

DAY 6

Note: The names for my HTML, CSS, and JavaScript files (like DAYTHREE.html, DAYTHREE.css, DAYTHREE.js) are **not recommended** as they are considered **bad practice** in professional projects. However, since this is a **mini project**, I prefer using these names to **distinguish them from my other files**. For **larger or professional projects**, it's better to use standard and descriptive file names like index.html, style.css, and script.js.

DAYSIX.HTML

❖ `<div class="group"> ... </div>`

This is the main container that holds all your content.

❖ `<h1 id="heading">DAY 6</h1>`

A heading saying "DAY 6".

❖ `<h1>Password Strength Checker</h1>`

Main heading. The word "**Strength**" is given a special style using the `` tag (you'll color it differently in CSS).

❖ `<label for="password">Password</label>`

A label telling the user to enter their password.

❖ `<div class="input-container"> ... </div>`

A box containing:

- The **password input field**
- The **eye icon** (☐) to show/hide the password

❖ `<input type="password" id="password" ... />`

This is the actual **password box**.

❖ ` ☐ `

This is the **eye icon** the user can click to toggle visibility of the password.

❖ `<label>Strength of password</label>`

Another label — telling the user the next box shows **how strong their password is**.

❖ `<div class="slide"> <div id="point"></div> </div>`

This is the **password strength bar**:

- slide is the outer bar (gray background)
- point is the inner bar that grows in width and changes color based on password strength

DAYSIX.CSS

Styling the whole page

```
body {  
  margin: 0;  
  font-family: monospace;  
  min-height: 100vh;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  background-color: rgb(5, 52, 50);  
  font-size: 20px;  
  color: white;  
}
```

- **margin: 0** → Removes default browser space around the page.
- **font-family: monospace** → All text uses a fixed-width font.
- **min-height: 100vh** → Page height is always at least full screen.
- **display: flex; justify-content: center; align-items: center;** → Centers everything in the middle of the screen.
- **background-color** → Dark green background.
- **font-size and color** → Base text size and color (white).

Styling the "DAY 6" heading

```
#heading {  
  color: white;  
  text-decoration: underline;  
}
```

- Makes the text **white** and **underlined**.

All main headings

```
h1 {  
  font-family: 'Times New Roman';  
  color: white;  
}
```

- All `<h1>` elements use a serif font and are white.

Only the word "STRENGTH"

```
#highlight {  
  color: darkorange;  
}
```

- Makes just the word **STRENGTH** in your heading orange for emphasis.

Wrapper for all content

```
.group {  
  text-align: center;  
}
```

- Centers all inner content (like text and labels).

Text above inputs

```
.group label {  
  display: block;  
  padding: 20px 0 10px;  
  color: white;  
}
```

- Labels take up the whole line (block).
- Adds spacing **above and below** the text.
- Keeps the color white.

Box that holds the password field and eye icon

```
.input-container {  
  position: relative;  
  width: 100%;  
}
```

- Sets up **relative positioning** so the icon can be placed inside.


Password field

```
.input-container input {  
  width: 100%;  
  padding: 15px 15px 15px 15px;  
  font-size: 18px;  
  border: none;  
  border-radius: 10px;  
  background-color: #eaeff2;  
  color: #3ba62f;  
  outline: none;  
}
```

- Full-width password box with padding inside.
- No border and rounded corners.
- Light gray background.
- Green text inside (#3ba62f).
- No outline when clicked.

Eye icon to show/hide password

```
.input-container span {
  position: absolute;
  right: 0px;
  top: 50%;
  transform: translateY(-50%);
  cursor: pointer;
  font-size: 18px;
}
```

- The  icon is **placed inside the input box**, on the right.
- cursor: pointer makes it clickable.

Outer password strength bar

```
.slide {
  background-color: #2E424D;
  width: 100%;
  height: 15px;
  border-radius: 5px;
  margin-top: 10px;
}
```

- Gray background bar to hold the strength meter.
- Rounded and full-width.

The colored strength level inside the bar

```
#point {
  background-color: #D73F40;
  width: 1%;
  height: 100%;
  border-radius: 5px;
  transition: 0.5s ease;
}
```

- Starts small (1% width), grows with stronger passwords.
 - Color and size change dynamically (via JavaScript).
 - Transition makes the bar **grow smoothly**.
-

The code has **two main purposes**:

1. To **check the strength of the password** as the user types.
2. To **show or hide the password** using the ☐ icon.

1. These lines get HTML elements:

```
let password = document.getElementById("password");  
let power = document.getElementById("point");  
let toggle = document.getElementById("togglePwd");
```

- password = refers to the password input field.
- power = refers to the strength bar (the colored one inside the grey bar).
- toggle = refers to the ☐ icon used to show/hide the password.

2. When the user types in the password box:

```
password.oninput = function () {
```

- This **runs the function every time** you type or change something in the password box.

3. Setup points, colors, and bar widths:

```
let point = 0;  
let value = password.value;  
let widthPower = ["1%", "25%", "50%", "75%", "100%"];  
let colorPower = ["#D73F40", "#DC6551", "#F2B84F", "#BDE952", "#3ba62f"];
```

- point = 0; → Starts with 0 strength.
- value stores the current password value.
- widthPower → Different bar sizes for different strengths.
- colorPower → Different colors to show weak ☐ to strong ☒ passwords.

4. Check if password is long enough:

```
if (value.length >= 6) {
```

- Only checks strength **if password is at least 6 characters long**.

5. Test the password using 4 checks:

```
let checks = /[0-9]/, /[a-z]/, /[A-Z]/, /^[^0-9a-zA-Z]/;  
checks.forEach((regex) => {  
  if (regex.test(value)) point++;  
});
```

- It uses **4 rules** to check your password:
 1. Has a **number**? (0-9)



- 2. Has a **small letter**? (a-z)
- 3. Has a **capital letter**? (A-Z)
- 4. Has a **symbol**? (@, #, \$, !, etc.)
- For **each match**, it adds 1 point. So max = 4 points.

6. Update the strength bar:

```
power.style.width = widthPower[point];
power.style.backgroundColor = colorPower[point];
};
```


- Based on how strong the password is, it:
 - Changes the **length of the bar**.
 - Changes the **color of the bar**.

Examples:

- 0 points → Red, small bar 
- 4 points → Green, full bar 



7. Show or hide the password when clicking

```
toggle.onclick = function () {
```

- When the user clicks the  icon, this part runs.


8. If it's hidden, show it:

```
if (password.type === "password") {
  password.type = "text";
  toggle.textContent = "👁";
  toggle.title = "Hide Password";
```

- If the password is **currently hidden**, it will:
 - Change input to **visible** (type = text).
 - Change icon from  to .
 - Change tooltip text to say "**Hide Password**".

9. If it's visible, hide it again:

```
} else {
  password.type = "password";
  toggle.textContent = "🔒";
  toggle.title = "Show Password";
}
};
```

- If it was **already visible**, it goes back to:
 - **Hidden password**
 - Icon goes back to 

- Tooltip becomes "Show Password"