Московский государственный технический университет им. Н.Э. Баумана

Лабораторная работа №6 по дисциплине “Разработка интернет приложений”

Работа с СУБД

ИСПОЛНИТЕЛЬ:

студент группы РТ5-51
Чечелев Д.С.

"__"_____2017 г.

Москва - 2017

Файл Urls.py

```
21  urlpatterns = [
22      url(r'^function_view/', function_view),
23      url(r'^class_b_view/', ClassBased.as_view()),
24      url(r'^admin/', admin.site.urls),
25      url(r'^$', main_page, name='main'),
26      url(r'^order/(?P<id>\d+)', OrderView.as_view(), name='order_url'),
27      url(r'^tests/', TestView.as_view(), name='tests'),
28      url(r'^users/', UserView.as_view(), name='users'),
29  ]
```

Файл views.py

```
59  class TestView(ListView):
60      model = Test
61      context_object_name = 'tests'
62
63      def get_queryset(self):
64          q = super().get_queryset()
65          return q.none()
66
67  class UserView(View):
68
69      def get(self, request):
70          users = User.objects.all()
71          return render(request, 'users.html')
```

Models.py

```
1   # Create your models here.
2
3   from django.db import models
4
5
6   class User(models.Model):
7       first_name = models.CharField(max_length=30)
8       last_name = models.CharField(max_length=30)
9       age = models.IntegerField()
10      email = models.EmailField()
11
12      def __str__(self):
13          return self.first_name
14
15
16  class Test(models.Model):
17      first_name = models.CharField(max_length=30)
18      last_name = models.CharField(max_length=30)
19
20      def __str__(self):
21          return self.first_name
```

templates/test_list.html

```
{% extends 'base.html' %}

{% block title %}Студенты{% endblock %}

{% block header %}<h2 class="text-left">Список студентов делавших эту лабу:</h2>{% endblock %}

{% block body %}
    {% for test in tests %}
        <h4>{{ test.first_name }}</h4>
        <p>{{ test.last_name }}</p>
        <hr>
    {% endfor %}
{% endblock %}
```

Реализация скриптов для работы с БД

```
import MySQLdb


class Connection:
    def __init__(self, user, password, db, host='localhost'):
        self.user = user
        self.host = host
        self.password = password
        self.db = db
        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 = MySQLdb.connect(
                host = self.host,
                user = self.user,
                passwd = self.password,
                db = self.db,
                use_unicode=True,
                charset='utf8'
            )

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


class Test:
    def __init__(self, db_connection, first_name, last_name):
        self.db_connection = db_connection.connection
        self.name = first_name
        self.description = last_name

    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO lab5_app_test (first_name, last_name) VALUES (%s, %s);", (self.name, self.description))
        self.db_connection.commit()
        c.close()


class Tests:
    def __init__(self, db_connection):
        self.db_connection = db_connection.connection
```

```
    def select_all(self):
        c = self.db_connection.cursor()
        c.execute("SELECT * FROM lab5_app_test;")
        output = c.fetchall()
        c.close()
        return output

    def delete_all(self):
        c = self.db_connection.cursor()
        c.execute("TRUNCATE table lab5_app_test;")
        self.db_connection.commit()
        c.close()


con = Connection('root', '1111', 'mydb')
with con:
    test = Test(con, 'Денис', 'Чечелев')
    test.save()
    tests = Tests(con)#
    select_tests = tests.select_all()
    print(select_tests)
    print('DELETE? 1/0')
    input1 = input()
    if (input1 == '1'):
        tests.delete_all()
    print('----------------------')
    select_tests = tests.select_all()
    print(select_tests)
```

Результат работы: