Отчет по лабораторной работе №6 «Работа с базами данных» по курсу «Разработка интернет приложений»

Выполнил студент ИУ5-51 Никитин К.И.

Цель работы

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

Скрипты

Скрипт с подключением к БД и несколькими запросами class Connection:

```
def __init__(self, user, password, db, host="localhost"):
        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):
        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 Book:
   def __init__(self, db_con, name, descr):
        self.db con = db con.connection
        self.name = name
        self.descr = descr
   def save(self):
        c = self.db con.cursor()
        c.execute("INSERT INTO books (name, description) VALUES (%s, %s);",
                  (self.name, self.descr))
        self.db con.commit()
```

```
c.close()
conect = Connection("kirill", "1234", "lab6 db")
with conect:
    book = Book(conect, 'New', "New description of book")
    book.save()
Модели вашей предметной области
class My user(models.Model):
    class Meta:
       db_table = "Users"
    first name = models.CharField(max length = 50)
    second name = models.CharField(max length=50)
    group = models.CharField(max length= 15)
    birthday = models.DateField()
class Ser f users(models.Model):
   class Meta():
       db table = "Serveses"
    price = models.IntegerField(verbose name="цена")
    type of serves = models.CharField(max length=50, verbose name='тип услуги')
    customer = models.ForeignKey(My user)
    use_of_serves = models.IntegerField(default=0)
View для отображения списка ваших сущностей
from django.shortcuts import render, redirect
from django.http.response import HttpResponse
from django.template import Context
from django.template.context processors import csrf
from django.views import generic
from bdwork.models import My_user, Ser_f_users
from .forms import ServesForm
# Create your views here.
def basicone(request):
   view = "basicone"
    html = "<html><body>This is %s view</body></html>" % view
    return HttpResponse(html)
class Users(generic.ListView):
    template name = "users0.html"
   def get queryset(self):
```

return render(request, 'users.html', {'users': My user.objects.all()})

args['servese_id'] = Ser_f_users.objects.filter(customer=user_id)

serv.customer = Ser f users.objects.get(id=user id)

args['user'] = My_user.objects.get(id=user_id)

return render(request, 'singleuser.html', args)

return My user.objects.all()

args.update(csrf(request))

args['form'] = serves form

form = ServesForm(request.POST)

serv = form.save(commit=False)

return redirect("/users/get/%s" % user id)

def singleuser(request, user_id = 1):
 serves form = ServesForm

def addserves(request, user id):

if form.is_valid():

form.save()

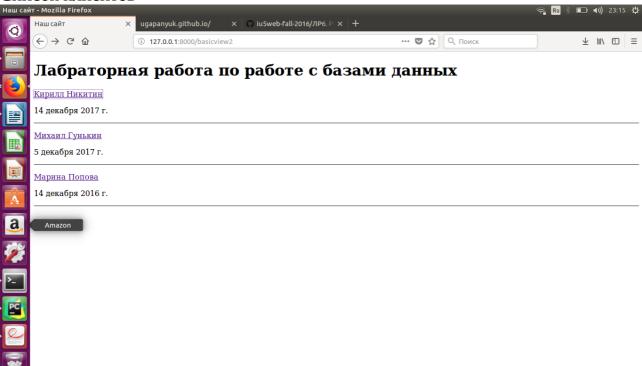
def users(request):

 $args = \{\}$

if request.POST:

Результаты

Список клиентов



Конкретный клиент

