**Пошаговый пример решения задания**

Рассмотрим пример решения задания 1 практического занятия.

1. На основе материалов лекции следует, что для хранения изображений поста необходимо:

a. Добавить в модель записи блога поле для картинки (см. рисунок 1):

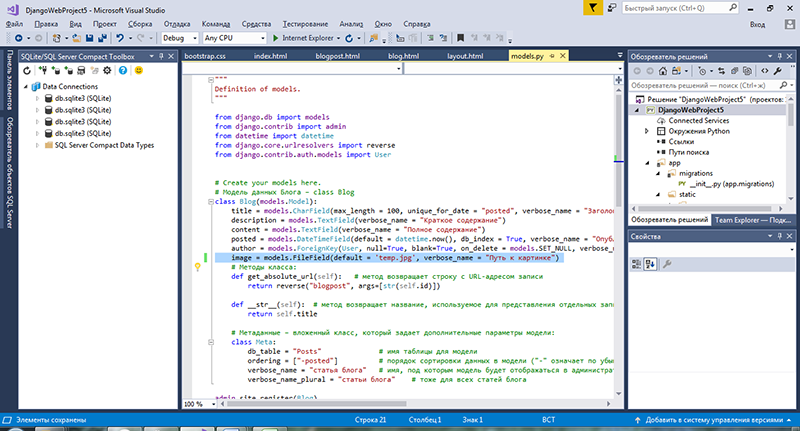


Рисунок 1 – Добавление в модель записи блога поле для картинки

b. Выполните миграцию для обновления базы данных:

* В меню **Проект** выбрать команду **Миграция сборок Django**
* В меню **Проект** выбрать команду **Миграция Django...**

c. В файл settings.py добавить определение переменных для загрузки графических файлов в папку media (см. рисунок 2).

d. В том же файле в список context\_processors (в переменной TEMPLATES в поле OPTIONS) добавить строку для подключения обработчика загрузки файлов (см. рисунок 3).

e. В файл urls.py добавить импорт функций для настройки доступа к загруженным файлам (см. рисунок 4).

f. В конец файла urls.py добавить строки шаблонов URL для загрузки файлов в папку media, вложенной в папку проекта (см. рисунок 5).

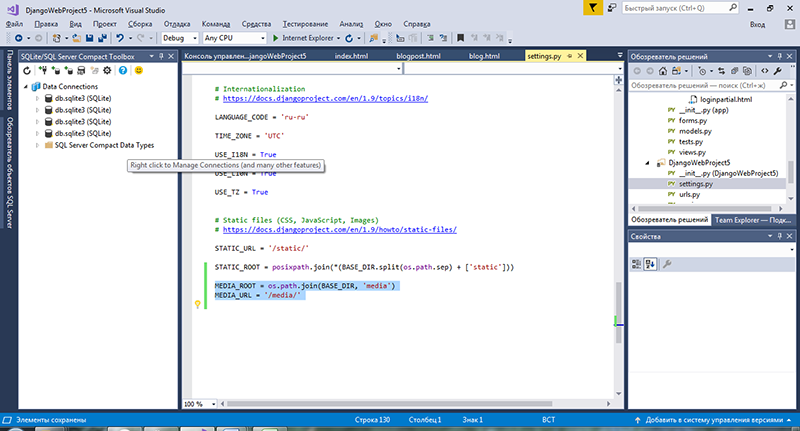
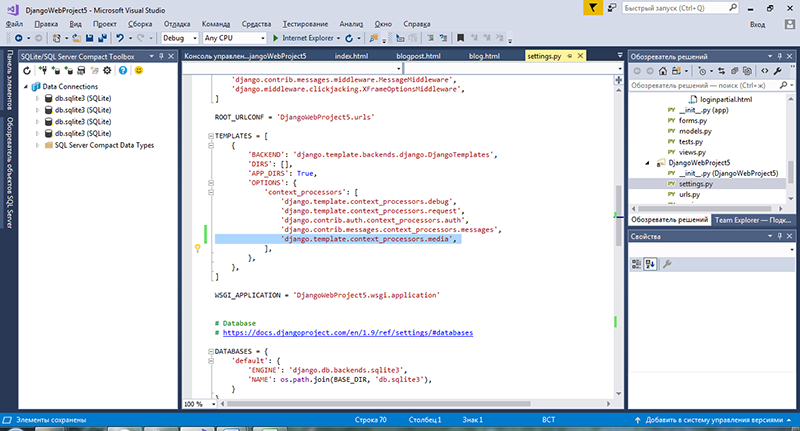


Рисунок 2 – Добавление в файл settings.py определение переменных для загрузки файлов в папку media

  
Рисунок 3 – Добавление строки для подключения обработчика загрузки файлов

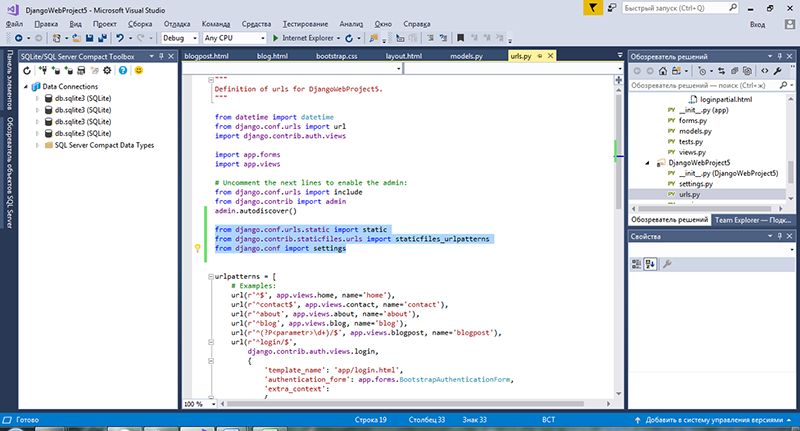


Рисунок 4 – Добавление в файл urls.py строк импорта функций для настройки доступа к загруженным файлам

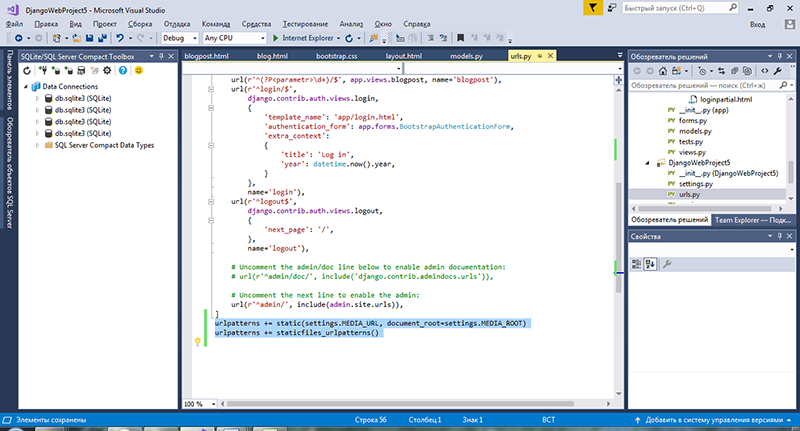


Рисунок 5 – Добавление строк шаблонов URL для загрузки файлов в папку media, вложенной в папку проекта

2. На основе материалов лекции и из теории в подразделе **Теория для практического занятия** следует, что для реализации добавления статей блога администратором на отдельной веб-странице сайта, необходимо:

a. Добавить в начало файла forms.py строку импорта модели блога (см. рисунок 6):

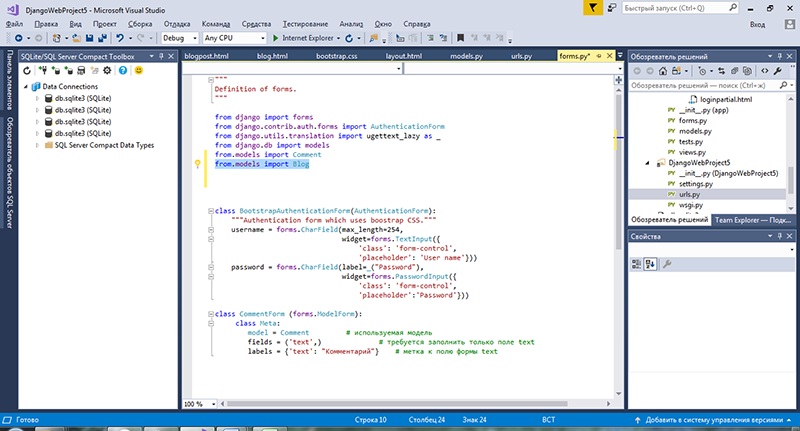


Рисунок 6 – Добавление в начало файла forms.py строку импорта модели блога

b. Добавить в конец файла forms.py код формы ввода статьи блога. Форма должна содержать поля для ввода данных статьи блога, например, для ввода заголовка, краткого содержания, полного содержания статьи, выбора файла картинки (см. рисунок 7).

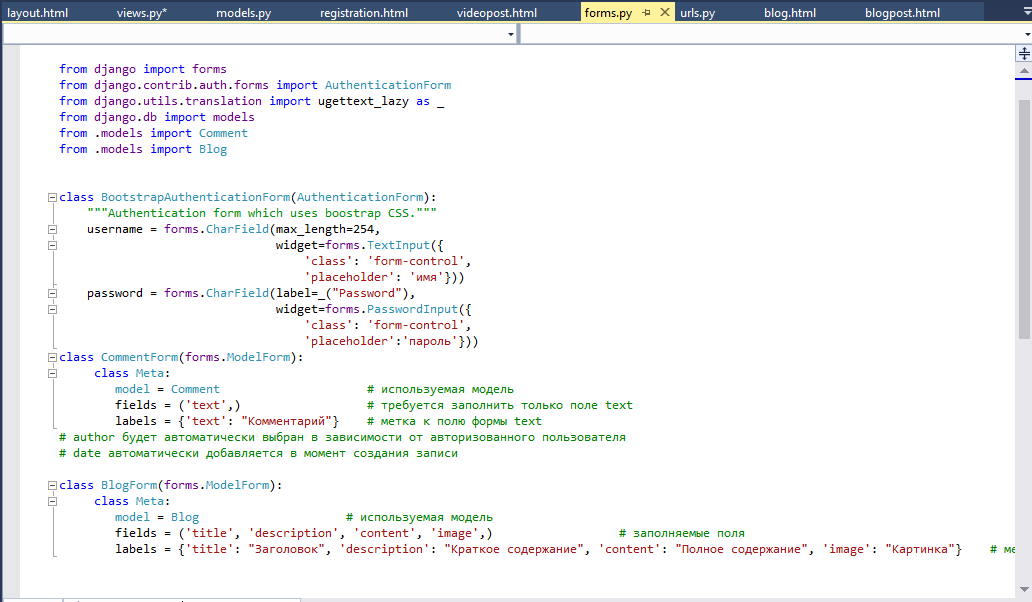


Рисунок 7 – Добавление в forms.py кода формы ввода статьи блога

c. Добавить в проект шаблон для html-файла для ввода статьи блога (например, newpost.html).

d. Добавить код для отображения формы ввода статьи блога в newpost.html. В тег формы добавить атрибут enctype="multipart/form-data" (см. рисунок 8).

e. Добавить в начало файла views.py код импорта формы для ввода статьи блога.

f. В файл views.py добавить код метода действия контроллера def newpost(request): (см. рисунок 9).

g. Добавить строку шаблона URL в файл urls.py для страницы добавления статьи блога (см. рисунок 10).

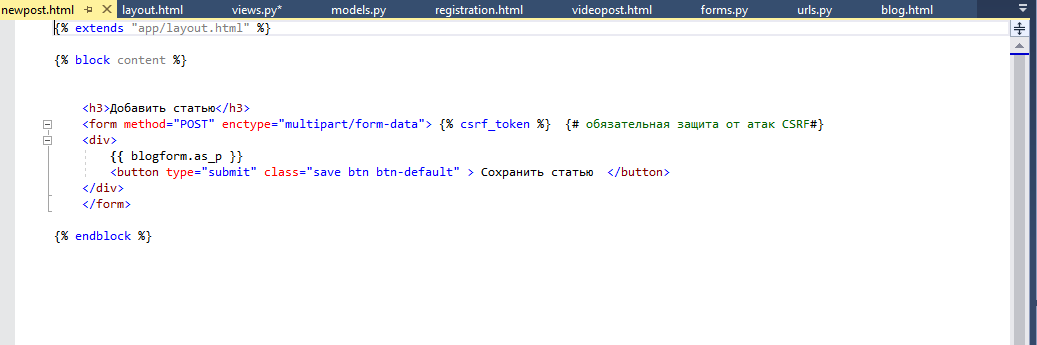


Рисунок 8 – Добавление кода для отображения формы ввода статьи блога в newpost.html

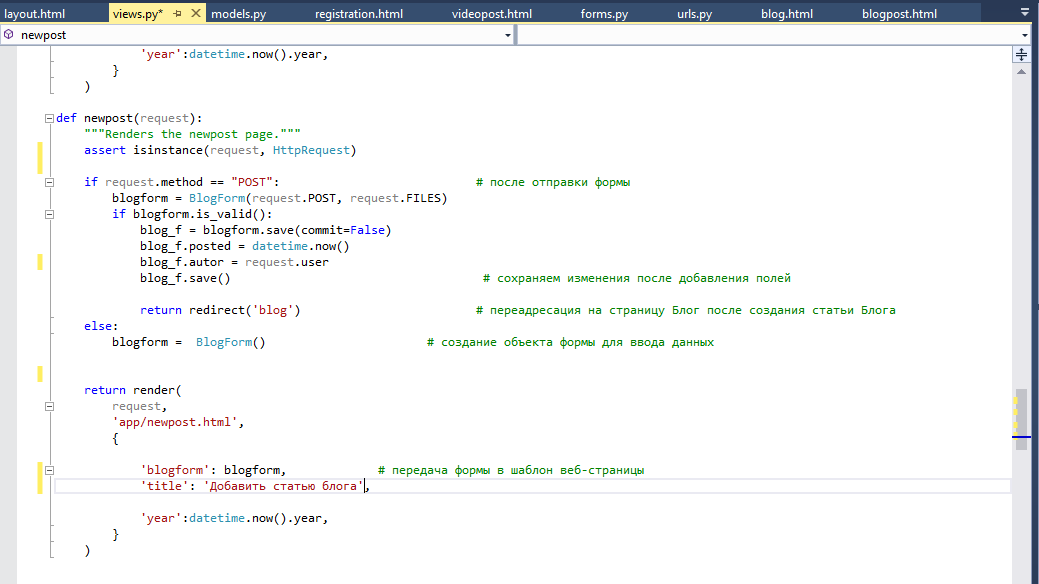


Рисунок 9 – Добавление кода метода действия контроллера def newpost(request):

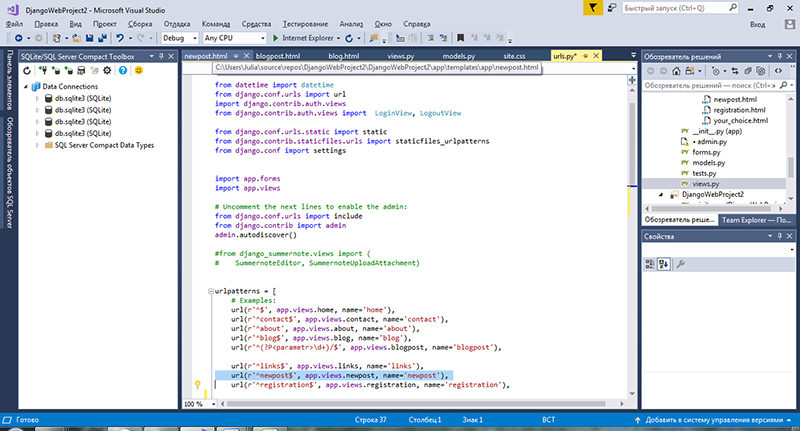


Рисунок 10 – Добавление строки шаблона URL в файл urls.py для страницы добавления статьи блога

h. Добавить пункт в меню сайта для страницы добавления статьи блога, доступный только для администратора – метод is\_superuser (см. рисунок 11).

i. В файл blogpost.html добавить вывод картинки записи блога перед содержанием статьи (см. рисунок 12).

j. Запустить проект на выполнение. Войти на сайт под администратором. Добавить статью блога с картинкой, добавить к ней комментарий. Проверить отображение новой статьи и комментария.

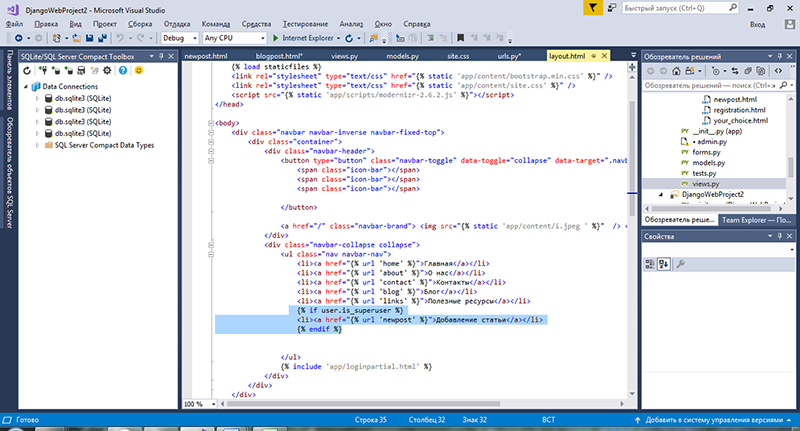


Рисунок 11 – Добавление пункта в меню сайта для страницы добавления статьи блога

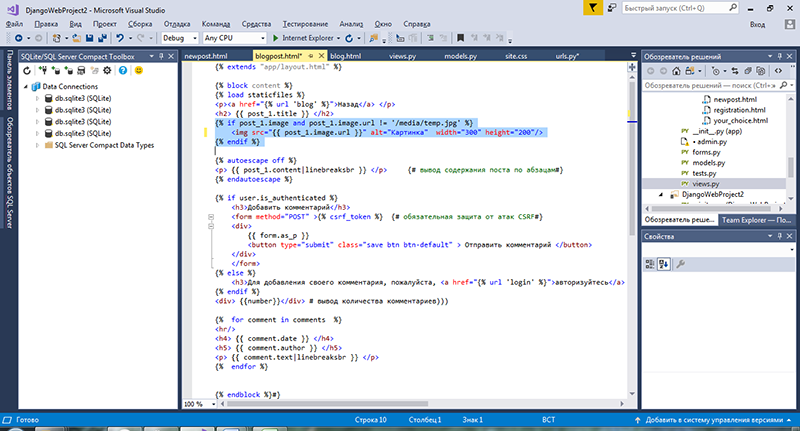


Рисунок 12 – Добавление вывода картинки записи блога перед содержанием статьи

3. На основе материалов лекции и из теории в подразделе **Теория для практического занятия** следует, что для реализации на отдельной веб-странице сайта (например, «Видео») отображение и воспроизведение двух видео-фрагментов формата .mp4 и длительностью не более 1 минуты необходимо:

a. Первоначально подготовить (или найти) 2 видео-фрагмента по тематике сайта формата .mp4, длительностью не более 1 минуты и картинки к ним для заставки (poster).

b. Скопировать подготовленные 2 видео-файла и картинки к ним (poster) в папку *media* веб-проекта.

c. Добавить пункт в меню сайта для страницы Видео (см. рисунок 13):

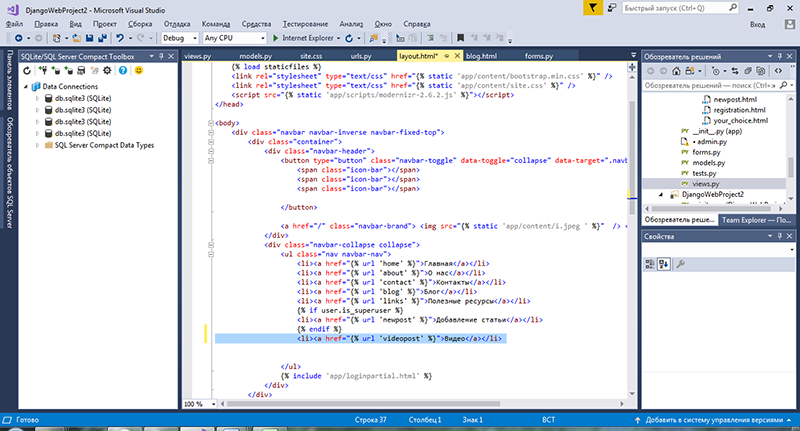


Рисунок 13 – Добавление пункта в меню сайта для страницы Видео

d. Добавить строку шаблона URL в файл urls.py для страницы Видео (см. рисунок 14).

e. Добавить в файл views.py код метода действия контроллера def videopost(request): (по аналогии с методом about).

f. Добавить в проект шаблон для html-файла для видео-контента (например, videopost.html).

g. Добавить код для отображения и воспроизведения видео-роликов в videopost.html (см. рисунок 15).

h. Запустить проект на выполнение. Проверить отображение и воспроизведение видео-роликов на странице сайта.

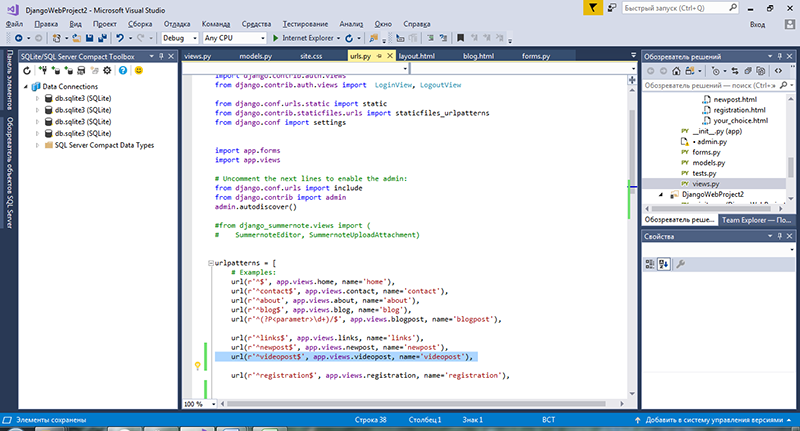


Рисунок 14 – Добавление строки шаблона URL в файл urls.py для страницы Видео

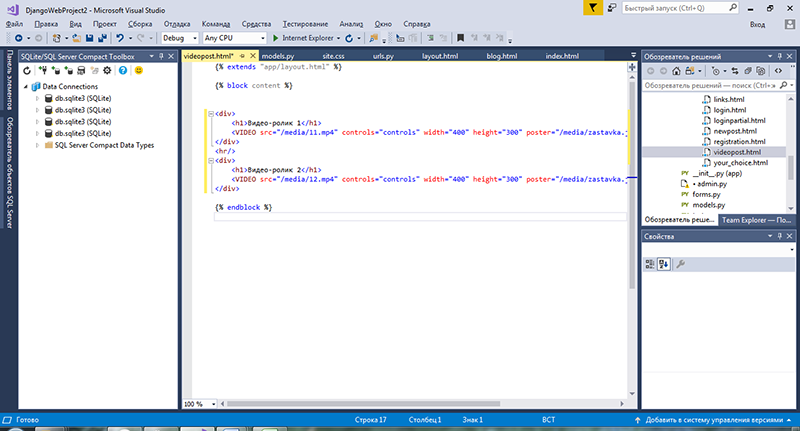


Рисунок 15 – Добавление кода для отображения и воспроизведения 2 видео-роликов в videopost.html