

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

Кафедра «Системы обработки информации и управления»

Разработка интернет приложений

Отчет по Лабораторной работе №6.

Исполнитель: студент группы РТ5-51

Бгатцев А. В.

Москва – 2016

Задания лабораторной:

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

1) Класс Connection - для соединения с базой данных

```
class Connection():
    def __init__(self, user, password, host, database):
        self.user=user
        self.password = password
        self.host = host
        self.database = database
        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 =
connector.connect(user=self.user,password=self.password,host=self.host,databa
se=self.database)
    def disconnect(self):
        if self._connection:
            self._connection.close()
```

2) Класс для работ с БД

```
class Catalog:
    def __init__(self,db_connection, product, category, price, quantity,
foto):
        self.db_connection=db_connection
        self.product=product
        self.category = category
        self.price = price
        self.quantity = quantity
        self.foto = foto

    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO user (product, category, price, quantity, foto)
VALUES (%s,%s,%s,%s,%s)", (self.product, self.category, self.price,
self.quantity, self.foto))
        self.db_connection.commit()
        c.close()

    def spisok(self):
        c = self.db_connection.cursor()
        c.execute("SELECT * FROM catalog")
        takeall=c.fetchall()
        arr = []
        for e in takeall:
            arr.append(Catalog(self.db_connection, e[1], e[2], e[3], e[4],
e[5]))
        return arr
```

```

conn=Connection('alex', '177', '127.0.0.1', 'shope')
with conn:
    prod=Catalog(conn.connection, 'поза', 'позы', '100', '7',
'http://file.mobilmusic.ru/8e/93/2f/1330096.jpg')
    #man.save()
    a = prod.spisok()
    print(a[0].foto)

```

3) Модель предметной области

```

from django.db import models
from django.utils import timezone

class Catalog(models.Model):
    id = models.AutoField(auto_created=True, primary_key=True,
serialize=False)
    product = models.CharField(max_length=255)
    category = models.CharField(max_length=255)
    price = models.CharField(max_length=255)
    quantity = models.CharField(max_length=255)
    foto = models.CharField(max_length=255)

    def publish(self):
        self.published_date = timezone.now()
        self.save()

    def __str__(self):
        return self.title

    class Meta:
        db_table = 'catalog'

```

4) View отображающая список сущностей

```

from django.shortcuts import render
from django.http import HttpResponse
from datetime import date
from .models import Catalog

def catalog(request):
    data = Catalog.objects.all()
    return render(request, 'catalog.html', {'catalog': data})

```

Страница

```

{% extends 'base.html' %}

{% block title %}Каталог{% endblock %}
{% block content %}
    <div class="container">
        <div class="jumbotron">
            <div class="row">
                {% for IDProduct in catalog %}
                    <div class="col-md-4">
                        <div class="thumbnail">
                            
                            <div class="caption">
                                <p>{{ IDProduct.product }}</p>
                                <p><a class="btn btn-lg btn-success" href="{%
url 'product' IDProduct.id %}" role="button">Купить</a></p>
                            </div>
                        </div>
                    </div>
                {% endfor %}
            </div>
        </div>
    </div>

```

```
</div>  
</div>  
</div>  
{ % endblock % }
```

Отображение страницы в браузере

