

Lesson 07 Demo 01

Styling Essential Elements Using Tailwind

Objectives: To style and configure essential UI elements, including buttons, dropdowns, badges, spinners, and progress bars, using Tailwind CSS utility classes, ensuring a responsive design and smooth user interaction

Tools required: Visual Studio Code, Web browser, Tailwind CSS

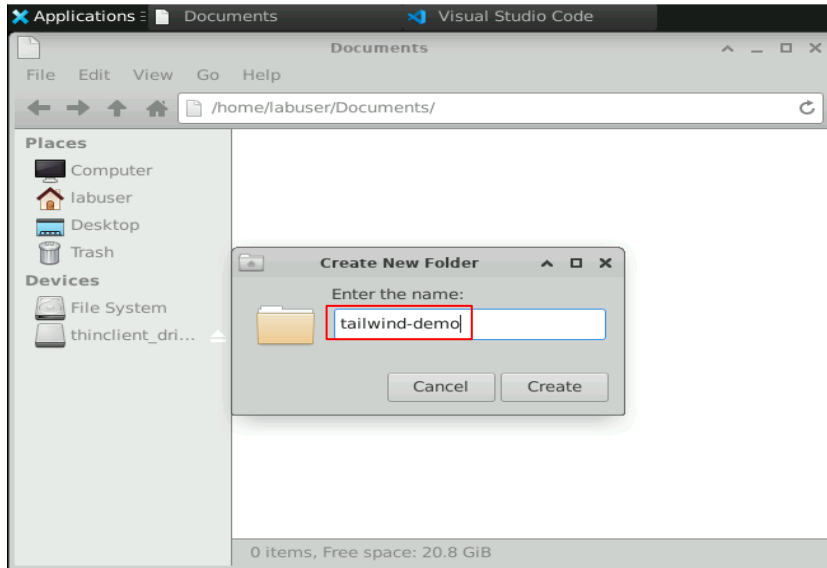
Prerequisites: NA

Steps to be followed:

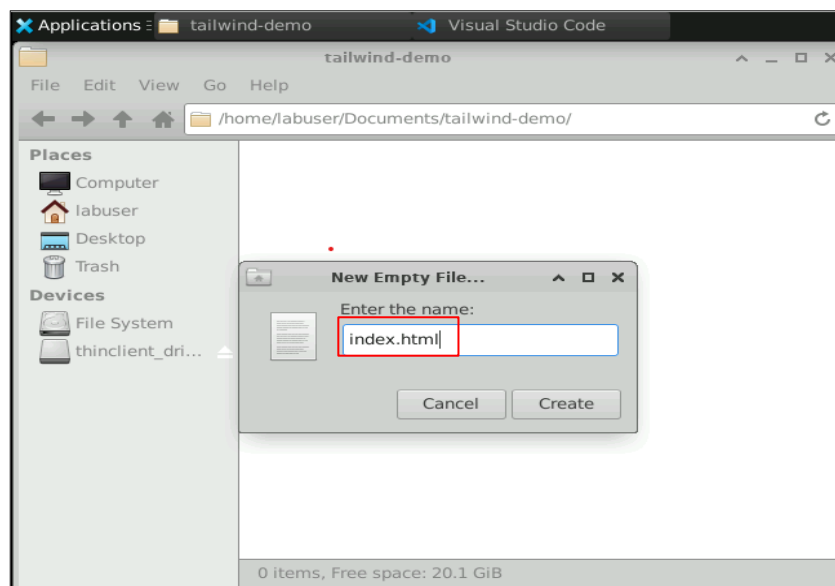
1. Set up the project folder and base HTML structure
2. Create and style the button components
3. Create a dropdown menu with three selectable options
4. Design badge elements
5. Add a loading spinner
6. Build a progress bar
7. Run the implementation in a web browser

Step 1: Set up the project folder and base HTML structure

1.1 Open Visual Studio Code and create a new project folder named **tailwind-demo**

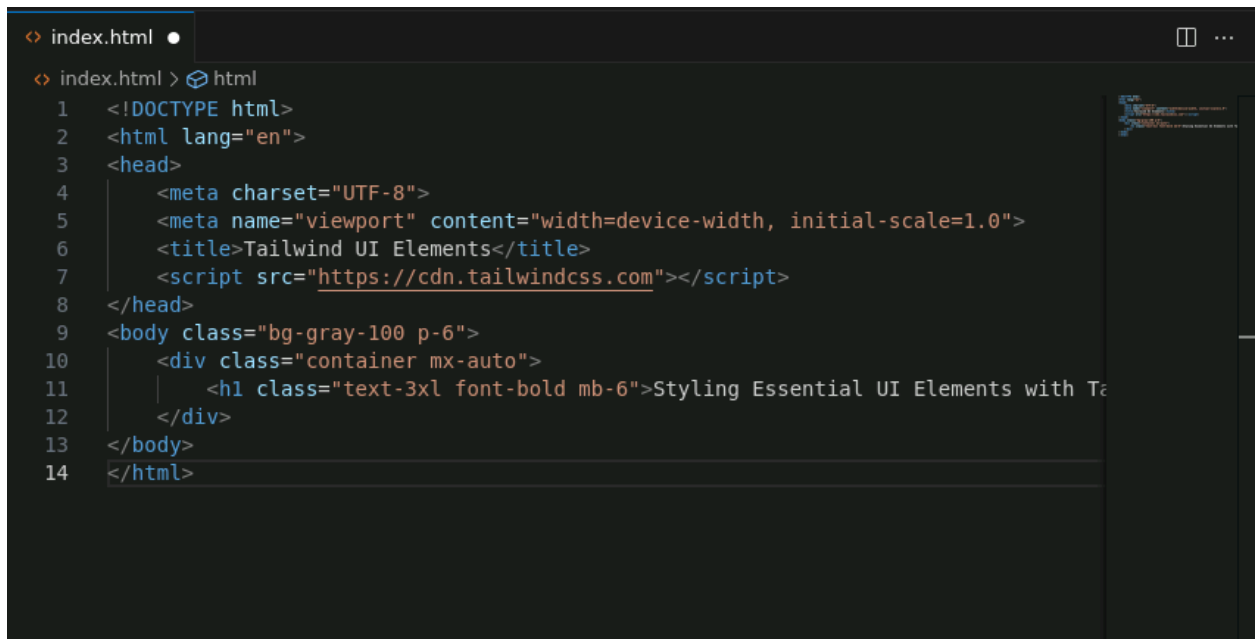


1.2 Inside this folder, create a file named **index.html**



1.3 Add the following boilerplate code to the index.html file:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tailwind UI Elements</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 p-6">
  <div class="container mx-auto">
<h1 class="text-3xl font-bold mb-6">Styling Essential UI Elements with Tailwind</h1>
  </div>
</body>
</html>
```

A screenshot of a code editor with a dark theme. The editor shows a file named 'index.html' with 14 lines of HTML boilerplate code. The code is color-coded: HTML tags are in blue, attributes and values are in orange, and text content is in white. The code matches the text provided in the previous block. On the right side of the editor, there is a small preview window showing a snippet of the rendered HTML, including the h1 tag and its content. The editor interface includes a tab at the top labeled 'index.html' and a breadcrumb 'index.html > html'.

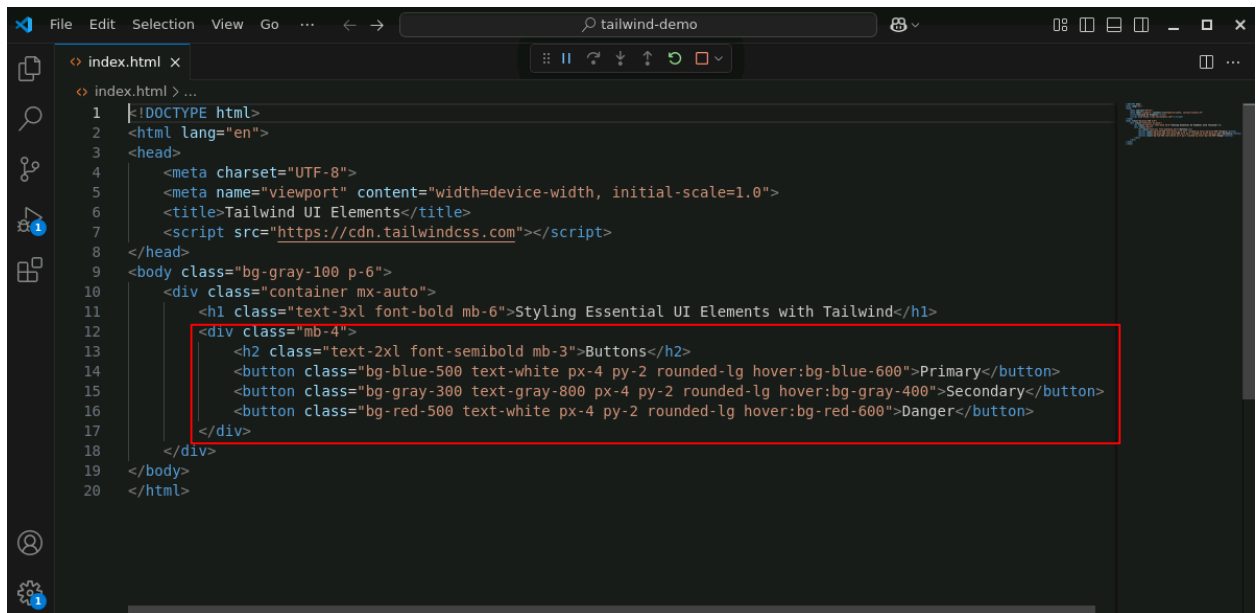
- 1.4 Save the file and open it in a web browser to verify that the basic structure of the web page is displayed correctly



Step 2: Create and style the button components

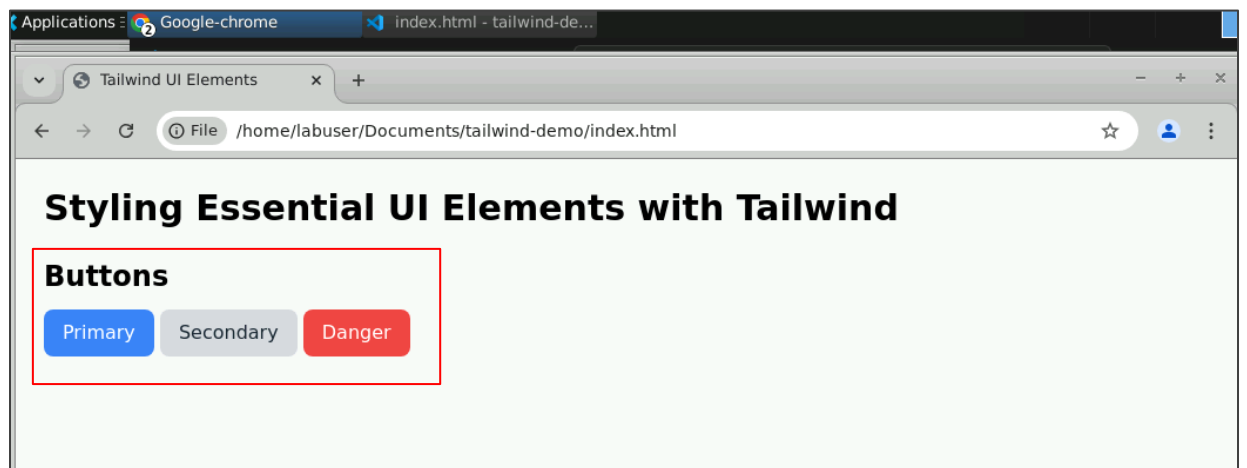
- 2.1 Modify the index.html file and add the following code for button styles inside the <body> tag:

```
<div class="mb-4">
  <h2 class="text-2xl font-semibold mb-3">Buttons</h2>
  <button class="bg-blue-500 text-white px-4 py-2 rounded-lg hover:bg-blue-600">Primary</button>
  <button class="bg-gray-300 text-gray-800 px-4 py-2 rounded-lg hover:bg-gray-400">Secondary</button>
  <button class="bg-red-500 text-white px-4 py-2 rounded-lg hover:bg-red-600">Danger</button>
</div>
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tailwind UI Elements</title>
7   <script src="https://cdn.tailwindcss.com"></script>
8 </head>
9 <body class="bg-gray-100 p-6">
10   <div class="container mx-auto">
11     <h1 class="text-3xl font-bold mb-6">Styling Essential UI Elements with Tailwind</h1>
12     <div class="mb-4">
13       <h2 class="text-2xl font-semibold mb-3">Buttons</h2>
14       <button class="bg-blue-500 text-white px-4 py-2 rounded-lg hover:bg-blue-600">Primary</button>
15       <button class="bg-gray-300 text-gray-800 px-4 py-2 rounded-lg hover:bg-gray-400">Secondary</button>
16       <button class="bg-red-500 text-white px-4 py-2 rounded-lg hover:bg-red-600">Danger</button>
17     </div>
18   </div>
19 </body>
20 </html>
```

2.2 Save and refresh the browser to see the styled buttons

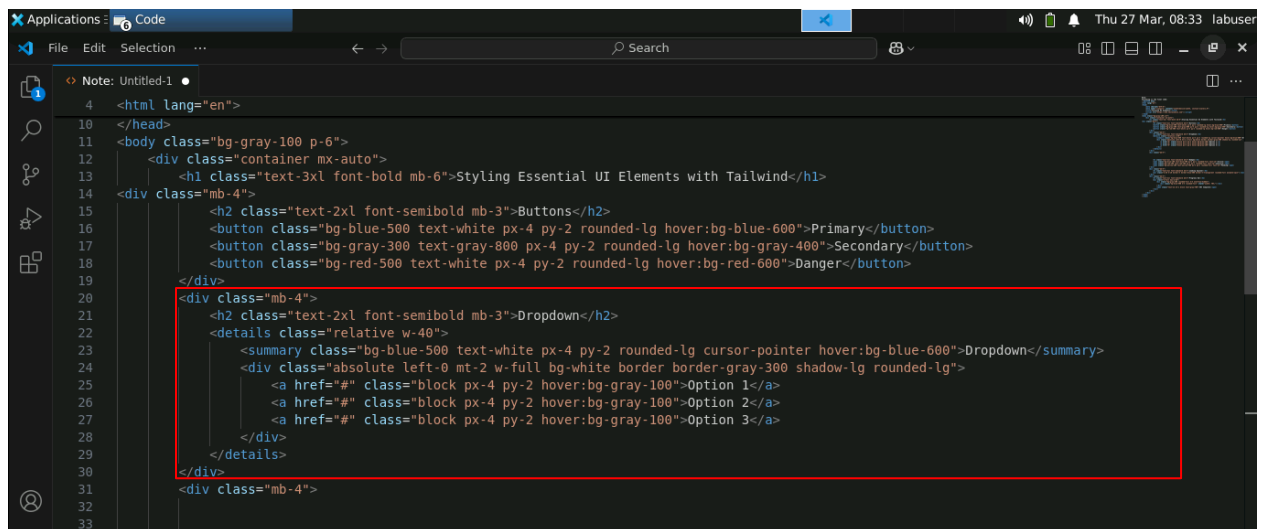


By completing this step, you have styled three types of buttons: Primary, Secondary, and Danger. Each button uses distinct colors.

Step 3: Create a dropdown menu with three selectable options

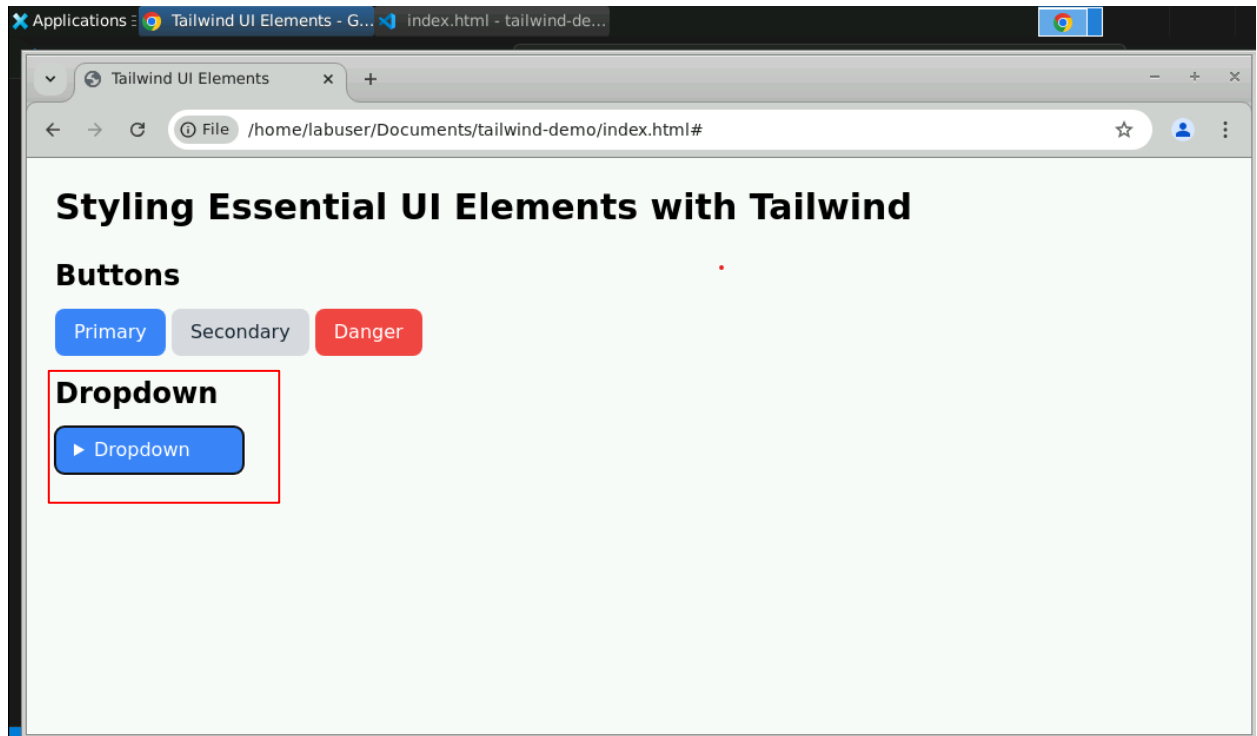
3.1 Modify the index.html file and add the following dropdown code inside the <body> tag to create an interactive menu with selectable options:

```
<div class="mb-4">
  <h2 class="text-2xl font-semibold mb-3">Dropdown</h2>
  <details class="relative w-40">
    <summary class="bg-blue-500 text-white px-4 py-2 rounded-lg cursor-pointer
hover:bg-blue-600">Dropdown</summary>
    <div class="absolute left-0 mt-2 w-full bg-white border border-gray-300
shadow-lg rounded-lg">
      <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option 1</a>
      <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option 2</a>
      <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option 3</a>
    </div>
  </details>
</div>
```



```
4 <html lang="en">
10 </head>
11 <body class="bg-gray-100 p-6">
12   <div class="container mx-auto">
13     <h1 class="text-3xl font-bold mb-6">Styling Essential UI Elements with Tailwind</h1>
14     <div class="mb-4">
15       <h2 class="text-2xl font-semibold mb-3">Buttons</h2>
16       <button class="bg-blue-500 text-white px-4 py-2 rounded-lg hover:bg-blue-600">Primary</button>
17       <button class="bg-gray-300 text-gray-800 px-4 py-2 rounded-lg hover:bg-gray-400">Secondary</button>
18       <button class="bg-red-500 text-white px-4 py-2 rounded-lg hover:bg-red-600">Danger</button>
19     </div>
20     <div class="mb-4">
21       <h2 class="text-2xl font-semibold mb-3">Dropdown</h2>
22       <details class="relative w-40">
23         <summary class="bg-blue-500 text-white px-4 py-2 rounded-lg cursor-pointer hover:bg-blue-600">Dropdown</summary>
24         <div class="absolute left-0 mt-2 w-full bg-white border border-gray-300 shadow-lg rounded-lg">
25           <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option 1</a>
26           <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option 2</a>
27           <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option 3</a>
28         </div>
29       </details>
30     </div>
31   </div>
32 </body>
33 </html>
```

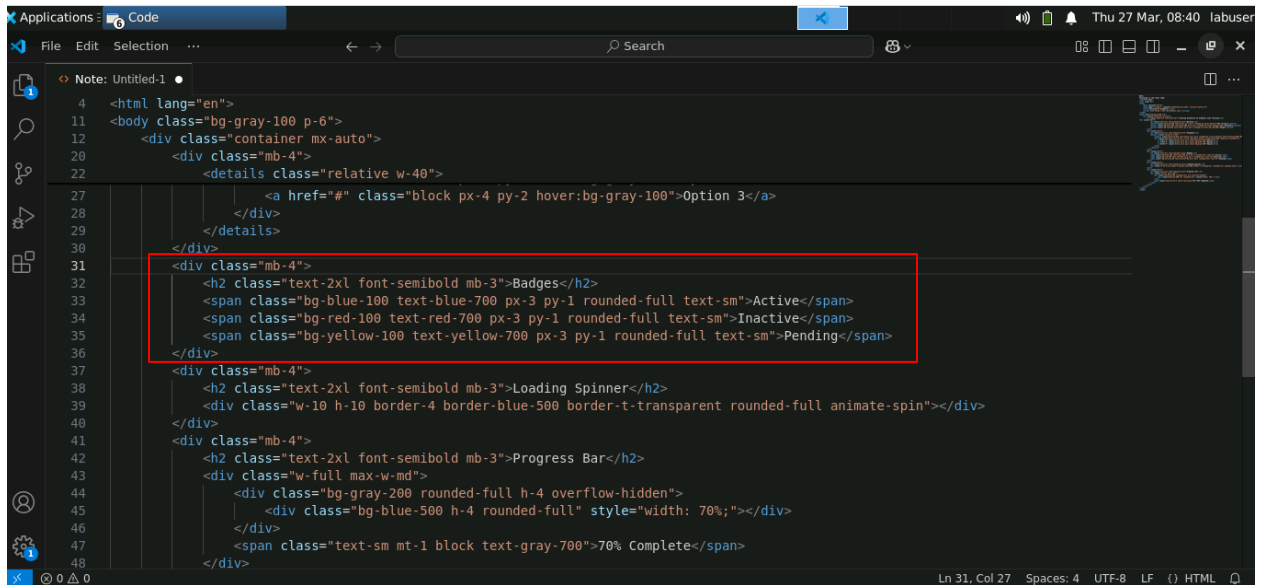
3.2 Save and refresh the browser to see the dropdown



Step 4: Design badge elements

4.1 Modify the index.html file and add the following code for badge elements inside the `<body>` tag:

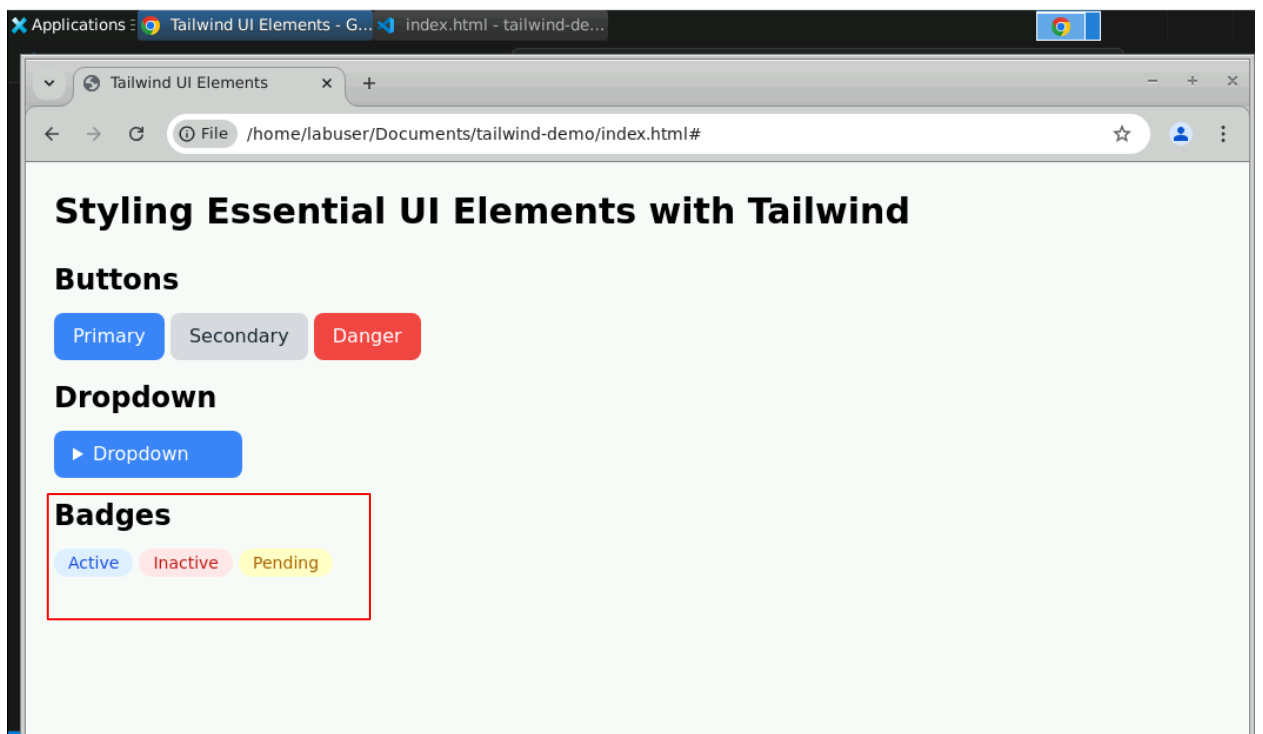
```
<div class="mb-4">
  <h2 class="text-2xl font-semibold mb-3">Badges</h2>
  <span class="bg-blue-100 text-blue-700 px-3 py-1 rounded-full text-sm">Active</span>
  <span class="bg-red-100 text-red-700 px-3 py-1 rounded-full text-sm">Inactive</span>
  <span class="bg-yellow-100 text-yellow-700 px-3 py-1 rounded-full text-sm">Pending</span>
</div>
```



The screenshot shows the VS Code editor with a file named 'Untitled-1'. The code is HTML and includes a section for 'Badges' which is highlighted with a red box. The code defines three spans: 'Active' (blue background), 'Inactive' (red background), and 'Pending' (yellow background). The code is as follows:

```
<html lang="en">
<body class="bg-gray-100 p-6">
  <div class="container mx-auto">
    <div class="mb-4">
      <details class="relative w-40">
        <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option 3</a>
      </div>
    </div>
    <div class="mb-4">
      <h2 class="text-2xl font-semibold mb-3">Badges</h2>
      <span class="bg-blue-100 text-blue-700 px-3 py-1 rounded-full text-sm">Active</span>
      <span class="bg-red-100 text-red-700 px-3 py-1 rounded-full text-sm">Inactive</span>
      <span class="bg-yellow-100 text-yellow-700 px-3 py-1 rounded-full text-sm">Pending</span>
    </div>
    <div class="mb-4">
      <h2 class="text-2xl font-semibold mb-3">Loading Spinner</h2>
      <div class="w-10 h-10 border-4 border-blue-500 border-t-transparent rounded-full animate-spin"></div>
    </div>
    <div class="mb-4">
      <h2 class="text-2xl font-semibold mb-3">Progress Bar</h2>
      <div class="w-full max-w-md">
        <div class="bg-gray-200 rounded-full h-4 overflow-hidden">
          <div class="bg-blue-500 h-4 rounded-full" style="width: 70%;"></div>
        </div>
        <span class="text-sm mt-1 block text-gray-700">70% Complete</span>
      </div>
    </div>
  </div>
</body>
</html>
```

4.2 Save and refresh the browser to see the badges

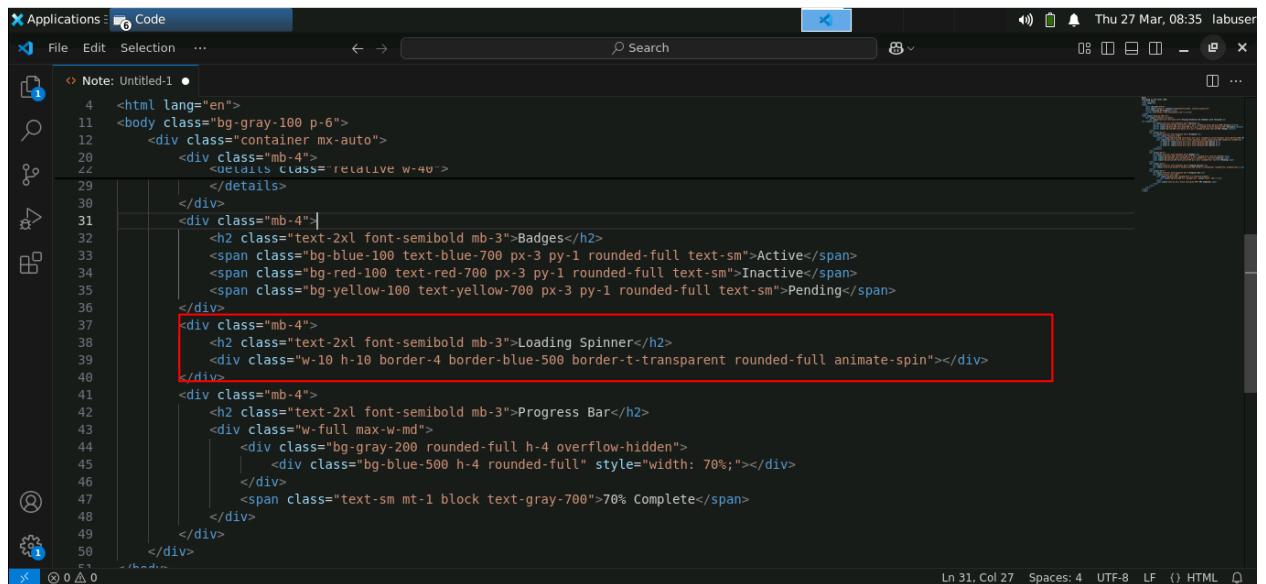


By following this step, we have designed the badges that can be used to visually highlight status indicators such as “Active,” “Inactive”, and “Pending.” Each badge will have a unique background and text color to represent its status clearly.

Step 5: Add a loading spinner

5.1 Modify the index.html file and insert code for a loading spinner inside the <body> tag:

```
<div class="mb-4">
  <h2 class="text-2xl font-semibold mb-3">Loading Spinner</h2>
  <div class="w-10 h-10 border-4 border-blue-500 border-t-transparent rounded-full
animate-spin"></div>
</div>
```

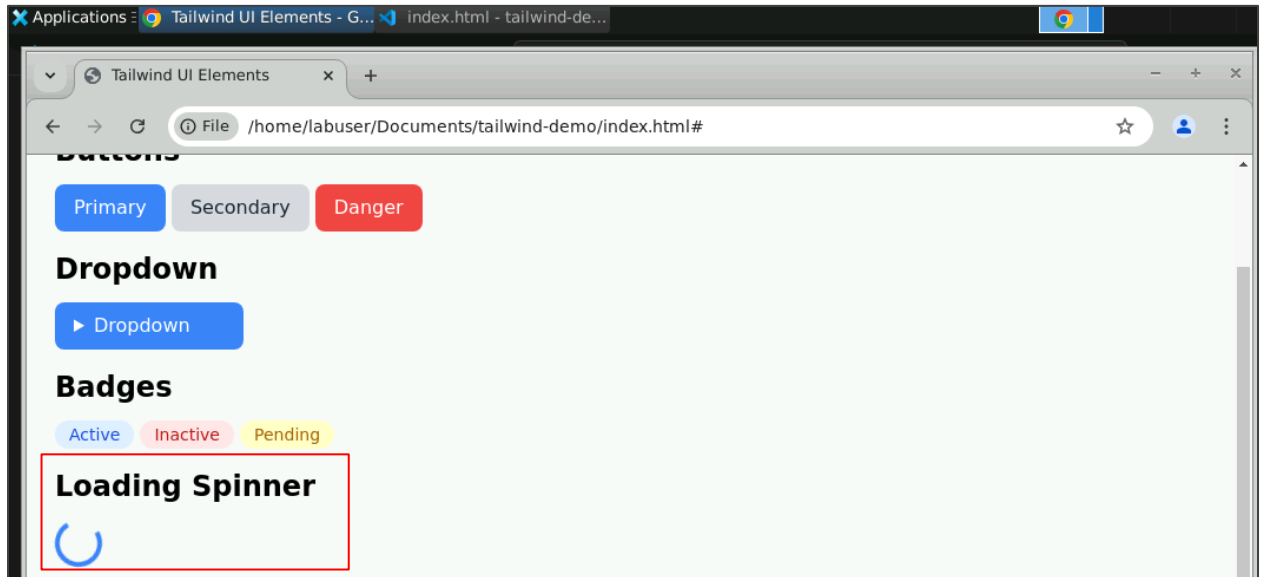


The screenshot shows a code editor window with the following code:

```
4 <html lang="en">
11 <body class="bg-gray-100 p-6">
12   <div class="container mx-auto">
20     <div class="mb-4">
21       <details class="relative w-40">
29       </details>
30     </div>
31     <div class="mb-4">
32       <h2 class="text-2xl font-semibold mb-3">Badges</h2>
33       <span class="bg-blue-100 text-blue-700 px-3 py-1 rounded-full text-sm">Active</span>
34       <span class="bg-red-100 text-red-700 px-3 py-1 rounded-full text-sm">Inactive</span>
35       <span class="bg-yellow-100 text-yellow-700 px-3 py-1 rounded-full text-sm">Pending</span>
36     </div>
37     <div class="mb-4">
38       <h2 class="text-2xl font-semibold mb-3">Loading Spinner</h2>
39       <div class="w-10 h-10 border-4 border-blue-500 border-t-transparent rounded-full animate-spin"></div>
40     </div>
41     <div class="mb-4">
42       <h2 class="text-2xl font-semibold mb-3">Progress Bar</h2>
43       <div class="w-full max-w-md">
44         <div class="bg-gray-200 rounded-full h-4 overflow-hidden">
45           <div class="bg-blue-500 h-4 rounded-full" style="width: 70%;"></div>
46         </div>
47         <span class="text-sm mt-1 block text-gray-700">70% Complete</span>
48       </div>
49     </div>
50   </div>
51 </body>
52 </html>
```

The code is displayed in a dark-themed editor. A red rectangle highlights the new loading spinner code block (lines 37-40). The status bar at the bottom indicates 'Ln 31, Col 27', 'Spaces: 4', 'UTF-8', 'LF', and 'HTML'.

5.2 Save and refresh the browser to see the loading spinner

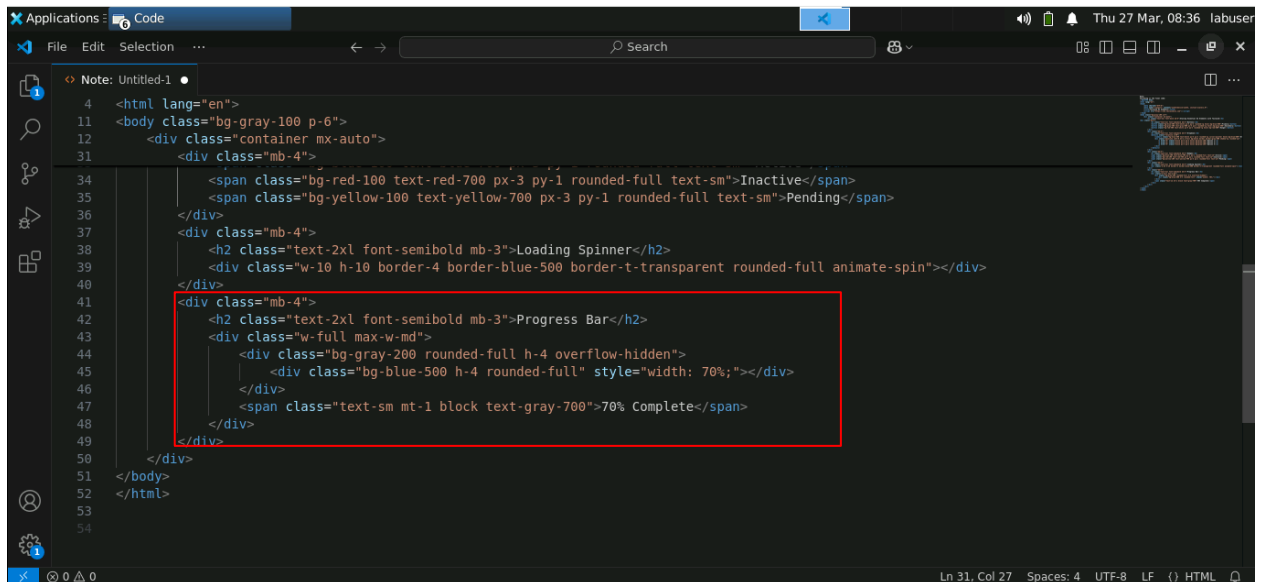


By following this step, you have inserted a loading spinner that visually indicates data is being fetched or a process is in progress. This helps enhance user experience by clearly signaling system activity during wait times.

Step 6: Build a progress bar

6.1 Modify the index.html file and add a code for progress bar inside the <body> tag:

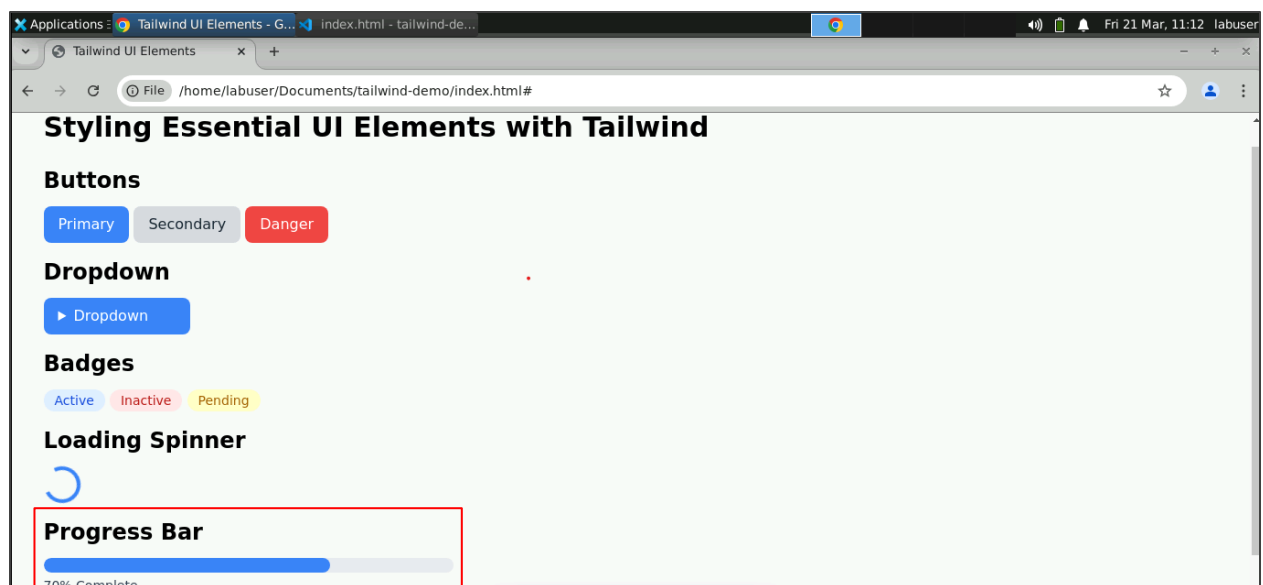
```
<div class="mb-4">
  <h2 class="text-2xl font-semibold mb-3">Progress Bar</h2>
  <div class="w-full max-w-md">
    <div class="bg-gray-200 rounded-full h-4 overflow-hidden">
      <div class="bg-blue-500 h-4 rounded-full" style="width: 70%;"></div>
    </div>
    <span class="text-sm mt-1 block text-gray-700">70% Complete</span>
  </div>
</div>
```



The screenshot shows the VS Code editor with a file named 'Untitled-1'. The code is HTML and uses Tailwind CSS classes. A red box highlights the following code block:

```
<div class="mb-4">
  <h2 class="text-2xl font-semibold mb-3">Progress Bar</h2>
  <div class="w-full max-w-md">
    <div class="bg-gray-200 rounded-full h-4 overflow-hidden">
      <div class="bg-blue-500 h-4 rounded-full" style="width: 70%;"></div>
    </div>
    <span class="text-sm mt-1 block text-gray-700">70% Complete</span>
  </div>
</div>
```

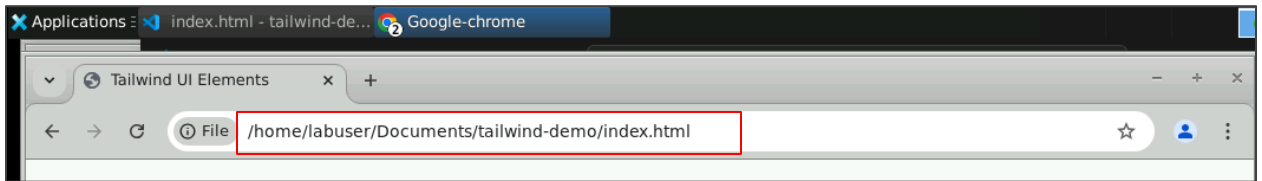
6.2 Save and refresh the browser to see the progress bar



By completing this step, you have added a progress bar that visually communicates the completion status of a task or process. It helps users understand how much of the task has been completed and how much remains, improving clarity and engagement.

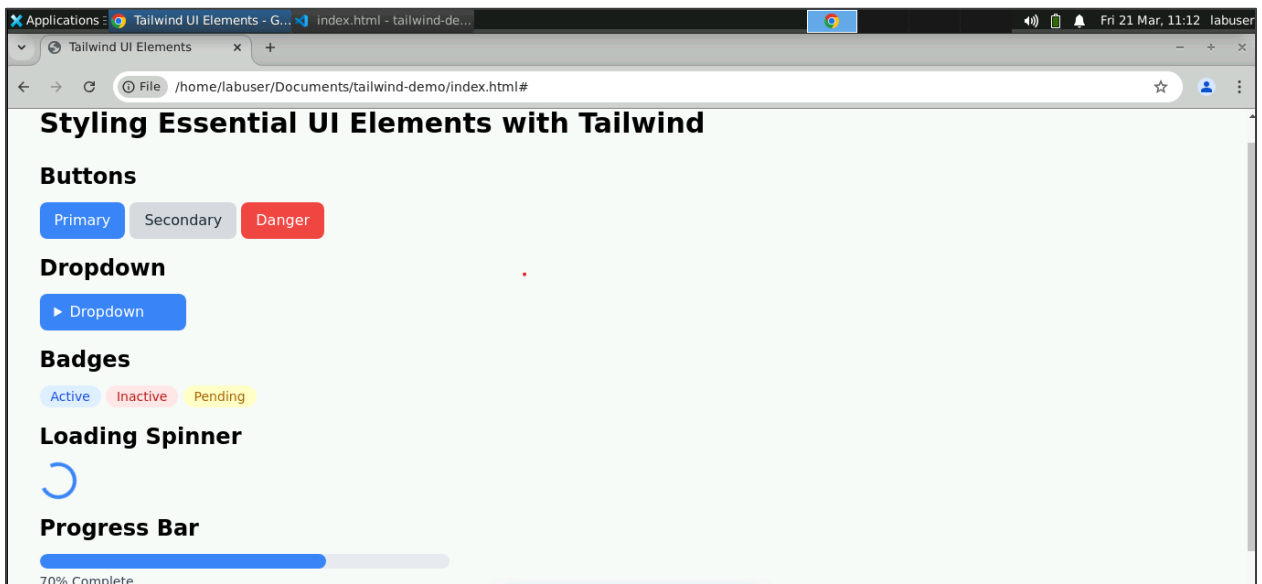
Step 7: Run the implementation in a web browser

7.1 Navigate to the project folder containing index.html file



By following this step, you navigate to the project folder named tailwind-demo and opens the index.html file in a browser.

7.2 Open the index.html in a web browser and verify all the UI elements



By following these steps, you have successfully styled essential UI elements using Tailwind CSS. Mastering Tailwind's utility classes allows you to build highly responsive, visually appealing, and maintainable web interfaces efficiently.