CSE485 – Công nghệ Web

Bài 04: PHP Form

dungkt@tlu.edu.vn

COMPUTER SCIENCE
AND ENGINEERING
THUYLOI UNIVERSITY

Nội dung

1) Cơ bản về Form
2) Form Upload File
3) Form – Các vấn đề bảo mật
4) Session và Cookie

1. Cơ bản về Form

Form: Khái niệm

- Form là thành phần không thể thiếu trong các ứng dụng Web vì form là nơi trao đổi và thu thập dữ liệu từ người dùng
- Form có 2 loại chính:
  - Form lấy thông tin dưới dạng căn bản
  - Form lấy thông tin dưới dạng các file được upload lên
- Có thể không sử dụng form để lấy thông tin từ người dùng không ?
  - Có thể, sử dụng kỹ thuật Ajax
- Hai phương thức thường được sử dụng khi gửi dữ liệu từ form là GET và POST
- Server phân biệt data gửi lên dựa vào thuộc tính name của các thẻ input form

Form: Khai báo

- Cú pháp khai báo HTML

  <form action='<url-xử-lý>' method='<tên-method>' >

  <các-thẻ-input-html>

  </form>

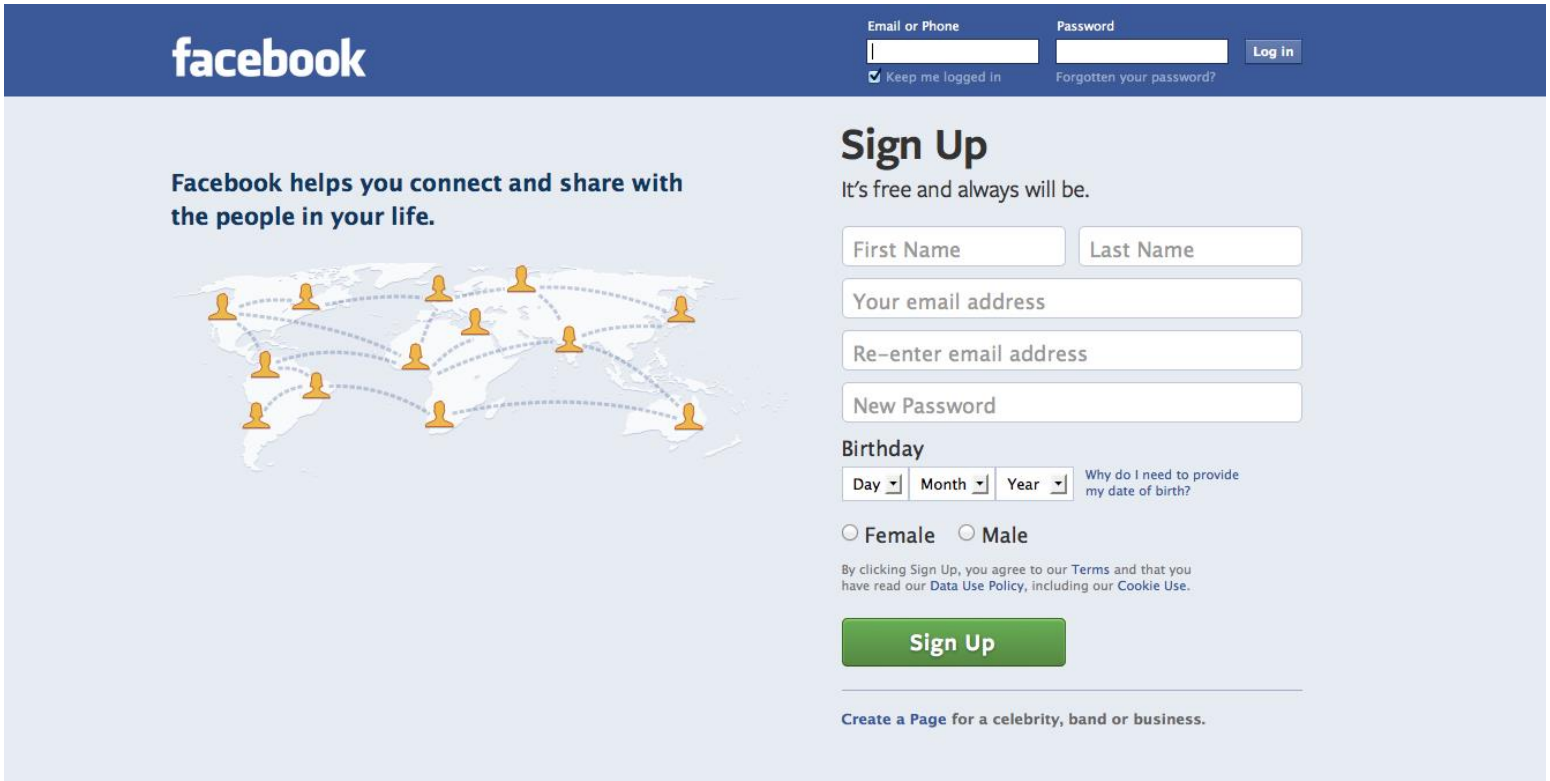
- Trong đó:

  form, action, method là các từ khóa

  <url-xử-lý>: url dùng để xử lý các dữ liệu được gửi lên từ form

  <tên-method>: phương thức truyền dữ liệu (POST/GET...)

Form: Ví dụ


facebook
Email or Phone
Password
Log in
Keep me logged in
Forgotten your password?
Facebook helps you connect and share with the people in your life.
Sign Up
It's free and always will be.
First Name
Last Name
Your email address
Re-enter email address
New Password
Birthday
Day
Month
Year
Why do I need to provide my date of birth?
Female
Male
By clicking Sign Up, you agree to our Terms and that you have read our Data Use Policy, including our Cookie Use.
Sign Up
Create a Page for a celebrity, band or business.

Form: Danh sách Input

Component	Sample	HTML
Text Entry		<input type="text">
Password Entry		<input type="password">
Checkbox		<input type="checkbox">
Radio Button		<input type="radio">
Submit Button	Submit Data	<input type="submit" value="Submit Data">
Reset Button	Reset	<input type="reset" value="Reset">
File Selection Entry	Choose File No file chosen	<input type="file">
Text Area (multiple lines)		<textarea cols="20" rows="2">
Select Menu (pop-up)	Red	<select size="1">
Select Menu (scrolling)	Red Blue Green	<select size="3">

Form: Lưu ý đặt tên phần tử Input

- Với các input mà chỉ có 1 giá trị duy nhất tại 1 thời điểm, thì thuộc tính name của input sẽ ở dạng đơn, VD: input text, password, textarea, radio, select ở trạng thái đơn, file ở trạng thái đơn
- Với các input mà có thể có nhiều giá trị tại 1 thời điểm, thì thuộc tính name của input đó sẽ ở dạng mảng. VD: checkbox, select kiểu multi, select file kiểu multi.
  - VD: <input type='checkbox' name='gender[ ]' />
- Để input select và file có thể ở trạng thái chọn multi, cần thêm thuộc tính multiple vào cho thẻ
- Với riêng kiểu input file, sẽ có cách riêng để xử lý lấy dữ liệu từ file, được trình bày trong buổi học sau

Form: Phương thức GET

- Phương thức GET gửi dữ liệu thông qua URL trên trình duyệt
- Các biến truyền lên có format: <URL>?param1=value1&param2=value2 ...
  - VD: http://google.com.vn?name=Anh&age=18
- Thường sử dụng cho truy vấn lấy dữ (GET) liệu từ server
- GET giới hạn độ dài chuỗi dữ liệu trên URL là 1024 ký tự
- PHP sử dụng biến $_GET lưu trữ toàn bộ dữ liệu gửi lên thông qua phương thức GET
- Không nên sử dụng GET để gửi các thông tin nhạy cảm như password...
- Dữ liệu gửi lên chỉ tồn tại khi submit form, vì vậy cần kiểm tra bằng hàm isset() để chắc chắn rằng đã submit form

Form: Phương thức POST

- Có tính bảo mật hơn GET, dữ liệu truyền đi không hiển thị trên trình duyệt
- Thường sử dụng cho các truy vấn thay đổi dữ liệu trong database(INSERT, UPDATE, DELETE)
- Không giới hạn độ dài dữ liệu như GET
- Tương tự như GET, PHP sử dụng biến $_POST được dùng để lưu trữ toàn bộ dữ liệu gửi lên thông qua phương thức POST

Form: Biến $_REQUEST

- Có thể được sử dụng để lấy dữ liệu gửi lên từ form của cả 2 phương thức GET và POST
- Chứa tất cả thông tin về các biến $_GET, $_POST và $_COOKIE ($_COOKIE là biến quản lý liên quan đến cookie trình duyệt)
- Thông thường trong form đã chỉ định cụ thể tên phương thức, nên sẽ ưu tiên sử dụng biến $_GET hoặc $_POST thay vì $_REQUEST

Form: Validate

- Là bước kiểm soát dữ liệu gửi lên từ user nhằm tăng tính bảo mật và tránh việc mất thời gian xử lý các dữ liệu rác.
- Là bước không thể thiếu khi xử lý form
- Các lỗi nhập từ người dùng phổ biến là dữ liệu trống và nhập sai định dạng
- Quá trình validate dữ liệu thường gồm 2 bước:
    1. Lấy dữ liệu từ user khi submit form và kiểm tra xem dữ liệu là hợp lệ hay chưa
    2. Nếu chưa hợp lệ, thông báo lỗi tới người dùng, đồng thời giữ nguyên các giá trị các trường mà user đã nhập đúng để tránh cho việc user phải nhập lại
    3. Có thể validate bằng Javascript, tuy nhiên trên server vẫn cần validate lại một lần nữa

Form: Submit

- Tùy thuộc vào phương thức gửi dữ liệu lên là gì (POST/GET) để gọi biến $_POST/$_GET tương ứng
- Server chỉ xử lý logic dữ liệu từ user sau khi đã hoàn thành bước validate dữ liệu
- Form có thể được submit bằng cách click input submit hoặc nhấn Enter

2. Form Upload File

Form Upload: Tổng quan

- Khai báo form cần thêm thuộc tính enctype="multipart/form-data“ để cho phép form upload file
- Phương thức của form phải là POST
- Tương tự $_POST, $_GET hay $_REQUEST, biến toàn cục dùng lưu trữ thông tin file upload lên là $_FILES
- Thẻ HTML dùng để chọn file upload là <input type=file />
- Trong trường hợp upload nhiều file, cần đặt name dạng mảng, và thêm thuộc tính multiple trong input file. Xử lý upload file với trường hợp này sẽ là xử lý trên mảng các file.

Form Upload: Cấu trúc biến $_FILES

- Là mảng 2 chiều có dạng như sau **$_FILES['nameInputFile']['properties']**
  - **nameInputFile**: thuộc tính name của thẻ HTML input file
  - **properties**: đại diện cho 5 thuộc tính sau:
    - name: tên file được upload
    - type: kiểu dữ liệu của file như png, jpg, xlsx v.v
    - tmp_name: đường dẫn tạm thời mà file đang được lưu trên server
    - size: kích cỡ của file upload, đơn vị là Byte
    - error: trạng thái lỗi trong qua trình upload file, ở dạng số nguyên, 1 số mã lỗi thường gặp:

      0 - không có lỗi, hay upload thành công
      1 - Báo lỗi file upload vượt quá dụng lượng cho phép
      2 – Báo lỗi số file upload vượt quá số lượng file cho phép của form
      3 – Báo lỗi file chỉ được upload 1 phần, không phải toàn bộ file
      4 – Báo lỗi không có file nào được upload

Form Upload: Hàm move_uploaded_file()

- Khi quá trình upload file không có lỗi (thuộc tính error của biến $_FILES = 0), sử dụng hàm move_uploaded_file() để lưu đường dẫn thật cho file
- Cú pháp: move_uploaded_file(tmp_path, destination)
  - tmp_path: đường dẫn tạm thời của file trên server
  - destination: đường dẫn thật sẽ lưu trữ file, phải là đường dẫn vật lý

3. Form – Các vấn đề bảo mật

Bảo mật Form: Khái niệm

- Bảo mật là vấn đề cực kỳ quan trọng đối với các hệ thống website
- Do form là thành phần chính của việc truyền dữ liệu từ client đến server nên thường là đích đến của các cuộc tấn công
- Hiện nay các framework đều hỗ trợ rất tốt việc chống các tấn công này
- Hai hình thức tấn công phổ biến trong form là
    - XSS (Cross-site scripting)
    - CSRF (Cross-site request forgery)

Bảo mật Form: Tấn công XSS

- Kỹ thuật này được thực hiện bằng cách chèn các mã script độc hại vào trong code của bạn
- Cách phòng chống
  - Không bao giờ tin tưởng dữ liệu mà user nhập, do vậy cần validate dữ liệu trước khi lưu vào database
  - Mã hóa các ký tự đặc biệt mà user nhập trong form thành các thực thể HTML, sử dụng hàm htmlspecialchars()

Bảo mật Form: Tấn công CSRF

- Kỹ thuật này giả mạo chính chủ sở hữu của hệ thống đó
  - Ví dụ: Hệ thống của bạn có link xóa user dạng delete-user?id=xxx, hacker biết được link này và gửi 1 mail cho bạn với 1 image có src là link trên, với id cụ thể, khi click vào ảnh đó đồng nghĩa bạn đã xóa user!
- Cách phòng chống: tạo key (token) cho form, với các bước xử lý như sau:
  - Thêm thẻ input ẩn với giá trị key và thuộc tính là hidden
  - Lưu key này vào session
  - Mỗi lần submit form sẽ kiểm tra key này có hợp lệ hay không, chỉ xử lý khi key hợp lệ

4. Session và Cookie

Session

- Được hiểu như 1 phiên làm việc của client, session lưu trữ thông tin giữa client và hệ thống
- Biến được lưu trong session có thể được truy cập từ mọi nơi trên hệ thống
- Session sẽ mất khi đóng trình ứng dụng
- Biến toàn cục trong PHP lưu trữ các thông tin về session là $_SESSION, có kiểu mảng
- Một số ứng dụng hay sử dụng session là đăng nhập, giỏ hàng .v.v

Session

- Khởi tạo: session_start();
- Thêm dữ liệu: $_SESSION['name'] = value;
- Lấy giá trị: $_SESSION['name'];
  - Lưu ý trước khi lấy giá trị cần kiểm tra session đã tồn tại hay chưa sử dụng lệnh isset()
- Xóa session
  - Xóa 1 phần tử cụ thể: unset($_SESSION['name']);
  - Xóa toàn bộ session trên hệ thống: session_destroy();

Cookie

- Thường được dùng để lưu các giá riêng của từng trang web cụ thể cho client
- Cookie không mất đi khi đóng ứng dụng, sự tồn tại của cookie phụ thuộc vào thời gian sống khi bạn set cho nó
- Cookie được lưu dưới trình duyệt của client
- Biến toàn cục trong PHP lưu trữ các thông tin về cookie là $_COOKIE, có kiểu mảng

Cookie

- Khởi tạo: setcookie(name, value, expire, path, domain);

  name: tên cookie muốn tạo

  value: giá trị cookie

  expire: thời gian sống của cookie

  path: đường dẫn lưu cookie, mặc định là '/'

  domain: tên domain

- Lấy giá trị: $_COOKIE['name'];
  - Cần sử dụng lệnh isset() trước khi lấy giá trị
- Xóa cookie: Sử dụng lại phương thức setcookie như lúc khởi tạo, nhưng set thời gian sống nhỏ hơn thời gian hiện tại

Session vs Cookie

- Giống nhau:
  - Đều dùng để lưu trữ thông tin giữa client và hệ thống
  - Đều có thể được truy cập từ mọi nơi trên hệ thống
- Khác nhau:
  - Session có tính bảo mật hơn cookie
  - Cookie được lưu trên trình duyệt của client trong khi session được lưu trên server
  - Session sẽ bị mất khi đóng ứng dụng, trong khi cookie thì không

Bài tập: Session vs Cookie

- Demo chức năng login đơn giản, với mô tả như sau:
  - Màn hình login gồm 2 ô nhập username và password và 1 nút submit có tên Login
  - Trường hợp user nhập đúng username = nvanh, password = 123456 thì chuyển hướng sang màn hình khác - Gọi là màn hình login success, tại màn hình này sẽ hiển thị username của user và 1 nút Logout.
  - Tại màn hình Login success, khi click nút Logout thì sẽ chuyển hướng về trang login, tại trang login này sẽ hiển thị thông báo "Đăng xuất thành công"
  - Trong trường hợp login thành công rồi thì không thể truy cập vào trang login nữa, mà sẽ chuyển hướng sang màn hình Login success
  - Trong trường hợp chưa login, khi cố tình truy cập vào trang Login success (giả sử bạn đã biết trước url) thì báo "Cần đăng nhập để truy cập trang này" và chuyển hướng về trang login

Bài tập: Session vs Cookie

- Tạo form đăng nhập cho người dùng, gồm 2 trường username và password, và checkbox ghi nhớ đăng nhập
  - Nếu user/password là admin/123456 thì báo đăng nhập thành công, ngược lại là đăng nhập thất bại
  - Khi đăng nhập thành công, lưu session cho username, chuyển hướng tới 1 file khác , trong file này sẽ hiển thị tên username vừa đăng nhập và nút logout
  - Nếu user checkbox vào ô ghi nhớ đăng nhập, thì sẽ tự động đăng nhập vào lần sau
  - Khi click nut Logout, thì chuyển hướng người dùng về trang đăng nhập