МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э.Баумана

Отчет по домашнему заданию

по курсу «Разработка интернет-приложений»

Создание прототипа веб-приложения с использованием фреймворка Django на основе базы данных

Подготовил

Ионов С.А.

ИУ5-52Б

1. Описание задания

Создать 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.
5. Текст программы
6. **Файл settings.py (в проекте)**

"""

Django settings for mysite project.

Generated by 'django-admin startproject' using Django 3.1.5.

For more information on this file, see

https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.1/ref/settings/

"""

from pathlib import Path

import os

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = '-\_kd2v1u6wn4xa3-b0x!+dgok#7x0qlo\*i5q\_lb=%us+th(%tx'

# 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',

    'food.apps.FoodConfig',

]

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 = 'mysite.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [],

        '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 = 'mysite.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': BASE\_DIR / 'db.sqlite3',

    }

}

# Password validation

# https://docs.djangoproject.com/en/3.1/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/3.1/topics/i18n/

LANGUAGE\_CODE = 'ru-RU'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC\_URL = '/static/'

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

1. **Файл urls.py (в проекте)**

from django.contrib import admin

from django.urls import path, include

from django.conf.urls.static import static

from django.conf import settings

urlpatterns = [

    path('admin/', admin.site.urls),

    path('food/', include('food.urls')),

] + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

1. **Файл apps.py (в приложении)**

from django.apps import AppConfig

class FoodConfig(AppConfig):

    name = 'food'

    verbose\_name = "Еда"

1. **Файл models.py (в приложении)**

from django.db import models

from mysite import settings

from django.utils.translation import ugettext\_lazy

class Category(models.Model):

    name = models.CharField(max\_length=30, verbose\_name="Название")

    def \_\_str\_\_(self):

        return self.name

    class Meta:

        verbose\_name = ugettext\_lazy("Категория")

        verbose\_name\_plural = ugettext\_lazy("Категории")

class Dish(models.Model):

    category = models.ForeignKey(Category, on\_delete=models.CASCADE, verbose\_name="Категория")

    name = models.CharField(max\_length=30, verbose\_name="Название")

    structure = models.CharField(max\_length=200, verbose\_name="Состав")

    price = models.PositiveIntegerField(default=0, verbose\_name="Цена")

    img = models.ImageField(upload\_to='food/images/', default='food/images/No\_image\_available.jpg', verbose\_name="Изображение")

    def \_\_str\_\_(self):

        return 'Название: {}. Состав: {}. Цена: {}.'.format(self.name, self.structure, self.price)

    class Meta:

        verbose\_name = ugettext\_lazy("Блюдо")

        verbose\_name\_plural = ugettext\_lazy("Блюда")

1. **Файл views.py (в приложении)**

from django.shortcuts import render

from .models import Category, Dish

from django.http import Http404

def index(request):

    category\_list = Category.objects.all()

    context = {'category\_list': category\_list}

    return render(request, 'food/index.html', context)

def dish(request, category\_id):

    category = Category.objects.get(pk=category\_id)

    category\_list = Category.objects.all()

    context = {'category': category, 'category\_list': category\_list}

    return render(request, "food/dish.html", context)

1. **Файл index.html (в приложении)**

<!DOCTYPE html>

{% load static %}

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Кафе "Восток"</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW" crossorigin="anonymous"></script>

        <link rel="stylesheet" type="text/css" href="{% static 'food/css/style.css' %}">

    </head>

    <body>

        <header>

            <nav class="navbar navbar-expand-lg navbar-dark navstyle">

                <div class="container-xxl">

                  <a class="navbar-brand" href="{% url 'index' %}">Восток</a>

                  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

                    <span class="navbar-toggler-icon"></span>

                  </button>

                  <div class="collapse navbar-collapse" id="navbarSupportedContent">

                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">

                      <li class="nav-item dropdown">

                        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

                          Меню

                        </a>

                        <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

                            {% for category in category\_list %}

                                <li><a class="dropdown-item" href="{% url 'dish' category.id %}">{{ category.name }}</a></li>

                            {% endfor %}

                        </ul>

                      </li>

                    </ul>

                  </div>

                </div>

              </nav>

        </header>

        <section class="main">

          <div class="py-3 container-xxl">

            <p>

              Мы рады приветствовать Вас на нашем сайте!<br>

              Здесь Вы найдёте всю необходимую информацию о кафе "Восток".

            </p>

          </div>

        </section>

        <footer class="container-xxl my-5 pt-5 text-muted text-small">

          <hr>

          <p class="mb-1">© 2020 Все права защищены</p>

        </footer>

    </body>

</html>

1. **Файл dish.html (в приложении)**

<!DOCTYPE html>

{% load static %}

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>{{ category.name }}</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW" crossorigin="anonymous"></script>

        <link rel="stylesheet" type="text/css" href="{% static 'food/css/style.css' %}">

    </head>

    <body>

        <header>

            <nav class="navbar navbar-expand-lg navbar-dark navstyle">

                <div class="container-xxl">

                  <a class="navbar-brand" href="{% url 'index' %}">Восток</a>

                  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

                    <span class="navbar-toggler-icon"></span>

                  </button>

                  <div class="collapse navbar-collapse" id="navbarSupportedContent">

                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">

                      <li class="nav-item dropdown">

                        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

                          Меню

                        </a>

                        <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

                          {% for category in category\_list %}

                                <li><a class="dropdown-item" href="{% url 'dish' category.id %}">{{ category.name }}</a></li>

                          {% endfor %}

                        </ul>

                      </li>

                    </ul>

                  </div>

                </div>

              </nav>

        </header>

        <section class="py-3 container-xxl">

          <div>

            <p class="h2">{{ category.name }}</p>

            {% if category.dish\_set.all %}

              <table class="table" >

                  <thead>

                      <tr>

                          <th scope="col">Название блюда</th>

                          <th scope="col">Цена за порцию, руб.</th>

                          <th scope="col">Состав</th>

                          <th scope="col">Изображение</th>

                      </tr>

                  </thead>

                  <tbody class="style\_rows">

                      {% for dish in category.dish\_set.all %}

                        <tr>

                            <td>{{ dish.name }}</td>

                            <td align="center">{{ dish.price }}</td>

                            <td>{{ dish.structure }}</td>

                            <td><img src="{{ dish.img.url }}" width="200" alt="Изображение"/></td>

                        </tr>

                      {% endfor %}

                  </tbody>

              </table>

            {% else %}

              <div>

                <p class="text-info" style="font-size: 16pt;">Раздел находится в разработке</p>

              </div>

            {% endif %}

          </div>

        </section>

        <footer class="container-xxl text-muted text-small">

          <hr>

          <p>© 2020 Все права защищены</p>

        </footer>

    </body>

</html>

1. **Файл urls.py (в приложении)**

from django.urls import path

from . import views

urlpatterns = [

    path('', views.index, name='index'),

    path('<int:category\_id>/', views.dish, name='dish'),

]

1. **Файл admin.py (в приложении)**

from django.contrib import admin

from .models import Category, Dish

from django.utils.safestring import mark\_safe

from django.utils.http import urlencode

from django.urls import reverse

from django.utils.html import format\_html

class DishAdmin(admin.ModelAdmin):

    readonly\_fields = ["preview"]

    list\_display = ("name", "structure", "price", "preview", )

    list\_filter = ("category", )

    search\_fields = ("name\_\_startswith", )

    def preview(self, obj):

        return mark\_safe(f'<img src="{obj.img.url}" height="100">')

    preview.short\_description = "Превью"

class CategoryAdmin(admin.ModelAdmin):

    list\_display = ("name", "view\_dishes\_link")

    def view\_dishes\_link(self, obj):

        count = obj.dish\_set.count()

        url = (

            reverse("admin:food\_dish\_changelist")

            + "?"

            + urlencode({"category": f"{obj.id}"})

        )

        return format\_html('<a href="{}">Количество блюд: {}</a>', url, count)

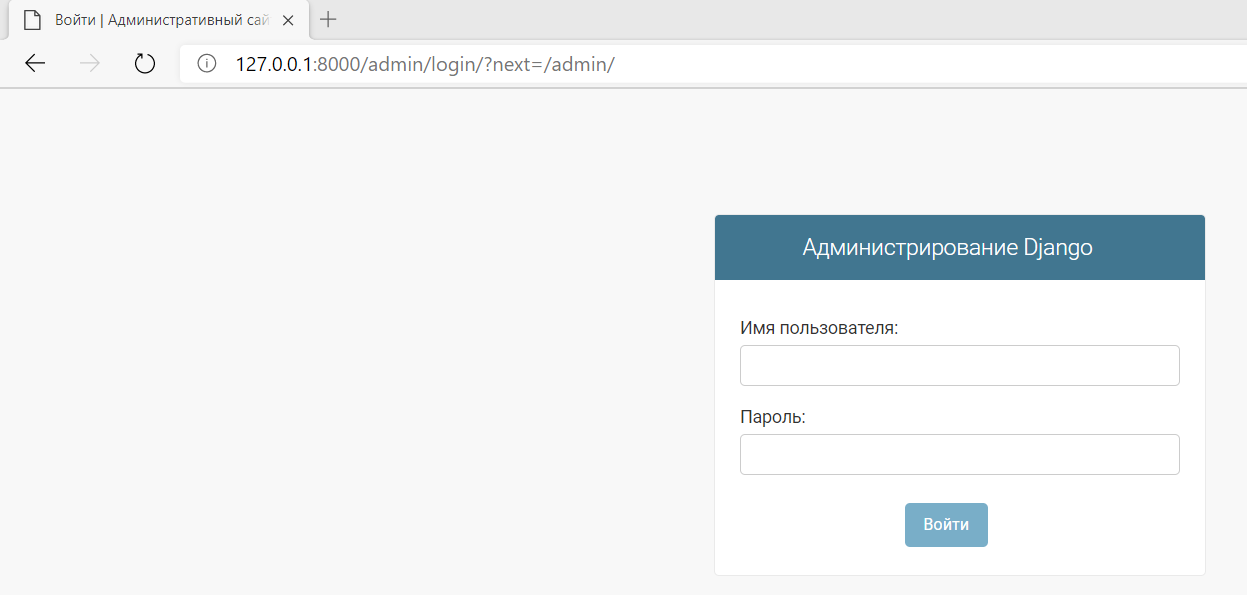
    view\_dishes\_link.short\_description = "Блюда"

admin.site.register(Category, CategoryAdmin)

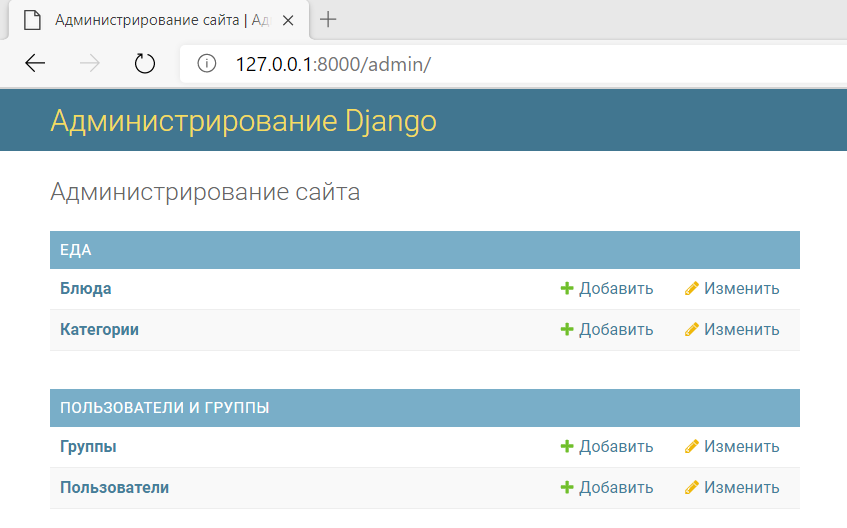
admin.site.register(Dish, DishAdmin)

1. Экранные формы с примерами выполнения программы
2. **Добавление категорий и блюд**

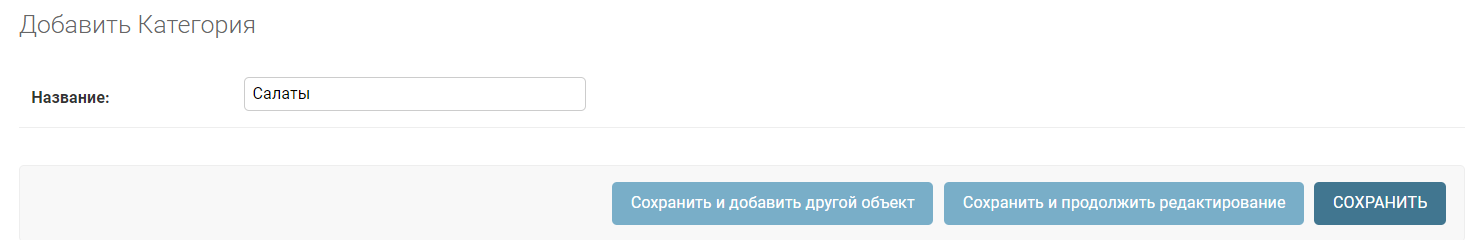
При входе в admin:



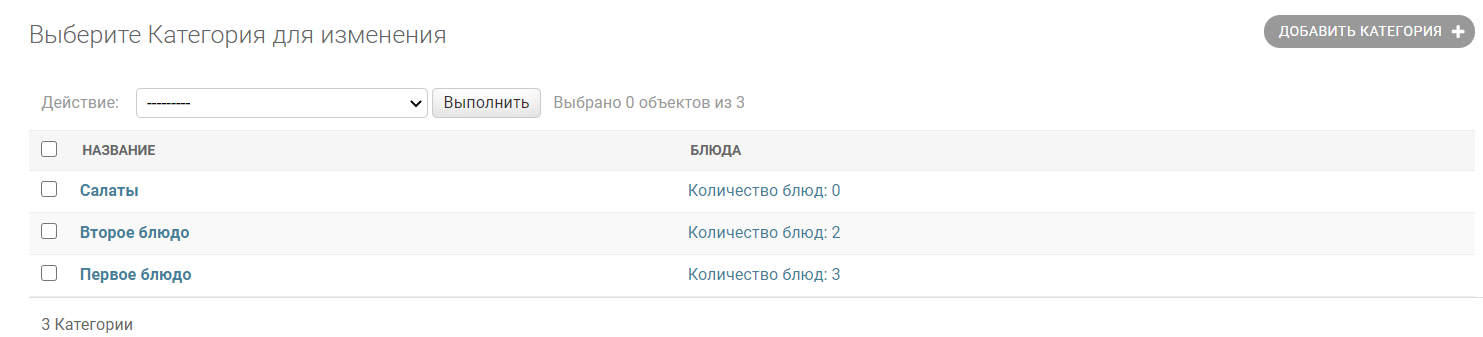
Можно добавить категорию, нажав на «Добавить» в «Категории»:



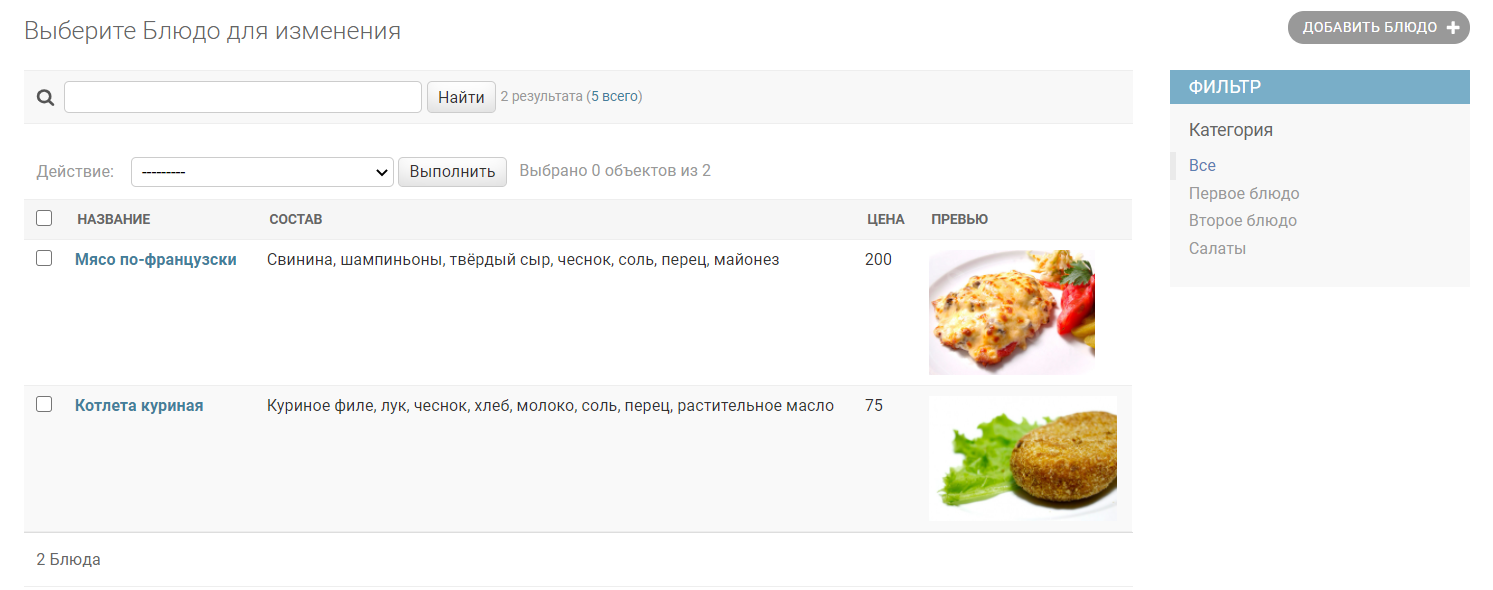
Вводим «Салаты» и нажимаем «Сохранить»:



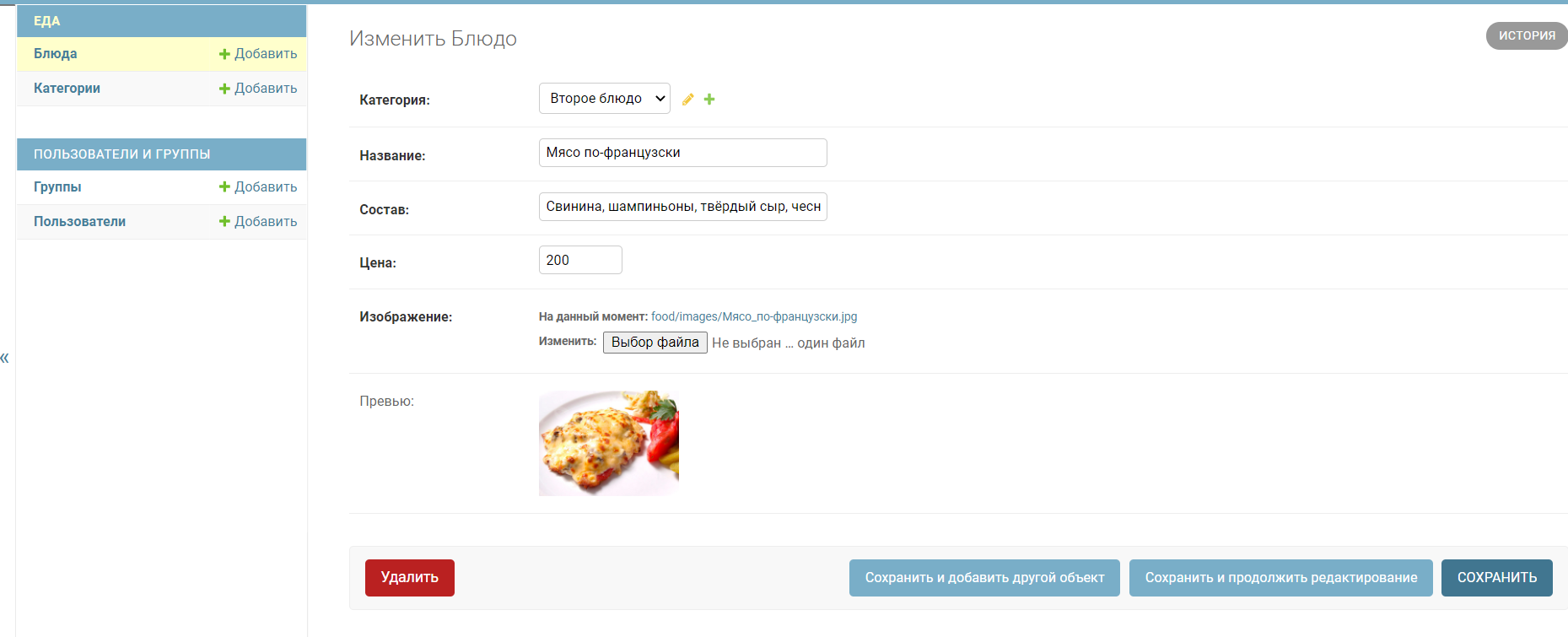
После этого категории выглядят следующим образом:



Можно нажать, например, на «2 блюда» и отобразить все блюда для категории «Второе блюдо»:

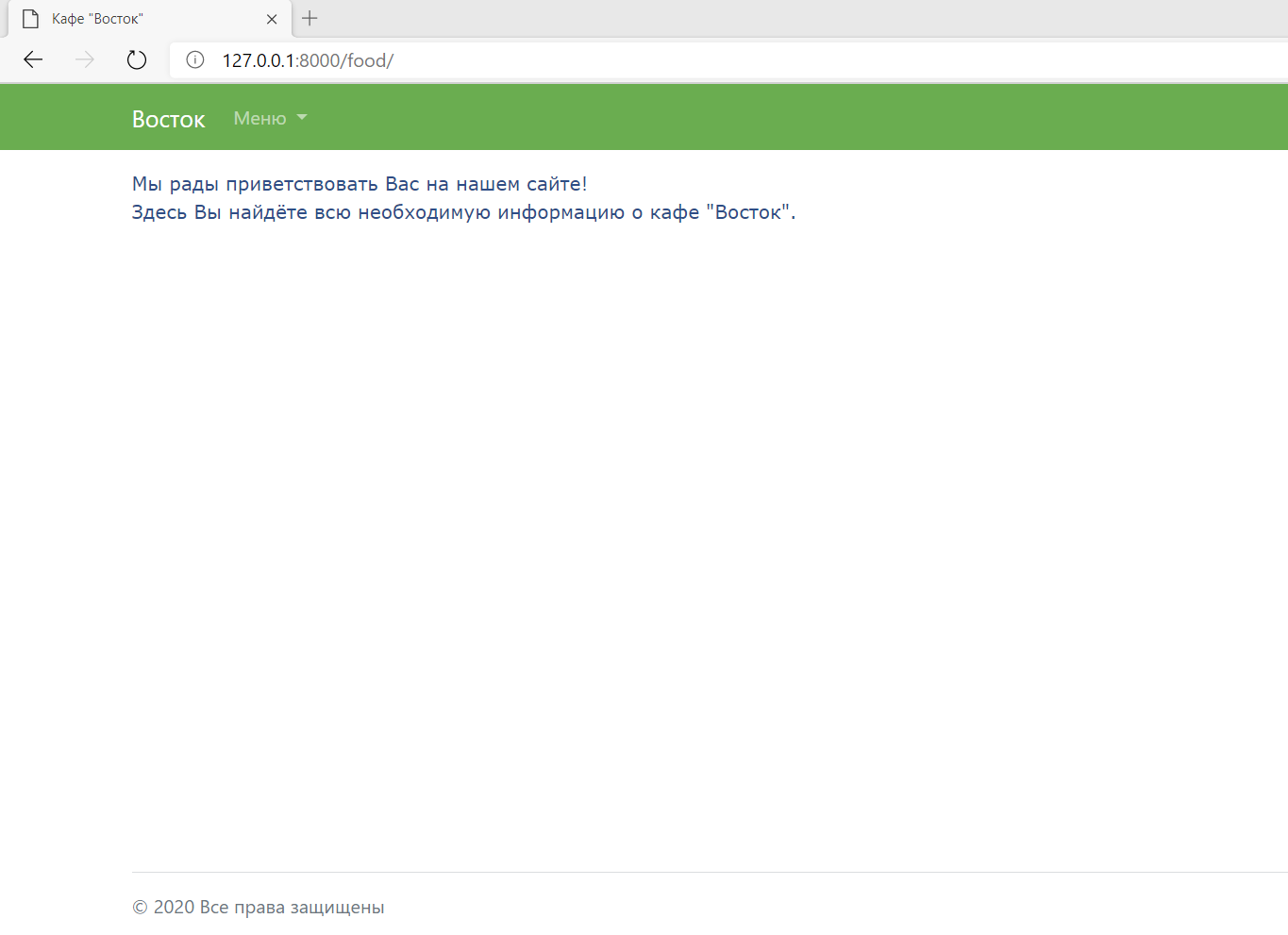


Аналогично можно добавлять или редактировать блюда:

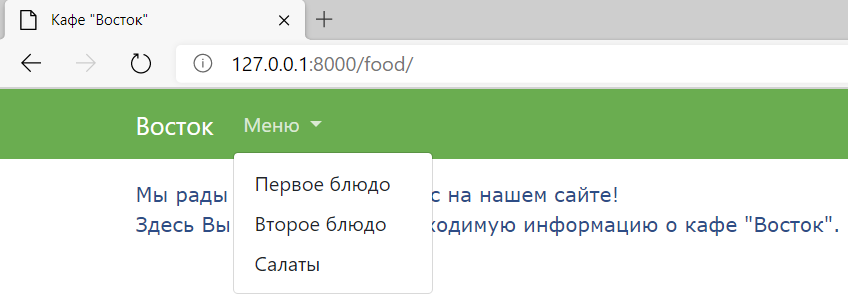


1. **Просмотр содержимого веб-приложения**

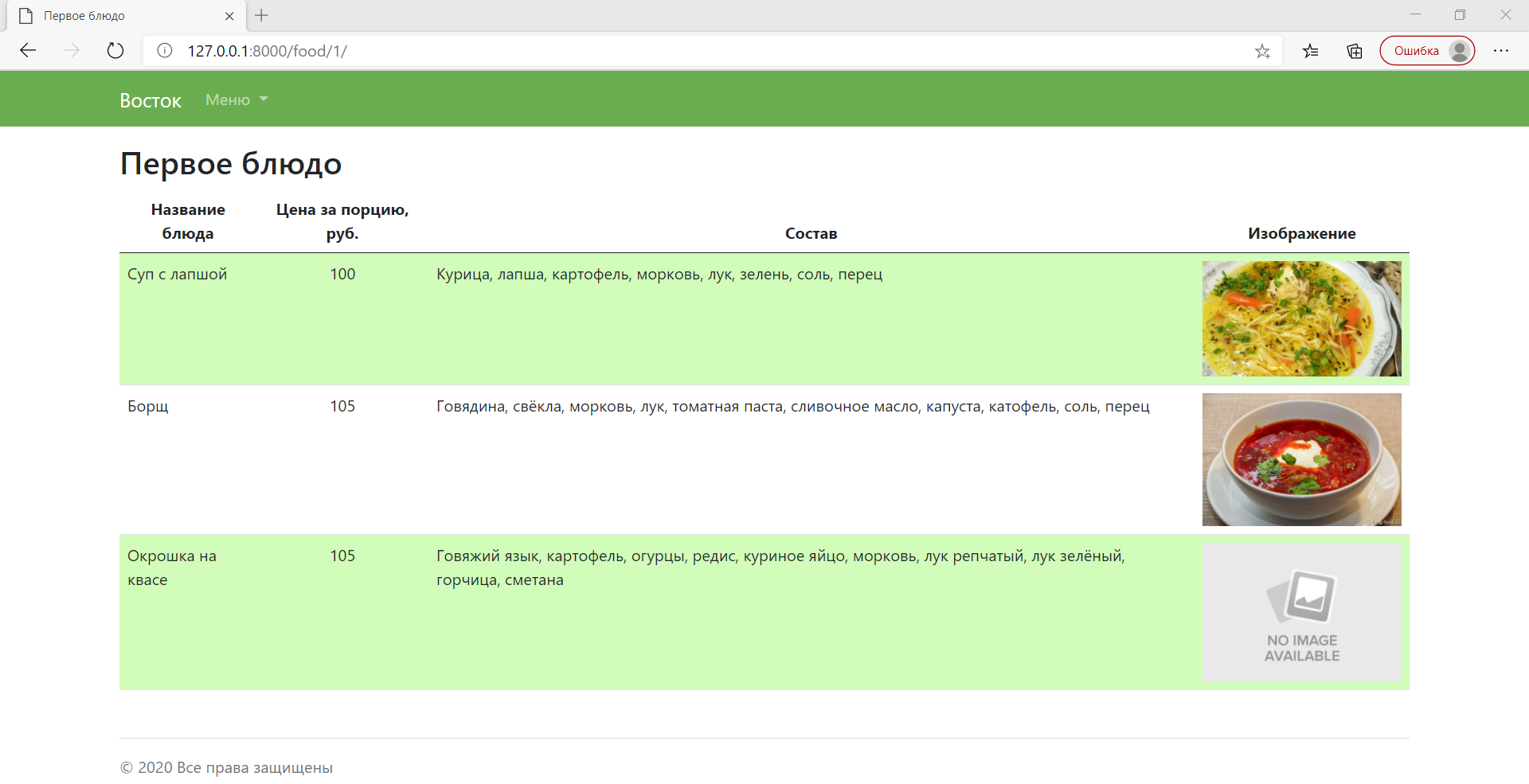
Главная страница:



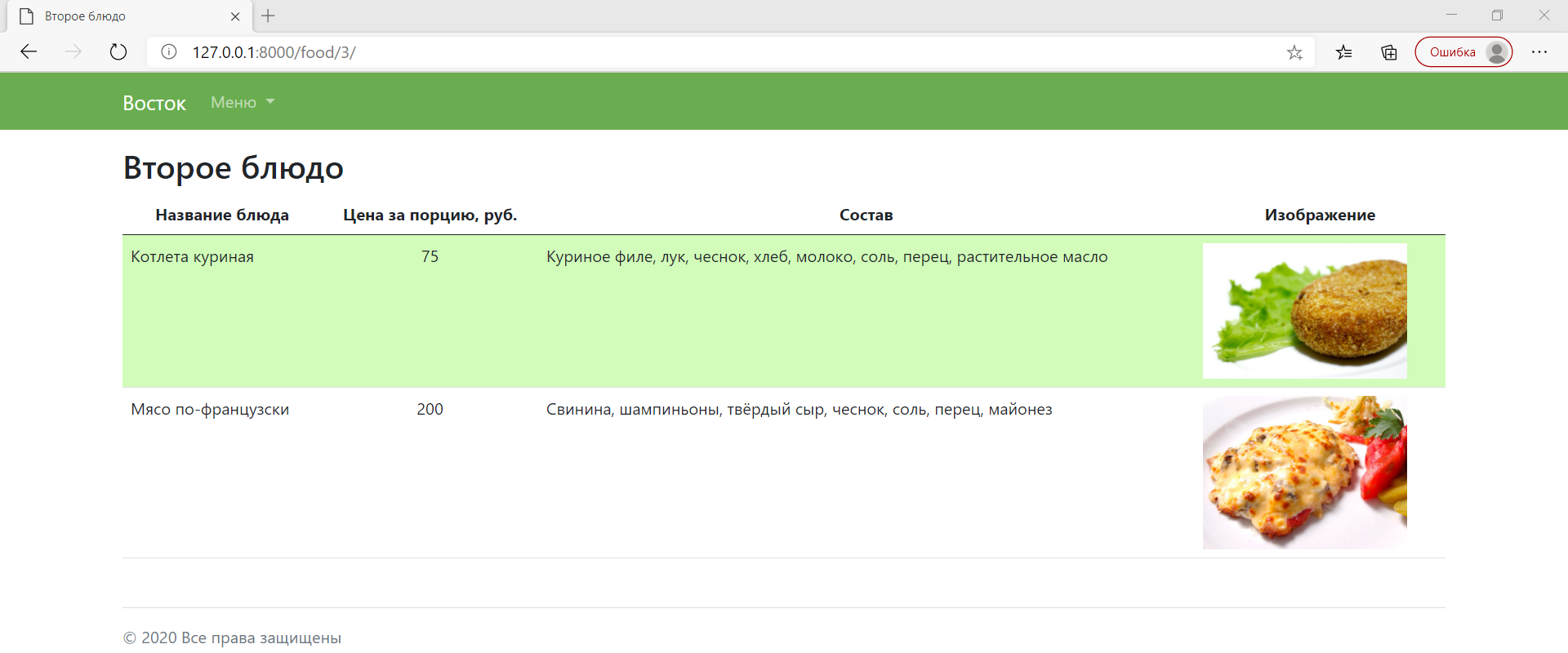
При нажатии на «Меню»:



При нажатии на «Первое блюдо»:



При нажатии на «Второе блюдо»:



При нажатии на «Салаты»:

