GUIÓN DE LA ACTIVIDAD PRACTIVA AEV4:

Título actividad

AEV4 - Seguridad con Symfony

Objetivos

- Demostrar el conocimiento adquirido durante el tema 8 sobre la seguridad del Framework Symfony.

Recursos generales

Presentaciones y videos del Tema 8, del tema 7 y del tema 3

Actividad / Enunciado

1. Vamos a crear una pequeña aplicación que servirá para registrarse en un restaurante y poder hacer reservas. El restaurante se llama **restaurante DWES.**
2. Crear una aplicación, con el framework de Symfony, en la que únicamente instalaremos los bundles que sean necesarios. No se puede instalar una aplicación completa (webapp):
3. Todas las pantallas, emails y mensajes deben mostrarse completamente en **castellano**. No puede haber palabras en inglés u otro idioma.
4. En todas las pantallas tendremos un link para poder volver a la ruta raíz ➔ "/"
    - **Raíz "/"**
      Acceso para todos los usuarios.
      En esta ruta mostraremos una breve descripción del restaurante, te inventas el párrafo de la descripción.
      Además, tendremos accesos directos a las diferentes secciones de la aplicación. (DEBERAS TENER EN CUENTA QUE LINK PUEDES MOSTRAR SEGÚN ESTES O NO LOGUEADO).
        - **Registro**
        - **Logueo**
        - **Administración de usuarios**
        - **Administración de reservas**
        - **Perfil de Usuario**
        - **Reservar**
        - **Carta**
        - **Encuéntranos**
        - **Quienes somos**

- **Registro "/registro"**
  Esta ruta es accesible por todo el mundo y nos muestra un formulario de registro.
  Nos tendremos que registrar mediante el email y nos validaremos al pinchar el link que se recibe en el correo.
  En el registro hay que introducir los siguientes datos del usuario:
    - Nombre
    - Apellidos
    - Email
    - Teléfono.
    - Contraseña

  (RECUERDA debes modificar todos los mensajes, plantillas, etc… que configura el framework por defecto)
  Al registrarnos deberemos acceder a la pantalla raíz.

- **Logueo "/login" o Desconectar "logout"**
  Acceso para todos los usuarios.
  En la pantalla de la raíz, si no estamos logueados deberemos ver en este punto el texto **"Logueo"** e ir a la ruta "/login", por el contrario, si estamos logueados con un usuario veremos el texto **"Desconectar"** e ir a la ruta /logout.
    - Logueo "/login"
      Mostrará una pantalla en la que podremos loguearnos en la aplicación mediante el email y la contraseña.
    - Desconectar "logout"
      Nos deslogueará de la aplicación y nos volverá a la pantalla de inicio.
- **Administración de usuarios "/user"**
  Esta pantalla únicamente tendrá acceso los usuarios que sean administradores y tengan el role: ROLE_ADMIN.
  En esta pantalla saldrá una lista de todos los usuarios en la que veremos en formato tabla con los siguientes datos:
    - Nombre
    - Apellidos
    - Email
    - Teléfono
    - Roles
    - Editar (Dentro de la edición nos permite eliminarlo, es el único sitio donde se puede)
- **Perfil de usuario "/user/profile"**
  Esta pantalla únicamente tendrá acceso el usuario logueado, tenga el rol que tenga. De forma que accederá únicamente a su perfil.
  En esta pantalla mostraremos los datos del usuario:
    - Nombre
    - Apellidos
    - Email
    - Teléfono
    - Roles
    - Editar (En esta pantalla no se puede eliminar al usuario)

En la pantalla de edición será prácticamente igual a la pantalla de edición del perfil de administración, salvo que no se puede eliminar al usuario.
Hay que añadir un método para poder actualizar la contraseña y que esta misma se codifique como en el método de registro.

- **Nueva Reserva "/reservas/new"**
  Esta pantalla puede acceder cualquier usuario que este logueado y tenga el rol ROLE_USER.
  Para insertar una reserva deberemos utilizar el usuario con el que estoy logueado. No pudiendo elegir otro usuario.
  Se elegirá el número de comensales y la fecha y hora de la reserva.
  Al introducir la reserva nos mostrará los detalles de esta: "/reservas/{id}"
  No se puede editar la reserva, pero sí eliminarla.
  Eliminar la reserva **"/reservas/{id}"** solamente puede hacerlo el usuario que ha realizado la reserva o el administrador.
- **Administración de reservas "/reservas"**
  Esta pantalla únicamente tendrá acceso los usuarios que sean administradores y tengan el role: ROLE_ADMIN.
  En esta pantalla nos mostrará la lista con todas las reservas, de todos los usuarios, donde podremos visualizarlas de forma individual, editarlas o borrarlas.
- **Carta "/carta"**
  Esta pantalla es de acceso público y debe mostrar un breve texto descriptivo del menú y luego un pequeño listado de productos del menú. No es necesario hacer nada con la BB.DD.
- **Encuéntranos "/location"**
  Esta pantalla es de acceso público.
  Debe mostrar un link a Google Maps con la dirección, así como un párrafo indicando la dirección y el horario.
- **Quienes somos "/about"**
  Esta pantalla es de acceso público.
  En ella tendremos una breve descripción de la historia del restaurante, te inventas el texto para que quede chulo.

5. La actividad debe crearse en un contendor de docker exclusivo para esta actividad. Y el contenedor debe contener además la BB.DD. correspondiente a esta actividad.
6. No es necesario trabajar con Git.
7. Hay que realizar un video poniendo en marcha la aplicación y mostrando cómo funciona. Debes seguir el siguiente guion en el video:
   - Debes poner la aplicación en funcionamiento desde el contenedor.
   - Después debes crear un nuevo usuario, verificar el email que se reciba y loguearte con ese usuario.
   - Ese usuario tendrá el rol de usuario y no lo modificaremos.
   - Con ese usuario debes demostrar que puede acceder o no a las rutas indicadas.
   - Después te deslogueas y muestras las rutas a las que puedes acceder sin necesidad de estar logueado.

- Tras ello, te logueas con uno de los usuarios que hayas dado de alta previamente al video y que hayas modificado en la BB.DD. para que tenga el rol de administrador.
- Entonces con ese usuario debes mostrar como puedes acceder a cada una de las rutas que solamente se puede acceder como administrador.
- El video debe durar como mínimo 5 minutos. No hay límite de tiempo, por si os tarda en renderizar las pantallas.
- El video debe estar alojado en el OneDrive o SharePoint proporcionado por la Florida con el MS Office 365 y compartido únicamente con el profesor a la cuenta fdiaz-alonso@florida-uni.es **cualquier otra forma de compartir el video no se considera valida y no se evaluará.**

8. Entregar autorúbrica del trabajo realizado según la plantilla adjunta.

Entregar:

- Se entregará un fichero zip con todo el contendor de docker. Como ya se hizo en las anteriores evaluables. Como el fichero no se va a poder subir a Florida Oberta, debe subirse a una carpeta de OneDrive o SharePoint y compartirla, al igual que el video.
- Recordar que en el zip debe estar también la BB.DD. y la carpeta del docker.
- Incluir el archivo de texto con la autorúbrica rellena.

Rúbrica

Existen cuatro niveles en los que se cataloga el resultado de la actividad:

1. **Mínimos:** (Este nivel es excluyente, si no se cumplen todos los puntos se considera la actividad suspendida con una nota máxima de 4) **(Máxima nota 5)**
   - La versión usada de Symfony debe ser la LTS (6.4).
   - Crear la aplicación de Symfony, con todas las rutas con acceso público funcionales.
   - Que existan y funcionen las rutas de logueo y deslogueo.
   - No deben existir más bundles de los necesarios.
   - Entregar el video con el tiempo mínimo estipulado.
   - Entregar la carpeta del docker completa y la carpeta de la bb.dd.
   - Entregar la autorúbrica.
2. **Suficiencia (Máxima nota 6)**
   - Cumplir todos los puntos de mínimos. **(OBLIGATORIAMENTE)**
   - Que se puedan registrar usuarios en la aplicación y se reciba el email correspondiente.
3. **Notoriedad (Máxima nota 8)**
   - Cumplir todos los puntos de suficiencia. **(OBLIGATORIAMENTE)**
   - Ha de modelarse correctamente en Symfony la tabla usuarios.
   - Además, tendremos completamente terminado el sistema de administración de usuarios.
4. **Excelencia. (Máxima nota 10)**
   - Cumplir todos los puntos de notoriedad. **(OBLIGATORIAMENTE)**
   - Ha de modelarse correctamente en Symfony la tabla reservas.
   - Además, tendremos completamente terminado el sistema de administración de reservas.

5. **Extras:** (estos puntos suman o restan en cualquiera de los cuatro niveles)
   - Debe de configurarse los registros de la BB.DD. lo más fiel posible, es decir si es nulo o no, si es único, etc…
   - Las asociaciones deben ser bidireccionales.
   - El tipeado de las variables en PHP y en Doctrine debe ser el correcto.
   - Contenido del vídeo debe ser entendible y su contenido adecuado a lo solicitado. Se debe mostrar en él:
   - Que no exista ningún corte entre el arranque de la aplicación, la apertura del navegador en la ruta de la raíz del proyecto.
   - Que la imagen del vídeo se pueda ver, y tenga una nitidez adecuada.
   - Que el audio del vídeo se pueda oír sin problemas.
   - Incluir la hoja de autorúbrica rellena y argumentados los porques.
   - Todo el texto de la aplicación tiene que estar en castellano, incluido el email recibido. Solamente se permiten las rutas en otros idiomas.

Autorúbrica

Para la Autorúbrica debes añadir un documento de texto que contenga una tabla como la siguiente y respondiendo todos los puntos:

Niveles	**Cumplido**	**Observaciones**
Leyenda →	Sí o no	Indicar porque se considera que se ha cumplido y porque no.
Mínimos		
Suficiencia		
Notoriedad		
Excelencia		