**Методические указания по выполнению практического задания**

**Целью** занятия является практическое освоение теоретических знаний, полученных в процессе изучения темы 7 «Использование баз данных в Django. SQLite. Модели в Django».

**Задачи**, реализуемые на практическом занятии:

* Закрепление представления о понятиях, введенных на лекции.
* Получение практических навыков создания модели Django и построения базы данных веб-проекта на её основе в среде Visual Studio 2017.

**Методические указания по выполнению практического задания**

1. Прежде чем приступить к выполнению задания 1, необходимо изучить конспект и глоссарий к лекции.
2. Для закрепления представления о понятиях, введенных на лекции, найдите в конспекте к лекции понятия Django ORM, класс QuerySet. По материалам лекции изучите структуру типичной модели Django, найдите команды, необходимые для создания таблицы базы данных по модели.
3. Для получения практических навыков создания модели Django и построения базы данных веб-проекта на её основе в среде Visual Studio 2017:
   * в конспекте найдите описание типов полей модели, перечень атрибутов полей модели;
   * прочитайте теорию для практического занятия;
   * выполните [задание 1](https://courses.openedu.ru/courses/course-v1:spbstu+WEBPYT+spring_2021/courseware/70d1547fcd5247398fc504824da170ae/61777faa95d848d48b43282f93a49c60/?child=first#yakor_1).

**Теория для практического занятия**

Для того, чтобы добавить возможность редактирования и создания статьи блога в административном разделе, необходимо:

1. В начало файла **models.py** необходимо добавить строку импорта модуля административного раздела: *from django.contrib import admin*
2. В файл **models.py** после объявления модели необходимо добавить строку регистрации модели блога (*class* *Blog*) в административном разделе: *admin.site.register(Blog)*

Для того, чтобы добавить в поле даты модели значение по умолчанию равное текущей дате, необходимо:

1. В начало файла **models.py** необходимо добавить строку импорта класса datetime из модуля datetime: *from datetime import datetime*
2. В поле даты добавить значение по умолчанию равное текущей дате: *posted = models.DateTimeField(default = datetime.now(), db\_index = True, verbose\_name = "Опубликована")*

При добавлении метода **get\_absolute\_url** в класс модели необходимо:

1. В начало файла **models.py** добавить строку импорта метода **reverse**: *from django.core.urlresolvers import reverse*

Для того, чтобы добавить статью блога в административном разделе и сохранить её в базе данных веб-проекта db.sqlite3, необходимо:

1. Запустить веб-проект на выполнение.
2. Войти в административный раздел (см. тему 6).
3. В строке «Статьи блога» нажать кнопку «**+Добавить**».
4. В загруженной форме добавления статьи блога необходимо ввести **Заголовок** статьи блога, **Краткое содержание** статьи, **Полное содержание** статьи, **Дату публикации** статьи (поле **Опубликована**).
5. Нажать кнопку «**Сохранить**». Статья добавляется в список статей и может быть **отредактирована при нажатии на гиперссылку названия статьи**.
6. Добавить следующую новую статью блога можно нажав на кнопку  и выполнить пункты 4 и 5.
7. Выйти из административного раздела.

**Задание 1**

Цель задания – получить практические навыки создания модели Django и построения базы данных веб-проекта на её основе в среде Visual Studio 2017.

1. Добавить в веб-проект Django модель для статей блога. Данные статьи блога должны добавляться в административном разделе. Каждая статья блога должна иметь заголовок, краткое содержание, полное содержание и дату публикации.
2. Создать таблицу в БД веб-проекта db.sqlite3 для сохранения статей блога.
3. Добавить в административном разделе три тестовые статьи блога (на определенную тему) и сохранить в базе данных веб-проекта db.sqlite3.