**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет “Информатика и системы управления”

Кафедра ИУ-5 “Системы обработки информации и управления”

Курс “Разработка интернет-приложений”

Отчет по домашнему заданию

Выполнила:

Студентка группы ИУ5-53Б

Желтова Александра

Москва, 2020

**Задание:**

Создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

1. Две модели, связанные отношением один-ко-многим.
2. Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.
3. Веб-приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.
4. Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фреймворк по желанию студента.

**Расширенное задание, добавляющее +1 балл на экзамене:**

Реализация связи много-ко-многим (с возможностью редактирования данных в пользовательском интерфейсе)

**Цель домашнего задания:** изучение возможностей создания прототипа веб-приложения на основе базы данных с использованием фреймворка Django.

**Текст программы:**

**Base.html**

<!doctype html>  
<html lang="ru">  
{% load static %}  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>{% block title %}{% endblock %}</title>  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">  
</head>  
<body style="background:linear-gradient(rgba(47, 23, 15, 0.65), rgba(47, 23, 15, 0.65)), url(' {% static 'chocolate/img/bg.jpg' %} ');">  
 <div class="d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3 bg-white border-bottom shadow-sm">  
 <h5 class="my-0 mr-md-auto fw-normal">Шоколад</h5>  
 <nav class="my-2 my-md-0 mr-md-3">  
 <a class="p-2 text-dark" href="{% url 'home' %}">Все о...</a>  
 <a class="p-2 text-dark" href="{% url 'about' %}">Виды ...</a>

<a class="p-2 text-dark" href="{% url 'view' %}">Шоколад и фабрики ...</a>  
 <a class="p-2 text-dark" href="{% url 'create' %}">Добавить ...</a>  
 </nav>  
 </div>  
  
 <div class="container">  
 {% block content %}{% endblock %}  
 </div>  
</body>  
</html>

**Index.html**

{% extends 'main/base.html' %}  
  
{% block title %}  
Шоколад  
{% endblock %}  
  
{% block content %}  
<h1><font color="#FFF5EE">Все о шоколаде</font></h1>  
<p><font color="#F4A460">Шокола́д — кондитерское изделие на основе масла какао, являющееся продуктом переработки какао-бобов — семян шоколадного дерева, богатых теобромином и кофеином.</p>  
  
 <p>Изделие получило широкое распространение во всём мире, став одним из самых популярных типов еды; оно часто применяется в кулинарии. Шоколад входит в состав большого количества продуктов — преимущественно десертных, таких как торты, пудинги, муссы, шоколадные пирожные и печенья. Многие конфеты наполнены или покрыты подслащённым шоколадом; также твёрдые плитки шоколада и батончики, покрытые шоколадом, едят в качестве «закуски-перекуса». Шоколад в качестве подарков выпускается в различных формах и является традиционным на некоторых праздниках, таких как Пасха и День святого Валентина. Кроме того, шоколад используется в горячих и холодных напитках, таких как шоколадное молоко и горячий шоколад.</p>  
 <p>Шоколадные изделия часто содержат ароматические добавки (кофе, спирт, коньяк, ванилин, перец, корицу), пищевые добавки (изюм, орехи, вафли, цукаты) или начинку.</font></p>  
{% endblock %}

**About.html**

{% extends 'main/base.html' %}  
  
{% block title %}  
{{ title }}  
{% endblock %}  
  
{% block content %}  
<h1><font color="#FFF5EE">Виды шоколада</font></h1>  
 {% if types %}  
 {% for el in types %}  
 <div class="alert alert-warning mt-2">  
 <h3>{{ el.title }}</h3>  
 <p>{{ el.task }}</p>  
 </div>  
 {% endfor %}  
 {% else %}  
 <p>База данных пуста</p>  
 {% endif %}  
  
<nav class="my-2 my-md-0 mr-md-3">  
 <a href="{% url 'black' %}"><font color="#FFF5EE">Чёрный...</font></a>  
 <a href="{% url 'milk' %}"><font color="#FFF5EE">Молочный ...</font></a>  
 <a href="{% url 'white' %}"><font color="#FFF5EE">Белый ...</font></a>  
</nav>  
{% endblock %}

**Create.html**

{% extends 'main/base.html' %}  
  
{% block title %}  
Добавление новых видов шоколада  
{% endblock %}  
  
{% block content %}  
 <h1>Добавление</h1>  
 <form method="post">  
 {% csrf\_token %}  
 {{ form.title }}<br>  
 {{ form.task }}<br>  
 <span>{{ error }}</span>  
 <button type="submit" class="btn btn-success">Добавить</button>  
 </form>  
{% endblock %}

**Models.py**

from django.db import models  
from django.urls import reverse  
  
  
class Factory(models.Model):  
 name = models.CharField('Название', max\_length=25)  
 place = models.TextField('Расположение')  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 verbose\_name = 'Фабрики'  
 verbose\_name\_plural = 'Фабрики'  
  
  
class Types(models.Model):  
 title = models.CharField('Вид', max\_length=20)  
 task = models.TextField('Описание')  
 factory = models.ManyToManyField(Factory, verbose\_name='Шоколад', related\_name='Types\_Factory' )  
  
 def \_\_str\_\_(self):  
 return self.title  
  
 def get\_absolute\_url(self):  
 return reverse("types\_detail", kwargs={"slug": self.url})  
  
 class Meta:  
 verbose\_name = 'Шоколад'  
 verbose\_name\_plural = 'Виды'

**Froms.py**

from .models import Types  
from django.forms import ModelForm, TextInput, Textarea  
  
  
class TypesForm(ModelForm):  
 class Meta:  
 model = Types  
 fields = ["title", "task"]  
 widgets = {  
 "title": TextInput(attrs={  
 'class': 'form-control',  
 'placeholder': 'Введите название'  
 }),  
 "task": Textarea(attrs={  
 'class': 'form-control',  
 'placeholder': 'Введите описание'  
 }),  
 }

**Urls.py**

from django.urls import path  
from . import views  
  
urlpatterns = [  
 path('', views.index, name='home'),  
 path('types', views.about, name='about'),  
 path('create', views.create, name='create'),  
 path('black', views.black, name='black'),  
 path('milk', views.milk, name='milk'),  
 path('white', views.white, name='white'),

path('view', views.TypesView.as\_view(), name='view')  
]

**Views.py**

from django.shortcuts import render, redirect  
from .models import Types  
from .forms import TypesForm  
  
  
def index(request):  
 return render(request, 'main/index.html')  
  
  
def about(request):  
 types = Types.objects.order\_by('id')  
 """[:3]"""  
 return render(request, 'main/about.html', {'title': 'Виды шоколада', 'types': types})  
  
  
def create(request):  
 error = ''  
 if request.method == 'POST':  
 form = TypesForm(request.POST)  
 if form.is\_valid():  
 form.save()  
 return redirect('about')  
 else:  
 error = 'Форма не верна'  
  
 form = TypesForm()  
 context = {  
 'form': form,  
 'error': error  
 }  
 return render(request, 'main/create.html', context)  
  
  
def black(request):  
 return render(request, 'main/black.html')  
  
  
def milk(request):  
 return render(request, 'main/milk.html')  
  
  
def white(request):  
 return render(request, 'main/white.html')

class TypesView(View):  
 def get(self, request):  
 types = Types.objects.all()  
 return render(request, 'main/types\_list.html', {'types\_list': types})

**Types\_list.html**

{% extends 'main/base.html' %}  
  
{% block title %}  
Связь многие-ко-многим  
{% endblock %}  
  
{% block content %}  
<h1><font color="#FFFF00" size="30">Фабрики производства</font></h1>  
{% for types in types\_list %}  
<ul class="list-group">  
 <li class="list-group-item">  
 <h3>{{ types.title }}</h3>  
 <table>  
 <tr><th>Фабрики</th><th>Расположение</th></tr>  
 {% for factory in types.factory.all %}  
 <tr><td>{{ factory.name }}</td><td>{{ factory.place }}</td></tr>  
 {% endfor %}  
 </table>  
 </li>  
</ul>  
{% endfor %}  
{% endblock %}

**Экранные формы сайта:**



    

