Apellido y nombre: Garrós Gastón
Legajo: 118286
Materia: 11086 – Programación en ambiente web

11086 - Programación en Ambiente Web – UNLU
Primer Parcial 2020

Entrega: archivo PDF por mail a la dirección paw@unlu.edu.ar antes del viernes 1 de mayo a las 23.59.59, es decir cuenta con 36hs para realizarlo. Además del envío por mail del PDF se solicita suban también dicho archivo PDF a un repositorio propio y envíen dicha dirección en mail aparte o por whatsapp/telegram para garantizar recepción a tiempo utilizando dos vías independientes y con timestamp.
Metodología: el examen es individual y si bien puede utilizar libros e Internet, el examen debe ser autocontenido y con respuestas «propias», no con URLs hacia material externo. Puede incluir esquemas o lo que considere necesario para ilustrar sus respuestas.
Nota Importante: En ninguno de los 10 puntos se solicita código. Responda cada punto considerando el escenario más real/auténtico que pueda imaginar y explicite cada una de sus asunciones.

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 cookies son una pequeña porción de datos que se almacenan del lado cliente por lo tanto no tiene el suficiente espacio para almacenar demasiada información comparado con las sesiones las cuales corren del lado Server y pueden usar los recursos de este para almacenar información con respectos a la app y luego enviarlas al cliente browser o del tipo correspondiente.

   En nuestro portal de noticias un uso para las cookies podría ser para setear la clases de noticias que le gusta recibir al usuario y si la app lo permite personalizar la apariencia de la web, el lenguaje que desea para las noticias o si el portal tiene algunas noticias de pago individual y no una suscripciones completa podría crear un carrito de compras para artículos y para guardar los item de ese carrito por un tiempo relativamente largo, permitiendo salir de la app y cuando vuelva a entrar el usuario los item seguirán cargados. También podría almacenar datos correspondientes al usuario que carga contenido al portal, como por ejemplo un campo de firma en los articulo que este redacta, etc.
   Las sesiones en esta app la usaría para la parte de registro de usuario y para el momento del ingreso de las tarjetas al momento de realizar los pagos, también para almacenar si el usuario tiene varias tarjetas y dejara una como default, esa información la guardaría del lado servidor por seguridad, ya que si se almacenara en la cookie, si alguien roba esa cookie estaría robando la información de esa tarjeta. Con respecto a la seguridad del login tendría que ser una conexión https y utilizar hash con salt en el password además de algunas otras políticas de seguridad.

2. ¿Qué ventajas ofrece el uso de Virtualhost en el contexto de servidores Web (en gral y en particular para esta app)?

   La ventaja que ofrece el uso de Virtualhost en los servidores web es que puede proporcionar diferentes servicios para diferentes dominios tanto ip, nombres y puertos, sin la necesidad de

contratar una ip ni un almacenamiento especifico para cada uno de estos servicios. Esto lo que permite es que se pueda subdividir los directorios de ese servidor otorgando una carpeta raiz para cada uno de los virtualhost (documentRoot) osea, podemos tener N sitios en un mismo servidor. Para que esto funcione le debe llegar trafico a la ip del servidor y según el dominio que venga en la petición esta macheara con los dominios, esto nos configurados en el servidor y cumplirá las reglas fijadas en este. Esto permite separar por ejemplo los log en diferentes directorios siendo el caso que haya varios virtualhost para una misma app.

En el caso de nuestra aplicación se puede considerar usar virtualhost para separar el registro de usuarios de nuestro sitio para aplicarle otras políticas de seguridad y tener mayor control sobre el registro, sacando del mismo ámbito a los usuarios que postean o leen noticias. Esto se puede configurar para que de pocos permisos y denegar ejecución de comandos en estos directorios.
Otro ejemplo podría ser separar la aplicación en otro dominio que pertenece a un blog del sitio pero se le quiere asignar otro tipo de recursos menores.
Otro ejemplo podría ser separar las claves de las aplicaciones en otro dominio y que este las sirvas de forma segura, esto serviría si con un ataque se roban el código de la app, no podrían acceder por ejemplo a la base de datos ya que en ningun lado del código estaría esa clave para la base de datos, esta se obtiene en la aplicación por el servicio virtualizado.

3. Defina con sus palabras la diferencia principal entre contenido estático y dinámico.

   El contenido dinámico de una pagina es el contenido que puede ir cambiando con el uso de la pagina por el usuario, osea, el usuario realiza una acción sobre el contenido de la pagina y esto provoca una respuesta por el servidor alterando el contenido de la pagina actual o sirviendo otra pagina nueva, o tal vez solo guardando información del lado servidor.
   Para una paginas dinámica se requiere de algún lenguaje de programación del lado servidor el cual según las peticiones del cliente procesara la información y creara una respuesta a la petición, generando un archivo html con las vista y la información necesaria. En nuestra app podría ser un ejemplo de esto la sección de agregar una noticia o buscar una noticia en la web.
   Por el contrario, las paginas estáticas son aquellas en las que el contenido no cambia ante la acción del usuario en ellas. Son un archivo el cual no es necesario que sea servido por un servidor web como apache, el cual puede tener problemas y caerse, esas es una ventaja de estas paginas, es muy difícil que las paginas estáticas no estén en funcionamiento, pero dependiente a esto viene lo limitada que son.
   El contenido estático pueden ser los archivos CSS, las imágenes, etc.

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.

   View:

   En la sección vista colocaría todos los elementos relacionados a la visualización de los componentes del app, serian las paginas que van a renderizar el contenido que se le mostrará al usuario. En nuestra aplicación puede ser el código php que nos mostrará el contenido para realizar el login a nuestra pagina web o la vista que nos muestra las noticias del dia cuando iniciamos la pagina del sitio.

   Controller:

   En esta sección controller ser realizaría la conexión de las vistas con los modelos de la app web, esta clase seria la encargada de instanciar las clases del modelo y cargarlas con los datos que vienen del lado cliente en las vistas y enviarlos al sistema de bases de datos para que este

los almacene. También se encarga del sentido contrario donde el usuario pide datos de la base de datos y el controller lo localiza los datos en la base de datos, a través del driver de bases de datos y este controller lo devuelve a la vista.
También en esta sección dedicaría un controller especifico a para la conexión con la base de datos para abstraer el tipo de la base de datos con el modelo en si, por si la base de datos en un futuro cambiara o si hubiera varias BD de distintos tipos el controlador independiente de la base de datos las manejaría como una sola base de datos y este machearía a cual hacer las consultas.

Models:

El la sección de modelos irían todas la clases que hacen a la app y sus relaciones que hacen la lógica de negocio, por ejemplo las clases que se involucran con las noticias y atributos que esta lleva, junto con las relaciones que tendrán esas noticias con otros elementos de la app.
También otro ejemplo que debería entrar en esta sección son los usuarios y las relaciones de estos. Una clase persistencia también podría introducirse en aquí para levantar métodos correspondieses a consultas o nacionalización de la base de datos.
La sección del carrito o las suscripciones también deben tener sus modelos en esta sección y relacionarse con un usuario único y con las noticias.

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

PDO mejora la seguridad en la capa de bases de datos porque es una abstracción del driver y usa prepared statements los cuales evitan la inyección sql en la base de datos ya que se parametrizan los elementos en las queries y no pueden manipular el sql como con otros métodos como por ejemplo mysql connect que si bien puede programarse de forma seguro hay que validar y resolver varias cuestiones para que no produzca problemas de seguridad.

Prepare-→ Bind → execute

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

Algunas de otras cuestiones en la capa de bases de datos son los permisos que se le otorgan a los usuarios que deben tener acceso a la base de datos, a los desarrolladores que necesiten tener permisos, pueden necesitar tener varios permisos pero no de todas las tablas, por lo tanto se le puede otorgar permisos “user o full” pero de ciertas tablas, no de todas. Al igual que los administrativos tiene que tener roles y restricciones para que por errores no eliminen datos importantes, igualmente para esto se deben tener sistemas de logs y backup para poder recuperar datos.
Se tienen que validar todos los datos del lado servidor para evitar datos incorrectos o intentos de ataques. Aun usando PDO se tiene que validar los datos para los parámetros.
También se tiene que tener en cuenta el capturar los errores de la base de datos para no mostrar información no deseada al usuario en caso de algún error y que puede mostrar algunos datos que pongan en riesgo de la seguridad.
En todas las operaciones de carga de datos se deben controlar tanto como en las operaciones con archivos e include, ya que se pueden introducir código malicioso y secuencias de comando que se

pueden ejecutar hasta con permisos root en la base de datos, una practica es usar pdo y también invalidar los comando especiales y comandos.

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?

   Si la web recibe criticas de diseño intentaría realizar un nuevo diseño de la web si es una pagina muy antiguo o mejorar los aspectos actuales de la pagina web, teniendo en cuenta que la web sea responsiva, que los elementos estén alineados de forma adecuada para la visualización desde dispositivos móviles, usar la paleta de colores correspondientes, no agregar tanta información en la pagina que no sea de utilidad en el sitio, siempre mantener la menor cantidad de elementos para el sitio, demasiados elementos puede hacer que el usuario se pierda en la búsqueda de información y una demora de unos segundos en el sitio puede provocar la perdida de ese usuario en el sitio.
   También se deberían realizar chequeo de accesibilidad, teniendo en cuenta a usuario con discapacidades para otorgar servicios para estos usuario de una mejor manera. Para llevar a cabo todas estas nuevas mejoras o revisiones se debería modificar el ccs de la web, también se podría utilizar imágenes con mejores diseños usar técnicas para hacer notar las noticias mas relevantes, por ejemplo, mostrar imágenes con cierta edición que atraigan a los usuarios, estos podrían trabajo de diseñadores pero podrían mejorar notablemente el consumo de los usuarios.
   De ser necesario revisar temas de performance si hay síntomas de deterioro en cuanto a la carga y descarga del sitio.

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?

   Si las funcionalidades incorporadas no son de carácter critico se podría crear una vista con una pantalla informativa de mantenimiento temporal para informar al usuario y mientras tanto controlar los errores.
   Para controlar el mas desempeño buscaría error en el código en cuanto a la validación de los datos del usuario, el mal funcionamiento en la producción, y buenos resultados en el testing (QA) me hace sospechar de alguna inyección sql o algún intento de ataque a la base de datos y por eso puede estar sobrecargando la base de datos. Si este fuese el caso buscaría bien en todas las secciones donde se ingresan datos, controlaría que las sentencias de PDO estén correctas si se trabaja con ellas y sino trabajar con PDO para mas seguridad y salvar la inyección de código.
   Se debería revisar todas las inclusiones a archivos teniendo en cuenta que estos pueden provocar que alguien ejecute código en el lado servidor. Una buena practica podría ser buscar los logs de las nuevas funcionalidades para ver si se encuentra en ellos algo fuera de lo normal.
   Validar que el registro de usuario se haga de un modo seguro con https y sesiones y no a través de cookies o si lo hicieran de esta forman controlar bien los datos por que son mas sensibles y pueden robar datos de allí para intentar ataques de fuerza bruta.
   Otro posible caso que puede estar causando problemas de las nuevas funcionalidades es que le estén pidiendo peticiones masivas sobre esa funcionalidad y produce sobrecarga en la base de datos en cuanto al procesamiento, esto puede solucionarse con balanceadores de carga del lado

servidor. También es posible que haya datos que no se están cacheando y se están consultando continuamente con los datos de la base de datos.

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?

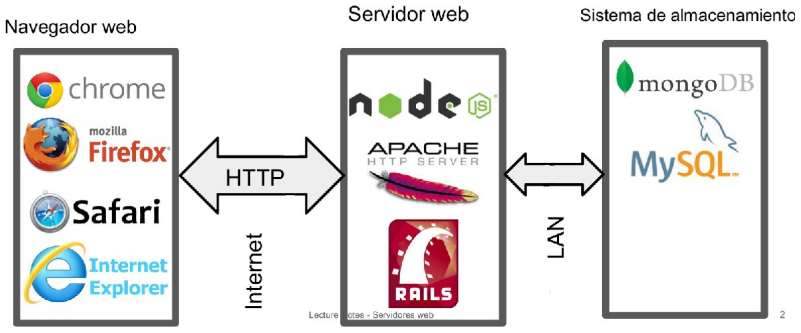
   Para realizar un paywall y poder tener accesos a las noticias pagando una membresia se debe poder registrar usuarios, la manera de hacer esto es a traves de sesiones para mayor seguridad en los pagos, con https y encriptando y hasheando las claves de los usuarios. Cuando el usuario se registra se le debe pedir una cuenta de correo electronico y un pasword con una cierta cantidad de carecteres que deben ser validarlos del lado cliente que cumpla con lo pedido y la confirmacion de dicho password, luego estas validaciones se deben hacer tambien del lado server. Dependiendo como sea la politica de usuarios, puede la app tener usuario pagos y usuarios con registro gratituos, en cualquiera de los dos casos se deben confirmar el registro con un mail.
   En caso de los usuarios pagos se tiene que tener mucho cuidado en el ingreso de la tarjeta de los usuarios que de ninguna forma se puede perder o puedan robar esa informacion tanto en la base de datos como en el envio desde el cliente al servidor, una posibilidad de las paginas de hoy en dia es delegar a un tercero el registro de usuarios solo para el registro confiando por ejemplo en facebook o google y también otra posibilidad es para la partes de pagos confiar en un tercero como mercadopago para los pago y recibir un tokens desde la Api para confirmar el pago y asegurara la confianza del cliente.
   En caso de no delegar las acciones de pago se debe controlar donde se almacenan las contraseñas de los usuarios, tanto de pagos como administradores, no se deben alojar en archivos a los cuales el usuario de algún tipo puede tener acceso, estas contraseñas de deben validar con las almacenadas en la BD para confirmar el acceso.

   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?

   Para mantener un control de los artículos que puede tener un usuario que no tienen una membresía en la app lo que se puede hacer es que el servidor utilice una cookie del cliente para tener un contador de accesos de ese cliente, esto seria, cuando el cliente pide un articulo en la cookie el contador aumenta en uno por cada articulo pedido hasta llegar al limite de diez. En la primer consulta del mes se le pude setear a la cookie una fecha de expiración la cual tendrá una valides de un mes, por lo tanto, la cookie mantendrá un registro de artículos pedidos por ese cliente durante el mes, pasada la fecha de expiración la cookie se descarta y volverá el servidor a pedir otra cookie por ese mes con un limite de diez artículos.
   Esto solo funciona del lado cliente, esta cookie podría ser editada y el control se perdería ya que el servidor no guarda datos sobre los usuarios sin membresía.

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?

Para realizar un buscador de noticias en la aplicación en las vistas se deben agregar todos los elementos de filtrado que el usuario necesite para encontrar una noticia como titulo, categoría, fecha, etc. Esto se debe maquetar de una forma que sea agradable para el usuario y la experiencia de este ante la búsqueda pueda ser la optima. Luego en la capa controller se deben realizar todos los métodos necesarios para que la vista le pueda pedir al modelo de la aplicación los datos que el usuario llego en la vista y luego devolver estos al usuario. Para realizar estas conexiones se tiene que tener los modelos correspondientes con su lógica para poder identificar artículos y poder conectarse con la base de datos y pedir los datos.
En una capa de abstracción mayor el cliente pide datos de búsquedas en su browser, estos datos son enviados al servidor por el método correspondiente y el servidor los procesa y valida haciendo la consulta en la base de datos



En este enfoque mas amplio dependiendo si es un usuario pago o no, el servidor podría almacenar datos sobre la búsqueda de esa sesión para crear un perfil de ese usuario y en próximas búsquedas o inicio sugerir articulos relacionados a sus búsquedas previas. El método por el que se envian los datos debería ser GET ya que al pasarlos de esta forma el usuario podría compartir esa búsqueda con diferentes usuarios.
Los problemas que puedo observar en la búsqueda de datos es un usuario malicioso puede inyectar código sql en las variables de las búsqueda, por lo tanto deben validarse del lado servidor.

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?

Una metodología para que la experiencia del sito sea uniforme para varios navegadores se puede cumplir siguiendo las normas estándar que se plantean para html5, css javascrip. Esta metodología se la puede conocer como cross browsinng, para la misma representación en varios browser. Sin embargo en algunos casos puede suceder que alguna versión del browser tenga conflictos se pueden intentar buscar algunas técnicas para encontrar los problemas que se involucran en el conflicto, pensar es si es necesario dar soporte a una versión muy antiguo, en este caso hablamos en un limite de 3 años por lo tanto no se debería tomar esa opción. En algún caso muy extraño se podría implementar una solución única para ese tipo de browser, por lo tanto tendría dos implementaciones una para ese browser especifico y otro para el resto de los browsers englobados en esa solución normalizadas por alguna entidad reguladora como w3schools.