

Защищено:  
Гапанюк Ю. Е.

Демонстрация:  
Гапанюк Ю. Е.

"\_\_" \_\_\_\_\_ 2016 г.

"\_\_" \_\_\_\_\_ 2016 г.

**Отчет по лабораторной работе № 6 по курсу  
Разработка интернет-приложений**

**Тема работы: "Работа с СУБД"**

5  
(количество листов)  
Вариант № 23

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-54

Тимошенко В.А.

\_\_\_\_\_  
(подпись)

"\_\_" \_\_\_\_\_ 2016 г.

## Задание и порядок выполнения

В этой лабораторной работе ознакомимся 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 connect.py

```
from mysql import connector as 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,
```

```
            )
```

```
    def disconnect(self):
```

```
        if self._connection:
```

```
            self._connection.close()
```

```
class User:
```

```
    def __init__(self, db_connection, name, age):
```

```
        self.db_connection = db_connection.connection
```

```

        self.name = name
        self.age = age
    def save(self):
        u = self.db_connection.cursor()
        u.execute("INSERT INTO users (`name`, age) VALUES (%s, %s)", (self.name, self.age,))
        self.db_connection.commit()
        u.close()
con = Connection('newuser', '123', 'new_schema')
with con:
    u = User(con, 'Pet', 22)
    u.save()

```

## файл models.py

**from** django.db **import** models

```

class TovarModel(models.Model):
    name = models.CharField(max_length=70)
    desc = models.CharField(max_length=300)
    cout = models.IntegerField()

```

## файл urls.py

**from** bk.views **import** index, ExampleView, OrdersView, OrderView

```

from django.conf.urls import url
urlpatterns = [
    url(r'^$', index),
    url(r'^example$', ExampleView.as_view()),
    url(r'^order$', OrdersView.as_view(), name='orders_url'),
    url(r'^order/(?P<id>(d+))$', OrderView.as_view(), name='order_url'),
]

```

## файл base.py

<!DOCTYPE **html**>

```

<html lang="en">
<head>
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.css'%}">
    <title>{% block title %}{% 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" href="#">Brand</a>
        </div>
        <!-- /.container-fluid -->
    </div>
</nav>
<div class="container-fluid">
    <div class="row">

```

```

    {% block body%} Переопределяется в наследниках
{% endblock %}
</div>
</div>
</body>
</html>

```

## файл orders.py

```

{% extends 'base.html' %}
{% block title %}Заказы №{{order.id}}{% endblock %}
{% block body%}
<div class="col-md-4">
    <div> Общая часть </div>
    <ul>
        {% for order in orders %}
            {%include 'order_element.html' with order=order %}
        {% empty %}
            <li> пустой список </li>
        {% endfor %}
    </ul>
</div>
{% endblock %}

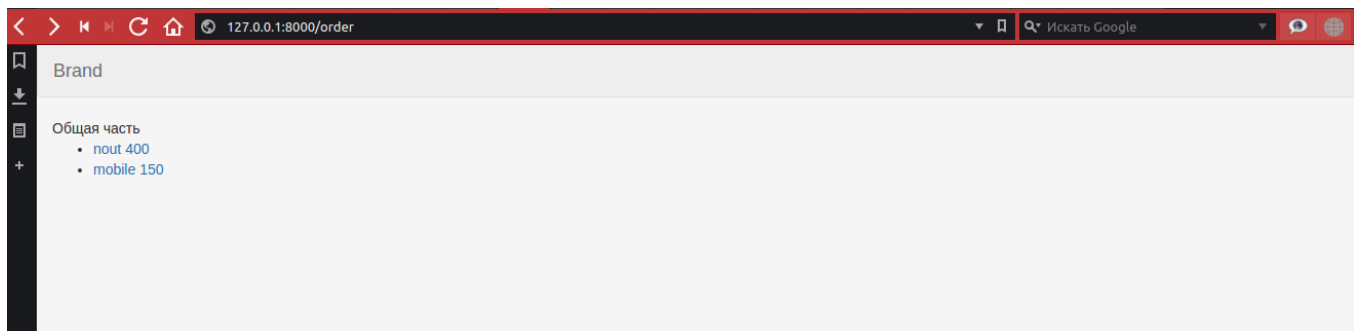
```

## файл setting.py

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

```



## Результаты выполнения работы

File Edit View Navigate Code Refactor Run Tools VCS Window Help

Database db.sqlite3 bk\_tovarmodel

Project

- db.py
- models.py
- views.py
- mysite
  - \_\_init\_\_.py
  - settings.py
  - urls.py
  - wsgi.py
- templates
  - base.html
  - example.html
  - order\_element.html
  - orders.html
- db.sqlite3
- manage.py
- External Libraries

bk\_tovarmodel x models.py x connect.py x order\_element.html x

Database

connect

2 rows

Tab-se...d (TSV) View Query

<Filter criteria>

| id | name   | desc | cout |
|----|--------|------|------|
| 2  | mobile | link | 150  |
| 2  | nout   | work | 400  |

Data DDL

db.sqlite3

- auth\_group
- auth\_group\_permissions
- auth\_permission
- auth\_user
- auth\_user\_groups
- auth\_user\_user\_permissions
- bk\_tovarmodel
  - id INTEGER
  - name TEXT
  - desc TEXT
  - cout INTEGER
  - (id)
- django\_admin\_log
- django\_content\_type

Terminal

```
[11/Jan/2017 13:43:26] "GET /order/2 HTTP/1.1" 200 1050
vovatin@vovatin-TravelMate-5742G:~/PycharmProjects/Lab5/mysite$ python3 manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
January 11, 2017 - 13:59:46
Django version 1.10.3, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[11/Jan/2017 13:59:58] "GET / HTTP/1.1" 200 13
[11/Jan/2017 14:00:04] "GET /order HTTP/1.1" 200 1119
[11/Jan/2017 14:03:23] "GET /order HTTP/1.1" 200 1126
[11/Jan/2017 14:03:23] "GET /static/css/bootstrap.css HTTP/1.1" 304 0
{'orders': <QuerySet [<TovarModel: TovarModel object>]>}>
[11/Jan/2017 14:03:26] "GET /order/1 HTTP/1.1" 200 1052
```

Run TODO Python Console Terminal Event Log

1:2