```tsx
import React, { useState } from 'react';

interface PasswordStrength {

length: boolean;

uppercase: boolean;

lowercase: boolean;

number: boolean;

specialChar: boolean;

}

const PasswordStrengthChecker = () => {
```

```
const [password, setPassword] = useState('');

const [passwordStrength, setPasswordStrength] = useState<PasswordStrength>({

  length: false,

  uppercase: false,

  lowercase: false,

  number: false,

  specialChar: false,

});


const checkPasswordStrength = (password: string) => {

  const strength: PasswordStrength = {
```

```
      length: password.length >= 8,

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

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

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

      specialChar: /[^A-Za-z0-9]/.test(password),

    };

    setPasswordStrength(strength);

};


  const handlePasswordChange = (e: React.ChangeEvent<HTMLInputElement>) => {

    const password = e.target.value;
```

```javascript
    setPassword(password);

    checkPasswordStrength(password);

  };


  const getStrengthColor = () => {

    const { length, uppercase, lowercase, number, specialChar } = passwordStrength;

    if (length && uppercase && lowercase && number && specialChar) {

      return 'text-green-500';

    } else if (length && (uppercase || lowercase) && (number || specialChar)) {

      return 'text-yellow-500';

    } else {
```

```jsx
    return 'text-red-500';

  }

};


  return (

    <div className="max-w-md mx-auto p-4 mt-10 bg-white rounded-lg shadow-md">

      <h2 className="text-lg font-bold mb-4">Password Strength Checker</h2>

      <input

        type="password"

        value={password}

        onChange={handlePasswordChange}
```

```jsx
        className="w-full p-2 mb-4 border border-gray-300 rounded-lg focus:outline-none focus:ring focus:ring-blue-500"

        placeholder="Enter your password"

      />

      <div className="mb-4">

        <p className="text-sm font-bold">Password Strength:</p>

        <ul>

          <li className={`text-sm ${passwordStrength.length ? 'text-green-500' : 'text-red-500'}`}>

            Length: {passwordStrength.length ? 'Strong' : 'Weak'}

          </li>

          <li className={`text-sm ${passwordStrength.uppercase ? 'text-green-500' : 'text-red-500'}`}>

            Uppercase: {passwordStrength.uppercase ? 'Present' : 'Absent'}
```

```
        </li>

        <li className={`text-sm ${passwordStrength.lowercase ? 'text-green-500' : 'text-red-500'}`}>

            Lowercase: {passwordStrength.lowercase ? 'Present' : 'Absent'}

        </li>

        <li className={`text-sm ${passwordStrength.number ? 'text-green-500' : 'text-red-500'}`}>

            Number: {passwordStrength.number ? 'Present' : 'Absent'}

        </li>

        <li className={`text-sm ${passwordStrength.specialChar ? 'text-green-500' : 'text-red-500'}`}>

            Special Character: {passwordStrength.specialChar ? 'Present' : 'Absent'}

        </li>

    </ul>
```

```
        </div>

        <p className={`text-lg font-bold ${getStrengthColor()}`}>

          Overall Strength: {getStrengthColor() === 'text-green-500' ? 'Strong' : getStrengthColor() === 'text-yellow-500' ? 'Medium' : 'Weak'}

        </p>

      </div>

  );

};



export default PasswordStrengthChecker;
```