**Software Requirements Specification (SRS) for Registration Form**

**1. Introduction**

The purpose of this document is to provide a detailed Software Requirements Specification (SRS) for the **Registration Form**. This form will allow users to register by providing essential information such as username, email, and password. Optional fields like phone number and address are also included.

**2. Scope**

The system is a web-based application that allows users to create an account by filling out a registration form. The form captures user information, validates the inputs, and submits the data to the backend for processing. It ensures that mandatory fields are completed correctly and optional fields are handled without compromising user experience.

**3. Functional Requirements**

**3.1. User Registration Form**

* **Description**: The registration form will collect user data, validate inputs, and submit the form.
* **Input Fields**:
  + Username (Mandatory)
  + Email (Mandatory)
  + Password (Mandatory)
  + Confirm Password (Mandatory)
  + Phone Number (Optional)
  + Address (Optional)

**3.2. Field Validations**

* **Username**:
  + Mandatory.
  + Must be between 6 and 50 characters.
  + Provides username suggestions (e.g., john\_doe, admin).
* **Email**:
  + Mandatory.
  + Must be a valid email format.
  + Maximum length is 255 characters.
  + Provides email suggestions (e.g., john@example.com).
* **Password**:
  + Mandatory.
  + Must be between 8 and 128 characters.
  + Must include both letters and numbers.
* **Confirm Password**:
  + Mandatory.
  + Must match the password field.
* **Phone Number**:
  + Optional.
  + Maximum length of 15 characters.
* **Address**:
  + Optional.
  + Maximum length of 300 characters.
  + Provides address suggestions.

**3.3. Error Messages**

* Display error messages next to invalid input fields.
* Red-colored error text for mandatory fields.

**3.4. Buttons**

* **Register**: Submits the form for validation and processing.
* **Reset**: Clears all fields in the form.

**3.5. Password Visibility Toggle**

* The form includes an option to show/hide the entered password and confirm password.

**3.6. Auto-Completion**

* **Username**, **Email**, and **Address** fields provide suggestions from a predefined list.

**4. Non-Functional Requirements**

**4.1. Usability**

* The form should be easy to use and provide clear error messages in case of invalid input.
* It should have a modern, clean design with intuitive navigation.

**4.2. Performance**

* The form validation should be quick and responsive.
* The form submission should not take more than 2 seconds to process.

**4.3. Security**

* Passwords should be masked by default to ensure privacy.
* Ensure that the form submits securely over HTTPS.

**4.4. Compatibility**

* The form should work across multiple devices, browsers, and screen sizes.
* Supported browsers include Chrome, Firefox, Safari, and Edge.

**5. User Interface Design**

**5.1. Layout**

* A clean, professional layout using HTML and CSS.
* The form should be centered within the page and visually appealing.
* Header with navigation links to the index and authentication home pages.

**5.2. Visual Elements**

* Use of a consistent color scheme.
  + **Header**: Dark background (#333), white text.
  + **Buttons**: Dark color for the primary button, lighter for the reset.
  + **Error Messages**: Red-colored text.

**6. System Interface**

The form is expected to interface with a backend system that handles form submission, user registration, and data persistence. This will involve an API that receives the form data, validates it further, and stores it in a database.

**7. Assumptions and Dependencies**

* The registration form is part of a larger authentication system.
* The form will use a backend API for user registration.
* Dependencies include a functioning internet connection and a compatible browser.

This SRS outlines the essential details required for developing a registration form that captures user information, validates it, and submits it securely.