**TASK – Djano Signal   
  
Scenario:**

You are developing a blog application. Every time a Post is published, you need to notify all the subscribers of the blog by adding a record to the Notification model.

**Exercise:**

**Step 1: Create the Post, Subscriber, and Notification Models**

1. **Post** model should contain the following fields:
   * title: A string field to store the post's title.
   * content: A text field to store the post's content.
   * published: A boolean field to indicate whether the post is published or not.
   * created\_at: A timestamp for when the post is created.
2. **Subscriber** model should contain the following fields:
   * email: An email field to store the subscriber’s email.
3. **Notification** model should contain the following fields:
   * subscriber: A ForeignKey to the Subscriber model.
   * post: A ForeignKey to the Post model.
   * message: A string field to store the notification message.
   * timestamp: A timestamp for when the notification is created.

**Step 2: Create a Signal**

Use Django signals to handle the following:

1. **Create a signal** that is triggered whenever a post's published field changes from False to True. The signal should create a Notification for each subscriber, notifying them that a new post has been published.

**Step 3: Connect the Signal**

* Connect the signal to the Post model, so that when a post is published, all subscribers are notified.

**Expected Outcome:**

1. When a post is first created with published=False, no notifications are created.
2. When a post is updated and published is changed to True, a notification is created for each subscriber, with a message like: "A new post titled '{post\_title}' has been published!"