

Демонстрация ЛР:
Гапанюк Ю.Е.

**Отчет по лабораторной работе № 6 по курсу
разработка интернет приложений**

"Работа с СУБД"

Вариант № <15>

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-54

Матвейчук И.А.

Задание

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

Текст программы

Файл connection.py:

```
import MySQLdb

class Connection:
    def __init__(self, user, password, db, host='localhost'):
        self.user = user
        self.host = host
        self.password = password
        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.password,
                db=self.db,
                charset='utf8',
                use_unicode=True
            )

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

class Computer:
    def __init__(self, db_connection, brand, price, processor_type, screen_size):
        self.db_connection = db_connection.connection
        self.brand = brand # производитель
        self.type = type # desktops, laptop, tablet
        self.screen_size = screen_size
        self.installed_OS = installed_OS
        self.processor_type = processor_type
```

```

#         self.RAM = RAM
        self.price = price

    def save(self):
        c = self.db_connection.cursor()
        c.execute('INSERT INTO my_lab6_computermodel (brand, price,
processor_type, screen_size) VALUES (%s, %s, %s, %s)',
                (self.brand, self.price, self.processor_type, self.screen_size))
        self.db_connection.commit()
        c.close()

    def update(self):
        c = self.db_connection.cursor()
        #self.price = '20 pyб'
        c.execute('UPDATE my_lab6_computermodel SET price = "0" WHERE id=2')
        self.db_connection.commit()
        c.close()

    def delete_item(self):
        c=self.db_connection.cursor()
        c.execute('DELETE FROM my_lab6_computermodel WHERE id=1')
        self.db_connection.commit()
        c.close()

con = Connection('dbuser', '123', 'shop_db')

with con:
    computer = Computer(con, 'Asus', '40000 pyб', 'Inspiron', 10.0)
    computer.save()
    computer.delete_item()
    computer.update()

```

Файл models.py:

```

from django.db import models

class ComputerModel (models.Model):
    brand = models.CharField(max_length=20, default='Dell') #производитель
    # type = models.CharField(max_length=10, default='laptop') #desktops, laptop,
    # tablet
    screen_size = models.FloatField(default=11)
    # installed_OS = models.CharField(max_length=20, default='Windows8')
    processor_type = models.CharField(max_length=20, default='Inspiron 11')
    # RAM = models.FloatField(default=2)
    price = models.CharField(max_length=20, default='30000 pyб')

class OrderModel (models.Model):
    order_num = models.IntegerField(unique=True)
    order_date = models.DateField()
    computer = models.ForeignKey('ComputerModel', null=True)

class UserModel (models.Model):
    order_num = models.ForeignKey('OrderModel', null=True)
    user_name = models.CharField(max_length=20)
    user_surname = models.CharField(max_length=20)
    user_thirdname = models.CharField(max_length=20)
    user_email = models.EmailField()
    user_phone = models.CharField(max_length=20)
    user_adress = models.CharField(max_length=50)

```

Файл urls.py:

```

from django.conf.urls import url
from django.contrib import admin
from my_lab6.views import ComputerView

```

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^computers/', ComputerView.as_view())
]
```

Файл views.py:

```
from django.shortcuts import render
from django.conf.urls import url
from django.views.generic import View
from my_lab6.models import ComputerModel

class ComputerView(View):
    def get(self, request):
        computers = ComputerModel.objects.all()
        return render(request, 'computer.html', {'computers':computers})
```

Файл base.html:

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel='stylesheet' type = 'text/css', href='{% static
"my_lab5/css/bootstrap.min.css" %}'>
    <title>{% block title %}{% endblock %}</title>
</head>
<body>

<div class="header">
    <div class="container" align="center">
        <h1>{% block visibletitle %}{% endblock %}</h1>
    </div>
</div>

<div>
    {% block body %}{% endblock %}
</div>
</body>
</html>
```

Файл computer.html:

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

{% block title %}{% endblock %}

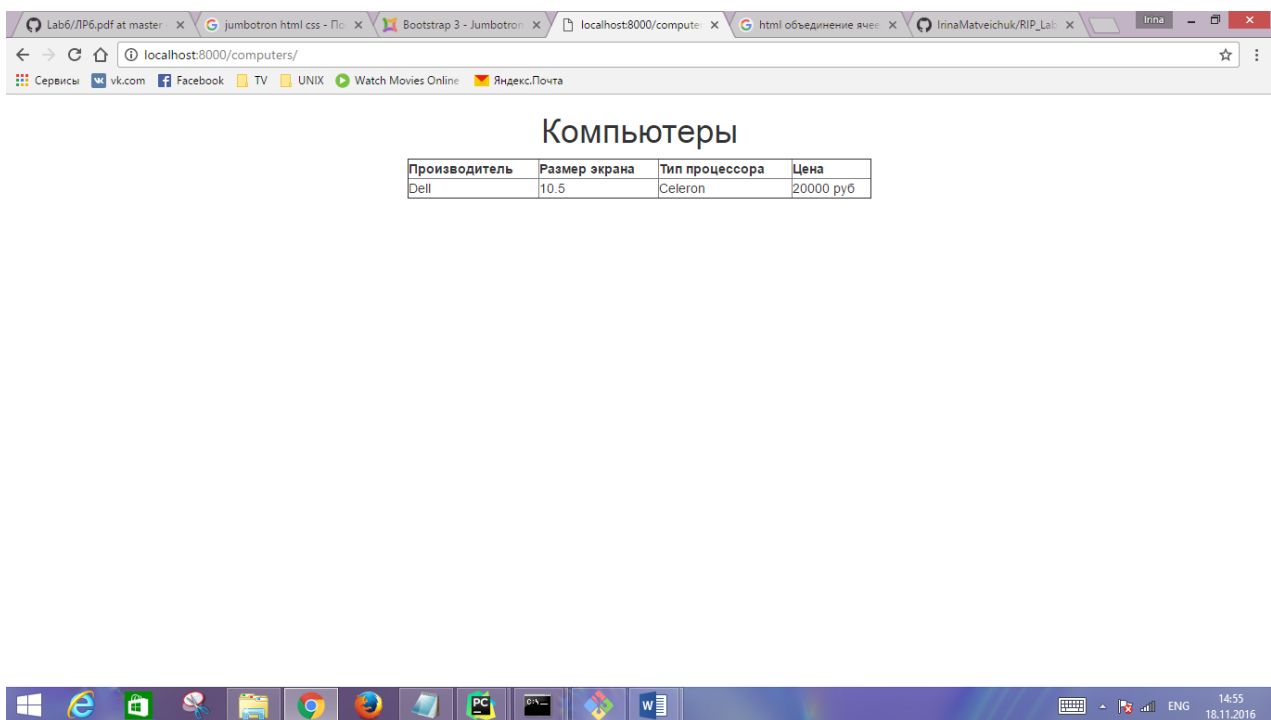
{% block visibletitle %}Компьютеры{% endblock %}

{% block body %}
    <table border="1" align="center" width="500">
        <tr><th align="middle">Производитель</th><th>Размер экрана</th><th>Тип
процессора</th><th>Цена</th></tr>
        {% for el in computers %}
            <tr><td>{{ el.brand }}</td><td>{{ el.screen_size }}</td><td>{{
el.processor_type }}</td><td>{{ el.price }}</td></tr>
            {% empty %}
                <tr><td colspan="4" align="center">Нет компьютеров</td></tr>
            {% endfor %}
        </table>
{% endblock %}
```

Изменения в файле settings.py:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'shop_db',  
        'USER': 'dbuser',  
        'PASSWORD': '123',  
        'HOST': 'localhost',  
        'PORT': 3306, # Стандартный порт MySQL  
        'OPTIONS': {'charset': 'utf8'},  
        'TEST_CHARSET': 'utf8',  
    }  
}
```

Результаты выполнения программы



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/computers/'. The page content includes a title 'Компьютеры' and a table with the following data:

Производитель	Размер экрана	Тип процессора	Цена
Dell	10.5	Celeron	20000 руб

The browser's taskbar at the bottom shows various application icons, including Windows Explorer, Google Chrome, and Microsoft Word. The system clock in the bottom right corner indicates the time is 14:55 on 18.11.2016.

Таблицы, созданные после миграции:

Query 1 x

Limit to 1000 rows

```
1 show tables
2
3
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Read Only

Tables_in_shop_db
django_content_type
django_migrations
django_session
my_lab6_computermodel
my_lab6_ordermodel
my_lab6_usermodel

Query 1 x

Limit to 1000 rows

```
1 select * from my_lab6_computermodel
2
3
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Read Only

	id	brand	price	processor_type	screen_size
▶	2	Samsung	0	Celeron	10
	3	Asus	40000 py6	Inspiron	10
	4	Asus	40000 py6	Inspiron	10
*	NULL	NULL	NULL	NULL	NULL