

Trabajo Practico IP

En este informe, proporcionaremos una breve explicación de las funciones implementadas, las nuevas funciones añadidas, las modificaciones realizadas en el código del programa, y las dificultades encontradas durante el proceso.

En primera instancia, las funciones implementadas en `service_nasa_image_gallery.py`

- La función **getAllImages(input=None)**: obtiene imágenes de otra función llamada `getAllImages` proveniente de la carpeta `transport.py`, las transforma en un formato específico y las devuelve como una lista.
- La función **getImagesBySearchInputLike(seach_input)**: Normaliza el parámetro de búsqueda, obtiene imágenes de una fuente externa según la palabra ingresado por el usuario, las transforma en un formato específico y las devuelve como una lista.
- La función **saveFavourite(request)**: guarda una instancia de **NASACard** como favorito asociado al usuario actual en la base de datos.

En esta primera instancia, no se presentaron complicaciones. Aunque inicialmente no teníamos conocimiento de como trabajaban esas funciones de la carpeta `mapper.py`, como por ejemplo **fromRequesIntoNASACard**, nos dábamos una idea de los resultados que retornaban.

En segunda instancia, las funciones implementadas en `views.py`

- Se implemento una función para el login llamada **“login_view(reques)”**: La cual maneja el inicio de sesión del usuario, autenticando al usuario con las credenciales proporcionadas y, si son correctas, inicia la sesión y redirige a la página principal. Si la autenticación falla, muestra un mensaje de error.
- La función **getAllImagesAndFavouritesList(request)**: obtiene todas las imágenes de la galería de la NASA y la lista de imágenes favoritas del usuario, y luego las devuelve.

- La función **home(request)**: obtiene todas las imágenes y la lista de imágenes favoritas del usuario, luego renderiza la plantilla `home.html` con estas listas para mostrar la galería de imágenes y los favoritos del usuario en la página principal.
- La función **search(request)**: Obtiene el término de búsqueda ingresado por el usuario desde la solicitud POST y lo convierte a minúsculas. Luego, utiliza este término para filtrar las imágenes a través de la función que se encuentra en **“services_nasa_image_gallery.getImagesBySearchInputLike”**. Si el usuario está autenticado, también obtiene su lista de favoritos. Finalmente, renderiza la plantilla `'home.html'` con las imágenes filtradas y la lista de favoritos.
- La función **getAllFavouritesByUser(request)**: Obtiene la lista de imágenes favoritas del usuario autenticado y las renderiza en la plantilla `'favourites.html'`.
- La función **saveFavourite(request)**: Esta función guarda una imagen en la lista de favoritos del usuario autenticado. Después de guardar el favorito, redirige al usuario a la página `'home'`.
- La función **deleteFavourite(request)**: Esta función elimina una imagen de la lista de favoritos del usuario autenticando. Después de eliminar el favorito, redirige al usuario a la página.

Los inconvenientes presentados en esta segunda instancia fueron varios. Primero, al intentar mostrar las imágenes en pantalla, solo se cargaban como archivos sin imagen, sin título ni descripción. Esto se solucionó teniendo en cuenta el código de otras funciones, ya que se retornaba solo un parámetro de la función **“getAllImagesAndFavouritesList(request)”** a la función **“home(request)”** y esta necesitaba ambos parámetros. Posteriormente, se implementó el extra de favoritos sin tener problemas. La implementación del inicio de sesión se realizó tomando en cuenta el video de referencia del TP, pero añadiendo una autenticación.

Modificaciones del código en **“urls.py”** y **“header.html”**

- Se modificaron las rutas del login y el logout para que funcionaran con la función “login_view(request)”. Además, se modificó el ensalce de navegación para el inicio de sesión y el cierre.

En resumen, logramos implementar las funciones principales para mostrar las imágenes con sus títulos y descripciones, el inicio de sesión, la gestión de favoritos por parte del usuario y la eliminación de estos favoritos.