

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



Лабораторные работы по курсу:

«Разработка Интернет Приложений»

ЛР6. Работа с СУБД

Исполнитель:

Студент группы РТ5-51

Житенев В.Г.

Преподаватель:

Гапанюк Ю. Е.

« ____ » _____



Москва 2017 г.

Задание и порядок выполнения

В этой лабораторной работе необходимо познакомиться с популярной СУБД MySQL, создать свою базу данных. Также нужно будет дополнить свои классы предметной области, связав их с созданной базой. После этого потребуется создать свои модели с помощью Django ORM, отобразить объекты из БД с помощью этих моделей и ClassBasedViews.

Исходный код:

settings.py

```
"""
Django settings for lab5 project.

Generated by 'django-admin startproject' using Django 1.11.6.

For more information on this file, see
https://docs.djangoproject.com/en/1.11/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/1.11/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '49!#b03k-$776(y!yk)1i==$de407791lydo(okzdn3($ca!c7'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'my_app.apps.MyAppConfig',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'lab5.urls'
```

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')]
    },
    {
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'lab5.wsgi.application'

# Database
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Password validation
# https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/1.11/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.11/howto/static-files/

```

```
STATIC_URL = '/static/'
```

lab6/urls.py

```
"""lab5 URL Configuration
```

The `urlpatterns` list routes URLs to views. For more information please see:
<https://docs.djangoproject.com/en/1.11/topics/http/urls/>

Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to `urlpatterns`: `url(r'^$', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to `urlpatterns`: `url(r'^$', Home.as_view(), name='home')`

Including another `URLconf`

1. Import the `include()` function: `from django.conf.urls import url, include`
2. Add a URL to `urlpatterns`: `url(r'^blog/', include('blog.urls'))`

```
"""
```

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

```
urlpatterns = [
```

```
    url(r'^admin/', admin.site.urls),
    url(r'', include('my_app.urls')),
    url(r'^orders', OrdersView.as_view(), name='orders_url'),
    url(r'^bd', bd, name='bd'),
    url(r'^tovar', TovarList.as_view(), name='tovar'),
    url(r'^shop1', ShopList1.as_view(), name='shop1'),
    url(r'^shop', ShopList.as_view(), name='shop'),
    url(r'^workers', WorkerList.as_view(), name='workers'),
    url(r'^main', main, name='main'),
    url(r'^login', login, name='login'),
    url(r'^success', success, name='success'),
    url(r'^registration2', registration2, name='registration2'),
    url(r'^registration', registration, name='registration'),
    url(r'^$', main, name='main'),
```

```
]
```

```
if settings.DEBUG:
```

```
    urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_URL)
```

views.py

```
from django.shortcuts import render, HttpResponseRedirect
from django.views.generic import View, ListView
from datetime import datetime
from .models import *
from .registration import *
from django.contrib.auth.models import User
from django.contrib import auth
from django.contrib.auth import authenticate
from django.contrib.auth.decorators import login_required
```

```
class OrdersView(View):
```

```
    def get(self, request):
        variable = 'Django'
        today_date = datetime.now()
        data = {
```

```

        'orders': [
            {'title': 'Первый заказ', 'id': 1},
            {'title': 'Второй заказ', 'id': 2},
            {'title': 'Третий заказ', 'id': 3}
        ]
    }
    return render(request, 'orders.html', locals())

class OrderView(View):
    def get(self, request, id):
        variable = 'Django'
        today_date = datetime.now()
        data = {
            'order': {
                'id': id
            }
        }
        return render(request, 'order.html', locals())

def main(request):
    return render(request, 'main.html', locals())
def bd(request):
    return render(request, 'bd.html', locals())

class TovarList(ListView):
    model = Tovar
    template_name = "tovar.html"

class ShopList(ListView):
    model = Shop
    template_name = "shop.html"

class ShopList1(ListView):
    model = Shop
    template_name = "assort.html"

    def get(self, request, *args, **kwargs):
        self.id = request.GET.get('id')
        return super(ShopList1, self).get(request, *args, **kwargs)
    def get_queryset(self):
        if self.id:
            return Shop.objects.filter(id=self.id)

class WorkerList(ListView):
    model = Worker
    template_name = "workers.html"

def registration(request):
    errors = {'username': '', 'password': '', 'password2': '', 'email': '',
'firstname': '', 'surname': ''}
    error_flag = False
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors['username'] = 'Введите логин'
            error_flag = True
        elif len(username) < 5:
            errors['username'] = 'Логин должен превышать 5 символов'
            error_flag = True
        elif User.objects.filter(username=username).exists():
            errors['username'] = 'Такой логин уже существует'
            error_flag = True
        password = request.POST.get('password')
        if not password:

```

```

        errors['password'] = 'Введите пароль'
        error_flag = True
    elif len(password) < 8:
        errors['password'] = 'Длина пароля должна превышать 8 символов'
    password_repeat = request.POST.get('password2')
    if password != password_repeat:
        errors['password2'] = 'Пароли должны совпадать'
        error_flag = True
    email = request.POST.get('email')
    if not email:
        errors['email'] = 'Введите e-mail'
    firstname = request.POST.get('firstname')
    if not firstname:
        errors['firstname'] = 'Введите имя'
    surname = request.POST.get('surname')
    if not surname:
        errors['surname'] = 'Введите фамилию'
    if not error_flag:
        user = User.objects.create_user(username=username, password=password,
email=email, first_name=firstname, last_name=surname)
        return HttpResponseRedirect('/login')
    return render(request, 'registration.html', locals())

def registration2(request):
    form = RegistrationForm(request.POST or None)
    if request.method == 'POST':
        if form.is_valid():
            user = User.objects.create_user(username=request.POST.get('username'),
email=request.POST.get('email'),
password=request.POST.get('password'),
first_name=request.POST.get('firstname'),
last_name=request.POST.get('surname'))
            return HttpResponseRedirect('/login')
    return render(request, 'registration2.html', {'form': form})

def login(request):
    error = ""
    username = None
    password = None
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        user = authenticate(username=username, password=password)
        if user:
            auth.login(request, user)
            return HttpResponseRedirect('/success')
        else:
            error = "Попробуй ещё раз"
    return render(request, 'login.html', locals())

@login_required()
def success(request):
    return render(request, 'success.html', locals())

def logout(request):
    auth.logout(request)
    return HttpResponseRedirect('/main/')

```

```

from django.db import models

# Create your models here.
class Users(models.Model):
    class Meta():
        db_table = "Users"
    email_user = models.EmailField(max_length=100)
    name_user = models.CharField(max_length=20)
    lastname_user = models.CharField(max_length=20)

class Tovar(models.Model):
    class Meta():
        db_table = "Tovar"
    id_tovar = models.AutoField(primary_key=True)
    name_tovar = models.CharField(max_length=50)
    type_tovar = models.CharField(max_length=50)

class Shop(models.Model):
    class Meta():
        db_table = "Table"
    id_shop = models.AutoField(primary_key=True)
    name_shop = models.CharField(max_length=20)
    adr_shop = models.CharField(max_length=100)
    assort_shop = models.ManyToManyField(Tovar)

    def get_assort(self):
        return [{'id': tovar.id_tovar, 'name_tovar': tovar.name_tovar, 'type_tovar':
tovar.type_tovar} for tovar in self.assort_shop.all()]

class Worker(models.Model):
    class Meta():
        db_table = "Worker"
    id_worker = models.AutoField(primary_key=True)
    name_worker = models.CharField(max_length=20)
    lastname_worker = models.CharField(max_length=20)
    email_worker = models.EmailField(max_length=100)
    workplace_worker = models.ForeignKey(Shop, on_delete=models.CASCADE)

```

base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Мой сайт</title>
    {% load static %}
    <link rel="stylesheet" href="static/Bootstrap/css/bootstrap.css">
    <link rel="stylesheet" href="static/Bootstrap/css/main.css">
    <link rel="stylesheet" href="static/Bootstrap/css/font-awesome.min.css">
</head>
<body>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                </button>
            <a class="navbar-brand" href="{% url 'main' %}">Welcome!</a>
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav navbar-right">
                {% block body1 %} <!-- Блок для кнопок -->

                {% endblock %}
            </ul>

```

```

        </div>
    </div>
</div>
<div id="headerwrap">
    {% block body2 %}<!--Блок для заголовков-->

    {% endblock %}
</div>
    {% block body3 %}<!--Блок для всего остального-->

    {% endblock %}
</body>
</html>

```

main.html

```

<!DOCTYPE html>
{% extends 'base.html' %}
{% load static %}
    {% block body1 %}
        <li><a href="{%url 'bd' %}">DataBase</a></li>
        <li><a href="{%url 'orders_url' %}">Orders</a></li>
        <li><a href="{%url 'login' %}">Login</a></li>
        <li><a href="{%url 'registration' %}">registration</a></li>
        <li><a href="{%url 'admin:index' %}"><i class="fa fa-bomb fa-
invert" aria-hidden="true"></i></a></li>
    {% endblock %}
{% block body2 %}
    <div class="container">
        <div class="row centered">
            <div class="col-lg-8 col-lg-offset-2">
                <h1>Выберите категорию</h1>
            </div>
            <br>
        </div>
    </div>
{% endblock %}

```



bd.html

```
<!DOCTYPE html>
{% extends 'base.html' %}
{% load static %}

{% block body1 %}
    <li class="active"><a href="{%url 'bd' %}">DataBase</a></li>
    <li><a href="{%url 'orders_url' %}">Orders</a></li>
    <li><a href="{%url 'login' %}">Login</a></li>
    <li><a href="{%url 'registration' %}">registration</a></li>
    <li><a href="{%url 'admin:index' %}"><i class="fa fa-bomb fa-
invert" aria-hidden="true"></i></a></li>
{% endblock %}

{% block body2 %}
<h1>Добро пожаловать!</h1>
<h2>Выберите категорию:</h2>
<br>
<div class="container">
    <div class = "row centered">
        <a class="navbar-brand" href="{%url 'tovar' %}">Просмотр товаров</a><br>
    </div>
    <div class = "row centered">
        <a class="navbar-brand" href="{%url 'shop' %}">Просмотр магазинов</a><br>
    </div>
    <div class = "row centered">
        <a class="navbar-brand" href="{%url 'workers' %}">Просмотр
работников</a><br>
    </div>
</div>
{% endblock %}
```

WELCOME!

DATABASE

ORDERS

LOGIN

REGISTRATION



Добро пожаловать!

Выберите категорию:

[Просмотр товаров](#)

[Просмотр магазинов](#)

[Просмотр работников](#)



товар.html

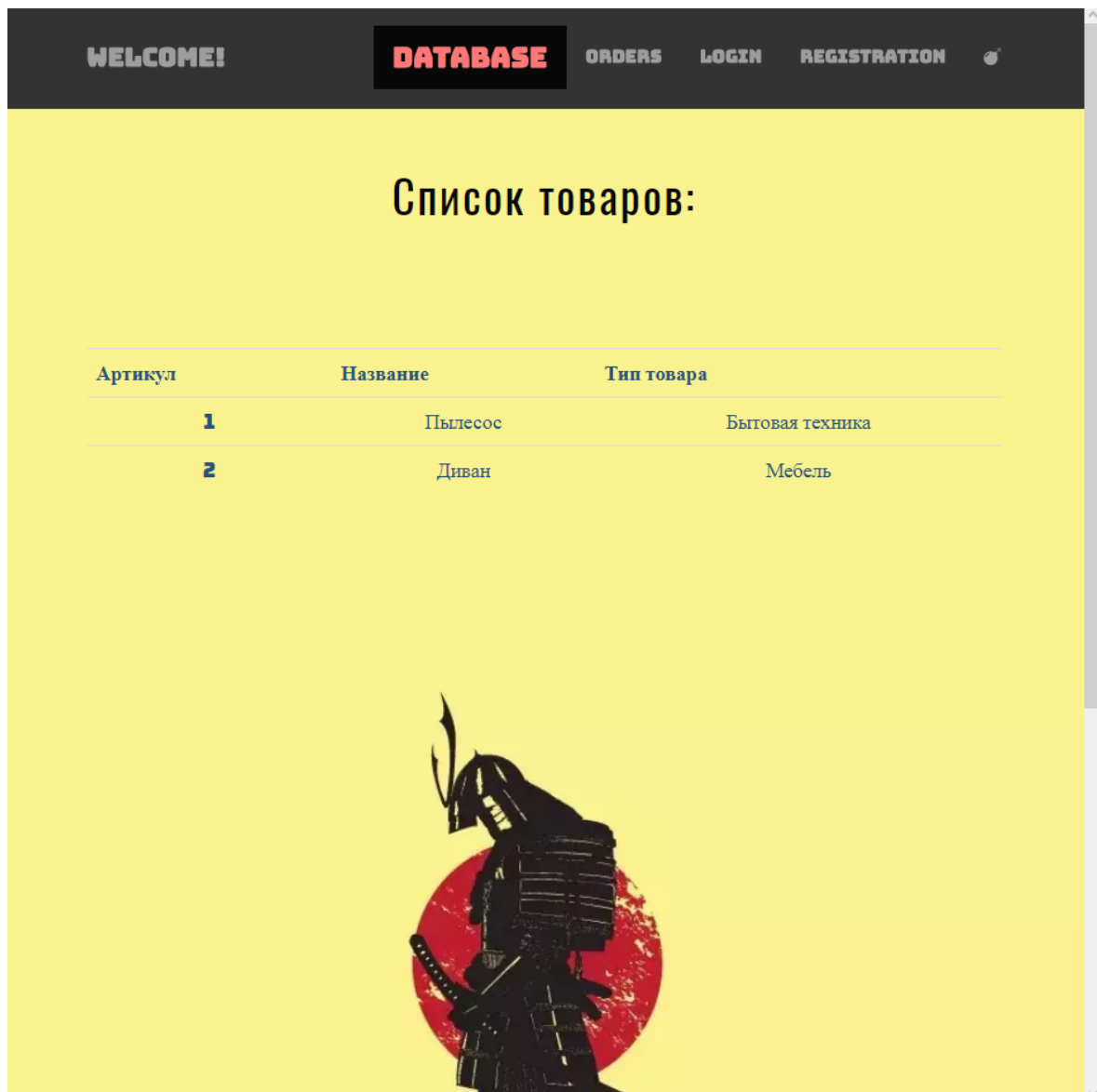
```
<!DOCTYPE html>
{% extends 'base.html' %}
{% load static %}

{% block body1 %}
    <li class="active"><a href="{%url 'bd' %}">DataBase</a></li>
    <li><a href="{%url 'orders_url' %}">Orders</a></li>
    <li><a href="{%url 'login' %}">Login</a></li>
    <li><a href="{%url 'registration' %}">registration</a></li>
    <li><a href="{%url 'admin:index' %}"><i class="fa fa-bomb fa-
invert" aria-hidden="true"></i></a></li>
{% endblock %}

{% block body2 %}
<h2>Список товаров:</h2>
    <br><br><br><br>
    <div class="container">
<table class="table table-dark">

    <tr>
        <th scope="col">Артикул</th>
        <th scope="col">Название</th>
        <th scope="col">Тип товара</th>
    </tr>
    {% for tovar in object_list %}
        <tr>
            <td>{{ tovar.id_tovar }}</td>
            <td>{{ tovar.name_tovar }}</td>
            <td>{{ tovar.type_tovar }}</td>
        </tr>
    {% endfor %}
</table>
    </div>
{% endblock %}
{% block body3 %}

{% endblock %}
```



shop.html

```
<!DOCTYPE html>
{% extends 'base.html' %}
{% load static %}
    {% block body1 %}
        <li class="active"><a href="{%url 'bd' %}">DataBase</a></li>
        <li><a href="{%url 'orders_url' %}">Orders</a></li>
        <li><a href="{%url 'login' %}">Login</a></li>
        <li><a href="{%url 'registration' %}">registration</a></li>
        <li><a href="{%url 'admin:index' %}"><i class="fa fa-bomb fa-
invert" aria-hidden="true"></i></a></li>
    {% endblock %}
{% block body2 %}
<h2>Список магазинов:</h2>
    <br><br><br><br>
    <div class="container">
<table class="table table-dark">

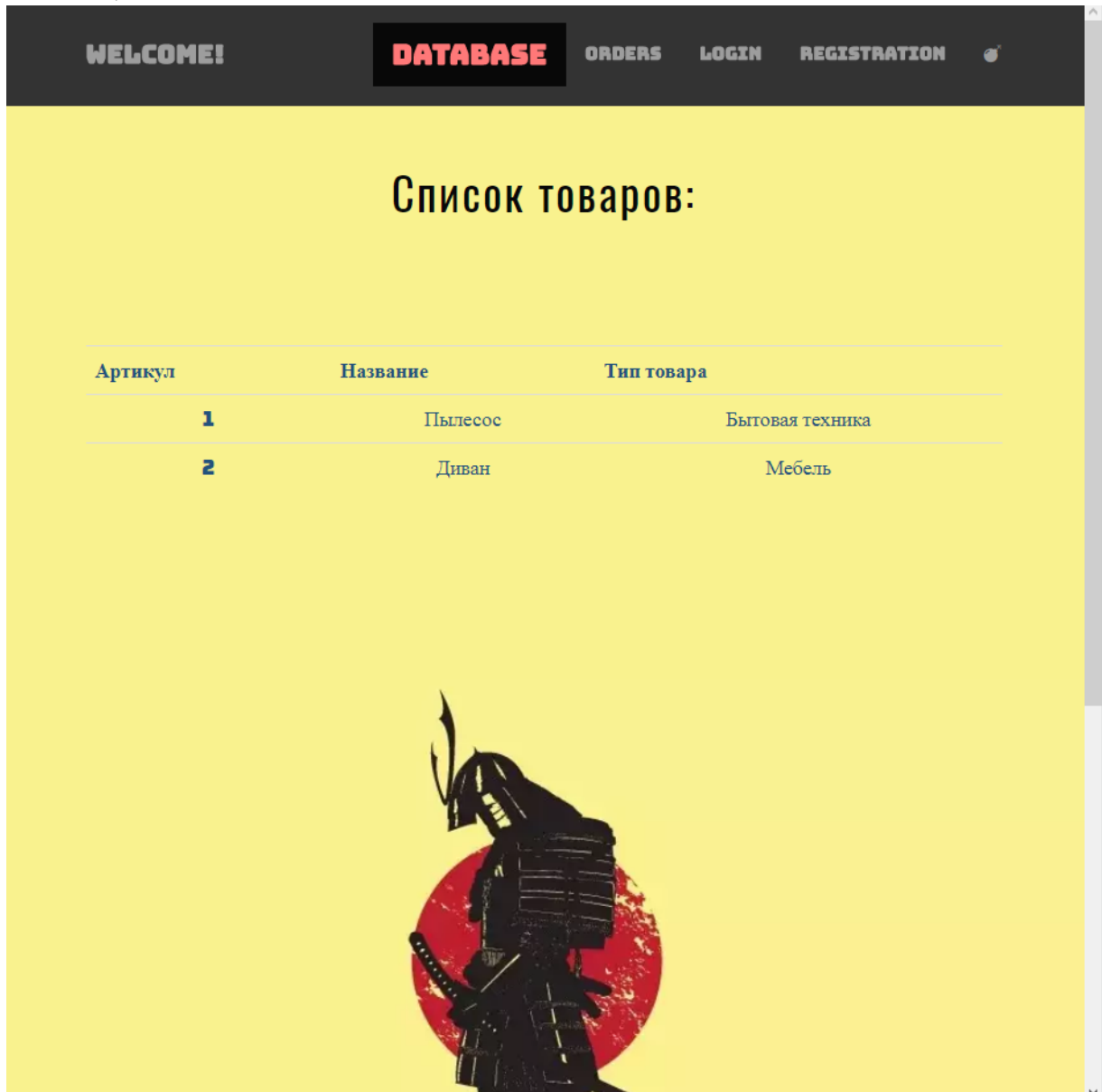
    <tr>
        <th scope="col">Название</th>
        <th scope="col">Ассортимент</th>
        <th scope="col">Адрес</th>
    </tr>
    {% for shop in object_list %}
        <tr>
            <td><a href="{% url 'shop1' %}">{{ shop.name_shop }}</a></td>
            <td>{{ shop.assort_shop.shop_id }}</td>
            <td>{{ shop.adr_shop }}</td>
```

```

    </tr>
  {% endfor %}
</table>
</div>
{% endblock %}
{% block body3 %}

{% endblock %}

```



workers.html

```

<!DOCTYPE html>
{% extends 'base.html' %}
{% load static %}

{% block body1 %}
  <li class="active"><a href="{%url 'bd' %}">DataBase</a></li>
  <li><a href="{%url 'orders_url' %}">Orders</a></li>
  <li><a href="{%url 'login' %}">Login</a></li>
  <li><a href="{%url 'registration' %}">registration</a></li>
  <li><a href="{%url 'admin:index' %}"><i class="fa fa-bomb fa-
invert" aria-hidden="true"></i></a></li>
{% endblock %}

{% block body2 %}
<h2>Список работников:</h2>
  <br><br><br><br>
  <div class="container">
<table class="table table-dark">

```

```

<tr>
  <th scope="col">Имя</th>
  <th scope="col">Фамилия</th>
  <th scope="col">Email</th>
  <th scope="col">Место работы</th>
  <th scope="col">Адрес работы</th>
</tr>
{% for worker in object_list %}
  <tr>
    <td>{{ worker.name_worker }}</td>
    <td>{{ worker.lastname_worker }}</td>
    <td>{{ worker.email_worker }}</td>
    <td>{{ worker.workplace_worker.name_shop }}</td>
    <td>{{ worker.workplace_worker.adr_shop }}</td>
  </tr>
{% endfor %}
</table>
</div>
{% endblock %}
{% block body3 %}

{% endblock %}

```

WELCOME!
DATABASE
ORDERS
LOGIN
REGISTRATION

Список работников:

Имя	Фамилия	EMAIL	Место работы	Адрес работы
Виталий	Гром	23QAD@MAIL.RU	Ашан	Пушкина 23
Евгения	Жук	Q52@GMAIL.COM	Ашан	Пушкина 23
Дмитрий	Колдун	AS34@MAIL.RU	Техносила	Суворова 15