**Урок 5: Работа с административной панелью**

**Активируем виртуальное окружение**

.\venv\Scripts\activate

**Переходим в папку с проектом**

cd .\recipesbook\

**Создаем приложение и добавляем его в settings.py в INSTALLED\_APPS**

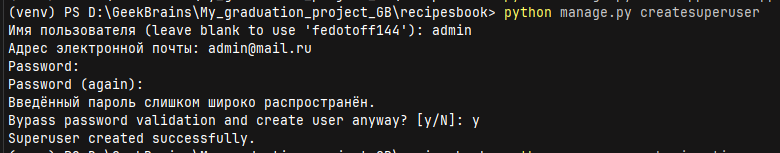
python manage.py startapp myapp

**Запускаем сервер и переходим на сайт в админ панель**

127.0.0.1:8000/admin

**Убеждаемся что проект запускается но пользователя нет, поэтому создаем нового суперпользователя, вводим и подтверждаем пароль для него**

python manage.py createsuperuser



**Создаем и применяем миграции**

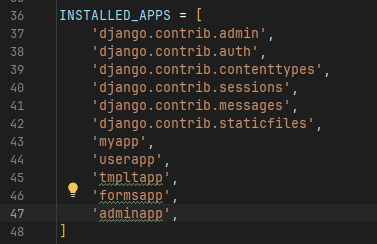
python manage.py makemigrations

python manage.py migrate

**Теперь мы можем заново запустить проект, войти в админ панель и можем добавлять новых пользователей и группы**

**Для того чтобы добавить модели наших приложений в админ панели нам необходимо:**

**-убедиться, что приложение добавление в список в settings.py проекта**

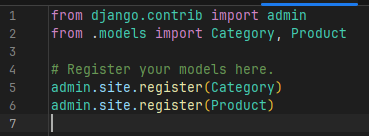
****

**-создать миграции и применить их:**

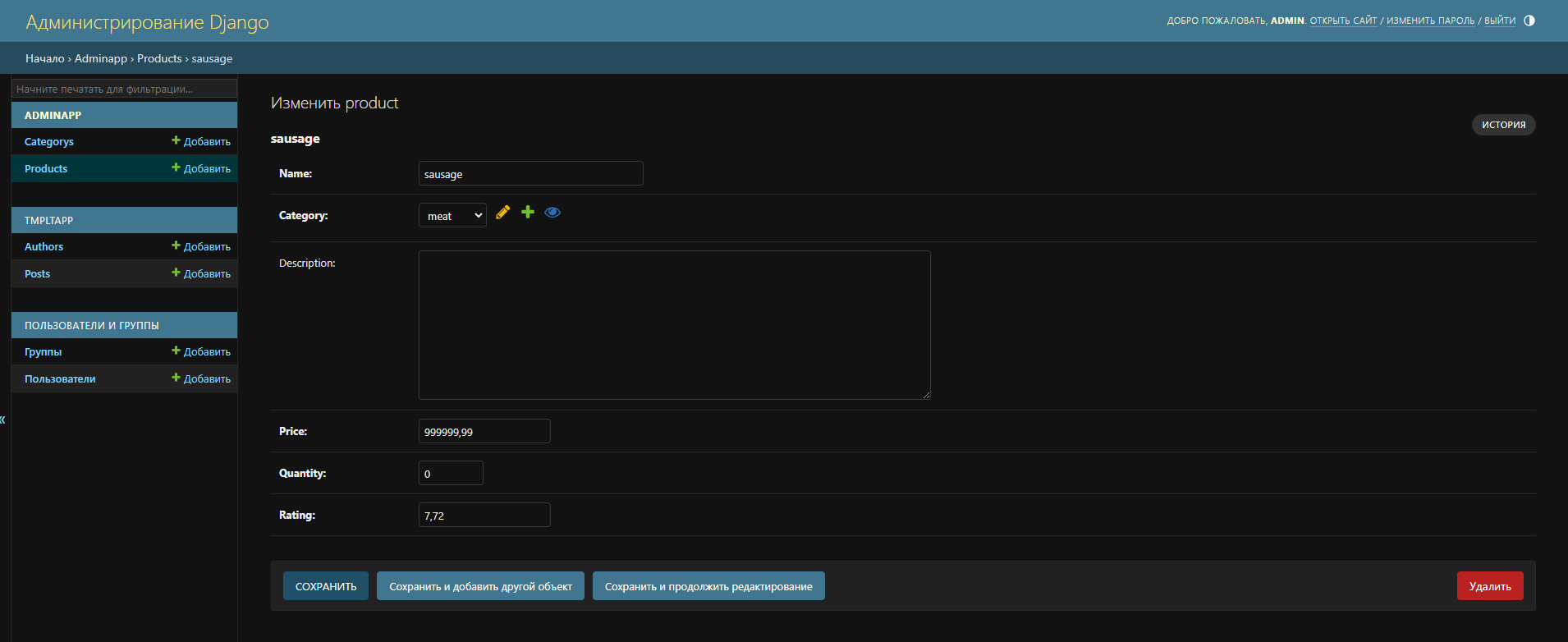
python manage.py makemigrations

python manage.py migrate

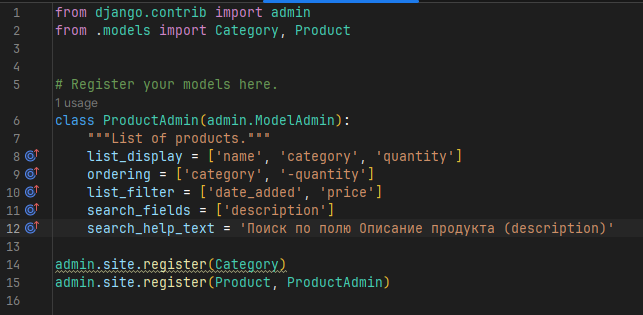
**в файле admin.py каждого приложения зарегистрировать модели в админке:**

****

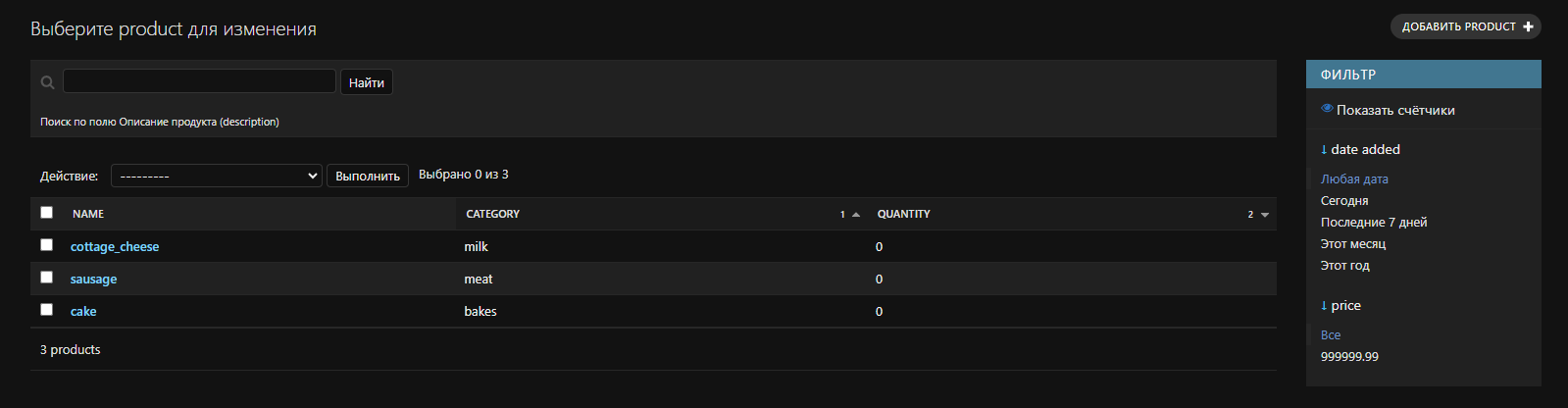
**ВНИМАНИЕ! После добавления моделей в админку, достаточно просто обновить страницу сайта и модели появятся.**

**Добавлять названия продуктов или новые категории из админ панели достаточно просто, нужно всего лишь воспользоваться интуитивно понятным интерфейсом.**

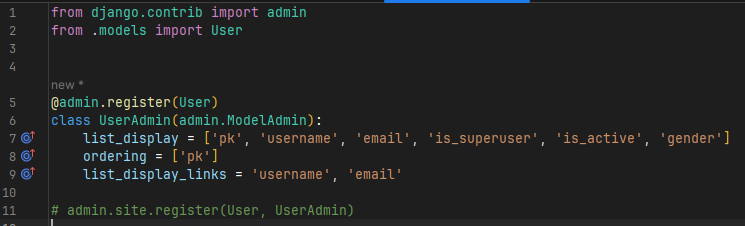
**Для того чтобы добавить отображение колонок, сортировку, фильтр, и поиск у продуктов в админ панели необходимо в файле admin.py нужного приложения с моделью продуктов, добавить класс ProductAdmin, перечислить все настройки и в конце подружить модель Product с классом ProductAdmin как указано на скрине:**

****

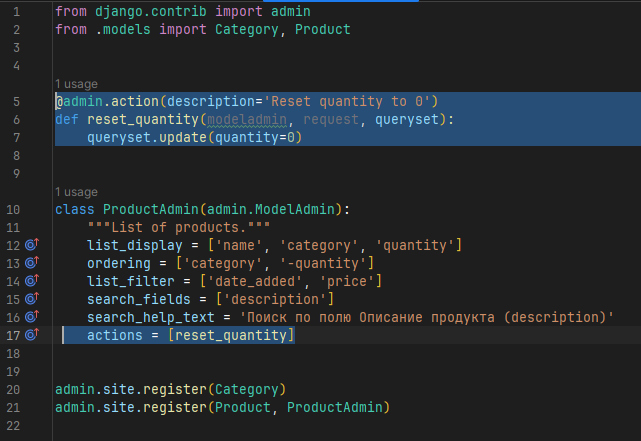
**Теперь имеем следующее отображение в админ панели:**

****

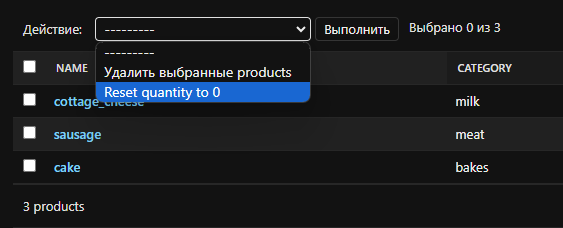
**ВНИМАНИЕ! Зарегистрировать модель в админ панели и настроить отображение необходимых полей можно и через декоратор как показано на скрине ниже.**

****

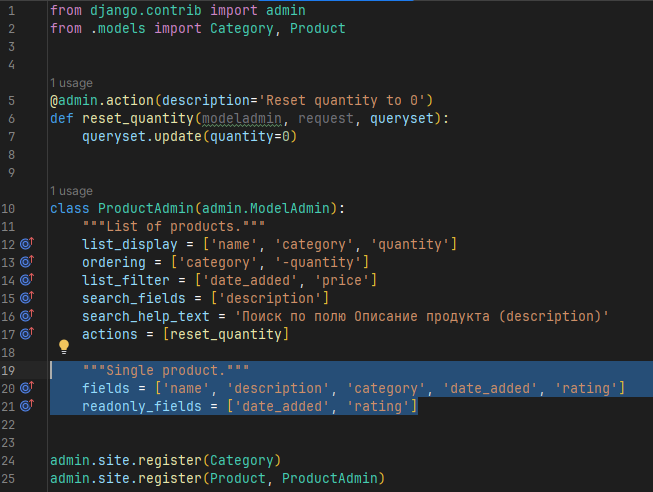
**Для добавления действий в поле «Действия» админ панели, в том же файле admin.py нужно добавить функцию с декоратором:**

****

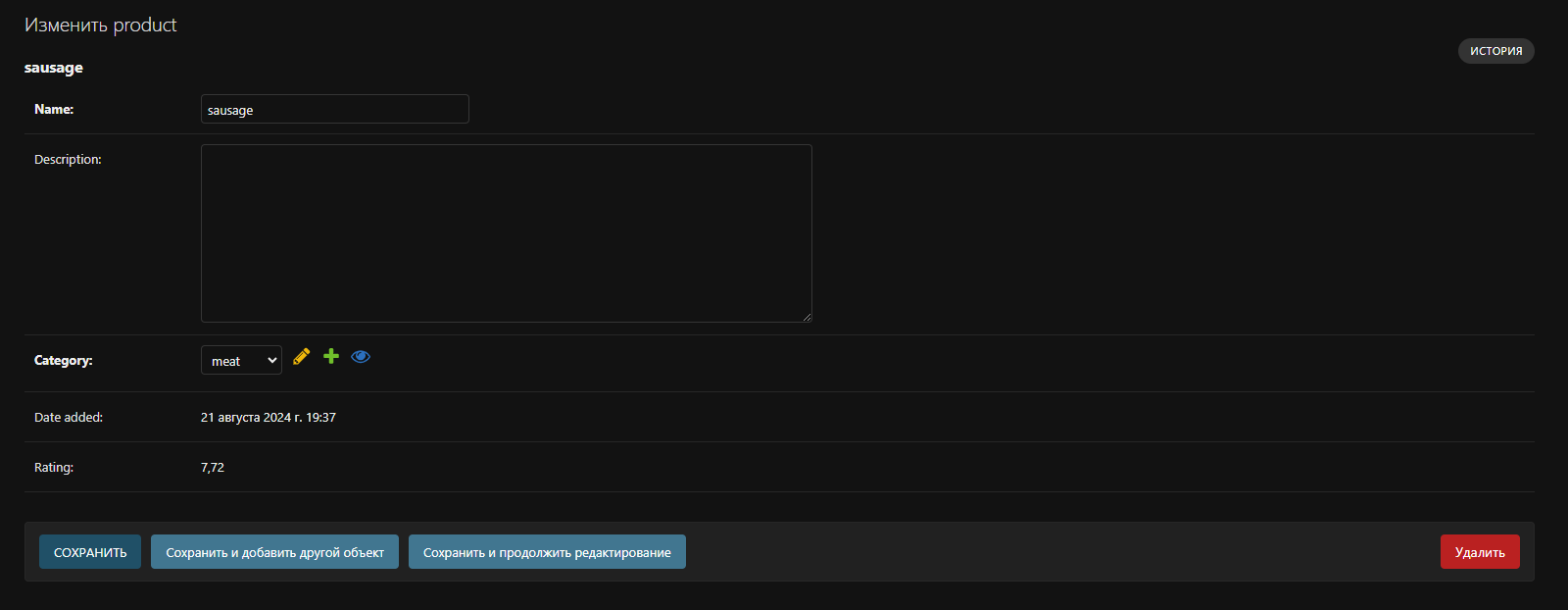
**Теперь имеем следующее отображение в админ панели:**

****

**Для изменения отображаемых полей в админ панели отдельного продукта необходимо, в файле admin.py добавить слеюующее:**

****

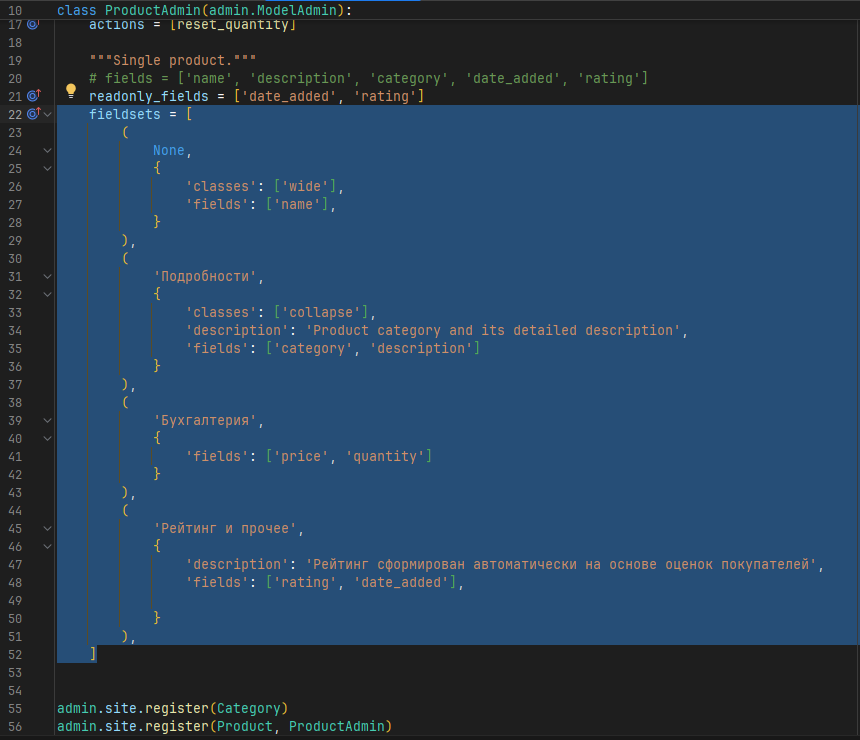
**Теперь имеем следующее отображение в админ панели отдельного продукта:**

****

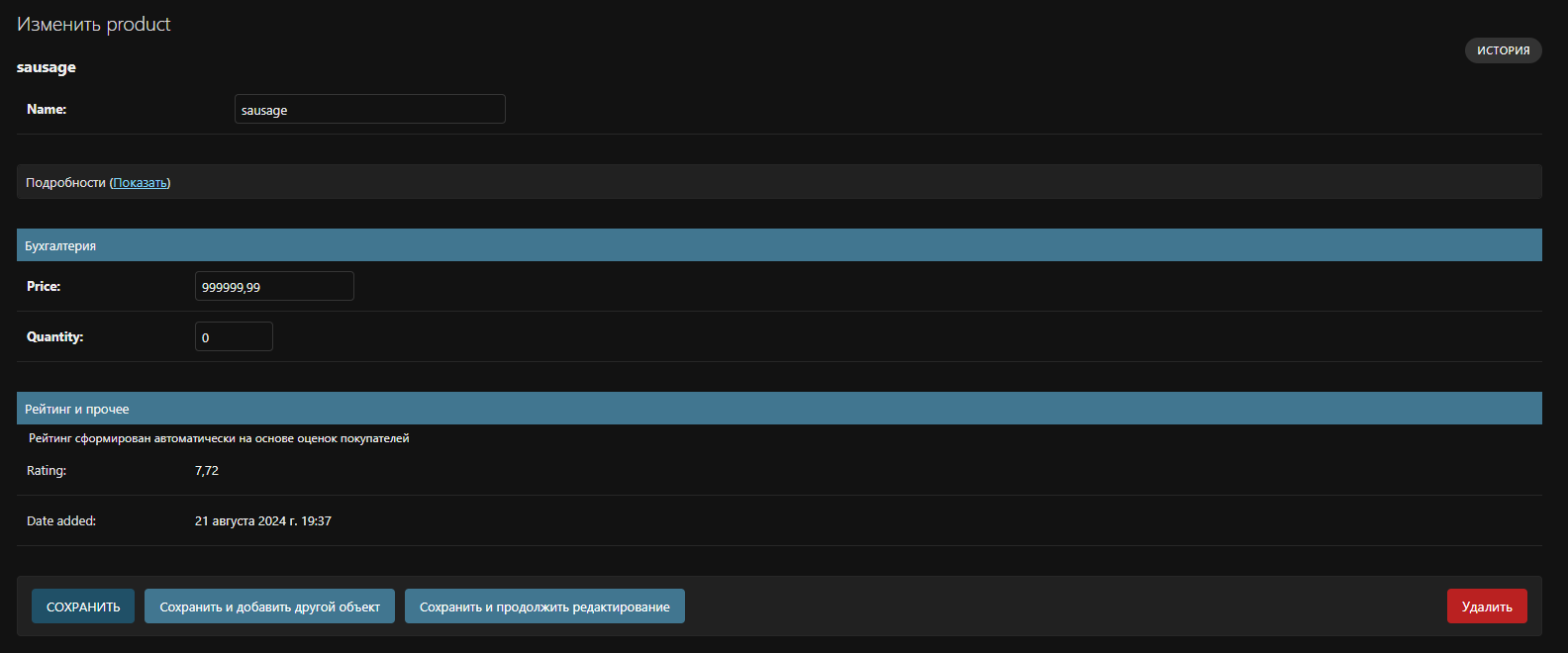
**Мы выбрали поля для отображения, среди которых указали поля только для чтения.**

**Для детальной настройки отображения полей внутри модели, можно воспользоваться переменной fieldsets:**

Внимание! **fieldsets не дружит с fields поэтому его мы закомментировали.**

****

**Теперь имеем следующее отображение в админ панели отдельного продукта:**

****

