

## Задание 1. Сохранение и вывод изображений низкоуровневыми средствами

**Цель:** отработать работу с `request.FILES` и модулем `os`.

1. Создайте форму для загрузки изображений, включающую поле выбора файла и поле с текстовым описанием.
  2. Реализуйте представление (`view`), которое принимает файл через `request.FILES`, сохраняет его низкоуровневыми средствами:
    - Имя файла формировать на основе временной метки и оригинального расширения.
    - Сохранять в папку `files` в корне проекта.
  3. Напишите отдельное представление, которое:
    - Сканирует папку `files` с помощью `os.scandir()`.
    - Формирует список всех загруженных файлов с их описаниями.
    - Для каждого файла выводит ссылку на его просмотр.
  4. Реализуйте представление, которое по имени файла из URL возвращает этот файл пользователю (с помощью `FileResponse`).
  5. Добавьте шаблон для вывода списка файлов с миниатюрами (миниатюры пока обычные `<img>` с уменьшенными размерами через HTML).
- 

## Задание 2. Использование `django-cleanup` и `easy-thumbnails`

**Цель:** закрепить навыки работы с моделями и внешними библиотеками.

1. Установите и подключите библиотеки `django-cleanup` и `easy-thumbnails`.
2. Создайте модель `Photo` с полями:
  - `image` — `ThumbnailerImageField`, с параметром `resize_source={'size': (400, 300), 'crop': 'scale'}`.
  - `description` — `TextField`.
3. Создайте форму `PhotoForm` на основе этой модели.
4. Реализуйте представление, позволяющее:
  - Загружать изображения и описание через форму.
  - Сохранять их в базу данных.
  - Выводить на главной странице список всех загруженных изображений:
    - Оригинал изображения.
    - Миниатюру с размером 100x100 пикселей, созданную с помощью `easy-thumbnails` (`thumbnail` шаблонный тег).
5. Убедитесь, что при удалении записи соответствующий файл и миниатюры автоматически удаляются (`django-cleanup`).
6. Дополнительно: добавьте в шаблоне кнопку «Удалить» рядом с каждой записью.