

Informe Trabajo Práctico Introducción a la Programación

Integrantes:

Brener, Matías

Herrera, Maximiliano

Villafañe, Hugo

Comisión 05

Nuestra tarea es completar el desarrollo de una de una página web en la que se muestran distintas imágenes de la Nasa. Las ilustraciones tienen que ser presentadas en forma de *cards*, e incluir el título y la descripción de cada una de ellas. Entre otras funcionalidades, debemos lograr que el usuario pueda registrarse para marcar imágenes como sus favoritas.

Realizamos lo siguiente:

En `nasa_image_gallery/views.py`:

```
def home(request):  
    images = []  
    favourite_list = []  
    images = services_nasa_image_gallery.getAllImages()  
    favourite_list = getAllImagesAndFavouriteList(request)  
    return render(request, 'home.html', {'images': images, 'favourite_list': favourite_list})  
)
```

Dentro de la función `home`, en la lista `images` guardamos los resultados obtenidos al invocar a la función `getAllImages()` desde `services_nasa_image_gallery.py`

Cuando el usuario está logueado, en la lista `favourite_list` se guarda el listado de imágenes al llamar a la función `getAllImagesAndFavouriteList()` con el argumento `request`.

En `services/services_nasa_image_gallery.py`:

Completamos la función `getAllImages()`:

En la lista `json_collection` llamamos a la función `getAllImages()` desde `transport.py` y mediante un condicional, hacemos que muestre un listado de acuerdo al argumento.

Luego hacemos un recorrido de los objetos `json` con un ciclo `for` y los agregamos a la lista `images`.

En la variable `nasa_card` se guardan las imágenes transformadas en cards con la función `fromRequestIntoNASACard()` desde `mapper.py`.

```
def getAllImages(input=None):
    json_collection = []
    images = []
    if input:
        json_collection = transport.getAllImages(input)
    else:
        json_collection = transport.getAllImages(None)

    for json in json_collection:
        nasa_card = mapper.fromRequestIntoNASACard(json)
        images.append(nasa_card)

    return images
```

De esta manera, al clicar en el link, se muestra en la galería el listado de imágenes de la Api.

Buscador:

Para hacerlo funcionar, completamos la función `search()` en `views.py` que, de acuerdo al request, llama a la función `getAllImages()` y devuelve un listado de imágenes relacionadas al tema buscado. En este caso le agregamos un condicional para que cuando el usuario no ingrese ninguna palabra en el buscador, este devuelva por defecto imágenes relacionadas al tema “Espacio”

```
def search(request):
    images, favourite_list = getAllImagesAndFavouriteList(request)
    search_msg = request.POST.get('query', '')
    if not search_msg:
        search_msg="space"
    else:
        images=services_nasa_image_gallery.getAllImages(search_msg) #

    return render(request, 'home.html', {'images': images, 'favourite_list': favourite_list})
```

Login:

En templates/header.html modificamos los vínculos correspondientes al inicio/cierre de sesión para que nos permita acceder a login.html.

```
{% if request.user.is_authenticated %}
    <a class="nav-link" href="{% url 'exit' %}">Salir</a> {% else %}
    <a class="nav-link" href="{% url 'login' %}">Iniciar sesión</a>
```

En main/urls.py/ agregamos la ruta path('accounts/',include('django.contrib.auth.urls')),

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('nasa_image_gallery.urls')),
    path('accounts/',include('django.contrib.auth.urls')),
]
```

Favoritos:

La funcionalidad de "Favoritos" permite a los usuarios crear una lista personalizada de sus imágenes preferidas. Los usuarios pueden añadir imágenes a esta lista mientras navegan por la galería, guardando aquellas que les resulten más interesantes o significativas. Esta lista de favoritos es accesible en cualquier momento, siempre que el usuario esté logueado.

Para lograr esta funcionalidad modificamos la función saveFavourite() de *repositories.py* que se utiliza para guardar favoritos, se genera la variable fav, la cual basada en el modelo favourite, guarda los datos de la imagen, título, descripción, url y usuario:

```
def saveFavourite(image):
    try:
        fav = Favourite.objects.create(title=image.title, description=image.description,
        image_url=image.image_url, date=image.date, user=image.user)
        return fav
    except Exception as e:
        print(f"Error al guardar el favorito: {e}")
        return None
```

Las siguientes funciones se utilizan para implementar la sección de favoritos: traer los favoritos de un usuario, guardarlos, eliminarlos y desloguearse de la app:

La función `getAllFavouritesByUser()` trae las imágenes guardadas como "favorito", se utiliza el modelo `favourite` con la acción "filter" basado en el dato "user" y retorna la lista de favoritos utilizada en el template:

```
def getAllFavouritesByUser(request):  
    favourite_list = Favourite.objects.filter(user=request.user)  
    return render(request, 'favourites.html', {'favourite_list': favourite_list})
```

```
def saveFavourite(request):  
    if request.method == 'POST':  
        try:  
            saved_favourite = services_nasa_image_gallery.save_Favourite(request)  
            if saved_favourite:  
                return redirect('favoritos') # Redirige a la vista de favoritos si el  
                favorito se guarda correctamente  
            else:  
                messages.error(request, 'Error al guardar el favorito.')  
        except Exception as e:  
            messages.error(request, f'Error: {str(e)}')  
    return redirect('home')
```

Eliminar:

Función para eliminar favoritos:

```
def deleteFavourite(request):  
    if request.method == "POST":  
        success = services_nasa_image_gallery.deleteFavourite(request)  
        if success:  
            return redirect('favoritos')  
        else:  
            messages.error(request, 'Error al guardar el favorito.')  
            return redirect('favoritos')  
    else:  
        return redirect('favoritos')
```