

Auxiliar 2

HTML5 y CSS

Prof: José Urzua

Auxiliar: Pablo Pizarro, Gabriel Iturra

Deployment de tareas

Para ejecutar el deployment dentro del servidor anakena, debemos dirgirnos a la sección preguntas frecuentes de la pagina del dcc subsección página web '[¿Cómo pongo una página web en mi cuenta?](#)'. Los pasos son los siguientes pasos:

- Nos conectamos al servidor anakena mediante ssh:
`mycomputer:~$ ssh user@anakena.dcc.uchile.cl`
- Una vez dentro del servidor debemos crear un directorio llamado 'public_anakena' (no puede tener otro nombre).
`user@anakena.dcc.uchile.cl:~$ mkdir public_anakena`
- Luego, debemos setear los permisos de listar archivos en nuestra cuenta del servidor.
`user@anakena.dcc.uchile.cl:~$ chmod 711 ~user`
- Y finalmente seteamos los permisos de escritura y lecturar del directorio 'public_anakena' eso debe ser para TODOS los archivos dentro del directorio.
`user@anakena.dcc.uchile.cl:~$ chmod -R 755 public_anakena`
- Para ver si todo funciona, deben dirigirse a la url `anakena.dcc.uchile.cl/~useranakena/` (con algún archivo previamente creado que tenga un nombre `index.html` o `index.php` de la carpeta `public_anakena`).
- Sí tienen un error de permisos es porque la url no esta bien escrita o bien no setearon los permisos.

Pregunta

Debe **validar** el formulario hecho en la primera versión de *Hackbox*. Para ello, considere las mismas restricciones del enunciado anterior:

- El **nombre del usuario** debe tener un mínimo de 10 caracteres y un máximo de 100. No puede tener números o símbolos que no sean a-z/A-Z. (Campo obligatorio)
- El **archivo** no debe estar vacío. (Campo obligatorio)
- El **selector de privacidad** no debe estar vacío. (Campo obligatorio)
- La **contraseña** debe tener un límite de 10 caracteres. Comprobar además que la contraseña no sean: *password*, *1234*, *pass*, *user*, *hackbox*. Para ello implemente listas con JS. (Campo opcional)
- El **autodestruir**, si existe, debe ser un número entre 1 y 3153600 (número de segundos en un año). (Campo opcional)
- El **comentario** debe tener un máximo de 1000 caracteres. (Campo opcional)

Además, para mejorar la experiencia de usuario, considere añadir el siguiente botón a su aplicación:

- **Botón reset:** Limpia todo el formulario y oculta el mensaje de error. Esto es sin recargar la página.

Al encontrar cualquier error en el formulario. Debe alertar al usuario de TODOS los errores encontrados. Si tiene dos o más, se debe mostrar un mensaje del estilo “*Su formulario falló en: Nombre, Contraseña, ... etc*”. Para ello puede usar un **alert()** o bien, como lo haremos en la auxiliar (sólo mostrando un mensaje de error genérico) usaremos un elemento `<div>` que mostraremos si el usuario cometió algún tipo de error.

Para añadir la funcionalidad del evento, considere añadir `onsubmit="myValFunction()";`^{en javascript}.

Ejercicio

Implementar toda la página web, y subir al servidor. Si no sube a anakena sufrirá un descuento en su nota final de 5 décimas. Responder además las siguientes preguntas de manera breve. Puede hacer un LaTeX o un documento tipo word. Enviar como PDF adjunto a los archivos de su tarea.

- Por qué es importante la validación “doble” usando javascript en vez de lo que ofrece HTML. ¿Es segura?
- En clase auxiliar se mencionó la existencia de librerías de alto nivel que permiten manipular el DOM de manera sencilla. Investigue, nombre y describa 5 de las librerías javascript más usadas por la industria para el desarrollo web (por ejemplo, **jQuery**). Si ud. tuviese que implementar su propia página web, ¿Qué librería utilizaría?