

Московский государственный технический
университет им. Н.Э. Баумана
Факультет «Информатика и системы управления» Кафедра
«Системы обработки информации и управления»



“Разработка интернет- приложений”

«Python-классы»

Лабораторная работа № 6

Студент группы ИУ5-53

_____ Атаманов В. В.

Преподователь

_____ Гапанюк Е. Ю.

Москва 2017

Задачение

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

Листинг:

ex.py

```
import MySQLdb

#! Открытие соединение с базой данных
db = MySQLdb.connect(
    host="localhost",
    user="dbuser",
    passwd="3298",
    db="lab_db"
)
db.set_character_set('utf8')
#! Получить курсор для работы с базой данных
c=db.cursor()

#! Выполнить вставку
c.execute("insert into labapp_departments (name, description) VALUES (%s, %s);", ('Отдел маркетинга', 'Этаж 4'))
#! Фиксирование изменений
db.commit()

#! Выполнить выборку
c.execute("select * from labapp_departments;")

#! Забрать все полученные записи
entries = c.fetchall()

#! Распечатать записи
for e in entries:
    print(e)

#! Закрытие курсора
c.close()
#! Закрытие соединения
db.close()
```

models.py:

```
from django.db import models

class Author(models.Model):
    first_name = models.CharField(max_length=32)
```

```

last_name = models.CharField(max_length=32)

def __str__(self):
    return '{} {}'.format(self.first_name, self.last_name)

class Meta:
    ordering = ('first_name', 'last_name')

class Book(models.Model):
    title = models.CharField(max_length=64)
    author = models.ForeignKey(Author)
    image = models.ImageField(upload_to='static/books')
    year = models.IntegerField(default= None)
    count = models.IntegerField(default= 0)

    def __str__(self):
        return self.title

class Meta:
    ordering = ('title', 'image', 'year', 'count')

```

Результат отображения списка

