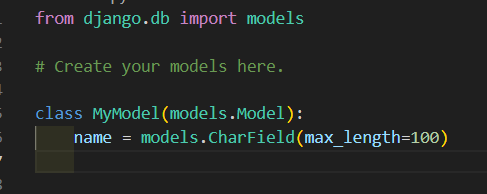
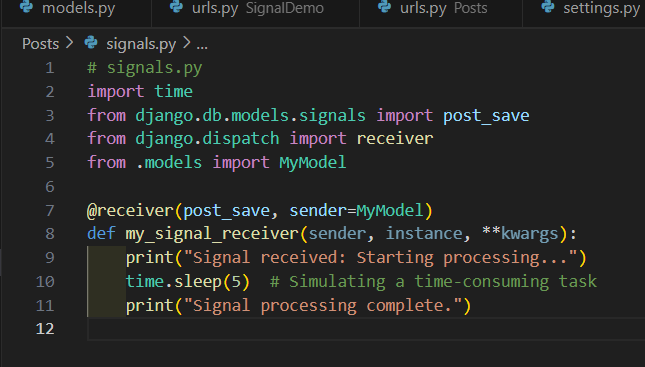
**Q1**.By default are django signals executed synchronously or asynchronously? Please support your answer with a code snippet that conclusively proves your stance. The code does not need to be elegant and production ready, we just need to understand your logic.

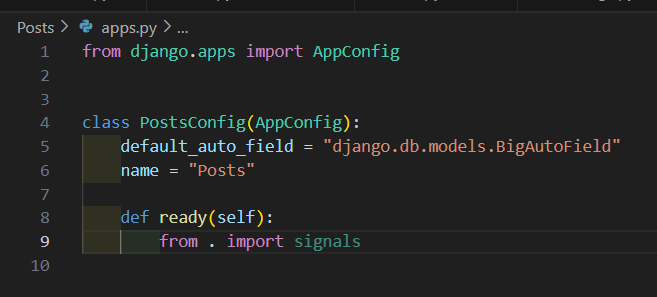
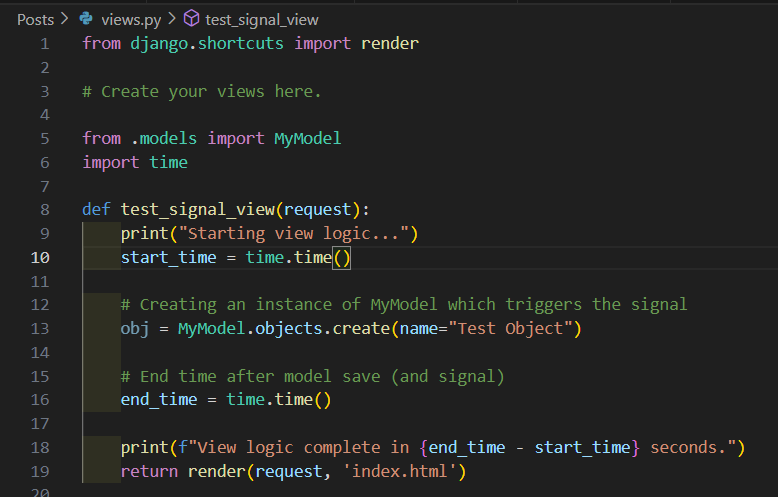
Ans:

By default, **Django signals are executed synchronously**. This means that when a signal is triggered, the handler (receiver function) is executed immediately in the same process. The entire process (like saving a model or handling a request) will wait until the signal’s handler finishes executing before forward.

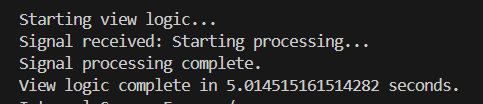
We can prove this by measuring the time taken for a signal handler to execute, and observing that the main process waits for it to complete.

creating model:

creating signal:



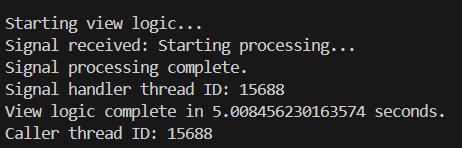
Output:



**Q2.** 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.

Ans:

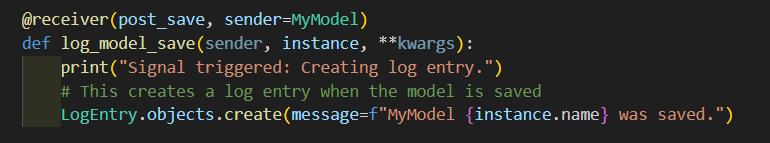
Yes, by default, **Django signals run in the same thread as the caller**. This means that the signal’s handler (receiver function) is executed in the same thread that triggered the signal. You can prove this by checking the thread IDs of both the caller and the signal handler to see if they are the same.

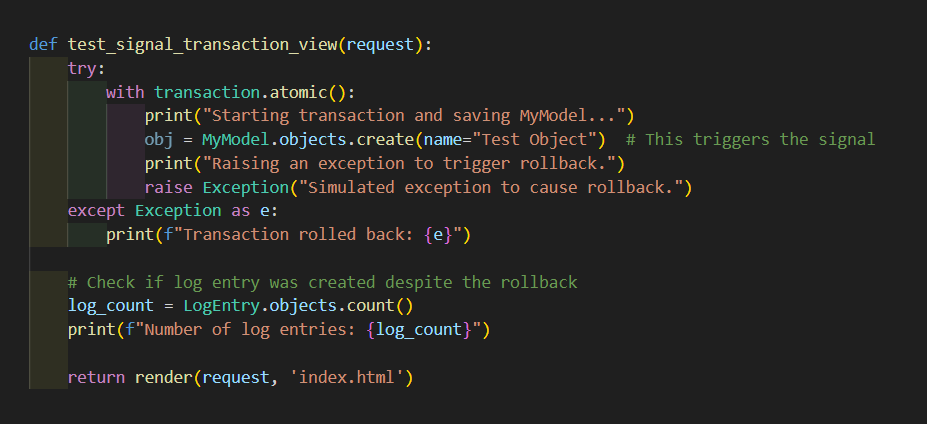


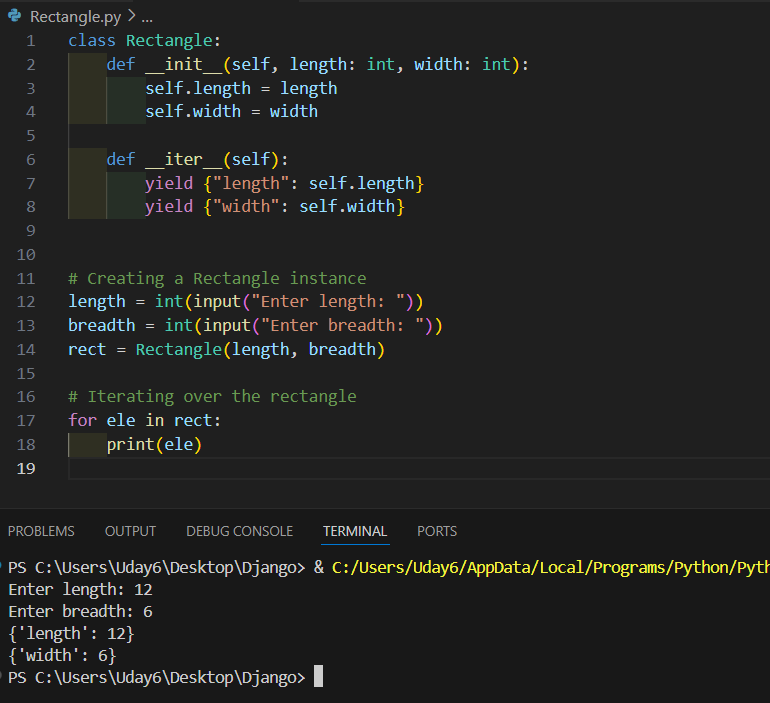
**Q3.** By default do django signals run in the same database transaction as the caller? Please support your answer with a code snippet that conclusively proves your stance. The code does not need to be elegant and production ready, we just need to understand your logic

Ans:

By default, **Django signals run in the same database transaction as the caller** if the signal is triggered during a transaction-bound operation like saving or deleting a model. This means that if the caller is in a transaction, the signal's receiver will execute within that same transaction, and if the transaction fails (e.g., due to an exception), any changes made by the signal will be rolled back as well.





**Program :**Description: You are tasked with creating a Rectangle class with the following requirements: An instance of the Rectangle class requires length:int and width:int to be initialized. We can iterate over an instance of the Rectangle class When an instance of the Rectangle class is iterated over, we first get its length in the format: {'length': <VALUE\_OF\_LENGTH>} followed by the width {width: <VALUE\_OF\_WIDTH>}