

Защищено:

Гапанюк Ю.Е.

Демонстрация ЛР:

Гапанюк Ю.Е.

"__" _____ 2016 г.

"__" _____ 2016 г.

**Отчет по лабораторной работе № 6 по курсу
Разработка интернет приложений**

«Работа с СУБД»

Вариант № <20>

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-54

Савельева М. А.

(подпись)

"__" _____ 2016 г.

Задание лабораторной работы

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.

Текст программы

Файл connection.py:

```
import MySQLdb

class Connection:

    def __init__(self, user, password, db, host='localhost'):
        self.user = user
        self.db = db
        self.password = password
        self.host = host
        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,
                charset='utf8',
                use_unicode=True
            )

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

class Services:
    def __init__(self, db_connection, service_price, service_type,
service_class_of_price):
```

```

        self.db_connection = db_connection.connection
        self.service_price = service_price
        self.service_type = service_type
        self.service_class_of_price = service_class_of_price

    def save(self):
        c = self.db_connection.cursor()
        c.execute('INSERT INTO lab6_dj_servicemodel (service_price, service_type,
service_class_of_price) '
                'VALUES (%s, %s, %s)',
                (self.service_price, self.service_type,
self.service_class_of_price))
        self.db_connection.commit()
        c.close()

    def delete(self):
        c = self.db_connection.cursor()
        c.execute('DELETE FROM lab6_dj_servicemodel WHERE service_price=5000')
        self.db_connection.commit()
        c.close()

    def update(self):
        c = self.db_connection.cursor()
        c.execute('UPDATE lab6_dj_servicemodel SET service_price = 0 WHERE id = 3')
        self.db_connection.commit()
        c.close()

con = Connection('dbuser', '123', 'service')

with con:
    service=Services(con, '7000', 'Окна', 'Низкий класс')
    service.save()
    #service.delete()
    service.update()

```

Файл views.py:

```

from django.shortcuts import render
from django.views.generic import View
from LAB6_dj.models import ServiceModel

class ServiceView(View):
    def get(self, request):
        services = ServiceModel.objects.all()
        return render(request, 'service.html', {'services':services})

```

Файл models.py:

```

from django.db import models

class ClientModel(models.Model):
    client_first_name = models.CharField(max_length=30)
    client_email = models.EmailField(max_length=80)
    client_phone = models.CharField(max_length=40)
    client_address = models.CharField(max_length=100)
    order_number = models.ForeignKey('OrderModel', null=True)

class ServiceModel (models.Model):
    service_price = models.CharField(max_length=20)
    service_type = models.CharField(max_length=40)
    service_class_of_price = models.CharField(max_length=20)

class OrderModel (models.Model):
    order_number = models.IntegerField(unique=True)

```

```
order_data = models.DateField()
service = models.ForeignKey('ClientModel', null=True)
```

Изменения в файле settings.py:

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

Файл base.html:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>LAB_6</title>
    <link href="/static/bootstrap-3.3.5-dist/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
    <link href="/static/bootstrap-3.3.5-dist/css/jumbotron.css" rel="stylesheet"
type="text/css">
</head>
<body>
    <nav class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <p class="navbar-brand" >{% block title %}{% endblock %}</p>
        </div>
    </nav>
    <div class="jumbotron">
        <div class="container">
            <h1>{% block head %}{% endblock %}</h1>
        </div>
    </div>
    <div class="container">
        {% block body %}{% endblock %}
    </div>
</body>
</html>
```

Файл service.html:

```
{% extends 'base.html' %}
{% block title %}{% endblock %}
{% block visibletitle %}Услуги{% endblock %}
{% block head %}Услуги клининговой компании{% endblock %}
{% block body %}
    <table class="table table-striped">
        <thead>
            <tr>
                <th>Цена услуги (в рублях)</th>
                <th>Услуга</th>
                <th>Класс цены</th>
            </tr>
        </thead>
        <tbody>
            {% for el in services %}
                <tr><td>{{ el.service_price }}</td><td>{{ el.service_type }}</td><td>{{
```

```
el.service_class_of_price }}</td></tr>
  {% empty %}
    <tr><td colspan="4" align="center">Услуг нет</td></tr>
  {% endfor %}
</tbody>
</table>
{% endblock %}
```

Результаты выполнения программы:

Услуги клининговой компании

Цена услуги (в рублях)	Услуга	Класс цены
3000	Влажная уборка	Низкий класс
2000	Глажка белья	Низкий класс
7000	Окна	Низкий класс
7000	Окна	Низкий класс