Tecnologies de Desenvolupament per a Internet i Web

Curs 2020-2021

Pràctica: *Botiga virtual.* Sessió 3

Índex

1 Dates **1**

2 Objectius **1**

3 Feina prèvia abans de la sessió **1**

4 Feina durant la sessió i abans de la següent sessió **1**
4.1 Llistat de productes d'una categoria amb AJAX 1
4.2 Detall de producte amb AJAX 3
4.3 Registre d'usuari 4
4.4 Menú desplegable d'usuari 4

Apèndixs **6**

A Consultes parametritzades **6**
A.1 Consultes parametritzades amb paràmetres nominals 6
A.1.1 Amb *arrays* associatius 6
A.1.2 Amb assignació manual de valors 7
A.2 Consultes parametritzades amb paràmetres de substitució . . . 8
A.2.1 Amb *arrays* de valors 9
A.2.2 Amb assignació manual de valors 10

Referències **11**

1 Dates

Grups	**Dia**
Grups A, B	04/11
Grups C, D, E	05/11
Grup F	06/11
Grups G, H, I, J	07/11
Grups K, L, M	08/11

2 Objectius

Pes	**Obligatori?**	**Funcionalitat**
Sistema de productes		
0'2	✓	Llistat de productes amb AJAX
0'1	✓	Detall de producte amb AJAX
Sistema d'usuaris		
0'4	✓	Registre: consultes parametritzades
0'4	✓	Registre: desar contrasenya xifrada
0'2	✓	El meu compte: menú amb `jQuery`

3 Feina prèvia abans de la sessió

- Mireu un tutorial de Javascript bàsic[3][2].
- Mireu un tutorial de jQuery[4].
- Mireu un tutorial de AJAX[1].
- Mireu com executar consultes SQL parametritzades amb PHP[7][9][10].

4 Feina durant la sessió i abans de la següent sessió

4.1 Llistat de productes d'una categoria amb AJAX

Heu de modificar la pàgina del llistat de categories de manera que, en fer clic al títol —o la imatge, si és que en teniu— de cadascuna d'ella, es mostrin la informació resumida dels seus productes sense necessitat de recarregar la pàgina. Per implementar aquesta funcionalitat heu de fer **crides AJAX amb jQuery**.

La informació dels productes que heu de mostrar a aquesta pàgina és la següent:

- Títol.
- Imatge (de moment no la teniu, però ho podeu anar preparant).
- Preu.
- Opcionalment, un botó d'afegir al cabàs.

A la figura 1 teniu un exemple del llistat de productes d'un dels webs del curs 2017-2018.

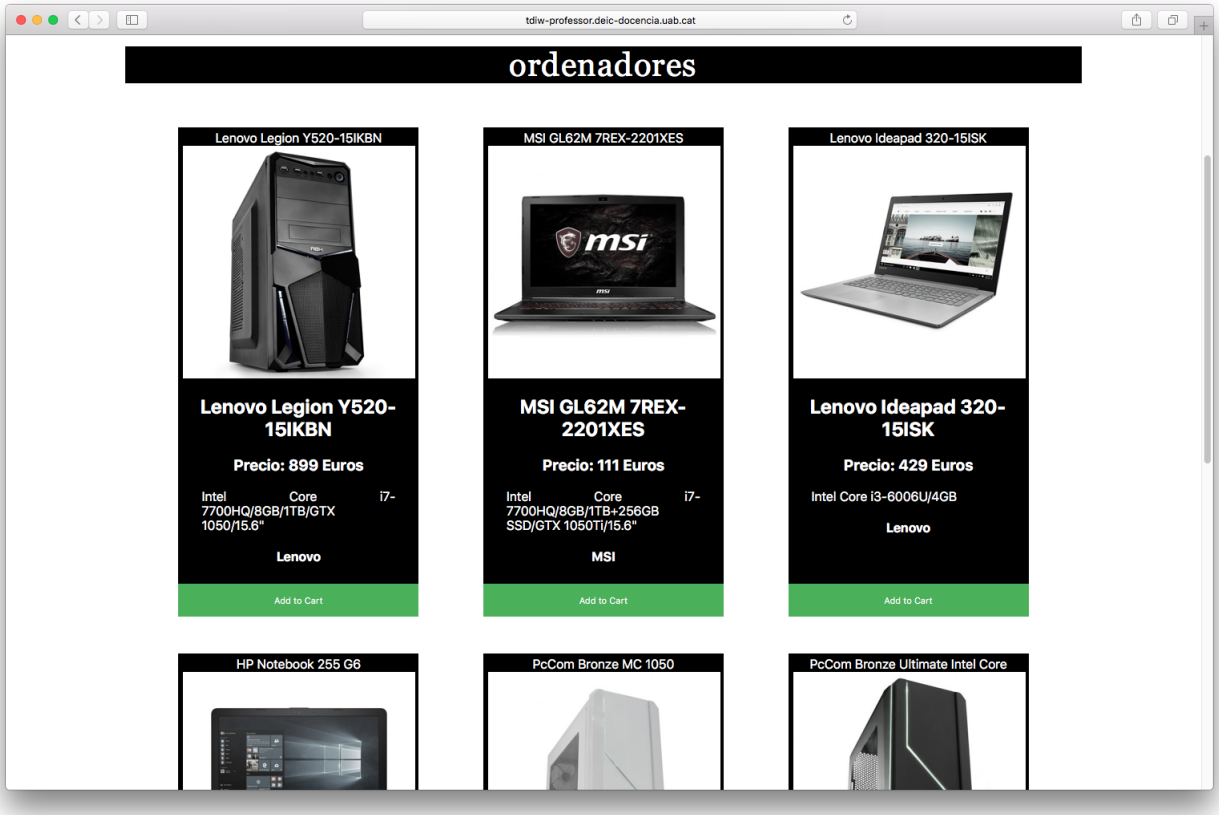

Figura 1: Exemple del llistat de productes d'un dels webs del curs 2017-2018

4.2 Detall de producte amb AJAX

Igual que en el punt anterior, cal que mostreu la pàgina del detall d'un producte —la informació estesa— en fer clic a un producte des del seu llistat, emprant crides AJAX i jQuery, és a dir, sense recarregar la pàgina.

La informació dels productes que heu de mostrar al seu detall és la següent:

- Títol.
- Descripció.
- Imatge (de moment no la teniu, però ho podeu anar preparant).
- Preu.
- Botó d'afegir al cabàs amb una quantitat —si escau—, que encara no serà funcional.

A la figura 2 teniu un exemple del detall de producte d'un dels webs del curs 2017-2018.

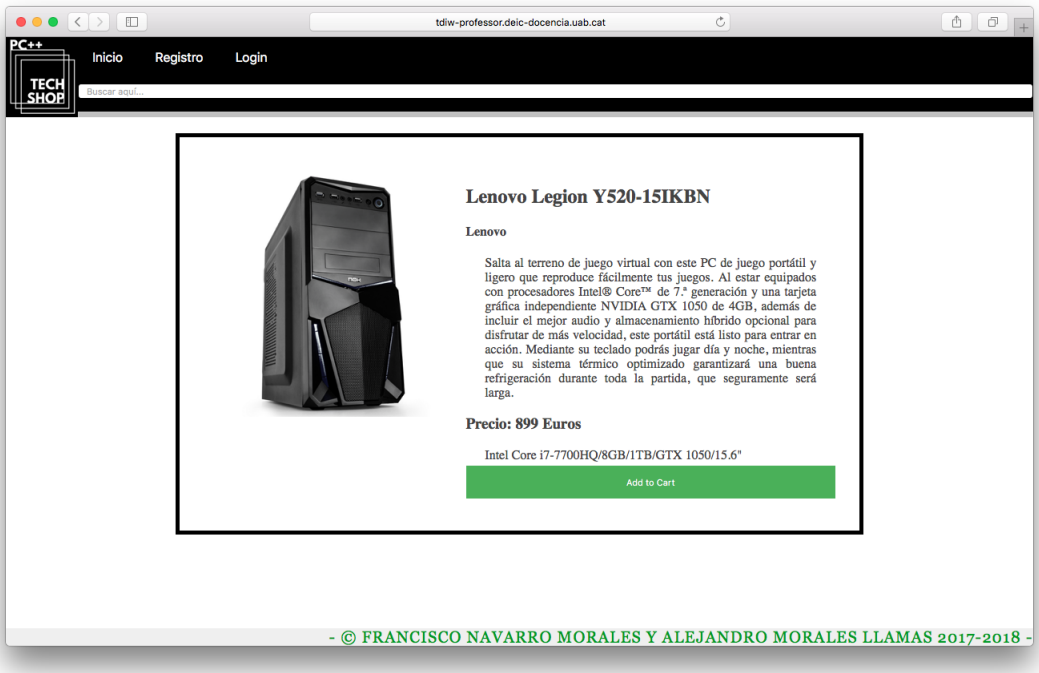

Figura 2: Exemple del detall de producte d'un dels webs del curs 2017-2018

4.3 Registre d'usuari

A partir del formulari de registre d'usuari que vau fer a la sessió 1, heu d'emmagatzemar les dades de l'usuari a la base de dades un cop s'enviï el formulari. De moment no us heu de preocupar de filtrar les dades ni a la part del client ni a la part del servidor, però sí que heu de tenir en compte dos aspectes de seguretat de vital importància:

- Heu de xifrar la contrasenya. A PHP és molt senzill, heu de fer servir la funció `password_hash`[6], que és criptogràficament segura. **Les contrasenyes sempre s'han de desar xifrades a la base de dades**.
- Heu de fer servir consultes parametritzades per emmagatzemar les dades de l'usuari. A la pràctica, per motius de planificació, només us demanem fer aquesta funcionalitat amb consultes parametritzades, però fóra bo que canviéssiu totes les consultes que ja teniu perquè s'executin com a consultes parametritzades, i que les noves consultes que feu a partir d'aquest moment les feu així. **Utilitzar consultes parametritzades permet evitar atacs d'injecció SQL**.

Tingueu present que, encara que feu el registre d'usuaris, de moment no poden iniciar sessió, això ho veurem a la següent sessió de pràctiques.

4.4 Menú desplegable d'usuari

Heu d'afegir al vostre menú de navegació un menú desplegable amb les opcions corresponents a la navegació de l'usuari; aquest menú l'heu d'implementar amb jQuery. Penseu que aquest menú encara no serà totalment funcional, i que variarà en funció de si l'usuari ha iniciat sessió al seu compte o no.

Si l'usuari no ha iniciat sessió, l'única opció disponible al menú serà la d'iniciar sessió.

En canvi, si l'usuari ha iniciat sessió, les opcions d'aquest menú seran les següents:

- El meu compte.
- Les meves compres.
- Tancar sessió.

A la figura 3 teniu un exemple del menú desplegable d'un dels webs del curs 2017-2018.

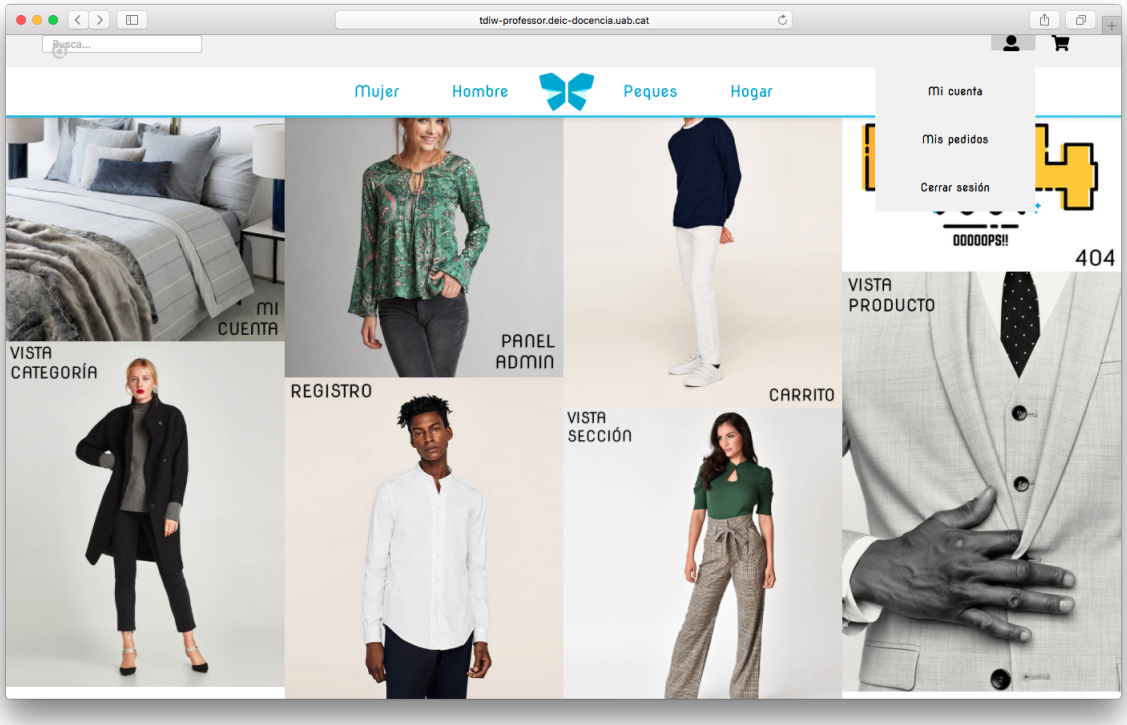

Figura 3: Exemple del menú desplegable d'un dels webs del curs 2017-2018

Apèndixs

A Consultes parametritzades

Els atacs d'injecció SQL són els més comuns a aplicacions web[5]. És una tècnica que pretén obtenir el control de la base de dades a partir d'una consulta de l'aplicació. Sovint, aquests atacs exposen dades privades dels usuaris.

Prevenir-los és senzill, i per aconseguir-ho s'utilitza el que s'anomenen consultes parametritzades[1], que consisteixen a utilitzar un espai reservat o *placeholder* a les consultes. A aquest espai després s'hi afegeix el valor a consultar de manera segura.

Afortunadament, PDO ens permet executar consultes parametritzades de manera simple i senzilla. Hi trobem dos mètodes equivalents, que només difereixen en la manera en què s'escriuen.

A.1 Consultes parametritzades amb paràmetres nominals

El primer mètode consisteix a afegir paràmetres amb un nom concret dins de la consulta SQL.

Suposem que volem agafar tots els productes d'una categoria, la que té l'identificador `3`. El codi PHP que tindríem per definir la consulta SQL podria ser com el següent:

Hem preparat la consulta anterior per tenir-hi un paràmetre, `:category_id`, el nom del qual hem escollit nosaltres. El que ens cal, doncs, és assignar el valor d'aquest paràmetre a l'hora de fer la consulta. Hi ha dues maneres de fer-ho, passant un *array* associatiu amb tots els valors de les variables, o amb una associació manual de valors.

A.1.1 Amb *arrays* associatius

Per fer la consulta amb *arrays* associatius, el que s'ha de fer és simplement passar un *array* al mètode `execute`. A aquest *array* associatiu, les claus són els *placeholders* que hem definit i els valors són els valors que necessitem substituir:

En executar la consulta, se substituirà `:category_id` pel valor de la variable `$categoryId`, que en aquest cas és `3`.

[1] Heu de tenir present que una consulta parametritzada no consisteix executar la consulta a la base de dades dins d'una funció que rep paràmetres.

A.1.2 Amb assignació manual de valors

Per fer la consulta amb assignació manual de valors, el que cal fer és associar cada *placeholder* a un valor amb el mètode `bindValue`[8]:

En executar la consulta, se substituirà `:category_id` pel valor de la variable `$categoryId`, que en aquest cas és `3`.

A.2 Consultes parametritzades amb paràmetres de substitució

Aquest mètode és equivalent a l'anterior però els *placeholders* es defineixen amb el símbol d'interrogació, `?` en lloc d'amb un nom.

Si fem el mateix exemple que al cas anterior, agafar tots els productes d'una categoria, la que té l'identificador `3`, el codi PHP amb què definiríem la consulta SQL podria ser com el següent:

Hem preparat la consulta anterior per tenir-hi un paràmetre, definit amb el símbol d'interrogació. El que ens cal, doncs, és assignar el valor d'aquest paràmetre a l'hora de fer la consulta. De manera anàloga al cas de consultes parametritzades amb valors nominals, existeixen dues maneres de fer-ho: o bé passant un *array* tots els valors de les variables, o bé amb una associació manual de valors.

A.2.1 Amb *arrays* de valors

Per fer la consulta amb *arrays* de valors, el que s'ha de fer és simplement passar un *array* amb els valors al mètode `execute`. A una consulta parametritzada pot haver-hi un nombre arbitrari de paràmetres, de manera que és important que els valors segueixin l'ordre definit amb els *placeholders*. El codi podria ser com el següent:

En executar la consulta, se substituiran els símbols d'interrogació —en aquest cas, només un— pels valors correlatius de l'*array* que s'ha passat com a paràmetre del mètode `execute`.

A.2.2 Amb assignació manual de valors

Per fer la consulta amb assignació manual de valors, el que cal fer és associar cada *placeholder* a un valor amb el mètode `bindValue`[8]:

Fixeu-vos, en aquest cas, que s'associa el valor en l'ordre definit a la consulta, començant per `1` —no per `0`, com comencen els *arrays*.

Referències

[1] Javier Eguiluz. Introducción a AJAX. http://www.librosweb.es/ajax/.

[2] Javier Eguiluz. Introducción a javascript. http://www.librosweb.es/javascript/.

[3] MDN. Tutorials | MDN. https://developer.mozilla.org/ca/docs/Web/Tutorials.

[4] Rebecca Murphey. Fundamentos de jQuery. http://librosweb.es/libro/fundamentos_jquery/.

[5] OWASP. Top 10-2017 Top 10. https://www.owasp.org/index.php/Top_10-2017_Top_10.

[6] php.net. How to calculate the hash of a password using a random salt. http://php.net/manual/en/function.password-hash.php.

[7] PHP.net. PHP: Connections and Connection management - Manual. http://php.net/manual/en/pdo.connections.php.

[8] PHP.net. PHP: PDOStatement::bindValue - Manual. http://php.net/manual/en/pdostatement.bindvalue.php.

[9] PHP.net. PHP: Prepared statements and stored procedures - Manual. http://php.net/manual/en/pdo.prepared-statements.php.

[10] P.I.E. Staff. Preventing SQL Injection in PHP Applications - the Easy and Definitive Guide - Paragon Initiative Enterprises Blog. view-source:https://paragonie.com/blog/2015/05/preventing-sql-injection-in-php-applications-easy-and-definitive-guide.