

Usuario:

Crea una aplicación web para generar contraseñas seguras. La aplicación debe incluir las siguientes funcionalidades:

1. Permitir al usuario seleccionar:
 - Longitud de la contraseña (entre 8 y 20 caracteres).
 - Si desea incluir letras mayúsculas, letras minúsculas, números y caracteres especiales.
2. Generar una contraseña segura basada en las preferencias seleccionadas.
3. Mostrar la contraseña generada en pantalla.
4. Incluir un botón para copiar la contraseña generada al portapapeles.
5. Diseñar una interfaz moderna y responsiva utilizando Tailwind CSS. Asegúrate de que sea fácil de usar, con botones y controles bien visibles.
6. Escribir todo el código en un solo archivo (HTML, CSS y JavaScript).
7. Cargar Tailwind CSS desde un CDN para simplificar el diseño.
8. Proporcionar instrucciones claras para ejecutar la aplicación:
 - El archivo debe ser funcional al abrirlo en un navegador haciendo doble clic.

Asistente:

Aquí tienes el código completo de la aplicación web para generar contraseñas seguras, diseñado en un único archivo HTML con Tailwind CSS cargado desde un CDN. Simplemente guarda este código en un archivo con extensión `.html` y ábrelo en cualquier navegador para usarlo:

Código:

```
<!DOCTYPE html>

<html lang="en">

<head>
```

```
<meta charset="UTF-8">

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

<title>Password Generator</title>

<script src="https://cdn.tailwindcss.com"></script>

</head>

<body class="bg-gray-100 flex items-center justify-center min-h-screen">

  <div class="bg-white shadow-lg rounded-lg p-6 max-w-md w-full">

    <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">Secure Password
Generator</h1>

    <form id="password-form" class="space-y-4">

      <div>

        <label for="length" class="block text-sm font-medium
text-gray-700">Password Length</label>

        <input type="number" id="length" name="length" min="8" max="20"
value="12" class="mt-1 block w-full p-2 border border-gray-300 rounded-md shadow-sm
focus:ring-blue-500 focus:border-blue-500" required>

      </div>

      <div class="flex items-center">

        <input type="checkbox" id="uppercase" name="uppercase" class="h-4 w-4
text-blue-600 border-gray-300 rounded">

        <label for="uppercase" class="ml-2 text-sm text-gray-700">Include
Uppercase Letters</label>

      </div>

      <div class="flex items-center">

        <input type="checkbox" id="lowercase" name="lowercase" class="h-4 w-4
text-blue-600 border-gray-300 rounded" checked>
```

```
        <label for="lowercase" class="ml-2 text-sm text-gray-700">Include
Lowercase Letters</label>

    </div>

    <div class="flex items-center">

        <input type="checkbox" id="numbers" name="numbers" class="h-4 w-4
text-blue-600 border-gray-300 rounded" checked>

            <label for="numbers" class="ml-2 text-sm text-gray-700">Include
Numbers</label>

        </div>

        <div class="flex items-center">

            <input type="checkbox" id="special" name="special" class="h-4 w-4
text-blue-600 border-gray-300 rounded">

                <label for="special" class="ml-2 text-sm text-gray-700">Include Special
Characters</label>

            </div>

            <button type="button" id="generate" class="w-full bg-blue-600 text-white
py-2 px-4 rounded-md hover:bg-blue-700">Generate Password</button>

        </form>

        <div id="result-container" class="mt-4 hidden">

            <p class="text-sm text-gray-700">Generated Password:</p>

            <div class="flex items-center space-x-2 mt-1">

                <input type="text" id="result" readonly class="w-full p-2 border
border-gray-300 rounded-md focus:outline-none">

                    <button id="copy" class="bg-green-600 text-white py-2 px-4 rounded-md
hover:bg-green-700">Copy</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script>
```

```
document.getElementById('generate').addEventListener('click', () => {

    const length = parseInt(document.getElementById('length').value);

    const includeUppercase = document.getElementById('uppercase').checked;
    const includeLowercase = document.getElementById('lowercase').checked;
    const includeNumbers = document.getElementById('numbers').checked;
    const includeSpecial = document.getElementById('special').checked;

    const uppercaseChars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    const lowercaseChars = 'abcdefghijklmnopqrstuvwxyz';
    const numberChars = '0123456789';
    const specialChars = '!@#$%^&*()-_+=[]{}|;:,.<>?';

    let charPool = '';

    if (includeUppercase) charPool += uppercaseChars;
    if (includeLowercase) charPool += lowercaseChars;
    if (includeNumbers) charPool += numberChars;
    if (includeSpecial) charPool += specialChars;

    if (charPool === '') {
        alert('Please select at least one character type.');
```

```
        return;
    }
}
```

```
let password = '';

for (let i = 0; i < length; i++) {

    const randomIndex = Math.floor(Math.random() * charPool.length);

    password += charPool[randomIndex];

}

document.getElementById('result').value = password;

document.getElementById('result-container').classList.remove('hidden');

});
```

```
document.getElementById('copy').addEventListener('click', () => {

    const passwordField = document.getElementById('result');

    passwordField.select();

    document.execCommand('copy');

    alert('Password copied to clipboard!');

});
```

```
</script>
```

```
</body>
```

```
</html>
```