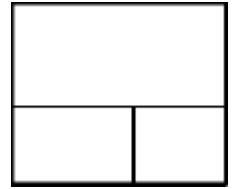


DWES Examen PHP 3-mar-2021



Se te habrá proporcionado un proyecto PhpStorm con este enunciado y tres carpetas, una por ejercicio. Las carpetas son la base sobre la que se recomienda realizar los ejercicios. Incluyen una exportación de la BD cuando es necesario.

[1] Anexar texto (2,00p)

(Solo PHP, sin DAO, sin Ajax...)

Programa una web que contenga un input en el que el usuario pueda introducir texto.

Al abrirse la web, si hay algún texto guardado de antes, se cargará en el input.

Habrà un botón que añadirà (anexará) dicho texto al texto que ya haya, intercalando un espacio.

Si al pulsar el botón el texto estaba vacío, la web debe volver a mostrar el formulario inicial con un mensaje de “Por favor, introduzca algo de texto”. No se pide controlar otro tipo de situaciones (han metido solo espacios, hay espacios a los lados, etc.)

Además, habrá un botón de Reiniciar que volverá al estado inicial, vacío.

Se pide que el texto perdure si cierro y vuelvo a abrir el navegador. Utiliza un mecanismo adecuado para ello, cookies y/o sesiones.

Crea el fichero/ficheros que necesites. Si tu solución tiene copiado de código (por ejemplo, copiado del formulario en dos PHP separados), aspiras a un punto menos que la puntuación del ejercicio. Si la haces monopágina, o bipágina pero evitando ese copiado, aspiras a la puntuación máxima.

[2] Concesionario DAO (4,00p)

(Sí se pide “DAO”, no se pide inicio de sesión ni nada de eso, ni se pide AJAX.)

Importa la BD incluida en el ZIP o, si tienes problemas al hacerlo, crea una nueva BD (schema) y una tabla en una base de datos: BD/schema: “concesionario”. Tabla: “coche”.

Ponle estos campos:

- id: INT, autoincrementable, no nulo, clave primaria
- matricula: VARCHAR(7)
- modelo: VARCHAR(30)
- precio: INT

Introduce algunos valores en la BD, o usa los de la BD incluida.

Se pide:

- Preparar toda la infraestructura que necesitaremos para crear los scripts/páginas que se piden. Esto incluye la(s) clase(s), el “DAO” y demás. Puedes utilizar, al igual que en el resto de ejercicios, material de clase para ello.
- (2,00p) Desarrollar un script para visualizar una tabla con un listado de todos los coches que hay en la BD.
 - La tabla incluirá la matrícula, el modelo y el precio.
 - La tabla incluirá también un link para bajar el precio. El texto del link será “Rebajar”. Este link estará en una columna propia en la tabla.
 - (No es necesario poner links para eliminar coches, ni dar la opción de editarlos como tal.)

- (2,00p) Desarrollar la funcionalidad para rebajar el precio de un coche:
(Si no te da tiempo, deja este ejercicio.)
 - Al pulsar el link en el listado para determinado coche, se hará en PHP lo que se tenga que hacer y se volverá a cargar automáticamente (sin más intervención del usuario) el mismo listado, esta vez, con el precio de ese coche rebajado en 1.000 euros (si costaba 15.400, ahora 14.400). Si el coche iba a pasar a tener un precio inferior a 5.000 euros, se debe dejar en 5.000 y mostrar en la parte superior de la página del listado un mensaje indicando que no se puede vender ningún coche por menos de 5.000.

Queremos utilizar el modelo DAO y queremos implementar las funciones de la manera más organizada posible, con el objetivo de crear una aplicación fácil de mantener y modificar, evitando copiapegas, etc.

Organiza el código lo mejor que puedas, dentro de estas premisas. Un objetivo en este ejercicio es demostrar que sabes organizar código. Por tanto, conseguir la funcionalidad indicada sin organización no es valorable.

No se pide un control de errores exhaustivo más allá de lo indicado explícitamente de los 5.000.

[3] AJAX (4,00p)

Se proporciona la aplicación funcionando según lo visto en clase.

Se pide implementar las siguientes funciones, de forma lo más “Ajax” posible, es decir, evitando uso de la red y distribuyendo las responsabilidades entre los distintos agentes de la forma más útil.

- (2,00p) Añadir un botón en la página llamado “Purgar”. Este botón ejecutará un borrado de las categorías que no contengan ninguna persona.
 - Se pide añadir lo necesario en HTML/JS/PHP/DAO, creando funciones, métodos y ficheros que sean adecuados.
 - Idea que puede utilizarse para la implementación: el servidor puede devolver un array de ids de categorías que han sido eliminadas. En PHP `json_encode($arrayDeNumeros)` funciona bien y desde JS se recibe un array de números que puede procesarse igual que un array de objetos.
- (2,00p) Actualmente, el servidor no valida los datos recibidos. Vamos a añadir una validación al teléfono.
(Si no te da tiempo, deja este ejercicio.)
 - `is_numeric($variable)` es una función PHP que permite comprobar si la `$variable` contiene un número o un string que sean solo números.
 - Consideraremos que el teléfono NO es válido si `is_numeric($telefono)` devuelve false.
 - Implementar, en el servidor, en el lugar adecuado, esta validación, tanto al crear una persona nueva como al modificar el teléfono de una persona existente.
 - El servidor devolverá al cliente un valor de control (un false, un string “ERR_VALIDACION”... Decídelo tú) que permita al cliente detectar esta situación e informar de ello al usuario de alguna manera (un `alert()` se considerará válido en ese sentido, en este ejercicio). El usuario deberá tener la oportunidad de remendar el error y volver a realizar la operación.

En general, para todo el examen: implementa primero lo básico y luego lo accesorio. Implementa primero los puntos que te resulten más rentables y luego los que menos. No te lées con detalles de control de errores si todavía no tienes la funcionalidad principal. Se pide explícitamente organizar el código según los modelos vistos en clase, u otros que sean objetivamente mejores en términos de calidad del código, simplicidad, genericidad y facilidad para su modificación. Dicho de otro modo, “que funcione” no es el objetivo principal de los ejercicios.