



Ссылки на документацию

https://docs.djangoproject.com/en/3.1/topics/pagination/ - пагинация
https://django-filter.readthedocs.io/en/stable/ - фильтры
https://docs.djangoproject.com/en/3.1/topics/class-based-views/generic-display/ - дженерики
https://docs.djangoproject.com/en/3.1/topics/forms/ - формы

Пагинация

from django.core.paginator import Paginator

<code>paginate_by = <число></code>	Указывает количество элементов на странице
<code>model = <Класс модели></code>	Указывает на модель которую надо выводить
Номер страницы	Передаётся в адресной строке гет параметрами например <code>?page=1</code>
<code>Paginator.has_next()</code>	Проверить можно ли перейти на следующую страницу

Fullstack разработчик на Python

<code>Paginator.has_previous()</code>	Проверить есть ли возможность перейти на следующую страницу
<code>Paginator.range()</code>	Возвращает диапазон страниц

Фильтры

Установить фильтры	<code>pip install django-filter</code>
Включить фильтры в приложение в настройках	<code>INSTALLED_APPS = [... 'django_filters',]</code>
Определить в фильтре класс мета с полями	<pre>class SomeFilter(FilterSet): class Meta: model = <Модель> fields = <поля модели через запятую или в списке></pre>

Модуль Фильтры и формы

Формы через html

Написание формы через html	<code><form method="<Метод">"> <Элементы формы через input> </form></code>
Передаются	Во все view попадают через <code>request.POST</code> в виде словаря. Чтобы параметры попали в этот словарь надо указать атрибут <code>"name"</code> у html инпутов.

Формы через *django.forms*

Form	Примитивная форма, которую можно собрать самому с помощью полей из модуля форм, например <code>django.forms.CharField</code> , <code>IntegerField</code> и т.д. (полный список полей доступен в документации в начале шпаргалка)
ModelForm	Позволяет составить форму автоматически на основе модели.



SKILLFACTORY

Fullstack разработчик на Python

Модуль Фильтры и формы

Включить форму в дженерики	Чтобы это сделать надо добавить атрибут <code>form_class</code> и передать в него класс формы. Форма будет доступна автоматически в html шаблоне как <code>{{ form }}</code>
----------------------------	--

Дженерики для основных операций

DetailView	Позволяет считывать информацию о конкретном объекте. Для работы. Для работы необходимо указать <code>queryset</code> и имя шаблона
CreateView	Позволяет создавать объект. Для работы, прописать <code>form_class</code> и <code>template_name</code>
UpdateView	Позволяет редактировать объект. Почти полностью идентичен CreateView, но для его работы надо переопределить метод <code>get_object</code>

DeleteView	Позволяет удалять объекты. Для работы нужно указать имя шаблона и в самом шаблоне прописать форму с POST запросом для подтверждения
------------	---

Ссылка на репозиторий с туториалом

https://github.com/greedWizard/23_django_filters-forms