Sign Up & Sign In Page

Name: Mohammed Waheed (Reg No: 240970144)

I. CSS Frameworks

Custom CSS

Usage: The <link rel="stylesheet" href="signup.css" /> tag is used in the HTML document's <head> section to link an external CSS file named "signup.css." This allows the styles defined in that CSS file to be applied to the HTML elements within the document.

Explanation: By linking the external stylesheet, the form inherits the styles that ensure a responsive design and consistent aesthetics. This separation of content (HTML) and presentation (CSS) not only promotes cleaner code but also allows for easier maintenance and updates to the design without altering the HTML structure**.**

II. Other Concepts Used

JavaScript Concepts

Input Validation

The JavaScript functions in this code are designed to validate user input in the registration form in real time. Each function checks a specific input field for compliance with predefined rules, such as format and completeness using regular expression, and provides immediate feedback to the user.

```
const phonePattern = /^\d{10}$/;
        if (!phonePattern.test(phone)) {
            errorElement.textContent = 'Enter a valid 10-digit phone number.';
            removeValidClass(inputElement);
            return false;
        }
```

Php Concepts

The code establishes a connection to a MySQL database using MySQLi and processes form submissions to register users. It retrieves user inputs, from the POST request. By employing prepared statements, it prevents SQL injection vulnerabilities while inserting data into the registration table. If the insertion is successful, it redirects the user to a confirmation page; otherwise, it displays any errors

The PHP code implements a login system for the by connecting to a MySQL database. It validates user credentials based on the mobile number and password entered. If the credentials are valid, the user is logged in, and session variables are set to maintain user state. Error messages are displayed for invalid inputs, ensuring a user-friendly experience.

III. Responsive Design Concepts

1.Meta Tag for Viewport

Usage: The viewport meta tag is included in the HTML to ensure the webpage adapts to different screen sizes by controlling the layout on mobile browsers.

Explanation: This tag allows the browser to adjust the page's width based on the device's screen width, enhancing the responsiveness of the layout across various devices.

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

2.CSS Media Queries

Usage: Media queries in the CSS file apply different styles based on the screen width, allowing for a responsive design that adapts to different devices.

Explanation: By using media queries, developers can create a customized layout for smaller screens, ensuring that the form and its elements remain user-friendly and visually appealing.

```
@media(max-width: 584px) {

 .container { max-width: 100%; }

 form .user-details .input-box { margin-bottom: 15px; width: 100%; }

 }
```

3.Responsive Containers

Usage: The .container class is styled using flexbox to center the form both vertically and horizontally within the viewport.

Explanation: This flexible layout approach allows the form to adapt dynamically to different screen sizes, providing a better user experience on both desktops and mobile devices.

```
body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

padding: 10px;

}
```

IV. Assistive Technology/Design Concepts

1.Accessible Form Labels

Usage: Each form input has a <label> tag associated with it to describe its function, aiding screen reader users.

Explanation: Labels with the for attribute enhance clarity by providing context for input fields, making it easier for all users, especially those with cognitive disabilities, to understand what information is required. They improve accessibility by allowing screen readers to announce the label when focused, which aids users with visual impairments in navigating forms..

HTML Code

<label for="details">Full Name</label>

<input type="text" placeholder="Enter your name" name="fname" id="fname" aria-required="true"required>

2.Keyboard Accessibility

Usage: The tabindex attribute is applied to HTML elements (such as form fields, buttons, and links) to define their focus order when users navigate using the Tab key.

Explanation: By using tabindex, developers can improve keyboard accessibility, allowing users who cannot use a mouse to navigate the webpage efficiently. The tabindex attribute allows users with mobility impairments to navigate web pages without a mouse.

HTML Code

<input type="text" placeholder="Enter your name" name="fname" id="fname" tabindex="1" aria-required="true" required>

3.Accessible Rich Internet Applications (ARIA)

3.1 **aria-required**

Usage: The aria-required attribute indicates that a specific form input must be filled out before submission. It helps assistive technologies understand which fields are mandatory.

Explanation: By using aria-required="true" on input elements, developers ensure that screen readers announce the requirement status to users. This is particularly helpful for users with visual impairments, as they will be informed of the necessity to complete certain fields.

HTML Code

<input type="text" placeholder="Enter your name" name="fname" id="fname" tabindex="1" aria-required="true" required>

3.2 **aria-describedby**

Usage: The aria-describedby attribute links an input field to additional descriptive text that provides context or guidance for users.

Explanation: When an input field has aria-describedby screen readers will read both the label of the input and the associated descriptive content when the input is focused. This ensures users receive not just the purpose of the field, but also any supplementary information that might aid them in filling it out.

<input type="password" placeholder="Enter your password" oninput="validatePassword()" aria-describedby="Enter Password with 8 characters" aria-required="true" required>

4.Reducing Motion for Users with Motion Sensitivity

Usage: The `@media (prefers-reduced-motion: reduce)` query in CSS detects users' motion preferences and disables animations and transitions.

Explanation: This guideline helps users with motion sensitivity by preventing animations that could cause discomfort or disorientation. By applying this CSS rule, the webpage provides a stable experience, allowing users to interact without triggering nausea or dizziness. For example, hover effects on containers and buttons are disabled, creating a more accessible environment.

```
@media (prefers-reduced-motion: reduce) {

    .container: hover  {  transform: none;  }
```

6.Error Messages and Validation Feedback

Usage: JavaScript functions are employed to validate form input fields and provide real-time error messages when necessary, ensuring users are alerted to any mistakes as they type.

Explanation: Real-time validation gives users immediate feedback, aiding in error correction without needing to re-submit the form. Error messages are text-based, so they can be read by screen readers.

7.High Contrast Colours

Usage: High contrast colors are implemented in the webpage design to create a strong visual distinction between text and background elements. For example, using dark text on a white background or vice versa allows users to easily read content.

Explanation: The web page employs a high-contrast colour palette ,which is advantageous for users with visual impairments or colour blindness ,as it improves the legibility of the text and content.