MIT World Peace University

Full Stack Development
Third Year B. Tech, Semester 5

Implementation of Form Validation, JQuery, and Ajax

Lab Assignment 3

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

September 21, 2023

Contents

1 Aim

Write a program to perform following form validations using JavaScript, where all fields are mandatory, and some fields like Phone number, Email Address, Zip code Validation etc. are included. Also include JavaScript to access and manipulate Document Object Model (DOM) objects in an HTML web page. Include JQuery to develop to develop your application as an Ajax based application.

2 Objectives

- To understand what is form validation.
- To learn basic functioning of DOM objects.
- To learn how to apply various techniques to implement it.

3 Problem Statement

Write a program to design Student registration form by using HTML, CSS having following fields: Username, Email, Phone number, Password, Confirm Password and write external javascript code to achieve following validations:

1. Fields should not be empty. If spaces are entered those should be considered empty
2. Phone number must accept only numeric values and it should be 10 digits
3. Password length must be at least 7 and it should contain at least one capital letter, one digit and one special character from the set (&,$,#,)
4. Value entered in password field and confirm password fields must match

Email address must contain @ sign and a ., there should be few letters before the @ sign, should be three letters between @ sign and a . There must be 3 or 2 letters after the . (hint: Use regular expression)

4 Theory

5 Types of Form Validation

Definition 1 *Form validation is the process of checking that a form has been filled in correctly before it is submitted. It is used to prevent users from submitting invalid data, such as an email address without an @ sign or a password that is too short.*

5.1 Types

1. **Client-side validation:** Client-side validation is performed by the web browser before the form is submitted to the server. It is used to provide immediate feedback to the user about any errors in their input.
2. **Server-side validation:** Server-side validation is performed by the web server after the form has been submitted. It is used to ensure that the data submitted by the user is valid and safe to process.

3. **Database validation:** Database validation is performed by the database management system after the data has been stored in the database. It is used to ensure that the data is consistent with the database schema and any constraints defined on the table.

4. **Business logic validation:** Business logic validation is performed by the application after the data has been retrieved from the database. It is used to ensure that the data is consistent with the business rules defined by the application.

6 The Document Object Model

Definition 2 *The Document Object Model (DOM) is a programming interface for HTML and XML documents. It defines the logical structure of a document and the way it is accessed and manipulated.*

1. The DOM is used by JavaScript to access and manipulate HTML documents.

2. It is also used by other programming languages, such as PHP and Python, to generate HTML documents dynamically.

3. The DOM is an object-oriented model that represents HTML documents as a collection of objects.

4. Each object has properties and methods that can be used to access and manipulate the document.

5. The DOM is a platform-independent model that can be used with any programming language.

6.1 DOM Tree

Definition 3 *The DOM tree is a hierarchical representation of the HTML document. It consists of nodes that represent elements, attributes, text, and comments.*

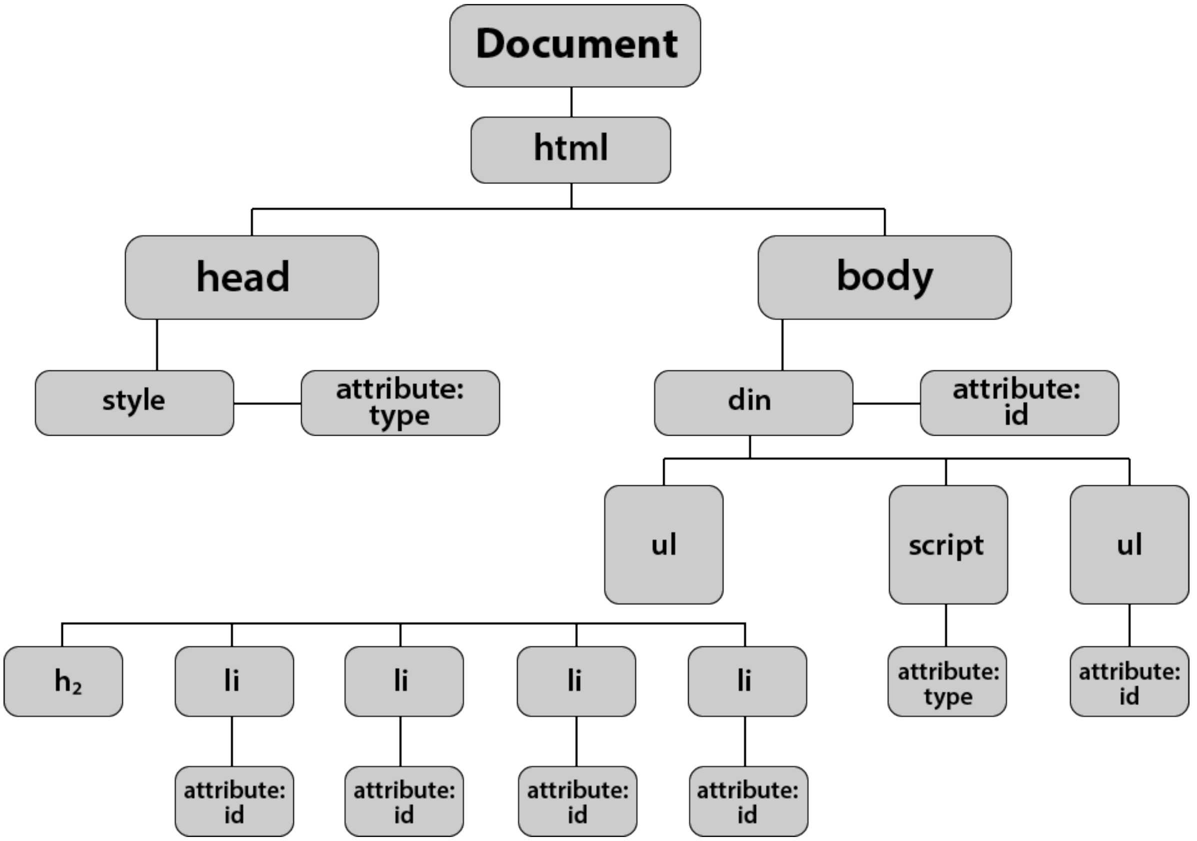


Figure 1: The DOM Tree

6.2 History

- The DOM was first introduced in 1998 by the World Wide Web Consortium (W3C).
- It was designed to be a platform-independent model for representing HTML documents.
- It is now supported by all major web browsers, including Internet Explorer, Firefox, Chrome, Safari, and Opera.

6.3 Advantages

1. The DOM is a platform-independent model that can be used with any programming language.
2. It is a standard model that is supported by all major web browsers.
3. It is an object-oriented model that represents HTML documents as a collection of objects.
4. It is a hierarchical model that represents the logical structure of an HTML document.
5. It is a dynamic model that can be used to access and manipulate HTML documents.
6. It is a flexible model that can be extended to support new features.
7. It is a powerful model that can be used to create interactive web applications.
8. It is a simple model that is easy to learn and understand.

9. It is a scalable model that can be used to create large web applications.

7 JQuery

Definition 4 *JQuery is a JavaScript library that simplifies the process of creating interactive web pages. It is used to manipulate HTML documents and handle events.*

7.1 History and Significance

1. JQuery was first released in 2006 by John Resig.
2. It was designed to simplify the process of creating interactive web pages.
3. It is now supported by all major web browsers, including Internet Explorer, Firefox, Chrome, Safari, and Opera.
4. It is used by many popular websites, such as Google, Facebook, Twitter, and YouTube.
5. It is used by many popular web applications, such as WordPress, Drupal, Joomla, and Magento.

7.2 Advantages

1. **Simplified DOM Manipulation:** jQuery simplifies DOM (Document Object Model) manipulation tasks. It provides easy-to-use methods to select, modify, and manipulate HTML elements.

```
<!-- HTML -->
<button id="myButton">Click Me</button>

<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // Selecting an element by ID
    var button = $("#myButton");

    // Modifying its text
    button.text("New Text");
</script>
```

Listing 1: Example: Selecting and Modifying Elements

2. **Event Handling:** jQuery simplifies event handling by providing methods like ‘.on()‘ and ‘.click()‘ for attaching event handlers to elements.

```
<!-- HTML -->
<button id="myButton">Click Me</button>

<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // Event handling with jQuery
    $("#myButton").click(function() {
        alert("Button Clicked!");
    });
</script>
```

Listing 2: Example: Event Handling

3. **AJAX Support:** jQuery streamlines AJAX (Asynchronous JavaScript and XML) requests, making it easier to fetch data from a server without reloading the entire page.

```
// jQuery AJAX request
$.ajax({
    url: "https://api.example.com/data",
    method: "GET",
    success: function(data) {
        console.log(data);
    },
    error: function(error) {
        console.error(error);
    }
});
```

Listing 3: Example: AJAX Request

4. **Cross-Browser Compatibility:** jQuery abstracts away browser-specific differences, ensuring consistent behavior across various browsers.

5. **Rich Set of Plugins:** jQuery offers a vast ecosystem of plugins that extend its functionality, allowing developers to add features like sliders, date pickers, and more without reinventing the wheel.

6. **Animation and Effects:** jQuery provides methods for creating animations and adding visual effects to web pages with ease.

```
<!-- HTML -->
<div id="box">Click Me</div>

<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // jQuery animation
    $("#box").click(function() {
        $(this).animate({ left: "200px" }, "slow");
    });
</script>
```

Listing 4: Example: Animation

7.3 JQuery Selectors

jQuery provides a variety of selectors that allow you to target and manipulate specific elements in the HTML DOM. Here are some commonly used jQuery selectors:

1. **Element Selector:** Selects all elements with a specified tag name.

```
<!-- HTML -->
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>

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

```
<script>
    // Select all <li> elements
    var listItems = $("li");
</script>
```

Listing 5: Example: Element Selector

2. **ID Selector:** Selects a single element with a specific ID attribute.

```
<!-- HTML -->
<div id="myDiv">Hello, World!</div>

<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // Select the element with ID "myDiv"
    var myElement = $("#myDiv");
</script>
```

Listing 6: Example: ID Selector

3. **Class Selector:** Selects all elements with a specific class attribute.

```
<!-- HTML -->
<p class="highlight">This is a highlighted text.</p>
<p>This is a normal text.</p>

<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // Select all elements with class "highlight"
    var highlightedElements = $(".highlight");
</script>
```

Listing 7: Example: Class Selector

4. **Attribute Selector:** Selects elements with a specific attribute and optional value.

```
<!-- HTML -->
<input type="text" id="username" name="username" value="John" />

<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // Select element with attribute "name"
    var usernameInput = $("[name='username']");
</script>
```

Listing 8: Example: Attribute Selector

5. **Descendant Selector:** Selects all elements that are descendants of a specified element.

```
<!-- HTML -->
<ul id="myList">
    <li>Item 1</li>
    <li>Item 2</li>
</ul>

<!-- jQuery -->
```

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // Select all <li> elements inside the element with ID "myList"
    var listItems = $("#myList li");
</script>
```

Listing 9: Example: Descendant Selector

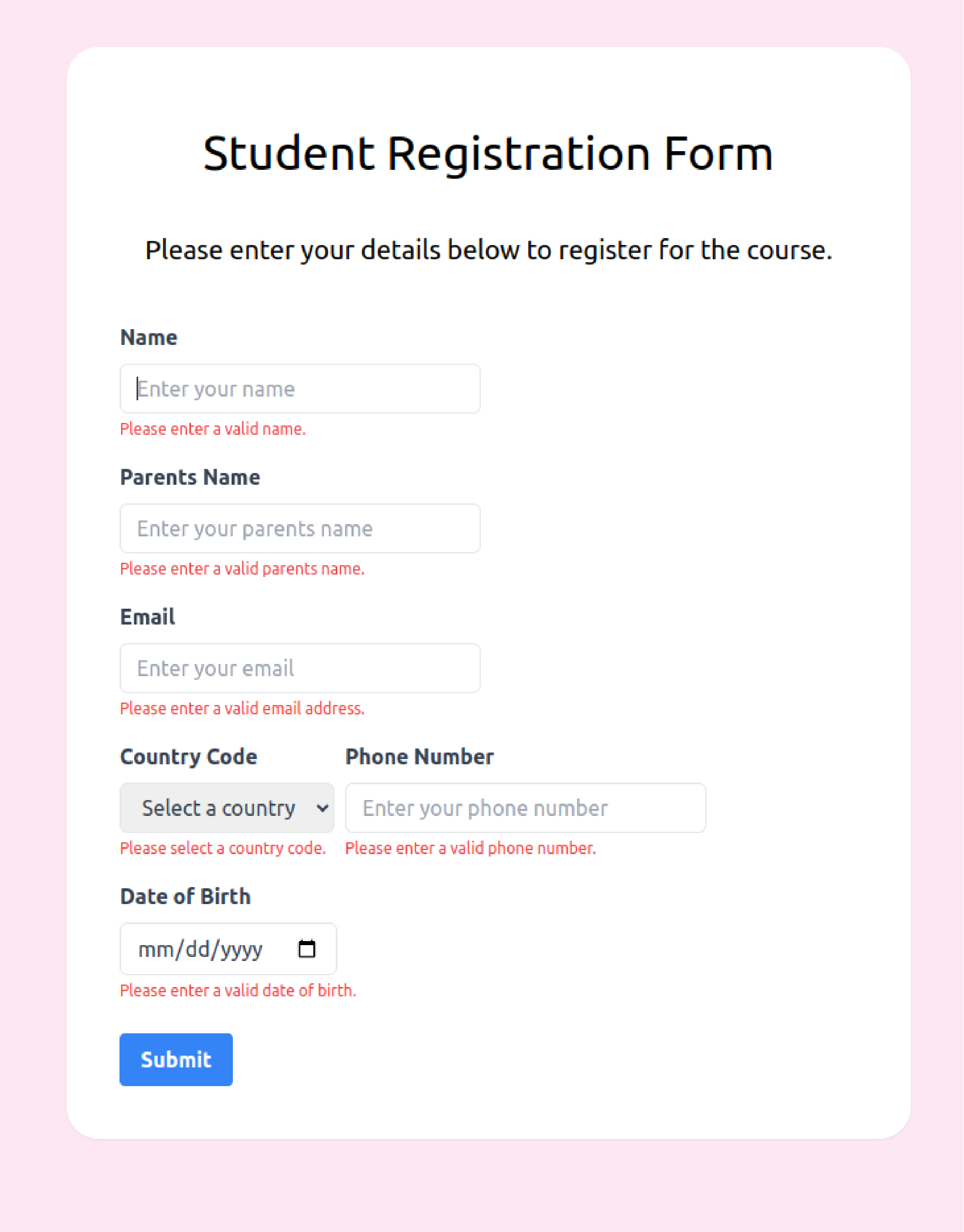
8 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code
Compilers or Interpreters: Brave Browser (Chromium v117.0.5938.88.)

9 Input and Output

1. The Student Validaiton form was created, and the relevant fields were added.
2. The validation was done on client side using Javascript Successfully.
3. The screenshots of the same are given below.

10 Screenshots



Student Registration Form

Please enter your details below to register for the course.

Name

Enter your name

Please enter a valid name.

Parents Name

Enter your parents name

Please enter a valid parents name.

Email

Enter your email

Please enter a valid email address.

Country Code

Select a country

Please select a country code.

Phone Number

Enter your phone number

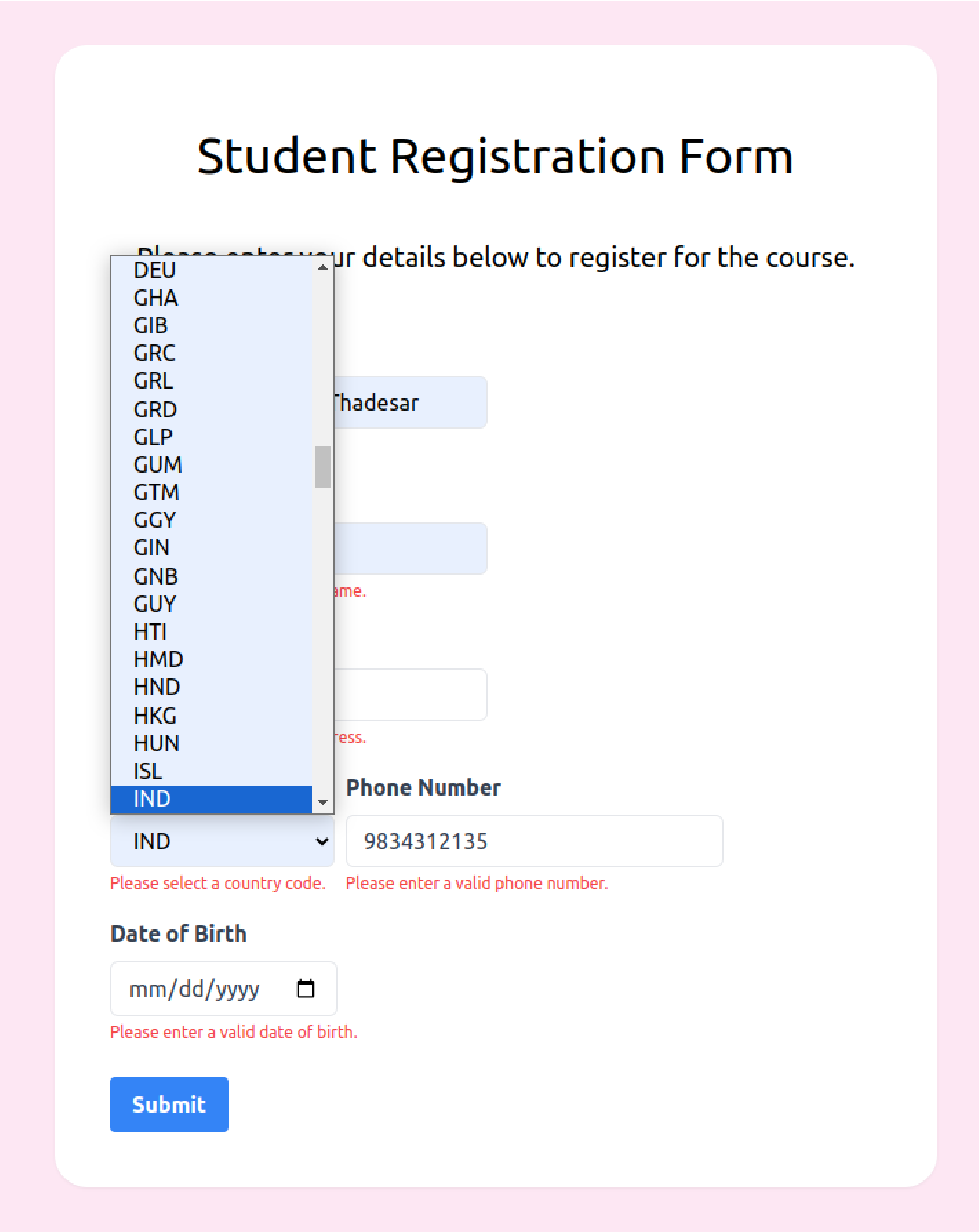
Please enter a valid phone number.

Date of Birth

mm/dd/yyyy

Please enter a valid date of birth.

Figure 2: The Student Validation form with Mandatory Fields. You cant submit without filling them. The Phone Number field only accepts numbers.



Student Registration Form

Please enter your details below to register for the course.

hadesar

ame.

ess.

Phone Number

IND

9834312135

Please select a country code.

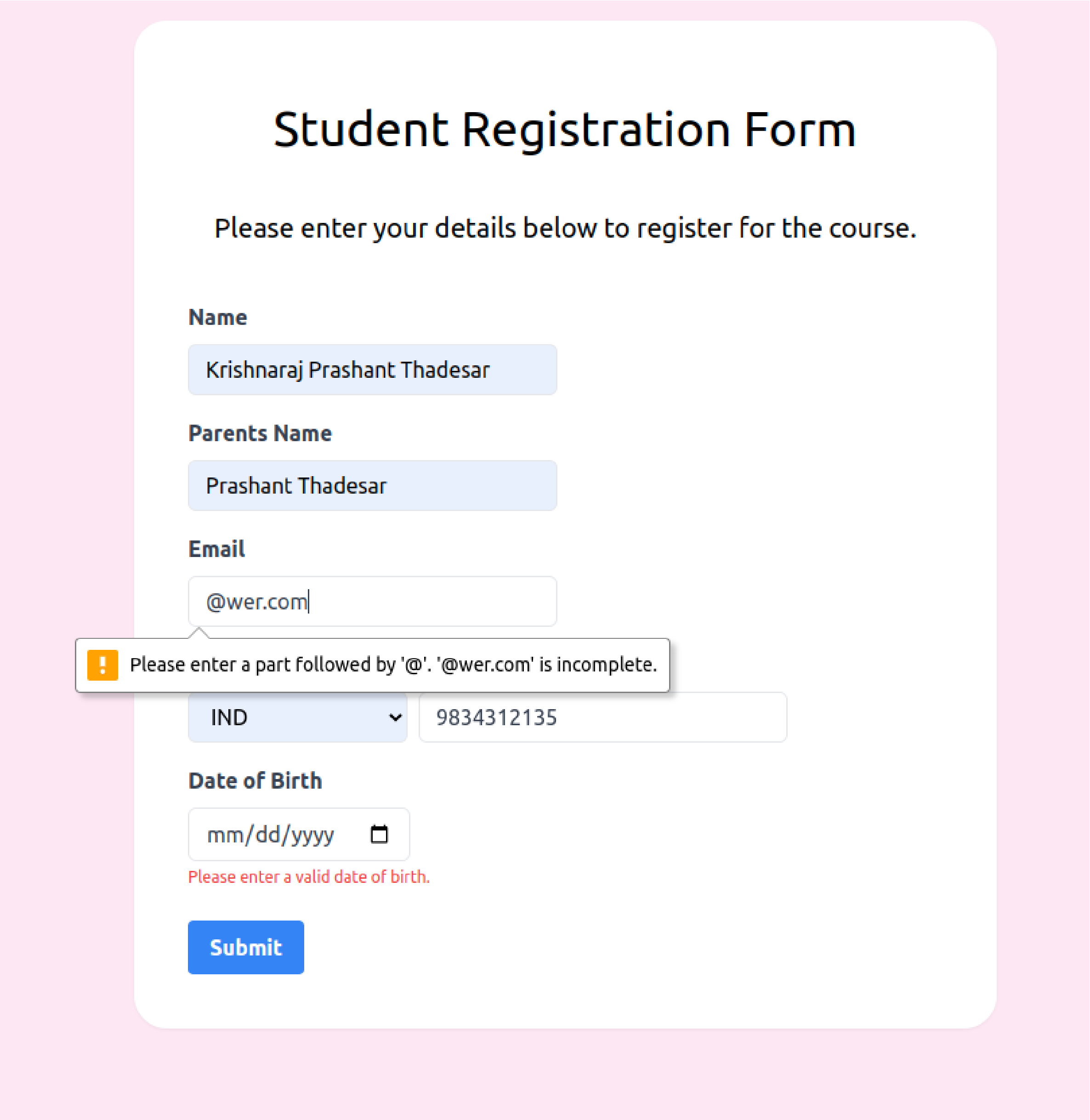
Please enter a valid phone number.

Date of Birth

mm/dd/yyyy

Please enter a valid date of birth.

Figure 3: The Country Code Selection Field. The Countries REST api is called by using Jquery AJAX to get the list of countries.



Student Registration Form

Please enter your details below to register for the course.

Name

Krishnaraj Prashant Thadesar

Parents Name

Prashant Thadesar

Email

@wer.com

Please enter a part followed by '@'. '@wer.com' is incomplete.

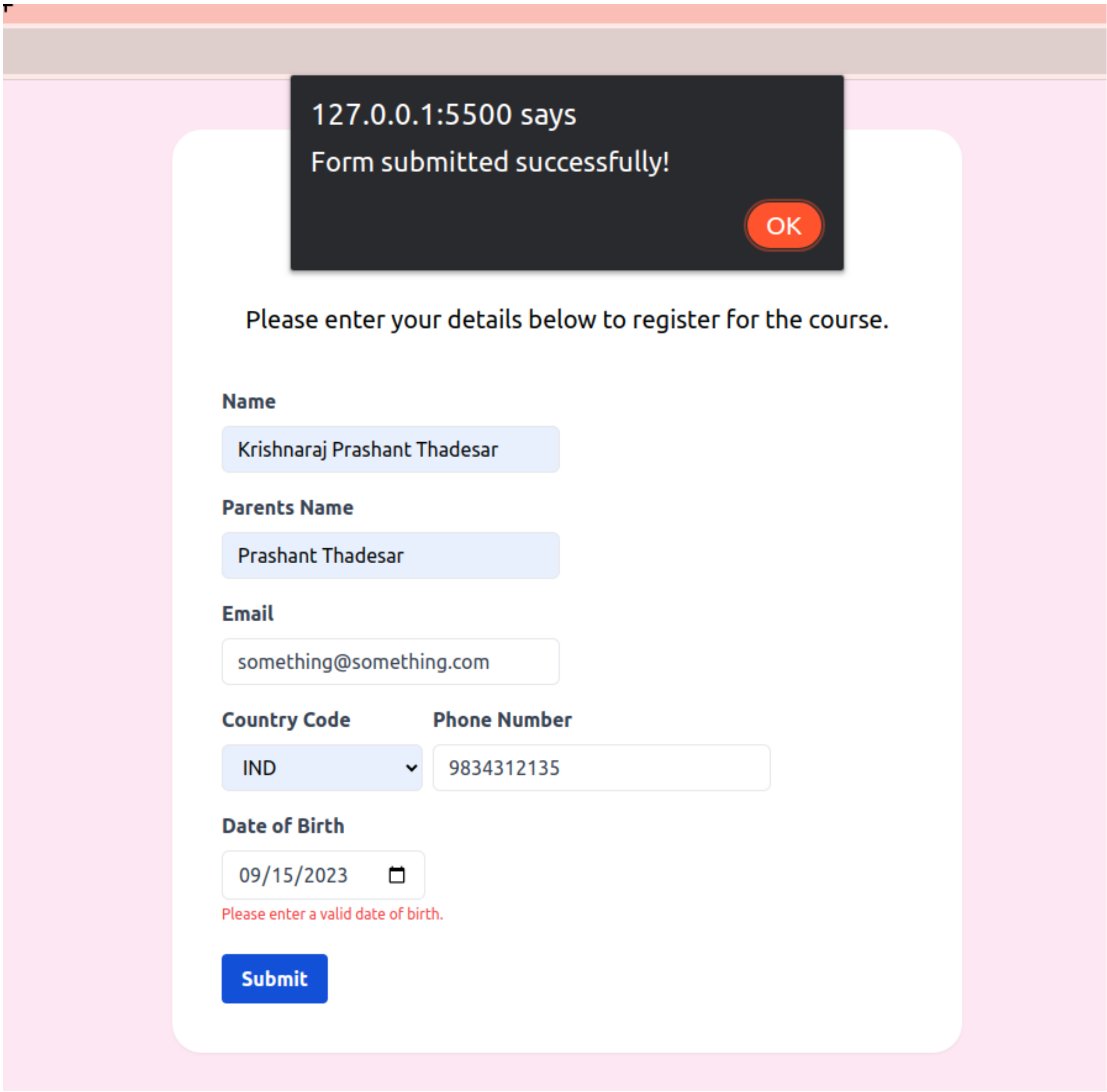
IND 9834312135

Date of Birth

mm/dd/yyyy

Please enter a valid date of birth.

Figure 4: Email Validation done on client side JS



127.0.0.1:5500 says

Form submitted successfully!

Please enter your details below to register for the course.

Name: Krishnaraj Prashant Thadesar

Parents Name: Prashant Thadesar

Email: something@something.com

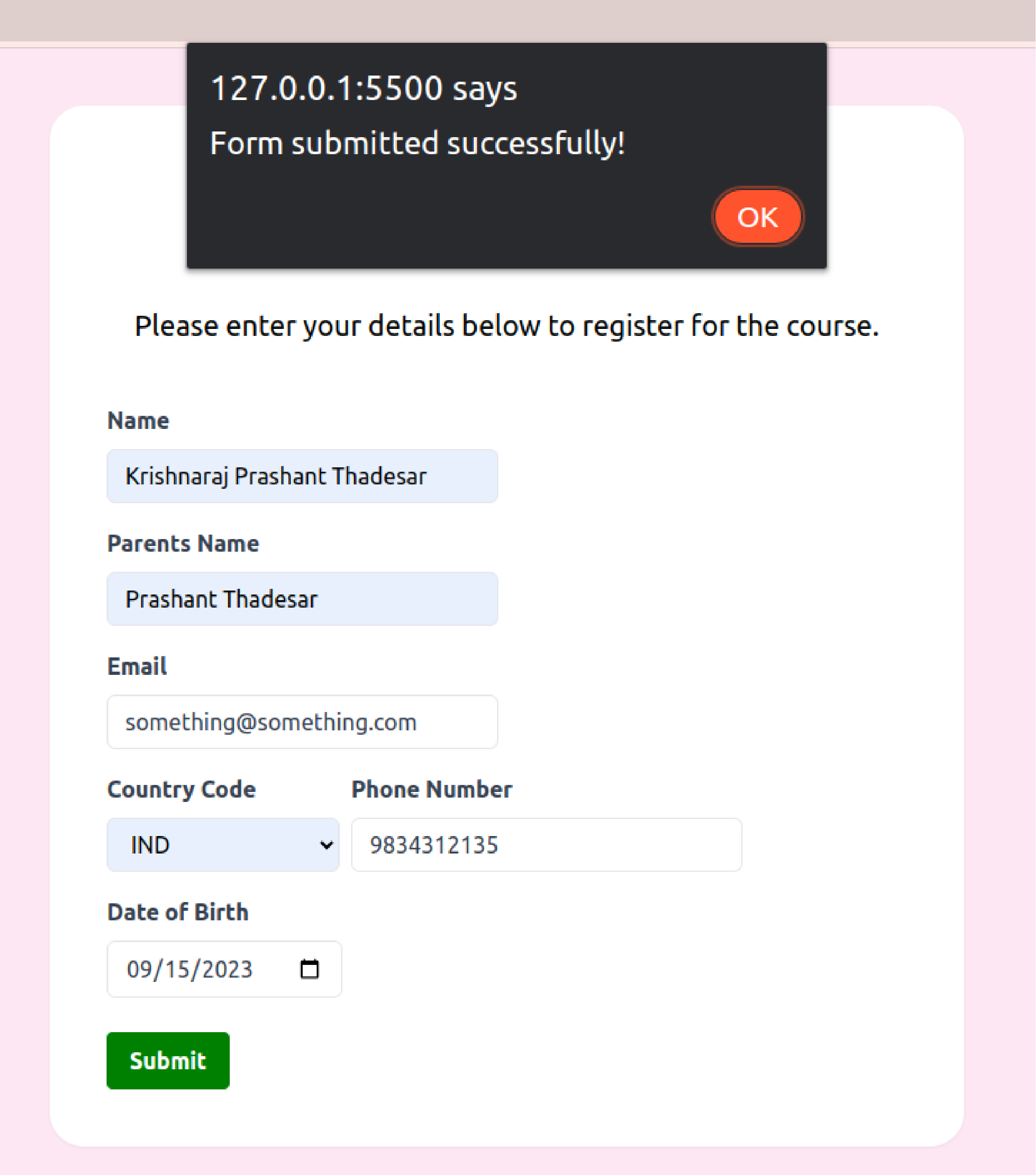
Country Code: IND

Phone Number: 9834312135

Date of Birth: 09/15/2023

Please enter a valid date of birth.

Figure 5: Form being submitted after entering everything. Notice the button is blue.



127.0.0.1:5500 says

Form submitted successfully!

Please enter your details below to register for the course.

Name: Krishnaraj Prashant Thadesar

Parents Name: Prashant Thadesar

Email: something@something.com

Country Code: IND

Phone Number: 9834312135

Date of Birth: 09/15/2023

Figure 6: The Color of the button changes to green for 3 seconds, as instructed in the problem statement. Only then the form submits.

11 Code

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Student Registration Form</title>
```

```html
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.7/tailwind.min.css"
    />
  </head>
  <body class="bg-pink-100">
    <div class="mx-auto my-10 bg-white rounded-3xl shadow-sm w-1/3 p-10">
      <h2 class="font-medium mb-10 text-4xl self-center text-center mt-5">
        Student Registration Form
      </h2>
      <div class="text-xl text-center my-10">
        Please enter your details below to register for the course.
      </div>
      <form>
        <div class="mb-4">
          <label
            class="block text-gray-700 font-bold mb-2"
            for="name"
          >
            Name
          </label>
          <input
            class="border rounded-md py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
            id="name"
            type="text"
            placeholder="Enter your name"
          />
          <p class="text-red-500 text-xs mt-1 hidden" id="name_error">
            Please enter a valid name.
          </p>
        </div>
        <div class="mb-4">
          <label
            class="block text-gray-700 font-bold mb-2"
            for="parents_name"
          >
            Parents Name
          </label>
          <input
            class="border rounded-md py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
            id="parents_name"
            type="text"
            placeholder="Enter your parents name"
          />
          <p
            class="text-red-500 text-xs mt-1 hidden"
            id="parents_name_error"
          >
            Please enter a valid parents name.
          </p>
        </div>
        <div class="mb-4">
          <label
            class="block text-gray-700 font-bold mb-2"
            for="email"
          >
```

```
            Email
          </label>
          <input
            class="border rounded-md py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
            id="email"
            type="email"
            placeholder="Enter your email"
          />
          <p
            class="text-red-500 text-xs mt-1 hidden"
            id="email_error"
          >
            Please enter a valid email address.
          </p>
        </div>
        <div class="mb-4 flex">
          <div class="mr-2">
            <label
              class="block text-gray-700 font-bold mb-2"
              for="phone_country_code"
            >
              Country Code
            </label>
            <select
              class="border rounded-md py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
              id="phone_country_code"
            >
              <option value="">Select a country</option>
            </select>
            <p
              class="text-red-500 text-xs mt-1 hidden"
              id="phone_country_code_error"
            >
              Please select a country code.
            </p>
          </div>
          <div class="flex-1">
            <label
              class="block text-gray-700 font-bold mb-2"
              for="phone_number"
            >
              Phone Number
            </label>
            <input
              class="border rounded-md py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
              id="phone_number"
              type="tel"
              placeholder="Enter your phone number"
            />
            <p
              class="text-red-500 text-xs mt-1 hidden"
              id="phone_number_error"
            >
              Please enter a valid phone number.
            </p>
          </div>
        </div>
        <div class="mb-4">
          <label
            class="block text-gray-700 font-bold mb-2"
            for="date_of_birth"
          >
            Date of Birth
          </label>
          <input
            class="border rounded-md py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
            id="date_of_birth"
            type="date"
          />
          <p
            class="text-red-500 text-xs mt-1 hidden"
            id="date_of_birth_error"
          >
            Please enter a valid date of birth.
          </p>
        </div>
        <div class="mt-6">
          <button
            class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline"
            type="submit"
          >
            Submit
          </button>
        </div>
      </form>
    </div>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script>
      $(document).ready(function () {
        const nameInput = $("#name");
        const parentsNameInput = $("#parents_name");
        const emailInput = $("#email");
        const phoneCountryCodeInput = $("#phone_country_code");
        const phoneNumberInput = $("#phone_number");
        const dateOfBirthInput = $("#date_of_birth");
        const nameError = $("#name_error");
        const parentsNameError = $("#parents_name_error");
        const emailError = $("#email_error");
        const phoneNumberError = $("#phone_number_error");
        const dateOfBirthError = $("#date_of_birth_error");
        const phoneCountryCodeError = $("#phone_country_code_error");

        function validateName() {
          if (nameInput.val().trim() === "") {
            nameError.removeClass("hidden");
            return false;
          } else {
            nameError.addClass("hidden");
            return true;
          }
        }

        function validateParentsName() {
          if (parentsNameInput.val().trim() === "") {
            parentsNameError.removeClass("hidden");
            return false;
          } else {
            parentsNameError.addClass("hidden");
            return true;
          }
        }

        function validateEmail() {
          const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
          if (!emailRegex.test(emailInput.val())) {
            emailError.removeClass("hidden");
            return false;
          } else {
            emailError.addClass("hidden");
            return true;
          }
        }

        function validatePhoneNumber() {
          const phoneRegex = /^[0-9]{10}$/;
          if (!phoneRegex.test(phoneNumberInput.val())) {
            phoneNumberError.removeClass("hidden");
            return false;
          } else {
            phoneNumberError.addClass("hidden");
            return true;
          }
        }

        function validateDateOfBirth() {
          const dobRegex = /^\d{4}-\d{2}-\d{2}$/;
          if (!dobRegex.test(dateOfBirthInput.val())) {
            dateOfBirthError.removeClass("hidden");
            return false;
          } else {
            dateOfBirthError.addClass("hidden");
            return true;
          }
        }

        function validatePhoneCountryCode() {
          if (phoneCountryCodeInput.val() === "") {
            phoneCountryCodeError.removeClass("hidden");
            return false;
          } else {
            phoneCountryCodeError.addClass("hidden");
            return true;
          }
        }

        function validateForm() {
          const isNameValid = validateName();
          const isParentsNameValid = validateParentsName();
          const isEmailValid = validateEmail();
          const isPhoneNumberValid = validatePhoneNumber();
          const isDateOfBirthValid = validateDateOfBirth();
          const isPhoneCountryCodeValid = validatePhoneCountryCode();
          return (
            isNameValid &&
            isParentsNameValid &&
            isEmailValid &&
            isPhoneNumberValid &&
            isDateOfBirthValid &&
            isPhoneCountryCodeValid
          );
        }

        $.ajax({
          url: "https://restcountries.com/v3.1/all",
          type: "GET",
          success: async function (data) {
            // Sort data by country name in ascending order
            await data.sort((a, b) =>
              a.name.common.localeCompare(b.name.common)
            );
            const countrySelect = $("#phone_country_code");
            $.each(data, function (index, country) {
              const option = $("<option>")
                .val(country.cca3)
                .text(country.cca3);
              countrySelect.append(option);
            });
          },
          error: function () {
            const countrySelect = $("#phone_country_code");
            const option = $("<option>")
              .val("")
              .text("Error loading countries");
            countrySelect.append(option);
          },
        });

        $("form").submit(function (event) {
          event.preventDefault();
          if (validateForm()) {
            const submitButton = $(’button[type="submit"]’);
            submitButton.css("background-color", "green");
            submitButton.prop("disabled", true);
            alert("Form submitted successfully!");
            setTimeout(function () {
              submitButton.css("background-color", "blue");
              submitButton.prop("disabled", false);
            }, 3000);
          }
        });
        $(document).ready(function () {
          $("form").submit(function (event) {
            event.preventDefault();
            if (validateForm()) {
              console.log("hi");
              const submitButton = $(’button[type="submit"]’);
              submitButton.css("background-color", "green");
              submitButton.prop("disabled", true);
              setTimeout(function () {
                submitButton.css("background-color", "blue");
                submitButton.prop("disabled", false);
              }, 3000);
            }
          });
        });
      });
    </script>
  </body>
</html>
```

Listing 10: "index.html"

12 Conclusion

Thus, we have successfully implemented form validation using Javascript and used Jquery as well as JQuery Ajax.

13 FAQ

1. **Reasons for Form Validations:**

   - **Data Integrity:** Form validations ensure that the data submitted by users is accurate and follows the expected format. This prevents incorrect or malicious data from being processed, maintaining the integrity of your application’s database.
   - **Enhanced User Experience:** Validations provide immediate feedback to users, helping them correct errors before submission. This enhances the user experience and reduces frustration.
   - **Security:** Proper form validations can prevent common security vulnerabilities like SQL injection and cross-site scripting (XSS) attacks by rejecting potentially harmful input.

2. **Modifying Attribute Value using DOM:** In JavaScript, you can modify an attribute value using the DOM (Document Object Model) as follows:

```
// Assuming ’element’ is a reference to the HTML element you want to modify.

// To change the ’src’ attribute of an image:
element.setAttribute(’src’, ’new_image.jpg’);

// To modify the ’href’ attribute of a link:
element.setAttribute(’href’, ’new_url.html’);

```

Listing 11: Example: Modifying Attribute Value using DOM

3. **jQuery Ajax:** jQuery Ajax is a feature in the jQuery library that allows you to make asynchronous HTTP requests to a server and handle the responses without requiring a full page reload. It offers a simplified way to perform tasks like fetching data from a server, sending data to a server, or updating a web page without the need for extensive JavaScript code.

```
// jQuery AJAX request
$.ajax({
    url: "https://api.example.com/data",
    method: "GET",
    success: function(data) {
        console.log(data);
    },
    error: function(error) {
        console.error(error);
    }
});

```

Listing 12: Example: AJAX Request