Sesión # 14 Componente Práctico

Método de autenticación basado en usuario y contraseña

Dentro del desarrollo de nuestra aplicación de registro para vacunación, vamos a empezar a trabajar con el manejo de sesiones de usuario, de manera que los usuarios ya registrados puedan acceder (autenticarse) y descargar los soportes de sus turnos. Para ello, utilice el esqueleto del proyecto adjunto e instale la librería session de flask: `pip install Flask-Session` y siga los siguientes pasos

1. Crea un archivo base.html
   a. Inserta un condicional de si la sesión existe, muestre la opción de logout.
   b. Si no hay sesión, dar opción de login y register.
   c. Sin importar si hay o no sesión, se debe mostrar un enlace para la página Contáctenos.

2. Modifica los archivos html creados hasta el momento para incluir una extensión de base.html

3. Crea un archivo send.html con un formulario para enviar mensajes que incluya los campos: para, asunto y mensaje. La acción 'submit' del formulario será la URL para send.

4. Dentro del archivo app.py
   a. Dentro de la petición de register, modifica el password antes de insertarlo en la base de datos. Usa la función generate_password_hash antes de realizar el insert. (realiza `pip install werkzeug` e importa: `from werkzeug.security import generate_password_hash, check_password_hash`)
   b. Crea la función `login_required` que valide si existe un usuario y sino, reenvíe al usuario a la vista /login.
   c. Modifica la función login para validar el password ingresado contra el almacenado. Para ello, puedes validar con la función check_password_hash.
   d. Modifica la función `login`. Si el usuario y la contraseña fueron ingresados exitosamente, vamos a eliminar la sesión anterior y asignar la nueva sesión al nuevo usuario.
   e. Crea una función para la ruta `/logout` que cierre la sesión y redireccione a login.html.
   f. Crea una función para la ruta /send que valide o haga llamado a `login_required`, valide los campos del formulario de la página send.html e inserte el mensaje enviado en la base de datos.
   g. Crea una función `app.before_request` que se ejecuta previo a cada request y valide el usuario de la sesión.
   h. Modifica la función de entrada para reenviar a la vista si existe un usuario o a la vista login si no existe.
   i. Modifica la función register, para que si existe el usuario envíe a send sino a register.
   j. Crear una función downloadpdf que retorne un archivo que se encuentre en resources/doc.pdf ejemplo: `return send_file("resources/doc.pdf", as_attachment=True)`.
   k. Crea una función downloadimage que retorne un archivo que se encuentre en resources/image.png.

l. Modifica la función login para que cuando se confirme la sesión, se cree una cookie del tipo ‘username’ y almacene el usuario.

m. Modifica la función send para que recupere la cookie creada y en los mensajes de error incluya el valor de username.

5. Modifica el archivo base.html y adiciona dos link para downloadpdf y downloadimage si existe el usuario.

Nota: En caso dado encuentres errores al momento de realizar `flask run`, ejecuta `python3 app.py`