DOCUMENTATION

html

This file provides the structure for the user registration form. The form collects essential user information such as full name, email, password, confirmation password, and date of birth.

- **<form>**: Defines the registration form with the id="registrationForm" for targeting in JavaScript.
  - **<h1>**: A heading for the form, titled "Registration Form".
  - **<label>**: Labels for each form field, associated with their respective input elements via the for attribute.
  - **<input>**: Input fields for user data collection, including full name, email, password, confirm password, and date of birth. Each input field has a corresponding id and name attribute for identification and data submission.
  - **<span>**: Elements to display error messages next to the respective input fields. Each span has a unique id corresponding to the input field it is associated with.
  - **<button>**: A submit button for form submission, identified by id="submitButton". **<script>**: Includes the external JavaScript file (hotel.js) to handle form validation.

JavaScript Code (hotel.js)

- This file contains the JavaScript logic for validating the form fields and providing real-time feedback to the user.

DOM Elements

- **form**: The registration form element.
- **fullNameInput, emailInput, passwordInput, confirmPasswordInput, dobInput, submitButton**: These variables reference the respective input fields and submit button in the form.

Validation Functions

1. **validateFullName()**:
   - Validates that the full name is at least 3 characters long and contains only alphabetic characters and spaces.
   - Displays an error message if the input is invalid, or clears the error message if valid.
2. **validateEmail()**:
   - Checks if the email address is in a valid format using a regular expression.
   - Displays an error message if the email format is incorrect.
3. **validatePassword()**:
   - Ensures the password is at least 8 characters long and contains both letters and numbers.
   - Displays an error message if the password doesn't meet the criteria.

4. **validateConfirmPassword()**:
    - Confirms that the "Confirm Password" input matches the "Password" input.
    - Displays an error message if the passwords do not match.
5. **validateDob()**:
    - Validates that the user is at least 18 years old based on their date of birth.
    - Displays an error message if the user is underage.

Utility Functions

- **showError(elementId, message)**: Displays an error message in the specified error span element.
- **clearError(elementId)**: Clears the error message from the specified error span element.
- **validateForm()**: Calls all individual validation functions and returns true if all fields are valid, otherwise false.

Event Listeners

- **form.addEventListener('submit', ...)**: Prevents the default form submission and validates all fields before allowing submission. If validation fails, it alerts the user to correct the errors.
- **document.addEventListener('DOMContentLoaded', ...)**: Ensures that the JavaScript code runs after the DOM has fully loaded, which prevents errors related to missing elements.

Summary

- The code is designed to provide a user-friendly registration form that gives real-time feedback and ensures that all input data meets specific validation criteria before submission. The JavaScript code is organized into functions that validate individual fields, enhancing maintainability and readability. The use of event listeners ensures that validation occurs both during form submission and as the user interacts with the form, improving the overall user experience.

Submitted by,

ANINA ABRAHAM

2447109, 1 MCA A