



**Министерство образования и науки Российской Федерации Федеральное
государственное бюджетное образовательное учреждение высшего
образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА

СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

**Отчет по лабораторной работе № 5
«Работа с СУБД»
по курсу “Разработка интернет-приложений”**

Исполнитель:

Студент группы ИУ5-52

Губайдуллина Карина

_____ 29.11.2018

Москва, 2018

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:

- Скрипт с подключением к БД и несколькими запросами.
- 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)
- Модели вашей предметной области
- View для отображения списка ваших сущностей

Исходный код

urls.py

```
from django.contrib import admin from django.conf.urls
import url from django.views.generic import RedirectView
from testApp.views import PicturesView, ListPicturesView
from django.urls import path

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^favicon\.ico$',
        RedirectView.as_view(url='/static/images/favicon.ico')),
    url(r'^$', ListPicturesView.as_view(), name='pictures-list'),
    path('pictures/', PicturesView.as_view()), #
    path('pictures_list', ListPicturesView.as_view()),
    url(r'^pictures/(?P<pictures_id>\d+)', PicturesView.as_view())
]

models.py
from __future__ import unicode_literals from
django.db import models
```

Create your models here.

```
class
Pictures(models.Model):
    name = models.CharField(max_length=50,)
    description = models.CharField(max_length=1000,)
    place = models.CharField(max_length=30,)
    def
    __str__(self):
        return ' '.join([self.name, ' in ', self.place, ])
```

views.py

```
from __future__ import unicode_literals
from django.shortcuts import render from
django.http import HttpResponse
from django.views.generic import View, ListView
from testApp.models import Pictures
```

```

# Create your views here.
def
function_view(request):
    return HttpResponse('response from function view')
class
ListPicturesView(ListView):
    model = Pictures
    template_name = 'pictures_list.html'
def get(self, request):
    data = {
        'pictures': [
            {'name': 'Black square', 'description': 'It was painting by
Malevich',
            'place': 'Tretiakov gallery'},
            {'name': 'The starry night', 'description': 'It was painting
by van Gogh',
            'place': 'Museum of Modern Art'},
        ]
    }
    return render(request, 'pictures_list.html', data)
class PicturesView(View):
def get(self, request):
    data = {
        'pictures': [
            {'description': 'Many people go there everyday jast to see
something wonderful',
            'place': 'Museums'},
        ]
    }
    return render(request, 'pictures.html', data)

```

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.use_unicode = True
        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,
        use_unicode=self.use_unicode,
charset=self.charset
)
    def disconnect(self):
if self._connection:
        self._connection.close()
    class Picture:
        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 pictures (name, description) VALUES(%s, %s);",
(self.name, self.description))

        self.db_connection.commit()
        c.close()
        connection = Connection('root', '22121998', 'pic',
'localhost') with connection:
            picture = Picture(connection,
                                'The Birth of Venus',
                                'Painting that depicts the emergence of Goddess Venus
from the sea as a beautiful woman') picture.save()

```

base.html

```

{% load staticfiles %}

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta name="description" content="">
        <meta name="place" content="">

        <title>{% block title %}{% endblock %}</title>
        <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
        <style>

body {
    padding-top: 30px;
    background-color: lavender;
}
</style>
</head>

<body>
    <nav class="navbar navbar-inverse navbar-fixed-top">
        <div class="container-fluid">
            <div class="navbar-header">

```

```

        <button type="button" class="navbar-toggle collapsed"
data-toggle="collapse" data-target="#bs-example-navbar-collapse-1"
ariaexpanded="false">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <!--a class="navbar-brand" href="/">lab4</a-->
    </div>
    <div class="collapse navbar-collapse" id="bs-example-
navbarmacollapse-1">
        {% block navbar-links %}
        {% endblock %}
    </div>
</div>
</nav>

<div class="container">
    <div class="starter-template">
        {% block body %}
        {% endblock %}
    </div>
</div>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></scr
ipt>
    <script src="{% static 'js/bootstrap.js' %}"></script>
</body> </html>

```

pictures.html

```

{% extends 'base.html' %}

{% block title %} {% endblock %}

{% block navbar-links %}

    <ul class="nav navbar-nav">
        <li><a href="/">back to pictures</a></li>
    </ul>
{% endblock %}

{% block body %}
    <ul>
        {% for picture in pictures %}
            <h1>{{ pictures.name }}</h1>
        <div class="media">
            <div class="media-body">
                <div class="jumbotron">
                    <h2>{{ picture.place }}</h2>
                </div>
                <div class="jumbotron">
                    <p>{{ picture.description }}</p>
                </div>
                <p><a class="btn btn-lg btn-success"
role="button">Watch</a></p>
            </div>
        </div>
        {% endfor %}
    </ul>

```

```

</ul>

{% endblock %}
pictures_list.html
{% extends 'base.html' %}

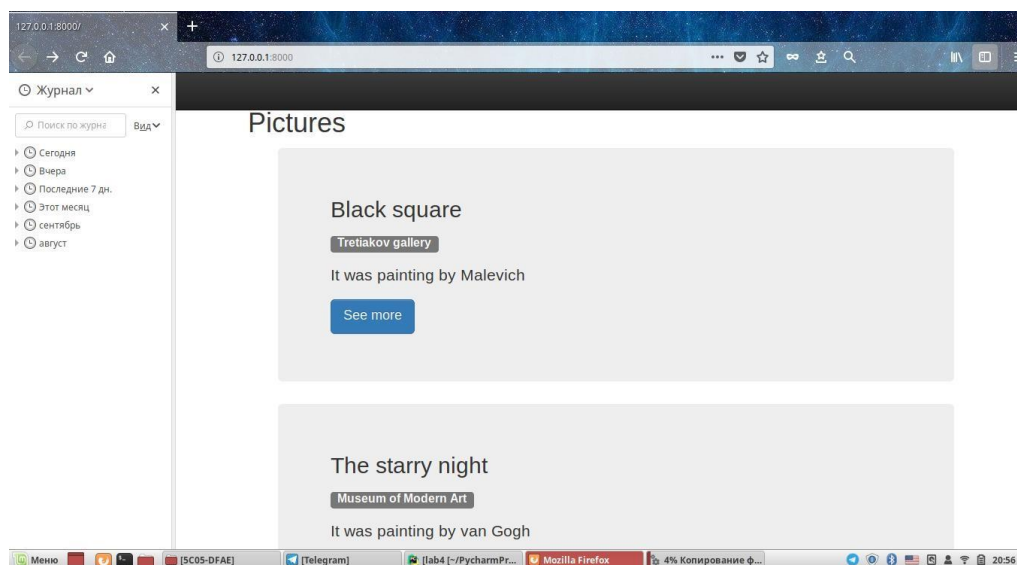
{% block title %} {% endblock %}

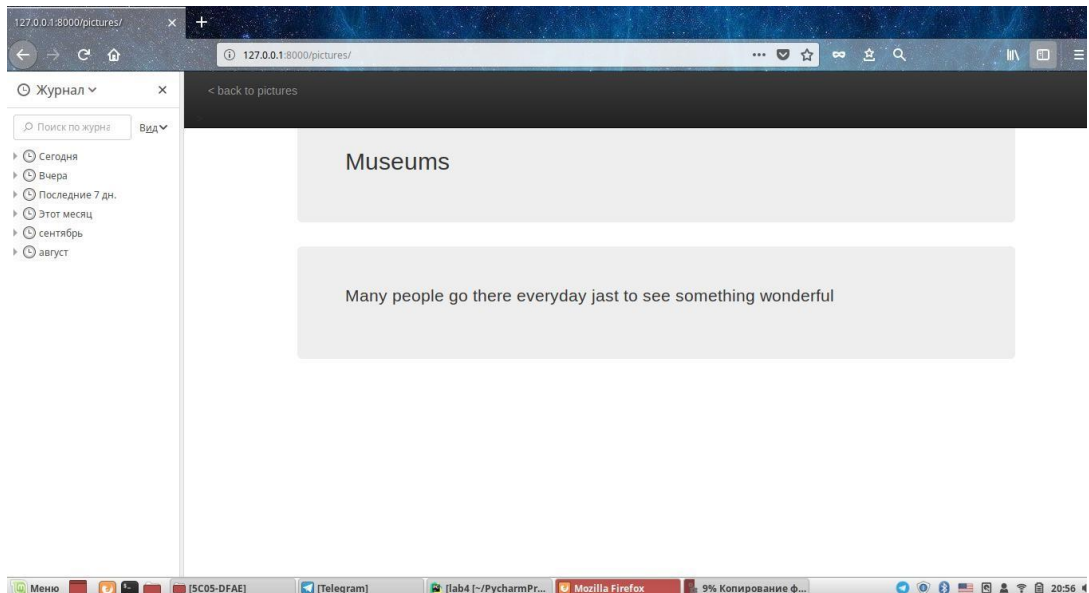
{% block navbar-links %} {% endblock %}

{% block body %}
<h1>Pictures</h1>
<ul>
{% for picture in pictures %}
<div class="jumbotron">
<div class="media-left media-bottom">
<a href="#">
<h1 class="label label-default" alt="{{ picture.name
}}"></h1>
</a>
</div>
<div class="media-body">
<h2>{{ picture.name }}</h2>
<p><span class="label label-default">{{ picture.place
}}</span></p>
<p>{{ picture.description }}</p>
<p><a class="btn btn-primary btn-lg" href="pictures/{{
picture.id }}" role="button">See more</a></p>
</div>
</div>
{% endfor %}
</ul>
{% endblock %}

```

Результаты выполнения





```
Администратор: Командная строка - mysql.exe -u root -p

mysql> select * from pictures;
+----+-----+-----+
| id | name  | description |
+----+-----+-----+
| 1  | Red Lenin | Andy Warhol's portrait of the Russian political leader |
| 2  | Mona Lisa | It is a half-length portrait painting by the Leonardo da Vinci |
| 3  | Mona Lisa | It is a half-length portrait painting by the Leonardo da Vinci |
| 4  | The Birth of Venus | Painting that depicts the emergence of Goddess Venus from the sea as a beautiful woman |
| 5  | The Birth of Venus | Painting that depicts the emergence of Goddess Venus from the sea as a beautiful woman |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql> delete from pictures where id = 3;
Query OK, 1 row affected (0.12 sec)

mysql> delete from pictures where id = 4;
Query OK, 1 row affected (0.01 sec)

mysql> select * from pictures;
+----+-----+-----+
| id | name  | description |
+----+-----+-----+
| 1  | Red Lenin | Andy Warhol's portrait of the Russian political leader |
| 2  | Mona Lisa | It is a half-length portrait painting by the Leonardo da Vinci |
| 5  | The Birth of Venus | Painting that depicts the emergence of Goddess Venus from the sea as a beautiful woman |
+----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```