

**Отчет по лабораторной работе № 5 по курсу
«Разработка интернет приложений»**

5
(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53
Янковский В.И.

ПРОВЕРИЛ:
Гапанюк Ю.Е.

(подпись)
" " _____ 201_ г

Лабораторная работа №6

Работа с СУБД

Задание и порядок выполнения

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

Реализованный проект:

Файл `udbconscript.py`:

```
import MySQLdb
import configparser

config = configparser.RawConfigParser()
config.read('config.ini')

class Event():
    id = 0
    name = ""
    address = ""
    time = ""
    desc = ""
    imageUrl = ""

    def get(self, connection, id):
        cur = connection.cursor()
        cur.execute("SELECT * FROM main_event WHERE id=%d" % (id,))
        res = cur.fetchall()[0]
        self.id = res[0]
        self.name = res[1]
        self.address = res[2]
        self.time = res[3]
        self.desc = res[4]
        self.imageUrl = res[5]
        cur.close()
        return self

    def __str__(self):
        return "Id=%d, Name=%s, time=%s, address=%s" \
            % (self.id, self.name, self.time, self.address)

if __name__ == "__main__":
    con = MySQLdb.connect(
        host=config["DATABASES"]["HOST"],
        user=config["DATABASES"]["USER"],
        passwd=config["DATABASES"]["PASSWORD"],
        db=config["DATABASES"]["NAME"]
    )

    ev = Event()
    print(ev.get(con, 5))
    con.close()
```

Файл `views.py`:

```
from django.shortcuts import render, get_object_or_404
from .models import Event
```

```
from django.views import View
from django.http import JsonResponse
```

```
class ObjectListView(View):
    def get(self, request, page_id):
        objects_on_list = 10
        page_id = int(page_id if page_id else 0)
        end = len(Event.objects.all())
        tmp = end - objects_on_list * page_id
        end = tmp if tmp > 0 else 0
        tmp = end - objects_on_list
        start = tmp if tmp > 0 else 0
        lst = Event.objects.all()[start:end:-1]
        return render(
            request,
            'index.html',
            context={
                'page': {'title': 'List'},
                'list': lst,
            }
        )
```

```
class ObjectView(View):
    def get(self, request, object_id):
        return render(
            request,
            'single_object.html',
            context={
                'object': get_object_or_404(Event, id=object_id),
            }
        )
```

Файл models.py:

```
from django.contrib.auth.models import User
from django.db import models
from django.utils import timezone
```

```
default_image_path = 'images/default.jpg'
```

```
class Event(models.Model):
    name = models.CharField(max_length=50)
    address = models.CharField(max_length=100)
    time = models.DateTimeField(default=timezone.now)
    desc = models.CharField(max_length=2000, null=True)
    imageUrl = models.CharField(
        default=default_image_path,
        max_length=256,
        null=False
```

```
)  
participation = models.ManyToManyField(User)  
  
def __str__(self):  
    return "Id=%d, Name=%s, time=%s, address=%s"\  
        % (self.id, self.name, self.time, self.address)
```

Результаты работы:

Аналогичны лабораторной №5 текущего курса.