

Защищено:

Гапанюк Ю. Е.

"__" _____ 2016 г.

Демонстрация:

Гапанюк Ю. Е.

"__" _____ 2016 г.

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

9

(количество листов)

ИСПОЛНИТЕЛЬ:

Студент группы ИУ5-54 _____

Повираева М. Л.

(подпись)

"__" _____ 2016 г.

Оглавление

Описание задания лабораторной работы.....	3
Реализация поставленной задачи	3
Листинг программы mysql_con.py	3
Листинг программы models.py	3
Листинг программы urls.py.....	4
Листинг программы views.py.....	4
Листинг программы settings.py	5
Листинг программы manage.py	7
Листинг программы base.html.....	7
Листинг программы lesson.html	8
Листинг программы lessons.html	8
Результаты работы программы.....	9

Описание задания лабораторной работы

Требуется создать:

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.

Реализация поставленной задачи

Листинг программы mysql_con.py

```
import MySQLdb

if __name__ == '__main__':
    db = MySQLdb.connect(
        host='localhost',
        user='mary',
        passwd='123mary123',
        db='lab6',
        charset='utf8'
    )

    cur = db.cursor()

    cur.execute("INSERT INTO lab6_lesson (name,description) VALUES (%s, %s);",
        ('Математика', 'Подготовим к ЕГЭ'))

    db.commit()

    cur.execute("SELECT * FROM lab6_lesson;")

    entries = cur.fetchall()

    for e in entries:
        print(e)

    cur.close()
    db.close()
```

Листинг программы models.py

```
from django.db import models

# Create your models here.

class Lesson(models.Model):
    name = models.CharField(max_length=30)
    description = models.CharField(max_length=255)
```

Листинг программы urls.py

"""Templating URL Configuration

*The `urlpatterns` list routes URLs to views. For more information please see:
<https://docs.djangoproject.com/en/1.9/topics/http/urls/>*

Examples:

Function views

- 1. Add an import: from my_app import views*
- 2. Add a URL to urlpatterns: url(r'^\$', views.home, name='home')*

Class-based views

- 1. Add an import: from other_app.views import Home*
- 2. Add a URL to urlpatterns: url(r'^\$', Home.as_view(), name='home')*

Including another URLconf

- 1. Add an import: from blog import urls as blog_urls*
- 2. Import the include() function: from django.conf.urls import url,*

include

- 3. Add a URL to urlpatterns: url(r'^blog/', include(blog_urls))*

"""

from django.conf.urls **import** url

from . **import** views

```
urlpatterns = [  
    url(r'^$', views.ExampleView.as_view()),  
    url(r'^lessons/', views.LessonsView.as_view()),  
    url(r'^lesson/(?P<id>\d+)', views.LessonView.as_view(),  
    name='lesson_url'),  
]
```

Листинг программы views.py

```
from django.shortcuts import render  
from django.http import HttpResponse  
from django.views import View  
from django.views.generic import ListView  
from django.views.generic import TemplateView
```

```
from lab6.models import Lesson
```

Create your views here.

```
class ExampleView(View):  
    def get(self, request):  
        return render(request, 'base.html')
```

```
class LessonsView(ListView):  
    model = Lesson  
    context_object_name = 'lessons'  
    template_name = 'lessons.html'  
  
    def get_queryset(self):  
        qs = Lesson.objects.all().order_by('id')  
        return qs
```

```
class LessonView(View):  
    def get(self, request, id):  
        data = Lesson.objects.get(id__exact=id)  
        return render(request, 'lesson.html', {'lesson':data})
```

Листинг программы settings.py

```
"""
Django settings for lab6mary project.

Generated by 'django-admin startproject' using Django 1.10.4.

For more information on this file, see
https://docs.djangoproject.com/en/1.10/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/1.10/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.10/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'je2h9qj!$@znf7h6igw8@ji8r7xq2ppz (@4&ur!!cnb$b!f(o+'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'lab6',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'lab6mary.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',

```

```

        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
]

WSGI_APPLICATION = 'lab6mary.wsgi.application'

# Database
# https://docs.djangoproject.com/en/1.10/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'lab6',
        'USER': 'mary',
        'PASSWORD': '123mary123',
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}

# Password validation
# https://docs.djangoproject.com/en/1.10/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
        'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
        'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
        'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
        'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/1.10/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

```

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.10/howto/static-files/

STATIC_URL = '/static/'
```

Листинг программы manage.py

```
#!/usr/bin/env python
import os
import sys

if __name__ == "__main__":
    os.environ.setdefault("DJANGO_SETTINGS_MODULE", "lab6mary.settings")
    try:
        from django.core.management import execute_from_command_line
    except ImportError:
        # The above import may fail for some other reason. Ensure that the
        # issue is really that Django is missing to avoid masking other
        # exceptions on Python 2.
        try:
            import django
        except ImportError:
            raise ImportError(
                "Couldn't import Django. Are you sure it's installed and "
                "available on your PYTHONPATH environment variable? Did you "
                "forget to activate a virtual environment?"
            )
        raise
    execute_from_command_line(sys.argv)
```

Листинг программы base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="{% static
'css/bootstrap.css' %}">
    <title>{% block title %}ТЕСТОВЫЙ ВЫВОД{% endblock %}</title>
</head>
<body>
<nav class="navbar navbar-default">
    <div class="container-fluid">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="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">ЛАБА #6</a>
        </div>

        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav">
                <li><a href="/lab6/">Базовый класс</a></li>
```

```

        <li><a href="/lab6/lessons/">Уроки</a></li>
    </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
<div>
    {% block body %}Redefined by spawnlings{% endblock %}
</div>
</body>
</html>

```

Листинг программы lesson.html

```

{% extends 'base.html' %}

{% block title %}Урок №{{ lesson.id }}{% endblock %}

{% block body %}
<h1>Урок №{{ lesson.id }} &ndash; {{ lesson.name }}</h1>
<p>{{ lesson.description }}</p>
{% endblock %}

```

Листинг программы lessons.html

```

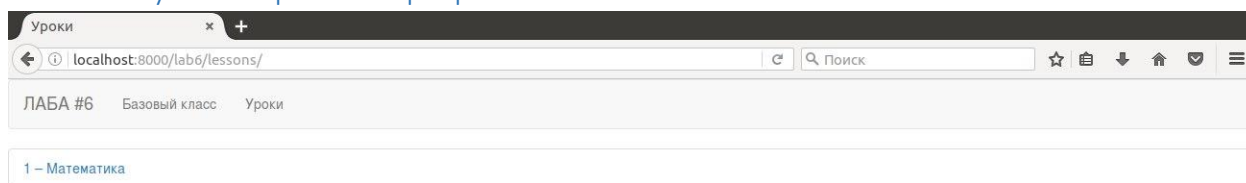
{% extends 'base.html' %}

{% block title %}Уроки{% endblock %}

{% block body %}
<ul class="list-group">
    {% for lesson in lessons %}
        <li class="list-group-item"><a href="{% url 'lesson_url'
lesson.id %}">{{ lesson.id }} &ndash; {{ lesson.name }}</a></li>
    {% empty %}
        <li class="list-group-item disabled">empty list</li>
    {% endfor %}
</ul>
{% endblock %}

```


Результаты работы программы



Урок №1 – Математика

Подготовим к ЕГЭ