

Задание: Страница регистрации с валидацией

Описание:

Создать страницу регистрации пользователя, включающую форму и скрипт для валидации введенных данных.

Валидация должна происходить в реальном времени (по мере ввода) и при финальной отправке формы.

Требования:

- HTML: Разработать форму с полями ввода для имени, логина, email, телефона, пароля и подтверждения пароля. Каждое поле должно иметь уникальный id.
- CSS: Оформить форму и поля ввода. При ошибке валидации поле должно подсвечиваться красной рамкой, а под ним должно появляться сообщение об ошибке.

JavaScript:

- Валидация полей с помощью регулярных выражений:
- Имя: Должно содержать только буквы (латиница или кириллица). Допускается использование пробелов и дефисов.
- Логин: Только буквы (латиница), цифры и символ подчёркивания (_).
- Email: Должен соответствовать стандартному формату email-адреса (например, user@domain.com).
- Телефон: Должен содержать 11 цифр. Допускаются пробелы, скобки и дефисы, которые должны игнорироваться при проверке.
- Пароль: Минимум 8 символов. Должен содержать хотя бы одну заглавную букву, одну строчную букву и одну цифру.
- Подтверждение пароля: Должно полностью совпадать с содержимым поля "Пароль".

Динамическая проверка (по желанию): При потере фокуса (blur) или вводе (input) в каждом поле должна запускаться проверка и отображаться соответствующее сообщение об ошибке, если данные некорректны.

Финальная проверка: При нажатии на кнопку "Отправить" должна происходить полная валидация всех полей. Если хотя бы одно поле не прошло проверку, отправка формы должна быть отменена, и должны быть показаны все сообщения об ошибках.

Дополнительно (по желанию):

- Добавить индикатор надёжности пароля, который меняет цвет и текст (например, "слабый", "средний", "сильный") в зависимости от сложности введённого пароля.
- Реализовать "глазик" рядом с полем пароля, который позволяет показать/скрыть введённый пароль.

Шпаргалка по регулярным выражениям (JS)

test(): Метод RegExp.prototype.test(). Возвращает true, если строка содержит совпадение с регулярным выражением, и false в противном случае.

```
const regex = /шаблон/;
```

```
if (regex.test(строка)) { ... }
```

match(): Метод String.prototype.match(). Возвращает массив со всеми найденными совпадениями или null, если совпадений нет.

```
const matches = строка.match(/шаблон/g);
```

replace(): Метод String.prototype.replace(). Возвращает новую строку, в которой некоторые или все совпадения заменены.

```
const новаяСтрока = строка.replace(/шаблон/g, 'замена');
```

Важно: Для телефона сначала используйте `replace()` для удаления всех пробелов и дефисов, а уже потом проверяйте длину и символы.