

Lesson 07 Demo 03

Styling Input Elements Using Tailwind CSS

Objective: To demonstrate how to style form input elements such as text fields, checkboxes, radio buttons, dropdowns, and file upload controls using Tailwind CSS utility classes. The example emphasizes enhancing UI with padding, borders, focus states, and responsive design.

Tools required: Visual Studio Code

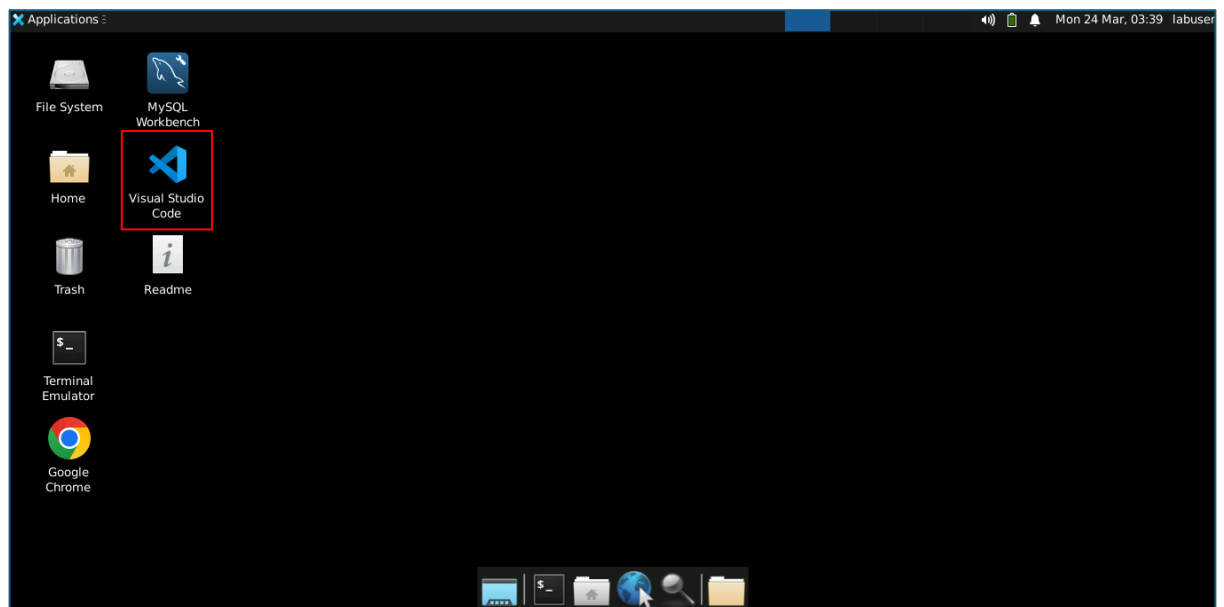
Prerequisites: HTML Basics

Steps to be followed:

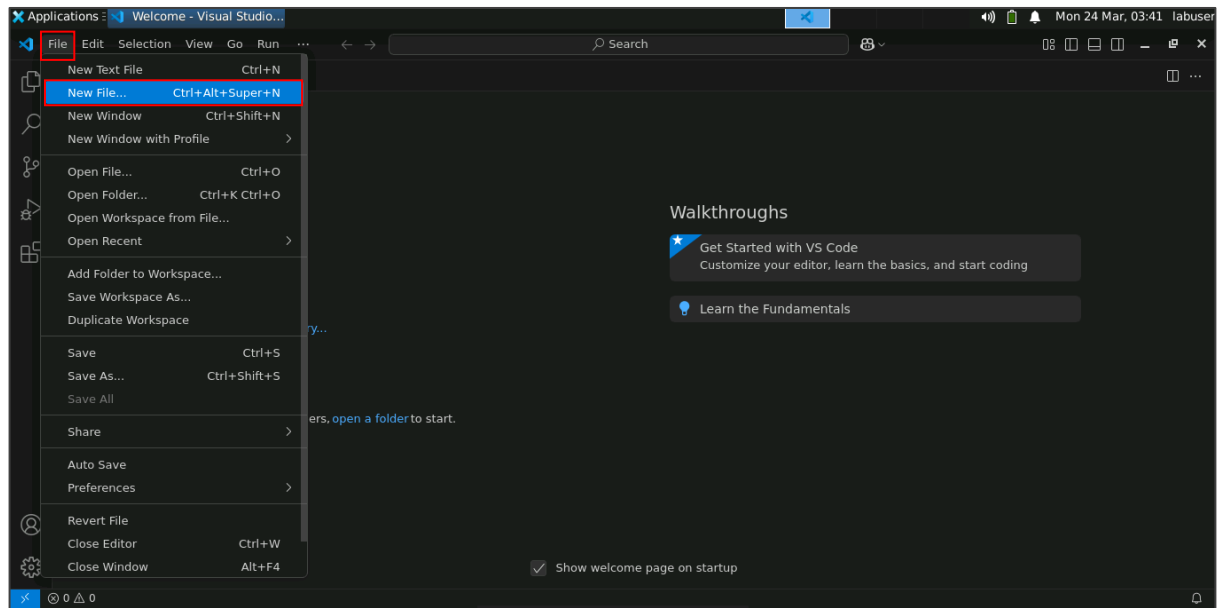
1. Open Visual Studio Code and create index.html
2. Implement the styled form layout

Step 1: Open Visual Studio Code and create index.html

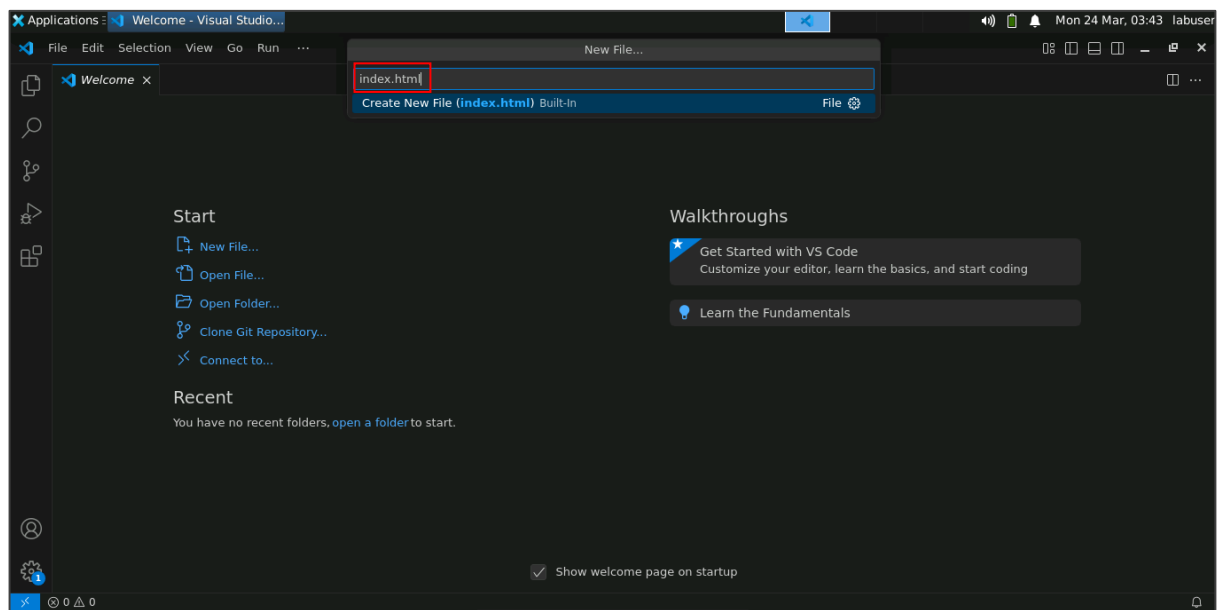
1.1 Open Visual Studio Code



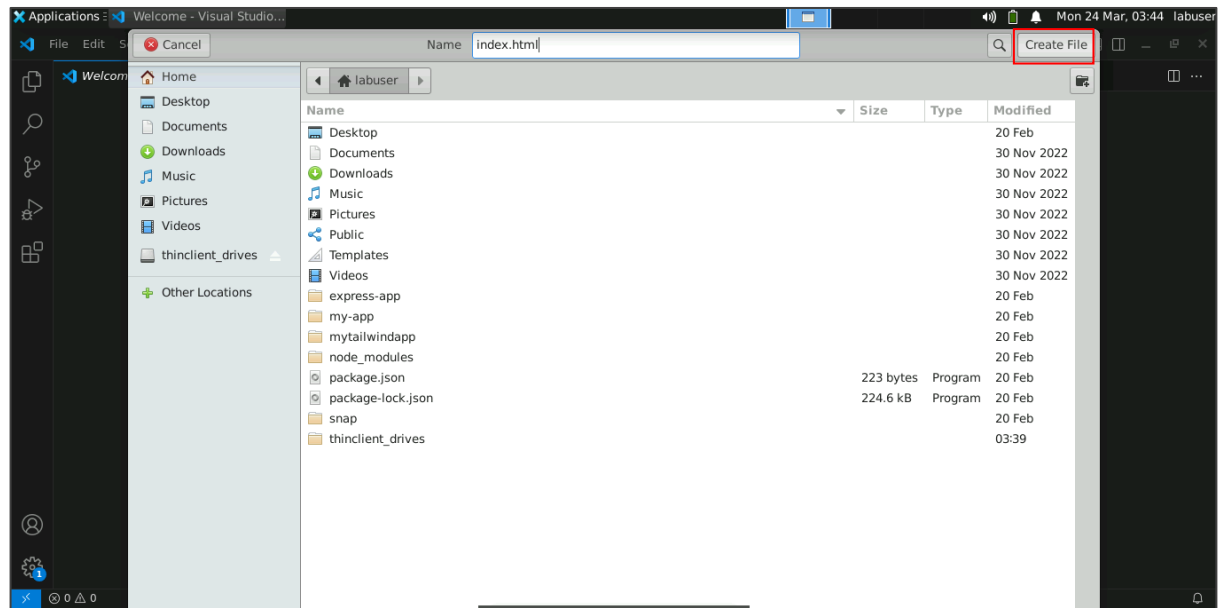
1.2 Click on **File**, then select the **New File...** option



1.3 Create an **index.html** file and press enter



1.4 Click **Create File**



1.5 Insert the following base HTML structure into 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>Form Elements</title>
```

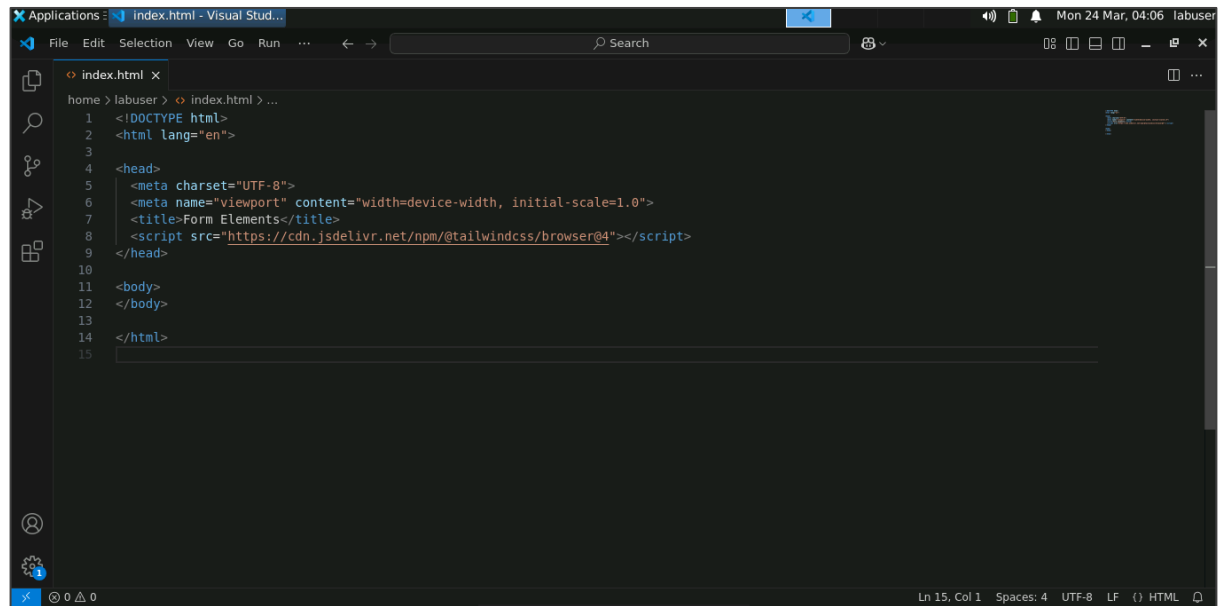
```
  <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```



Step 2: Implement the styled form layout

- 2.1 Modify the index.html file to style the hero section and form elements using Tailwind CSS:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

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

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

```
  <title>Form Elements</title>
```

```
  <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
```

```
</head>
```

```
<body>
```

```
  <div class="container mx-auto min-h-screen p-4">
```

```
    <!-- Hero Section -->
```

```
    <section class="bg-blue-600 text-white py-10 px-5 text-center rounded-lg">
```

```
      <div class="max-w-3xl mx-auto">
```

```
        <h1 class="text-5xl font-bold">Register with us</h1>
```

```
      </div>
```

```
    </section>
```

```

<!-- Form Card -->
<div class="bg-white p-6 mt-10 rounded-lg shadow-md w-full max-w-md mx-
auto">
  <h2 class="text-2xl font-bold mb-4 text-gray-700">Registration Form</h2>
  <form class="space-y-4" autocomplete="on" novalidate>

    <!-- Full Name -->
    <div class="flex flex-col">
      <label for="name" class="text-lg font-medium text-gray-600">Full
Name</label>
      <input type="text" id="name" name="name" required
      class="mt-1 p-2 border border-gray-300 rounded-md focus:outline-none
focus:ring-2 focus:ring-blue-400"
      autocomplete="name" />
      <span id="nameError" class="text-red-500 text-xs mt-1 hidden">Name is
required</span>
    </div>

    <!-- Email -->
    <div class="flex flex-col">
      <label for="email" class="text-lg font-medium text-gray-
600">Email</label>
      <input type="email" id="email" name="email" required
      class="mt-1 p-2 border border-gray-300 rounded-md focus:outline-none
focus:ring-2 focus:ring-blue-400"
      autocomplete="email" />
      <span id="emailError" class="text-red-500 text-xs mt-1 hidden">Invalid
email address</span>
    </div>

    <!-- Password -->
    <div class="flex flex-col">
      <label for="password" class="text-lg font-medium text-gray-
600">Password</label>
      <input type="password" id="password" name="password" required
      class="mt-1 p-2 border border-gray-300 rounded-md focus:outline-none
focus:ring-2 focus:ring-blue-400"
      autocomplete="new-password" />
      <span id="passwordError" class="text-red-500 text-xs mt-1
hidden">Password must be at least 6 characters</span>

```

</div>

<!-- Checkbox -->

<div class="flex items-center">

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

<label for="terms" class="ml-2 text-lg text-gray-700">Accept Terms and Conditions</label>

</div>

<!-- Radio Buttons -->

<label class="block text-lg font-medium text-gray-600">Gender</label>

<div class="flex items-center space-x-4">

<label class="flex items-center">

<input type="radio" name="gender" value="male" required class="w-4 h-4 text-blue-600 border-gray-300">

Male

</label>

<label class="flex items-center">

<input type="radio" name="gender" value="female" required class="w-4 h-4 text-blue-600 border-gray-300">

Female

</label>

</div>

<!-- Select Dropdown -->

<div>

<label for="country" class="block text-lg font-medium text-gray-600">Country</label>

<select id="country" required

class="mt-1 p-2 w-full border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-blue-400">

<option value="">Select Country</option>

<option value="IN">India</option>

<option value="US">United States</option>

<option value="UK">United Kingdom</option>

</select>

</div>

<!-- File Upload -->

<div>

```
    <label for="file" class="block text-lg font-medium text-gray-600">Upload  
Profile Picture</label>
```

```
    <input type="file" id="file" required  
      class="mt-1 px-2 py-1 block w-full text-lg text-gray-700 border border-  
gray-300 rounded-md cursor-pointer focus:outline-none focus:ring-2 focus:ring-  
blue-400" />  
  </div>
```

```
  <!-- Submit Button -->  
  <div>  
    <button type="submit"  
      class="w-full bg-blue-500 text-white p-2 rounded-md hover:bg-blue-600  
transition duration-200">  
      Register  
    </button>  
  </div>
```

```
</form>  
</div>  
</div>
```

```
<script>  
  const form = document.querySelector('form');  
  const nameInput = document.getElementById('name');  
  const emailInput = document.getElementById('email');  
  const passwordInput = document.getElementById('password');  
  const nameError = document.getElementById('nameError');  
  const emailError = document.getElementById('emailError');  
  const passwordError = document.getElementById('passwordError');  
  
  form.addEventListener('submit', (e) => {  
    let isValid = true;  
  
    if (nameInput.value.trim() === '') {  
      nameError.classList.remove('hidden');  
      isValid = false;  
    } else {  
      nameError.classList.add('hidden');  
    }  
  
    if (!/\S+@\S+\.\S+/.test(emailInput.value)) {
```

```

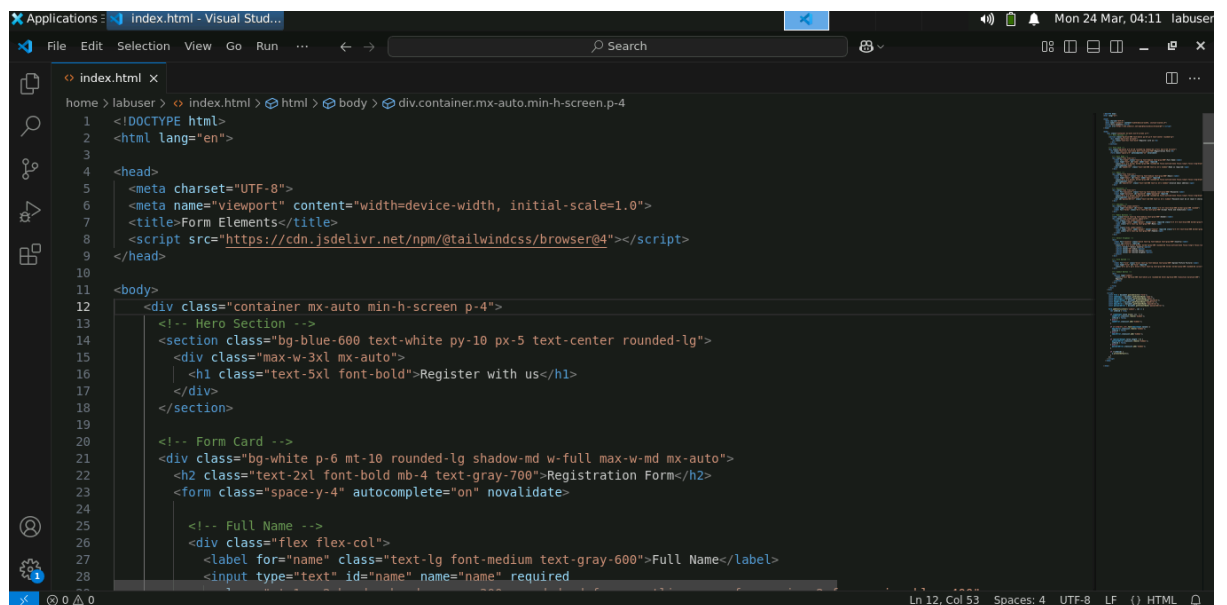
        emailError.classList.remove('hidden');
        isValid = false;
    } else {
        emailError.classList.add('hidden');
    }

    if (passwordInput.value.length < 6) {
        passwordError.classList.remove('hidden');
        isValid = false;
    } else {
        passwordError.classList.add('hidden');
    }

    if (!isValid) {
        e.preventDefault();
    }
});
</script>
</body>

</html>

```

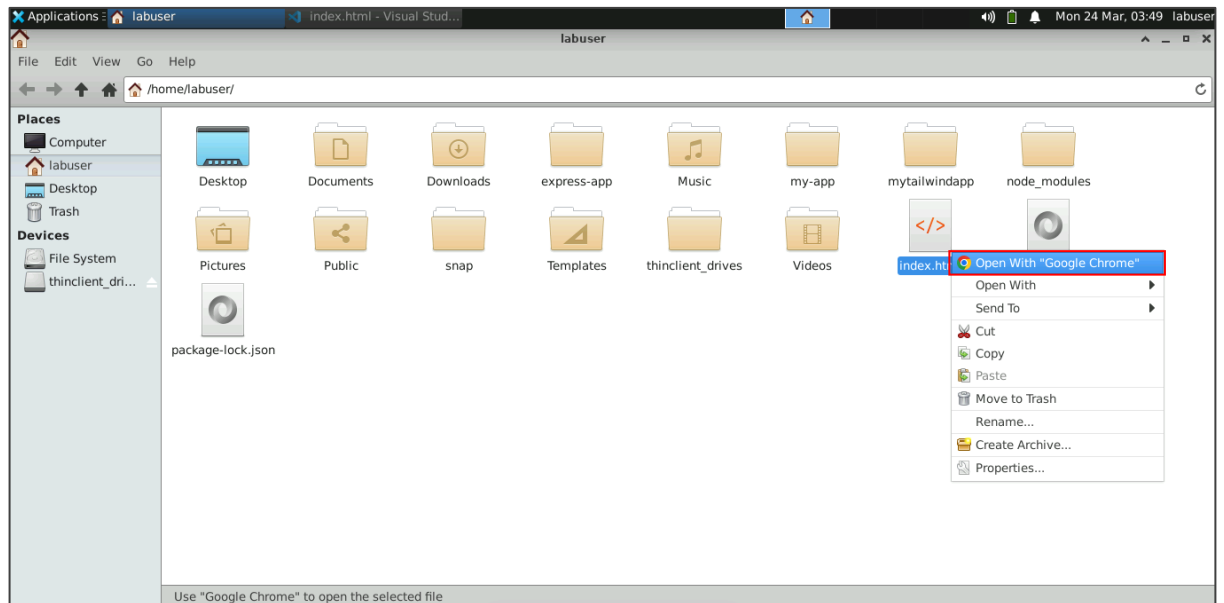


```

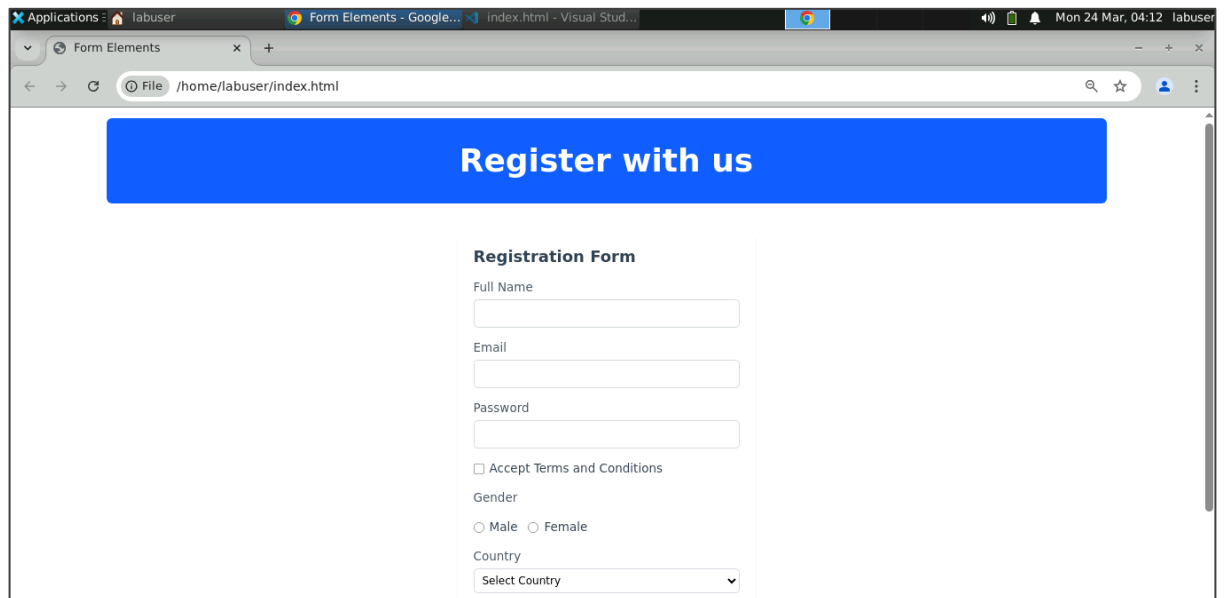
index.html x
home > labuser > index.html > html > body > div.container.mx-auto.min-h-screen.p-4
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Form Elements</title>
8    <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
9  </head>
10
11 <body>
12   <div class="container mx-auto min-h-screen p-4">
13     <!-- Hero Section -->
14     <section class="bg-blue-600 text-white py-10 px-5 text-center rounded-lg">
15       <div class="max-w-3xl mx-auto">
16         <h1 class="text-5xl font-bold">Register with us</h1>
17       </div>
18     </section>
19
20     <!-- Form Card -->
21     <div class="bg-white p-6 mt-10 rounded-lg shadow-md w-full max-w-md mx-auto">
22       <h2 class="text-2xl font-bold mb-4 text-gray-700">Registration Form</h2>
23       <form class="space-y-4" autocomplete="on" novalidate>
24
25         <!-- Full Name -->
26         <div class="flex flex-col">
27           <label for="name" class="text-lg font-medium text-gray-600">Full Name</label>
28           <input type="text" id="name" name="name" required

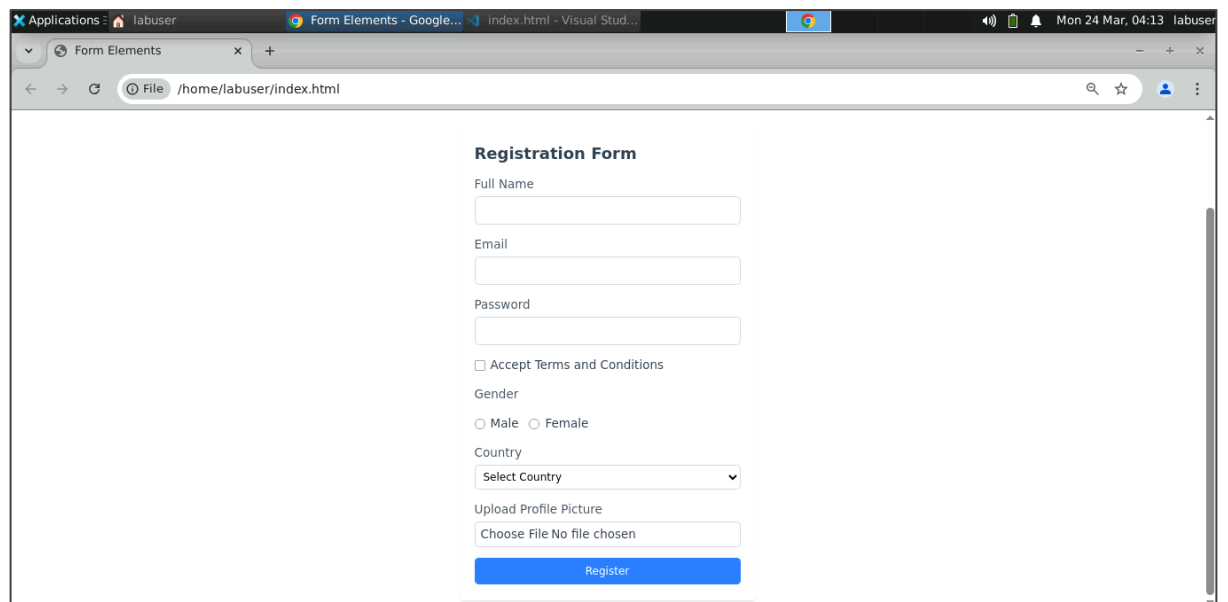
```


2.2 Navigate to the index.html file, right-click on the file, and select **Open With “Google Chrome”**



2.3 Verify that the hero section and form elements are displayed as expected:





By following the above steps, you have successfully styled a responsive registration form using Tailwind CSS utility classes, enhancing the user interface with consistent spacing, borders, focus states, and accessible form controls.