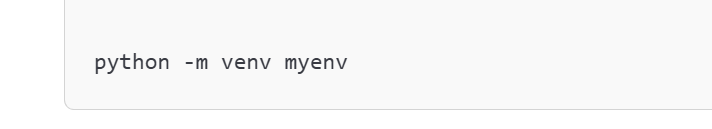
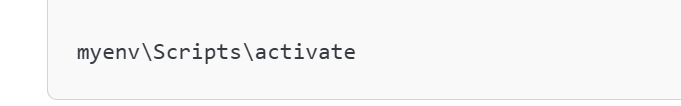
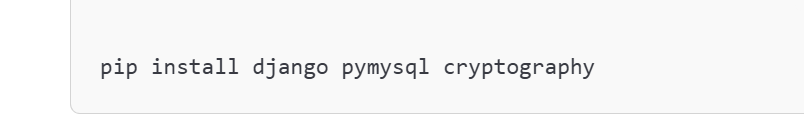
**在你的项目根目录中，创建虚拟环境 python -m venv myenv**

****

**激活虚拟环境CMD： myenv\Scripts\activate**

****

**安装依赖：pip install django pymysql cryptography**

****

**在 settings.py 中，配置 MySQL 数据库。找到 DATABASES 配置部分，修改为：**

**DATABASES = {**

**'default': {**

**'ENGINE': 'django.db.backends.mysql',**

**'NAME': 'inventory\_db', # 你的数据库名**

**'USER': 'inv\_user', # 你的 MySQL 用户名**

**'PASSWORD': 'your\_password', # 该用户的密码**

**'HOST': 'localhost', # 数据库地址（一般为 localhost）**

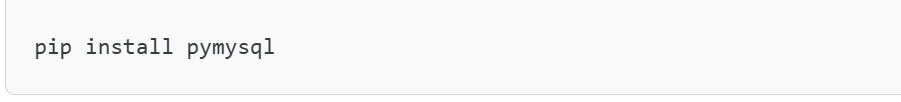
**'PORT': '3306', # MySQL 默认端口**

**}**

**}**

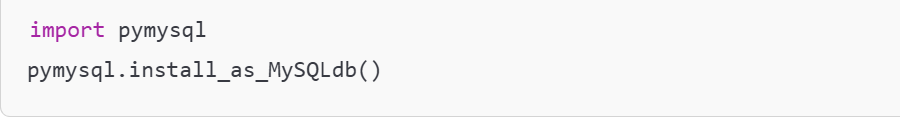
****

**你需要安装 pymysql 库来连接 MySQL 数据库。如果没有安装，请运行：pip install pymysql**

****

**然后，在 settings.py 的顶部加入：import pymysql**

**pymysql.install\_as\_MySQLdb()**

****

**创建数据库 inventory\_db 尚未创建，可以在 MySQL 中执行以下命令：CREATE DATABASE inventory\_db;**

****

**在 Django 应用中找到 models.py 编辑 models.py，写入以下内容：**

**from django.db import models**

**class Product(models.Model):**

**name = models.CharField(max\_length=100) # 产品名称**

**price = models.DecimalField(max\_digits=10, decimal\_places=2) # 产品价格**

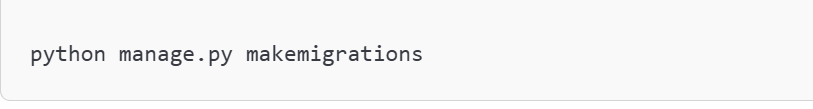
**stock = models.IntegerField() # 库存数量**

**def \_\_str\_\_(self):**

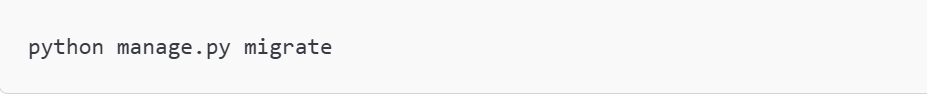
**return self.name**

****

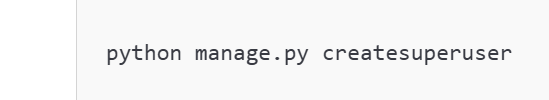
****生成迁移文件**：修改完模型后，运行：python manage.py makemigrations**

****

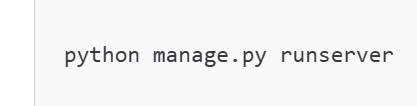
****应用迁移到数据库**：运行迁移，将模型同步到 MySQL 数据库：python manage.py migrate**

****

****创建超级管理员账户，**为了访问 Django 管理后台，创建一个管理员账户：python manage.py createsuperuser 按提示输入用户名、电子邮箱和密码。创建完成后会显示“Superuser created successfully.”。**



****启动开发服务器并访问后台，**在项目根目录，运行开发服务器：python manage.py runserver**



**默认启动在 http://127.0.0.1:8000/。在浏览器访问 http://127.0.0.1:8000/admin/，输入上一步创建的管理员账号，即可进入 Django 后台管理界面。此时可以看到刚才注册的 Product 模型，添加或修改库存数据。**