**Практическая работа 8. Вывод данных**

**Задание:**

К проекту ИС для торговой компании (созданному в предыдущем задании) добавить следующее:

Добавить на каждую страницу сайта отображения данных из соответствующих таблиц. (например, на страницу с информацией о складе вывести в виде таблицы все записи из таблицы Склад)

**Пример:**

**1. В файле views.py в функции products** добавить запросы для получения данных из таблиц (добавленные строки выделены цветом)

def products(request):  
 productForm = ProductForm()  
 providerForm = ProviderForm()  
   
 product = Product.objects.all()  
 provider = Provider.objects.all()  
 data = {**'productForm'**: productForm, **'providerForm'**: providerForm,  
 **'products'**: product, **'providers'**: provider}  
 return render(request, **'products.html'**, context=data)

**2. На странице products.html добавить код** для вывода данных переменных в таблицы (добавленные строки выделены цветом):

{% extends 'base.html' %}  
{% block header %}  
<h1>Список товаров магазина</h1>  
{% endblock header %}  
{% block content %}  
<h2>Добавление данных</h2>  
<form method**="post"** action**="add\_product"**>  
 {% csrf\_token %}  
 <table class**="form-table"**>  
 {{productForm}}  
 </table>  
 <input class**="btn"** type**="submit"** value**="Добавить продукт"**>  
</form>  
  
<form method**="post"** action**="add\_provider"**>  
 {% csrf\_token %}  
 <table>  
 {{providerForm}}  
 </table>  
 <input class**="btn"** type**="submit"** value**="Добавить поставщика"**>  
</form>  
  
<h2>Список продуктов</h2>  
<table class**="table"**>  
 <tr>  
 <th>Номер</th>  
 <th>Название продукта</th>  
 <th>Описание продукта</th>  
 </tr>  
 {% for p in products %}  
 <tr>  
 <td>{{p.id}}</td>  
 <td>{{p.name}}</td>  
 <td>{{p.description}}</td>  
 </tr>  
 {% endfor %}  
</table>  
<h2>Список поставщиков</h2>  
<table class**="table"**>  
 <tr>  
 <th>Номер</th>  
 <th>Название компании</th>  
 </tr>  
 {% for p in providers %}  
 <tr>  
 <td>{{p.id}}</td>  
 <td>{{p.name}}</td>  
 </tr>  
 {% endfor %}  
</table>  
{% endblock content %}

**3. В файле views.py в функции store** добавить запросы для получения данных из таблицы (добавленные строки выделены цветом)

def store(request):  
 if request.method == **"POST"**:  
 product = StoreForm(request.POST)  
 if product.is\_valid():  
 product.save()  
 return HttpResponseRedirect(**"store"**)  
 storeForm = StoreForm()  
 storeProducts = Store.objects.all()  
 return render(request, **'store.html'**, context={**'storeForm'**: storeForm, **'store'**: storeProducts})

**4. На странице store.html добавить код** для вывода данных переменных в таблицы

{% extends 'base.html' %}  
{% block header %}  
<h1>Наличие товаров на складе</h1>  
{% endblock header %}  
{% block content %}  
<h2>Добавление данных</h2>  
<form method**="post"**>  
 {% csrf\_token %}  
 <table>  
 {{storeForm}}  
 </table>  
 <input class**="btn"** type**="submit"** value**="Добавить"**>  
</form>  
<h2>Наличие на складе</h2>  
<table class**="table"**>  
 <tr>  
 <th>Номер</th>  
 <th>Название продукта</th>  
 <th>Компания поставщик</th>  
 <th>Дата</th>  
 <th>Количество в наличии</th>  
 <th>Цена товара</th>  
 </tr>  
 {% for p in store %}  
 <tr>  
 <td>{{p.id}}</td>  
 <td>{{p.product}}</td>  
 <td>{{p.provider}}</td>  
 <td>{{p.date}}</td>  
 <td>{{p.quantity}}</td>  
 <td>{{p.price}}</td>  
 </tr>  
 {% endfor %}  
</table>  
{% endblock content %}

5. Теперь данные, отправленные через формы должны отображаться в таблицах. Далее на рисунках приведены примеры таблиц:

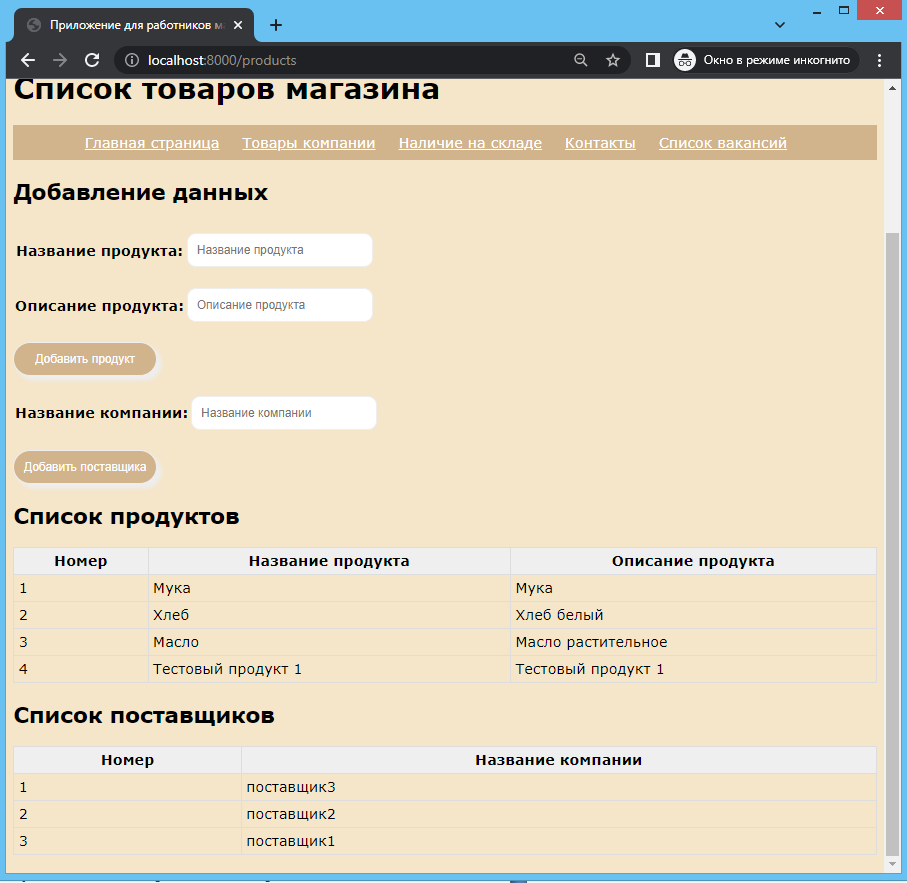


Рисунок. Пример – страница продукты

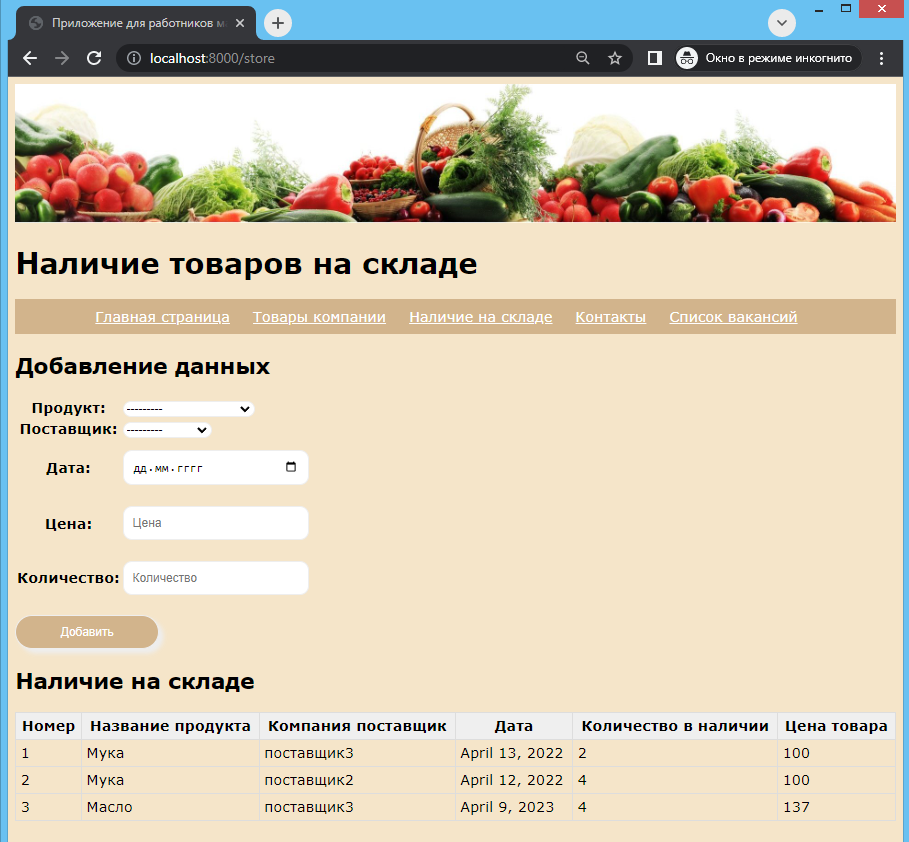


Рисунок. Пример – страница склада