**Московский государственный технический**

**университет им. Н.Э. Баумана.**

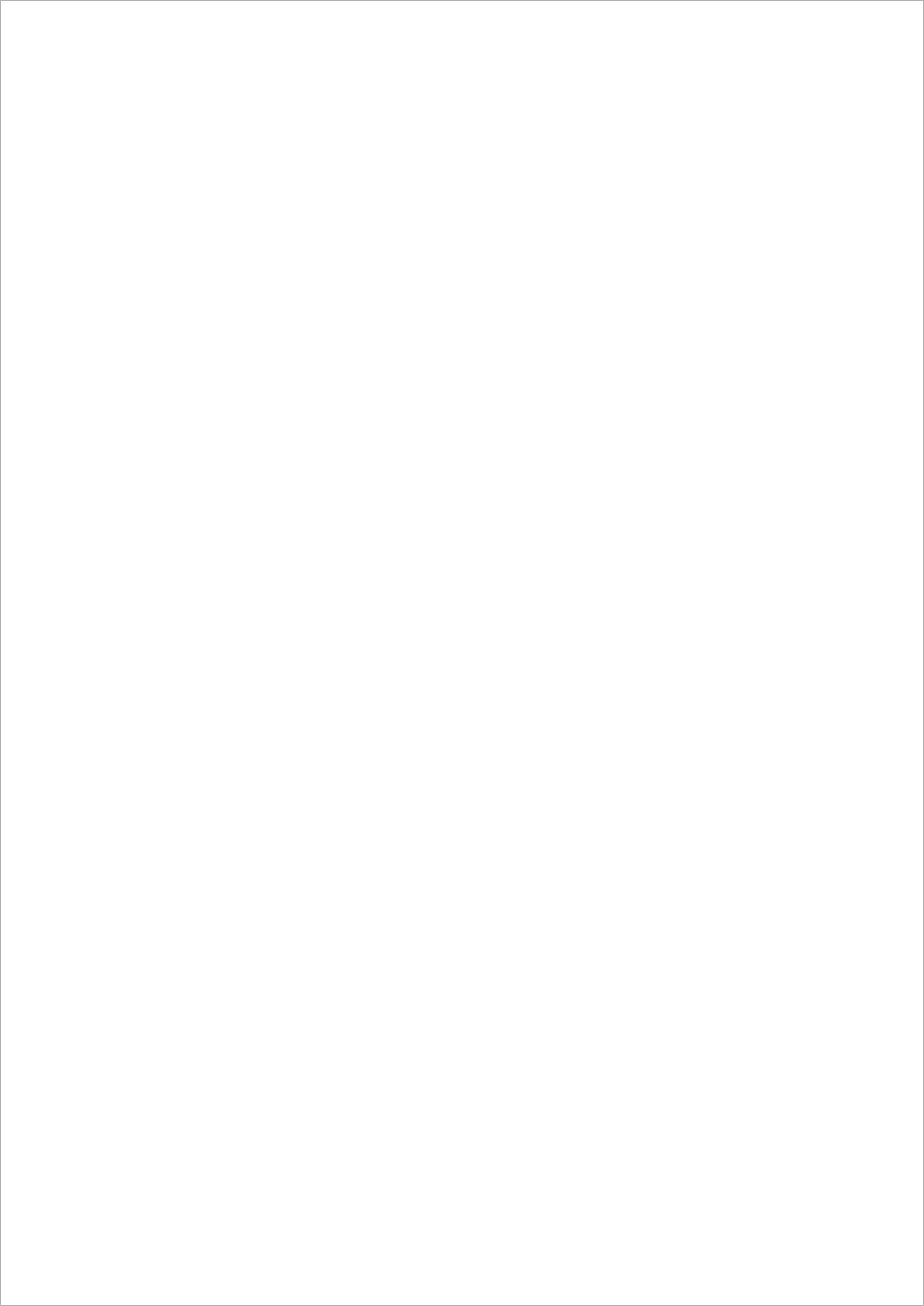
Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №5

|  |  |  |
| --- | --- | --- |
| Выполнила: |  | Проверил: |
| студентка группы ИУ5-51 |  | преподаватель каф. ИУ5 |
| Цветкова А.К. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

г. Москва, 2018 г.



Задание лабораторной работы.

* 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 должны быть:

1.Скрипт с подключением к БД и несколькими запросами.

2. Набор классов предметной области с привязкой к СУБД (класс должен уметь

хотя бы получать нужные записи из БД и преобразовывать их в объекты этого класса).

1. Модели предметной области.
2. View для отображения списка сущностей.

Исходный код/

1. Обращение к БД из Python:

import pymysql

pymysql.install\_as\_MySQLdb()

db=pymysql.connect(

host='localhost',

user='dbuser',

passwd='123',

db='first\_db'

)

c=db.cursor()

c.execute('SELECT \*FROM books;')

entries=c.fetchall()

for e in entries:

print(e)

c.close()

db.close()

1. Написание классов предметной области с соединением с БД.

import pymysql

pymysql.install\_as\_MySQLdb()

class Connection:

def \_\_init\_\_(self, user, password, db, host='localhost'):

* ***Сохраняем параметры соединения*** self.user = user self.password = password self.db = db

self.host = host

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):

***"""***

***Открытие соединения***

***"""***

if not self.\_connection:

self.\_connection = pymysql.connect(

host=self.host,

user=self.user,



passwd=self.password,

db=self.db

)

def disconnect(self):

***"""***

***Закрытие соединения***

***"""***

if self.\_connection:

self.\_connection.close()

class books:

def \_\_init\_\_(self, db\_connection, id, name, autor):

self.db\_connection = db\_connection.connection

self.id=id

self.name = name

self.autor = autor

def save(self):

c = self.db\_connection.cursor()

c.execute("INSERT INTO Books (id, name, autor) VALUES (%s, %s, %s);", (self.id, self.name, self.autor))

self.db\_connection.commit()

c.close()

con = Connection('dbuser', '123', 'first\_db')

with con:

book = books(con, '6', 'Книга Книжная', 'Автор Авторный')

book.save()

1. Django ORM. models.py:

from django.db import models

class BookModel(models.Model):

class Meta:

db\_table = 'books'

name = models.CharField(max\_length=25)

author = models.CharField(max\_length=50)

def \_\_str\_\_(self):

return "{}, {}".format(self.name, self.author)

views.py:

from django.shortcuts import render

* ***Create your views here.*** from django.views import View

from library.models import BookModel

class BookView(View):

def get(self, request):

books = BookModel.objects.all()

data = {

'books': books

}

return render(request, 'bookinfo.html', data)

bookinfo.html:

{% extends 'base.html' %}

{% block title %}Some data{% endblock %}

{% block body %}

<ul>

{% for book in books %}

<li>

Название - {{ book.name }}<br>

Автор - {{ book.author }}.</li>

{% endfor %}

</ul>

{% endblock %}