**Московский государственный технический университет**

**им. Н.Э. Баумана**

**(МГТУ им. Н.Э. Баумана)**

**Радиотехнический факультет (РТ)**

Отчёт по лабораторной работе № 5

«Обработка данных с использованием Django ORM»

|  |  |
| --- | --- |
| Проверила: | Выполнил: |
| Преподаватель кафедры ИУ-5 | студент группы РТ5-51Б |
| Гапанюк Ю.Е. | Гудилин Д.С. |
| Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_ | Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_ г. | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_ г. |

Москва, 2019

**1.** **Задание**

ВэтойлабораторнойработевыпознакомитесьспопулярнойСУБДMySQL, создадите свою базу данных. Также вам нужно будет дополнить свои классы предметной области, связав их с созданной базой. После этого вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей и ClassBasedViews.

Для сдачи вы должны иметь:

1. Скрипт с подключением к БД и несколькими запросами.

2. Набор классов вашей предметной области с привязкой к СУБД (класс должен уметь хотя бы получать нужные записи из БД и преобразовывать их в объекты этого класса)

3. Модели вашей предметной области

4. View для отображения списка ваших сущностей

**2.** **Выполнение** **задания** **и** **исходный** **код** **проекта** **Работа** **с** **MySQL:**

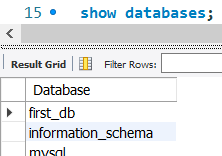
1) Создание пользователя

CREATE USER ' dbuser '@'localhost' IDENTIFIED BY '123';

2) Создание и использование базы данных

CREATE DATABASE mydb

USE mydb;

3) Привилегии пользователю:

GRANT ALL PRIVILEGES ON mydb.\* TO ' dbuser '@'localhost';

4) Добавление базы данных в настройки

|  |
| --- |
| DATABASES = { |
|  | 'default': { |
|  | 'ENGINE': 'django.db.backends.mysql', |
|  | 'NAME': "mydb", |
|  | 'USER': "dbuser", |
|  | 'PASSWORD': "SECRET", |
|  | 'HOST': "localhost", |
|  | 'PORT': 3306, |
|  | 'OPTIONS': {'charset': 'utf8'}, |
|  | 'TEST\_CHARSET': 'utf8', |
|  | } |
|  | } |

5) Создание модели:

|  |
| --- |
| class Book(models.Model): |
|  | STATUS\_CHOICES = ( |
|  | ('o', 'Opened'), |
|  | ('c', 'Closed'), |
|  | ) |
|  | id = models.AutoField(auto\_created=True, primary\_key=True, serialize=False, verbose\_name='ID') |
|  | # Название |
|  | title = models.CharField(max\_length=250) |
|  | # Для URL |
|  | slug = models.SlugField(max\_length=250, unique\_for\_date='publish') |
|  | # Автор |
|  | author = models.CharField(max\_length=250) |
|  | # Описание |
|  | desc = models.TextField() |
|  | # Дата публикации |
|  | publish = models.DateTimeField(default=timezone.now) |
|  | # Дата выставления на форум |
|  | created = models.DateTimeField(auto\_now\_add=True) |
|  | # Последний отзыв |
|  | updated = models.DateTimeField(auto\_now=True) |
|  | # Статус обсуждения: закрыт/Открыт |
|  | status = models.CharField(max\_length=10, choices=STATUS\_CHOICES, default='draft') |
|  | # Картинка |
|  | image = models.ImageField(upload\_to='books/%Y/%m/%d', blank=True) |

6) Работа с данной моделью:

|  |
| --- |
| class BookListView(LoginRequiredMixin, AjaxListView): |
|  | # Контекстная переменная (на странице) |
|  | context\_object\_name = 'books' |
|  | # Название шаблона |
|  | template\_name = 'list.html' |
|  | page\_template = 'entry\_list\_page.html' |
|  | def get\_queryset(self): |
|  | return Book.objects.all() |

7)View для вывода информации

|  |
| --- |
| <div class="wrapper"> |
|  | <div class="kartinka"> |
|  | <img src="{{book.image.url}}" alt="{{ book.title }}" width="100" height="100"> |
|  | </div> |
|  | <div class="info" style="padding-left:10px"> |
|  | <h2> |
|  | <a href="{{ book.id }}">{{ book.title }}</a> |
|  | </h2> |
|  | <p class="author"> |
|  | Автор {{ book.author }} |
|  | </p> |
|  | <p class="date"> |
|  | Добавлена {{ book.publish }} |
|  | </p> |
|  | {{ book.body|truncatewords:30|linebreaks }} |
|  | </div> |
|  | </div> |