Задание и порядок выполнения ЛР №5

В этой лабораторной работе вы познакомитесь с популярной СУБД MySQL, 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

Работа с MySQL:

```
mysql> CREATE USER 'dbuser'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE DATABASE `first_db` CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON first_db.* TO 'dbuser'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> use first_db;
Database changed
```

```
mysql> CREATE TABLE products (id INT NOT NULL AUTO_INCREMENT, name VARCHAR(30) NOT NULL, description VARCHAR(255)
NOT NULL, PRIMARY KEY(id));
Query OK, 0 rows affected (0.48 sec)

mysql> desc products
    -> ;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int(11)      | NO   | PRI | NULL    | auto_increment |
| name        | varchar(30)  | NO   |     | NULL    |                |
| description | varchar(255) | NO   |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
3 rows in set (0.34 sec)

mysql> INSERT INTO products VALUES(1, 'Book', 'Beautiful story about little boy Freddy and his friends');
Query OK, 1 row affected (0.36 sec)

mysql> SELECT * FROM products;
+----+------+-------------------------------------------------------+
| id | name | description                                           |
+----+------+-------------------------------------------------------+
|  1 | Book | Beautiful story about little boy Freddy and his friends |
+----+------+-------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

```
C:\Users\belun>python -m pip install pymysql
Collecting pymysql
  Downloading https://files.pythonhosted.org/packages/ed/39/15045ae46f2a123019aa
968dfcba0396c161c20f855f11dea6796bcaae95/PyMySQL-0.9.3-py2.py3-none-any.whl (47k
B)
    |                                | 51kB 656kB/s
Installing collected packages: pymysql
Successfully installed pymysql-0.9.3
```

Код программы

connection.py

```
import pymysql

pymysql.install_as_MySQLdb()


class Connection:

    def __init__(self, user, password, db, host='localhost'):
        self.host = host
        self.user = user
        self.password = password
        self.db = db
        self.charset = "utf8"
        self._connection = None

    @property
    def connection(self):
        return self._connection

    def __enter__(self):
        self.connect()

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.disconnect()

    def connect(self):
        if not self._connection:
            self._connection = pymysql.connect(
                host=self.host,
                user=self.user,
                password=self.password,
                db=self.db,
                charset=self.charset
            )

    def disconnect(self):
        if self._connection:
            self._connection.close()


class Product:
    def __init__(self, db_connection, name, description):
        self.db_connection = db_connection.connection
        self.name = name
        self.description = description

    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO 'products' ('name', 'description') VALUES (%s, %s);", (self.name,
self.description))

        self.db_connection.commit()
        c.close()


connection = Connection('dbuser', '123', 'first_db', 'localhost')
with connection:
    product = Product(connection, 'Computer', 'Electronic desktop device for working with files and
programming')
    product.save()
```

dbshow.py

```
import pymysql
pymysql.install_as_MySQLdb()


db = pymysql.connect (
     host="localhost",
     user="dbuser",
     passwd="123",
     db="first_db"
)

# курсор
c = db.cursor()

c.execute('SELECT * FROM products;')
entries = c.fetchall()
for e in entries:
    print(e)
c.close()
db.close()
```

views.py

```
from django.shortcuts import render

from django.http import HttpResponse
from django.views.generic import View, ListView

from lab5.models import Product, Review
import math


# Список продуктов
class ListProductView(ListView):
    model = Product
    template_name = 'product_list.html'
    context_object_name = 'products'
    paginate_by = 3

    def get(self, request, page=1):

        # Количество продуктов на странице
        elements_on_page = 9

        # Количество продуктов в строке
        elements_in_row = 3

        products = Product.objects.all()
        pages_count = math.ceil(len(products) / elements_on_page)

        start_index = (int(page) - 1) * elements_on_page
        end_index = start_index + elements_on_page
        products = products[start_index:end_index]

        index = 1
        rows = []
        row = []
        for product in products:
            row.append(product)

            if index == elements_in_row:
                rows.append(row)
                row = []
                index = 1
            else:
                index += 1
```

```
        if len(row) > 0:
            rows.append(row)

        return render(request, 'product_list.html', {"products": rows, "page": page, "pages_count":
pages_count})


# Страница с информацией о продукте и отзывами
class ProductView(View):

    def get(self, request, product_id):

        elements_in_row = 2
        product = Product.objects.get(id=product_id)
        reviews = Review.objects.filter(product_id=product_id)
        reviews_count = len(reviews)

        index = 1
        rows = []
        row = []
        for review in reviews:
            row.append(review)

            if index == elements_in_row:
                rows.append(row)
                row = []
                index = 1
            else:
                index += 1

        if len(row) > 0:
            rows.append(row)

        if len(rows) == 0:
            rows = None

        return render(request, 'product.html', {"product": product, "reviews": rows, "reviews_count":
reviews_count})
```

models.py

```
# coding=utf-8
from django.db import models
from django.contrib.auth.models import User


class Product(models.Model):

    # Название товара
    name = models.CharField(max_length=255)

    # Описание товара
    description = models.CharField(max_length=1000)

    # Продавец
    seller = models.CharField(max_length=255)

    # Короткое описание товара
    def short_description(self):
        return self.description[:126]

    def __str__(self):
        return ' '.join([
            self.name,
            ' from ',
            self.seller,
        ])
```

```python
class Review(models.Model):

    # Пользователь, который оставил отзыв
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    # Товар, под которым оставлен отзыв
    product = models.ForeignKey(Product, on_delete=models.CASCADE)

    # Текст отзыва
    description = models.CharField(
        max_length=500,
    )

    def __str__(self):
        return ' '.join([
            'review \'',
            str(self.description),
            ' \' from user @',
            str(self.user.username),
        ])
```

settings.py

```python
DATABASES = {
       'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'first_db',
        'USER': 'dbuser',
        'PASSWORD': '123',
        'HOST': 'localhost',
        'PORT': 3306,
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
```

urls.py

```python
from django.conf.urls import url
from django.contrib import admin

from lab5.views import ProductView, ListProductView

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^product/(?P<product_id>\d+)', ProductView.as_view()),
    url(r'^$', ListProductView.as_view(), name='base_template'),
]
```

Скриншоты выполнения

```
C:\Users\belun\Miniconda3\python.exe C:/Users/belun/Documents/РИП/Lab_5/lab5/lab5/dbshow.py
(1, 'Book', 'Beautiful story about little boy Freddy and his friends')
(2, 'Computer', 'Electronic desktop device for working with files and programming')
(3, 'Computer', 'Electronic desktop device for working with files and programming')
```