Документация проекта

**Введение:**

**Описание проекта**

Валидатор паролей - это веб-приложение, которое позволяет пользователям проверять надежность своих паролей. Проект предоставляет:

Визуальную оценку сложности пароля

Проверку соответствия стандартным требованиям безопасности

Генератор надежных паролей.

Возможность копирования пароля в буфер обмена

Целевая аудитория

Кто будет использовать ваш проект? Опишите целевую аудиторию.

**Целевая аудитория:**

Обычные пользователи, желающие проверить надежность своих паролей

Разработчики, которым нужен инструмент для проверки паролей

Администраторы систем, проверяющие политики паролей

**Установка.**

**Требования:**

Проект не требует серверной части и может работать в любом современном браузере:

Любой современный браузер (Chrome, Firefox, Safari, Edge)

Доступ к интернету (для загрузки шрифтов и иконок)Использование

**Шаги по установке:**

Создайте папку на рабочем столе

Импортируйте код ниже выделенным текстом:

//1 . Модуль элементов HTML

document.addEventListener('DOMContentLoaded', function() {

    const passwordInput = document.getElementById('pass');

    const togglePassword = document.getElementById('tpass');

    const strengthBar = document.getElementById('sb');

    const strengthText = document.getElementById('st');

    const generateBtn = document.getElementById('g-3');

    const copyBtn = document.getElementById('cbtn');

    const tooltip = document.getElementById('too');

//2. Модуль управления паролем

    // Элементы требований

    const requirements = {

        length: document.getElementById('l-r'),

        lower: document.getElementById('lr'),

        upper: document.getElementById('ur'),

        number: document.getElementById('nr'),

        special: document.getElementById('sr')

    };

    // Переключение видимости пароля

    togglePassword.addEventListener('click', function() {

        const type = passwordInput.getAttribute('type') === 'password' ? 'text' : 'password';

        passwordInput.setAttribute('type', type);

        this.innerHTML = type === 'password'

            ? '<i class="far fa-eye"></i>'

            : '<i class="far fa-eye-slash"></i>';

    });

    // Проверка пароля при вводе

    passwordInput.addEventListener('input', function() {

        const password = this.value;

        const result = checkPasswordStrength(password);

        updateStrengthIndicator(result.strength);

        updateRequirements(result.checks);

    });

    // Генерация пароля

    generateBtn.addEventListener('click', function() {

        const password = generateStrongPassword();

        passwordInput.value = password;

        passwordInput.dispatchEvent(new Event('input'));

    });

    // Копирование пароля

    copyBtn.addEventListener('click', function() {

        if (passwordInput.value) {

            passwordInput.select();

            document.execCommand('copy');

            // Показываем подсказку

            tooltip.classList.add('show');

            setTimeout(() => {

                tooltip.classList.remove('show');

            }, 2000);

        }

    });

    // Проверка сложности пароля

    function checkPasswordStrength(password) {

        const checks = {

            length: password.length >= 8,

            lower: /[a-z]/.test(password),

            upper: /[A-Z]/.test(password),

            number: /\d/.test(password),

            special: /[!@#$%^&\*]/.test(password)

        };

        let strength = 0;

        // За длину (максимум 50 баллов)

        strength += Math.min(password.length \* 5, 50);

        // За разнообразие символов

        const fulfilledChecks = Object.values(checks).filter(Boolean).length;

        strength += fulfilledChecks \* 10;

        // Штраф за повторяющиеся символы

        if (/(.)\1\1/.test(password)) {

            strength -= 20;

        }

        // Проверка на слабые пароли

        const weakPasswords = ['123456', 'password', 'qwerty', '111111'];

        if (weakPasswords.includes(password.toLowerCase())) {

            strength = Math.min(strength, 20);

        }

        return {

            strength: Math.max(0, Math.min(100, strength)),

            checks

        };

    }

// 3. Модуль обновления интерфейса

    // Обновление индикатора сложности

    function updateStrengthIndicator(strength) {

        strengthBar.style.width = `${strength}%`;

        if (strength < 40) {

            strengthBar.style.backgroundColor = 'var(--danger)';

            strengthText.textContent = 'Слабый пароль';

            strengthText.style.color = 'var(--danger)';

        } else if (strength < 70) {

            strengthBar.style.backgroundColor = 'var(--warning)';

            strengthText.textContent = 'Средний пароль';

            strengthText.style.color = 'var(--warning)';

        } else {

            strengthBar.style.backgroundColor = 'var(--success)';

            strengthText.textContent = 'Надежный пароль';

            strengthText.style.color = 'var(--success)';

        }

    }

    // Обновление отображения требований

    function updateRequirements(checks) {

        for (const [key, isValid] of Object.entries(checks)) {

            const icon = requirements[key].querySelector('i');

            if (isValid) {

                icon.classList.remove('fa-circle', 'far');

                icon.classList.add('fa-check', 'fas', 'valid');

                requirements[key].style.color = 'var(--dark)';

            } else {

                icon.classList.remove('fa-check', 'fas', 'valid');

                icon.classList.add('fa-circle', 'far');

                requirements[key].style.color = '#858796';

            }

        }

    }

//4. Основной модуль

    // Генерация надежного пароля

    function generateStrongPassword() {

        const length = 12;

        const lowercase = 'abcdefghjkmnpqrstuvwxyz';

        const uppercase = 'ABCDEFGHJKLMNPQRSTUVWXYZ';

        const numbers = '23456789';

        const symbols = '!@#$%^&\*';

        let charset = '';

        let password = '';

        // Добавляем хотя бы по одному символу из каждой категории

        password += getRandomChar(lowercase);

        password += getRandomChar(uppercase);

        password += getRandomChar(numbers);

        password += getRandomChar(symbols);

        // Формируем полный набор символов

        charset = lowercase + uppercase + numbers + symbols;

        // Добираем остаток пароля

        for (let i = password.length; i < length; i++) {

            password += getRandomChar(charset);

        }

        // Перемешиваем символы

        return shuffleString(password);

    }

    // Получение случайного символа из строки

    function getRandomChar(string) {

        const randomValues = new Uint32Array(1);

        window.crypto.getRandomValues(randomValues);

        return string[randomValues[0] % string.length];

    }

    // Перемешивание строки

    function shuffleString(string) {

        const array = string.split('');

        for (let i = array.length - 1; i > 0; i--) {

            const j = Math.floor(Math.random() \* (i + 1));

            [array[i], array[j]] = [array[j], array[i]];

        }

        return array.join('');

    }

});

Затем создайте файл style.css и вставьте этот код:

:root{

    --primary: #4e73df;

    --success: #1cc88a;

    --warning: #f6c23e;

    --danger: #e74a3b;

    --dark: #5a5c69;

    --light: #f8f9fc;

}

\*{

    box-sizing: border-box;

    margin: 0;

    padding: 0;

    font-family: 'Nunito', sans-serif;

}

body{

    background: linear-gradient(135deg, #f8f9fc 0%, #e3e6f0 100%);

    display: flex;

    justify-content: center;

    align-items: center;

    min-height: 100vh;

    padding: 20px;

}

.m-1{

    background-color: white;

    border-radius: 15px;

    box-shadow: 0 10px 35px rgba(0, 0, 0, 0.08);

    padding: 40px;

    width: 100%;

    max-width: 500px;

    transition: all 0.3s ease;

}

.m-1:hover{

    transform: translateY(-5px);

    box-shadow: 0 15px 40px rgba(0, 0, 0, 0.12);

}

h1{

    color: var(--dark);

    text-align: center;

    margin-bottom: 30px;

    font-weight: 700;

    font-size: 28px;

}

.in-gr{

    margin-bottom: 25px;

    position: relative;

}

label {

    display: block;

    margin-bottom: 10px;

    font-weight: 600;

    color: var(--dark);

    font-size: 16px;

}

.pass-w {

    position: relative;

}

input{

    width: 100%;

    padding: 15px 20px;

    border: 2px solid #d1d3e2;

    border-radius: 10px;

    font-size: 16px;

    transition: all 0.3s;

    background-color: var(--light);

}

input:focus {

    outline: none;

    border-color: var(--primary);

    box-shadow: 0 0 0 3px rgba(78, 115, 223, 0.25);

}

.t-pass {

    position: absolute;

    right: 15px;

    top: 50%;

    transform: translateY(-50%);

    background: none;

    border: none;

    cursor: pointer;

    color: #b7b9cc;

    font-size: 18px;

    transition: color 0.3s;

}

.t-pass:hover {

    color: var(--primary);

}

.s-m {

    height: 10px;

    background-color: #eaecf4;

    border-radius: 5px;

    margin: 20px 0;

    overflow: hidden;

}

.s-b {

    height: 100%;

    width: 0%;

    transition: all 0.5s cubic-bezier(0.65, 0, 0.35, 1);

    border-radius: 4px;

}

.s-t {

    font-size: 15px;

    font-weight: 600;

    text-align: center;

    margin-bottom: 25px;

    height: 20px;

}

.r {

    margin-bottom: 30px;

}

.r-0 {

    display: flex;

    align-items: center;

    margin-bottom: 12px;

    font-size: 14px;

    color: #858796;

}

.r-0 i {

    margin-right: 10px;

    font-size: 14px;

    width: 20px;

    text-align: center;

}

.valid {

    color: var(--success);

}

.invalid {

    color: #d1d3e2;

}

.b-t {

    display: flex;

    gap: 15px;

}

button {

    flex: 1;

    padding: 15px;

    border: none;

    border-radius: 10px;

    font-weight: 600;

    cursor: pointer;

    transition: all 0.3s;

    font-size: 16px;

    display: flex;

    align-items: center;

    justify-content: center;

    gap: 8px;

}

.g-btn {

    background-color: var(--primary);

    color: white;

}

.g-btn:hover {

    background-color: #3a5bd9;

    transform: translateY(-2px);

}

.c-btn {

    background-color: #e3e6f0;

    color: var(--dark);

}

.c-btn:hover {

    background-color: #d7dae7;

    transform: translateY(-2px);

}

.too {

    position: absolute;

    top: -40px;

    right: 0;

    background-color: var(--dark);

    color: white;

    padding: 8px 12px;

    border-radius: 6px;

    font-size: 13px;

    opacity: 0;

    transition: opacity 0.3s;

    pointer-events: none;

}

.too.show {

    opacity: 1;

}

@media (max-width: 576px) {

    .m-1{

        padding: 30px 20px;

    }

    h1{

        font-size: 24px;

    }

    .b-t{

        flex-direction: column;

    }

}

Затем HTML код

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

    <link rel="stylesheet" href="css/style.css">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

    <script src="js/main.js" defer></script>

    <title>Проверка надежности пароля</title>

</head>

<body>

    <div class="m-1">

        <h1>Проверка надежности пароля</h1>

        <div class="in-gr">

            <label for="pass">Ваш пароль</label>

            <div class="pass-w">

                <input type="password" id="pass" placeholder="Введите пароль">

                <button class="t-pass" id="tpass">

                    <i class="far fa-eye"></i>

                </button>

            </div>

        </div>

        <div class="s-m">

            <div class="s-b" id="sb"></div>

        </div>

        <div class="s-t" id="st">Введите пароль для проверки</div>

        <div class="r">

            <div class="r-0" id="l-r">

                <i class="far fa-circle"></i>

                <span>Минимум 8 символов</span>

            </div>

            <div class="r-0" id="lr">

                <i class="far fa-circle"></i>

                <span>Строчные буквы (a-z)</span>

            </div>

            <div class="r-0" id="ur">

                <i class="far fa-circle"></i>

                <span>Заглавные буквы (A-Z)</span>

            </div>

            <div class="r-0" id="nr">

                <i class="far fa-circle"></i>

                <span>Цифры (0-9)</span>

            </div>

            <div class="r-0" id="sr">

                <i class="far fa-circle"></i>

                <span>Спецсимволы (!@#$)</span>

            </div>

        </div>

        <div class="b-t">

            <button class="g-btn" id="g-3">

                <i class="fas fa-key"></i>

                Сгенерировать

            </button>

            <button class="c-btn" id="cbtn">

                <i class="far fa-copy"></i>

                Копировать

            </button>

            <div class="too" id="too">Скопировано!</div>

        </div>

    </div>

</body>

</html>

Скопируйте HTML-код в файл с расширением .html, затем установите файл main.js и styless.css.

Откройте файл в браузере

**Использование**

Основные функции

Проверка пароля:

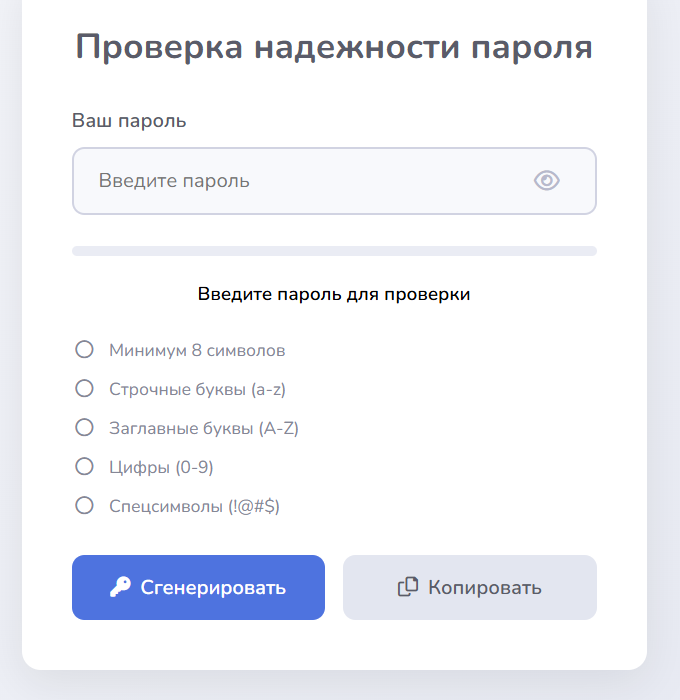
Введите пароль в поле ввода

Система автоматически оценит его надежность

Проверит соответствие требованиям

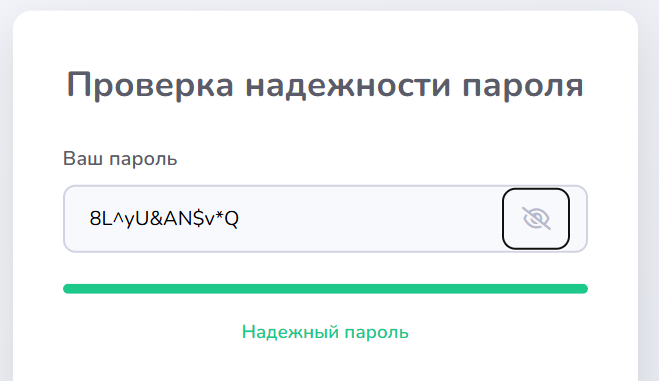
Генерация пароля:

Нажмите кнопку "Сгенерировать"

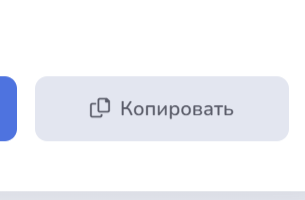


В поле появится надежный пароль

Пароль автоматически проверяется



Копирование пароля:



Нажмите кнопку "Копировать"

Пароль будет скопирован в буфер обмена

Появится уведомление об успешном копировании

**Архитектура**

Структура проекта:

Проект состоит из одного HTML-файла, содержащего:

HTML-разметку - структура интерфейса

CSS-стили - оформление элементов

JavaScript-код - логика работы валидатора

**Основные компоненты**

Поле ввода пароля с возможностью показать/скрыть пароль

Индикатор сложности (полоса и текстовая оценка)

Список требований с визуализацией выполнения

Кнопки управления (генерация, копирование)

**Вклад**

Как внести вклад:

Сообщайте об ошибках через Issues

Предлагайте улучшения интерфейса

Добавляйте новые функции проверки паролей

**Лицензия**

Проект распространяется под лицензией MIT.

**Авторы:** Овчинников Максим, Дон Вадим, Мусаев Арсений