Отчет по лабораторной работе №7
по курсу «Разработка интернет-приложений»
«Авторизация, работа с формами и Django Admin»

Выполнил:	Преподаватель:
Калиниченко Ирина, ИУ5-52	Гапанюк Ю.Е.
____________________	____________________

2016 г.

Задание

Основная цель данной лабораторной работы – научиться обрабатывать веб-формы на стороне приложения, освоить инструменты, которые предоставляет Django, по работе с формами. Также в этой лабораторной работе вы освоите инструменты Django по работе с авторизацией и реализуете простейшую авторизацию. Напоследок, вы познакомитесь с инструментом администрирования Django – как в несколько строчек кода сделать панель администратора сайта.

1. Создайте view, которая возвращает форму для регистрации.

   Поля формы:
   - Логин
   - Пароль
   - Повторный ввод пароля
   - Email
   - Фамилия
   - Имя

2. Создайте view, которая возвращает форму для авторизации.

   Поля формы:
   - Логин
   - Пароль

3. При отправке формы регистрации во view проверять каждый параметр по правилам валидации, если валидация всех полей пройдена, то создавать пользователя и делать перенаправление на страницу логина, а ошибки, если они есть, выводить над формой.

   Правила валидации:
   - Логин не меньше 5 символов
   - Пароль не меньше 8 символов
   - Пароли должны совпадать
   - Все поля должны быть заполнены
   - Логин – уникален для каждого пользователя

4. При возникновении ошибок в момент отправки формы, введенные значения в полях ввода, кроме пароля, не должны исчезать.

5. Переписать view регистрации с использованием Django Form, правила валидации удалить из view, использовать встроенный механизм валидации полей.

6. Во view авторизации реализовать логин при POST запросе. При успешной авторизации должен происходить переход на страницу успешной авторизации.

7. Страница успешной авторизации должна проверять, что пользователь авторизован. Иначе делать перенаправление на страницу авторизации.

8. Реализовать view для выхода из аккаунта.

9. Заменить проверку на авторизацию на декоратор login_required

10. Добавить superuser'a через комманду manage.py

11. Подключить django.contrib.admin и войти в панель администрирования.

12. Зарегистрировать все свои модели в django.contrib.admin

13. Для выбранной модели настроить страницу администрирования:
    - Настроить вывод необходимых полей в списке
    - Добавить фильтры
    - Добавить поиск
    - Добавить дополнительное поле в список

Код программы

shop/urls.py

```
from django.conf.urls import url
from . import views
from shop.views import NewView, BasicView, SaleView
from django.contrib.auth import views as auth_views
from django.contrib.auth.decorators import login_required

urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^home/', views.navig, name='navigate'),
    url(r'^show_new/', login_required(redirect_field_name='',
login_url='/login')(NewView.as_view())),
    url(r'^show_all/', login_required(redirect_field_name='',
login_url='/login')(BasicView.as_view())),
    url(r'^show_sweet/', login_required(redirect_field_name='',
login_url='/login')(SaleView.as_view())),
    url(r'^reg/', views.registration, name='registration'),
    url(r'^login/', views.log, name='login'),
    url(r'^logout/', views.logout_view, name='logout'),
    url(r'^login/$', auth_views.login),
]
```

lab6/urls.py

```
from django.conf.urls import include, url
from django.conf.urls.static import static
from django.conf import settings
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'', include('shop.urls')),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

view.py

```
from django.shortcuts import render, redirect
from shop.models import Category, Item
from django.views import View
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth import authenticate, logout
from django.contrib.auth.hashers import make_password
from django.contrib import auth
from django.contrib.auth.decorators import login_required


@login_required(redirect_field_name='', login_url='/login')
def index(request):
    return render(request, 'index.html')


@login_required(redirect_field_name='', login_url='/login')
def navig(request):
    data = Category.objects.all()
    return render(request, 'navigate.html', context={'menu': data})


class NewView(View):
    def get(self, request):
        data_show_n = Item.objects.filter(category_id=1).all()
        if len(data_show_n) == 0:
            return render(request, 'empty.html')
        else:
            return render(request, 'show.html', context={'search':
data_show_n})


class BasicView(View):
    def get(self, request):
        data_show_a = Item.objects.filter(category_id=2).all()
        if len(data_show_a) == 0:
            return render(request, 'empty.html')
        else:
            return render(request, 'show.html', context={'search':
data_show_a})


class SaleView(View):
    def get(self, request):
        data_sweet = Item.objects.filter(category_id=3).all()
        if len(data_sweet) == 0:
            return render(request, 'empty.html')
        else:
            return render(request, 'show.html', context={'search':
data_sweet})


class RegistrationForm(forms.Form):
    username = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'id':
'username', 'placeholder': 'Enter login', }), \
        min_length=5, label='Login:')
    name = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'id': 'name',
'placeholder': 'Enter name', }), \
        max_length=30, label='Name:')
    surname = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'id':
'surname', 'placeholder': 'Enter surname', }), \
        max_length=30, label='Surname:')
    email = forms.EmailField(
        widget=forms.EmailInput(attrs={'class': 'form-control', 'id':
'email', 'placeholder': 'Enter email', })
    )
    password = forms.CharField(min_length=8, label='Password:',
widget=forms.PasswordInput(
        attrs={'class': 'form-control', 'id': 'password', 'placeholder':
'Enter password', }))
    password2 = forms.CharField(min_length=8, label='Confirm password:',
widget=forms.PasswordInput(
        attrs={'class': 'form-control', 'id': 'password2', 'placeholder':
'Confirm password', }))

    def clean_password2(self):
        p1 = self.cleaned_data.get('password')
        p2 = self.cleaned_data.get('password2')
        if p1 != p2:
            raise forms.ValidationError('Passwords does not match')

    def save(self):
        u = User()
        u.username = self.cleaned_data.get('username')
        u.password = make_password(self.cleaned_data.get('password'))
        u.first_name = self.cleaned_data.get('name')
        u.last_name = self.cleaned_data.get('surname')
        u.email = self.cleaned_data.get('email')
        u.is_staff = False
        u.is_active = True
        u.is_superuser = False
        u.save()

    def clean_username(self):
        username = self.cleaned_data.get('username')
        try:
            u = User.objects.get(username=username)
            raise forms.ValidationError('This login already is used')
        except User.DoesNotExist:
            return username


def registration(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('/login')
        return render(request, 'registration.html', {'form': form})
    else:
        form = RegistrationForm()
    return render(request, 'registration.html', {'form': form})


def log(request):
    errors = []
    username = ''
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        if not username:
            errors.append('Input login')
        elif not password:
            errors.append('Input password')
        else:
            user = authenticate(username=username, password=password)
            if user:
                auth.login(request, user)
                if not request.POST.get('remember'):
                    request.session.set_expiry(0)
                return redirect('/')
            else:
                errors.append('Wrong login or password')
    return render(request, 'login.html', {'errors': errors, 'name':
username})


def logout_view(request):
    logout(request)
    return render(request, 'logout.html')
```

base.html

```
{% load staticfiles %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
    <link rel="stylesheet" type="text/css" href="/static/css/bootstrap.css">
    <link href="/static/css/jumbotron-narrow.css" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Calligraffitti"
rel="stylesheet">
    <style>
    body {
        background: url(/static/3.jpg)  no-repeat center center fixed;
        -webkit-background-size: cover;
        -moz-background-size: cover;
        -o-background-size: cover;
        background-size: cover;
   }
        h3 {
            font-family: 'Calligraffitti';
            font-size: xx-large;
        }
  </style>
</head>
<body>
    <div class="container-fluid">
        <style>
            .link1 {
                font-size: 50px;
                color: darkblue;
            }
        </style>
        <div class="text-center">
            <h3><a href="/" class="link1"> Enjoy with cooking </a></h3>
        </div>
        {% if user.username != '' %}
            <div class="col-md-offset-9">
                <p class="user">You are signed in as {{ user.username }}</p>
                <a href="/logout" class="btn btn-primary"
role="button">Logout</a>
                <br>
                <br>
            </div>
        {% endif %}
    </div>
{% block content %}{% endblock %}

</body>
</html>
```

index.html

```
{% extends 'base.html' %}

{% block title %}Home{% endblock %}
{% block content %}
    <div class="jumbotron">
        <p class="lead">Hello. This is my test project. Nice to meet you!
Click on the link below to begin. </p>
        <a href="/home" class="btn btn-primary" role="button" >Cooking</a>
    </div>
{% endblock %}
```

empty.html

```
{% extends 'base.html' %}
{% block title %}Search{% endblock %}
{% block content %}
    <div class="container-fluid">
        <div class="blog-header">
            <h1 class="blog-title col-lg-offset-5"><b><i>
                <mark>Best recipies! Do it!</mark>
            </i></b></h1>
        </div>
        <div class="row">
            <div class="col-lg-8 blog-main">
                <h1 class="blog-title col-lg-offset-4">There are no
recipies:( But we will add something soon! Don't
                    worry! </h1>
            </div>
        </div>
    </div>

{% endblock %}
```

navigate.html:

```
{% extends 'base.html' %}

{% block title %}menu{% endblock %}
{% block content %}
    <div class="pp menu">
        {% for category in menu %}
            <div class="col-lg-5">
                    <div class="panel panel-default">
                         <div class="panel-body bg-info">
                    <style>
                        .s1 {
                            text-align: center;
                            font-weight: 700;
                        }
                    </style>
                              <h4> <p class="s1">{{ category.name }}</p></h4>
                              <div class="caption">
                                  <p><div style="text-align: center;">{{
category.description }}</div></p>
```

```
                                        <div class="s1">
                                            {% if category.name == 'New recipies' %}
                                                <a href="/show_new" class="btn btn-
primary" role="button">{% include 'show_q.html' with field=category.name
%}</a>
                                            {% elif category.name == 'All recipies'
%}
                                                <a href="/show_all" class="btn btn-
primary" role="button">{% include 'show_q.html' with field=category.name
%}</a>
                                            {% elif category.name == 'Cake and
sweet' %}
                                                <a href="/show_sweet" class="btn
btn-primary" role="button">{% include 'show_q.html' with field=category.name
%}</a>
                                            {% endif %}
                                        </div>
                                    </div>
                                </div>
                           </div>
                   </div>
               {% endfor %}
           </div>
{% endblock %}
```

show.html:

```
{% extends 'base.html' %}
{% block title %}Search{% endblock %}
{% block content %}
    <div class="container-fluid">
        <div class="blog-header">
            <h1 class="blog-title col-lg-offset-5"><b><i><mark>Best recipies!
Do it!</mark></i></b></h1>
        </div>
        <div class="row">
            <div class="col-lg-8 blog-main">
                {% for item in search %}
                    <section class="panel panel-search">
                    <div class="panel-heading bg-info">
                        <div class="panel-title">
                            <h4> {{  item.name}}<small> <span class="pull-
right">{{ item.a }} {{ item.level }}</span></small> </h4>
                        </div>
                    </div>
                    <div class="panel-body ">
                        <div class="row">
                            <div class="col-lg-4">
                                <img src="{{ item.image }}" width="100%"/>
                            </div>
                            <div class="col-lg-4">
                                {{ item.description }}
                            </div>
                        </div>
                    </div>
                </section>
                {% endfor %}
            </div>
            <div class="col-lg-3 blog-sidebar">
                <section class="panel panel-search">
                    <div class="panel-heading bg-info">
                        <div class="panel-title">
                            <h3>Search</h3>
                        </div>
                    </div>
                    <h4><div class="col-lg-offset-1">Filters</div></h4>
                    <ol class="list-unstyled text-center">
                        <li><a href="#">Level low to high</a></li>
                        <li><a href="#">Level high to low</a></li>
                    </ol>
                    <div class="panel-body bg-info">
                        <div class="input-group">
                            <form data-
key="2af70d95e12e1e4e9344fa7468f8213d00434d93" action="search" method="get"
style="margin-bottom:10px;">
                            <span class="input-group-btn">
                                    <input type="text" class="form-control"
name="query" placeholder="Enter here" value="" />
                                    <button type="submit" class="btn btn-
default">Find</button>
                                </span>
                                </form>
                            </div>
                        </div>
                </section>
            </div>
        </div>
    </div>

{% endblock %}
```

show_q.html

```
DO IT!!!
```

login.html

```
{% extends 'base.html' %}
{% csrf_token %}
{% block title %}Sign in{% endblock %}
{% block content %}
<form action="/login/" method="post">
    {% csrf_token %}
    <div class="form-group" style="margin: 100px 375px 0px;">
            {% if errors %}
                <div class="alert alert-warning">
                <a href="" class="close" data-dismiss="alert">×</a>
                {% for e in errors %}
                    <p>{{ e }}</p>
                {% endfor %}
            {% endif %}
        </div>
    </div>
    <form class="form-inline">

    <div class="form-inline"  style="margin: 0px 0px 0px 375px;">
        <div class="form-group">
            <label class="sr-only" >Login</label>
            <input type="text" class="form-control" placeholder="Login"
id="username" name="username" autocomplete="on" value="{{ name }}">
        </div>
        <div class="form-group">
            <label class="sr-only" for="inputPassword">Password</label>
            <input type="password" class="form-control" id="password"
placeholder="Password" name="password">
```

```
        </div>
        <div class="checkbox">
            <label><input name="remember" id="remember"
type="checkbox">Remember me</label>
        </div>
        <button type="submit" class="btn btn-default">Sign in</button>
    </div>
    </form>
    <div class="form-group" style="margin: 10px 0px 0px 375px;">
        <label>Not a member?</label>
        <a href="/reg">
            <span style="color: #111;">
            Sign up here!
            </span>
        </a>
    </div>

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

logout.html

```
{% extends 'base.html' %}

{% block title %}Goodbye!{% endblock %}
{% block content %}
    <div class="jumbotron">
        <p class="user">We are waiting for you again!</p>
    </div>
{% endblock %}
```

registration.html

```
{% extends 'base.html' %}
{% block title %}Become a member of our club{% endblock %}
{% block content %}
<form class="form-horizontal" action="/reg/" method="POST">
{% csrf_token %}
    <div class="form-group">
        {% if form.errors%}
            <div class="col-md-4 col-md-offset-2">
            <div class="alert alert-warning">
                <a href="/reg" class="close" data-dismiss="alert">×</a>
                {% for f in form %}
                    {% if f.errors %}
                     <p>{{ f.label }}</p>
                    {% for e in f.errors %}
                        <p>{{ e }}</p>
                    {% endfor %}
                    {% endif %}
                {% endfor %}
            </div>
            </div>
        {% endif %}
        </div>
    </div>
    <div class="form-group">
        <label class="col-xs-2 control-label">Login:</label>
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon"><i class="glyphicon
glyphicon-user"></i></span>
                {{ form.username }}
            </div>
        </div>
    </div>
    <div class="form-group">
        <label class="col-xs-2 control-label">Name:</label>
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon"><i class="glyphicon
glyphicon-user"></i></span>
                {{ form.name }}
            </div>
        </div>
    </div>
    <div class="form-group">
        <label for="enterSurname" class="col-xs-2 control-
label">Surname:</label>
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon"><i class="glyphicon
glyphicon-user"></i></span>
                {{ form.surname }}
            </div>
        </div>
    </div>
    <div class="form-group">
        <label for="inputEmail" class="col-xs-2 control-label">Email:</label>
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon"><i class="glyphicon
glyphicon-envelope"></i></span>
                {{ form.email }}
            </div>
        </div>
    </div>
    <div class="form-group">
        <label for="inputPassword" class="col-xs-2 control-
label">Password:</label>
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon"><i class="glyphicon
glyphicon-pencil"></i></span>
                {{ form.password }}
            </div>
        </div>
    </div>
    <div class="form-group">
        <label for="confirmPassword" class="col-xs-2 control-label">Confirm
password:</label>
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon"><i class="glyphicon
glyphicon-pencil"></i></span>
                {{ form.password2 }}
            </div>
        </div>
    </div>

    <div class="form-group">
        <div class="col-xs-offset-2 col-xs-5">
            <button type="submit" class="btn btn-
default">Registration</button>
</form >
{% endblock %}
```

models.py

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


class Category(models.Model):
    name = models.CharField(max_length=255, unique=True)
    description = models.TextField(max_length=500)

    def __str__(self):
        return self.name


class Item(models.Model):
    name = models.CharField(max_length=255)
    date = models.DateTimeField(default=timezone.now)
    category = models.ForeignKey(Category)
    description = models.TextField(max_length=500)
    image = models.URLField(max_length=100, default="https:\\")

    def __str__(self):
        return self.name
```

db-class-test.py

```python
import pymysql as MySQLdb


class Connection:
    def __init__(self, user, password, db, host='localhost'):
        self.user = user
        self.host = host
        self.password = password
        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 = MySQLdb.connect(
                host = self.host,
                user = self.user,
                password = self.password,
                db = self.db
            )

    def disconnect(self):
        if self.__connection:
            self.__connection.close()
```

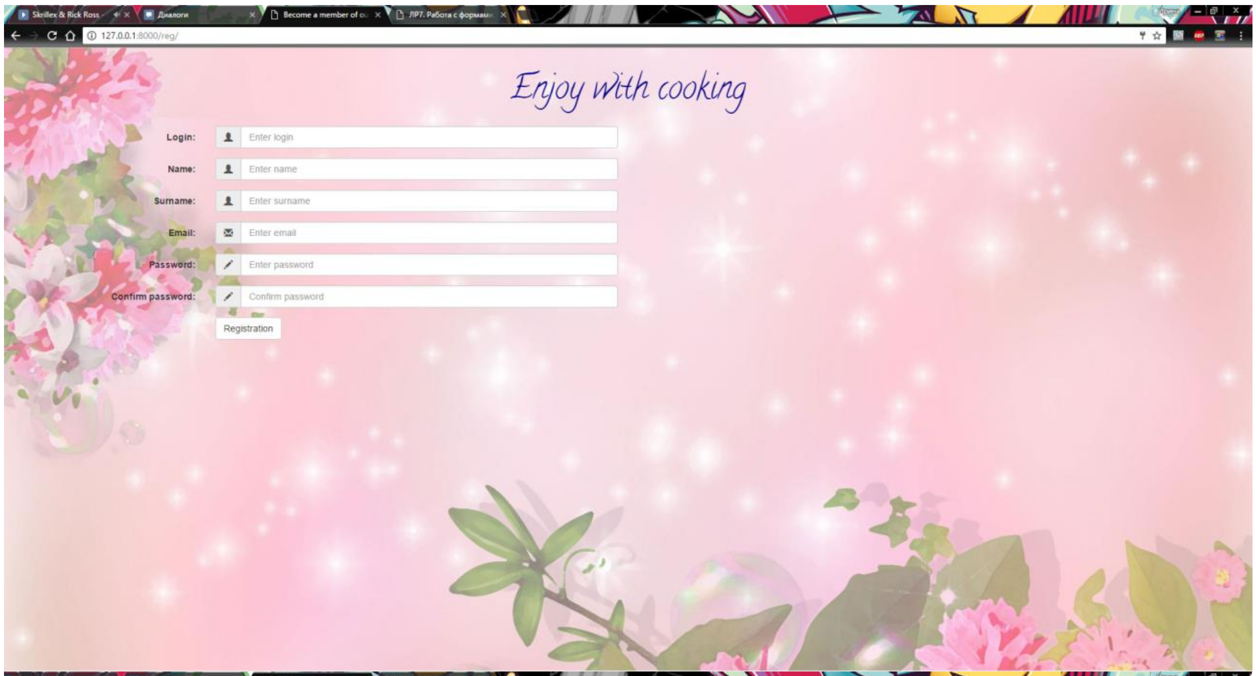
```
class Category:
    def __init__(self, db_connection, name, description, id=None,):
        self.db_connection = db_connection.connection
        self.name = name
        self.description = description
        self.__id = id

    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO category_test (name, description) values (%s,
%s);", (self.name, self.description))
        self.__id = self.db_connection.insert_id()
        self.db_connection.commit()
        c.close()

    def select_all(self):
        c = self.db_connection.cursor()
        c.execute("SELECT * from category_test")
        items = c.fetchall()
        c.close()
        return items

    def truncate_table(self):
        c = self.db_connection.cursor()
        c.execute("TRUNCATE table category_test")
        self.db_connection.commit()
        c.close()


con = Connection('dbuser', '123', 'cook')

with con:
    category = Category(con, 'New recipies', 'Cook something new')
    category.save()
    category = Category(con, 'All recipies', 'Choose what you want to do')
    category.save()
    categories = list(category.select_all())
    print(categories)
    category.truncate_table()
    categories = list(category.select_all())
    print(categories)
```

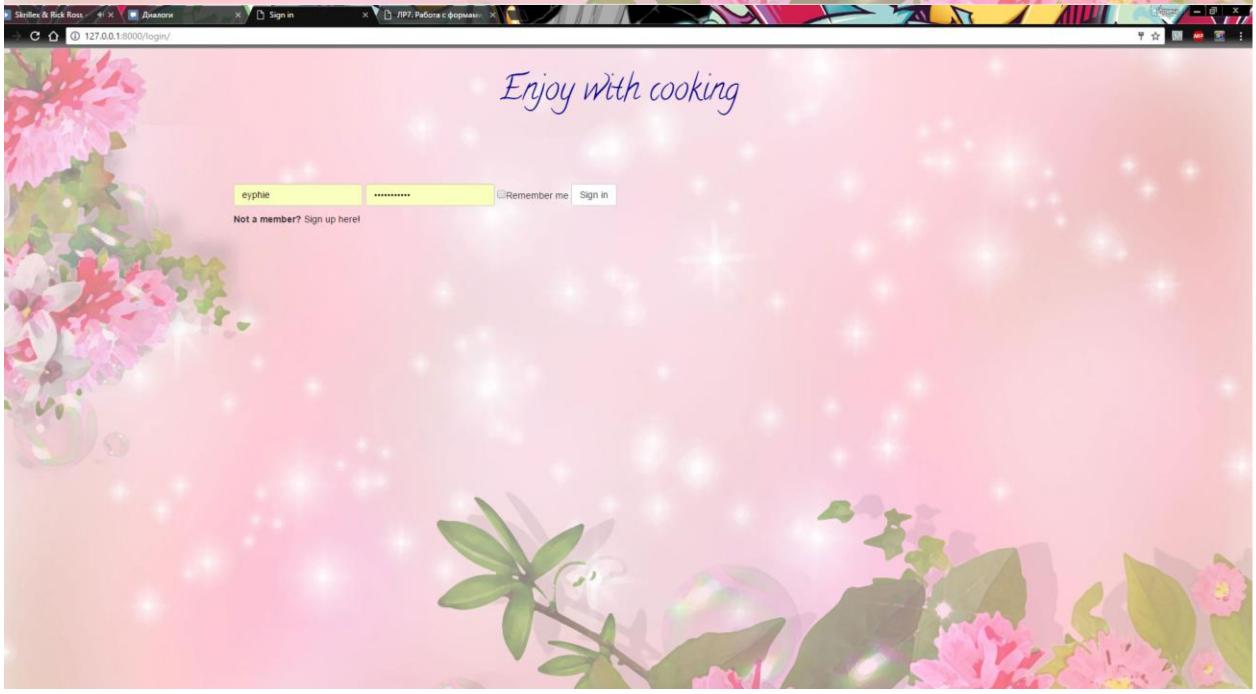
db-test.py

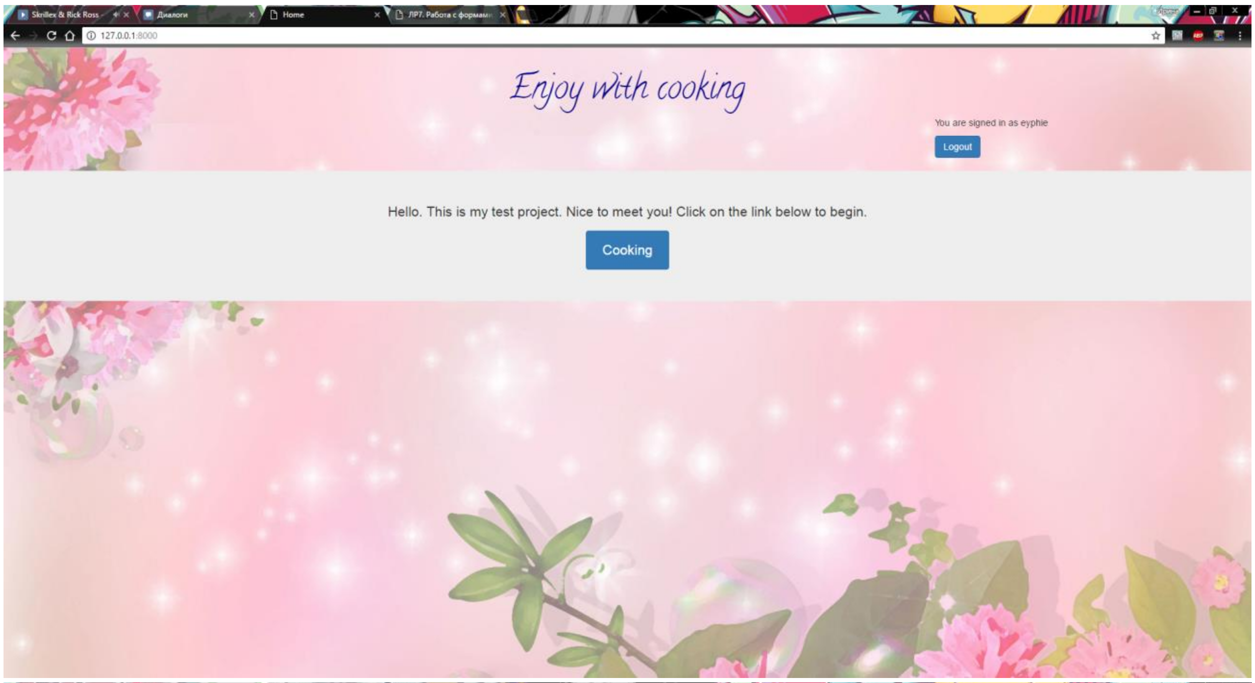
```
import pymysql as MySQLdb


db = MySQLdb.connect(user='dbuser', password='123', host='127.0.0.1',
database='cook')
c = db.cursor(MySQLdb.cursors.DictCursor)

c.execute("TRUNCATE TABLE category_test")
db.commit()

c.execute('INSERT INTO category_test (name, description) values (%s,%s),
(%s,%s)',\
          ('New recipies', 'Cook something new', 'All recipies', "Choose what
you want to do"))
db.commit()

c.execute("SELECT * FROM category_test")
categories=c.fetchall()
for category in categories:
   print("{}:{}".format(category['name'], category['description']))
```

```
c.execute("DELETE FROM category_test where id=1;")
db.commit()

c.execute ("SELECT * FROM category_test;")
print("After DELETE")
categories=c.fetchall()
for category in categories:
   print("{}:{}".format(category['name'], category['description']))

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

Результат работы программы







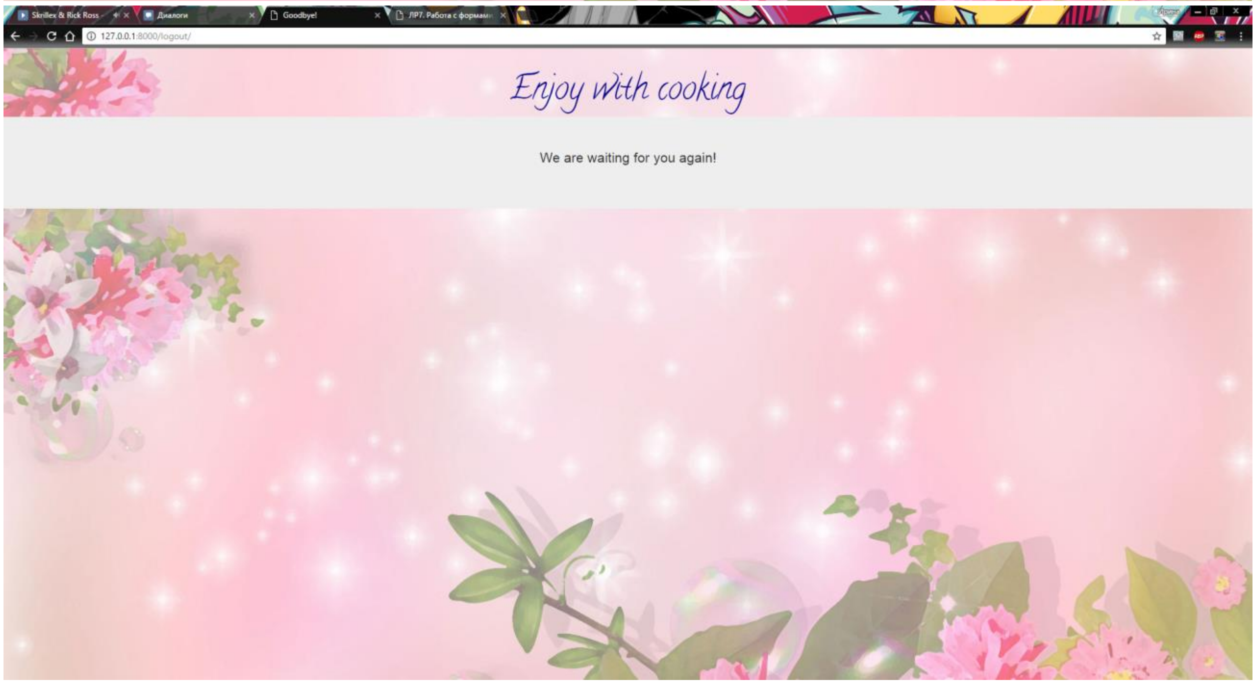
Enjoy with cooking

You are signed in as eyphie

Logout

Hello. This is my test project. Nice to meet you! Click on the link below to begin.

Cooking



Enjoy with cooking

We are waiting for you again!