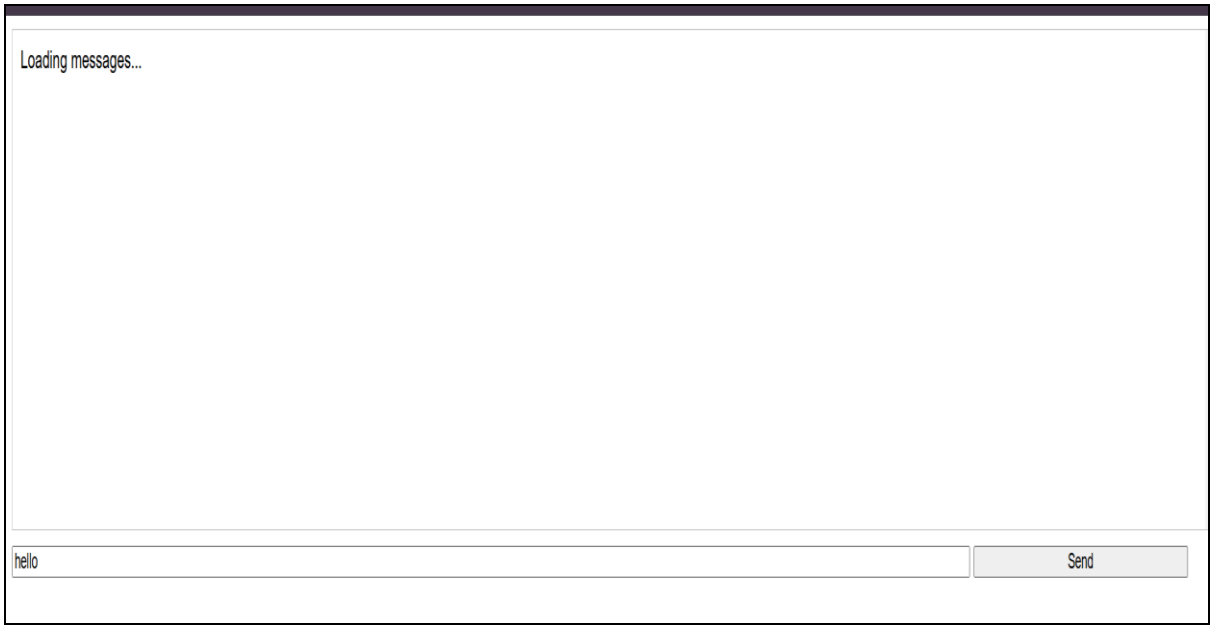
```
  <script>
    function fetchMessages() {
      const xhr = new XMLHttpRequest();
      xhr.open('GET', "C:/xampp/htdocs/fetch_messages.php", true);
      xhr.onload = function() {
        if (this.status === 200) {
          document.getElementById('chat-box').innerHTML = this.responseText;
          document.getElementById('chat-box').scrollTop = document.getElementById('chat-box').scrollHeight;
        }
      }
      xhr.send();
    }
    document.getElementById('send-btn').addEventListener('click', function() {
      const message = document.getElementById('message').value;
      if (message.trim() !== "") {
        const xhr = new XMLHttpRequest();
        xhr.open('POST', "C:/xampp/htdocs/save_message.php", true);
        xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
        xhr.onload = function() {
          if (this.status === 200) {
            fetchMessages(); // Refresh messages after sending
            document.getElementById('message').value = ''; // Clear message input
          } }
        xhr.send('message=' + encodeURIComponent(message));
      }
    });
    setInterval(fetchMessages, 2000);
    fetchMessages();
    </script>
    </body>
</html>
```

save_message.php

```
<?php
if (isset($_POST['message'])) {
    $message = htmlspecialchars($_POST['message']);
    $message = date('H:i:s') . " - " . $message . "\n";
    file_put_contents('chat.txt', $message, FILE_APPEND);
    echo 'Message saved!';
}
?>
```

fetch_messages.php

```
<?php
if (file_exists('chat.txt')) {
    echo nl2br(file_get_contents('chat.txt'));
} else {
    echo "No messages yet.";
}
?>
```



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

```
<!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;
      margin: 0;
      padding: 20px;
    }
    #content {
      margin: 20px 0;
    }
    #itemList {
      list-style-type: none;
      padding: 0;
    }
    #itemList li {
      margin: 10px 0;
      padding: 10px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    #loadMoreBtn {
      padding: 10px 15px;
      font-size: 16px;
      cursor: pointer;
      background-color: #007BFF;
      color: white;
```

```
        border: none;
        border-radius: 5px;
      }
      #loadMoreBtn:hover {
        background-color: #0056b3;
      }
    </style>
</head>
<body>
    <h1>Load More Example</h1>
    <div id="content">
      <ul id="itemList"></ul>
    </div>
    <button id="loadMoreBtn">Load More</button>
    <script>
      let currentPage = 1;
      const itemsPerPage = 3; // Number of items to load per request
      const loadMoreBtn = document.getElementById('loadMoreBtn');
      const itemList = document.getElementById('itemList');
      loadMoreBtn.addEventListener('click', () => {
        fetchItems(currentPage);
      });
      function fetchItems(page) {
        const apiUrl =
`https://jsonplaceholder.typicode.com/posts?_page=${page}&_limit=${itemsPerPage}`;
        fetch(apiUrl)
          .then(response => {
            if (!response.ok) {
              throw new Error('Network response was not ok');
            }
            return response.json();
          })
          .then(data => {
```

```
                appendItems(data);
                currentPage++;
                if (data.length < itemsPerPage) {
                    loadMoreBtn.style.display = 'none';
                }
            })
            .catch(error => {
                console.error('There was a problem with the fetch operation:', error);
            });
        }
        function appendItems(items) {
            items.forEach(item => {
                const li = document.createElement('li');
                li.textContent = item.title; // Change this to your item's property
                itemList.appendChild(li);
            });
        }
        fetchItems(currentPage);
    </script>
</body>
</html>
```

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.

```
<!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>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 20px;
      background-color: #f0f8ff;
      color: #333;
    }
    h1 {
      text-align: center;
    }
    #weatherForm {
      display: flex;
      justify-content: center;
      margin-bottom: 20px;
    }
    #cityInput {
      padding: 10px;
      font-size: 16px;
      width: 200px;
      border: 1px solid #ccc;
      border-radius: 4px;
    }
    #getWeatherBtn {
```

```
            padding: 10px 15px;
            font-size: 16px;
            cursor: pointer;
            background-color: #007BFF;
            color: white;
            border: none;
            border-radius: 4px;
            margin-left: 10px;
        }
        #getWeatherBtn:hover {
            background-color: #0056b3;
        }
        #weatherInfo {
            display: flex;
            flex-direction: column;
            align-items: center;
            margin-top: 20px;
        }
        #weatherIcon {
            width: 100px;
            height: 100px;
        }
        .temperature {
            font-size: 2rem;
            margin: 10px 0;
        }
        .description {
            font-size: 1.2rem;
        }
    </style>
</head>
<body>
```

```
<h1>Weather App</h1>
<div id="weatherForm">
  <input type="text" id="cityInput" placeholder="Enter city name" />
  <button id="getWeatherBtn">Get Weather</button>
</div>
<div id="weatherInfo">
  <img id="weatherIcon" src="" alt="" />
  <div class="temperature" id="temperature"></div>
  <div class="description" id="description"></div>
</div>
<script>
  const apiKey = '3316ccbd5e177f5f81936aeb2ac0e447';
  const cityInput = document.getElementById('cityInput');
  const getWeatherBtn = document.getElementById('getWeatherBtn');
  const weatherIcon = document.getElementById('weatherIcon');
  const temperature = document.getElementById('temperature');
  const description = document.getElementById('description');
  getWeatherBtn.addEventListener('click', () => {
    const cityName = cityInput.value;
    if (cityName) {
      fetchWeather(cityName);
    } else {
      alert("Please enter a city name.");
    }
  });
  function fetchWeather(city) {
    const apiUrl =
`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric`;
    fetch(apiUrl)
      .then(response => {
        if (!response.ok) {
          throw new Error('City not found');
        }
```

```
            return response.json();
        })
        .then(data => {
            displayWeather(data);
        })
        .catch(error => {
            alert(error.message);
            clearWeatherInfo();
        });
    }
    function displayWeather(data) {
        const temp = data.main.temp;
        const weatherDescription = data.weather[0].description;
        const iconCode = data.weather[0].icon;
        temperature.textContent = `${temp}°C`;
        description.textContent = weatherDescription.charAt(0).toUpperCase() +
weatherDescription.slice(1);
        weatherIcon.src = `https://openweathermap.org/img/wn/${iconCode}@2x.png`;
        weatherIcon.alt = weatherDescription;
    }
    function clearWeatherInfo() {
        temperature.textContent = '';
        description.textContent = '';
        weatherIcon.src = '';
        weatherIcon.alt = '';
    }
  </script>
</body>
</html>
```

Weather App

London

Get Weather

12.67°C

Broken clouds