Московский государственный технический университет

им. Н.Э.Баумана

Кафедра

«Системы обработки информации и управления»

(ИУ-5)

Отчёт по лабораторной работе №6

по курсу «Разработка интернет приложений»

Выполнил:

студент гр. ИУ5-53

Казаков Л.С.

Москва – 2017 г.

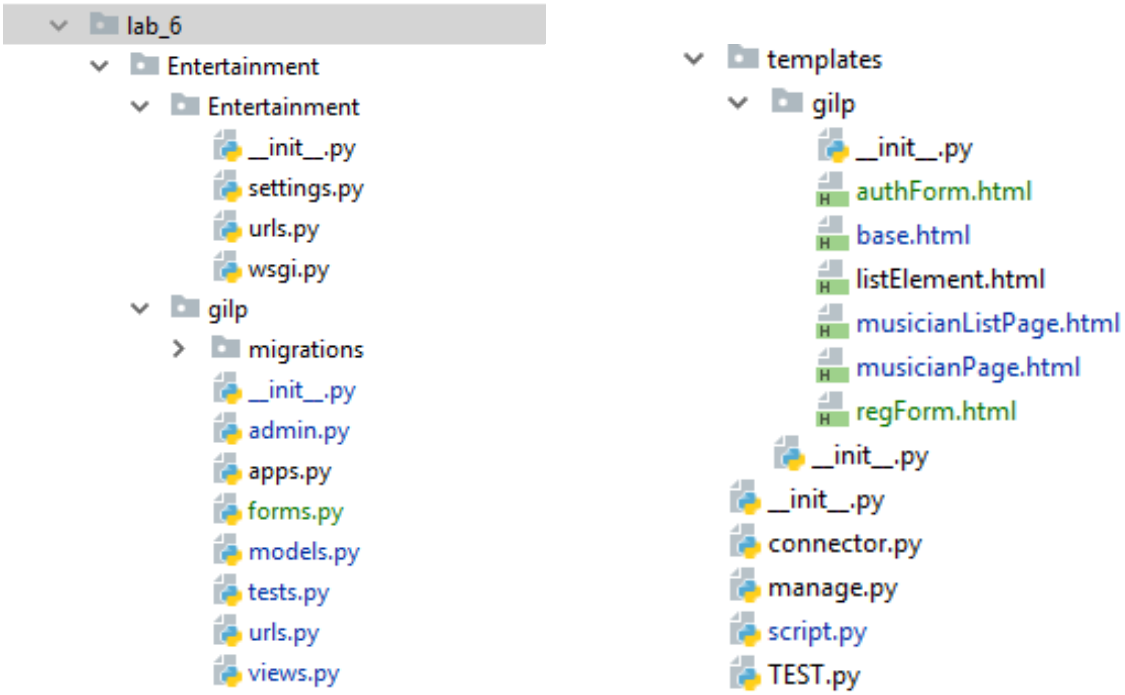
Задание и порядок выполнения

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

Структура проекта:



В файле settings.py добавили приложения:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'gilp',
    'bootstrap3'
]
```

Entertainment/urls.py:

```
from django.conf.urls import url, include
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^gilp/', include('gilp.urls')),
]
```

Entertainment/gilp/models.py:

```
from django.db import models

# Create your models here.


class MusicalGroup(models.Model):
    class Meta:
        db_table = 'musicalgroup'
    title = models.CharField(max_length=100, null=False)
    style = models.CharField(max_length=120, null=True)
    country = models.CharField(max_length=40, null=True)

    def __str__(self):
        return self.title


class Musician(models.Model):
    class Meta:
        db_table = 'musician'
    name = models.CharField(max_length=100, null=False)
    birth = models.DateField(null=True)
    role = models.CharField(max_length=70, null=True)
    group = models.ManyToManyField(MusicalGroup, through='Membership')

    def __str__(self):
        return self.name


class Membership(models.Model):
    class Meta:
        db_table = 'membership'
    musicalGroup = models.ForeignKey(MusicalGroup)
    musician = models.ForeignKey(Musician)
    entered = models.DateField(null=True)
    left = models.DateField(null=True)
```

Entertainment/gilp/urls.py:

```
from django.conf.urls import url

from . import views

app_name = 'gilp'

urlpatterns = [
    #url(r'^$', views.index, name='index'),
    url(r'^musicians/$', view=views.MusicianListView.as_view(),
name='musician_list_view'),
```

```
    url(r'^musicians/(?P<id>\d+)/', view=views.MusicianView.as_view(),
name='musician_view')
]
```

Entertainment/gilp/views.py:

```
from django.shortcuts import render
from django.views import View
from django.views.generic import ListView, DetailView
from .models import Musician, MusicalGroup, Membership
from django.http import HttpResponse, HttpRequest, HttpResponseRedirect
import MySQLdb
from .forms import *


# Create your views here.


class MusicianListView(ListView):
    model = Musician
    context_object_name = 'musicians'
    template_name = 'musicianListPage.html'


class MusicianView(View):
    def get(self, request, id):
        mus = Musician.objects.get(id=int(id))
        data = {
            'musician': {
                'id': mus.id,
                'name': mus.name,
                'birth': mus.name,
                'role': mus.role
            }
        }
        return render(request, 'musicianPage.html', {'musician': data})
```

Entertainment/templates/gilp/base.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
    <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    <div>
        {% block content %}{% endblock %}
    </div>
</body>
</html>
```

Entertainment/templates/gilp/musicianListPage.html:

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

{% block title %}List of Musicians{% endblock %}

{% block content %}
    <div class="container">
        <div class="row"><h1>Musicians</h1></div>
        <div class="row">
            <div class="col-xs-2"><b>Id</b></div>
            <div class="col-xs-2"><b>Name</b></div>
            <div class="col-xs-2"><b>Birth</b></div>
            <div class="col-xs-2"><b>Role</b></div>
        </div>
        <br>
        {% for person in musicians %}
        <div class="row">
            <div class="col-xs-2">{{ person.id }}</div>
            <div class="col-xs-2">{{ person.name }}</div>
            <div class="col-xs-2">{{ person.birth }}</div>
            <div class="col-xs-2">{{ person.role }}</div>
        </div>
        {% endfor %}
    </div>
{% endblock %}
```

Entertainment/templates/gilp/musicianPage.html:

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

{% block title %} Musician {% endblock %}

{% block content %}
    <div class="container">
        <h2>List Element</h2>
        <div class="container">
            <div class="row well">ID: {{ id }}</div>
            <div class="row well">Name: {{ name }}</div>
            <div class="row well">birth: {{ birth }}</div>
            <div class="row well">role: {{ role }}</div>
            <!--
            <div class="row">
                <a class = "btn btn-default" href="{% url 'gilp:musician_view'
%}">Back</a>
            </div>
            -->
        </div>
    </div>

{% endblock %}
```

Entertainment/script.py:

```
import MySQLdb
import datetime

class Connection:
    def __init__(self, user, passwd, db, host='localhost'):
        self.host = host
```

```
        self.user = user
        self.passwd = passwd
        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.passwd,
                db=self.db,
            )

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


class Musician:
    def __init__(self, db_connection, name=None, birth=None, role=None):
        self.db_connection = db_connection.connection
        self.name = name
        self.birth = birth
        self.role = role

    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO musician (name, birth, role) VALUES (%s, %s, %s);",
(self.name, self.birth, self.role))
        self.db_connection.commit()
        c.close()

    def edit(self, selected_name):
        c = self.db_connection.cursor()
        c.execute("UPDATE musician SET name = %s, birth = %s, role = %s WHERE name =
%s;",
                  (self.name, self.birth, self.role, selected_name))
        self.db_connection.commit()
        c.close()


new_connection = Connection('dbuser', '123', 'mydb', 'localhost')
choice = input('[add]/[edit]: ')
if choice == 'add':
    m_name = input('name: ')
    m_birth = input('birth (yyy-mm-dd): ')
    m_role = input('role: ')
    ans = input('Save [y/n]: ')
    if ans == 'y':
        with new_connection:
            musician = Musician(new_connection, m_name, m_birth, m_role)
            musician.save()
    else:
        pass
elif choice == 'edit':
    selected_name = input('musician name to edit: ')
```

```
    new_name = input('new name: ')
    new_birth = input('new birth (yyy-mm-dd): ')
    new_role = input('new role: ')
    ans = input('Save [y/n]: ')
    if ans == 'y':
        with new_connection:
            musician = Musician(new_connection, new_name, new_birth, new_role)
            musician.edit(selected_name)
    else:
        pass
```

Entertainment/conntector.py:

```
import MySQLdb

db = MySQLdb.connect(
    host='localhost',
    user='dbuser',
    passwd='123',
    db='mydb'
)
c = db.cursor()
db.commit()
c.execute('SELECT * FROM musician;')
entries = c.fetchall()
for item in entries:
    print(item)
c.close()
db.close()
```