Clase 12 – Ejercicio - Flask y Jinja2
Curso: Introducción al Desarrollo de Páginas Web – II Semestre 2024 – Grupo 50
Profesor: Adalberto Cubillo Rojas

Indicaciones

Con el uso de Python, Flask y Jinja2 vamos a crear una aplicación web capaz de generar dinámicamente una página HTML. Este sitio web permitirá registrarse o iniciar sesión, y una vez iniciada la sesión se va a mostrar una página con la información del usuario (perfil de usuario) en donde se podría agregar/modificar la información.

Estructura del proyecto

La estructura del proyecto puede ser la siguiente:

- **app.py**: Contiene el código de la aplicación Flask.
- **templates/**: Carpeta donde estarán las plantillas HTML que serán renderizadas por Jinja2.
- **static/**: Aquí puedes incluir archivos estáticos como CSS o imágenes.

Detalles de cada página

Para cada una de estas páginas vamos a emplear plantillas de Jinja2.

En el registro de usuario, se va a emplear correo electrónico, nombre de usuario y contraseña para registrarse. Todos los campos son requeridos, el formato del correo electrónico debe ser válido, el nombre de usuario no puede estar repetido y la contraseña debe llevar un largo de 10 caracteres.

En el inicio de sesión, se utilizará la lista de usuarios registrados para entrar al sitio web. Esto a través del nombre de usuario y contraseña. Ambos campos son requeridos.

Una vez el usuario inicia sesión, debe mostrarse el perfil del usuario con la siguiente información:

- foto/avatar.
- correo electrónico.

- nombre de usuario.
- nombre completo.
- dirección.
- fecha de nacimiento.
- URL para LinkedIn.
- URL para Facebook/Instagram/X/etc. (este campo solo permite agregar 1 red social al perfil).
- contraseña (obscurecida con ******* a la hora de mostrar el perfil).

Inicialmente, de estos campos solo correo electrónico, nombre de usuario y contraseña van a tener datos. Pero el perfil de usuario debe mostrar una opción de editar el perfil en donde se pueda agregar/cambiar toda la información excepto el correo electrónico. Una vez guardados los cambios se vuelve a cargar la información del perfil personal.

Finalmente, el perfil debe mostrar un botón para cerrar sesión. Y volver a la pantalla de inicio de sesión o registro de usuario.

Servicios web

Para el ejercicio se requieren los siguientes servicios web desarrollados utilizando Flask:

- **login/ (POST)**: recibe nombre de usuario y contraseña. Devuelve:

  {
  "status": "true/false" (resultado del inicio de sesión),
  "username": "string" (el nombre del usuario que inicio sesión),
  "error": "string" (en caso de status = false, devuelve un error).
  }

- **signup/ (POST)**: recibe correo electrónico, nombre de usuario y contraseña. Devuelve:

  {
  "status": "true/false" (resultado del registro de usuario, es false si el nombre de usuario ya existe),
  "error": "string" (en caso de status = false, devuelve un error).
  }

- **getProfile/<username> (GET)**: en el URL recibe el nombre del usuario. **Y devuelve la información del usuario solo si este es el usuario que inicio sesión previamente**.

  {

"status": "true/false" (resultado de la solicitud, false en caso que el usuario solicitado no es el que se autenticó),
"profile": {
"username": "string",
"email": "string",
"pictureURL": "string/path",
"fullName": "string",
"address": "string",
"birthday": "string/date",
"linkedIn": "string/URL",
"socialMedia": "string/URL"
},
"error": "string" (en caso de status = false, devuelve un error).
}

- **editProfile/<username> (PUT)**: envía los cambios al perfil y los almacena. **Solo se puede editar si el nombre de usuario que se envía es el que inicio sesión**.

{
"status": "true/false" (resultado de la solicitud, false en caso que el usuario solicitado no es el que se autenticó),
"profile": {
"username": "string",
"email": "string",
"pictureURL": "string/path",
"fullName": "string",
"address": "string",
"birthday": "string/date",
"linkedIn": "string/URL",
"socialMedia": "string/URL"
},
"error": "string" (en caso de status = false, devuelve un error).
}

- **signout/<username> (POST)**: envía una solicitud para cerrar sesión con el nombre de usuario actual.

Manejo de datos

Para el manejo de las sesiones se puede utilizar la siguiente guía como base https://flask.palletsprojects.com/en/3.0.x/quickstart/#sessions

Se utilizará por defecto SQLite para almacenar la información. Esta guía sirve de base para poder desarrollar el ejercicio https://flask.palletsprojects.com/en/3.0.x/tutorial/database/