

## 1. Mục tiêu

- Nhúng bootstrap
- Sử dụng grid
- Các thành phần bootstrap hay sử dụng:
  - Menu
  - Form
  - Table
  - Modal
- Cài đặt và sử dụng extension visual code cho bootstrap
  - Bootstrap 4, Font awesome 4, Font Awesome 5 Free & Pro snippets

## 2. Tóm tắt lý thuyết

### 2.1 Bootstrap là gì?

**Bootstrap** là một thư viện html, css và js. Bootstrap sử dụng html, css và js tạo ra những mẫu cơ bản như: form, button, table, navigation... giúp nhà phát triển web dễ dàng tạo ra những giao diện web tương thích với màn hình điện thoại.

### 2.2 Cách thêm thư viện Bootstrap vào website.

Có 2 cách để sử dụng bootstrap:

- Cách 1: tải bootstrap từ <https://getbootstrap.com/>
- Cách 2: dẫn đường link trực tiếp từ CDN



### Install via package manager

Install Bootstrap's source Sass and JavaScript files via npm, RubyGems, Composer, or Meteor. Package managed installs don't include documentation or our full build scripts. You can also [use any demo from our Examples repo](#) to quickly jumpstart Bootstrap projects.

```
$ npm install bootstrap@5.3.2
```



```
$ gem install bootstrap -v 5.3.2
```





## Include via CDN

When you only need to include Bootstrap's compiled CSS or JS, you can use [jsDelivr](#). See it in action with our simple [quick start](#), or [browse the examples](#) to jumpstart your next project. You can also choose to include Popper and our JS [separately](#).

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Tc2EqChoung9h6i/2CmhYvK64w7JhoQFj6p/vGm9yDBshGF0fYweqZbz0WL9n20CW" crossorigin="anonymous" />
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6Rerhg8pp1kYSG4+k7Tit9iJWBUgeLpbQ:Wqst3eF+0tZ/7TKyiLg1TRqMzsMT/IBt+saie6DKNJs935QyU3VAGC8N70AE" crossorigin="anonymous" />
```

### 2.3 Website sử dụng bootstrap cơ bản

Bootstrap bắt buộc các yếu tố html và css ở định dạng HTML5. Luôn bao gồm định dạng HTML5 **DOCTYPE** ở đầu trang, cùng với nó là các thuộc tính **lang** (xác định ngôn ngữ web) và thẻ **<meta>** với thuộc tính **charset="utf-8"** (định dạng bộ gõ ký tự cho web).

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
  </head>
</html>
```

Bootstrap được thiết kế để hiển thị tốt trên các thiết bị di động. Để đảm bảo chiều rộng của trang và tỷ lệ zoom của trang trên trình duyệt, cần phải thêm thẻ **<meta>** với các thuộc tính sau:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

### 2.4 Grid system trong Bootstrap

#### 2.4.1 Breakpoint

Breakpoint (điểm ngắt) chính là thứ sẽ giúp ta tùy chỉnh layout trên mỗi viewport sizes (khung nhìn) khác nhau. Bootstrap bao gồm **6 breakpoint mặc định** hay còn gọi là các **grid tiers** (tầng lưới), với mỗi breakpoint này chúng ta có thể áp dụng các class khác nhau của bootstrap để có thể tùy chỉnh layout theo ý muốn của mình. Chúng ta sẽ hiểu rõ hơn về nó qua những ví dụ sẽ đề cập sau.

| Breakpoint        | Class infix | Dimensions |
|-------------------|-------------|------------|
| X-Small           | <i>None</i> | <576px     |
| Small             | <i>sm</i>   | ≥576px     |
| Medium            | <i>md</i>   | ≥768px     |
| Large             | <i>lg</i>   | ≥992px     |
| Extra large       | <i>xl</i>   | ≥1200px    |
| Extra extra large | <i>xxl</i>  | ≥1400px    |

### 2.4.2 Container

Container (vùng chứa) dùng để bao quanh nội dung, nội dung bên trong container sẽ có kích thước tùy thuộc vào màn hình hoặc mỗi điểm ngắt. Container có 3 loại class chính gồm:

- **container-fluid**: Chiều rộng luôn là 100% kích thước màn hình.
- **container**: Chiều rộng sẽ thay đổi tùy theo kích thước màn hình (theo bảng bên dưới).
- **container-{breakpoint}**: Chiều rộng luôn là 100% đến khi nào đạt tới điểm ngắt của breakpoint.

|                               | Extra small<br><576px | Small<br>≥576px | Medium<br>≥768px | Large<br>≥992px | X-Large<br>≥1200px | XX-Large<br>≥1400px |
|-------------------------------|-----------------------|-----------------|------------------|-----------------|--------------------|---------------------|
| <code>.container</code>       | 100%                  | 540px           | 720px            | 960px           | 1140px             | 1320px              |
| <code>.container-sm</code>    | 100%                  | 540px           | 720px            | 960px           | 1140px             | 1320px              |
| <code>.container-md</code>    | 100%                  | 100%            | 720px            | 960px           | 1140px             | 1320px              |
| <code>.container-lg</code>    | 100%                  | 100%            | 100%             | 960px           | 1140px             | 1320px              |
| <code>.container-xl</code>    | 100%                  | 100%            | 100%             | 100%            | 1140px             | 1320px              |
| <code>.container-xxl</code>   | 100%                  | 100%            | 100%             | 100%            | 100%               | 1320px              |
| <code>.container-fluid</code> | 100%                  | 100%            | 100%             | 100%            | 100%               | 100%                |

### 2.4.3 Grid system

Hệ thống lưới của bootstrap sử dụng các **container** (vùng chứa), **row** (hàng) và **col** (cột) để xây dựng layout một cách linh hoạt, có thể bố trí và căn chỉnh nội dung theo từng kích thước màn hình khác nhau.

#### a. Cách hoạt động

- **Grid system** sử dụng 6 breakpoint để có thể dễ dàng responsive. 6 breakpoint này được dựng trên min-width media queries trong css. Do vậy mỗi thuộc tính của breakpoint sẽ ảnh hưởng lên chính nó và các breakpoint ở trên.

Ví dụ: `col-sm-4`, phần tử chứa class này sẽ có độ rộng là 4 cột từ breakpoint `sm` trở lên.

- **Container** sẽ chứa nội dung của ta và căn chỉnh nó theo chiều rộng của màn hình thiết bị, nó cũng sẽ tự động thêm padding để tránh content bị sát lề khi chiều rộng màn hình quá nhỏ.
- **Row** sẽ bao bọc các **Column**, class row sẽ có thuộc tính `display: flex` để khiến cho các cột nằm ngang. Mỗi cột sẽ có 1 phần padding ở 2 bên (được gọi là gutter) để điều chỉnh khoảng cách giữa các cột. Do vậy class row sẽ có thêm `margin horizontal âm` để xóa bỏ phần padding thừa ở 2 cột ngoài cùng 2 bên.
- Các **Column** rất linh hoạt, mỗi hàng sẽ có đến 12 cột cho phép ta có thể tạo ra các phần tử trên một hàng có số cột khác nhau, chiều rộng khác nhau.

Ví dụ: `col-6`, phần tử chứa class này sẽ chiếm chiều rộng bằng 1 nửa container chứa nó (`50% width`).

- **Gutters** được sử dụng để điều chỉnh khoảng cách giữa các phần tử trên cùng 1 hàng.

Ví dụ: `gx-*` điều chỉnh khoảng cách ngang, `gy-*` điều chỉnh khoảng cách dọc và `g-*` điều chỉnh khoảng cách cả 2 chiều.

- Chúng ta có thể sử dụng Bootstrap bằng `source Sass` để có thể dễ dàng tùy chỉnh grid system.

**b. Breakpoint trên grid system**

Với mỗi breakpoint, container sẽ có chiều rộng khác nhau (như hình dưới) và chúng ta cũng có thể điều chỉnh độ rộng của mỗi cột bằng cú pháp `col-{breakpoint}-{number}`

|                            | <b>xs</b><br><576px               | <b>sm</b><br>≥576px   | <b>md</b><br>≥768px   | <b>lg</b><br>≥992px   | <b>xl</b><br>≥1200px  | <b>xxl</b><br>≥1400px  |
|----------------------------|-----------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| <b>Container max-width</b> | None (auto)                       | 540px                 | 720px                 | 960px                 | 1140px                | 1320px                 |
| <b>Class prefix</b>        | <code>.col-</code>                | <code>.col-sm-</code> | <code>.col-md-</code> | <code>.col-lg-</code> | <code>.col-xl-</code> | <code>.col-xxl-</code> |
| <b># of columns</b>        | 12                                |                       |                       |                       |                       |                        |
| <b>Gutter width</b>        | 1.5rem (.75rem on left and right) |                       |                       |                       |                       |                        |
| <b>Custom gutters</b>      | <a href="#">Yes</a>               |                       |                       |                       |                       |                        |
| <b>Nestable</b>            | <a href="#">Yes</a>               |                       |                       |                       |                       |                        |
| <b>Column ordering</b>     | <a href="#">Yes</a>               |                       |                       |                       |                       |                        |

**c. Chiều rộng column tự động**

- Các **Column** có chiều rộng bằng nhau:

Chúng ta có thể sử dụng class `col` để khiến các cột có chiều rộng bằng nhau.

```
<div class="container">
  <div class="row">
    <div class="col" style="background-color: red;">
      1 of 3
    </div>
    <div class="col" style="background-color: yellow;">
      2 of 3
    </div>
    <div class="col" style="background-color: green;">
      3 of 3
    </div>
  </div>
</div>
```



- *Chỉ định chiều rộng cho 1 Column riêng biệt*

Ta cũng có thể chỉ định chiều rộng cho một cột riêng biệt và các cột còn lại sẽ tự động điều chỉnh chiều rộng để lấp đầy dòng.

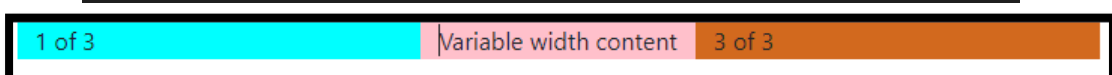
```
<div class="container">
  <div class="row">
    <div class="col" style="background-color: grey;">
      1 of 3
    </div>
    <div class="col-6" style="background-color: red;">
      2 of 3 (wider)
    </div>
    <div class="col" style="background-color: grey;">
      3 of 3
    </div>
  </div>
</div>
```



- *Chiều rộng của Column theo content*

Sử dụng cú pháp `col-{breakpoint}-auto` để có được một cột có chiều dài dựa trên nội dung bên trong của nó.

```
<div class="container">
  <div class="row justify-content-md-center">
    <div class="col col-lg-2" style="background-color: aqua;">
      1 of 3
    </div>
    <div class="col-md-auto" style="background-color: pink;">
      Variable width content
    </div>
    <div class="col col-lg-2" style="background-color: chocolate;">
      3 of 3
    </div>
  </div>
</div>
```



*d. Responsive với Grid system*

- *Áp dụng breakpoint*

Chúng ta có thể sử dụng breakpoint để chỉ định độ rộng của cột cho từng màn hình cụ thể.

```
<div class="container">
  <div class="row">
    <div class="col-sm-8">col-sm-8</div>
    <div class="col-sm-4">col-sm-4</div>
  </div>
</div>
```



Đoạn code trên sẽ hiển thị 1 dòng với 2 cột từ màn hình **sm** trở lên, cột một có độ rộng là 8 ô còn cột 2 có độ rộng là 4 ô.

- *Kết hợp nhiều breakpoint*

Nếu ta muốn bố cục hiển thị khác trên một breakpoint nào đó thì hãy kết hợp chúng lại. Ví dụ nếu ta muốn 1 bố cục hiển thị như sau:



Hiển thị trên màn hình **md** trở xuống:



Thì hãy sử dụng đoạn dưới đây. Mỗi cột trong dòng sẽ chiếm 6 ô nhưng từ màn hình **md** trở lên mỗi cột sẽ chiếm 4 ô.

```
<div class="container">
  <div class="row">
    <div class="col-6 col-md-4"> .col-6 .col-md-4</div>
    <div class="col-6 col-md-4"> .col-6 .col-md-4</div>
    <div class="col-6 col-md-4"> .col-6 .col-md-4</div>
  </div>
</div>
```

- *Tùy chỉnh số cột ở mỗi dòng*

Chúng ta có thể sử dụng class `.row-cols-{number}` cho mỗi dòng để đặt nhanh độ rộng của các cột trong hàng thay vì phải set riêng cho từng cột riêng biệt (*lưu ý number chỉ nhận giá trị là 'auto' hoặc từ 1 đến 6*).

|        |        |        |        |        |
|--------|--------|--------|--------|--------|
| Column | Column | Column | Column | Column |
| Column | Column | Column |        |        |

```
<div class="container">
  <div class="row row-cols-5">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

- *Lồng nhau*

Trong mỗi cột có thể lồng thêm các dòng, như ví dụ sau:

|                    |                           |                           |
|--------------------|---------------------------|---------------------------|
| Level 1: .col-sm-3 | Level 2: .col-8 .col-sm-6 | Level 2: .col-4 .col-sm-6 |
|--------------------|---------------------------|---------------------------|

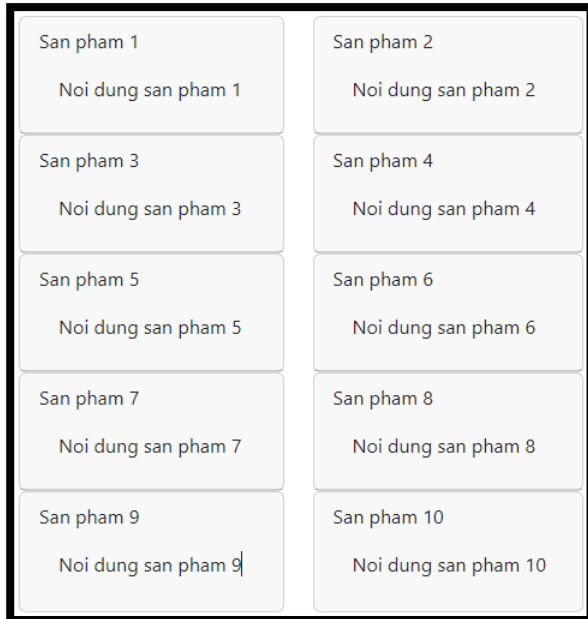
```
<div class="container">
  <div class="row">
    <div class="col-sm-3">
      Level 1: .col-sm-3
    </div>
    <div class="col-sm-9">
      <div class="row">
        <div class="col-8 col-sm-6">
          Level 2: .col-8 .col-sm-6
        </div>
        <div class="col-4 col-sm-6">
          Level 2: .col-4 .col-sm-6
        </div>
      </div>
    </div>
  </div>
</div>
```

- *Tip tạo Grid*

Ví dụ:

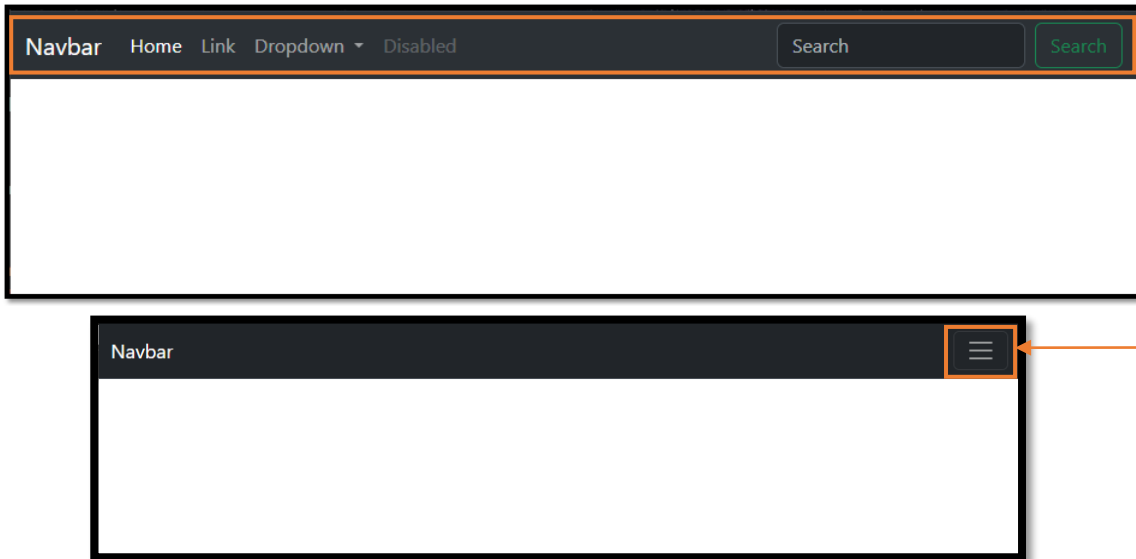
```
.row>.col-xm-12.col-md-6.col-xl-4*10>.card>.card-header>{San pham $}+.card-
body>{Noi dung san pham $}
```

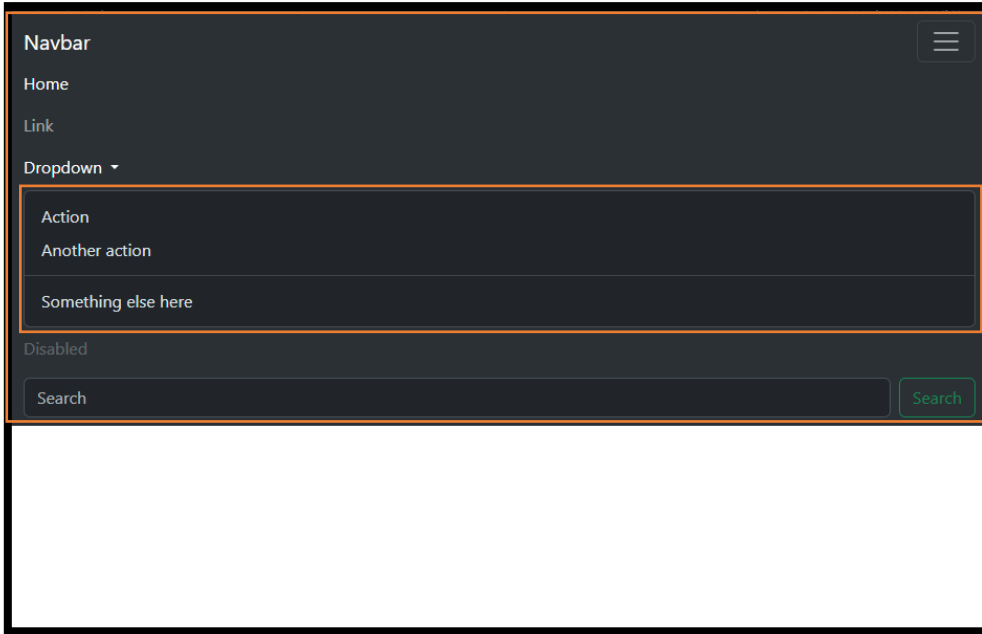
```
Emmet Abbreviation x
<div class="row">
  <div class="col-xm-12 col-md-6 col-xl-4">
    <div class="card">
      <div class="card-header">
        San pham 1
      <div class="card-body">Noi
dung san pham 1</div>
    </div>
  </div>
  <div class="col-xm-12 col-md-6 col-xl-4">
    <div class="card">
      <div class="card-header">
        San pham 2
      <div class="card-body">Noi
dung san pham 2</div>
    </div>
  </div>
  <div class="col-xm-12 col-md-6 col-xl-4">
    <div class="card">
      <div class="card-header">
        San pham 3
      <div class="card-body">Noi
```



### 3. Các thành phần Bootstrap hay sử dụng.

#### a. Menu (Navbar)





```

<nav class="navbar navbar-expand-lg bg-body-tertiary" data-bs-theme="dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
      aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li>
              <hr class="dropdown-divider">
            </li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" aria-disabled="true">Disabled</a>
        </li>
      </ul>
      <form class="d-flex" role="search">
        <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
        <button class="btn btn-outline-success" type="submit">Search</button>
      </form>
    </div>
  </div>
</nav>

```

Những lưu ý sau đây ta cần biết trước khi bắt đầu với thanh điều hướng (*Navbar*):

- Thanh điều hướng yêu cầu sử dụng class `.navbar` với `.navbar-expand{-sm|-md|-lg|-xl|-xxl}` cho chức năng tương thích (reponsive) và phối màu ([color scheme](#)).
- Thanh điều hướng và nội dung của chúng theo mặc định rất linh hoạt. Thay đổi [container](#) để giới hạn chiều rộng – ngang.
- Sử dụng các lớp [spacing](#) và [flex](#) để chỉnh khoảng cách và căn chỉnh trong thanh điều hướng.
- Thanh điều hướng phản hồi theo mặc định nhưng ta có thể dễ dàng chỉnh sửa. Hoạt động phản hồi phụ thuộc vào plugin Collapse JavaScript của chúng ta.
- Đảm bảo khả năng truy cập bằng cách sử dụng `<nav>` hoặc nếu sử dụng `<div>`, phải thêm `role="navigation"` vào mỗi thanh điều hướng để xác định rõ khu vực cần thay đổi riêng đối với người dùng.
- Chỉ ra mục hiện tại bằng cách sử dụng `aria-current="page"` cho trang hiện tại hoặc `aria-current="true"` cho đối tượng trong 1 tập đối tượng hiện tại.
- Tính năng mới trong **v5.2.0**: Thanh điều hướng có thể được tạo theo chủ đề bằng các biến CSS nằm trong phạm vi của class `.navbar`. Class `.navbar-light` không còn được sử dụng nữa, thay vào đó class `.navbar-dark` đã được viết lại để ghi đè các biến CSS thay vì bổ sung thêm các style.

Thanh điều hướng còn hỗ trợ cho một số thành phần nội dung phụ như:

- `.navbar-brand` đặt nhãn tên.
- `.navbar-nav` chỉnh độ cao (bao gồm tạo cho menu Dropdown).
- `.navbar-toggler` để tạo nút bấm số xuống .
- Tiện ích [spacing](#) và [flex](#) tác động đến các thành phần trong form.
- `.navbar-text` để thêm các văn bản được căn giữa theo chiều dọc.
- `.collapse.navbar-collapse` để nhóm và ẩn nội dung thanh điều hướng theo breakpoint.
- `.navbar-scroll` để đặt `max-height` và kích hoạt chức năng cuộn số xuống.

**b. Form:**

Địa chỉ Email

Chúng tôi sẽ không chia sẻ địa chỉ của bạn ở bất kì nơi khác.

Mật khẩu

Kiểm tra

**Nộp**

```
<form>
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Địa chỉ Email</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
    <div id="emailHelp" class="form-text">Chúng tôi sẽ không chia sẻ địa chỉ của bạn ở bất kì nơi khác.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Mật khẩu</label>
    <input type="password" class="form-control" id="exampleInputPassword1">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Kiểm tra</label>
  </div>
  <button type="submit" class="btn btn-primary">Nộp</button>
</form>
```

Các thành phần trong form phải được đặt tên để truy vấn. Cách đơn giản nhất là sử dụng `<label>`. Trong trường hợp sử dụng nút `<button>` - cần mô tả đầy đủ như một phần nội dung của cặp thẻ `<button>...</button>`.

Trường hợp không muốn hiển thị nội dung với thẻ `<label>`, có thể sử dụng các cách sau:

- Sử dụng class `.visually-hidden` cho các thẻ `<label>`
- Trở tới một phần tử hiện có có thể hoạt động như một nhãn bằng cách sử dụng `aria-labelledby`
- Cung cấp một thuộc tính `title`
- Đặt tên phần tử rõ ràng để truy cập bằng cách sử dụng `aria-label`
- Sử dụng thuộc tính `placeholder` cho mô tả trong `<input>` và `<textarea>`

### c. *Table*:

Khai báo sử dụng class `.table` trước tiên. Sử dụng class `.table-striped` để định dạng các hàng (rows) xen kẽ. Lưu ý là các rows phải nằm trong cặp thẻ `<tbody>`.

Tương tự với định dạng cột ta sử dụng class `.table-striped-columns` hoặc class `.table-hover` cho *hiệu ứng rê chuột* vào các rows.

⇒ Các class có thể kết hợp với nhau để tạo ra kết quả như mong muốn. Như hình ví dụ dưới đây kết hợp định dạng bảng xen kẽ theo màu nền **Dark Mode**.

```
<table class="table table-striped table-dark">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>Larry</td>
      <td>the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

| # | First | Last     | Handle   |
|---|-------|----------|----------|
| 1 | Mark  | Otto     | @mdo     |
| 2 | Jacob | Thornton | @fat     |
| 3 | Larry | the Bird | @twitter |

⇒ Theo tài liệu hướng dẫn trên trang Bootstrap từ **v5.2.0**, một số kỹ thuật (*thứ tự thực hiện*) cần được áp dụng để các định dạng hoạt động tốt hơn khi kết hợp với nhau như:

- Đặt nền cho các ô (cell) trong bảng với thuộc tính tùy chỉnh `--bs-table-bg` trước rồi mới đặt thuộc tính cho từng ô để tô màu cho các ô đó.
- Sau đó, thêm định dạng bóng đổ (shadow) vào các ô với `box-shadow: inset 0 0 0 9999px var(--bs-table-bg-state, var(--bs-table-bg-type, var(--bs-table-accent-bg)));` rồi tới `background-color`, nó sẽ ghi đè thuộc tính `box-shadow`. Vì `--bs-table-accent-bg` được đặt mặc định thành `transparent` nên chúng ta không có định dạng bóng đổ mặc định.
- Ngoài ra khi các class `.table-striped`, `.table-striped-columns`, `.table-hover` hoặc `.table-active` được thêm vào, `--bs-table-bg-type` hoặc `--bs-table-bg-state` (theo mặc định được đặt thành `initial`) được đặt thành màu bán trong suốt (`--bs-table-striped-bg`, `--bs-table-active-bg` hoặc `--bs-table-hover-bg`) để tô màu nền và ghi đè mặc định `--bs-table-accent-bg`.
- Đối với các biến thể của bảng, chúng ta sử dụng `--bs-table-accent-bg` tạo màu nền có độ tương phản cao nhất dựa trên màu đó. Ví dụ: màu sẽ đậm hơn với `.table-primary` so với `.table-dark` thì màu nhẹ hơn.
- Màu văn bản (text) và đường viền (border) cũng áp dụng như cách trên và màu của chúng được kế thừa mặc định.

d. *Modal: (tạo hộp thoại/cửa sổ thông báo)*

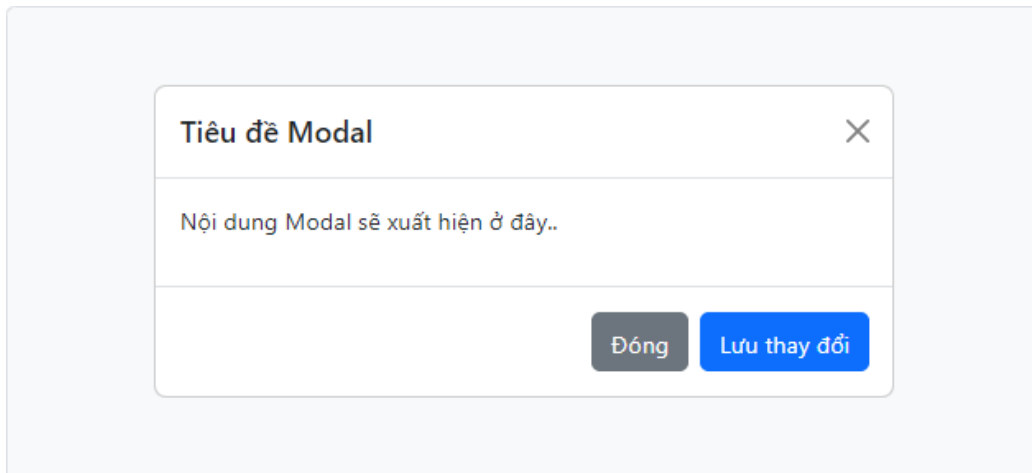
Cần lưu ý khi sử dụng các thành phần Modal của Bootstrap sau đây:

- Modal được xây dựng bằng HTML, CSS và JavaScript để tác động tới mọi phần tử trong trang. Nó loại bỏ chức năng cuộn (scroll) trong thẻ <body> thay bằng cuộn trong nội dung bên trong nó.
- Mặc định khi click bên ngoài hộp thoại (*Modal*) sẽ tự động đóng.
- Bootstrap chỉ hỗ trợ một hộp thoại/cửa sổ thông báo tại một thời điểm.
- Modal sử dụng `position: fixed` để cố định vị trí hiển thị. Nên đặt vị trí cấp cao nhất để tránh các phần tử khác đè lên.
- Với thiết bị di động cần phải lưu ý khi sử dụng `position: fixed`. Chi tiết sinh viên xem thêm tại [Modals and dropdowns on mobile #](#).
- Với HTML5, thuộc tính [autofocus](#) không có tác dụng trong các modal của Bootstrap. Có thể sử dụng một số tùy chỉnh JavaScript sau:

```
var myModal = document.getElementById('myModal')
var myInput = document.getElementById('myInput')

myModal.addEventListener('shown.bs.modal', function () {
  myInput.focus()
})
```

```
<div class="modal" tabindex="-1">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Tiêu đề Modal</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <p>Nội dung Modal sẽ xuất hiện ở đây..</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Đóng</button>
        <button type="button" class="btn btn-primary">Lưu thay đổi</button>
      </div>
    </div>
  </div>
</div>
```



Bên trên là một ví dụ về *static modal*. Giá trị thuộc tính **position** và **display** đã bị ghi đè bao gồm phần tiêu đề (*header*), nội dung (*body*) và chân trang (*footer*). Ở ví dụ trên class **.modal-title** khai báo trong thẻ `<h5>` để tránh vấn đề phân cấp tiêu đề. Lý tưởng nhất nên sử dụng thẻ `<h1>`.

Ngoài ra ta cũng có thể sử dụng các class **font-size** để định dạng như: **fs-1, fs-2, fs-3, fs-4, fs-5, fs-6**.

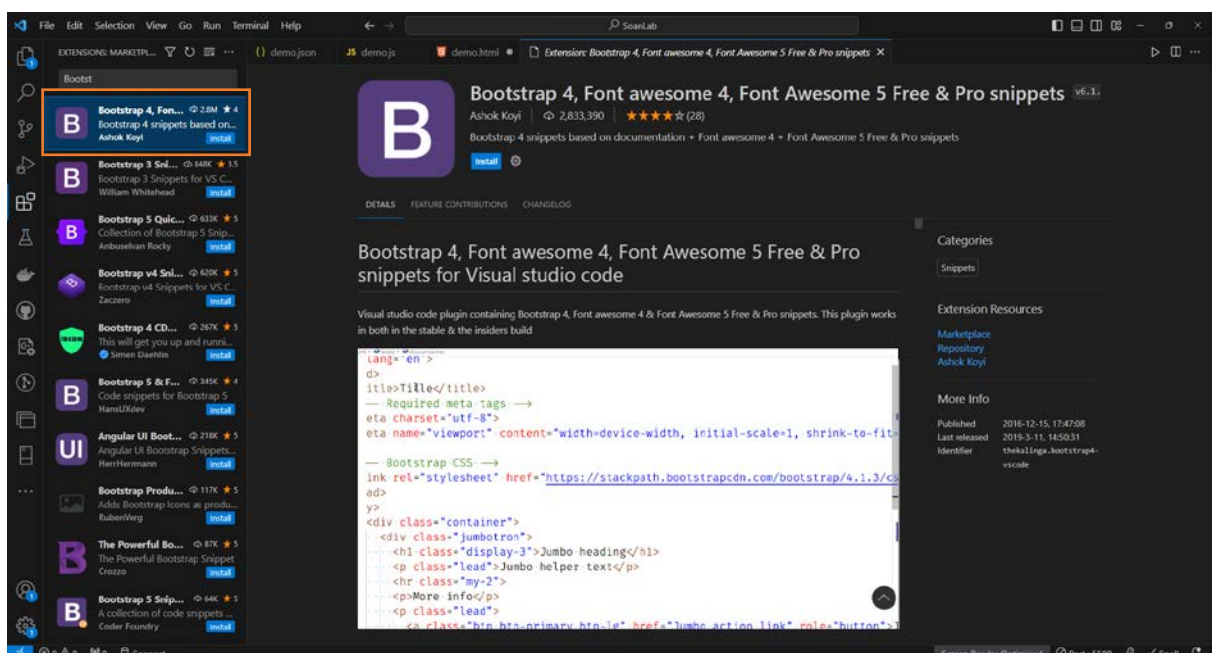
```
<h1 class="modal-title fs-5" >Modal title</h1>
```

#### 4. Sử dụng Extension visual code.

Cài đặt extension Bootstrap rất dễ dàng, bấm tổ hợp phím **Ctrl + Shift + X** và nhập từ khóa *Bootstrap* => chọn **Bootstrap 4, Font awesome 4, Font Awesome 5 Free & Pro snippets for Visual studio code** => **Install**

**Link tham khảo:**

<https://marketplace.visualstudio.com/items?itemName=thekalinga.bootstrap4-vscode>



## 5. Thời gian thực hành

120 phút

## 6. Đánh giá

Thang điểm tối đa: 10 điểm/Lab

## 7. Yêu cầu thực hành

Hãy thiết kế cho mình một trang web đơn giản sử dụng **Bootstrap**. Trong đó trang web phải có đầy đủ bố cục gồm: (**header**, **body** và **footer**).

- **Header:** Menu, Logo, nút Login... (2.5đ)
- **Body:** Hình Banner, search, content, chuẩn bị ít nhất 3 bức hình... (2.5đ)
- **Footer:** Logo, text (địa chỉ, email, map, ...), các logo ngân hàng liên kết (*nếu là web bán hàng hoặc dịch vụ*) (2.5đ).
- **Responsive** (2.5đ).

Tham khảo: [Lab10/Vi\\_du/index.html](#).



Logo TRANG CHỦ DỊCH VỤ THỊ TRƯỜNG ĐẦU TƯ LIÊN HỆ

### DỊCH VỤ

Những thứ chúng tôi đã cung cấp

|  |  |  |
|--|--|--|
| <br><b>POWER</b><br>Khách hàng rất vui.. | <br><b>LOVE</b><br>Khách hàng rất vui..      | <br><b>JOB DONE</b><br>Khách hàng rất vui..  |
| <br><b>GREEN</b><br>Khách hàng rất vui.. | <br><b>CERTIFIED</b><br>Khách hàng rất vui.. | <br><b>HARD WORK</b><br>Khách hàng rất vui.. |

localhost/00-TH NMWeb/Soan-lab-2023/Lab10/Vi\_du/#services

Logo TRANG CHỦ DỊCH VỤ THỊ TRƯỜNG ĐẦU TƯ LIÊN HỆ

### THỊ TRƯỜNG


Chọn gói thanh toán phù hợp với bạn

|   |   |  |
|---|---|--|
| <b>Cơ bản</b><br>20 Điểm VIP<br>15 Dịch vụ<br>5 Yêu cầu<br>2 Coin<br>Độc quyền<br><b>100.000đ</b><br>1 tháng<br><a href="#">Đăng Kí</a> | <b>Tiến bộ</b><br>50 Điểm VIP<br>25 Dịch vụ<br>10 Yêu cầu<br>5 Coin<br>Độc quyền<br><b>250.000đ</b><br>6 tháng<br><a href="#">Đăng Kí</a> | <b>Phát triển</b><br>100 Điểm VIP<br>50 Dịch vụ<br>25 Yêu cầu<br>10 Coin<br>Độc quyền<br><b>500.000đ</b><br>1 năm<br><a href="#">Đăng Kí</a> |
|---|---|--|


Logo      TRANG CHỦ      DỊCH VỤ      THI TRƯỜNG      **ĐẦU TƯ**      LIÊN HỆ

### ĐẦU TƯ


Những nhà đầu tư hàng đầu



**VIETCOMBANK**  
Vâng, chúng tôi đã xây dựng VIETCOMBANK



**VIB**  
Vâng, chúng tôi đã xây dựng VIB



**TECHCOMBANK**  
Vâng, chúng tôi đã xây dựng TECHCOMBANK

### KHÁCH HÀNG ĐÃ NÓI GÌ VỀ CHÚNG TÔI

◀      "Một từ thôi... **CHÀ!**"  
Nguyễn Văn B      ▶

○ ● ○

### LIÊN HỆ

Hãy liên hệ với chúng tôi và chúng tôi sẽ liên hệ lại với bạn trong vòng 240 giờ.

📍 HCM, VN  
☎ +84 1515151515  
✉ myemail@something.com