



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

Курс «Разработка интернет-приложений»

Отчет по домашней работе

Выполнил:
студент группы ИУ5-54Б

Ли М.В.

Преподаватель:

Гапанюк Ю.Е

Описание задания:

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.

Расширенные задания, добавляющие +1 балл на экзамене:

1. Реализация домашнего задания с использованием фреймворка для разработки SPA-приложений (React, Angular, ...).
2. Реализация связи много-ко-многим (с возможностью редактирования данных в пользовательском интерфейсе) и развертывание приложения на облачном сервисе (например, heroku).
3. Реализация в отчете (пункт 3 стандартного задания) графика на основе данных отчета с использованием библиотек JavaScript (например, <https://c3js.org/>) и развертывание приложения на облачном сервисе (например, heroku).
4. Подготовка черновика статьи по тематике курса (тематика статьи согласовывается с преподавателем).

Текст программы:

Manage.py

```
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'HomeWork.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
```

```

        "available on your PYTHONPATH environment variable? Did you "
        "forget to activate a virtual environment?"
    ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()

```

urls.py

```

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('carshowroom.urls'))
]

```

settings.py

```

import os
from pathlib import Path
import django_heroku

# 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 = 'k3$nk68^w@apizh((kbr(wajgrvngx3v5vq715yi*9oa1=k%y'

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

ALLOWED_HOSTS = ['showtime3.herokuapp.com']

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'carshowroom'
]

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 = 'HomeWork.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 = 'HomeWork.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'

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/'
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
django_heroku.settings(locals())

```

wsgi.py

```

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'HomeWork.settings')

application = get_wsgi_application()

```

asgi.py

```

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'HomeWork.settings')

application = get_asgi_application()

```

Carshowroom

admin.py

```

from django.contrib import admin
from .models import Dealer, Car_model, Supplier

admin.site.register(Dealer)

admin.site.register(Car_model)

admin.site.register(Supplier)

application = get_asgi_application()

```

apps.py

```

from django.apps import AppConfig

class CarshowroomConfig(AppConfig):
    name = 'carshowroom'

```

forms.py

```

from django import forms
from .models import Supplier

class ArticleForm(forms.ModelForm):
    class Meta:
        model = Supplier
        fields = '__all__'

    def __init__(self, *args, **kwargs):

```

```

super().__init__(*args,**kwargs)
for field in self.fields:
    self.fields[field].widget.attrs['class'] = 'form-control'

```

models.py

```

from django.db import models
from django.urls import reverse
# Create your models here.
class Supplier(models.Model):
    name = models.CharField('Название поставщика ', max_length=50)
    field = models.CharField('Поставляемый товар ', max_length=500)
    adress = models.CharField('Адрес поставщика ', max_length=500,
default="")
    def __str__(self):
        return self.name

class Dealer(models.Model):
    name = models.CharField('Марка автомобиля ', max_length=50)
    number = models.CharField('Контактный номер ', max_length=50, default=1)
    suppliers= models.ManyToManyField(Supplier, blank=True)
    def __str__(self):
        return self.name

class Car_model(models.Model):
    dealer = models.ForeignKey(Dealer, on_delete=models.CASCADE)
    name = models.CharField('Название модели', max_length=50)
    quantity = models.IntegerField('Количество', max_length=1000)
    price = models.IntegerField('Цена', max_length=100000)
    description = models.CharField('Описание', max_length=1000)
    def __str__(self):
        return self.name

```

urls.py

```

from django.urls import path
from . import views

app_name = 'carshowroom'
urlpatterns = [
    path('', views.cars, name="home"),
    path('report/', views.report, name="report"),
    path('suppliers/', views.supply, name="supply"),
    path('edit-page', views.edit_page, name="edit_page"),
    path('update-page/<int:pk>', views.update_page, name="update_page"),
    path('delete-page/<int:pk>', views.delete_page, name="delete_page"),
]

application = get_asgi_application()

```

views.py

```

import operator
from django.views.generic import View, TemplateView
from django.http import JsonResponse
from django.shortcuts import render, get_object_or_404, redirect
from .models import Dealer
from .models import Car_model
from .models import Supplier
from .forms import ArticleForm
from django.urls import reverse

def index(request):
    auto_list = Dealer.objects.all()
    context = {'auto_list': auto_list}

```

```

        return render(request, 'carshowroom/main.html', context)

def cars(request):
    auto = Car_model.objects.all().select_related('dealer')
    dealers = Dealer.objects.all()
    dealersstr = []
    for d in dealers:
        dealersstr.append(d)
    order = {key: i for i, key in enumerate(dealersstr)}
    auto_model = sorted(auto, key=lambda auto: order.get(auto.dealer, 0))
    return render(request, 'carshowroom/main.html', {'auto_model':
auto_model})

def report(request):
    auto = Car_model.objects.all().select_related('dealer')
    dealers = Dealer.objects.all()
    dealersstr = []
    for d in dealers:
        dealersstr.append(d)
    order = {key: i for i, key in enumerate(dealersstr)}
    auto_model = sorted(auto, key=lambda auto: order.get(auto.dealer, 0))
    return render(request, 'carshowroom/view.html', {'auto_model':
auto_model})

def supply(request):
    dealers = Dealer.objects.all().prefetch_related('suppliers')
    return render(request, 'carshowroom/supply.html', {'dealers': dealers})

def edit_page(request):
    success = False
    if request.method == 'POST':
        form = ArticleForm(request.POST)
        if form.is_valid():
            form.save()
            success = True
    list_suppliers=Supplier.objects.all()
    context = {
        'list_suppliers': list_suppliers,
        'form': ArticleForm(),
        'success': success
    }
    return render(request, 'carshowroom/edit-page.html', context)

def update_page(request, pk):
    get_article = Supplier.objects.get(pk=pk)
    if request.method == 'POST':
        form = ArticleForm(request.POST, instance=get_article)
        if form.is_valid():
            form.save()
    context = {
        'get_article': get_article,
        'update': True,
        'form': ArticleForm(instance=get_article),
    }
    return render(request, 'carshowroom/update-page.html', context)

def delete_page(request, pk):
    get_article = Supplier.objects.get(pk=pk)
    get_article.delete()
    return redirect(reverse('carshowroom:edit_page'))

```

Templates

Base.html

```
<!doctype html>
<html lang="en">
<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 href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKhr8RbDVddVHyTfAAsrekwKmP1"
crossorigin="anonymous">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.min.css"
integrity="sha512-
/zs32ZEJh+/EO2N1b0PEdoA10JkdC3zJ8L5FTiQu82LR9S/rOQNfQN7U59U9BC12swNeRAz3HSzIL
2vpp4fv3w==" crossorigin="anonymous" />
  <script src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.0-
beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65KgF800JFdrafW"
crossorigin="anonymous"></script>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-light rounded" aria-
label="Twelfth navbar example">
    <div class="container-fluid">
      <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarsExample10" aria-
controls="navbarsExample10" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse justify-content-md-center"
id="navbarsExample10">
        <ul class="navbar-nav">
          <li class="nav-item active">
            <form class="container-fluid justify-content-start">
              <button class="btn btn-outline-dark" type="button"
onClick='location.href="{% url 'carshowroom:home' %}"'>AutoCenterCity</button>
            </form>
          </li>
          <li class="nav-item">
            <form class="container-fluid justify-content-start">
              <button class="btn btn-outline-dark" type="button"
onClick='location.href="{% url 'carshowroom:supply' %}"'>Поставщики</button>
            </form>
          </li>
          <li class="nav-item">
            <form class="container-fluid justify-content-start">
              <button class="btn btn-outline-dark" type="button"
onClick='location.href="{% url 'carshowroom:report' %}"'>Отчет об
ассортименте</button>
            </form>
          </li>
          <li class="nav-item">
            <form class="container-fluid justify-content-start">
              <button class="btn btn-outline-dark" type="button"
onClick='location.href="{% url 'carshowroom:edit_page' %}"'>Редактирование
```



```

данных</button>
        </form>
    </li>
</ul>
</div>
</div>
</nav>
{% block content %}{% endblock %}
</body>
</html>

```

Edit-page.html

```

{% extends 'carshowroom/base.html' %}

{% block title %}
Редактирование данных
{% endblock %}

{% block content %}
<div class="container">
    <h3>Редактирование данных</h3>

    <button type="button" class="btn btn-lg btn-outline-dark mx-auto mb-3"
data-bs-toggle="modal" data-bs-target="#exampleModal">
        Добавить нового поставщика
    </button>

    {% if success %}
        <p style="color:red">Новый поставщик добавлен</p>
    {% endif %}
    <table class="table table-bordered">
        <thead>
            <tr>
                <th scope="col">Название компании</th>
                <th scope="col">Поставляемый продукт</th>
                <th scope="col">Адрес поставщика</th>
                <th scope="col">Редактирование</th>
                <th scope="col">Удаление</th>
            </tr>
        </thead>
        <tbody>
            {% for i in list_suppliers%}
            <tr>
                <th scope="row">{{i.name}}</th>
                <td>{{i.field}}</td>
                <td>{{i.adress}}</td>
                <td><a class="link-dark" href="{% url 'carshowroom:update_page'
i.id %}">Редактировать</a></td>
                <td><a class="link-dark" href="{% url 'carshowroom:delete_page'
i.id %}">Удалить</a></td>
            </tr>
            {% endfor %}
        </tbody>
    </table>

    <!-- Modal -->
    <div class="modal fade" id="exampleModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="exampleModalLabel">Поставщик</h5>
                    <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
                </div>

```

```

        <div class="modal-body">
            <form id="add form" method="post">
                {% csrf_token %}
                {{form}}
            </form>
        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Закрыть</button>
            <button form="add form" type="submit" class="btn btn-
primary">Добавить</button>
        </div>
    </div>
</div>
{% endblock %}

```

Main.html

```

{% extends 'carshowroom/base.html' %}

{% block title %}
Главная страница
{% endblock %}
{% block content %}
    <div class="container">
        <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">
            {% for model in auto_model %}
                <div class="col mt-4">
                    <div class="card shadow">
                        <div class="card-body">
                            <h4 class="p-1 mb-0 bg-dark text-white card-title
mx-auto text-center">{{ model.dealer.name }}</h4>
                            <p class="p-1 bg-dark text-white mx-auto text-
center">{{model.name}}</p>
                            <p><strong>В наличии:</strong><span class="font-
monospace"> {{model.quantity}}</span></p>
                            <p><strong>Стартовая цена:</strong> <span
class="font-monospace">{{model.price}}$</span></p>
                            <p><strong>Описание:</strong> <span class="font-
monospace">{{model.description}}</span></p>
                        </div>
                    </div>
                </div>
            {% endfor %}
        </div>
    </div>
{% endblock %}

```

Supply.html

```

{% extends 'carshowroom/base.html' %}

{% block title %}
Поставщики
{% endblock %}
{% block content %}
    <div class="container">
        <div class="row">
            {% for d in dealers %}
                <div class="col mt-4">
                    <div class="card shadow">
                        <div class="card-body">
                            <h5 class="p-3 mb-0 bg-dark text-white card-title

```

```

mx-auto text-center">{{ d.name }}</h5>
        <p class = "p-2 mb-auto mx-auto text-
center"><strong>Контактный номер:</strong> {{d.number}}</p>
        <div class="mb-0 mx-auto ">
            {% for supplier in d.suppliers.all %}
                <p class = "p-1 mb-0 bg-dark text-white mx-
auto text-center"><strong class="pb-1 ">Поставщик: {{ supplier
}}</strong></p>
                <p class = "ps-2 mb-1 mx-auto text-
center"><strong class="pb-1 ">Поставляемый продукт:</strong> {{
supplier.field }}</p>
                <p class = "ps-2 mb-1 mx-auto text-
center"><strong class="pb-1 ">Адрес поставщика:</strong> {{ supplier.address
}}</p>
                {% endfor %}
                <button type="button" class="w-100 btn btn-lg
btn-outline-dark" onClick='location.href="{% url 'carshowroom:edit_page'
%}">Редактировать</button>
            </div>
        </p>
    </div>
</div>
</div>
    </div>
    {% endfor %}
</div>
</div>
{% endblock %}

```

Update-page.html

```

{% extends 'carshowroom/base.html' %}

{% block title %}
Главная страница
{% endblock %}

{% block content %}
<div class="container ">
    <h3>Изменение</h3>

    {% if success %}
        <p style="color:red">Новый поставщик добавлен</p>
    {% endif %}
    {% if update %}
        <p>Редактирование</p>
        <form id="update form" method="post">
            {% csrf_token %}
            {{form}}
        </form>
        <button form="update form" type="submit" class="btn btn-outline-dark mx-
auto mt-2">ОБНОВИТЬ</button>
        <button type="button" class="btn btn-outline-secondary mx-auto mt-2
data-bs-dismiss="modal" onClick='location.href="{% url
'carshowroom:edit_page' %}">Вернуться</button>
    {% else %}
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th scope="col">Название компании</th>
                    <th scope="col">Поставляемый продукт</th>
                    <th scope="col">Адрес поставщика</th>
                    <th scope="col">Редактирование</th>
                    <th scope="col">Удаление</th>
                </tr>
            </thead>

```

```

<tbody>
  {% for i in list_suppliers%}
  <tr>
    <th scope="row">{{i.name}}</th>
    <td>{{i.field}}</td>
    <td>{{i.adress}}</td>
    <td><a href="{% url 'carshowroom:update_page' i.id
%}">Редактировать</a></td>
    <td><a href="">Удалить</a></td>
  </tr>
  {% endfor %}
</tbody>
</table>
{% endif %}
</div>
{% endblock %}

```

View.html

```

{% extends 'carshowroom/base.html' %}

{% block title %}
Отчеты
{% endblock %}

{% block content %}
<div class="container">
  <h1>Отчет об автомобилях: {{ header }}</h1>
  {% if auto_model %}
    <table class="table table-hover table-striped table-bordered border-
dark table-sm mx-auto">
      <tr class="text-center"><th>Модель
автомобиля</th><th>Производитель</th><th>Номер
производителя</th><th>Цена</th><th>Количество</th><th>Описание</th></tr>
      {% for model in auto_model %}
        <tr class="text-center"><td>{{ model.name }}</td><td>{{
model.dealer }}</td><td>{{ model.dealer.number }}</td><td>{{ model.price
}}$</td>
          <td>{{ model.quantity }}</td><td>{{ model.description
}}</td></tr>
      {% endfor %}
    </table>
  {% else %}
    <p>Database is empty.</p>
  {% endif %}
  <canvas id="myChart"></canvas>
  <script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
  <script>
    var ctx = document.getElementById('myChart');
    var myChart = new Chart(ctx, {
      type: 'bar',
      data: {
        labels: [{% for model in auto_model %}, '{{ model.name }}', {%
endfor %}],
        datasets: [{
          label: 'Сравнение цен на модели автомобилей',
          data: [{% for model in auto_model %}, '{{ model.price }}',
{% endfor %}],
          backgroundColor: [
            'rgba(255, 99, 132, 0.2)',
            'rgba(54, 162, 235, 0.2)',
            'rgba(255, 206, 86, 0.2)',
            'rgba(75, 192, 192, 0.2)',
            'rgba(153, 102, 255, 0.2)',
            'rgba(255, 159, 64, 0.2)',

```

```

        'rgba(255, 255, 80, 0.4)',
        'rgba(153, 102, 255, 0.2)',
        'rgba(255, 159, 64, 0.2)',
        'rgba(255, 255, 80, 0.4)',
        'rgba(255, 99, 132, 0.2)',
        'rgba(254, 162, 235, 0.2)',
        'rgba(255, 206, 86, 0.2)',
        'rgba(255, 99, 132, 0.2)',
        'rgba(0, 255, 0, 0.4)',
        'rgba(0, 255, 0, 0.4)',
        'rgba(0, 255, 0, 0.4)',

    ],
    borderColor: [
        'rgba(255, 99, 132, 0.2)',
        'rgba(54, 162, 235, 0.2)',
        'rgba(255, 206, 86, 0.2)',
        'rgba(75, 192, 192, 0.2)',
        'rgba(153, 102, 255, 0.2)',
        'rgba(255, 159, 64, 0.2)',
        'rgba(255, 255, 80, 0.4)',
        'rgba(153, 102, 255, 0.2)',
        'rgba(255, 159, 64, 0.2)',
        'rgba(255, 255, 80, 0.4)',
        'rgba(255, 99, 132, 0.2)',
        'rgba(254, 162, 235, 0.2)',
        'rgba(255, 206, 86, 0.2)',
        'rgba(255, 99, 132, 0.2)',
        'rgba(0, 255, 0, 0.4)',

    ],
    borderWidth: 1
    }]
    },
    options: {
        scales: {
            yAxes: [{
                ticks: {
                    beginAtZero: true
                }
            }]
        }
    }
    });
</script>
<canvas id="myChart2"></canvas>
<script>
    var ctx = document.getElementById('myChart2');
    var myChart2 = new Chart(ctx, {
        type: 'polarArea',
        data: {
            labels: ['', 'Impreza WRX STi III', '', 'LX', '',
'Forester','', 'Camry 3.5', '', 'Land Cruiser 200', '', 'Avalon','', 'LC500',
'', 'LX', '', 'NX'],
            datasets: [{
                label: '# of Votes',
                data: [{% for model in auto_model %},{ model.quantity
}}', {% endfor %}],
                backgroundColor: [
                    'rgba(255, 255, 255, 1)',
                    'rgba(54, 162, 235, 0.2)',

```

```

'rgba(255, 255, 255, 1)',
'rgba(75, 192, 192, 0.2)',
'rgba(255, 255, 255, 1)',
'rgba(255, 159, 64, 0.2)',
'rgba(255, 255, 255, 1)',
'rgba(153, 102, 255, 0.2)',
'rgba(255, 255, 255, 1)',
'rgba(255, 255, 80, 0.4)',
'rgba(255, 255, 255, 1)',
'rgba(254, 162, 235, 0.2)',
'rgba(255, 255, 255, 1)',
'rgba(255, 99, 132, 0.2)',
'rgba(255, 255, 255, 1)',,
'rgba(255, 255, 255, 1)',
'rgba(0, 255, 0, 0.4)',

],

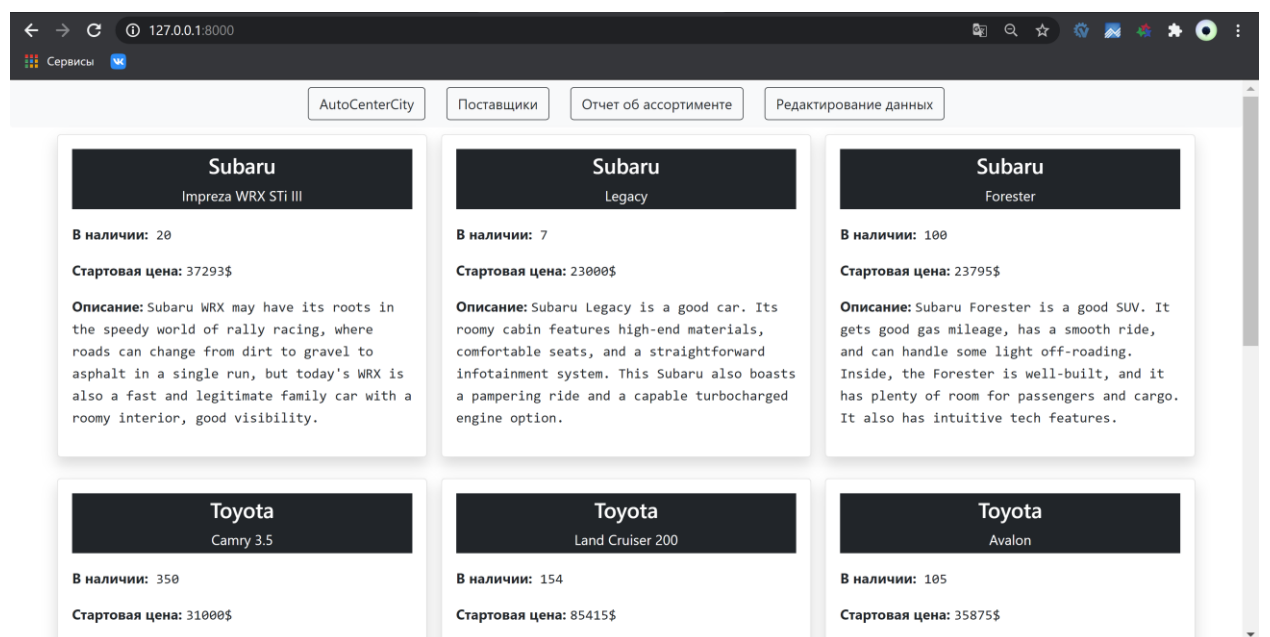
}],

});
</script>
</div>

{% endblock %}

```

Экранные формы с примерами выполнения программы:



AutoCenterCity

Поставщики

Отчет об ассортименте

Редактирование данных

Отчет об автомобилях:

Модель автомобиля	Производитель	Номер производителя	Цена	Количество	Описание
Impreza WRX STi III	Subaru	785923492	37293\$	20	Subaru WRX may have its roots in the speedy world of rally racing, where roads can change from dirt to gravel to asphalt in a single run, but today's WRX is also a fast and legitimate family car with a roomy interior, good visibility.
Legacy	Subaru	785923492	23000\$	7	Subaru Legacy is a good car. Its roomy cabin features high-end materials, comfortable seats, and a straightforward infotainment system. This Subaru also boasts a pampering ride and a capable turbocharged engine option.
Forester	Subaru	785923492	23795\$	100	Subaru Forester is a good SUV. It gets good gas mileage, has a smooth ride, and can handle some light off-roading. Inside, the Forester is well-built, and it has plenty of room for passengers and cargo. It also has intuitive tech features.
Camry 3.5	Toyota	891312313	31000\$	350	The 2021 Toyota Camry is a great midsize car. Among its strengths are a simple infotainment system, a roomy cabin, a high safety score, great fuel economy, an engaging ride, and a punchy V6 engine option. Its downsides are fairly minor: Some road and wind noise permeate the cabin, and some interior materials in lower trims are not impressive.
Land Cruiser 200	Toyota	891312313	85415\$	154	Toyota Land Cruiser is a decent luxury large SUV. It boasts a powerful V8 engine, composed handling on- and off-road, and a sterling reputation for reliability. Its first- and second-row seats are comfy and spacious, and there are lots of standard convenience and safety features. The Land Cruiser also offers hefty towing ability and a sizable cargo are.
Avalon	Toyota	891312313	35875\$	105	The 2021 Avalon is a great large car. Look inside, and you'll find top-notch cabin materials and plenty of

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ ADMIN ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало · Carshowroom · Dealers · Lexus

CARSHOWROOM

Car_models + Добавить

Dealers + Добавить

Suppliers + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

Изменить dealer

ИСТОРИЯ

Марка автомобиля: Lexus

Контактный номер: 8941741772

Suppliers:

Yokohama Michelin Brembo Bosch

Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений.

Удалить

Сохранить и добавить другой объект.

Сохранить и продолжить редактирование.

СОХРАНИТЬ

← → ↺ 127.0.0.1:8000/suppliers/ Сервисы

AutoCenterCity

Поставщики

Отчет об ассортименте

Редактирование данных

Subaru

Контактный номер: 785923492

Поставщик: Michelin

Поставляемый продукт: Tire

Адрес поставщика: Moscow

Поставщик: Bosch

Поставляемый продукт: Brakes

Адрес поставщика: Germany, Berlin

Редактировать

Toyota

Контактный номер: 891312313

Поставщик: Yokohama

Поставляемый продукт: Tire

Адрес поставщика: Sakhalin

Реализация доп.задания

