11086 - Programación en Ambiente Web - UNLU
Primer Parcial 2020

Nombre: German Fernandez
Legajo: 127390
Email: germanf.08@hotmail.com

Imagine una aplicación web "portal de noticias" y responda las siguientes consignas:

1. ¿Por qué las sesiones pueden guardar mucha más información que las cookies? ¿Qué almacenaría para esta app en cookies y/o sesiones?
Las sesiones se guardan del lado del servidor, a diferencia de las cookies que se almacenan en el navegador del cliente. El almacenamiento de cookies se encuentra limitado por la capacidad del navegador para las mismas, por lo tanto, debido a que el servidor tiene mucha más capacidad, las sesiones permiten alojar mucha más información, guardando únicamente una cookie en el cliente como identificador para asociar la sesión.

Para esta aplicación se almacenaría la siguiente información:
Para este caso se podrían utilizar sesiones para almacenar toda la información de cada cliente, por lo tanto, se guardaría una cookie en el navegador del cliente el cual contiene el ID de la sesión asociada como se explicó anteriormente.
Luego en cada sesión podríamos almacenar, por ejemplo, el nombre y password del usuario, datos acerca de la cantidad de veces q visitó el sitio, el idioma de preferencia del usuario del sitio, categorías de noticias favoritas, notas visualizadas para saber que le interesa al usuario y todo tipo de información similar que podría utilizarse un acceso posterior.

2. ¿Qué ventajas ofrece el uso de Virtualhost en el contexto de servidores Web (en gral y en particular para esta app)?
La ventaja del uso de Virtualhost es que nos permite asignar varios sitios web y sus nombres de dominio a una misma dirección IP, por lo tanto, los estaríamos ejecutando en el mismo servidor como si estuviesen en distintos. Esto nos resulta más económico ya que no tendríamos que pagar por un servicio de alojamiento a otra empresa.
Para nuestra app pueden considerarse algunas ventajas, entre ellas una es que nos permite navegar entre los distintos dominios y explorar de manera offline como si lo hiciéramos en internet y así testear nuestro sitio web. Otra ventaja es el poder personalizar los servidores virtuales, podríamos configurarlos según la necesidad del cliente. Se puede decir que es escalabre ya que, al modificar las capacidades según la necesidad, no estaríamos comprometiendo el funcionamiento del servidor. Además, puede tolerar fallos, en el sentido que permite crear backup de información para recuperarla por si ocurre algún problema de perdida de datos en nuestro servidor virtual. Incluso en la pérdida del servicio podemos levantar rápidamente otro virtualhost.

3. Defina con sus palabras la diferencia principal entre contenido estático y dinámico.
El contenido estático se refiere a aquel que no cambia en una página web, lo diseñamos para que quede fijo y no necesite modificaciones frecuentes, por ejemplo, el menú del sitio o las imágenes principales, como el logo de nuestro portal de noticias.
Por otro lado, el contenido dinámico va cambiando con el tiempo, como por ejemplo en el portal de

noticias, cada noticia del diario corresponde al contenido dinámico ya q se van actualizando las noticias más recientes, ya sea el título de la misma, el detalle o incluso el enlace a la página interior de la noticia misma.

4. ¿Cómo aplicaría el modelo MVC para el diseño de esta app?. No necesita escribir código alguno, sino argumentar conceptualmente como separaría la lógica de la app en estos tres elementos.

En primer lugar, voy a definir todas las interfaces de usuario en las Vistas. Aquí se guardarán todos los distintos escenarios, los cuales voy a mostrar al usuario para que interactúe y pueda llevar a cabo sus solicitudes, ya sea la página inicial del diario, la devolución de algún error, la lectura de una noticia, un formulario de registro al sitio web, etc.

En la siguiente capa, el Controlador, voy a definir todo el código que se utilice como intermediario y controle el flujo entre el modelo y la vista. Va a ser quien reciba la información de las vistas, por ejemplo, cuando se solicite ingresar a una noticia en particular del diario, el controlador va recibir esta información de la vista, va a gestionar la misma, es decir, verificar el evento que se realizó, y en base a eso va a enviar la petición al modelo correspondiente, recibiendo luego la información necesaria para actualizar la vista correspondiente al usuario que solicitó la noticia.

Por último, la capa del Modelo se encarga de todo el manejo de los datos que se encuentren en el sistema, ya sea el acceso a la información. Si un usuario solicita ver una determinada sección de noticias, el modelo se va a encargar de buscarla en la base de datos, y enviárselas a través del controlador a una vista. También se va a encargar de la actualización de los datos, como en este caso puede ser con respecto al registro de los usuarios.

5. a) ¿Por qué es posible afirmar que PDO mejora la seguridad en la capa de base de datos de una app PHP?

PDO te permite proporcionar una capa de abstracción de acceso a los datos, esto significa que permite realizar consultas a la base de datos mediante un conjunto de instrucciones o sentencias preparadas, las cuales son soportadas en muchos tipos de bases de datos, y por lo tanto nos permite utilizarlo en tipos de bases de datos distintos. Esto nos ayuda a mejorar la seguridad en la capa de base de datos de una app PHP ya que nos protege de que alguien pueda inyectar código malicioso en nuestras consultas, y acceder a datos guardados o incluso modificarlos.

b) ¿Qué otras cuestiones debemos tener en cuenta en la capa de base de datos en el sentido de la seguridad?

Además, para garantizar la seguridad en la capa de base de datos deberíamos tener en cuenta ciertas cuestiones. Por ejemplo, deberíamos identificar la información sensible que ingresa el usuario (Ej. contraseñas, tarjetas de crédito) y cifrarla mediante algún algoritmo fuerte, así mientras se encuentre almacenada sea ilegible para cualquier persona que llegue a ella sin autorización.
Otra cuestión podría ser monitorizar la actividad en la base de datos, llevar un control en el momento de los accesos a la misma para detectar acciones sospechosas en tiempo real, por ejemplo, si en un corto plazo de tiempo se registran demasiados intentos de accesos no autorizados a la información, se podría aplicar un bloqueo.

6. La app muestra signos de "envejecimiento" en cuanto al diseño, tanto usuarios finales como redactores del portal lo informan a diario. ¿Qué ideas se le ocurren al respecto?

Como diseñadores web debemos tratar que nuestro sitio web se mantenga actualizado para garantizar una mejor usabilidad, y por lo tanto que los usuarios y redactores del portal que van a

interactuar con la app, lo hagan de manera sencilla, intuitiva y agradable.
Si estos últimos nos informan seguido que el diseño se encuentra de alguna manera dicho "envejecido", podríamos modificar y actualizar mediante una plantilla (template) más actual volviéndolo estándar, mejorando así la usabilidad en los usuarios.
Un ejemplo de esto es el framework bootstrap que contiene distintas plantillas de diseño basados en HTML y CSS, las cuales podemos elegir para actualizar el diseño del sitio en el caso de nuestra app de noticias.

7. Se le informa al equipo de desarrollo que las nuevas funcionalidades están repercutiendo negativamente en la performance de esta app web en el ambiente productivo, no así en el ambiente de testing (QA). DevOps informa que existe últimamente mucha carga a nivel de bases de datos. ¿Qué se le ocurre hacer en su rol de Desarrollador Web?
En mi rol de Desarrollador Web yo debo diseñar una app que permita proporcionar toda información y además ser funcional, por lo tanto, a la hora de la recuperación de información en la base de datos, debo tener en cuenta la relación funcionalidad y velocidad. Esto llevado a nuestro ejemplo del sitio de noticias podría implementarse, estableciendo un límite en la cantidad de noticias que se devuelven como respuesta a la petición de búsqueda de un usuario, de esta forma no sobrecargaríamos la base de datos consultando todas las noticias, sino que le entregaríamos al usuario solo un conjunto limitado de noticias más relevantes o actuales sobre lo cual esta interesado.

8. Imagine ahora que el "portal de noticias" debe considerar tener un "paywall" (ciertos contenidos se vuelven pagos) y por ende almacenará tarjetas de débito / crédito de los clientes.
a. ¿Cuáles son las implicancias de seguridad de esta nueva funcionalidad?
El almacenamiento de tarjetas de débito/crédito de los clientes implica un cuidado especial en dicha información, ya que, si no lo haríamos, un atacante podría obtener estos datos, con las malas consecuencias que eso podría tener.
Por eso, para este ejemplo, al desarrollar el portal de noticias debería tener ciertas consideraciones para almacenar dicha información.
En primer lugar, debería fomentar el uso de contraseñas fuertes y que se cambien con regularidad ya que sino cualquiera que la consiga podría entrar y obtener los datos propios del usuario.
Esta información sensible por ejemplo sobre las tarjetas de crédito/debito almacenadas en el portal, deberían ser encriptadas para ser guardadas, y se podría configurar el servidor web para usar HTTPS el cual encripta los datos que se envían entre el cliente y el servidor y así asegura que la información en tránsito sea menos disponible a los atacantes.
Otra medida podría ser limitar la información, enseñando solo lo necesario, por ejemplo, mostrarle al usuario solo una parte de los números de la tarjeta de crédito, y así de esta forma el usuario pueda identificarla, y que no sea suficiente para ser copiada por un atacante.

b. ¿Cómo implementaría algún límite sobre la cantidad de noticias que puede ver un usuario que no paga, e.g. puede ver sólo 10 artículos por mes calendario?
Yo implementaría una entrada en la sesión que generé del cliente, que lleve a cabo un contador de visitas de los artículos de noticias a los que el usuario ingresa. Además del contador, voy a llevar un registro para saber si el usuario es o no es pago, de esta forma si no es pago y alcanza el límite de visitas a noticias, le voy a restringir el acceso. A principio de cada mes tengo que actualizar este valor para que el usuario vuelva a tener su contador de visitas nuevamente en 0.

9. Se requiere implementar un buscador de noticias dentro de esta app. Explique qué responsabilidades tiene cada capa de la aplicación en la resolución de la búsqueda. ¿Qué método HTTP le parece el más adecuado para implementar esto? ¿Qué problemas observa?

En el caso de la vista se podrían observar dos responsabilidades, en primer lugar, la vista para la introducción de la búsqueda del usuario. En segundo lugar, se encuentra el resultado de la búsqueda realizada.

Para el caso del modelo, se encargará de buscar los resultados de la búsqueda, y devolverlos para que el usuario pueda visualizarlos.

Y el controlador, se va a comunicar con el modelo para que devuelva las noticias que coincidan con la búsqueda que el cliente realizo, y las devolverá a través de la vista correspondiente.

Esto puede generar sobrecarga en la base de datos por obtener demasiadas noticias relacionadas, por lo tanto, en el modelo podríamos limitar la cantidad de consultas a realizar para devolver.

El método GET de HTTP sería el más adecuado ya que se utiliza para realizar peticiones de información y obtenerlas del servidor web. El problema que observo para este caso es el de la inyección, ya que un atacante podría inyectarnos código en la petición al servidor, accediendo a información almacenada.

10. Se requiere que la experiencia del sitio sea uniforme en versiones de Chrome/Firefox/IE de hasta 3 años atrás. ¿Cómo puede cumplir con dicho requisito? ¿Qué estrategias adoptaría desde el punto de vista del diseño e implementación?

Para permitir que la experiencia del sitio sea uniforme en distintas versiones de hasta 3 años atrás ya sea de Chorme, Firefox o Internet Explorer debemos lograr que nuestro sitio sea cross-browser. Esto es, que se comporte de igual forma en cualquier navegador que lo esté interpretando. Para esto existen algunas técnicas o soluciones que podremos implementar:

Una estrategia es cargar una hoja de estilo diferente según la versión utilizada del navegador, por ejemplo, en Internet Explorer existen varios problemas de compatibilidad entre versiones por lo tanto podríamos aplicar una condición que según la versión utilizada de IE, se cargara la hoja de estilo correspondiente.

Otra estrategia es la validación de nuestro código, tanto HTML como CSS. Es recomendable seguir los estándares que aconseja el W3C, esto se puede verificar mediante los validadores oficiales del mismo. Si nuestro sitio esta validado por el W3C va a ser mucho más compatible para la mayoría de los navegadores.

Por ultimo una técnica mas es la de Reset CSS. Los navegadores suelen emplear hojas de estilos por defecto propios, esto puede generar múltiples diferencias visuales a la hora de mostrar la página. Para solucionar esto se definen valores por defecto de algunas propiedades para que el navegador las tome como referencia, sobrescribiendo las propias del navegador.