**Практическая работа 9. Личный кабинет администратора**

**Задание:**

Добавим в приложение персонализацию для каждого работника (пользователя ИС), в котором будет отображаться информация о его продажах. Для этого подключим таблицу User, и создадим таблицу Сделка, содержащую следующие столбцы:

* Дата (формат дата),
* товар (поле ссылается на таблицу Товар),
* количество (число),
* цена за единицу (число)
* работник – проводивший сделку (ссылка на таблицу User)).

Создадим пользователя с правами администратора (superuser). Зайдем в личный кабинет администрантора django (по ссылке /admin) и добавим еще несколько пользователей (работников предприятия).

Добавить несколько записей в таблицу Сделка (через страницу администратора).

Далее добавить страницу Сделки, на которой вывести строки таблицы Сделки для каждого из работников.

**Пример:**

**1. В файле с описанием моделей models.py** добавим модель Пользователь и Сделка. Добавим следующий код:

from django.contrib.auth.models import User

class Deal(models.Model):  
 product = models.ForeignKey(Product, on\_delete=models.CASCADE, verbose\_name=**"Продукт"**)  
 employee = models.ForeignKey(User, on\_delete=models.CASCADE, verbose\_name=**"Сотрудник"**)  
 date = models.DateField(verbose\_name=**"Дата"**)  
 price = models.IntegerField(verbose\_name=**"Цена за ед."**)  
 quantity = models.IntegerField(verbose\_name=**"Количество"**)  
  
 def \_\_str\_\_(self):  
 return str(self.product) +” ”+ str(self.price)

После этого создадим и выполним миграции для того, чтобы в БД была создана таблица Сделка.

**2. Создадим пользователя – администратора**: в терминале напишем команду python manage.py createsuperuser. После этого заполним предлагаемые поля: имя пользователя, эл.почта, пароль.

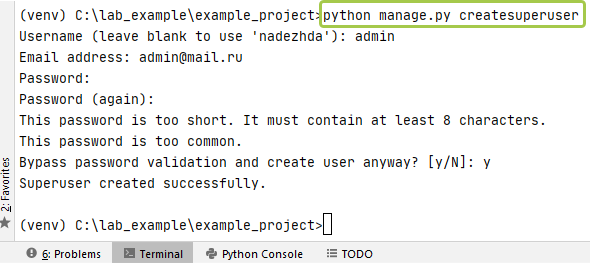


Рисунок. Создание администратора

**3. В файле admin.py зарегистрируем все модели** следующим образом:

from django.contrib import admin  
from .models import \*  
  
admin.site.register(Product)  
admin.site.register(Provider)  
admin.site.register(Store)  
admin.site.register(Deal)

**4. Зайдем в личный кабинет администратора** с созданным ранее логином и паролем.

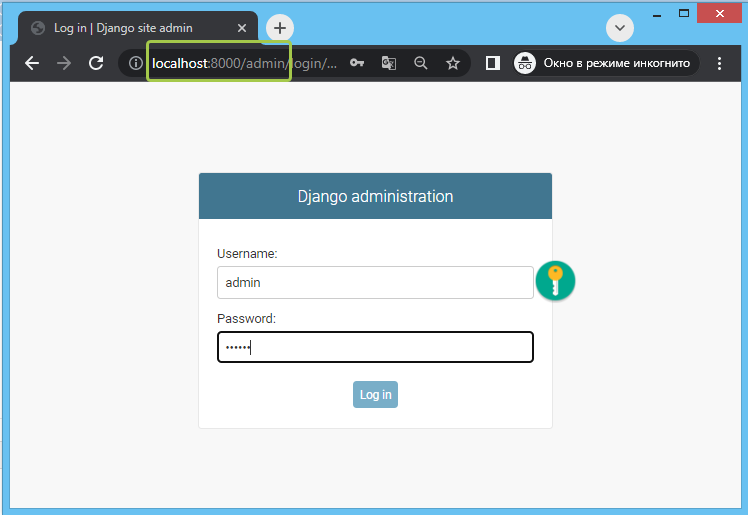


Рисунок. Пример входа в личный кабинет администратора

В панели администратора будет список созданных таблиц и возможность редактировать данные таблицы. Пример.

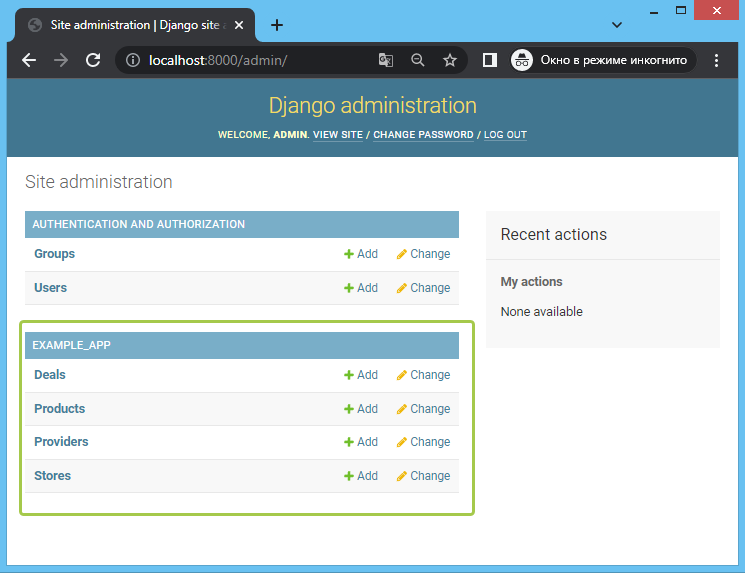


Рисунок. Пример отображения таблиц в панели администратора

В панели администратора добавим записи о пользователях и сделках.

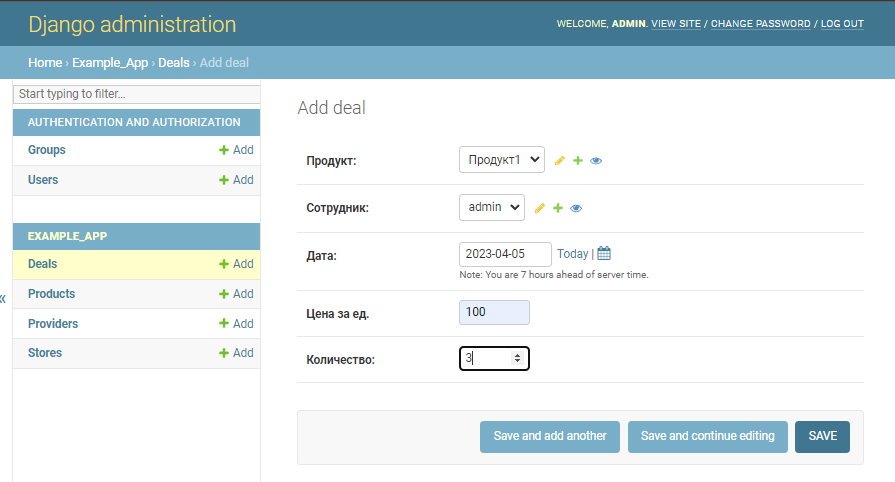


Рисунок. Пример добавления записи в таблицу

**5. Создадим в файле views.py функцию deal(),** в которой будет запрос к таблице Сделки.

def deal(request):  
 deals = Deal.objects.all()  
 return render(request, **'deal.html'**, context={**'deals'**: deals})

**6. Создадим в папке templates страницу deal.html** и добавим на нее код вывода данных из таблицы Сделки

{% extends 'base.html' %}  
{% block header %}  
<h1>Список проданных товаров</h1>  
{% endblock header %}  
{% block content %}  
<table class="table">  
 <tr>  
 <th>Номер</th>  
 <th>Название продукта</th>  
 <th>Сотрудник</th>  
 <th>Дата</th>  
 <th>Количество</th>  
 <th>Цена товара</th>  
 </tr>  
 {% for p in deals %}  
 <tr>  
 <td>{{p.id}}</td>  
 <td>{{p.product}}</td>  
 <td>{{p.employee}}</td>  
 <td>{{p.date}}</td>  
 <td>{{p.quantity}}</td>  
 <td>{{p.price}}</td>  
 </tr>  
 {% endfor %}  
</table>  
{% endblock content %}

**7. Добавим маршрут к странице** сделки в файле urls.py.

from django.urls import path  
from .views import \*  
  
urlpatterns = [  
 path(**''**, index),  
 path(**'contacts'**, contacts),  
 path(**'vacancy'**, vacancy),  
 path(**'products'**, products),  
 path(**'store'**, store),  
 path(**'add\_product'**, add\_product),  
 path(**'add\_provider'**, add\_provider),  
 path(**'deal'**, deal)  
  
]

**8. Перейдем на созданную страницу в браузере.** Далее на рисунке приведен пример внешнего вида полученной таблицы.

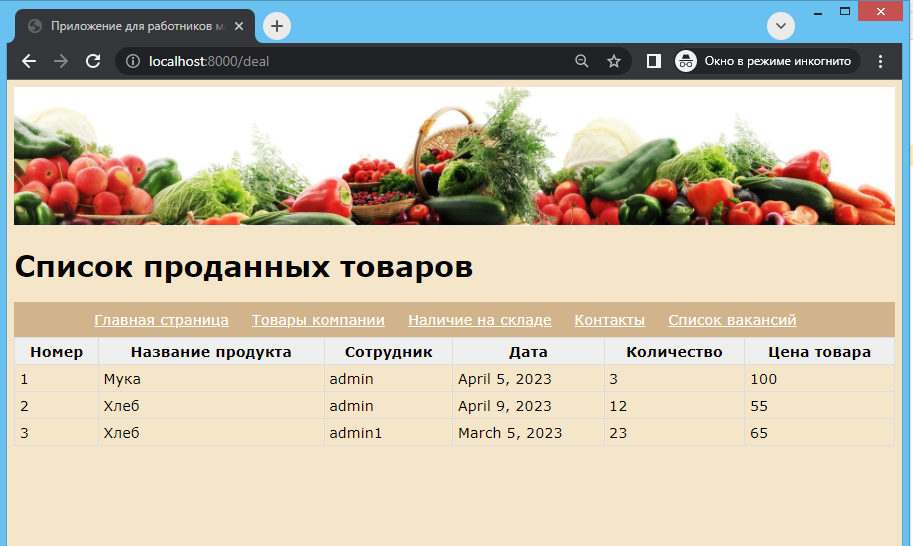


Рисунок. Пример отображения страницы сделка в браузере