message framework

Quite commonly in web applications, you need to display a one-time notification message (also known as “flash message”) to the user after processing a form or some other types of user input.

For this, Django provides full support for cookie- and session-based messaging, for both anonymous and authenticated users. The messages framework allows you to temporarily store messages in one request and retrieve them for display in a subsequent request (usually the next one). Every message is tagged with a specific **level** that determines its priority (e.g., **info**, **warning**, or **error**).

https://docs.djangoproject.com/en/2.1/ref/contrib/messages/

views.py

```
from django.contrib import messages

messages.success(request,"Post has been successfully created”)

return redirect(‘post_list')

#All type of messages{

#messages.debug(request, '%s SQL statements were executed.' % count)
#messages.info(request, 'Three credits remain in your account.')
#messages.success(request, 'Profile details updated.')
#messages.warning(request, 'Your account expires in three days.')
#messages.error(request, 'Document deleted.’)
#}
```

post_list.html

```
 {% include 'blog/alerts.html' %}
```

alerts.html

```
<div class="messages">

      {% if messages %}

         {% for message in messages %}

            <p class="alert alert-{{ message.tags }}">{{ message }}</p>

         {% endfor %}

      {% endif %}

</div>
```

Js

```
   $(function(){
       setTimeout(function(){
         $('.alert').slideUp(2000);
       }, 5000);
      });
```