

**Отчет по лабораторной работе №6
по курсу «Разработка Интернет-приложений»**

Вариант № <21>

ИСПОЛНИТЕЛЬ:
студент группы ИУ5-52
Ромичева Е.В.

Содержание

Описание задания лабораторной работы	3
Исходный код.....	3

Описание задания лабораторной 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

Исходный код

Обращение к БД из Python

```
db=pymysql.connect(
    host='localhost',
    user='root',
    passwd='1234',
    db='my_db'
)

c=db.cursor()
c.execute(
    'insert into horses (horse_id, name, ratio, age)
    values(%s,%s,%s,%s);', (2, 'holly', 58,4)
)
db.commit()
c.execute(
    'select * from horses;'
)
res_horses=c.fetchall()
for h in res_horses:
    print(h)

c.close()
db.close()
```

Написание классов предметной области с соединением с БД

```
import sys
import pymysql
class Connection:
    def __init__(self, user, password, db, host='localhost'):
        self.user = user
        self.password=password
        self.host=host
        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=pymysql.connect(
            host=self.host,
            user=self.user,
            passwd=self.password,
            db=self.db
        )

def disconnect (self):
    if self._connection:
        self._connection.close()

class Jockey:
    def __init__(self, db_connection, jockey_id, name, contacts,
ratio):
        self.db_connection=db_connection.connection
        self.jockey_id=jockey_id
        self.name=name
        self.contacts=contacts
        self.ratio=ratio

    def save(self):
        c=self.db_connection.cursor()
        c.execute('INSERT INTO jockeys (jockey_id, name, contacts,
ratio) VALUES (%s,%s,%s,%s);',
            (self.jockey_id, self.name, self.contacts,
self.ratio)
        )
        self.db_connection.commit()
        c.close()

con= Connection ('root', '1234', 'my_db')
with con:
    jockey=Jockey(con, 2, 'verv', '1244@mail', '5657')
    jockey.save()

```

Django ORM

Views.py

```

from django.shortcuts import render
from .models import User

```

```

from django.views import View
from pip._vendor import requests

class user_view(View):
    def get(self, request):
        users = User.objects.all()
        return render (request, 'bookmaking/users.html', {'users':
users})

def main_page(request):
    return render (request, 'bookmaking/main_page.html', {})

```

models.py

```

from django.db import models
import pymysql
pymysql.install_as_MySQLdb()

class User(models.Model):
    user_id = models.IntegerField()
    name = models.CharField(max_length=50)
    age = models.IntegerField()
    stake = models.FloatField()
    contacts = models.CharField(max_length=50)
    bank_account = models.CharField(max_length=50)

class Horse(models.Model):
    horse_id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=50)
    age = models.IntegerField()
    ratio = models.FloatField()

```

urls.py

```

from django.conf.urls import url
from . import views
from bookmaking.views import user_view

urlpatterns = [
    url(r'^$', views.main_page),
    url(r'^user/$', user_view.as_view()),
]

```

main_page.html

```

{% extends 'bookmaking/base.html' %}
{% load staticfiles %}

{% block content %}
<div class='jumbotron'>

```

```
<div class='span8'>
    <h2> welcome! </h2>
    
</div>
</div>
```

```
{% endblock content %}
```

users.html

```
{% extends 'bookmaking/base.html' %}
{% load staticfiles %}

{% block content %}
    <h4>Список пользователей</h4>
    <div class="container">

        <table class="table table-striped">
            <thead>
                <tr>
                    <th>Имя</th>
                    <th>Возраст</th>
                    <th>Контактные данные</th>
                    <th>Размер ставки</th>
                </tr>
            </thead>

            <tbody>
                {% for user in users %}
                <tr>
                    <td>{{user.name}}</td>
                    <td>{{user.age}}</td>
                    <td>{{user.contacts}}</td>
                    <td>{{user.stake}}</td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
{% endblock content %}
```