



Lập trình ứng dụng web nâng cao

Công nghệ Thông tin (Trường Đại học Sài Gòn)



Scan to open on Studeersnel

1. Nguyên tắc hoạt động website

1.1 Khái niệm và thuật ngữ

Website là một tập hợp các trang web liên kết với nhau, được lưu trữ trên một máy chủ web và có thể truy cập thông qua mạng Internet. Nó là một không gian trực tuyến cho các tổ chức, doanh nghiệp, cá nhân hoặc tổ chức phi lợi nhuận để chia sẻ thông tin, sản phẩm, dịch vụ hoặc nội dung với người dùng trên khắp thế giới.

Mỗi trang web trong một website có thể chứa nội dung đa dạng như văn bản, hình ảnh, video, âm thanh, biểu đồ và các phương tiện truyền thông khác. Người dùng có thể truy cập vào website bằng cách sử dụng một trình duyệt web như Google Chrome, Mozilla Firefox,... và nhập địa chỉ của website (tên miền) vào thanh địa chỉ.

Một số khái niệm khác:

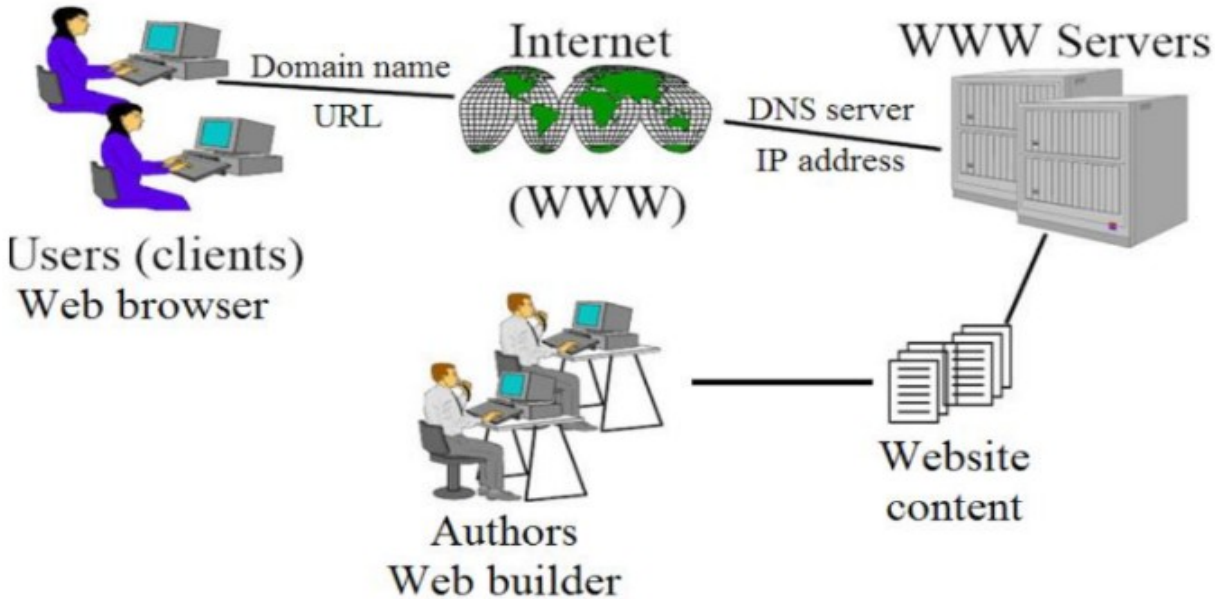
Domain name (Tên miền): Là địa chỉ duy nhất để truy cập vào website. Tên miền thường được mua từ các công ty đăng ký tên miền và được cung cấp trong thời hạn thuê. Ví dụ: google.com, facebook.com

DNS server (Domain Name System server): Là một hệ thống phân giải tên miền sang địa chỉ IP. Khi người dùng gõ một tên miền vào trình duyệt, trình duyệt sẽ gửi yêu cầu đến máy chủ DNS để lấy địa chỉ IP tương ứng với tên miền đó.

URL (Uniform Resource Locator): Là địa chỉ cụ thể của một tài nguyên trên Internet. Nó bao gồm ba phần chính: giao thức (ví dụ: "http://" hoặc "https://"), tên miền (ví dụ: "google.com") và đường dẫn tới tài nguyên cụ thể trên trang web (ví dụ: "/search"). URL định danh chính xác một tài nguyên cụ thể trên mạng.

Hosting: Là dịch vụ cung cấp không gian lưu trữ trên máy chủ web để lưu trữ các tệp tin và dữ liệu của một website. Website của bạn được lưu trữ trên máy chủ web đó, và khi người dùng truy cập vào tên miền của bạn, máy chủ web sẽ cung cấp nội dung của website cho người dùng.

1.2 Nguyên tắc hoạt động website dựa trên hình



Khi người dùng nhập URL của một trang web vào thanh địa chỉ của trình duyệt, trình duyệt sẽ kết nối với Internet và gửi yêu cầu đến máy chủ DNS. Máy chủ DNS sẽ phân giải tên miền trong URL thành địa chỉ IP tương ứng của máy chủ web. Dựa vào địa chỉ IP đó, trình duyệt có thể gửi yêu cầu (request) đến máy chủ. Máy chủ web nhận yêu cầu từ trình duyệt và trả về nội dung của website (Website content). Các người quản trị website (Authors Web Builder) có thể tùy chỉnh website theo ý muốn ở Website content.

1.3 Web hosting service

a) Khái niệm về dịch vụ lưu trữ web, và cho thuê domain

Dịch vụ lưu trữ web là một dịch vụ cung cấp không gian lưu trữ trên máy chủ (server) để lưu trữ các tệp tin, dữ liệu và các thành phần của trang web. Khi một trang web được tạo ra, các tệp tin HTML, CSS, hình ảnh, video và các thành phần khác sẽ được lưu trữ trên máy chủ thông qua dịch vụ lưu trữ web. Dịch vụ này đảm bảo rằng trang web có thể được truy cập và hoạt động trên Internet một cách liên tục.

Ta có thể truy cập trực tiếp website bằng địa chỉ ip của máy chủ website. Tuy nhiên, tên miền (domain) sẽ giúp cho website dễ nhớ hơn.

Dịch vụ cho thuê domain là một dịch vụ cung cấp khả năng đăng ký và sở hữu tên miền cho người dùng. Khi bạn sử dụng dịch vụ này,

bạn có thể tìm và chọn một tên miền duy nhất và đăng ký nó để sử dụng cho trang web, email hoặc các dịch vụ trực tuyến khác.

Sau khi bạn đã đăng ký tên miền, nhà cung cấp dịch vụ sẽ cung cấp một máy chủ DNS (DNS server) mặc định hoặc cho phép bạn chỉ định máy chủ DNS riêng của mình. Công việc của DNS server là quản lý bản ghi DNS (DNS records) liên kết tên miền với các địa chỉ IP tương ứng. Khi một yêu cầu truy cập đến tên miền của bạn, máy chủ DNS sẽ trả về địa chỉ IP của máy chủ web mà tên miền đó liên kết. Điều này cho phép trình duyệt của người dùng biết địa chỉ IP để kết nối và truy cập trang web.

Một số bản ghi DNS phổ biến bao gồm (*thông tin thêm*)

- A record (bản ghi A): Liên kết một tên miền với địa chỉ IPv4 của máy chủ web.
- AAAA record (bản ghi AAAA): Liên kết một tên miền với địa chỉ IPv6 của máy chủ web.
- CNAME record (bản ghi CNAME): Liên kết một tên miền với một tên miền khác, thường được sử dụng cho việc tạo bí danh (alias) cho tên miền.
- MX record (bản ghi MX): Xác định máy chủ email (mail server) được sử dụng cho tên miền để nhận và gửi email.
- TXT record (bản ghi TXT): Chứa thông tin bổ sung về tên miền, thường được sử dụng cho mục đích xác thực và chứng thực.

b) Giải thích các thông số: dung lượng lưu trữ (disk space), băng thông (bandwidth), tên miền con (sub-domain), email (có domain riêng), ngôn ngữ lập trình web và cơ sở dữ liệu được web server hỗ trợ.

Dung lượng lưu trữ (disk space): Đây là khoảng không gian được cung cấp bởi dịch vụ lưu trữ web để lưu trữ các tệp tin, hình ảnh, dữ liệu và các thành phần khác của trang web. Dung lượng lưu trữ quyết định khối lượng dữ liệu mà bạn có thể lưu trữ trên máy chủ web.

Băng thông (bandwidth) là thông số chỉ mức dung lượng truyền tải dữ liệu giữa website với người truy cập web trong một

thời gian nhất định. Nếu trang web của bạn có lượng truy cập lớn hoặc chứa nhiều tệp tin lớn, bạn cần có băng thông cao để đảm bảo truy cập nhanh và không bị giới hạn.

Tên miền con (sub-domain): Đây là một phần của tên miền chính và được sử dụng để tạo ra các địa chỉ web phụ trong một tên miền. Ví dụ, nếu tên miền chính là "example.com", bạn có thể tạo ra các tên miền con như "blog.example.com" hoặc "shop.example.com".

Email (có domain riêng): Khi bạn có một tên miền riêng, bạn có thể tạo ra các địa chỉ email dựa trên tên miền đó, ví dụ: "contact@example.com" hoặc "info@example.com". Điều này cho phép bạn tạo ra các địa chỉ email chuyên nghiệp và liên quan đến tên miền của bạn. Bạn có thể gửi và nhận email thông qua các địa chỉ email này

Ngôn ngữ lập trình web: Đây là các ngôn ngữ được sử dụng để phát triển trang web và xây dựng các chức năng backend (PHP, Python, Nodejs, Ruby, C#, Java, ...) và giao diện frontend của trang web (HTML, CSS, Javascript, ...) . Mỗi ngôn ngữ có các đặc điểm và ứng dụng khác nhau, và bạn có thể sử dụng một hoặc nhiều ngôn ngữ để xây dựng trang web theo yêu cầu của mình.

Cơ sở dữ liệu là nơi lưu trữ và quản lý dữ liệu của trang web. Các máy chủ web thường hỗ trợ một số loại cơ sở dữ liệu phổ biến như MySQL, PostgreSQL hoặc MongoDB.

2. Giao diện website

2.1 Các khái niệm về website

a) *Phân loại website:* gồm web tĩnh và web động

Website tĩnh (Static website): Đây là loại website mà nội dung của trang web được tạo ra và lưu trữ sẵn trước, không thay đổi theo tác động của người dùng hoặc dữ liệu đầu vào. Mỗi trang web tĩnh được xem là một tệp HTML độc lập và được gửi

đến người dùng một cách không thay đổi. Các trang web tĩnh thường hiển thị các thông tin cố định như thông tin công ty, sản phẩm, dịch vụ, v.v. Các trang web tĩnh dễ triển khai, đơn giản và tốn ít tài nguyên máy chủ.

Website động (Dynamic website): Đây là loại website mà nội dung của trang web được tạo ra dựa trên yêu cầu của người dùng hoặc dữ liệu đầu vào. Thông thường, website động được xây dựng bằng cách sử dụng các ngôn ngữ lập trình và cơ sở dữ liệu để tạo ra và cập nhật nội dung động trên trang web. Website động thường có tính tương tác cao và có thể cung cấp các chức năng như đăng ký, đăng nhập, giỏ hàng, tìm kiếm, v.v. Với website động, nội dung và giao diện của trang web có thể thay đổi dựa trên tác động của người dùng hoặc các sự kiện xảy ra.

b) Các thành phần của website:

Một trang web thông thường bao gồm các thành phần chính sau:

- **Header:** Vùng đầu trang của trang web chứa logo, menu điều hướng và các thành phần khác.
- **Nội dung:** Khu vực chứa thông tin, hình ảnh, video và các thành phần khác hiển thị trên trang web.
- **Sidebar:** Vùng bên cạnh nội dung chính thường chứa các liên kết, danh sách, widget và thông tin phụ.
- **Footer:** Vùng cuối trang web chứa thông tin liên hệ, các liên kết quan trọng và các thành phần khác.

c) Sitemap là một tệp hoặc một trang web đặc biệt chứa danh sách toàn bộ các URL của trang web. Nó giúp công cụ tìm kiếm và người dùng dễ dàng hiểu cấu trúc và nội dung của trang web. Sitemap thường được sử dụng để cải thiện khả năng tìm kiếm của trang web và tăng cơ hội được xếp hạng cao hơn trong kết quả tìm kiếm.

d) *Trang chủ* (home page) là trang đầu tiên mà người dùng thấy khi truy cập vào một trang web. Trang chủ thường chứa thông tin quan trọng nhất và cung cấp điểm khởi đầu để điều hướng đến các phần khác của trang web.

e) *Hệ thống liên kết* trên một trang web cho phép người dùng di chuyển giữa các trang và nội dung khác nhau. Các liên kết có thể được tạo thành các liên kết văn bản, nút, hình ảnh hoặc biểu tượng. Hệ thống liên kết cung cấp tính năng điều hướng và tương tác giữa các trang trong trang web.

f) *Giao diện cho người dùng cuối*: Là phần của trang web mà người dùng cuối thấy và tương tác trực tiếp. Nó bao gồm giao diện người dùng, màu sắc, bố cục, hình ảnh và các phần tử tương tác như nút, mẫu nhập liệu. Mục tiêu của giao diện này là cung cấp trải nghiệm tốt nhất cho người dùng và tạo ra một trang web hấp dẫn và dễ sử dụng.

Giao diện cho người quản trị web: Là phần của trang web dành riêng cho người quản trị để quản lý và điều khiển trang web. Nó bao gồm các chức năng quản lý nội dung, người dùng, bảo mật và các công cụ quản trị khác. Giao diện này được thiết kế để người quản trị dễ dàng sử dụng và thao tác trên trang web một cách hiệu quả, như thêm, chỉnh sửa và xóa nội dung, quản lý người dùng và kiểm soát các cài đặt quan trọng của trang web.

2.2 Giao diện website

a) *Quan điểm khi thiết kế giao diện website*

Khi thiết kế giao diện website, quan điểm quan trọng là tập trung vào người dùng, đơn giản và trực quan, sử dụng hình ảnh và màu sắc hợp thời trang, tuân thủ nguyên tắc thiết kế, tạo ra trải nghiệm tiện ích và dễ sử dụng, tối ưu hóa tốc độ tải trang và liên tục cải tiến.

b) *Nguyên tắc C.R.A.P*

- Contrast (Tương phản): Tổ chức thiết kế của bạn, thiết lập hệ thống phân cấp, nhấn mạnh tiêu điểm và thêm sự quan tâm trực quan

- Repetition (Lặp lại): Tạo sự nhất quán, tăng khả năng học hỏi của người dùng, giảm nhầm lẫn
- Alignment (Căn chỉnh): Tổ chức và nhóm các thành phần, tạo nhịp điệu, từ trật tự đến hỗn loạn
- Proximity (Tiệm cận): Tạo sự kết nối giữa các thành phần, mỗi thành phần đều dễ hiểu và có ý nghĩa

c) *Tính tiện dụng trong thiết kế giao diện*

Tính tiện dụng trong thiết kế giao diện đảm bảo giao diện dễ sử dụng, tối ưu hóa trải nghiệm người dùng, tăng cường khả năng tìm kiếm, cung cấp hướng dẫn và hỗ trợ, tự động hoá và tiết kiệm thời gian, và đáp ứng trên các thiết bị khác nhau.

- Một số nguyên tắc thiết kế tiện dụng:
 - + Đảm bảo tính thống nhất trong thiết kế
 - + Sắp xếp bố cục hợp lý
 - + Tránh lạm dụng các hiệu ứng hình ảnh
 - + Tránh sử dụng quá nhiều chi tiết trên một thiết kế
 - + Cung cấp các phản hồi hành động
 - + Màu sắc giao diện đồng bộ, thể hiện giá trị thương hiệu
 - + Thiết kế giao diện tương thích thiết bị di động
- Một số sai lầm thường gặp khi thiết kế giao diện website:
 - + Không có thanh công cụ tìm kiếm
 - + Chữ khó đọc
 - + Giao diện không thân thiện với điện thoại di động
 - + Nội dung bài viết kém chất lượng
 - + Chức năng đăng kí, tạo tài khoản phức tạp
 - + Sử dụng quá nhiều chữ trong website

2.3 Phân biệt lập trình winform và lập trình webform. Cho ví dụ minh họa.

- Lập trình WinForms là việc xây dựng ứng dụng desktop trên hệ điều hành Windows.

Ví dụ: Xây dựng một ứng dụng quản lý cửa hàng trên máy tính cá nhân. Giao diện người dùng được tạo bằng các thành phần Windows Forms như các nút, hộp văn bản, danh sách, và các

bảng điều khiển khác. Ứng dụng này chạy trên máy tính cục bộ và không cần kết nối Internet.

- Lập trình WebForms là việc xây dựng ứng dụng web trên nền tảng web.

Ví dụ: Xây dựng một trang web thương mại điện tử để bán hàng trực tuyến. Giao diện người dùng được tạo bằng các thành phần WebForms như các nút, hộp văn bản, danh sách, và các bảng điều khiển khác. Trang web này chạy trên máy chủ web và được truy cập qua trình duyệt web thông qua kết nối Internet.

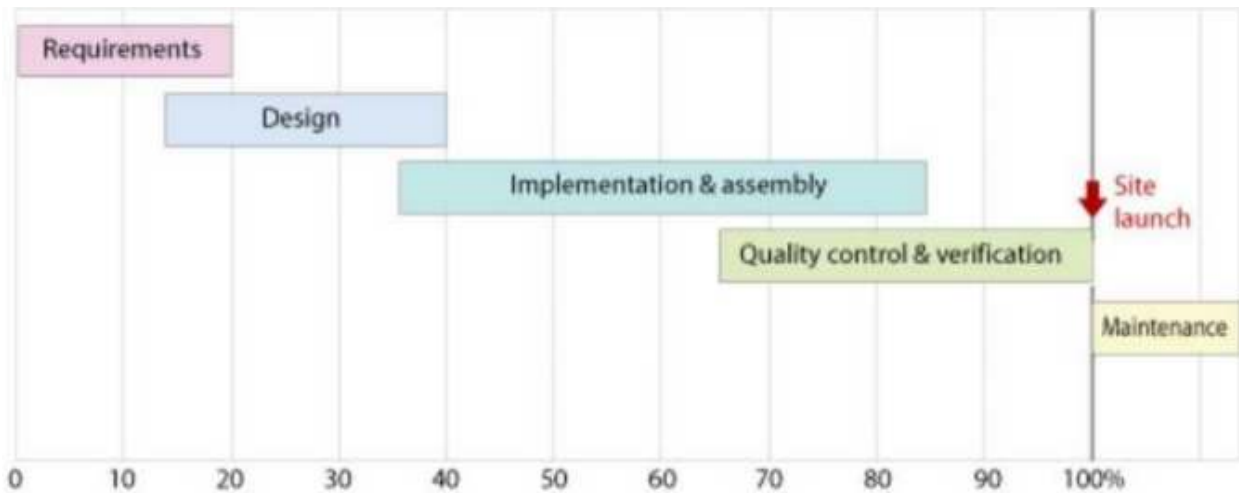
4. Quá trình xây dựng website

4.1 Tổng quan quá trình xây dựng website

1. Thu thập yêu cầu và mục tiêu của website.
2. Lập kế hoạch và xác định cấu trúc.
3. Thiết kế giao diện và trình bày nội dung.
4. Phát triển và triển khai website.
5. Kiểm thử và sửa lỗi.
6. Triển khai và vận hành website.
7. Tiếp tục cải tiến và duy trì website.

4.2 Chi tiết mỗi giai đoạn trong quá trình này

- **Requirements**: giai đoạn xác định các yêu cầu chức năng và phi chức năng mà hệ thống phần mềm cần có. KQ là tài liệu đặc tả yêu cầu.
- **Design**: là giai đoạn thực hiện phân tích thiết kế để hệ thống phần mềm đáp ứng đúng yêu cầu của khách hàng
- **Implement & assembly**: giai đoạn thực hiện sản phẩm dựa trên đặc tả yêu cầu và tài liệu thiết kế module
- **Quality control & verification**: giai đoạn kiểm thử các thành phần hệ thống
- **Maintenance**: giai đoạn bảo trì



4.3 Vai trò của mỗi nhóm người (người thiết kế, người lập trình, khách hàng) trong quá trình xây dựng website. **Nhấn mạnh, phân biệt công việc thiết kế và lập trình web**

Trong quá trình xây dựng website, mỗi nhóm người có vai trò và công việc riêng, bao gồm:

1. **Người thiết kế:**

Vai trò: Tạo ra giao diện và trải nghiệm người dùng hấp dẫn.

Công việc: Thiết kế các mockup, layout, hình ảnh và các yếu tố giao diện khác. Xác định cấu trúc và luồng điều hướng của trang web.

2. **Người lập trình:**

Vai trò: Triển khai các yêu cầu chức năng và xử lý dữ liệu.

Công việc: Sử dụng các ngôn ngữ lập trình và công nghệ web để xây dựng website. Tạo ra các chức năng tương tác, tích hợp cơ sở dữ liệu và xử lý các yêu cầu của người dùng.

3. **Khách hàng:**

Vai trò: Đưa ra yêu cầu, mục tiêu và phản hồi về website.

Công việc: Cung cấp thông tin về mục tiêu, nội dung và yêu cầu của website. Cung cấp phản hồi và hỗ trợ trong quá trình xây dựng để đảm bảo website đáp ứng được mục tiêu và mong đợi.

Người thiết kế tập trung vào khía cạnh giao diện và trải nghiệm người dùng, trong khi người lập trình tập trung vào triển

khai chức năng và xử lý dữ liệu. Khách hàng đóng vai trò quan trọng trong việc định hình yêu cầu và đảm bảo website phù hợp với mục tiêu và mong đợi của họ.

5. Khái niệm client-script và sử dụng NNLT Javascript

5.1 Kiến thức về NNLT thông dịch client-script Javascript

Javascript là một ngôn ngữ lập trình phổ biến trong phát triển ứng dụng web. Nó được thực thi trên trình duyệt của người dùng và tương tác với HTML và CSS để tạo ra các chức năng động trên trang web. Javascript sử dụng cú pháp giống với nhiều ngôn ngữ lập trình khác và hỗ trợ các kiểu dữ liệu và tính năng phong phú như biến, hàm, điều kiện, vòng lặp và xử lý sự kiện.

5.2 Kiến thức về sử dụng Javascript. Vd: cú pháp, kiểu dữ liệu, biến, hàm...

Cú pháp: `alert('Nội dung')`, `console.log()`, `prompt()`

Khai báo biến: `var`, `let`, `const`

Hàm: `function hello(){}`

5.3 Khái niệm cơ bản về XSS. Từ khóa: “cross-site scripting”

XSS (Cross-Site Scripting) là một lỗ hổng bảo mật phổ biến trong ứng dụng web, cho phép tin tặc chèn mã độc vào trang web để tấn công người dùng. Mã độc XSS có thể được chèn qua các tham số truy vấn, form data hoặc lưu trữ trên máy chủ. Khi người dùng truy cập vào trang bị ảnh hưởng, mã độc XSS sẽ được thực thi trên trình duyệt của họ, gây ra các hậu quả đáng ngại như đánh cắp thông tin cá nhân hoặc kiểm soát trình duyệt của người dùng. Để ngăn chặn XSS, cần xử lý đúng dữ liệu đầu vào, áp dụng phương pháp bảo vệ và kiểm tra rủi ro.

6. Khái niệm & sử dụng mô hình Javascript + HTML DOM, xử lý sự kiện

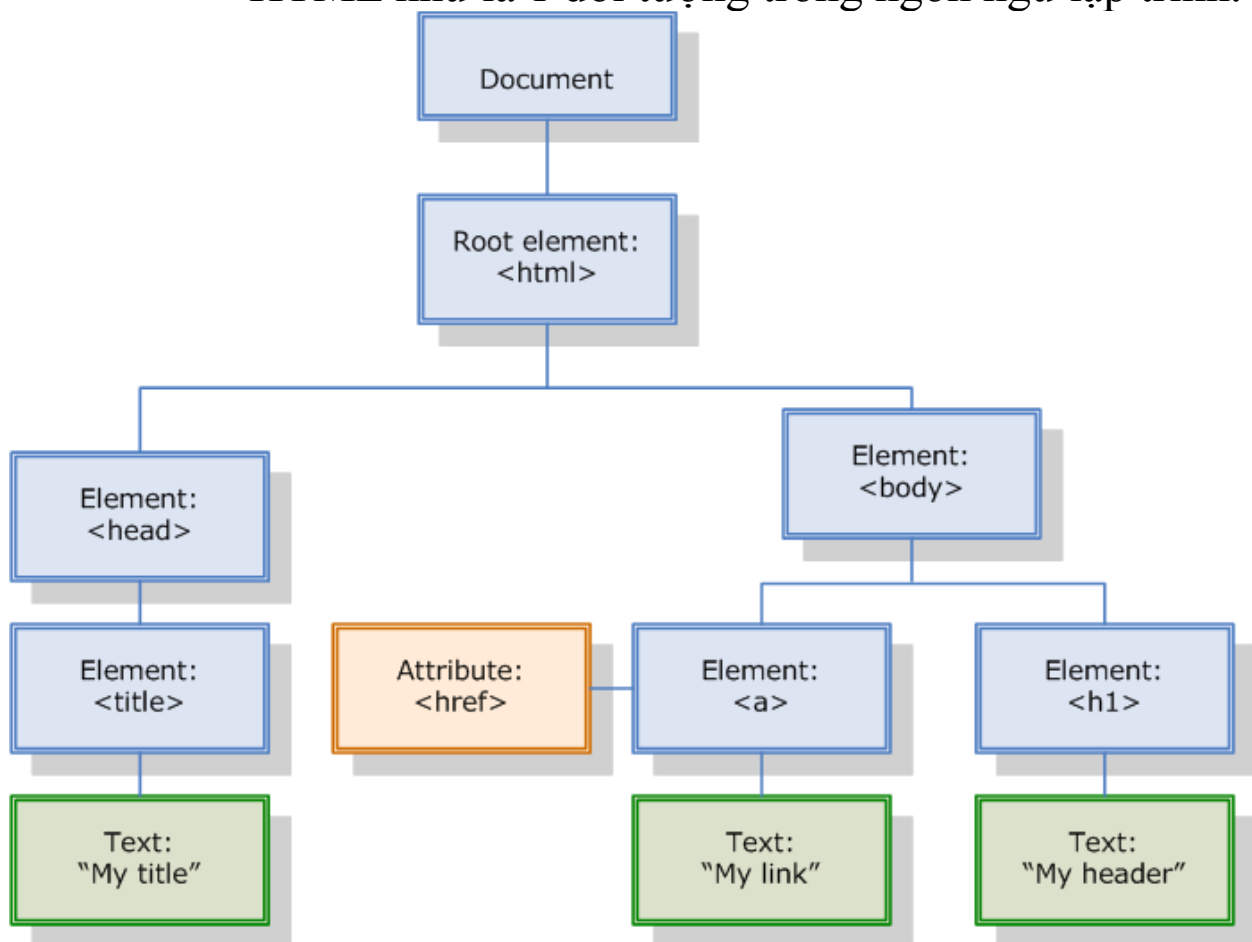
6.1 Kiến thức và xử lý sự kiện

DOM = Document Object Model (HTML DOM)

- Mô hình mô tả 1 tài liệu có cấu trúc như là 1 cây có nhiều node.
- Mỗi nút là 1 đối tượng có thể xử lý trong ngôn ngữ lập trình
- Ngôn ngữ lập trình hỗ trợ DOM: có thể đọc và phân tích tài liệu có cấu trúc thành các đối tượng lập trình.

File HTML là 1 file có cấu trúc thể hiện qua các thẻ

- Mỗi thẻ là 1 thành phần (element) trong mô hình DOM và là 1 nút trong cây DOM
- Javascript hỗ trợ DOM: có thể xử lý các thẻ HTML như là 1 đối tượng trong ngôn ngữ lập trình.



Ta có thể dùng `document.getElementById` hoặc `document.querySelector` để lấy element

6.2 Ứng dụng xử lý một số sự kiện thông dụng

Tên	Giải thích
onClick	Xảy ra khi nhấp chuột
onMouseDown/ onMouseUp	Xảy ra khi nhấn chuột xuống / thả ra
onMouseOver/ onMouseOut	Xảy ra khi di chuyển chuột vào / ra
onKeyDown / onKeyUp	Xảy ra khi 1 phím được ấn xuống / thả ra
onKeyPress	Xảy ra khi 1 phím được nhấn

Thêm hiệu ứng click vào thẻ: `<p onclick="doSomething()"></p>`

Hoặc dùng: `element.addEventListener("click", function() {
doSomething();
});`

6.3 Sử dụng javascript có ứng dụng regex để kiểm tra định dạng khi nhập liệu.

```
const phoneNumberRegex = /^\\d{3}-\\d{3}-\\d{4}$/;  
const phoneNumber1 = "123-456-7890";  
const phoneNumber2 = "abc-def-ghij";  
console.log(phoneNumberRegex.test(phoneNumber1)); // true  
console.log(phoneNumberRegex.test(phoneNumber2)); // false
```

7. Khái niệm server-script và sử dụng NNLT PHP

7.1 Kiến thức về NNLT thông dịch server-script PHP

- Là ngôn ngữ server-side-script thực thi phía WebServer trả kết quả về cho client.
- Cho phép người dùng tương tác với web site

- Thiên về xử lý nghiệp vụ.
- Tập tin PHP có phần mở rộng là **.php**
- Cú pháp ngôn ngữ giống ngôn ngữ C++
 - Ưu điểm:
 - Đa môi trường: web-server, hệ điều hành, hệ quản trị cơ sở dữ liệu,...
 - Tốc độ truy xuất nhanh, Free, Open Source
 - Hướng đối tượng.
 - Khuyết điểm:
 - Cú pháp rắc rối, chức năng quản lý lỗi.

7.2 Phân biệt cơ chế hoạt động client-script và server-script

Client-script (script phía client): Mã lập trình thực thi trên trình duyệt người dùng, cho phép tương tác trực tiếp với giao diện người dùng và thực hiện các tác vụ động trên trang web.

Server-script (script phía máy chủ): Mã lập trình thực thi trên máy chủ web trước khi gửi trang web đến trình duyệt người dùng, được sử dụng để xử lý dữ liệu, tương tác với cơ sở dữ liệu và thực hiện các tác vụ phức tạp.

7.3 Kiến thức về sử dụng PHP. Vd: cú pháp, kiểu dữ liệu, biến, hàm...

Mã php được đặt trong thẻ `<?php` và thẻ đóng `>`

Khai báo biến: `$val = 0; $arr = [];`

Khai báo hàm: `function helloworld($param){}`

Vòng lặp foreach: `foreach($item as $i){}, foreach($item as $key => $value){}`

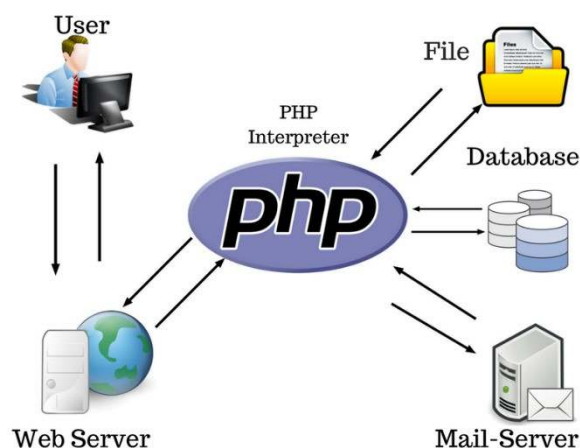
Vòng lặp for: `for ($i = 0; $i < 10; $i++) {};`

7.4 Mục đích và cú pháp PHP

Mục đích chính của PHP là tạo ra các ứng dụng web động và tương tác với cơ sở dữ liệu trên máy chủ. PHP cho phép bạn thực hiện các tác vụ phức tạp như xử lý biểu mẫu, xác thực người dùng, tạo và quản lý phiên làm việc, gửi và nhận cookie,

tạo trang web động, và thực hiện nhiều chức năng khác phía máy chủ.

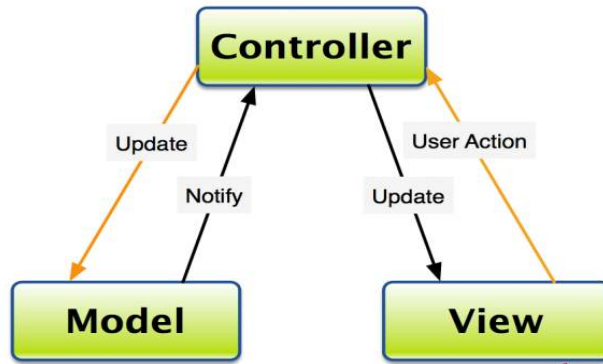
Cách hoạt động của PHP: Khi người sử dụng gọi trang PHP, Web Server sẽ triệu gọi PHP Engine để thông dịch dịch trang PHP và trả kết quả cho người dùng như hình bên dưới.



Hình: Cách hoạt động của PHP

Khái niệm về mô hình MVC

MVC là chữ viết tắt của Model - View - Controller, đây là một mô hình kiến phần mềm được tạo ra với mục đích quản lý và xây dựng dự án phần mềm có hệ thống hơn. Mô hình này được dùng khá rộng rãi và đặc biệt là trong các ngôn ngữ lập trình web. Trong PHP hiện tại có khá nhiều Framework và tất cả đều xây dựng từ mô hình MVC.



Hình: Mô hình MVC

Model chính là thành phần chứa những nghiệp vụ tương tác với dữ liệu hay hệ quản trị cơ sở dữ liệu. Phần này bao gồm các class hay function có khả năng xử lý nhiều nghiệp vụ như: kết nối database, thêm/ xóa/ sửa dữ liệu, truy vấn dữ liệu...

View chính là phần chứa những giao diện tương tự như một nút bấm, khung nhập, menu, hình ảnh. View đảm nhiệm chức trách hiển thị dữ liệu, giúp người dùng tương tác với hệ thống.

Controller là phần tiếp nhận mọi yêu cầu xử lý của người dùng. Nó bao gồm những class/ function có khả năng xử lý nhiều nghiệp vụ logic. Giúp lấy dữ liệu đúng những thông tin cần thiết nhờ vào các nghiệp vụ lớp Model cung cấp và hiển thị các dữ liệu đó ra cho người dùng.

Ưu điểm và nhược điểm mô hình MVC

Ưu điểm:

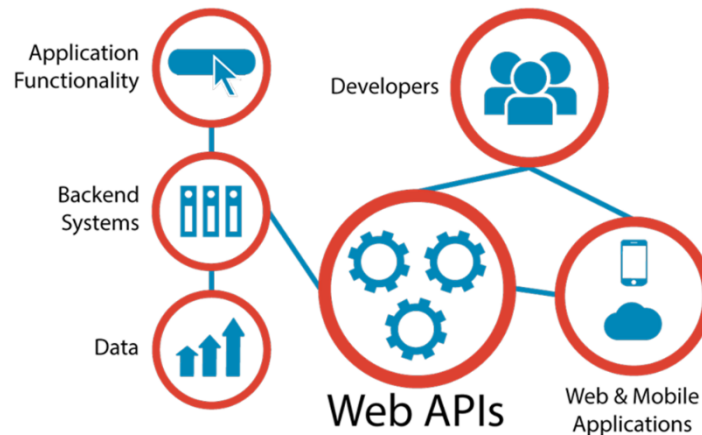
- Bảo trì mã dễ dàng, giúp mở rộng và phát triển.
- Sự phát triển của các thành phần khác nhau có thể được thực hiện song song.
- Nó giúp tránh sự phức tạp bằng cách chia một ứng dụng thành ba phần: Model, View và Controller.
- Cung cấp hỗ trợ tốt nhất để phát triển dựa trên thử nghiệm.
- Nó hoạt động tốt cho các ứng dụng Web được hỗ trợ bởi các nhóm các nhà thiết kế và phát triển web.
- Công cụ tìm kiếm tối ưu hóa (SEO) thân thiện.
- Tất cả các layer (lớp) và các đối tượng là độc lập với nhau để có thể kiểm tra chúng một cách riêng biệt.

Nhược điểm:

- Hệ thống sẽ chạy chậm hơn PHP thuần
- Xây dựng cầu kì và mất thời gian để xây dựng thư viện, cấu trúc
- Việc điều hướng tổ chức có thể đôi khi phức tạp vì nó giới thiệu các layer mới đòi hỏi người dùng phải thích ứng với các tiêu chí của MVC.
- Không hỗ trợ xác nhận chính thức.
- Tăng độ phức tạp và kém hiệu quả của dữ liệu.
- Khó khăn khi sử dụng MVC với giao diện người dùng.
- Cần có nhiều lập trình viên để tiến hành lập trình song song.
- Phải có lượng kiến thức về công nghệ phải nhiều vì nó rất cần thiết.
- Duy trì rất nhiều mã trong controller (bộ điều khiển).

Web API

- **API** là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của **Application Programming Interface** – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.



Hình: Cách hoạt động của Web APIs

Web API là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng [JSON](#) hoặc XML thông qua giao thức HTTP hoặc HTTPS.

Cách hoạt động của Web API

- Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.

- Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.
- Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.
- Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...

Cú pháp JSON

Dữ liệu JSON (JSON data)

- ✓ Cú pháp JSON bắt nguồn từ cú pháp kí hiệu đối tượng trong Javascript.
- ✓ Dữ liệu sẽ có dạng: “key”:”value” và các dấu ngoặc kép là bắt buộc.
- ✓ Dữ liệu được ngăn cách bằng dấu ,.
- ✓ Dấu {} bao ngoài dữ liệu.
- ✓ Dấu [] để ám chỉ mảng.

Dữ liệu JSON là một cặp tên trường/ thuộc tính (field name/ key) và giá trị (value) cách nhau dấu:

Key: value

Các key của JSON bắt buộc phải đặt trong dấu nháy kép “ “, nếu ta đặt nó trong dấu nháy đơn thì đây không phải là một chuỗi JSON đúng chuẩn. Key của JSON nên đặt chữ cái không dấu hoặc số, dấu _ và không có khoảng trắng., ký tự đầu tiên không nên đặt là số.

Giá trị có thể là số, chuỗi bao nháy kép, logic, đối tượng bao ngoặc móc, và mảng bao ngoặc vuông, null. Trường hợp trong value có chứa dấu nháy kép thì hãy dùng dấu (\) để đặt trước nó, ví dụ học \”JSON là gì\”

```
“firstName”:”John”
```

Đối tượng JSON (JSON object)

Đối tượng JSON gồm các cặp tên trường/ thuộc tính: giá trị cách nhau dấu phẩy, bao bởi ngoặc móc

```
{“firstName”:”John”, “lastName”:”Doe”}
```

Mảng các đối tượng (JSON array)

Mảng các đối tượng JSON bao gồm các đối tượng JSON ngăn cách bởi dấu phẩy, bao bởi dấu ngoặc vuông

```
{“employees”: [  
    {“firstName”:”John”, “lastName”:”Doe”},  
  
    {“firstName”:”Anna”, “lastName”:”Smith”}  
] }
```

Truy cập phần tử mảng:

```
Employees [0] [“firstName”]  
Employees [0].firstName + “ ” +  
employees[0].lastName  
  
employees[0][“firstName”]=”Gilbert”;  
employees[0].firstName=”Gilbert”;
```

Giống và khác nhau giữa JSON và XML

Giống nhau:

- ✓ Định dạng mô tả có thể đọc
- ✓ Có cấu trúc phân cấp
- ✓ Có thể duyệt và sử dụng bởi bất kỳ ngôn ngữ lập trình
- ✓ Có thể duyệt với một đối tượng XMLHttpRequest
- ✓ Được sử dụng để nhận dữ liệu từ web server

Khác nhau:

- ✓ JSON không sử dụng end tag
- ✓ JSON ngắn gọn hơn, dễ hiểu, đọc ghi nhanh hơn XML
- ✓ JSON có thể sử dụng mảng
- ✓ XML phải duyệt với một XML parser, JSON có thể duyệt bởi một hàm JavaScript chuẩn

7.5 PHP cơ bản và xử lý nhập xuất dữ liệu trên web

a) Nhập dữ liệu từ form HTML :



```

xuly.php

<?php
if ($_SERVER["REQUEST_METHOD"] != "POST") {
    $name = $_POST["name"];
    echo "Xin chào, $name!";
}
?>

```

```

<DOCTYPE html>
<html>
<head>
  <title>
    Nhập xuất form
  </title>
</head>
<body>
  <form action="xuly.php" method="post">
    <label for="name">Nhập tên</label>
    <input type="text" name="name" id="name">
    <input type="submit" value="Gửi">
  </form>
</body>
</html>

```

b) Xuất dữ liệu từ PHP



```

index.php

<?php
    $name = "Name"
    echo "<h1>Hello, $name</h1>"
?>

```

c) Lấy tham số từ URL



7.6 PHP session và cookies

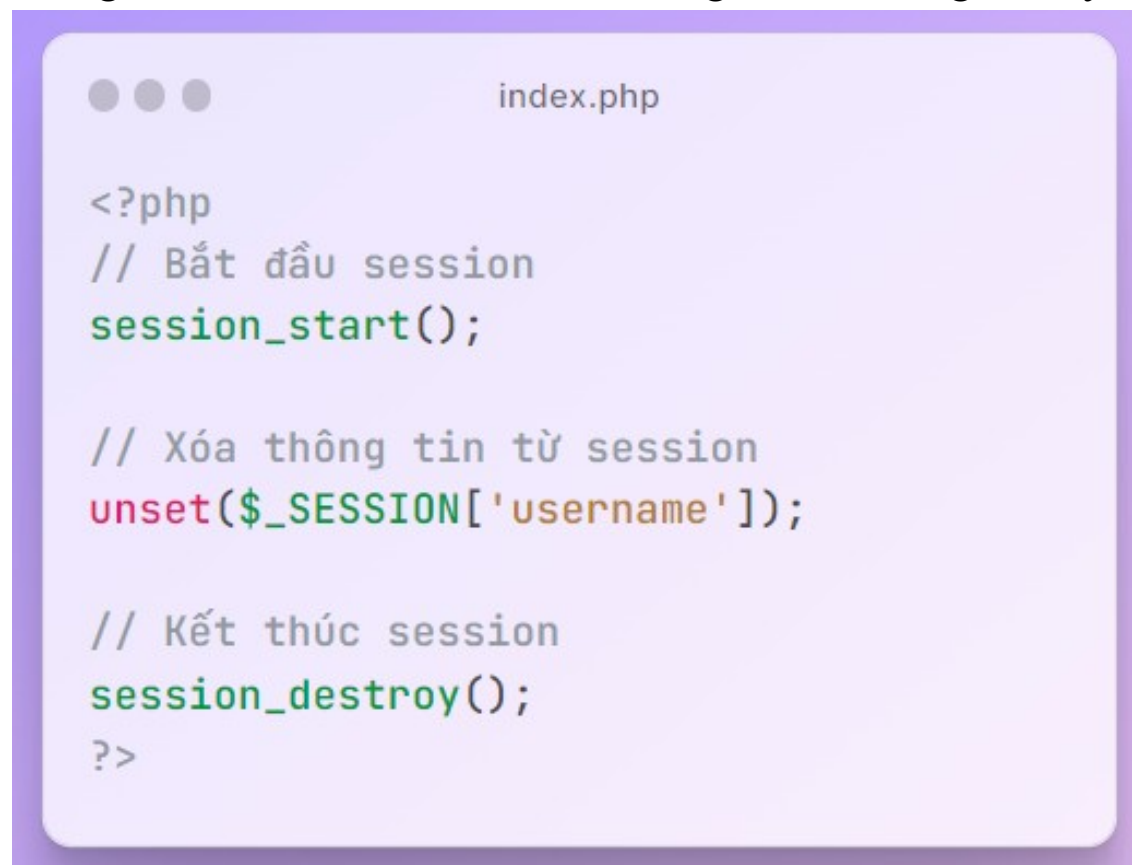
a) Session

- Session là một cơ chế lưu trữ thông tin trạng thái của người dùng trên máy chủ.
- Khi một người dùng truy cập vào một trang web, máy chủ tạo một session duy nhất và gửi một session ID đến trình duyệt của người dùng.
- Session ID này được lưu trữ trong cookie hoặc truyền qua URL để duy trì phiên làm việc.
- Thông tin trạng thái được lưu trữ trên máy chủ và có thể truy cập và sửa đổi trong suốt phiên làm việc của người dùng trên trang web.
- Session thường được sử dụng để lưu trữ thông tin như thông tin đăng nhập, giỏ hàng, lịch sử duyệt web, v.v.

Hàm `session_start()` dùng để bắt đầu 1 session

Hàm `session_unset()` dùng để xóa các biến của session và session vẫn tồn tại

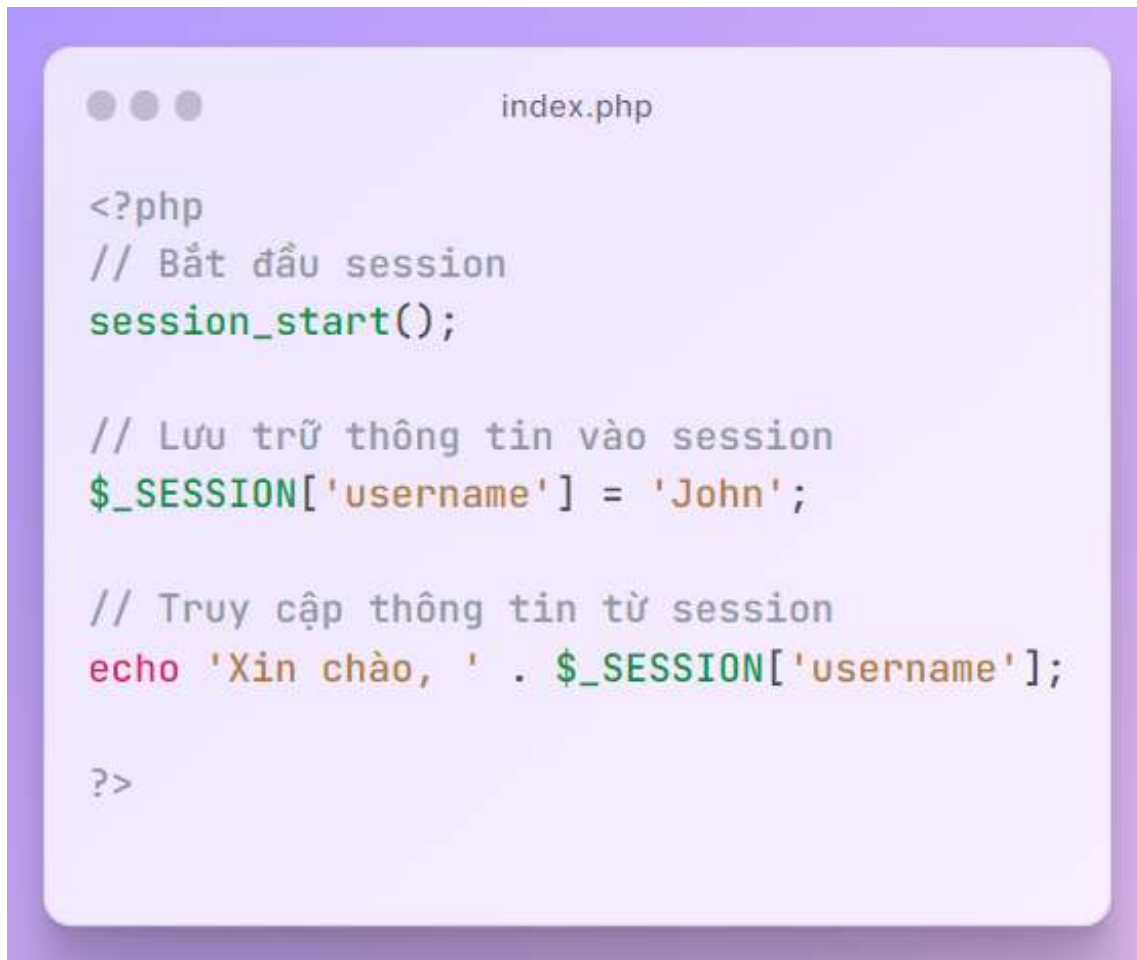
Hàm `session_destroy()` dùng để phá hủy session hiện tại, mọi thông tin như `PHPSESSIONID` trong cookie cũng bị hủy.

A code editor window with a light purple background and a darker purple border. The title bar at the top shows three window control buttons (red, yellow, green) and the filename 'index.php'. The code is written in a monospaced font with syntax highlighting: PHP tags are grey, comments are grey, function names are green, and variable names are red. The code demonstrates starting a session, unsetting a specific session variable, and then destroying the session.

```
<?php
// Bắt đầu session
session_start();

// Xóa thông tin từ session
unset($_SESSION['username']);

// Kết thúc session
session_destroy();
?>
```

```
index.php

<?php
// Bắt đầu session
session_start();

// Lưu trữ thông tin vào session
$_SESSION['username'] = 'John';

// Truy cập thông tin từ session
echo 'Xin chào, ' . $_SESSION['username'];

?>
```

b) Cookies

- Cookie là một tệp nhỏ, có thời hạn được lưu trữ trên máy tính của người dùng.
- Khi máy chủ gửi trang web cho trình duyệt, nó có thể gửi và lưu trữ một hoặc nhiều cookie trên máy tính của người dùng.
- Mỗi lần người dùng truy cập vào trang web, trình duyệt sẽ gửi cookie tương ứng với trang web đó đến máy chủ.
- Cookie thường được sử dụng để lưu trữ thông tin như thông tin đăng nhập, cài đặt ngôn ngữ, tùy chọn cá nhân, v.v.

```

index.php

<?php
// Tạo cookie
setcookie('username', 'John', time() + 3600); // Hết hạn sau 1 giờ

// Truy cập thông tin từ cookie
echo 'Xin chào, ' . $_COOKIE['username'];

// Xóa cookie
setcookie('username', '', time() - 3600);

// Lấy giá trị mặc định từ cookie
$username = isset($_COOKIE['username']) ? $_COOKIE['username'] : 'Guest';
?>

```

7.7 Lập trình hướng đối tượng trong PHP

a) Khởi tạo đối tượng

```

<?php
class Fruit {
    public static $info = "Data";
    public $name;
    public $color;

    //constructor
    function __construct($name) {
        $this->name = $name;
    }

    function get_name() {
        return $this->name;
    }
}

$apple = new Fruit("Apple");
echo $apple->get_name();
echo Fruit::$info;
?>

```

b) Inheritance (kế thừa)

```
index.php

<?php
class Fruit {
    public $name;
    public $color;
    public function __construct($name, $color) {
        $this->name = $name;
        $this->color = $color;
    }
    public function intro() {
        echo "The fruit is {$this->name} and the color is {$this->color}.";
    }
}
```

```
index.php

class Strawberry extends Fruit {
    public function message() {
        echo "Am I a fruit or a berry? ";
    }
}

$strawberry = new Strawberry("Strawberry", "red");
$strawberry->message();
$strawberry->intro();
?>
```

c) Abstract (trình tượng)

```

index.php

<?php
// Parent class
abstract class Car {
    public $name;
    public function __construct($name) {
        $this->name = $name;
    }
    abstract public function intro() : string;
}

// Child classes
class Audi extends Car {
    public function intro() : string {
        return "Choose German quality! I'm an $this->name!";
    }
}

$audi = new audi("Audi");
echo $audi->intro();
?>

```

d) Interface

```

index.php

<?php
interface Animal {
    public function makeSound();
}

class Cat implements Animal {
    public function makeSound() {
        echo "Meow";
    }
}

$animal = new Cat();
$animal->makeSound();
?>

```

So sánh Interface và Abstract

- Interface không thể có các thuộc tính, còn abstract thì có thể
- Tất cả phương thức của interface đều là public, abstract thì có thể là public hoặc protected
- Tất cả phương thức của interface đều là abstract nên chúng ta không thể viết code bên trong. Abstract có thể có các phương thức không cần sử dụng abstract method

8. Kiến thức cơ bản lập trình với CSDL: PHP & MySQL

8.1 Khái niệm lập trình với CSDL

Lập trình với CSDL (Cơ sở dữ liệu) là quá trình tạo ra và quản lý các ứng dụng máy tính để tương tác với các hệ quản trị cơ sở dữ liệu. CSDL là nơi lưu trữ và tổ chức thông tin theo cấu trúc nhất định, và lập trình với CSDL cho phép chúng ta thực hiện các thao tác như truy xuất, thêm, sửa, xóa dữ liệu trong CSDL.

Trong lập trình với CSDL, ngôn ngữ phổ biến được sử dụng là SQL (Structured Query Language), được dùng để tương tác với các hệ quản trị CSDL như MySQL, Oracle, SQL Server, và PostgreSQL. SQL cung cấp các câu lệnh như SELECT, INSERT, UPDATE, DELETE để thực hiện các thao tác trên dữ liệu.

Ngoài việc sử dụng SQL, lập trình với CSDL còn bao gồm việc sử dụng các ngôn ngữ lập trình khác như Java, C#, Python để kết nối với CSDL, xử lý dữ liệu và thực hiện các tác vụ khác như xử lý logic, giao diện người dùng, và quản lý kết nối đến CSDL.

8.2 Khái niệm cơ bản phân biệt CSDL quan hệ. Vd: MySQL và SQL Server

CSDL quan hệ (Relational Database Management System - RDBMS) là một loại hệ quản trị cơ sở dữ liệu dựa trên mô hình quan hệ, trong đó dữ liệu được tổ chức thành các bảng có mối quan hệ với nhau thông qua khóa chính và khóa ngoại.

1. MySQL:

- MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở (open-source) phát triển bởi Oracle Corporation.

- MySQL có sự đơn giản, nhẹ nhàng và dễ sử dụng, phù hợp cho các ứng dụng nhỏ và trung bình.
- Hỗ trợ nhiều ngôn ngữ lập trình như PHP, Java, Python, C++,... và là sự lựa chọn phổ biến cho ứng dụng web.
- MySQL có tính bảo mật tốt, khả năng xử lý cao và khả năng mở rộng linh hoạt.

2. SQL Server:

- SQL Server là một hệ quản trị cơ sở dữ liệu phát triển bởi Microsoft Corporation.
- SQL Server chủ yếu được sử dụng trong môi trường doanh nghiệp với quy mô lớn và yêu cầu tính năng mạnh mẽ và bảo mật cao.
- Hỗ trợ cho các ngôn ngữ lập trình như C#, Java, Python và cung cấp tích hợp tốt với các sản phẩm và công nghệ Microsoft khác.
- SQL Server có nhiều tính năng mở rộng như hỗ trợ phân vùng, dịch vụ tích hợp, quản lý dữ liệu di động, và tích hợp công nghệ đám mây.

8.3 Kiến thức cơ bản lập trình PHP & mySQL. Vd: các lỗi thường gặp khi kết nối CSDL, kiểu recordset, đóng kết nối khi thao tác xong...

a) Kết nối CSDL:

Để kết nối với CSDL MySQL trong PHP, bạn có thể sử dụng hàm `mysqli_connect()` hoặc PDO (PHP Data Objects).

Một lỗi thường gặp là kết nối CSDL không thành công. Điều này có thể xảy ra khi thông tin kết nối (host, tên người dùng, mật khẩu, tên CSDL) không chính xác hoặc máy chủ CSDL không hoạt động.

Ví dụ kết nối MySQL trong PHP sử dụng `mysqli_connect()`:

```

index.php

<?php
$servername = "localhost";
$username = "your_username";
$password = "your_password";
$dbname = "your_database";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Kết nối CSDL thất bại: " . mysqli_connect_error());
}

```

b) Kiểu recordset

Trong PHP, để thực hiện truy vấn SQL và lấy dữ liệu từ CSDL, bạn sử dụng các hàm như `mysqli_query()` hoặc `PDO::query()`.

Hàm `mysqli_query()` trả về một đối tượng `mysqli_result` chứa kết quả của truy vấn. Đối tượng này được gọi là recordset.

Ví dụ lấy dữ liệu từ CSDL bằng PHP và **MySQL**:

```

index.php

<?php
$sql = "SELECT * FROM users";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        echo "ID: " . $row["id"] . " - Name: " . $row["name"] . "<br>";
    }
} else {
    echo "Không có bản ghi nào.";
}

```

c) Đóng kết nối

Sau khi thực hiện các thao tác với CSDL, quan trọng là đóng kết nối để giải phóng tài nguyên và tránh việc chiếm dụng kết nối không cần thiết.

Sử dụng hàm `mysqli_close()` hoặc phương thức `PDO::close()` để đóng kết nối CSDL.

Ví dụ đóng kết nối MySQL trong PHP:

```
mysqli_close($conn);
```

8.4 Truy xuất CSDL bằng PHP

```
<?php
```

```
$servername = "localhost";
```

```
$username = "your_username";
```

```
$password = "your_password";
```

```
$database = "your_database";
```

```
$conn = new mysqli($servername, $username, $password, $database);
```

```
if ($conn->connect_error) {
```

```
    die("Kết nối CSDL thất bại: " . $conn->connect_error);
```

```
}
```

```
// Thêm người dùng mới (Create)
```

```
$name = "John Doe";
```

```
$email = "johndoe@example.com";
```

```
$password = "password123";
```

```
$stmt = $conn->prepare("INSERT INTO users (name, email, password) VALUES  
(?, ?, ?)");
```

```
$stmt->bind_param("sss", $name, $email, $password);
```

```
if ($stmt->execute()) {
```

```
    echo "Thêm người dùng thành công";
```

```
} else {
```

```
    echo "Lỗi: " . $stmt->error;
```

```
}
```

```
$stmt->close();
```


// Lấy một người dùng dựa trên ID (Read)

```
$user_id = 1;
```

```
$stmt = $conn->prepare("SELECT * FROM users WHERE id = ?");
```

```
$stmt->bind_param("i", $user_id);
```

```
$stmt->execute();
```

```
$result = $stmt->get_result();
```

```
if ($result->num_rows > 0) {
```

```
    $row = $result->fetch_assoc();
```

```
    echo "ID: " . $row["id"] . " - Name: " . $row["name"] . " - Email: " .
```

```
$row["email"];
```

```
} else {
```

```
    echo "Không tìm thấy người dùng.";
```

```
}
```

```
$stmt->close();
```

// Lấy danh sách người dùng (Read)

```
$stmt = $conn->prepare("SELECT * FROM users");
```

```
$stmt->execute();
```

```
$result = $stmt->get_result();
```

```
if ($result->num_rows > 0) {
```

```
    while ($row = $result->fetch_assoc()) {
```

```
        echo "ID: " . $row["id"] . " - Name: " . $row["name"] . " - Email: " .
```

```
$row["email"] . "<br>";
```

```
    }
```

```
} else {
```

```
    echo "Không có người dùng nào.";
```

```
}
```

```
$stmt->close();
```

// Cập nhật thông tin người dùng (Update)

```
$user_id = 1;
```

```
$new_name = "Jane Smith";
```

```
$stmt = $conn->prepare("UPDATE users SET name = ? WHERE id = ?");
```

```
$stmt->bind_param("si", $new_name, $user_id); //s là string của name, i là int id
```

```

if ($stmt->execute()) {
    echo "Cập nhật thông tin người dùng thành công";
} else {
    echo "Lỗi: " . $stmt->error;
}

$stmt->close();

// Xóa người dùng (Delete)
$user_id = 2;

$stmt = $conn->prepare("DELETE FROM users WHERE id = ?");
$stmt->bind_param("i", $user_id); // i là integer

if ($stmt->execute()) {
    echo "Xóa người dùng thành công";
} else {
    echo "Lỗi: " . $stmt->error;
}

$stmt->close();

$conn->close();
?>

```

Trong *bind_param*: i - integer, d - double, s - string, b – blob

8.5 Kết hợp HTML Form và CSDL bằng PHP

a) Form đăng nhập

```

<form method="POST" action="login.php">
    <label for="username">Tên người dùng:</label>
    <input type="text" name="username" id="username" required><br>
    <label for="password">Mật khẩu:</label>
    <input type="password" name="password" id="password"
required><br>
    <input type="submit" value="Đăng nhập">
</form>

```

```

<?php
// Kết nối đến cơ sở dữ liệu (thay đổi thông tin kết nối theo cấu hình của
bạn)
$servername = "localhost";
$username = "your_username";
$password = "your_password";
$dbname = "your_database";

$conn = new mysqli($servername, $username, $password, $dbname);

// Kiểm tra kết nối
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Lấy dữ liệu từ form
$username = $_POST['username'];
$password = $_POST['password'];

// Chuẩn bị truy vấn SQL với prepare statement
$sql = "SELECT * FROM users WHERE username = ? AND password
= ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ss", $username, $password);
$stmt->execute();

$result = $stmt->get_result();

if ($result->num_rows == 1) {
    // Đăng nhập thành công
    echo "Đăng nhập thành công!";
} else {
    // Đăng nhập không thành công
    echo "Tên người dùng hoặc mật khẩu không đúng!";
}

```

```
}
```

```
// Đóng kết nối  
$stmt->close();  
$conn->close();  
?>
```

b) Form đăng kí

```
<form method="POST" action="register.php">  
  <label for="username">Tên người dùng:</label>  
  <input type="text" name="username" id="username" required><br>  
  <label for="password">Mật khẩu:</label>  
  <input type="password" name="password" id="password"  
required><br>  
  <label for="email">Email:</label>  
  <input type="email" name="email" id="email" required><br>  
  <input type="submit" value="Đăng ký">  
</form>
```

```
<?php  
// Kết nối đến cơ sở dữ liệu (thay đổi thông tin kết nối theo cấu hình của  
bạn)  
$servername = "localhost";  
$username = "your_username";  
$password = "your_password";  
$dbname = "your_database";
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Kiểm tra kết nối  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}
```

```
// Lấy dữ liệu từ form  
$username = $_POST['username'];
```

```

$password = $_POST['password'];
$email = $_POST['email'];

// Chuẩn bị truy vấn SQL để chèn dữ liệu vào cơ sở dữ liệu
$sql = "INSERT INTO users (username, password, email) VALUES
('$username', '$password', '$email')";

if ($conn->query($sql) === TRUE) {
    echo "Đăng ký thành công!";
} else {
    echo "Lỗi: " . $sql . "<br>" . $conn->error;
}

// Đóng kết nối
$conn->close();
?>

```

8.6 Khái niệm cơ bản về SQL injection. Từ khóa: “php mysql SQL injection”:

SQL Injection là một kỹ thuật tấn công phổ biến mà tin tặc sử dụng để chèn các câu lệnh SQL độc hại vào câu lệnh SQL được thực thi bởi ứng dụng. Điều này có thể xảy ra khi ứng dụng không xử lý đầu vào từ người dùng một cách an toàn, cho phép tin tặc chèn mã SQL độc hại vào các trường dữ liệu đầu vào.

Cách phòng tránh mysql injection trong php:

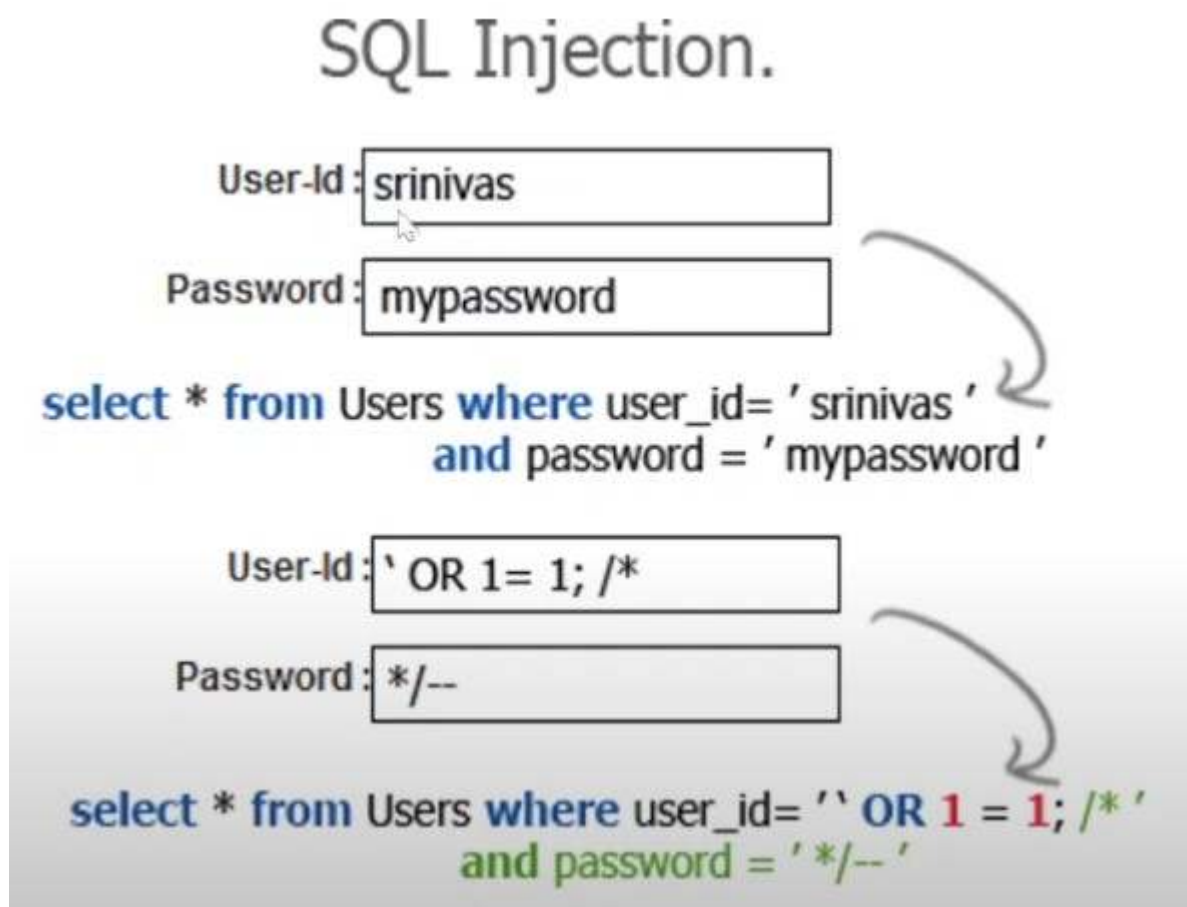
- Sử dụng Prepared Statements hoặc Parameterized Queries để truyền các giá trị đầu vào từ người dùng một cách an toàn.
- Kiểm tra và xử lý đầu vào từ người dùng để loại bỏ hoặc hạn chế các ký tự đặc biệt không an toàn.
- Sử dụng các hàm bảo mật như `mysqli_real_escape_string` hoặc `PDO::quote` để xử lý đầu vào từ người dùng trước khi sử dụng trong câu lệnh SQL.
- Hạn chế quyền truy cập CSDL chỉ cho phép ứng dụng thực hiện các hoạt động cần thiết.

- Cập nhật và bảo mật ứng dụng bằng cách sử dụng phiên bản mới nhất của PHP và MySQL và áp dụng các bản vá bảo mật và cập nhật phần mềm thường xuyên.

Ví dụ về câu lệnh sql bị injection:

SELECT * FROM USER WHERE USERNAME = 'admin'
AND PASSWORD = ''or 1 = '1'

Phần gạch dưới là phần được người dùng nhập vào và bị tấn công



9. Các câu hỏi về lập trình bao gồm thiết kế và xử lý các form liên quan đến đồ án (bao gồm phần end user và phần admin)

XÂY DỰNG BỘ THỰC HÀNH HƯỚNG DẪN SỬ DỤNG PHP KẾT NỐI CƠ SỞ DỮ LIỆU MYSQL, HIỂN THỊ NỘI DUNG LÊN GIAO DIỆN, THÊM, SỬA, XÓA, TÌM KIẾM DỮ LIỆU.

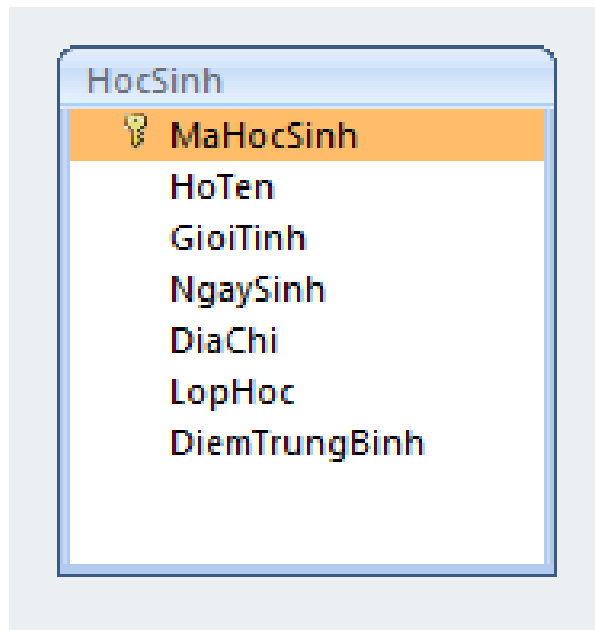
Yêu cầu:

Xây dựng trang web quản lý học sinh:

Table hocsinh gồm các trường dữ liệu sau:

(**MaHocSinh**, HoTen, GioiTinh, NgaySinh, DiaChi, LopHoc, DiemTrungBinh)

- Hiển thị danh sách tất cả học sinh.
- Hiển thị tất cả học sinh nam.
- Thêm học sinh.
- Cập nhật thông tin học sinh theo mã học sinh.
- Xóa học sinh theo Mã học sinh.



Hướng dẫn:\

Các bước thực hiện kết nối đến database

1. Kết nối với CSDL
2. Xây dựng câu truy vấn dữ liệu
3. Thực thi câu truy vấn
4. Xử lý kết quả trả về từ câu truy vấn
5. Ngắt kết nối CSDL

Xây dựng các trang để xử lý bao gồm:

ThemHS.php

CapNhatHS.php?MaHS=1

XoaHS.php?MaHS=1

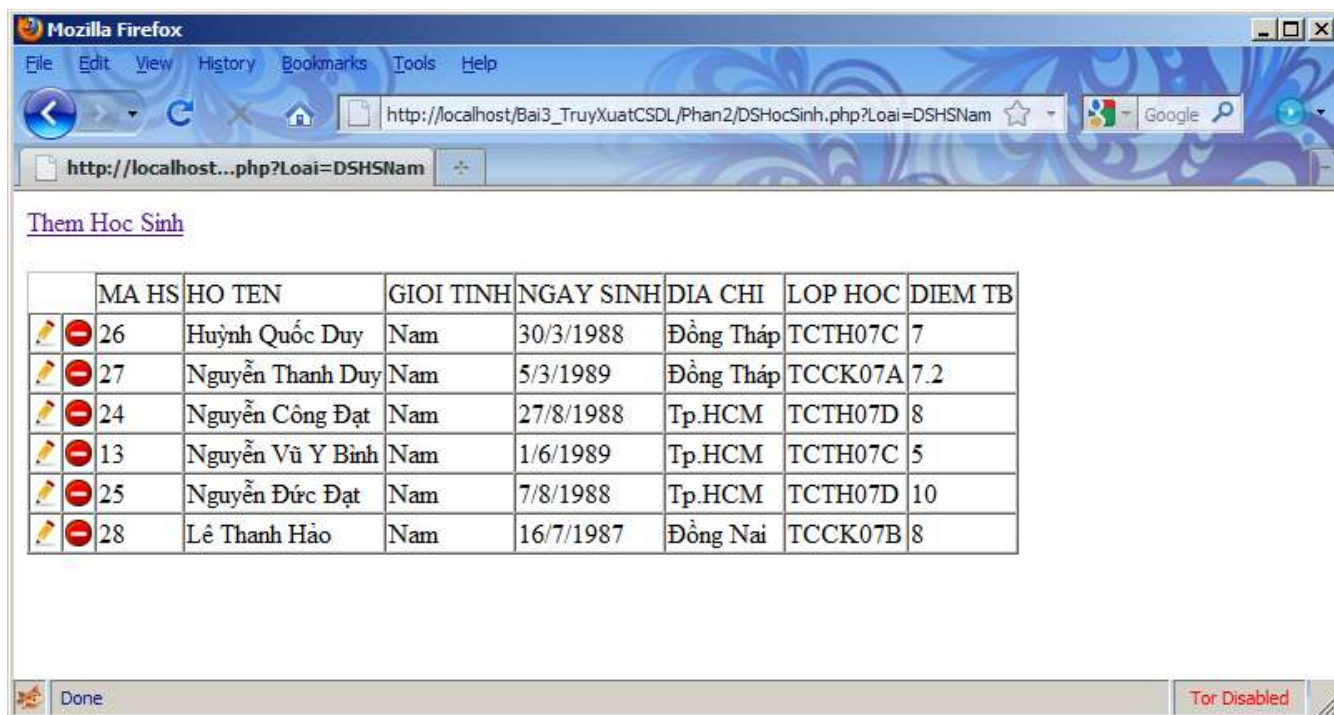
DSHocSinh.php?Loai=TatCa

DSHocSinh.php?Loai=DSHSNam

Để thực hiện được các chức năng này sinh viên phải hoàn thành các câu truy vấn select, insert, update, delete. Sau đó sử dụng các câu truy vấn vào thực hiện các chức năng: hiển thị, thêm, sửa, xóa và tìm kiếm.

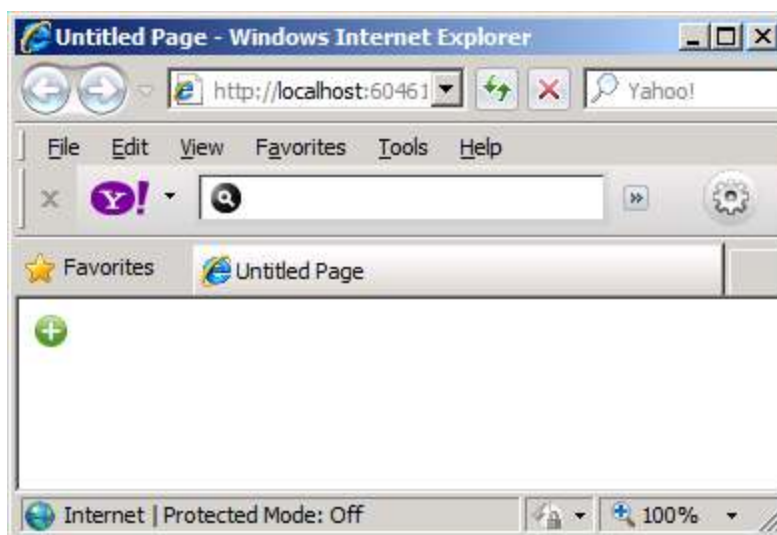
CÁC MÀN HÌNH GIAO DIỆN TƯƠNG ỨNG VỚI CÁC CHỨC NĂNG.

Trang Thêm Học sinh (ThemHS.php)



Hình: hiển thị thêm/xóa/sửa

Hiển thị link có chứa các icon thêm dạng HTML:



Mã nguồn:

```
<a id='h1' href='ThemHS.php>
```

```
<img src='Icon/insert-icon.png' alt='Thêm HS'style='border-  
width:0px;'/>  
</a>
```

Hình: thêm học sinh

CODE DEMO

```
<html>  
<head>  
    <meta http-equiv="content-type" content="text-html; charset=utf-8">  
</head>  
  
<body>  
<?php  
if (isset($_POST['HoTen']))  
{  
    $con = mysqli_connect("localhost","root","");  
  
    if (!$con)  
    {  
        die('Could not connect: ' . mysqli_error());  
    }  
  
    mysqli_select_db($con , "qlhocsinh");  
    mysqli_query($con , "SET NAMES 'UTF8'");
```

```
$sql="INSERT INTO hocsinh (HoTen, GioiTinh, NgaySinh, DiaChi,
LopHoc, DiemTrungBinh)
```

```
VALUES
```

```
('$_POST[HoTen]','$_POST[GioiTinh]','$_POST[NgaySinh]','$_POST[Dia
Chi]','$_POST[LopHoc]','$_POST[DiemTB]");
```

```
if (!mysqli_query($con, $sql))
```

```
{
```

```
    die('Error: ' . mysqli_error());
```

```
}
```

```
echo "Da them thanh cong <br/>";
```

```
mysqli_close($con);
```

```
}
```

```
?>
```

```
<form action="" method="post">
```

```
Ho ten: <input type="text" name="HoTen" /><br/>
```

```
Gioi tinh: <input name="GioiTinh" type="radio" value="1"/> Nam
```

```
    <input name="GioiTinh" type="radio" value="0"/> Nu<br/>
```

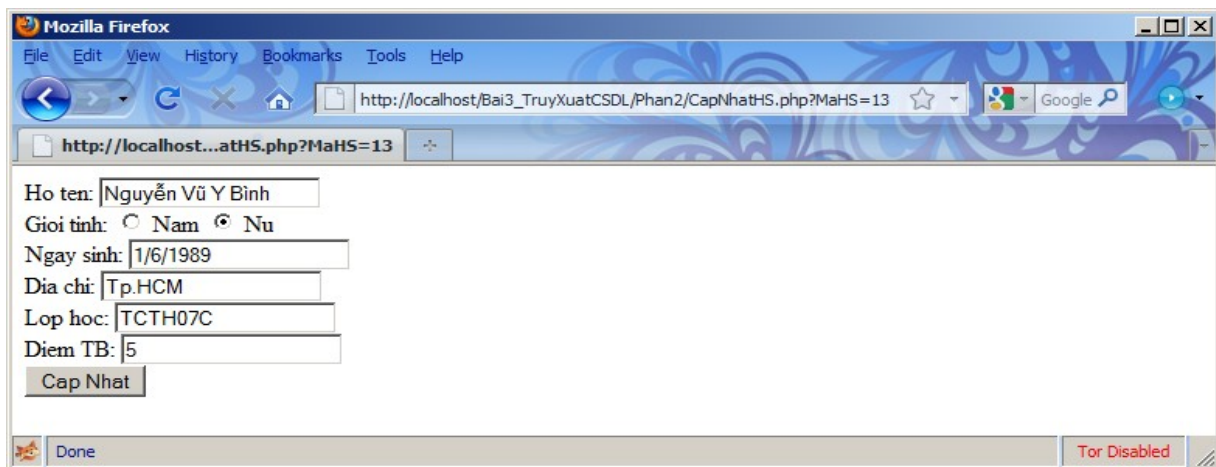
```
Ngay sinh: <input type="text" name="NgaySinh" /><br/>
```

```
Dia chi: <input type="text" name="DiaChi" /><br/>
```

```
Lop hoc: <input type="text" name="LopHoc" /><br/>
```

```
Diem TB: <input type="text" name="DiemTB" /><br/>
<input type="submit" value="Them"/>
</form>
</body>
</html>
```

Cập Nhật học sinh: Khi nhấn vào nút cập nhật gọi trang CapNhatHS.php và truyền tham số là MaHS mà muốn cập nhật thông tin (CapNhatHS.php?MaHS=1).



Hình: cập nhật học sinh

CODE DEMO

```
<html>
<head>
    <meta http-equiv="content-type" content="text-html;
    charset=utf-8">
</head>
<body>
```

```

<?php
//load du lieu doc tu chuo querystring (CapNhatHS.php?MaHS=1)
if (isset($_GET['MaHS']))
{
    $con = mysqli_connect("localhost","root","");
    if (!$con)
    {
        die('Could not connect: ' . mysqli_error());
    }
    mysqli_select_db($con , "qlhocsinh");
    mysqli_query($con , "SET NAMES 'utf8'");
    $MaHocSinh = $_GET['MaHS'];
    $sql = "SELECT * FROM hocsinh WHERE MaHocSinh=" .
$MaHocSinh ."";
    $result = mysql_query($con , $sql);
    $HoTen = "";
    $GioiTinh = "";
    $LopHoc = "";
    $DiaChi = "";
    $DiemTB = "";

    if($row = mysqli_fetch_array($result))
    {

```

```

        $HoTen = $row['HoTen'];
        $GioiTinh = $row['GioiTinh'];
        $NgaySinh = $row['NgaySinh'];
        $DiaChi = $row['DiaChi'];
        $LopHoc = $row['LopHoc'];
        $DiemTB = $row['DiemTrungBinh'];
    }
    mysqli_close($con);
}

//Cap nhat du lieu
if (isset($_POST['btncapnhat']))
{
    $con = mysqli_connect("localhost","root","");
    if (!$con)
    {
        die('Could not connect: ' . mysqli_error());
    }
    mysqli_select_db($con , "qlhocsinh");
    mysqli_query($con , "SET NAMES 'UTF8'");

    $sql="UPDATE hocsinh SET HoTen='$_POST[HoTen]',
GioiTinh='$_POST[GioiTinh]', NgaySinh='$_POST[NgaySinh]',
DiaChi='$_POST[DiaChi]', LopHoc='$_POST[LopHoc]',
DiemTrungBinh='$_POST[DiemTB]' WHERE MaHocSinh="

```

```

$MaHocSinh . """;

    if (!mysqli_query($con, $sql))
    {
        die('Error: ' . mysqli_error());
    }

    echo "Da cap nhat thong tin hoc sinh thanh cong <br/>";
    mysqli_close($con);
}
?>

<form action="" method="post">
Ho ten: <input type="text" name="HoTen" value='<?php if
(isset($HoTen)) echo $HoTen?>'/><br/>
Gioi tinh: <input name="GioiTinh" type="radio" value="1"
checked='<?php if (isset($GioiTinh) && $GioiTinh=="1") echo
"CHECKED" ?>'/> Nam

        <input name="GioiTinh" type="radio" value="0" checked='<?
php if (isset($GioiTinh) && $GioiTinh=="0") echo "CHECKED" ?
>'/> Nu<br/>
Ngay sinh: <input type="text" name="NgaySinh" value='<?php if
(isset($NgaySinh)) echo $NgaySinh?>'/><br/>
Dia chi: <input type="text" name="DiaChi" value='<?php if
(isset($DiaChi)) echo $DiaChi?>'/><br/>
Lop hoc: <input type="text" name="LopHoc" value='<?php if
(isset($LopHoc)) echo $LopHoc?>'/><br/>
Diem TB: <input type="text" name="DiemTB" value='<?php if

```

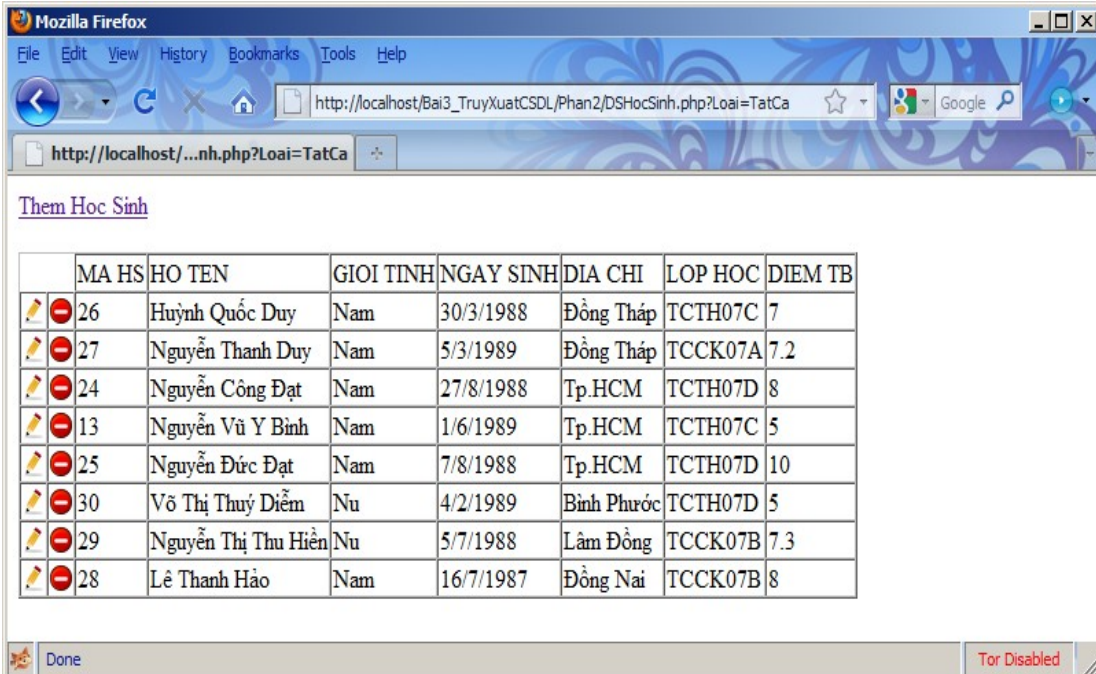
```
(isset($DiemTB)) echo $DiemTB?>'><br/>
<input type="submit" name="btncapnhat" value="Cap Nhat"/>
</form>
</body>
</html>
```









Danh sách Tất cả Học sinh:

<DSHocSinh.php?Loai=TatCa>

hoặc

DSHocSinh.php



	MA HS	HO TEN	GIOI TINH	NGAY SINH	DIA CHI	LOP HOC	DIEM TB
	26	Huỳnh Quốc Duy	Nam	30/3/1988	Đồng Tháp	TCTH07C	7
	27	Nguyễn Thanh Duy	Nam	5/3/1989	Đồng Tháp	TCCK07A	7.2
	24	Nguyễn Công Đạt	Nam	27/8/1988	Tp.HCM	TCTH07D	8
	13	Nguyễn Vũ Y Bình	Nam	1/6/1989	Tp.HCM	TCTH07C	5
	25	Nguyễn Đức Đạt	Nam	7/8/1988	Tp.HCM	TCTH07D	10
	30	Võ Thị Thủy Diễm	Nu	4/2/1989	Bình Phước	TCTH07D	5
	29	Nguyễn Thị Thu Hiền	Nu	5/7/1988	Lâm Đồng	TCCK07B	7.3
	28	Lê Thanh Hào	Nam	16/7/1987	Đồng Nai	TCCK07B	8

Hiển thị table dạng HTML

MA HS	HO TEN	GIOI TINH	NGAY SINH	DIA CHI	LOP HOC	DIEM TB
HS001	A	Nam	01/12/1990	TPHCM	CDTH07B	9.0
HS002	B	Nữ	01/02/1990	DONG NAI	CDTH07A	8.5

CODE THAM KHẢO

```
<table cellpadding='0' border='1' id='gv_DSHocSinh'>
```

```
  <tr>
```

```
    <td>MA HS</td>
```

```
    <td>HO TEN</td>
```

```
    <td>GIOI TINH</td>
```

```
    <td>NGAY SINH</td>
```

```
    <td>DIA CHI</td>
```

```
    <td>LOP HOC</td>
```

```
    <td>DIEM TB</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>HS001</td>
```

```
    <td>A</td>
```

```
    <td>Nam</td>
```

```
    <td>01/12/1990</td>
```

```
    <td>TPHCM</td>
```

```
    <td>CDTH07B</td>
```

```
    <td>9.0</td>
```

```
  </tr>
```

```
<tr>
  <td>HS002</td>
  <td>B</td>
  <td>Nữ</td>
  <td>01/02/1990</td>
  <td>DONG NAI</td>
  <td>CDTH07A</td>
  <td>8.5</td>
</tr>
</table>
```

CODE DEMO

DSHocSinh.php

```
<html>
<head>
  <meta http-equiv="content-type" content="text-html; charset=utf-8">
</head>

<body>
<?php
  $con = mysqli_connect("localhost","root","");
  if (!$con)
```

```
{
    die('Could not connect: ' . mysql_error());
}
mysqli_query($con, "SET NAMES 'utf8'");
mysqli_select_db($con , "qlhocsinh");
$result = mysqli_query($con , "SELECT * FROM hocsinh");
while($row = mysqli_fetch_array($result))
{
    echo $row['MaHocSinh'] . " " . $row['HoTen'];
    echo "<br />";
}
mysqli_close($con);
?>
</body>
</html>
```

Danh sách Tất cả Học sinh Nam:

DSHocSinh.php?Loai=DSHSNam

Them Hoc Sinh

MA HS	HO TEN	GIOI TINH	NGÀY SINH	ĐỊA CHỈ	LỚP HỌC	ĐIỂM TB
26	Huỳnh Quốc Duy	Nam	30/3/1988	Đồng Tháp	TCTH07C	7
27	Nguyễn Thanh Duy	Nam	5/3/1989	Đồng Tháp	TCCK07A	7.2
24	Nguyễn Công Đạt	Nam	27/8/1988	Tp.HCM	TCTH07D	8
13	Nguyễn Vũ Y Bình	Nam	1/6/1989	Tp.HCM	TCTH07C	5
25	Nguyễn Đức Đạt	Nam	7/8/1988	Tp.HCM	TCTH07D	10
28	Lê Thanh Hào	Nam	16/7/1987	Đồng Nai	TCCK07B	8

Hình: Danh sách tất cả học sinh nam

Danh sách học sinh (tất cả, học sinh nam)

```
<html>
<head>
    <meta http-equiv="content-type" content="text-html; charset=utf-8">
</head>

<body>
    <a id="ThemHS" href="ThemHS.php">Them Hoc Sinh</a> <br/>
    <br/>
<?php
    $con = mysqli_connect("localhost","root","");
    if (!$con)
    {
        die('Could not connect: ' . mysqli_error());
    }
}
```

```

}
mysqli_query($con, "SET NAMES 'utf8'");
mysqli_select_db($con, "qlhocsinh");
$sql = "";
if (isset($_GET['Loai']))
{
    $Loai = $_GET['Loai'];
    if ($Loai == "TatCa")
        $sql = "SELECT * FROM hocsinh";
    else if ($Loai == "DSHSNam")
        $sql = "SELECT * FROM hocsinh WHERE GioiTinh=1";
}
else
{
    $sql = "SELECT * FROM hocsinh";
}
$result = mysqli_query($con, $sql);
//Tao the table dung de hien thi danh sach hoc sinh
$s = "";
$s = $s . "<table cellpadding='0' border='1' id='dsHocSinh'> ";
$s = $s . "<tr>";
$s = $s . "<td></td>";
$s = $s . "<td></td>";
$s = $s . "<td>MA HS</td>";
$s = $s . "<td>HO TEN</td>";
$s = $s . "<td>GIOI TINH</td>";

```

```

$s = $s . "<td>NGAY SINH</td>";
$s = $s . "<td>DIA CHI</td>";
$s = $s . "<td>LOP HOC</td>";
$s = $s . "<td>DIEM TB</td>";
$s = $s . "</tr>";

while($row = mysqli_fetch_array($result))
{
    //Lay thong tin hoc sinh
    $MaHS      = $row['MaHocSinh'];
    $HoTen     = $row['HoTen'];
    $GioiTinh = "";
    if ($row['GioiTinh'] == "1")
        $GioiTinh = "Nam";
    else
        $GioiTinh = "Nu";

    $NgaySinh = $row['NgaySinh'];
    $DiaChi   = $row['DiaChi'];
    $LopHoc   = $row['LopHoc'];
    $DiemTB   = $row['DiemTrungBinh'];

    $s = $s . "<tr>";
    $s = $s . "<td><a id='h1' href='CapNhatHS.php?MaHS=".$MaHS .
"> <img src='Icon/edit-icon.png' alt='cap nhat' style='border-
width:0px;'/></a></td>";
    $s = $s . "<td><a id='h1' href='XoaHS.php?MaHS=" . $MaHS . ">

```

```

<img src='Icon/delete-icon.png' alt='xoa' style='border-width:0px;'/></a></td>";

        $s = $s . "<td>" . $MaHS . "</td>";
        $s = $s . "<td>" . $HoTen . "</td>";
        $s = $s . "<td>" . $GioiTinh . "</td>";
        $s = $s . "<td>" . $NgaySinh . "</td>";
        $s = $s . "<td>" . $DiaChi . "</td>";
        $s = $s . "<td>" . $LopHoc . "</td>";
        $s = $s . "<td>" . $DiemTB . "</td>";
        $s = $s . "</tr>";

    }
    $s = $s . "</table>";

    //xuat ket qua
    echo $s;
    mysqli_close($con);
?>
</body>
</html>

```

Xóa học sinh: Khi nhấn vào nút xóa gọi trang XoaHS.php và truyền tham số là MaHS mà muốn xóa thông tin (XoaHS.php?MaHS=1).

```

<html>
<body>
<?php

```

```

if (isset($_GET['MaHS']))
{
    $con = mysqli_connect("localhost","root","");
    if (!$con)
    {
        die('Could not connect: ' . mysqli_error());
    }
    mysqli_select_db($con , "qlhocsinh");
    $sql="DELETE FROM hocsinh WHERE MaHocSinh=" . $_GET['MaHS'] .
    "";
    if (!mysqli_query($con, $sql))
    {
        die('Error: ' . mysqli_error());
    }
    echo "Da xoa thong tin hoc sinh thanh cong <br/>";
    mysqli_close($con);
    header("Location: DSHocSinh.php");
}
?>

```

Tìm kiếm học sinh: (tương tự như hiển thị học sinh theo phái)

**XÂY DỰNG BỘ THỰC HÀNH HƯỚNG DẪN SỬ DỤNG
SESSION XỬ LÝ CHO FORM ĐĂNG NHẬP.**

Registration

Họ và tên

Địa chỉ

Gmail

Số Điện Thoại

Tên Đăng Nhập

Mật khẩu

Xác nhận mật khẩu

Đăng ký

Đã sở hữu tài khoản ? **Đăng nhập**

Hình: Form đăng ký tài khoản người dùng

CODE THAM KHẢO

```
<div class="register-form">
  <div class="register-container">
    <div class="register-title">Registration</div>
    <div id="note"></div>
    <form name="frmregister" action="XuLy/XuLyRegister.php"
id="RegisterForm" method="get" onSubmit="return
check_register1();">
      <div class="user-details">
        <div class="reg-input-box">
          <span class="details">Họ và tên</span>
          <input type="text" id="reg-fullname" name="reg-
fullname" placeholder="Nhập họ tên đầy đủ" >
        </div>
        <div class="reg-input-box">
```

```

        <span class="details">Tên Đăng Nhập</span>
        <input type="text" id="reg-name" name="reg-name"
placeholder="Nhập tên đăng nhập" >
    </div>

    <div class="reg-input-box">
        <span class="details">Địa chỉ</span>
        <input type="text" id="reg-address" name="reg-address"
placeholder="Nhập địa chỉ đang cư trú" >
    </div>

    <div class="reg-input-box">
        <span class="details">Mật khẩu</span>
        <input type="password" id="reg-password" name="reg-
password" placeholder="Nhập mật khẩu" >
    </div>

    <div class="reg-input-box">
        <span class="details">Gmail</span>
        <input type="email" id="reg-mail" name="reg-mail"
placeholder="Nhập địa chỉ gmail" >
    </div>

    <div class="reg-input-box">
        <span class="details">Xác nhận mật khẩu</span>
        <input type="password" id="reg-repassword"
placeholder="Nhập lại mật khẩu" >
    </div>

    <div class="reg-input-box">

```

```

        <span class="details">Số Điện Thoại</span>
        <input type="text" id="reg-phone" name="reg-
phone"placeholder="Nhập số điện thoại" >
    </div>
</div>
<div class="register-button">
    <input id="reg-submit-btn" type="submit" value="Đăng
ký">
</div>

<div class="register-bottom-text">
    <p class="reg-text1">Đã sở hữu tài khoản ?</p>
    <a href="login.php"><p class="reg-text2">Đăng
nhập</p></a>
</div>

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

```

Javascript kiểm tra dữ liệu nhập và thông báo khi sai định dạng.

```

var pass, repass, phone, regname;
function check_register() {
    pass = document.getElementById('reg-password');

```

```

repass = document.getElementById('reg-repassword');
phone = document.getElementById('reg-phone').value;
regname = document.getElementById('reg-name').value;
fullname = document.getElementById('reg-fullname').value;
address = document.getElementById('reg-address').value;
mail = document.getElementById('reg-mail').value;
if (fullname == "") {
    document.getElementById('note').innerText = "Vui lòng nhập đầy đủ họ và tên";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-fullname').focus();
    return false;
}
else if (regname == "" || regname.length < 4) {
    document.getElementById('note').innerText = "Tên đang nhập phải dài hơn 4 kí tự";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-name').focus();
    return false;
}
else if (address == "") {
    document.getElementById('note').innerText = "Vui lòng điền đầy đủ địa chỉ cư trú";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-address').focus();
    return false;
}

```

```
else if (pass.value.length < 6) {
    document.getElementById('note').innerText = "Mật khẩu phải dài hơn 6 kí  
tự";
    document.getElementById('note').style.color = "red";
    pass.focus();
    return false;
}
else if (mail == "") {
    document.getElementById('note').innerText = "Vui lòng điền đầy đủ địa chỉ  
Gmail";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-mail').focus();
    return false;
}
else if (pass.value != repass.value) {
    document.getElementById('note').innerText = "Mật khẩu không khớp";
    document.getElementById('note').style.color = "red";
    repass.focus();
    pass.focus();
    return false;
}
else if (phone[0] != 0 || phone.length < 4) {
    document.getElementById('note').innerText = "Số điện thoại không hợp  
lệ";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-phone').focus();
    return false;
}
```

```
}  
    return true;  
}
```

Kiểm Tra Định Dạng Form Theo Biểu Thức Chính Quy.

```
function validateEmail(email) {  
    var re = /^[S+@\S+\.\S+]/;  
    return re.test(email);  
}  
  
function validatePhone(phone) {  
    var re = /^0(\d{9}|9\d{8})$/;  
    return re.test(phone);  
}  
  
function validateName(regname)  
{  
    // bắt đầu là KH tiếp theo là 3 ký tự số  
    var re=/^KH(\d{3})$/  
    return re.test(regname);  
}  
  
function validatePass(pass)  
{  
    // Tối thiểu là 8 ký tự  
    re=/.{8,}/  
    return re.test(pass);  
}
```

```
function validatePass_Repass(pass, repass)
{
    if(pass==repass)
        return true;
    else
        return false;
}

function check_register()
{
    pass = document.getElementById('reg-password').value;
    repass = document.getElementById('reg-repassword').value;
    phone = document.getElementById('reg-phone').value;
    regname = document.getElementById('reg-name').value;
    fullname = document.getElementById('reg-fullname').value;
    address = document.getElementById('reg-address').value;
    mail = document.getElementById('reg-mail').value;
    if(!validatePhone(phone))
    {
        document.getElementById('note').innerText = "Số điện thoại không hợp  
lệ";
        document.getElementById('note').style.color = "red";
        document.getElementById('reg-phone').focus();
        document.getElementById('reg-phone').select();
        return false;
    }
    else if(!validateEmail(mail))
```

```

{

    document.getElementById('note').innerText = "Email không hợp lệ";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-mail').focus();
    document.getElementById('reg-mail').select();
    return false;
}
else if(!validateName(regname))
{
    document.getElementById('note').innerText = "Tên đăng nhập không hợp
lệ";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-name').focus();
    document.getElementById('reg-name').select();
    return false;
}
else if(!validatePass(pass))
{
    document.getElementById('note').innerText = "Password nhập không hợp
lệ";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-password').focus();
    document.getElementById('reg-password').select();
    return false;
}
else if(!validatePass_Repass(pass, repass))

```



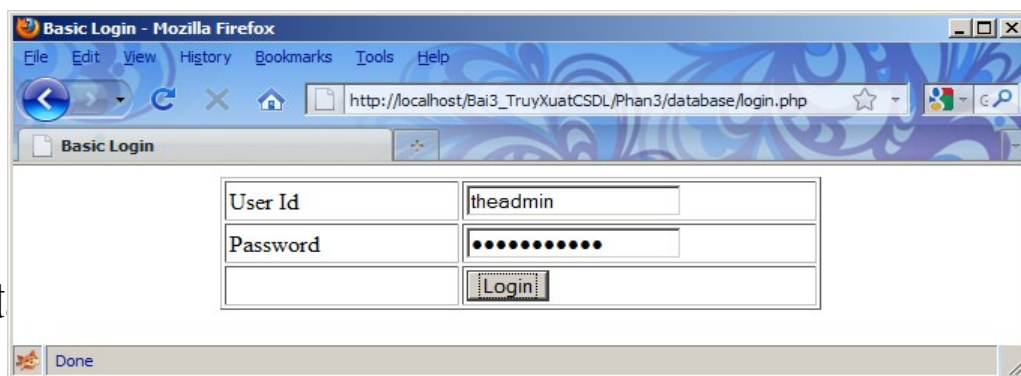
```

{
    document.getElementById('note').innerText = "Password và Repass không hợp lệ";
    document.getElementById('note').style.color = "red";
    document.getElementById('reg-repassword').focus();
    document.getElementById('reg-repassword').select();
    return false;
}
return true;
}

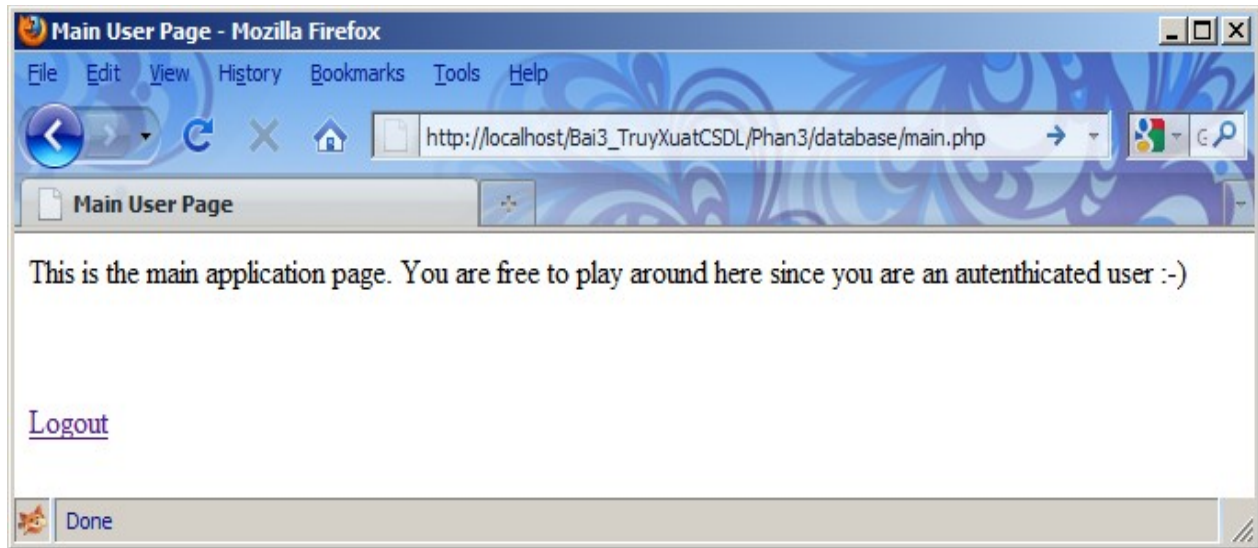
```

Chức năng đăng nhập với thông tin cơ bản (User Id, Password) từ CSDL

Khi vào trang web sẽ xuất hiện màn hình đăng nhập (login.php), trong bài này User Id và Password được lấy từ CSDL (bảng tbl_auth_user (user_id, user_password)).



Nếu đã đăng nhập thành công, người dùng sẽ được chuyển đến màn hình chính (main.php). Trong màn hình này có link đến trang đăng xuất (logout.php).



Mã nguồn minh hoa:

config.php

```
<?php
// db properties
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$dbname = 'tutor';
?>
```

opendb.php

```
<?php

$conn = mysqli_connect ($dbhost, $dbuser, $dbpass) or die ('I cannot connect to
the database because: ' . mysqli_error());

mysqli_select_db ($con, $dbname);
```

```
?>
```

closedb.php

```
<?php  
  
//mysql_free_result($result);  
  
mysqli_close($conn);  
  
?>
```

main.php

```
<?php  
  
// Bat dau session (quan trong)  
session_start();  
  
  
//kiem tra xem da dang nhap (log in) chua?  
if (!isset($_SESSION['db_is_logged_in']) ||  
$_SESSION['db_is_logged_in'] !== true)  
{  
    //neu trang nay chua dang nhap (log in) thi chuyen huong sang trang  
    login.php  
        header('Location: login.php');  
        exit;  
}  
  
?>  
  
<html>
```

```
<head>
<title>Main User Page</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
</head>
<body>
<p>This is the main application page. You are free to play around here
since you
are an authenticated user :-) </p>
<p>&nbsp;</p>
<p><a href="logout.php">Logout</a> </p>
</body>
</html>
```

login.php

```
<?php
// Bat dau session (quan trong)
session_start();

$errorMessage = "";
if (isset($_POST['txtUserId']) && isset($_POST['txtPassword']))
{
    //goi thuc thi 2 trang config.php va opendb.php
    include("library/config.php");
    include ("library/opendb.php");
```

```

//lay du lieu tu form (method="post")
$userId = $_POST['txtUserId'];
$password = $_POST['txtPassword'];

//kiem tra neu user_id va user_password ton tai trong CSDL
$sql = "SELECT user_id
        FROM tbl_auth_user
        WHERE user_id = '$userId' AND user_password =
PASSWORD('$password')";

//truy van
$result = mysqli_query($con, $sql) or die('Query failed. ' .
mysqli_error());

if (mysqli_num_rows($result) == 1)
{
    // username va password hop le
    // Tao bien session "db_is_logged_in" va gan gia tri bang
true
    $_SESSION['db_is_logged_in'] = true;

    //sau khi dang nhap thanh cong thi chuyen sang trang
main.php
    header('Location: main.php');
    exit;

```

```

    }
    else
    {
        //xuất thông báo lỗi khi username hoặc password không
        hợp lệ
        $errorMessage = 'Sorry, wrong user id / password';
    }

    //gọi thực thi trang closedb.php
    include ("library/closedb.php");
}
?>
<html>
<head>
<title>Basic Login</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
</head>

<body>
<?php
if ($errorMessage != "") {
?>
<p align="center"><strong><font color="#990000"><?php echo
$errorMessage; ?></font></strong></p>

```

```
<?php
}
?>

<form action="" method="post" name="frmLogin" id="frmLogin">
  <table width="400" border="1" align="center" cellpadding="2"
  cellspacing="2">
    <tr>
      <td width="150">User Id</td>
      <td><input name="txtUserId" type="text" id="txtUserId"></td>
    </tr>
    <tr>
      <td width="150">Password</td>
      <td><input name="txtPassword" type="password"
id="txtPassword"></td>
    </tr>
    <tr>
      <td width="150">&nbsp;</td>
      <td><input name="btnLogin" type="submit" id="btnLogin"
value="Login"></td>
    </tr>
  </table>
</form>
</body>
</html>
```

logout.php

```
<?php

// Bat dau session (quan trong)
session_start();

//Neu nguoi dung da dang nhap thanh cong, thi huy bien session
if (isset($_SESSION['db_is_logged_in']))
{
    unset($_SESSION['db_is_logged_in']);
}

//Da dang xuat, quay tro lai trang login.php
header('Location: login.php');
?>
```