**Full Stack Development Laboratory 03**

**Name:Janhavi Pawar**

**Roll no:1**

**Panel:H**

**PRN:1032231531**

**Aim: Client-side Form Validations using JavaScript, DOM real-time update, JQuery to develop Ajax based applications.**

**Aim:** Write a program to perform following form validations using JavaScript:

a) All fields mandatory,

b) Phone number, Email Address, Zip code Validation etc.

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.

**Objectives:**

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

**Theory:**

1. Different types of form validations.

Ans:-

### **Required Field Validation**

* **Purpose**: Ensures that certain fields are filled out before submission.
* **Example**: A login form requiring both a username and password.

**2. Data Type Validation**

* **Purpose**: Checks that the input matches a specific data type.
* **Examples**:
  + **Text**: Ensures input is a string (e.g., names).
  + **Number**: Checks that the input is numeric (e.g., age or quantity).
  + **Email**: Verifies that the input follows the standard email format.
  + **Date**: Ensures input is in a valid date format.

**3. Format Validation**

* **Purpose**: Checks that the input adheres to a specified format.
* **Examples**:
  + **Phone Number**: Validates that the number matches a phone number pattern.
  + **Postal Code**: Ensures the postal code adheres to a particular format (e.g., ZIP codes).

**4. Length Validation**

* **Purpose**: Ensures that the input falls within a specific length range.
* **Examples**:
  + **Minimum Length**: Requires a minimum number of characters (e.g., passwords).
  + **Maximum Length**: Limits the number of characters (e.g., username).

**5. Range Validation**

* **Purpose**: Checks that numeric input falls within a specified range.
* **Examples**:
  + **Age**: Must be between 0 and 120.
  + **Quantity**: Must be between 1 and 100.

**6. Pattern Validation**

* **Purpose**: Uses regular expressions (regex) to enforce a specific input pattern.
* **Examples**:
  + **Email Address**: Validates email format using a regex pattern.
  + **Password Complexity**: Enforces rules like requiring uppercase letters, digits, and special characters.

**7. Confirmation Validation**

* **Purpose**: Ensures that two related fields match.
* **Examples**:
  + **Password Confirmation**: Confirms that the "Password" and "Confirm Password" fields match.
  + **Email Confirmation**: Checks that email address fields match.

**8. Custom Validation**

* **Purpose**: Allows for more complex validation logic tailored to specific needs.
* **Examples**:
  + **Unique Username**: Checks that a username is not already taken by querying a database.
  + **Age Verification**: Ensures that the user meets age requirements based on additional conditions.

**9. Asynchronous Validation**

* **Purpose**: Validates input against data that requires an asynchronous request.
* **Examples**:
  + **Username Availability**: Checks if a username is available via an API call.
  + **Email Uniqueness**: Verifies that an email address is not already registered.

**10. Client-Side vs. Server-Side Validation**

* **Client-Side**: Performed in the browser using JavaScript or HTML5 attributes, providing immediate feedback to users.
* **Server-Side**: Performed on the server after form submission, ensuring data integrity and security, as it cannot be bypassed by the user.

1. HTML Document Object Model.

Ans:-

The HTML Document Object Model (DOM) is a programming interface for web documents. It represents the structure of an HTML document as a tree of objects, where each node corresponds to a part of the document. Here’s a brief overview:

**Key Concepts:**

1. **Tree Structure**:
   * The HTML document is represented as a tree of nodes. Each node corresponds to a part of the document, such as elements, attributes, and text.
2. **Nodes**:
   * **Element Nodes**: Represent HTML elements (e.g., <div>, <p>).
   * **Attribute Nodes**: Represent attributes of elements (e.g., class, id).
   * **Text Nodes**: Represent the text content within elements.
3. **Document Object**:
   * The root of the DOM tree is the document object, which provides methods and properties to interact with the content of the web page.
4. **Access and Manipulation**:
   * The DOM allows for dynamic access and manipulation of HTML elements. You can use JavaScript to change content, structure, and style of the document.
5. **Events**:
   * The DOM supports event handling, allowing scripts to respond to user actions like clicks, keystrokes, and mouse movements.

**Example:**

For an HTML document like this:

html

<!DOCTYPE html>

<html lang="en">

<head>

<title>My Page</title>

</head>

<body>

<h1>Hello, World!</h1>

<p id="description">This is a description.</p>

</body>

</html>

The DOM structure would look like:

* document
  + html
    - head
      * title ("My Page")
    - body
      * h1 ("Hello, World!")
      * p (id="description", "This is a description.")

**Common DOM Methods:**

* **getElementById(id)**: Selects an element by its ID.
* **querySelector(selector)**: Selects an element using a CSS selector.
* **createElement(tagName)**: Creates a new HTML element.
* **appendChild(node)**: Adds a new child node to an element.

1. What is JQuery? Write various JQuery Selectors.

Ans:-

* jQuery is a fast, lightweight, and feature-rich JavaScript library.
* It simplifies HTML document traversal, event handling, animation, and Ajax interactions for rapid web development.
* By providing a consistent API that works across different browsers, jQuery makes it easier to manipulate the DOM and handle events without writing complex JavaScript code.

**Key Features of jQuery:**

* **Simplified Syntax**: Makes DOM manipulation and event handling easier.
* **Cross-browser Compatibility**: Ensures code works uniformly across different browsers.
* **Plugins**: Extensible with numerous plugins for additional functionality.
* **AJAX**: Simplifies asynchronous HTTP requests and responses.

**Various jQuery Selectors:**

1. **Basic Selectors**:
   * $(selector): Selects elements based on the provided CSS selector.
     + Example: $('p') selects all <p> elements.
2. **ID Selector**:
   * $('#id'): Selects an element with a specific ID.
     + Example: $('#header') selects the element with ID header.
3. **Class Selector**:
   * $('.class'): Selects all elements with a specific class.
     + Example: $('.highlight') selects all elements with the class highlight.
4. **Attribute Selector**:
   * $('[attribute]'): Selects elements with a specific attribute.
     + Example: $('[type="text"]') selects all <input> elements with a type attribute of text.
5. **Descendant Selector**:
   * parent descendant: Selects elements that are descendants of a specified parent.
     + Example: $('ul li') selects all <li> elements inside <ul> elements.
6. **Child Selector**:
   * parent > child: Selects elements that are direct children of a specified parent.
     + Example: $('ul > li') selects all direct <li> children of <ul> elements.
7. **Pseudo-class Selectors**:
   * :first: Selects the first element in a set.
     + Example: $('li:first') selects the first <li> element.
   * :last: Selects the last element in a set.
     + Example: $('li:last') selects the last <li> element.
   * :eq(index): Selects the element at a specific index.
     + Example: $('li:eq(2)') selects the third <li> element (index starts at 0).
8. **Form Selectors**:
   * :input: Selects all form elements.
     + Example: $('input') selects all <input> elements.
   * :checked: Selects checked input elements (e.g., checkboxes, radio buttons).
     + Example: $('input:checked') selects all checked <input> elements.
9. **Hierarchy Selectors**:
   * $('ancestor descendant'): Selects elements that are descendants of a specified ancestor.
     + Example: $('div p') selects all <p> elements inside <div> elements.

**Example Usage:**

$(document).ready(function() {

// Select all paragraphs and change their color

$('p').css('color', 'blue');

// Select an element with the ID 'main' and hide it

$('#main').hide();

// Select all elements with the class 'active' and add a border

$('.active').css('border', '1px solid red');

// Select all input elements with the type 'text' and set their background color

$('[type="text"]').css('background-color', 'yellow');

});

**FAQ:**

1. Write 3 reasons why Form validations are important.

Ans:-

 **Data Integrity**:

* **Purpose**: Ensures that the data submitted by users adheres to specified formats and constraints.
* **Benefit**: Prevents incorrect or incomplete data from being stored in the database, maintaining the accuracy and quality of the information.

 **Enhanced User Experience**:

* **Purpose**: Provides immediate feedback to users about errors or missing information.
* **Benefit**: Reduces frustration by guiding users to correct mistakes before submission, leading to a smoother and more intuitive interaction with the form.

 **Security**:

* **Purpose**: Helps prevent malicious input and attacks, such as SQL injection or cross-site scripting (XSS).
* **Benefit**: Protects the application and its data by validating and sanitizing user input, reducing the risk of vulnerabilities and breaches.

1. What is jQuery Ajax?

Ans:-

**jQuery AJAX** is a feature of the jQuery library that simplifies making asynchronous HTTP requests. It allows web pages to send and retrieve data from a server without reloading the page.

**Key Points:**

* **Asynchronous Requests**: Enables background communication with the server, allowing web pages to update dynamically.
* **Simplified Syntax**: Provides a simple and easy-to-use API for sending and handling requests.
* **Common Methods**: Includes methods like $.ajax(), $.get(), and $.post() for various types of HTTP requests.

**Usage Example:**

$.ajax({

url: 'server-endpoint',

type: 'GET',

success: function(data) {

// Handle the response data

},

error: function(error) {

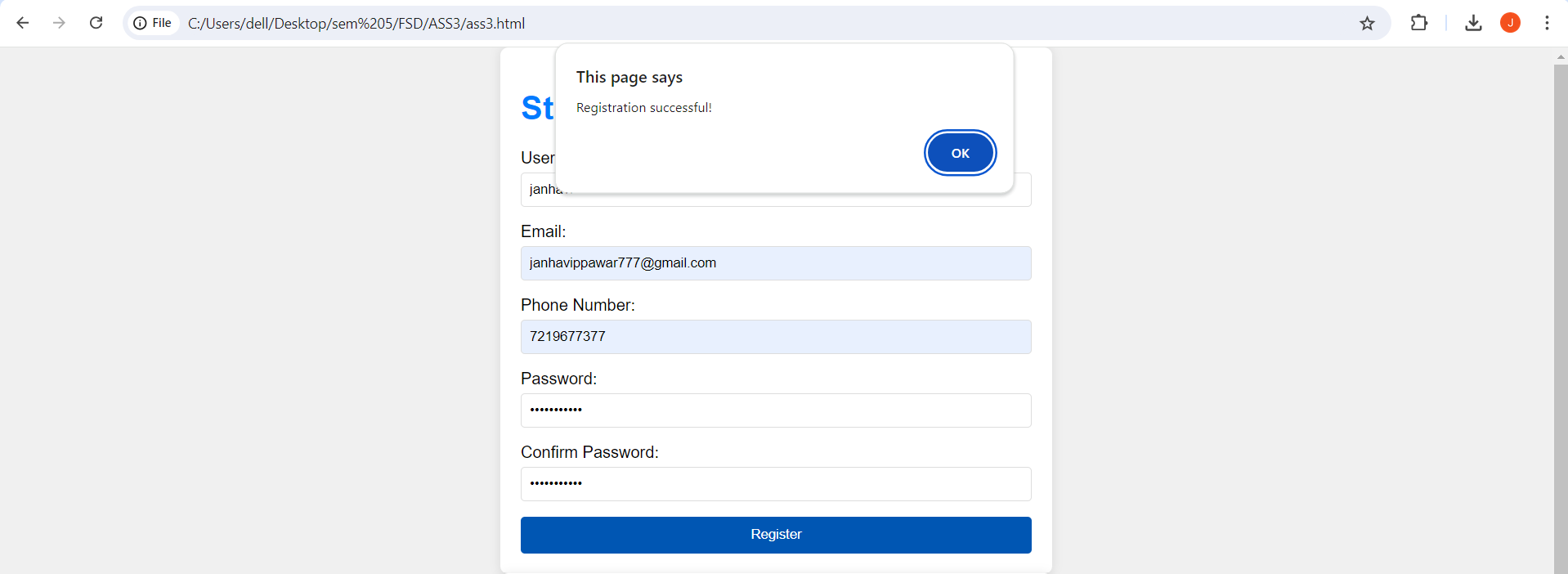
// Handle any errors

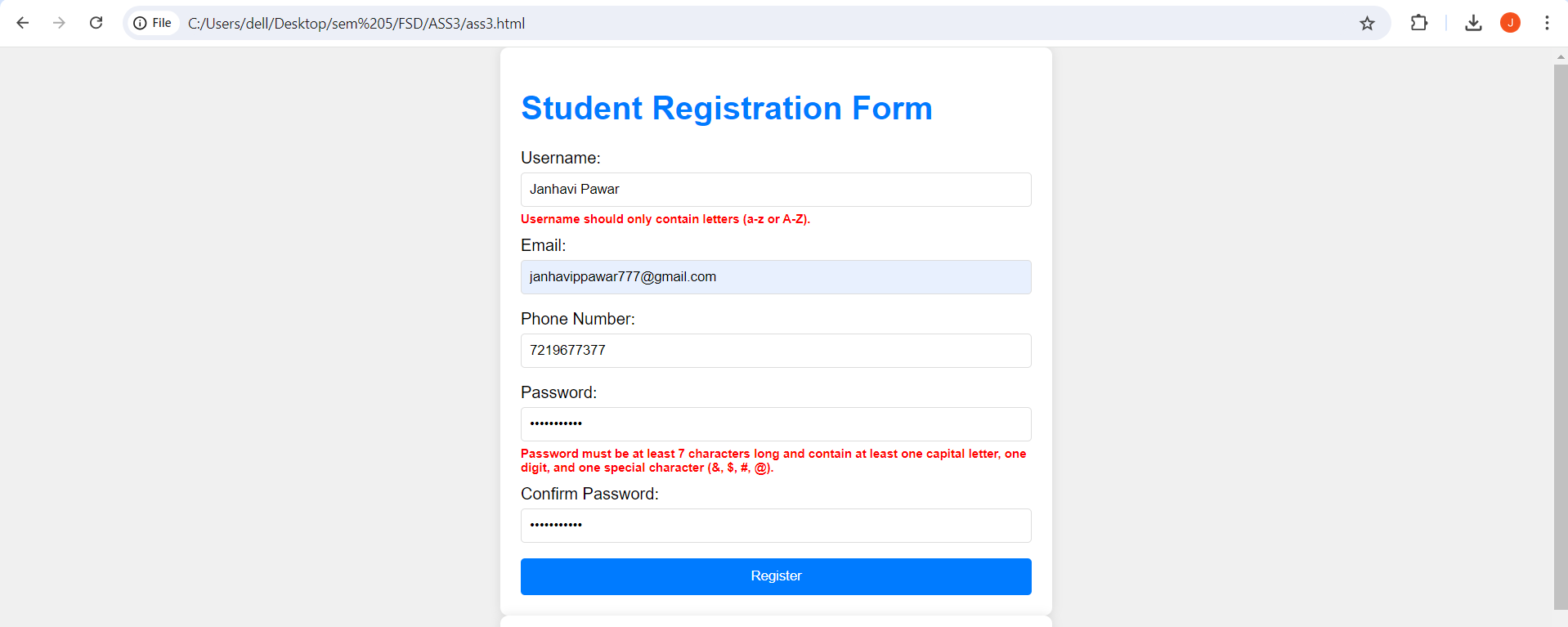
}

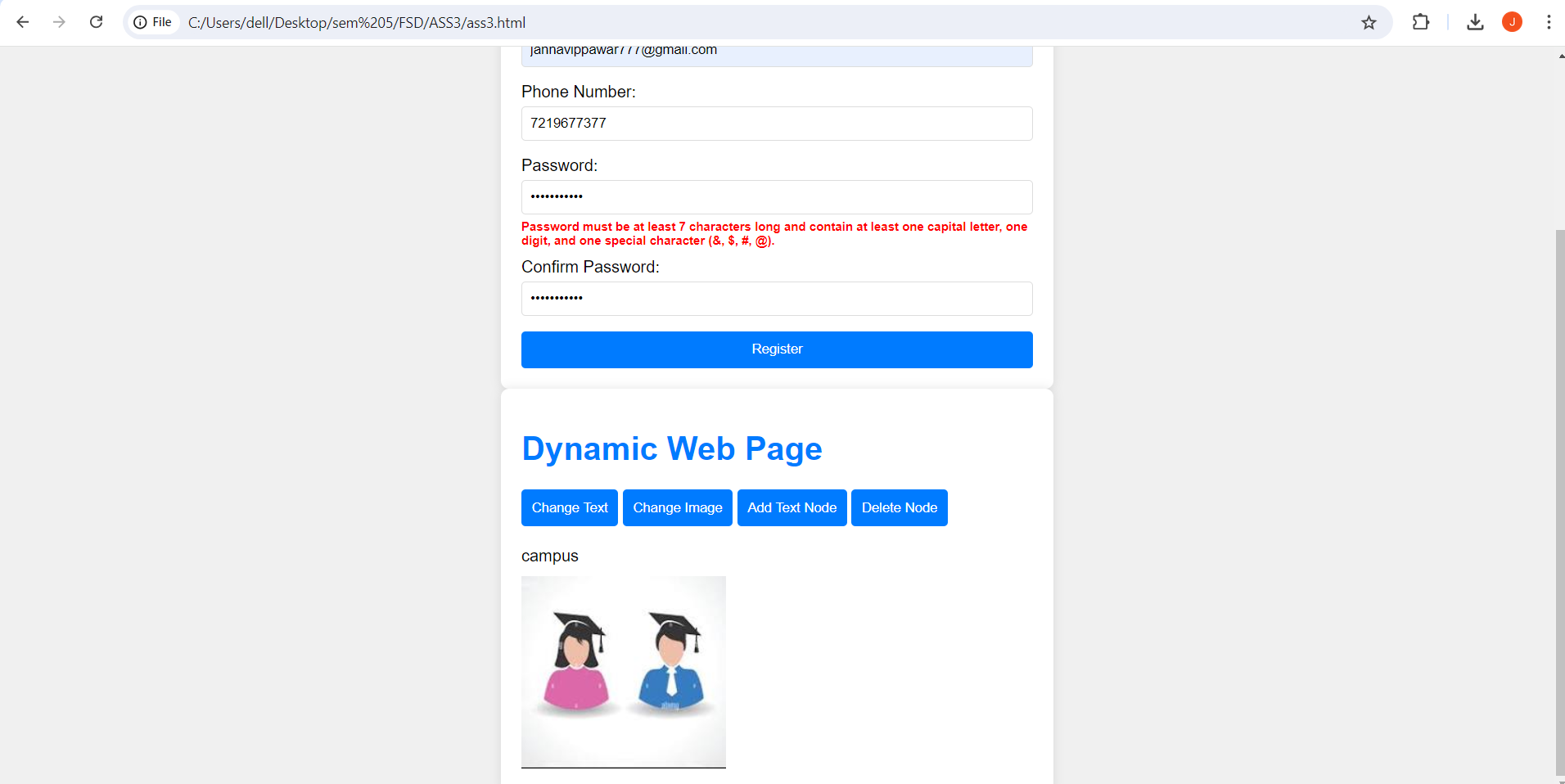
});

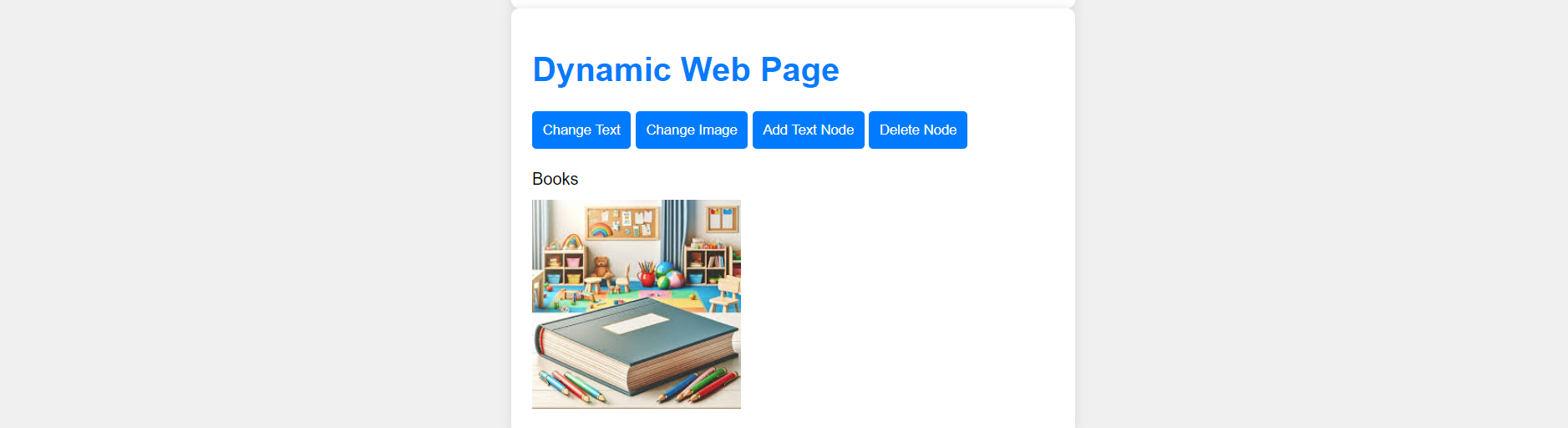
**Benefit**: Enhances user experience by allowing content updates and interactions without full page reloads.

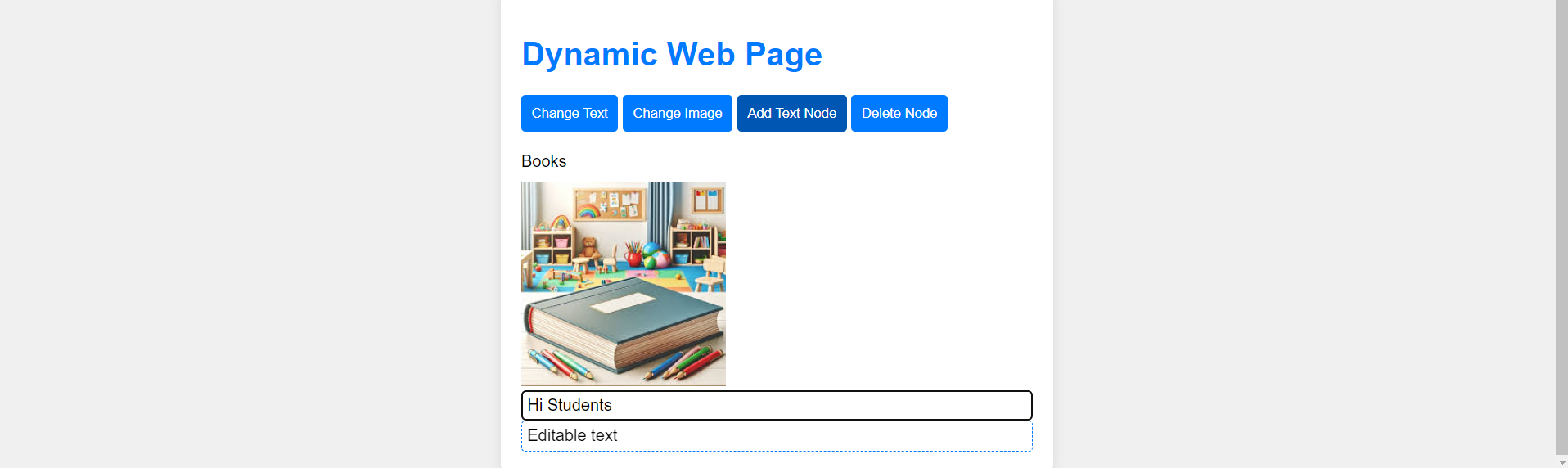
**Output: Screenshots of the output to be attached.**

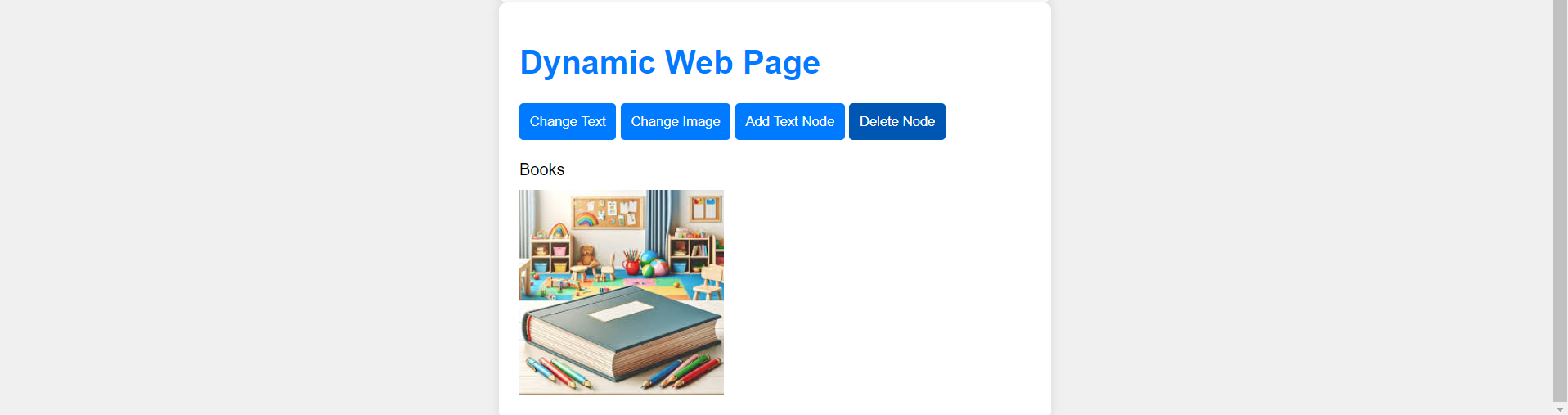
****

****

****

****

****

****

**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

* Fields should not be empty. If spaces are entered those should be considered empty
* Phone number must accept only numeric values and it should be 10 digits
* 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 (&,$,#@)
* 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, there should be three letters between @ sign and a . There must  be 3 or 2 letters after the  .   (hint: Use regular expression)

Write a client-side script with JavaScript to access and manipulate Document Object Model (DOM) objects in an HTML web page. Develop a dynamic web page using javascript and DOM. Make use of the following for accessing elements

* getElementById, getElementsByTagName,getElementsByClassName
* Change the text using innerHTML property
* Change the CSS properties like color, position of a particular element on the page
* Change the image source after clicking on a button
* Add a text node and attach it to a parent node
* Delete a node

Include jQuery to perform following operations:

* Change button text using jQuery.
* Set background-image using jQuery CSS property.
* Access HTML form data using jQuery.
* Add attribute using jQuery

Use this reference link for jQuery : https://www.w3resource.com/jquery-exercises/part1/index.php