

4

BÀI TẬP

Session & Cookies & Web Storage

Lập trình Ứng dụng Web

GVHD: Đỗ Thị Hương Lan

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Tìm hiểu và phân biệt các kỹ thuật dùng session, cookie và web storage trong lưu trữ trạng thái.
- Xây dựng chức năng đăng nhập, đăng ký.
- Xây dựng chức năng lưu trữ trạng thái, thông tin. Ví dụ
 - o Những sản phẩm khách hàng đã xem
 - o Giỏ hàng
 - o Các thông tin: menu của trang, vị trí của khách hàng v.v...

2. Môi trường

- Ngôn ngữ xử lý và Database: tự chọn
- IDE: Notepad, Notepad++, IDE Visual Studio, hoặc IDE phù hợp với ngôn ngữ Backend lựa chọn v.v ...

3. Liên quan

- Sinh viên cần nắm được flowchart làm việc của các chức năng cần xây dựng. Ví dụ: Đăng ký, Đăng nhập, Thêm vào Giỏ hàng, v.v...
- Tìm hiểu thêm về các kỹ thuật mã hóa mật khẩu
- Tìm hiểu về cách thức hoạt động của các API mạng xã hội (Facebook, Google, Twitter v.v...)
- Tham khảo thêm tại Website w3school.com hoặc một số website liên quan khác để nắm vững kiến thức

B. KIẾN THỨC NỀN TẢNG

Cookie, Session, và Web Storage là các cơ chế **lưu trữ dữ liệu** trong các ứng dụng web, giữa chúng có một số khác biệt quan trọng về cách chúng lưu trữ và quản lý dữ liệu.

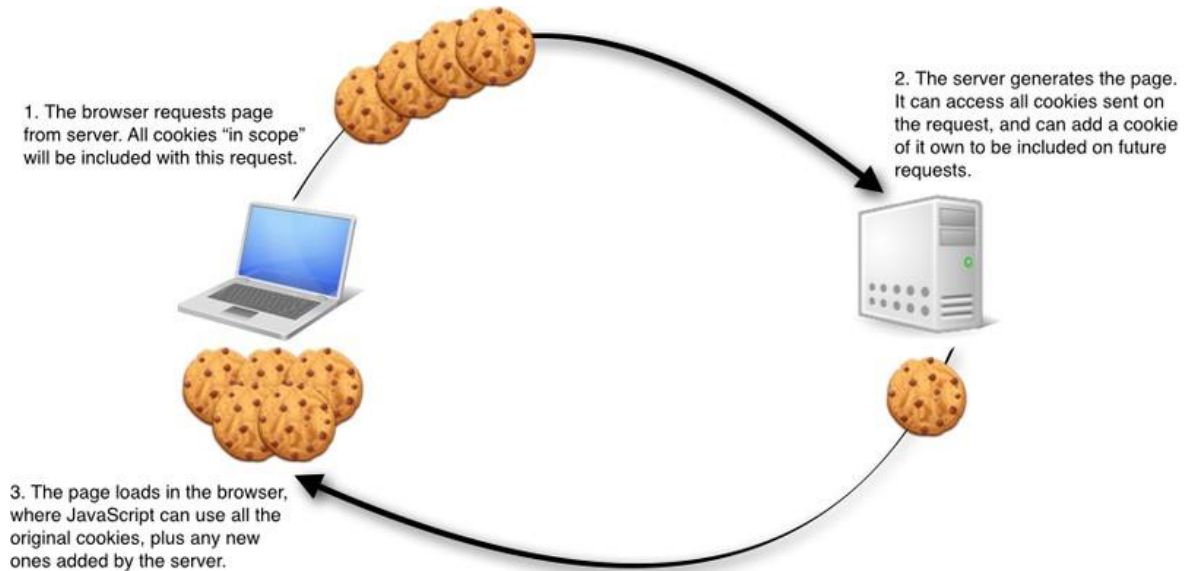
- **Phạm vi lưu trữ:**
 - **Cookie:** Lưu trữ dữ liệu ở cả hai phía, client và server. Cookie được gửi giữa client và server trong mỗi yêu cầu.
 - **Session:** Dữ liệu được lưu trữ chỉ trên phía server. Một session ID được gửi đến client trong cookie để xác định phiên làm việc.
 - **Web Storage:** Lưu trữ dữ liệu chỉ trên phía client và không được gửi đến server mỗi lần yêu cầu.
- **Kích thước lưu trữ:**
 - **Cookie:** Giới hạn khoảng 4 KB dữ liệu.
 - **Session:** Không có giới hạn cụ thể nhưng phụ thuộc vào cấu hình máy chủ và nguồn lực.

- **Web Storage:** Thường lớn hơn, với giới hạn 5-10 MB trên mỗi trang web.
- **Thời gian tồn tại:**
 - **Cookie:** Có thể được thiết lập để tồn tại trong một khoảng thời gian cụ thể hoặc chỉ tồn tại trong suốt phiên làm việc của trình duyệt.
 - **Session:** Tồn tại cho đến khi người dùng đăng xuất hoặc đóng trình duyệt.
 - **Web Storage:** Có thể tồn tại vĩnh viễn hoặc theo cấu hình cụ thể.
- **Truy cập dữ liệu:**
 - **Cookie:** Dữ liệu có thể được truy cập cả ở phía client và server.
 - **Session:** Dữ liệu chỉ có thể được truy cập ở phía server, nhưng một session ID được sử dụng để xác định phiên tương ứng của client.
 - **Web Storage:** Dữ liệu chỉ có thể được truy cập ở phía client.
- **An toàn và bảo mật:**
 - **Cookie:** Có thể bị thay đổi bởi người dùng hoặc bị đánh cắp (cross-site scripting, CSRF).
 - **Session:** An toàn hơn so với cookie vì dữ liệu được lưu trữ ở phía server.
 - **Web Storage:** Dữ liệu cũng có thể bị thay đổi bởi người dùng, nhưng có các biện pháp bảo mật nhất định như Same-Origin Policy.
- **Sử dụng cho mục đích nào:**
 - **Cookie:** Thích hợp cho việc lưu trữ thông tin như thông tin đăng nhập, tùy chọn người dùng nhỏ.
 - **Session:** Thích hợp cho việc lưu trữ thông tin phiên làm việc, như giữ trạng thái đăng nhập.
 - **Web Storage:** Thích hợp cho việc lưu trữ dữ liệu lớn hơn và cần sự tương tác nhanh từ phía client.

1. Cookie

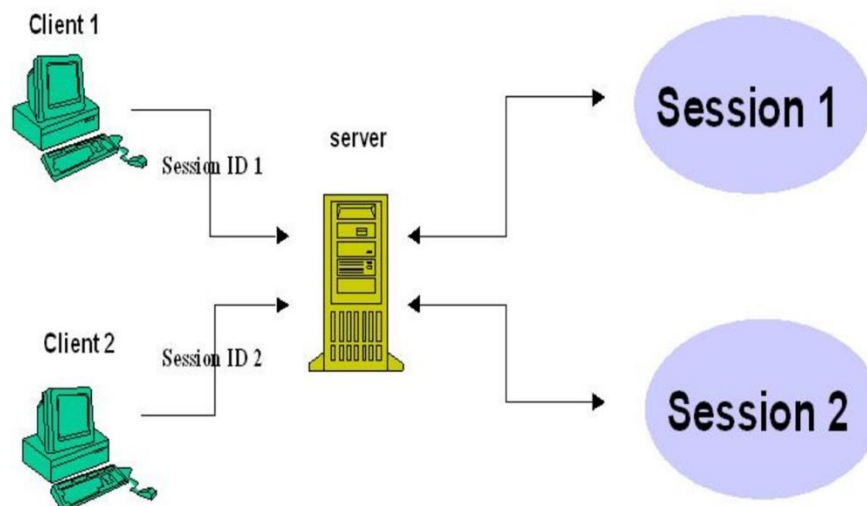
- Cookie là một đoạn văn bản ghi thông tin được tạo ra và lưu trên trình duyệt của máy người dùng. Cookie thường được tạo ra khi người dùng truy cập một website, cookie sẽ ghi nhớ những thông tin như tên đăng nhập, mật khẩu, các tùy chọn do người dùng lựa chọn đi kèm. Các thông tin này được lưu trong máy tính để nhận biết người dùng khi truy cập vào một trang web.
- Cookie thường được dùng để lưu trữ các tùy chọn riêng của trang web từng user, nó là một file nhỏ được Server chỉ định lưu trữ trên máy tính của Client và PHP có thể truy xuất tới được. Và để sử dụng được Cookie thì trình duyệt phải hỗ trợ chức năng này, nếu không thì Cookie trở nên vô dụng.

- Cookie sẽ không bị mất khi bạn đóng ứng dụng, nó phụ thuộc vào thời gian sống mà bạn thiết lập cho nó. Ví dụ bạn thiết lập Cookie lưu trữ thông tin đăng nhập trong vòng 15 phút thì sau 15 phút mà bạn không có một thao tác thay đổi trên nó thì Cookie của bạn sẽ bị mất.

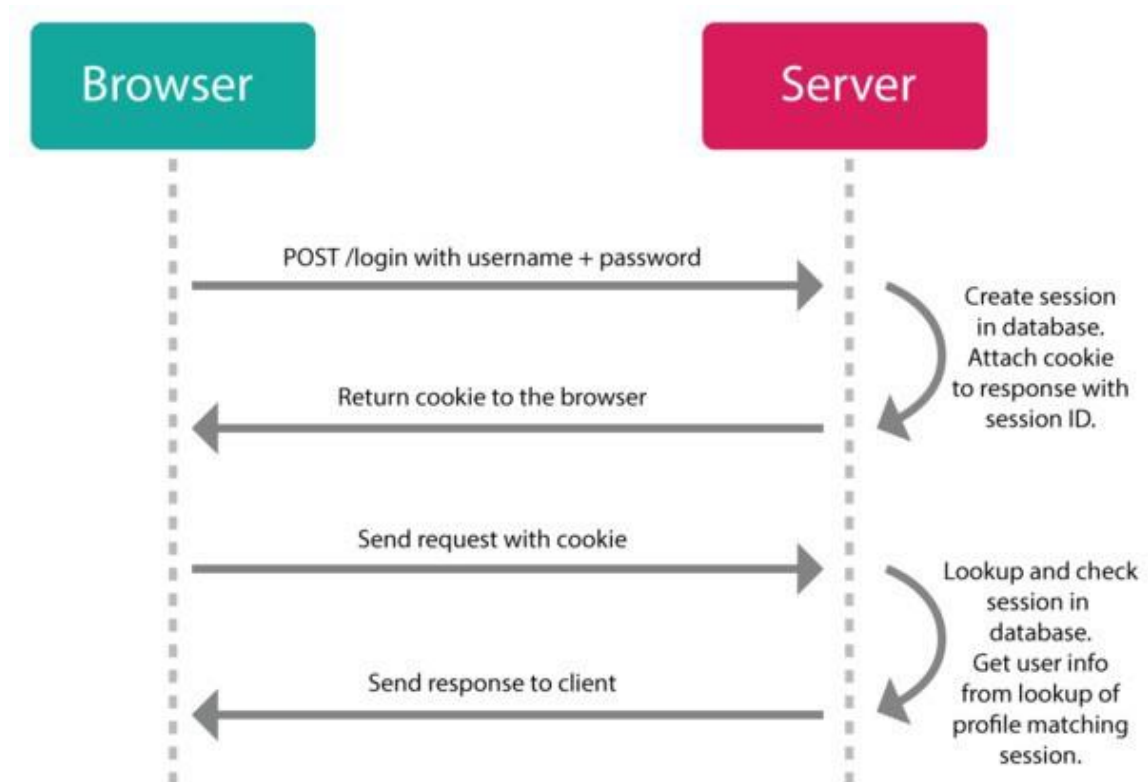


2. Session

- Session hay còn gọi là một phiên làm việc đơn giản là 1 cách để lưu lại dữ liệu của người dùng sử dụng website một cách tạm thời. Một session bắt đầu khi client gửi request đến sever, nó tồn tại xuyên suốt từ trang này đến trang khác trong ứng dụng và chỉ kết thúc khi hết thời gian timeout hoặc khi bạn đóng ứng dụng. Giá trị của session sẽ được lưu trong một tệp tin trên máy chủ.
- Bạn có thể tùy ý quyết định xem nên lưu trữ những thông tin nào vào Session, Thông thường chúng ta chỉ nên lưu trữ những thông tin tạm thời trong session VD như: thông tin đăng nhập, thông tin các sản phẩm trong giỏ hàng (đối với các trang web thương mại điện tử).
- Với mỗi session sẽ được cấp phát một định danh duy nhất SessionID. Khi kết thúc một phiên làm việc và bắt đầu một phiên mới, dĩ nhiên sẽ được cấp một SessionID khác với trước đó.



- Cơ chế hoạt động



• So sánh giữa cookie và session

	Session	Cookies
Không gian lưu trữ	Session lưu trữ ở phía Server	Cookie là các tệp phía máy khách chứa thông tin người dùng
Thời hạn	Một Session kết thúc khi người dùng đóng trình duyệt	Cookie kết thúc tùy thuộc vào thời gian đặt ra
Kích thước	Có thể lưu trữ bao nhiêu dữ liệu tùy thích. Tuy nhiên, nếu lưu trữ quá nhiều sẽ đẩy áp lực về phía Server	Kích thước Cookie tối đa là 4KB
Độ an toàn	Dữ liệu session không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ	Dữ liệu cookie dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client

3. Web Storage

Web Storage là một cơ chế lưu trữ dữ liệu trên trình duyệt web, cho phép ứng dụng web lưu trữ thông tin dưới dạng cặp **key-value (khóa-giá trị)** một cách nhanh chóng và dễ dàng. Có hai loại Web Storage là **localStorage** và **sessionStorage**.

localStorage:

- *Phạm vi lưu trữ:* Dữ liệu được lưu trữ vĩnh viễn trên trình duyệt và không có thời hạn hết hạn (trừ khi bị xóa bởi người dùng hoặc thông qua mã lệnh của ứng dụng).
- *Chia sẻ dữ liệu giữa các trang web:* Dữ liệu lưu trữ trong localStorage sẽ được chia sẻ giữa tất cả các trang web của một miền cụ thể (the same-origin policy).

sessionStorage:

- *Phạm vi lưu trữ:* Dữ liệu được lưu trữ trong suốt phiên làm việc của trình duyệt. Khi phiên làm việc kết thúc (ví dụ: khi trình duyệt đóng cửa sổ), dữ liệu trong sessionStorage sẽ bị xóa.
- *Chia sẻ dữ liệu giữa các trang web:* Dữ liệu lưu trữ trong sessionStorage chỉ được chia sẻ giữa các trang thuộc cùng một cửa sổ trình duyệt.

Web Storage cung cấp một API đơn giản để thao tác với dữ liệu:

- **setItem(key, value):** Đặt một giá trị dưới dạng cặp key-value.
- **getItem(key):** Lấy giá trị dựa trên key.
- **removeItem(key):** Xóa một cặp key-value.
- **clear():** Xóa tất cả key-value trong Web Storage.

Web Storage thường được ưu tiên sử dụng khi cần lưu trữ dữ liệu lớn hơn so với Cookie và khi không cần gửi dữ liệu về server mỗi lần yêu cầu. Nó thích hợp để lưu trữ các thông tin như cài đặt người dùng, trạng thái ứng dụng, hoặc dữ liệu cache. Tuy nhiên, do dữ liệu lưu trữ trên client, cần phải cẩn thận để không lưu trữ thông tin quan trọng và nhạy cảm mà có thể bị thay đổi bởi người dùng.

C. BÀI TẬP

1. Bài 01 – Chức năng Đăng ký

Giao diện minh họa:

ĐĂNG KÝ THÀNH VIÊN

Username*

Name

Email*

Password*

Confirm Password*

Phone number

*Required fields

Yêu cầu:

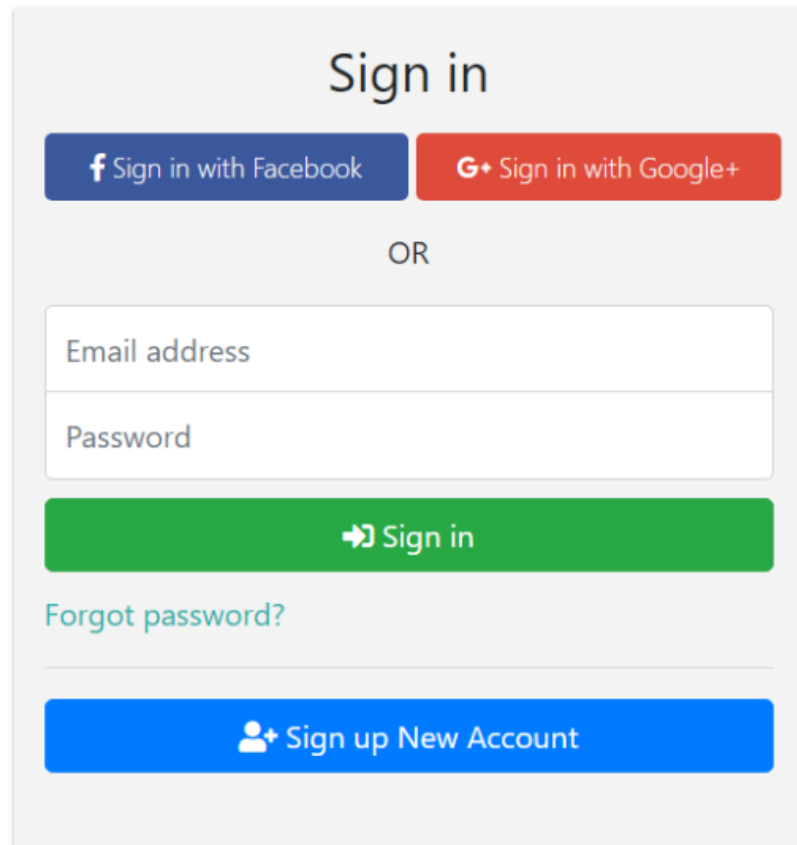
- Thiết kế form Đăng ký với những thông tin cần như giao diện minh họa
- Lập trình xử lý validate dữ liệu trước khi lưu trữ vào CSDL
- Mã hóa mật khẩu trước khi lưu trữ vào CSDL
- Lập trình chức năng kiểm tra username/email đã tồn tại trong CSDL hay chưa
- Lập trình chức năng xử lý việc lưu trữ thông tin đăng ký vào CSDL

Gợi ý:

- Sử dụng jQuery Validate để kiểm tra dữ liệu nhập (Tham khảo Bài tập 2/Lab 2)
- Nên sử dụng hàm băm (hash) để mã hóa mật

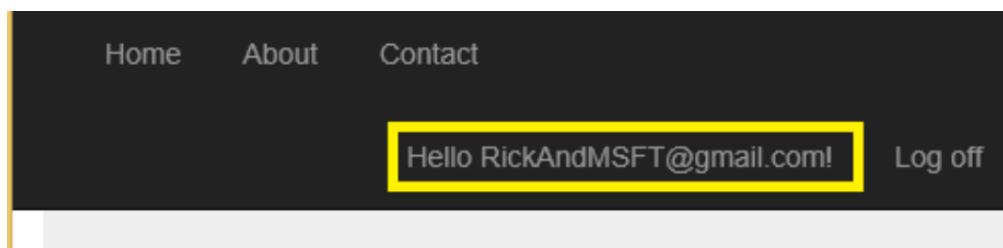
2. Bài 02 – Chức năng Đăng nhập

Giao diện minh họa:



Yêu cầu:

- Thiết kế form như hình minh họa
- Lập trình chức năng kiểm tra thông tin người dùng trong CSDL, nếu tồn tại thì đăng nhập thành công, chuyển về trang trước đó hoặc trang chủ; nếu không thì báo không thành công
- Nếu thành công, tạo **session** cho người dùng, và hiển thị thông tin người dùng ở trang Hồ sơ cá nhân hoặc ở đầu trang

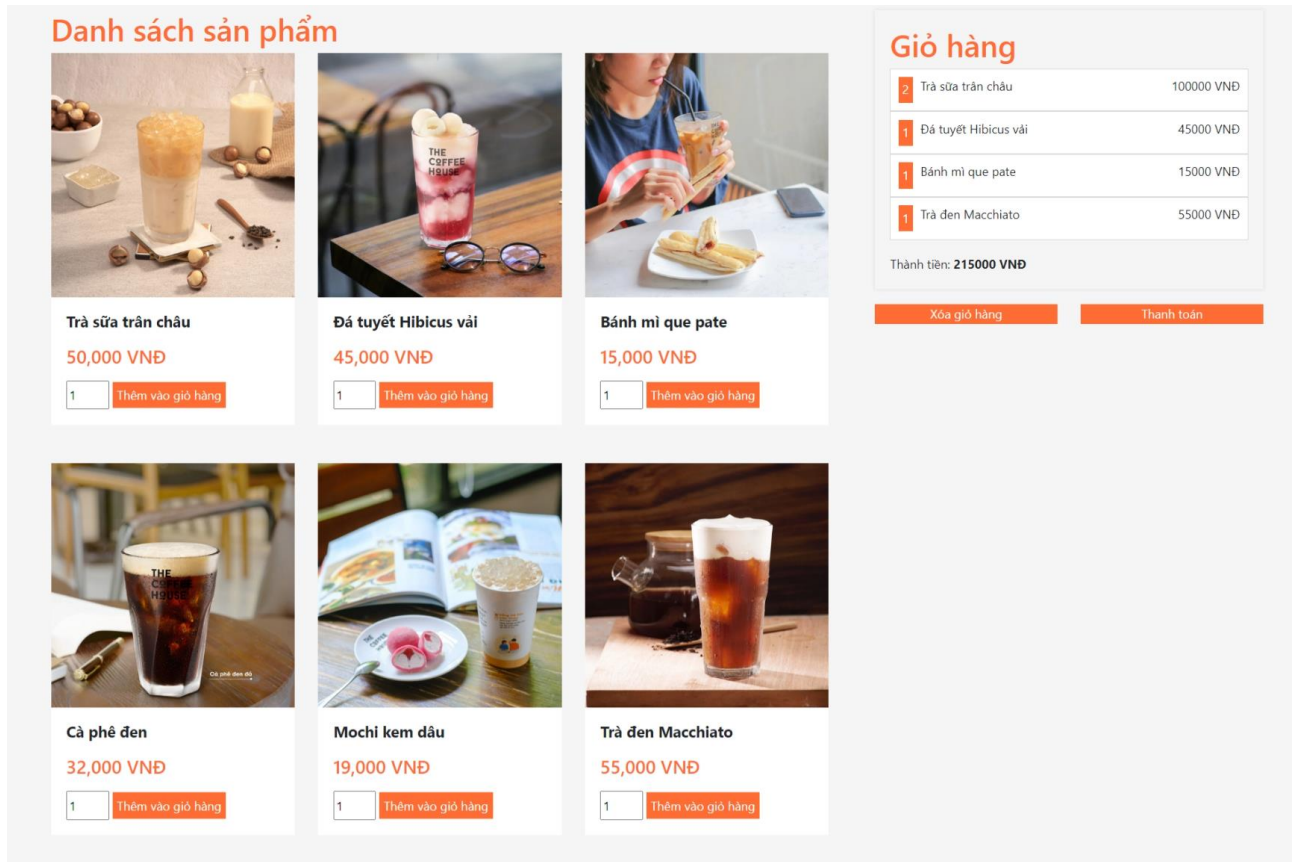


Mở rộng:

- Thiết kế giao diện và chức năng cho phép đăng nhập thông qua các tài khoản Mạng xã hội

3. Bài 03 – Giỏ hàng

Giao diện minh họa:



Yêu cầu:

Viết chức năng cho phép thêm sản phẩm vào giỏ hàng của người dùng

Gợi ý:

- Có thể sử dụng 1 trong các kỹ thuật Cookie, Session, Web Storage để lưu trữ thông tin giỏ hàng (Lưu vào Session ở phía Server, hoặc lưu vào Web Storage ở phía Client)

D. THAM KHẢO

[1] So sánh localStorage, sessionStorage và cookies [Online] Available at: <https://viblo.asia/p/so-sanh-localstorage-sessionstorage-va-cookies-Ljy5VQQklra>

[2] Web Storage API [Online] Available at: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API

[3] Shopping Cart with Local Storage Available at Moodle courses.uit.edu.vn

HẾT