Práctica 4

Dotando de dinamismo a una web usando JavaScript

En esta práctica vamos a utilizar JavaScript (JS) para dotar de dinamismo a nuestra página en la parte del cliente. Se debe utilizar la P3 como base sobre la que incluir las modificaciones necesarias para cumplir con la especificación de este enunciado.

Gestión dinámica de un pedido

Crear una página "productos.html" con la funcionalidad necesaria para implementar un carrito de la compra.

- **Funcionalidad básica:**

El formulario de será similar a la siguiente imagen donde se muestra un ejemplo con 3 líneas añadidas al pedido:

Cantidad	Producto	Precio	Subtotal
1	Producto 1	100	100
2	Producto 2	200	400
4	Producto 3	300	1200

Total: 1700€

Cada vez que el usuario pulse sobre el botón "Añadir" un producto al carrito, se añadirá una nueva línea usando la información de entrada. En este caso, se utilizan como entrada los siguientes componentes: un input para la cantidad, un select para los productos y un botón para añadir.

Además, se gestionará una tabla auxiliar donde registrar las líneas que se vayan añadiendo. Es decir, en el carrito no deberán aparecer productos repetidos. En su lugar, se añadirá a cada línea del carrito dos campos "Cantidad" y "Subtotal". "Cantidad" mostrará, por cada producto añadido al menos una vez, cuantas unidades se desean comprar. "Subtotal" contendrá el precio del producto por la cantidad. También se deberá gestionar el precio total del pedido (suma de los subtotales), que se irá actualizado al añadir nuevas líneas. Tened en cuenta que los productos deben tener precios diferentes.

- **Gestión de IVA:**

Se deberá añadir al carrito la posibilidad de seleccionar el tipo de IVA aplicable al pedido usando un select (o similar). En las opciones del select se registrará el tipo de IVA a aplicar, estando una de ellas seleccionadas por defecto. Al cambiar el valor del select se deberá recalcular el carrito según la nueva elección. Se deberá mostrar en el carrito el IVA de cada producto (o línea de producto) y junto al total del carrito el importe del IVA del pedido.

Además, se deberán modificar las propiedades CSS dinámicamente para que, según el tipo de IVA seleccionado, alguno de los colores del carrito cambie (ej, color de fondo de las celdas que contienen información del IVA, color de fondo de la tabla…). Ojo, se valorará que este cambio de estilo se consiga sin rehacer la tabla, es decir, cambiando solo las propiedades CSS.

Persistencia en el navegador (HTML Web Storage o Cookies)

Se deberá implementar una estrategia de persistencia del estado del carrito. Es decir, la información del carrito deberá guardarse, y aunque el usuario cierre la página del navegador volverá a cargarse cuando el usuario la vuelva a abrir. Además, incluir un botón "Limpiar carrito" que al pulsarlo elimine toda la información de los artículos añadidos hasta el momento.

Validación de formularios con JS

En esta parte se piden varios ejercicios relacionados con la validación de formularios. Recordad que cuando no se cumpla la validación se deberá proporcionar al usuario información sobre el motivo de los campos erróneos.

Formulario de contacto

Editar la página "contacto.html" para que únicamente se permita enviar mensajes de contacto a usuarios que cumplan estas dos condiciones:

- **Fecha de nacimiento:** solo se permitirá el registro a usuarios mayores de 18 años. En caso contrario se informará del error.
- **Mail:** el mail debe pertenecer a la UMH finalizando en "umh.es".

Formulario de registro

Implementar una página de "registro.html" con los siguientes campos: nombre, apellidos, DNI, contraseña, repetir contraseña. Al pulsar en el botón validar, se deberá enviar la información de registro a mquesada@umh.es. Además, se deberá incluir la validación de los datos cumpliendo con la siguiente especificación:

- **DNI:** incluir un nuevo campo DNI cuyo formato debe coincidir con ocho números más una letra. Además, se deberá comprobar que la letra es válida; para ello se podrá reutilizar parte del código JS del siguiente enlace: https://tinyurl.com/v49mmvv.
- **Contraseña:** se deberá comprobar que las contraseñas introducidas en los dos campos son la misma. En otro caso se informará del error.

Información de errores alineada con HTML5

En este apartado se valorará que la información sobre los errores detectados quede lo más integrada posible en la estrategia ofrecida por HTML5; por ejemplo, cambiando el estilo de aquellos campos erróneos y mostrando información descriptiva sobre ellos.

PD: si los mensajes de error se muestran usando alerts o similar no se obtendrá ningún punto en este apartado.

Gestión del aviso de Cookies

Leer el siguiente tutorial de A. Fenollosa sobre cómo dotar de dinamismo al panel de cookies oculto que introdujimos en la P3:

- https://programadorwebvalencia.com/Javascript-ejemplo-aviso-o-cartel-de-cookie

Reutilizar solo aquellas partes que sean necesarias para conseguir la siguiente funcionalidad: por defecto el cartel deberá aparecer en la página e informar al usuario del uso de cookies. Dicho cartel tendrá dos botones "Consentir" o "No consentir".

- Si el usuario pincha "Consentir", se deberá ocultar el cartel y guardar en una cookie esta decisión para no volver a molestar al usuario con el cartel en un futuro (que aparecerá oculto).

- Si el usuario pincha en "No consentir" se deberá informar al usuario que la funcionalidad sin cookies todavía no está disponible y redirigirlo a la página principal de la UMH.

Validación de Cuenta y envío de correos con API

Incluir las modificaciones necesarias en "productos.html para incluir un botón "Realizar pedido" que, al ser pulsado, solicite al usuario el número de cuenta. Una vez introducido este número de cuenta se deberá validar que es correcto siguiendo la siguiente especificación:

- **Número de cuenta:** validar que el número de cuenta bancaria que se ha introducido es válido reutilizando parte del código JS del siguiente enlace https://tinyurl.com/shumy7d.

Si el número de cuenta es válido, se deberá enviar la información del pedido. Para ello se deberán utilizar las opciones de un API externa como EmailJS que permite enviar correos electrónicos directamente desde JS (https://www.emailjs.com/).

Tened en cuenta que para configurar el API de EmailJS debéis utilizar vuestro correo que será desde el que se enviarán los correos de vuestra página web. En la **memoria** se deberá demostrar que el funcionamiento es correcto, es decir que se ha enviado y recibido el mail correctamente.

Animación y programa oculto

En la página "podcast.html" crear una capa que se desplace aleatoriamente por la pantalla y desaparezca en un tiempo de entre 5-10 segundos. Dicha capa informará de la existencia de un programa de podcast oculto (ver p.4 de P3) que será visible solo si: (1) el usuario está utilizando un navegador concreto, y (2) se pincha la imagen antes de que desaparezca.

Configurar como variables globales de JS el navegador de la oferta y el tiempo del banner.

Video-Memoria

Una vez finalizada la práctica, crear una video-memoria (duración 6-8 minutos) donde se explique brevemente el trabajo realizado y decisiones que habéis tomado para modelar cada solución. Ambos miembros del equipo deberán contribuir a la explicación aportada. Las explicaciones deberán realizarse sobre el código implementado, y durante las mismas

se deberá mostrar que los ejercicios funcionan. Finalmente, incluir una breve valoración personal de la práctica.

Para la realización de la memoria, podéis usar herramientas como la extensión del navegador Google Chrome Screencastify (https://www.screencastify.com/).

Normas de entrega

La práctica se realizará **en grupos de dos alumnos**. La **fecha de entrega** de la tarea se comunicará en clase y estará visible en el Campus Virtual. Revisar las condiciones de Opción Continua y Opción Final en la presentación de la asignatura.

La práctica deberá entregarse a través de la tarea disponible en el Campus Virtual (Moodle) de la asignatura. Se deberá adjuntar a la tarea **un único fichero** en formato **ZIP** con todos los **ficheros generados.** En todos los ficheros deben aparecer, en un comentario, los nombres y apellidos de los dos miembros del grupo.

- Si el tamaño del ZIP supera los 20MB, seguir el tutorial "Tutorial subida de tareas en varios ficheros de 20MB" para dividirlo y subir a la tarea las diferentes partes.

Calificación y evaluación

Esta práctica tendrá un peso del **25% de la nota de prácticas de la asignatura**. Además, se deberá tener en cuenta lo siguiente:

Apartado	Puntuación máxima (en puntos)	¿Requisito mínimo?
Funcionalidad básica	**2**	**Sí**
Gestión de IVA	0,5	No
Persistencia en el navegador	1	No
Formulario de contacto	0,75	**Sí, al menos 1**
Formulario de registro	0,75	
Información de errores	0,5	No
Gestión del aviso de Cookies	1	No
Validación de cuenta	0,5	No
Envío de correos con API	1,5	No
Animación y programa oculto	**1**	**Sí**
Video-Memoria	**0,5**	**Sí**

Para que la tarea sea corregida, se deberán cubrir y desarrollar los **requisitos mínimos** descritos en la tabla anterior. Además, se podrá requerir la **defensa de la práctica (obligatoria en la Opción Final)** donde el alumno/a deberá hacer modificaciones y responder a las preguntas planteadas sobre la práctica. No realizar o superar la defensa supondrá una calificación de 0 y sus respuestas podrán influir en la nota final.