Topic: **Django Signals**

Question 2:

Do django signals run in the same thread as the caller? Please support your answer with a code snippet that conclusively proves your stance. The code does not need to be elegant and production ready, we just need to understand your logic

Answer:

Yes, by default, Django signals execute on the same thread as the caller. This indicates that the signal handler, or receiver, is running in the same thread as the signal and will halt the sender's execution until the handler is finished.

To prove this, we can use the threading module to display the threads IDs of the both sender(the functions that sends the signal) and the receiver (the signal handler). If they have the same thread ID, it confirms that they are running in the same thread.

Program

Following code demonstrate that Django signals run in the same thread:

#models.py

```
import threading
from django.db import models
from django.db.models.signals import post_save
from django.dispatch import receiver

class MyModel(models.Model):
    name = models.CharField(max_length=100)

receiver(post_save, sender=MyModel)
def my_signal_handler(sender, instance, **kwargs):
    print(f"Signal handler running in thread:
{threading.current_thread().name}, ID: {threading.get_ident()}")
```

#views.py

```
from django.shortcuts import HttpResponse
from .models import MyModel
import threading

def create_model_instance(request):
    print(f"View function running in thread:
{threading.current_thread().name}, ID: {threading.get_ident()}")
    instance = MyModel.objects.create(name="Test Instance")
    return HttpResponse("Instance created")
```

Output:

```
View function running in thread: MainThread, ID: 350635073885007
Signal handler running in thread: MainThread, ID: 350635073885007
```