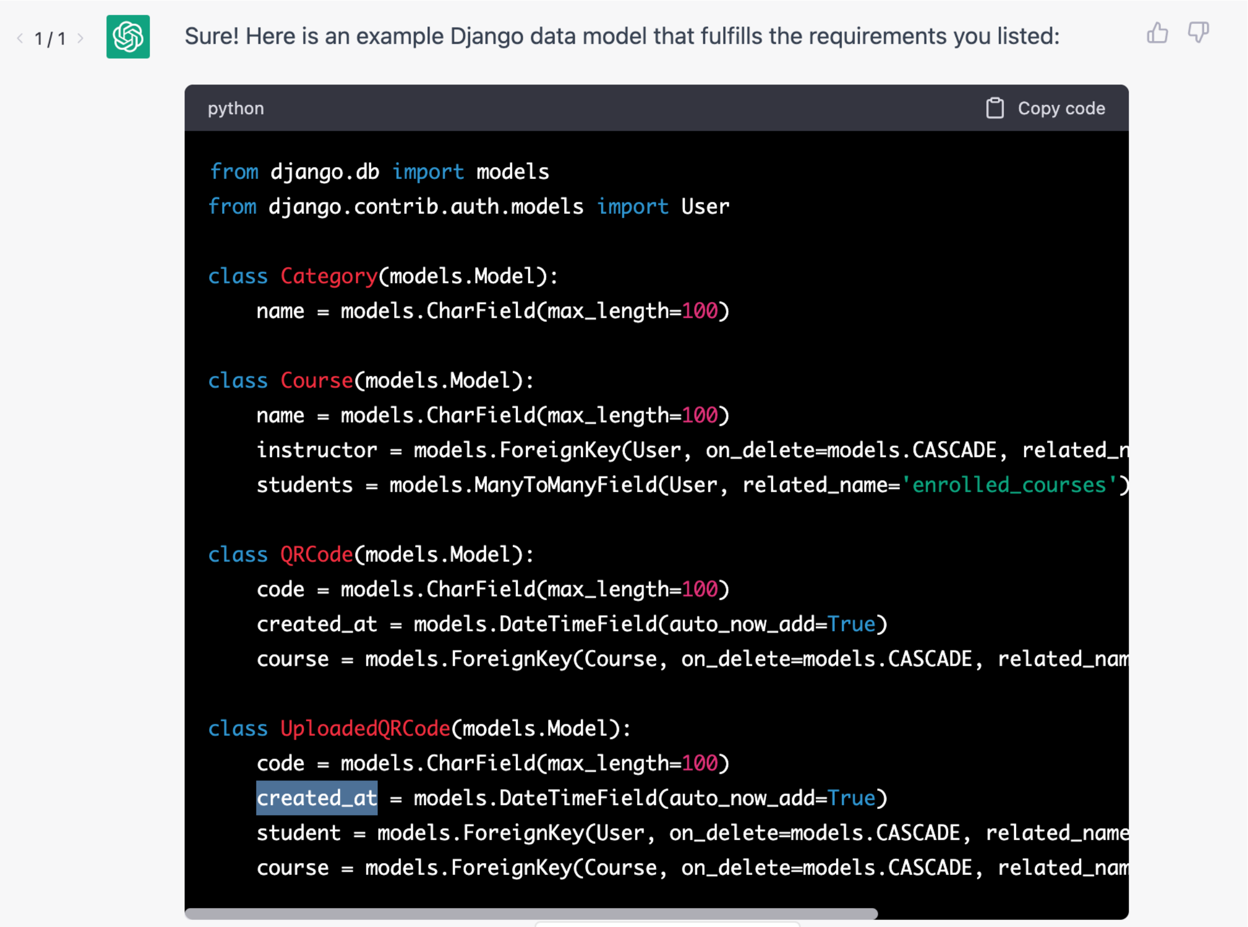
My ChatGPT model produced the following:



Sure! Here is an example Django data model that fulfills the requirements you listed:

```python
from django.db import models
from django.contrib.auth.models import User

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

class Course(models.Model):
    name = models.CharField(max_length=100)
    instructor = models.ForeignKey(User, on_delete=models.CASCADE, related_n
    students = models.ManyToManyField(User, related_name='enrolled_courses')

class QRCode(models.Model):
    code = models.CharField(max_length=100)
    created_at = models.DateTimeField(auto_now_add=True)
    course = models.ForeignKey(Course, on_delete=models.CASCADE, related_nam

class UploadedQRCode(models.Model):
    code = models.CharField(max_length=100)
    created_at = models.DateTimeField(auto_now_add=True)
    student = models.ForeignKey(User, on_delete=models.CASCADE, related_name
    course = models.ForeignKey(Course, on_delete=models.CASCADE, related_nam
```

While this model is simpler because it has less classes and less parts per class (fields), I prefer my version because of the logic. The Chat version utilizes **User**, which we have not fully discussed. I would prefer to have students and teachers separate, as to avoid using a boolean or another Foreign/ManyToMany key. Also, I used the ImageField for both QR code classes, which is the more desirable input for the purposes of our project. (However, I was pleasantly surprised that my model was similar to Chat, and I took that as a win!)