

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «РИП»

Отчет по лабораторным работам №5 и №6

«Шаблонизация», «Работа с СУБД»

Выполнил:

студент группы ИУ5-53
Гатауллин И. И.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2017 г.

Задание и порядок выполнения ЛР №5

В этой ЛР вы создадите Django-проект, покажете пользователю статичную страницу, познакомитесь с конструкциями шаблонизаторов: переменные, теги, наследование шаблонов.

Создать проект

Реализовать view, в которых генерируются html-страницы

В шаблонах должны быть использованы рассмотренные конструкции:

переменные, вложенные значения, циклы, условия

Все шаблоны должны расширять базовый шаблон

Для элементов списка использовать тег include

По нажатии на элемент списка должна открываться страница информации об элементе

Для верстки необходимо использовать Bootstrap

Задание и порядок выполнения ЛР №6

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

Код программы

urls.py

```
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.MainPageView.as_view(), name='services_list'),
    url(r'^services/$', views.ServiceList.as_view()),
    url(r'^service/(?P<id>\d+)', views.ServicePageView.as_view(),
    name='service'),
]
```

base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link href="/static/css/bootstrap.css" rel="stylesheet">
    <title>{% block title %}{% endblock %}</title>
```

```

</head>
<body>
<div class="container-fluid">
  <div class="row">
    <div class="col-md-12">
      <h1><a href="/"> Services </a></h1>
    </div>
  </div>
  <div class="col-md-3">
    <h2>Menu</h2>
    <ul class="list-group">
      <li class="list-group-item">Services
        <span class="badge">{% block count %}{% endblock %}</span>
      </li>
      <li class="list-group-item">Services
        <span class="badge">25</span>
      </li>
      <li class="list-group-item">Popular services
        <span class="badge">2</span>
      </li>
      <li class="list-group-item">New Services
        <span class="badge">3</span>
      </li>
    </ul>
  </div>
  <div class="col-md-9">
    <div>{% block body %}{% endblock %}</div>
  </div>
</div>
</body>
</html>

```

index.html

```

{% extends 'services/base.html' %}

{% block title %}Services{% endblock %}
{% block body %}
  <h1>Добро пожаловать</h1>
  <ul>
    {% for service in top_services %}
      {% if service.id %}
        <li> <a href="{% url 'service' service.id %}">{{ service.title
      }}</a> </li>
      {% include 'services/service_short_info.html' with id=service.id %}
      {% endif %}
      {% empty %}
        <li>Нет услуг</li>
      {% endfor %}
    </ul>
  </div>
{% endblock %}

```

service.html

```

{% extends 'services/base.html' %}

{% block title %}Service №{{ service.id }}{% endblock %}
{% block body %}
  <h3>Информация об услуге № {{ service.id }}</h3>
{% endblock %}

```

service_short_info.html

```
<p>Краткая информация об услуге №{{ service.id }}</p>
```

services_list.html

```
{% extends 'services/base.html' %}
{% block count %}{{ services.count }}{% endblock %}
{% block body %}
    <h2>Services</h2>
    <ul>
        {% for service in services %}
            <li> <a href="{% url 'service' service.id %}"> {{ service.title }}
        {% endfor %}
    </ul>
{% endblock %}
```

admin.py

```
from django.contrib import admin

from .models import CustomUser, Service, Order

admin.site.register(CustomUser)
admin.site.register(Service)
admin.site.register(Order)
```

apps.py

```
from django.apps import AppConfig
```

```
class ServicesConfig(AppConfig):
    name = 'services'
```

models.py

```
from django.db import models
from django import forms
from datetime import date
from django.contrib.auth.models import User
from django.core.validators import MaxValueValidator, MinValueValidator
```

```
# Create your models here.
```

```
class CustomUser(User):
    pass
```

```
class Service(models.Model):
    title = models.CharField(max_length=254)
    category = models.CharField(max_length=254)
    company = models.CharField(max_length=254)
    price = models.DecimalField
```

```

class Meta:
    unique_together = ('title', 'category')

```

```

class Order(models.Model):
    user = models.ForeignKey(User)
    service = models.ForeignKey(Service)
    odate = models.DateField

```

views.py

```

from django.shortcuts import render
from django.views import View
from django.views.generic import ListView
from .models import Service

```

```

def services_list(request):
    return render(request, 'services/services_list.html', {})

```

```

class MainPageView(View):
    def get(self, request):
        data = {
            'top_services': [
                {'title': 'service №1', 'id': 1},
                {'title': 'service №2', 'id': 2},
                {'title': 'service №3', 'id': 3},
            ]
        }
        return render(request, 'services/index.html', data)

```

```

class ServicePageView(View):
    def get(self, request, id):
        data = {
            'service': {
                'id': id
            }
        }
        return render(request, 'services/service.html', data)

```

```

class ServiceList(ListView):
    model = Service
    template_name = 'services/services_list.html'
    context_object_name = 'services'

```

handle_db.py

```

import MySQLdb

```

open connection

```

db = MySQLdb.connect(
    host="localhost",
    user='Adm-User',
    passwd='222',
    db='services_db',
    charset="utf8",
    use_unicode=True
)

```

```

cursor = db.cursor()

```

```

cursor.execute('INSERT INTO User (name, age) VALUES (%s, %s);', ('Рылёва',
int(20)))
db.commit()

cursor.execute('SELECT * FROM User ')
entries = cursor.fetchall()

for e in entries:
    print(e)

cursor.execute('DELETE FROM User')
db.commit()

cursor.execute('SELECT * FROM User')
entries = cursor.fetchall()

for e in entries:
    print(e)

cursor.close()
db.close()

```

handle_db_classes.py

```

import MySQLdb

class Connection:
    def __init__(self, user, passwd, db, host='localhost', charset='utf-8',
use_unicode=True):
        self.user = user
        self.password = passwd
        self.db = db
        self.host = host
        self.charset = charset
        self.use_unicode = use_unicode
        self._connection = None
        self.cursor = 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
            )
            self.cursor = self._connection.cursor()

```

```

def _disconnect(self):
    if self._connection:
        self.cursor.close()
        self._connection.close()

def commit(self):
    self._connection.commit()

#=====

class User:
    def __init__(self, connection: Connection, name, age):
        self.name = name
        self.age = age
        self.connection = connection

    def save(self):
        self.connection.cursor.execute("INSERT INTO User (name, age) VALUES
(%s, %s);",
                                     (self.name, self.age))
        self.connection.commit()

    def show_table(self):
        self.connection.cursor.execute('SELECT * FROM User')
        entries = self.connection.cursor.fetchall()
        for e in entries:
            print(e)

connect = Connection(host="localhost",
                    user='Adm-User',
                    passwd='222',
                    db='services_db',
                    charset="utf8",
                    use_unicode=True)

with connect:
    user = User(connection=connect, name='Саша', age=20)
    user.save()
    user.show_table()

```

Скрины выполнения

Lab4

Services

Menu

Services	
Services	25
Popular services	2
New Services	3

Добро пожаловать

- [service №1](#)
Краткая информация об услуге №1
- [service №2](#)
Краткая информация об услуге №2
- [service №3](#)
Краткая информация об услуге №3

Services

Menu

Services	
Services	25
Popular services	2
New Services	3

Информация об услуге № 2

Lab5

Services

Menu

Services	3
Services	25
Popular services	2
New Services	3

Services

- Ремонт iPhone
- Билет в кино
- Заказ такси

Данные берутся из нашей БД

Django administration

WELCOME, SASHA. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Services > Services > Service object

Change service

HISTORY

Title:

Заказ такси

Category:

Обслуживание

Company:

"Доедем"

Delete

Save and add another

Save and continue editing

SAVE