Tecnologies de Desenvolupament per a Internet i Web

Curs 2024-2025

Pràctica: *Botiga virtual.* Sessió 4

Índex

1 Dates

Grups	**Dia**
Grups A, B	18/11
Grup D	19/11
Grups E, F, G, H	20/11
Grups I, J, K	21/11
Grups L, M, N	22/12

2 Objectius

Pes	**Obligatori?**	**Funcionalitat**
Sistema de productes		
0'2	✓	Llistat de categories: filtratge de dades
Sistema d'usuaris		
0'3	✓	Registre: validació a la part de client
0'3	✓	Registre: validació a la part de servidor
0'2	✓	Registre: inici de sessió
Cabàs de compra		
0'3	✓	Cabàs visible a tot el web
0'3	✓	Afegir productes al cabàs
0'3	✓	Pàgina del cabàs

3 Feina prèvia abans de la sessió

- Mireu com filtrar i validar els camps del formulari al servidor[8].
- Mireu com mantenir dades durant tota la sessió de navegació amb PHP[3].

4 Feina durant la sessió i abans de la següent sessió

4.1 Validació del registre d'usuaris a la banda del client

Heu de validar a la part de client el formulari de registre d'un usuari. Podeu fer la validació amb Javascript o HTML5, com vosaltres vulgueu. La validació emprant Javascript serveix per als casos en què es vol personalitzar l'aparença i el missatge de l'error[1].

Recordeu que la validació a la part de client és important sobretot per l'experiència d'usuari, però que sempre cal validar les dades al servidor, que és el següent punt a fer.

4.2 Validació del registre d'usuaris a la banda del servidor

En aquest punt el que heu de fer és validar les dades que us arriben del formulari de registre a la part del servidor, amb PHP per assegurar-nos que són correctes. Amb PHP podeu fer servir la funció `filter_var`[4][2]. Trobareu un llistat dels diferents filtres de validació (**`FILTER_VALIDATE_*`**) que podeu utilitzar a [9].

4.3 Filtratge de dades

Mai no es pot confiar en les dades que un usuari afegeix a les aplicacions web, encara que sigui l'administrador, perquè poden inserir codi maliciós que posi en risc la privacitat dels usuaris. Aquests atacs, anomenats XSS —*Cross-Site Scripting*—, s'eviten filtrant les dades a l'hora de mostrar-les a l'usuari. Filtrar vol dir escapar tots els caràcters que puguin fer que el navegador interpreti el contingut com un *script* de Javascript.

Validar i filtrar són dues coses diferents i complementàries, una no substitueix l'altra: validant ens assegurem que les dades que ens arriben tenen un format vàlid —per exemple, una adreça de correu electrònic. I filtrant ens assegurem que no ens insereixen codi maliciós.

Per exemple, suposem que tenim aquest contingut en un camp d'una taula de la base de dades:

```
<script>alert('TDIW');</script>
```

Si mostreu el contingut d'aquest camp directament per pantalla, us apareixerà un `alert` per pantalla:

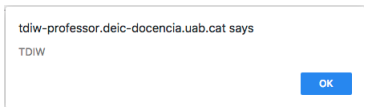

Figura 1: `alert` que es mostra per pantalla si no es filtren les dades

Per prevenir això, el que heu de fer és escapar el contingut que mostrareu de pantalla, de manera que els caràcters especials no seran interpretats pel navegador. Per fer-ho, en PHP podeu utilitzar la següent instrucció:

```php
<?php

$string = htmlentities($string, ENT_QUOTES | ENT_HTML5, 'UTF-8');
```

D'aquesta manera, el contingut que s'enviarà al navegador és el següent:

```
&lt;script&gt;alert&lpar;&apos;TDIW&apos;&rpar;&semi;&lt;&sol;script&gt;
```

Tots els caràcters especials són escapats, i el navegador els interpretarà com a caràcters de text.

Per a la vostra pràctica, només us demanem fer-ho al llistat de categories amb el nom de les categories.

Per fer proves, podeu desar la cadena `<script>alert('TDIW');</script>` al camp de nom de la categoria. Si, quan entreu a la pàgina del llistat de categories us apareix el missatge de l'`alert`, no esteu escapant correctament el contingut.

4.4 Inici de sessió d'usuari

Utilitzarem la variable *superglobal* `$_SESSION`[3] per al manteniment de sessions a PHP. Vegeu un exemple de com funciona a l'apendix A. Mitjançant sessions implementarem l'inici de sessió d'un usuari a la vostra botiga. Per fer-ho, heu d'aprofitar la pàgina d'inici de sessió (login) que vau fer a la primera sessió de pràctiques, de manera que envieu les dades d'inici de sessió d'un usuari i valideu que siguin correctes. Per validar que la contrasenya sigui correcta heu d'utilitzar la funció `password_verify`[5] de PHP.

4.5 Afegir productes al cabàs de compra

Utilitzant les sessions de PHP, heu d'implementar la funcionalitat d'afegir productes al cabàs de compra. A la pàgina de detall de producte que ja vau fer a la sessió anterior, heu de desenvolupar la funcionalitat d'afegir el producte al cabàs, especificant-ne la quantitat. El cabàs s'ha d'actualitzar, sense recarregar la pàgina, a través d'una crida AJAX (amb

jQuery o *Fetch*), i ha de mostrar un missatge de resposta indicant si el producte s'ha afegit correctament o si, per contra, hi ha hagut cap error en fer el procés.

Per implementar aquesta funcionalitat heu de fer servir sessions. En teniu una explicació a l'apèndix A.

4.6 Cabàs visible des de tot el web

Al vostre web heu de tenir visible a totes les pàgines una petita secció amb el resum del cabàs de la compra, indicant-ne, almenys, el nombre total de productes que conté i el seu import total. Aquests valors els heu d'actualitzar quan afegiu productes al cabàs des del detall d'un producte (el que heu fet al punt previ).

A la figura 2 teniu un exemple del cabàs visible a tot el web d'un dels webs del curs 2017-2018.

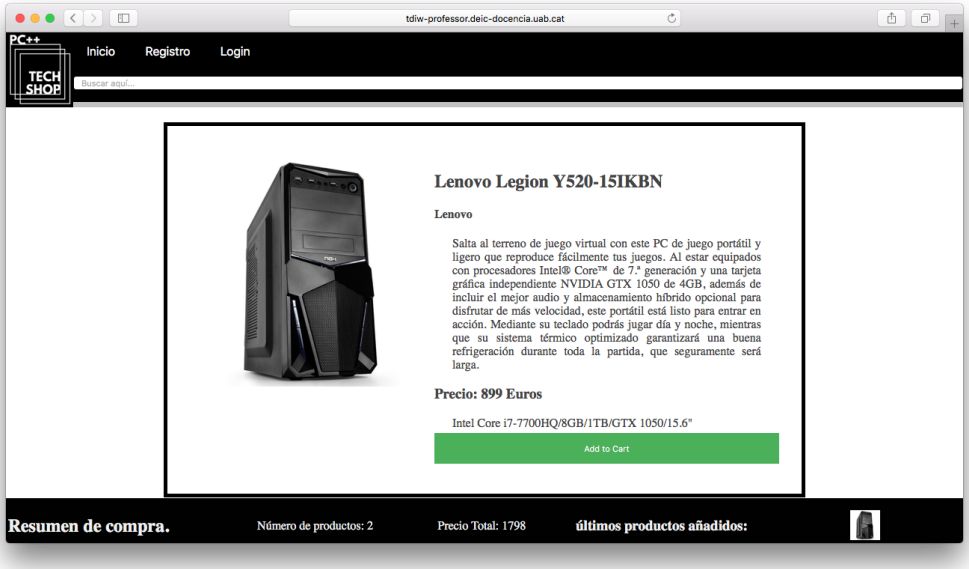

Figura 2: Exemple del cabàs visible a tot el web d'un dels webs del curs 2017-2018

4.7 Pàgina del cabàs

Heu d'afegir una pàgina a la vostra botiga on es mostri el llistat de productes del carret amb les seves quantitats i imports. A aquesta pàgina heu d'afegir un botó —no funcional encara— per acabar la compra.

A la figura 3 teniu un exemple de la pàgina del cabàs d'un dels webs del curs 2017-2018.

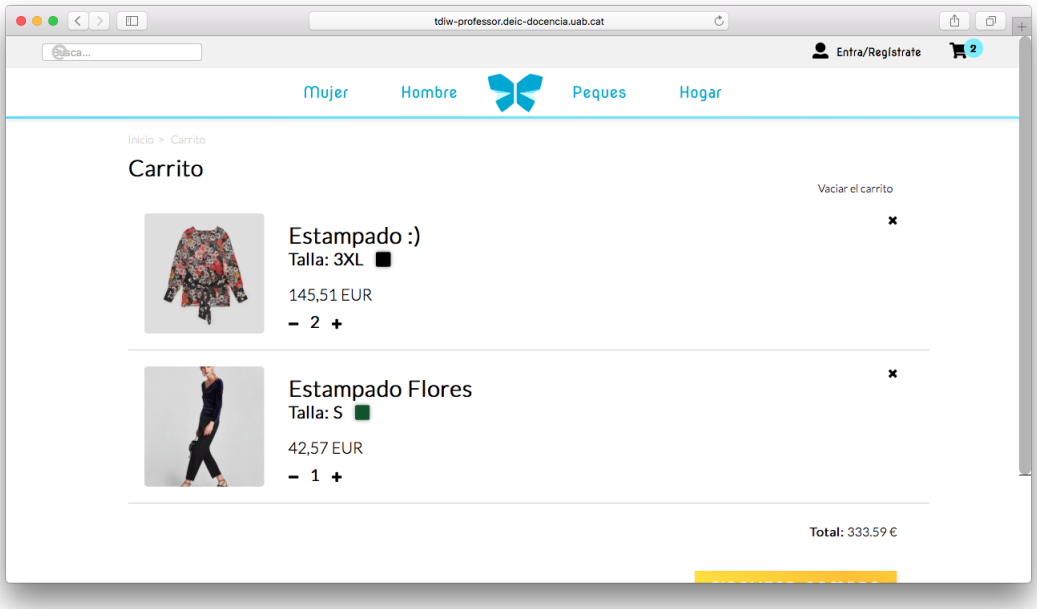

Figura 3: Exemple de pàgina del cabàs d'un dels webs del curs 2017-2018

Apèndixs

A Ús de sessions

Hi ha vegades en què necessitem desar informació de l'usuari entre diferents peticions consecutives, com per exemple els productes que hagi pogut afegir al cabàs de la compra. Per aconseguir això, es fan servir `sessions`[6].

A cada usuari que entra al vostre web se li assigna un identificador únic —l'identificador de sessió— que, amb la configuració per defecte de PHP, s'envia mitjançant galetes. Quan PHP rep aquest identificador a una petició, recupera els valors emmagatzemats de la sessió i els torna a fer disponibles durant tot el flux d'execució.

A PHP existeix una variable *superglobal*, `$_SESSION`, similar a les variables súperglobals que ja coneixeu, com `$_GET` o `$_POST`. Això no obstant, aquesta variable no està sempre disponible, sinó que només ho està un cop heu inicialitzat la sessió.

Per fer-ho, només cal cridar a la funció `session_start()`[7]. Si heu seguit les nostres recomanacions per implementar el patró MVC, només haureu de cridar aquesta funció un cop, a l'inici de tot del vostre encaminador `index.php`.

Un cop inicialitzada la sessió, ja tindreu disponible la variable súperglobal `$_SESSION`, on podreu emmagatzemar els valors que us calguin per a la vostra botiga. Hi ha una cosa que heu de tenir en compte en l'ús d'aquesta variable *superglobal*: la variable `$_SESSION` és un *array* associatiu (diccionari), de manera que les claus (keys) del diccionari han de ser `strings`. És a dir, podreu fer això:

```php
<?php
session_start();

$_SESSION['user_id'] = 1;
```

Però no podreu fer això:

```php
<?php
session_start();

$_SESSION[0] = 1;
```

Referències

[1] MDN. Form data validation - Learn web development | MDN. https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Form_validation#Validating_forms_using_JavaScript.

[2] php.net. Function filter_var for form validation in php. http://php.net/manual/en/filter.examples.validation.php.

[3] PHP.net. PHP: Basic usage - Manual. http://www.php.net/manual/en/session.examples.basic.php.

[4] PHP.net. PHP: filter_var - Manual. http://www.php.net/manual/en/function.filter-var.php.

[5] php.net. PHP: password_verify - Manual. http://php.net/manual/en/function.password-verify.php.

[6] PHP.net. PHP: Sessions - Manual. http://php.net/manual/en/book.session.php.

[7] PHP.net. PHP: session_start - Manual. http://www.php.net/manual/en/function.session-start.php.

[8] P.I.E. Staff. Everything You Need to Know About Preventing Cross-Site Scripting Vulnerabilities in PHP. https://paragonie.com/blog/2015/06/preventing-xss-vulnerabilities-in-php-everything-you-need-know.

[9] w3schools.com. PHP Predefined Filter Constants. https://www.w3schools.com/php/php_ref_filter.asp.