**HTML - Form controls**

**1.1.1 Các control phổ biến**

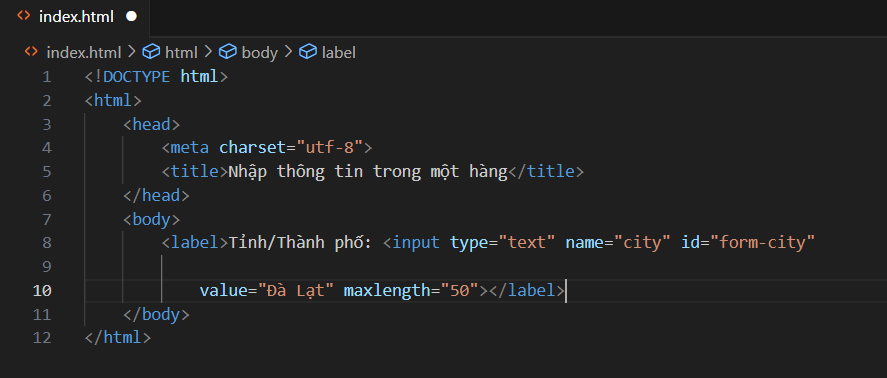
**Control nhập văn bản**

Một trong những công việc phổ biến nhất khi người dùng làm việc với web form là nhập thông tin dạng văn bản. Người dùng có thể nhập thông tin trong một hàng hoặc nhiều hàng.

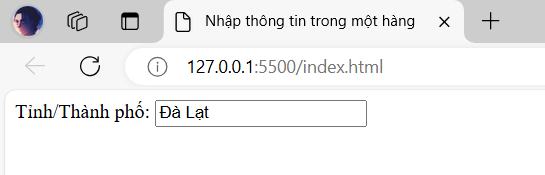
Nhập thông tin trong một hàng

Để người dùng có thể nhập thông tin trong một hàng, chúng ta sử dụng control có tên là input với thuộc tính type=“text”, đây là thuộc tính mặc định cho control input, nghĩa là nếu bỏ qua thuộc tính type=“text” thì control input sẽ cho phép người dùng nhập thông tin trong một hàng.

Ví dụ,



Kết quả:



Giải thích thêm về đoạn mã trên,

– name: là thuộc tính bắt buộc phải có, chứa tên của biến (city)

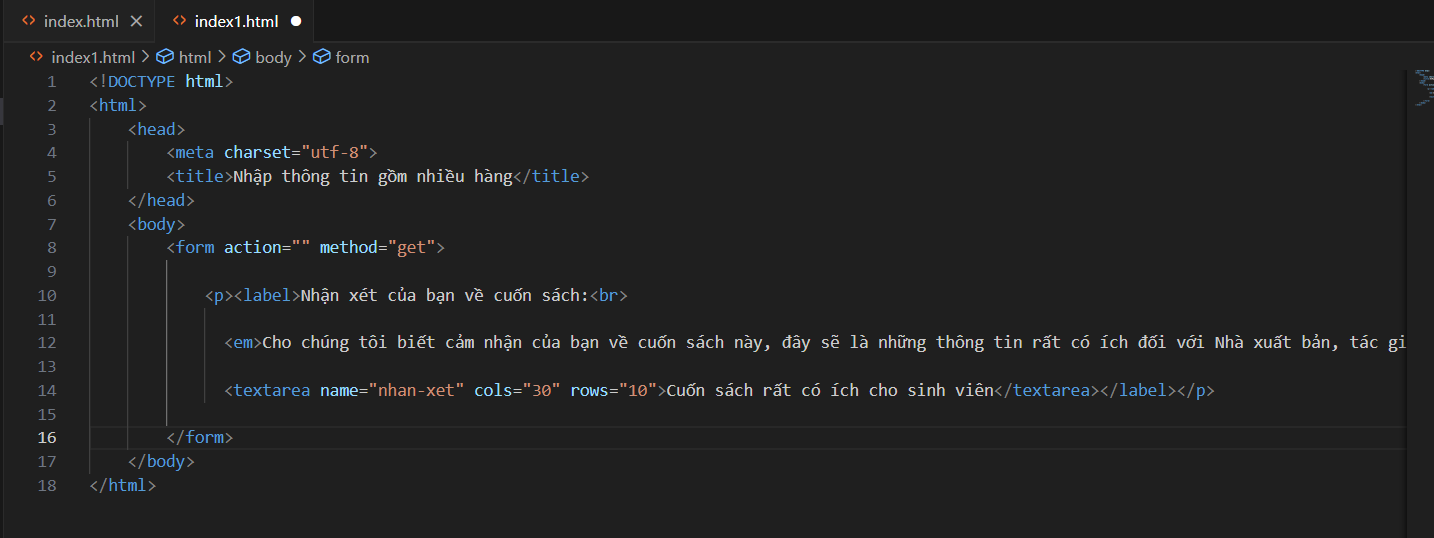
– value: là thuộc tính chứa nội dung văn bản sẽ được thiết lập mặc định khi form được hiển thị, hoặc sau khi reset form (Đà Lạt)

– maxlength: mặc định, control này cho phép người dùng nhập số kí tự không giới hạn, tuy nhiên, lập trình viên có thế giới hạn số lượng kí tự được phép nhập bằng thuộc tính maxlength(50).

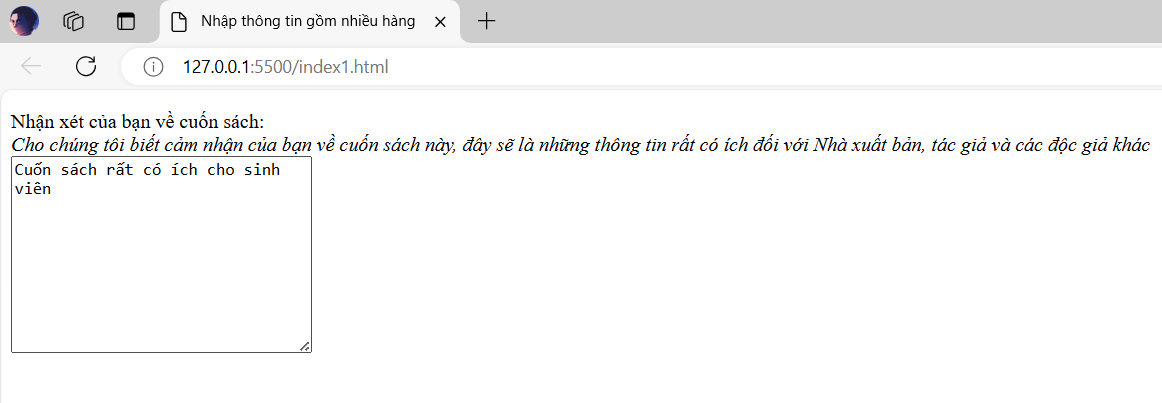
**Nhập thông tin gồm nhiều hàng**

Nếu người dùng cần nhập thông tin gồm nhiều hàng dữ liệu, thì chúng ta sử dụng control textarea. Khác với phần tử input, phần tử textarea cho phép đặt nội dung giữa thẻ mở và thẻ đóng, phần nội dung này sẽ được hiển thị trong trường nhập văn bản và được gửi về web server nếu người dùng gửi form (submit), vì vậy nên cẩn thận về điều này. Thực tế, các lập trình viên thường để trống phần nội dung, mà chỉ hiển thị thông tin gợi ý bằng thuộc tính title hoặc placeholder.

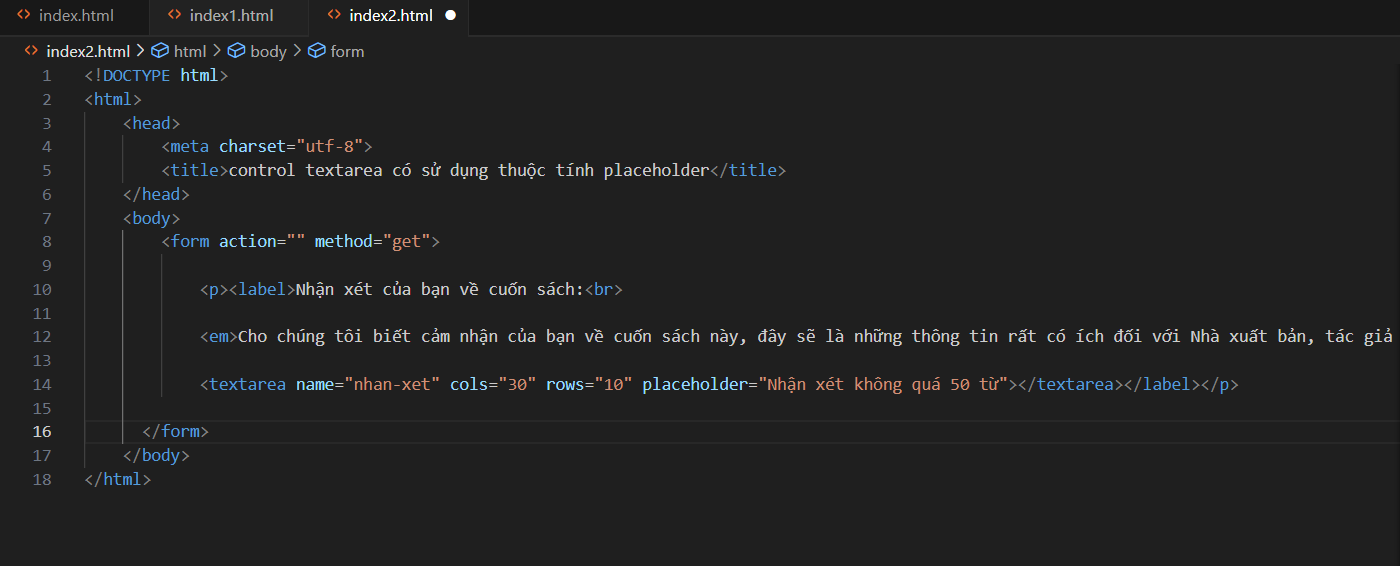
Ví dụ, control textarea có nội dung giữa thẻ mở và thẻ đóng,



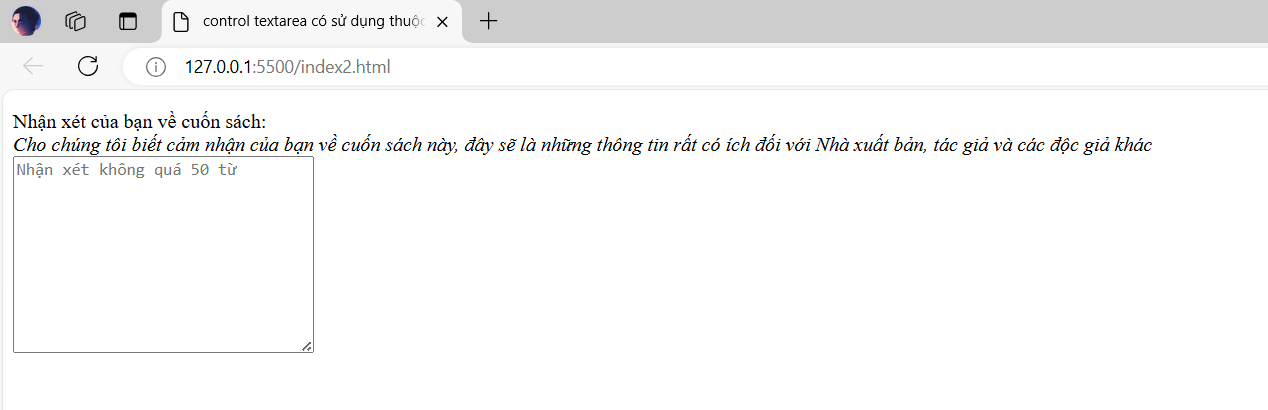
Kết quả:



Ví dụ, control textarea có sử dụng thuộc tính placeholder,



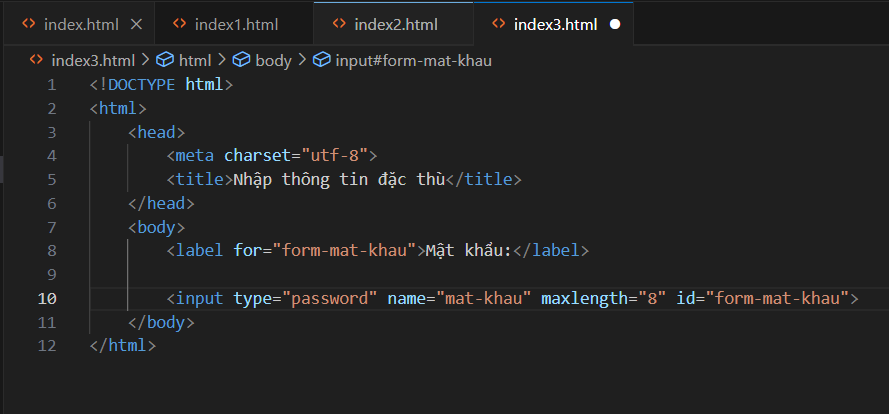
Kết quả:



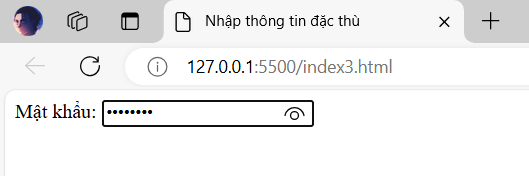
**Nhập thông tin đặc thù**

Bên cạnh control cho phép nhập thông tin trên một hàng thông thường, với phần tử input và thuộc tính type=“text”, HTML còn cung cấp một số control cho phép nhập các thông tin có tính đặc thù, như: mật khẩu (password), từ khóa tìm kiếm (search terms), địa chỉ email (email addresses), số điện thoại (telephone number) và các URL.

– Nhập mật khẩu



Kết quả:



**Phần tử datalist**

Phần tử datalist cho phép lập trình viên tạo ra một trình đơn xổ xuống gồm các giá trị đã được thiết lập sẵn, người dùng chỉ việc lựa chọn giá trị phù hợp.

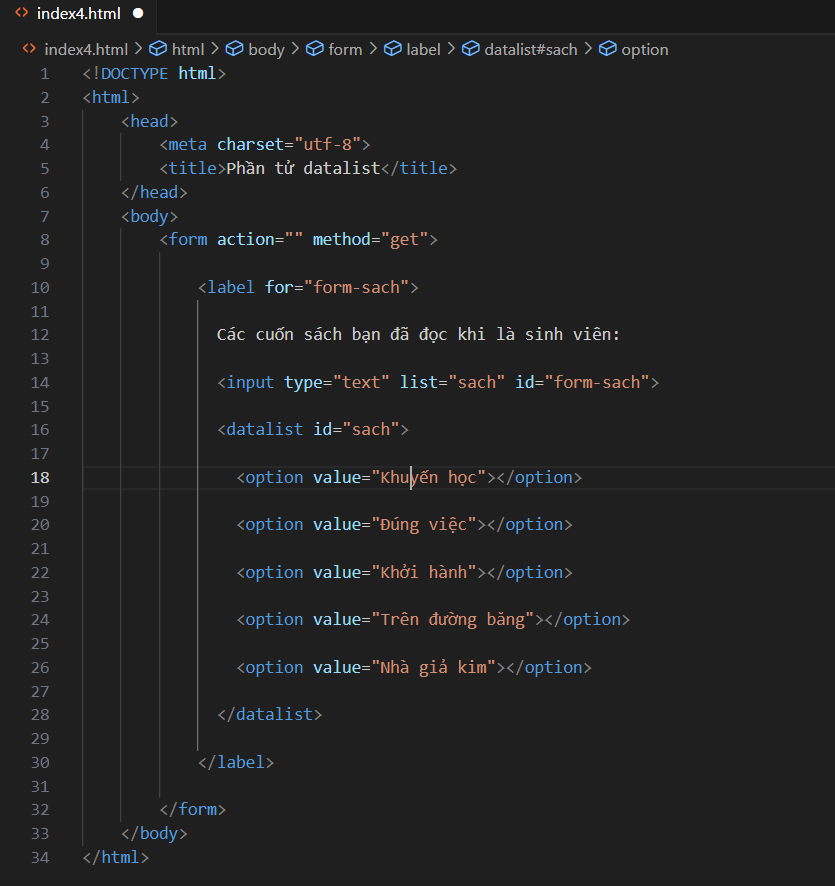
Tuy nhiên, nếu trong trình đơn không có giá trị phù hợp với người dùng thì sao? Giải pháp là kết hợp phần tử datalist và phần tử input với type=“text”. Cách làm:

– Tạo một input với type=“text”

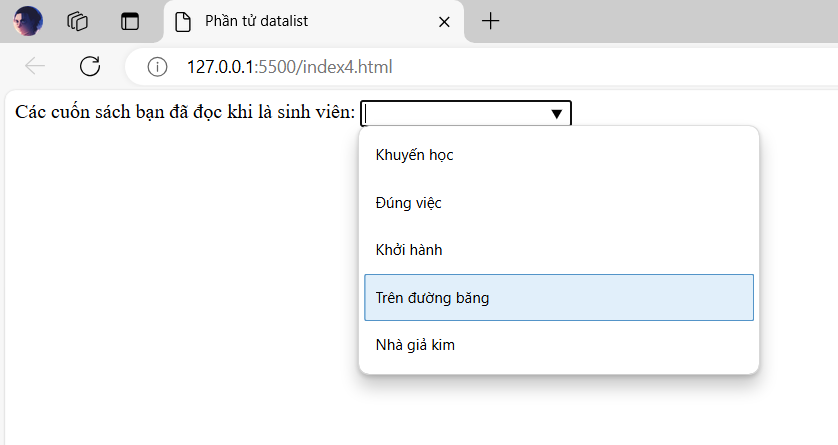
– Tạo một datalist, trong datalist, mỗi mục chọn ứng với một phần tử option

– Trong input, thêm thuộc tính list trỏ tới id của datalist

Xem ví dụ,



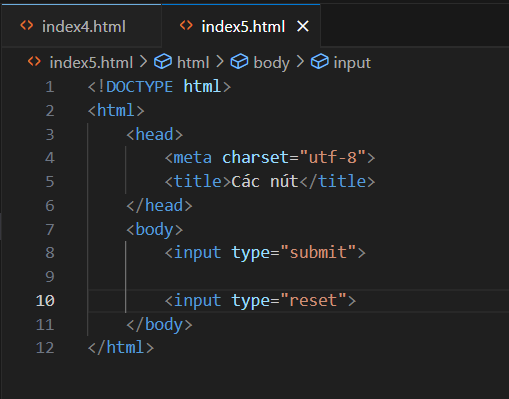
Kết quả:



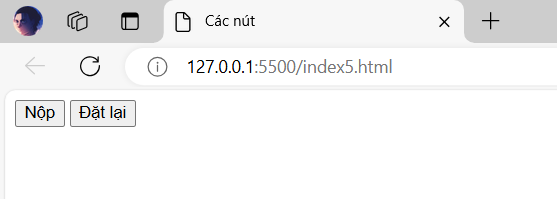
**Các nút**

Có thể chèn vào web form một số loại nút khác nhau, trong đó quan trọng nhất là nút submit. Submit có nghĩa là gửi đi hoặc đệ trình. Khi người dùng bấm hoặc chạm vào nút này, toàn bộ dữ liệu của form sẽ được gửi về web server để xử lý.

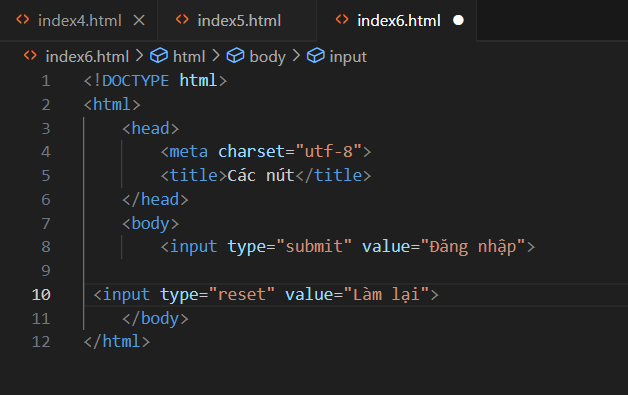
Ví dụ, hai nút với tên mặc định,



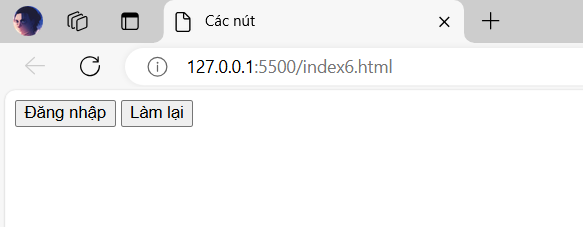
Kết quả:



Ví dụ, hai nút với tên được thiết lập trong thuộc tính value,



Kết quả:

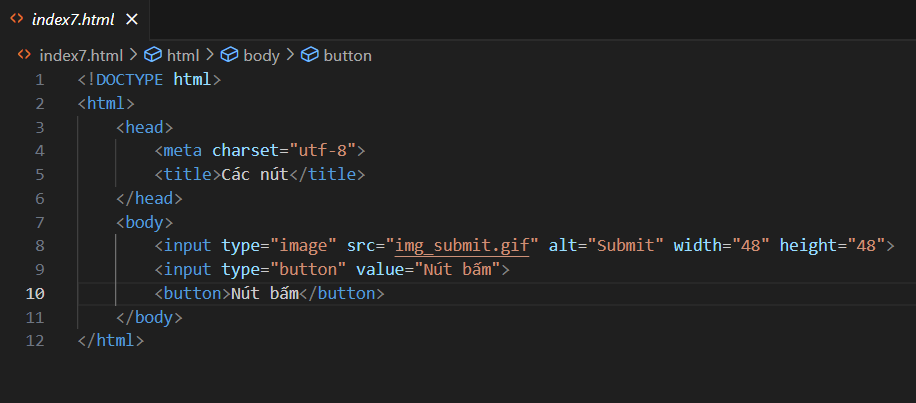


Ngoài hai nút trên, cũng có thể chèn vào form ba loại nút sau:

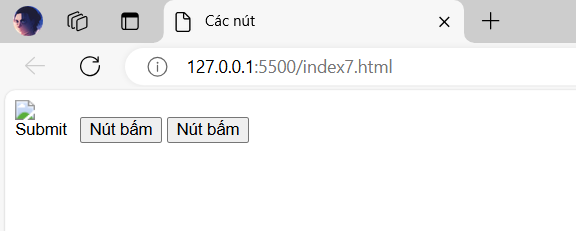
– Nút hình ảnh, nút này có chức năng tương tự như nút submit, chỉ khác là phần hiển thị của nút chính là một hình ảnh, cần thiết lập giá trị của thuộc tính alt cho phù hợp, phòng trường hợp không tải được hình ảnh. Ví dụ,

– Nút tùy biến, thiết lập thuộc tính của input là type=“button” để tạo ra một nút cho phép tùy biến bằng JavaScript, nút này chưa được thiết lập chức năng cụ thể. Ví dụ,

– Phần tử button, đây là phần tử giúp tạo nút một cách linh hoạt tương tự như phần tử input, nội dung nằm giữa thẻ <button> và </button> (văn bản hoặc hình ảnh) sẽ được hiển thị trên nút. Ví dụ,



Kết quả:



**Nút radio**

Để tạo nút radio, sử dụng input với thuộc tính type=“radio”, cú pháp là,

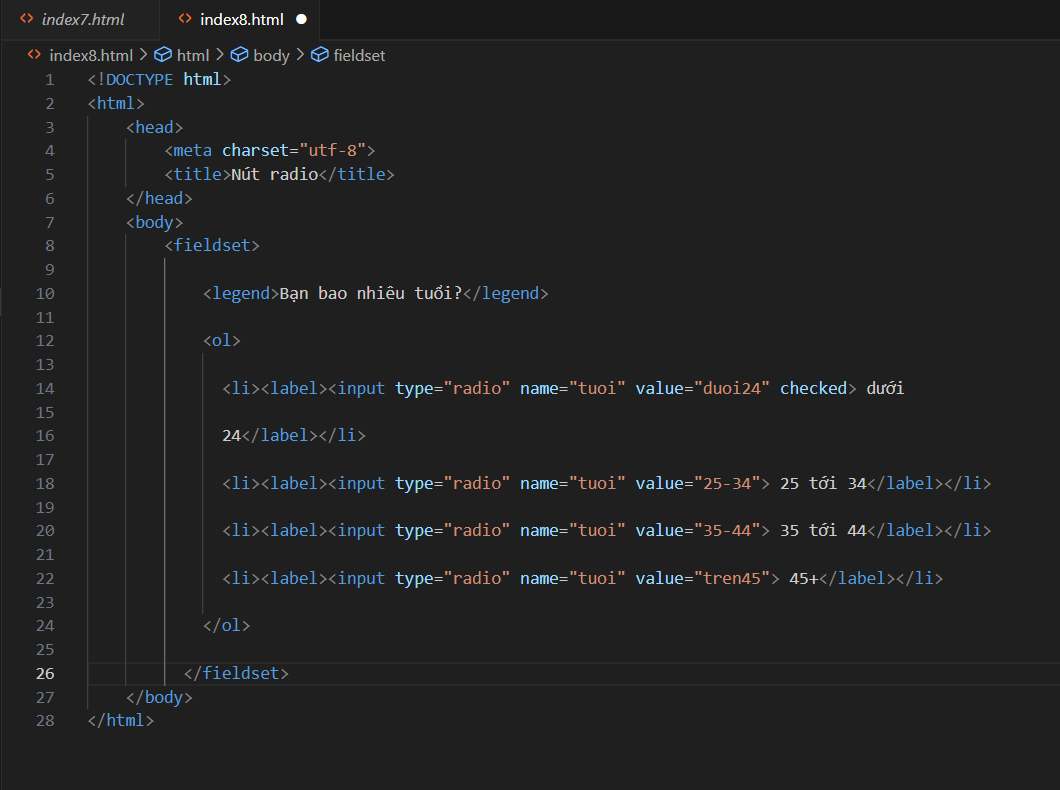
<input type="radio" name="variable" value="value">

Trong đó,

– Thuộc tính name là bắt buộc, sử dụng thuộc tính này để kết buộc (bind) các radio lại với nhau, những radio có giá trị name giống nhau sẽ tạo thành một nhóm, các nút chọn trong nhóm này có tính loại trừ nhau. Giá trị của thuộc tính name cũng là tên biến để server tham chiếu tới, nhằm lấy giá trị (value).

– Thuộc tính value chứa giá trị sẽ được gửi về server nếu nút radio tương ứng được chọn. Mỗi radio có một giá trị value khác nhau.

Ví dụ, hiển thị các mục chọn về độ tuổi của người dùng,



Kết quả:



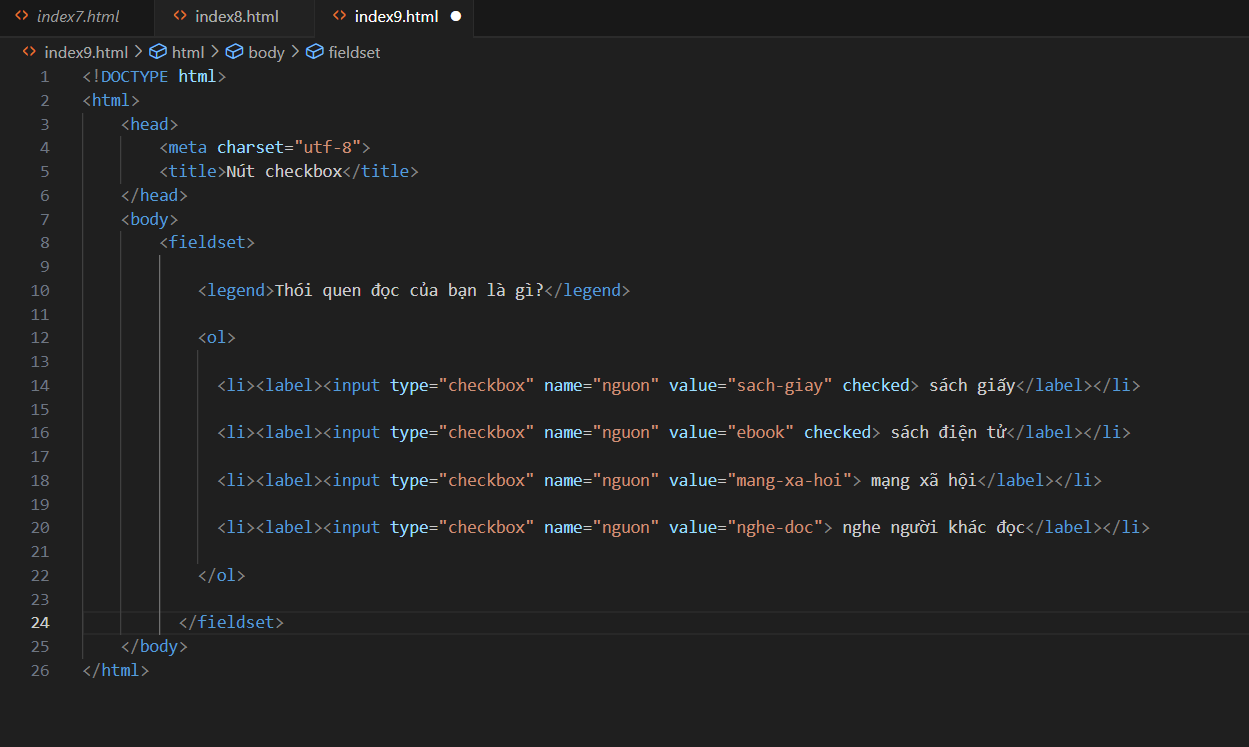
**Nút checkbox**

Để tạo nút checkbox, sử dụng input với thuộc tính type=“checkbox”, cú pháp là,

<input type="checkbox" name="variable" value="value">

Cũng giống như nút radio, các checkbox sử dụng giá trị thuộc tính name giống nhau để tạo thành một nhóm, chỉ khác là checkbox cho phép chọn cùng lúc nhiều hơn một mục. Khi gửi form, checkbox nào được chọn thì giá trị (value) của nó sẽ được gửi về server.

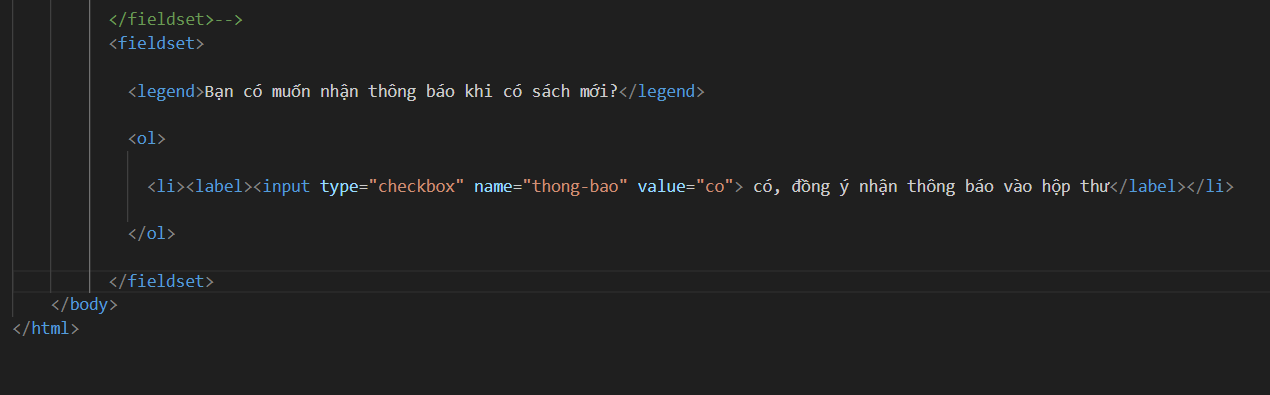
Ví dụ,



Kết quả:



Tất nhiên, checkbox không nhất thiết phải sử dụng theo nhóm, mà có thể dùng nó một mình. Ví dụ,



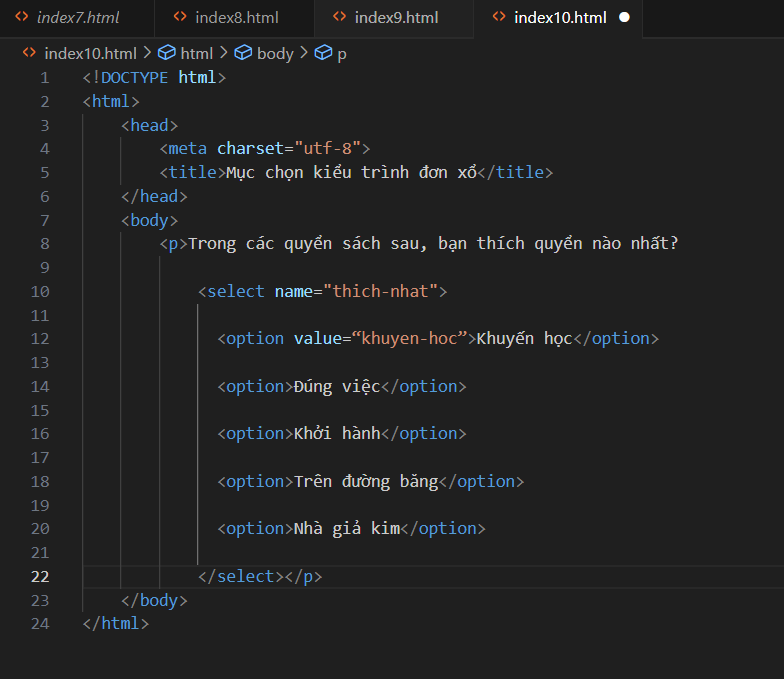
Kết quả:



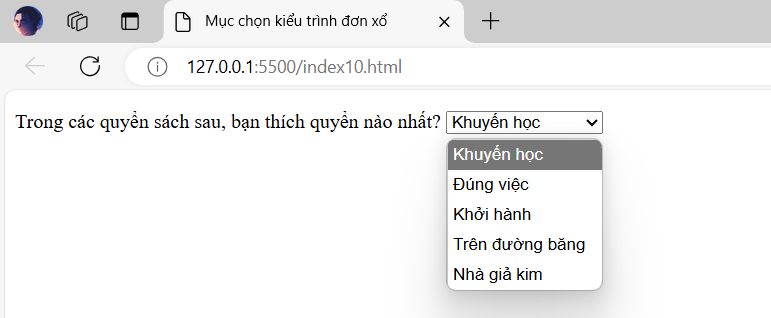
**Mục chọn kiểu trình đơn xổ**

Ở chế độ mặc định, phần tử select được hiển thị dưới dạng một trình đơn xổ, khi đó giá trị của thuộc tính size sẽ là 1. Thuộc tính size được sử dụng để xác định số mục mà người dùng được phép chọn trong trình đơn. Vậy, trong trình đơn xổ, người dùng chỉ được phép chọn một mục tại một thời điểm.

Ví dụ,



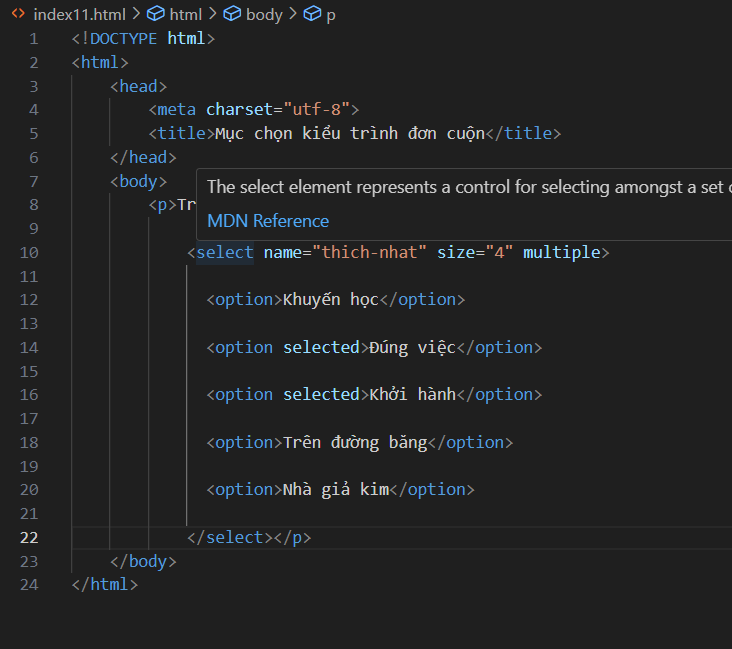
Kết quả:



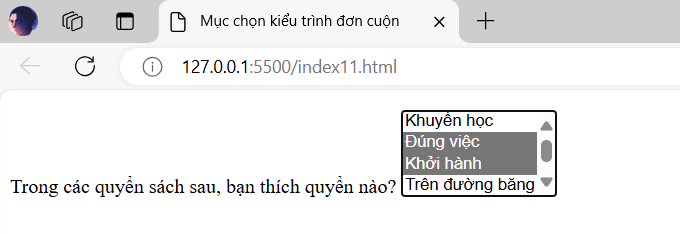
**Mục chọn kiểu trình đơn cuộn**

Cách tạo “mục chọn kiểu trình đơn cuộn” cũng tương tự như “mục chọn kiểu trình đơn xổ”, chỉ khác là giá trị của thuộc tính size phải lớn hơn 1.

Ví dụ,

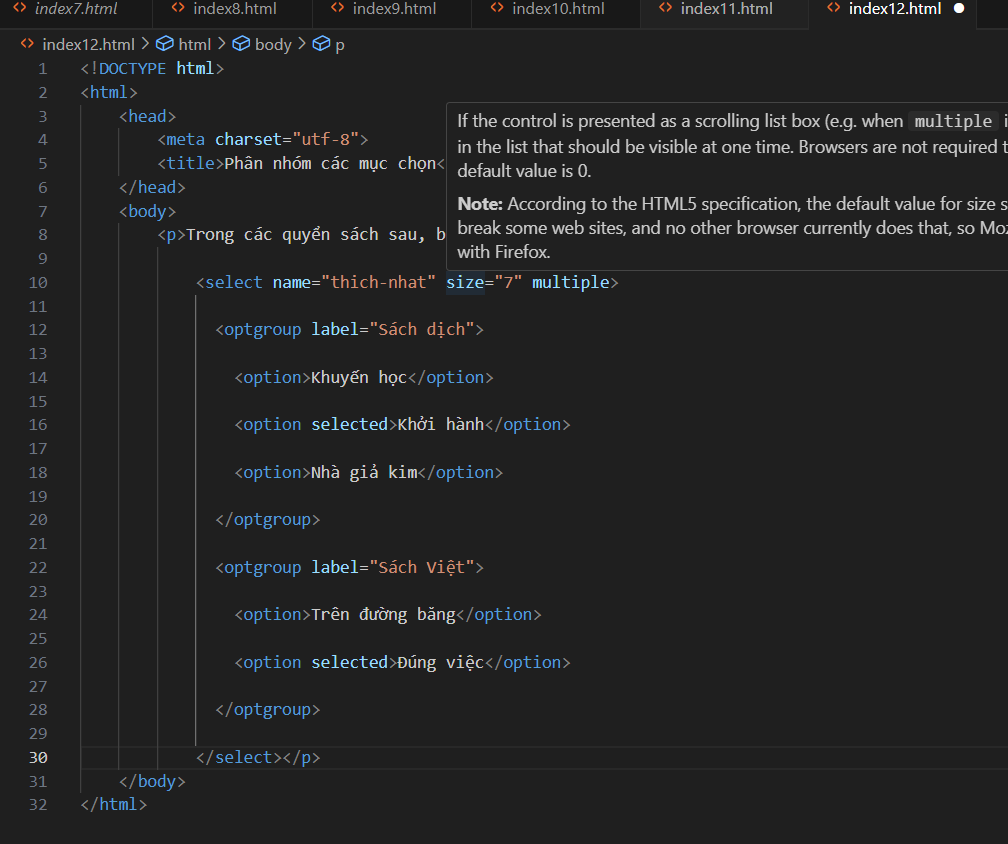


Kết quả:

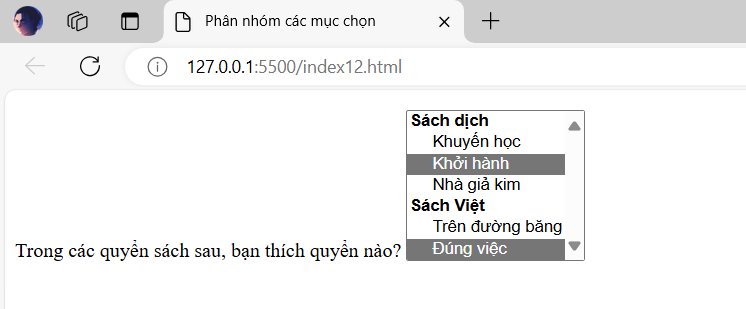


**Phân nhóm các mục chọn**

Để phân nhóm các mục chọn, sử dụng phần tử optgroup, tên của nhóm được đặt trong thuộc tính label (đừng nhầm thuộc tính label với phần tử label). Ví dụ,



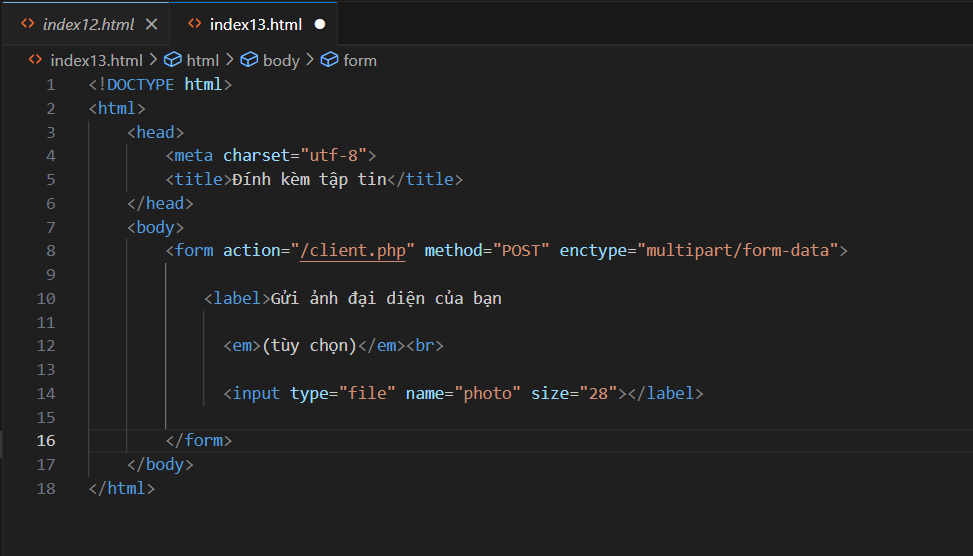
Kết quả:



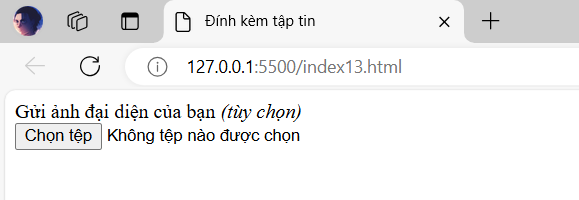
**Đính kèm tập tin**

Web form không chỉ cho người dùng nhập thông tin, mà nó còn cho đính kèm các tập tin từ đĩa của người dùng. Ví dụ, trang web của cửa hàng in ấn sẽ cho khách hàng gửi các tập tin cần in, hoặc một tạp chí sẽ cho người dùng đăng ảnh dự thi.

Sử dụng input với type=“file” để tạo một control cho người dùng đính kèm tập tin. Ví dụ,



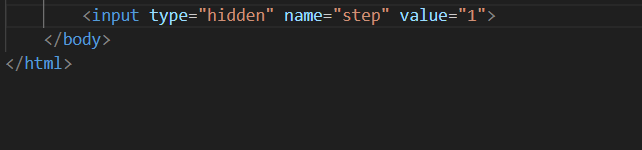
Kết quả:

**Control ẩn**

Trong một số trường hợp, bên cạnh dữ liệu do người dùng nhập vào, lập trình viên cũng muốn gửi thêm dữ liệu về chương trình/ứng dụng web bên server. Để làm việc này, chúng ta có thể sử dụng control ẩn. Với control ẩn, dữ liệu sẽ được gửi về server khi người dùng gửi form (submit), tuy nhiên, control ẩn sẽ không được hiển thị ra ngoài giao diện.

Để tạo ra control ẩn, sử dụng phần tử input với thuộc tính type=“hidden”. Mục đích duy nhất của control ẩn là gửi một cặp dữ liệu, có kiểu name/value về server khi người dùng gửi form.

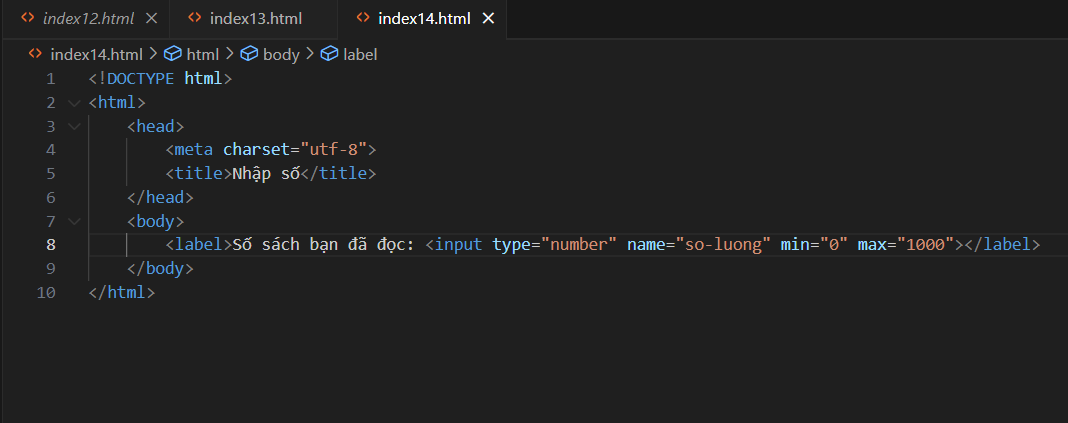
Ví dụ,



**Nhập số**

Để tạo control nhập số, HTML5 sử dụng phần tử input với type là number hoặc range.

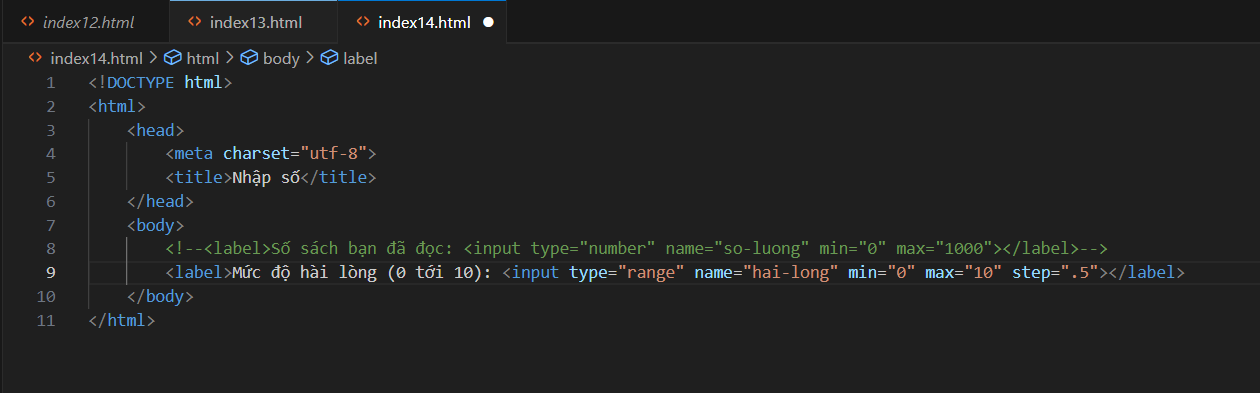
Ví dụ một control nhập số với type=“number” và giá trị nhập vào phải nằm trong khoảng [0, 1000],



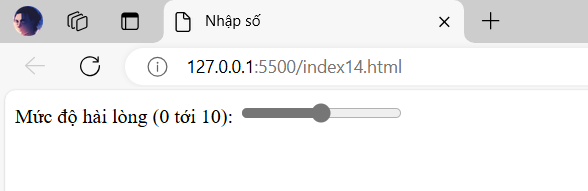
Kết quả:



Ví dụ một control nhập số với type=“range” và giá trị nằm trong khoảng [0, 10], độ tăng/giảm mỗi mức của thanh trượt là 0.5.



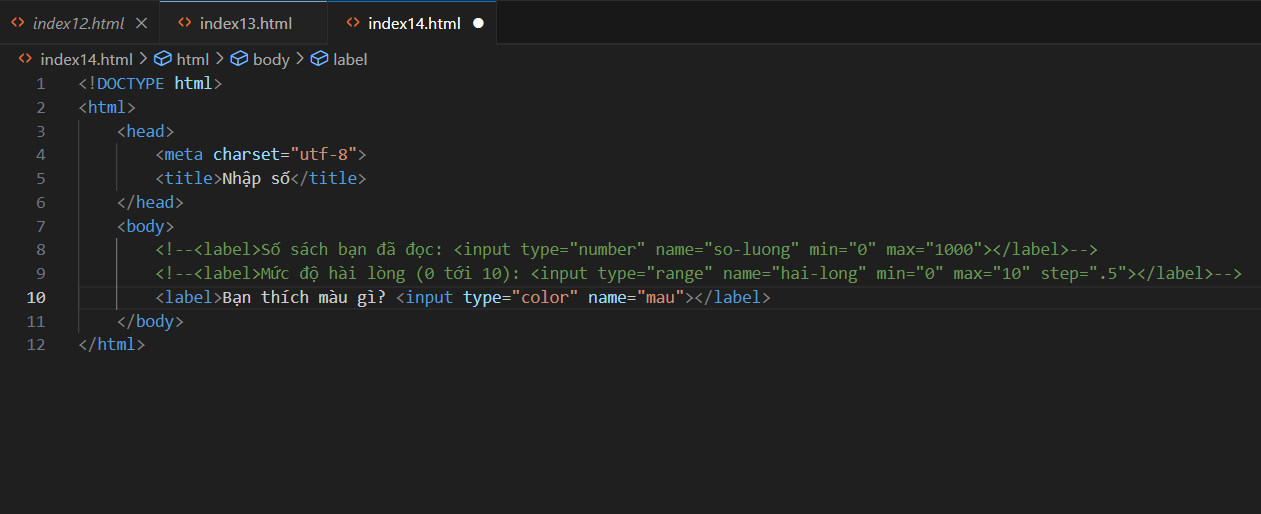
Kết quả:



**Bảng màu**

Để hiển thị một bảng màu cho phép người dùng chọn từng màu cụ thể, HTML5 sử dụng phần tử input với type=“color”. Giá trị màu được xác định bằng RGB hệ 16 (#CCAA23).

Ví dụ,



Kết quả:

