**Тема**: **Разработка веб–приложения для аренды автомобилей**

Содержание

[Введение 5](#_Toc501726971)

[1. Техническое задание 6](#_Toc501726972)

[1.1 Основания для разработки 6](#_Toc501726973)

[1.2 Назначение разработки 6](#_Toc501726974)

[1.3 Требования к приложению 6](#_Toc501726975)

[Требования к функциональным характеристикам 6](#_Toc501726976)

[Требования к составу и параметрам технических средств 6](#_Toc501726977)

[Требования к информационной и программной совместимости 7](#_Toc501726978)

[2. Стадии и этапы разработки 8](#_Toc501726979)

[2.1. Анализ предметной области 8](#_Toc501726980)

[2.2. Разработка схемы базы данных 10](#_Toc501726981)

[2.3. Разработка структур данных, предназначенных для представления сущностей в оперативной памяти 12](#_Toc501726982)

[2.4. Выбор СУБД 14](#_Toc501726983)

[2.5. Разработка структуры веб приложения 15](#_Toc501726984)

[2.6. Разработка программного кода, реализующего функциональные требования к приложению 20](#_Toc501726985)

[Заключение 25](#_Toc501726986)

[Список используемых источников 26](#_Toc501726987)

[Приложение 1. Исходный код 27](#_Toc501726988)

[**Project** 27](#_Toc501726989)

[Manage.py 27](#_Toc501726990)

[Settings.py 27](#_Toc501726991)

[Urls.py 29](#_Toc501726992)

[Wsgi.py 29](#_Toc501726993)

[**MyCar** 30](#_Toc501726994)

[Admin.py 30](#_Toc501726995)

[Apps.py 30](#_Toc501726996)

[Forms.py 30](#_Toc501726997)

[Models.py 32](#_Toc501726998)

[Tests.py 33](#_Toc501726999)

[Urls.py 33](#_Toc501727000)

[Views.py 34](#_Toc501727001)

[Style.css 38](#_Toc501727002)

[**Templates** 40](#_Toc501727003)

[about.html 40](#_Toc501727004)

[autos.html 40](#_Toc501727005)

[base.html 40](#_Toc501727006)

[footer.html 41](#_Toc501727007)

[header.html 41](#_Toc501727008)

[main\_page.html 41](#_Toc501727009)

[model.html 43](#_Toc501727010)

[Operators.html 44](#_Toc501727011)

[tariff.html 45](#_Toc501727012)

[**Admin** 45](#_Toc501727013)

[Fuel/show.html 45](#_Toc501727014)

[Mark/show.html 46](#_Toc501727015)

[Model/show.html 47](#_Toc501727016)

[Operator/show.html 47](#_Toc501727017)

[Rank/show.html 48](#_Toc501727018)

[Rent/show.html 49](#_Toc501727019)

[Tariff/show.html 50](#_Toc501727020)

[admin\_add.html 51](#_Toc501727021)

[admin\_delete.html 51](#_Toc501727022)

[admin\_edit.html 51](#_Toc501727023)

[admin\_menu.html 52](#_Toc501727024)

1. Техническое задание
2. Назначение разработки

Веб-приложение предназначено для аренды автомобилей удаленным способом с устройства, имеющего доступ в глобальную сеть интернет.

1. Требования к приложению

**Требования к функциональным характеристикам**

Веб-приложение должно обеспечивать возможность выполнения перечисленных ниже функций:

1. Администрирования: добавление, удаление, редактирование записей базы данных (БД). БД содержит всю информацию, необходимую для аренды автомобиля.
2. Возможность аренды автомобиля, посредством заполнения формы заказа информацией со стороны пользователя.
3. Простая схема навигации.
4. Браузеры: Opera, Google Сhrome, Safari, Microrosft Edge, Mozilla Firefox, Internet Explorer.
5. Просмотр вспомогательной информации, такой, как например: контакты, инструкции и т.д.

**Требования к составу и параметрам технических средств**

В состав технических средств должно входить устройство, имеющее выход в глобальную сеть интернет, экран и периферийные устройства (мониторы, мыши, клавиатуры), если необходимо. Такими устройствами могут стать смартфоны, персональные компьютеры, ноутбуки. Системные требования зависят от используемого браузера.

**Требования к информационной и программной совместимости**

Требования к ОС клиента: Windows 7 и выше, Linux Ubuntu 10.0 и выше

1. Стадии и этапы разработки
   1. Анализ предметной области

В рамках данной курсовой работы рассматривается предметная область «Прокат автомобиля». Она обоснована развитием коммерческих организаций по прокату автомобилей, в которых необходим компьютеризированный учет имеющихся машин. Эта информация обширна и разрозн ена. Чтобы вести учет всех автомобилей, которые можно взять в прокат, в организации имеется потребность в структурировании данных об автомобилях. Отсутствие такой возможности приводит к проблеме утери данных и большим временным затратам на выборку данных.

Прокат автомобилей (или аренда) - весьма распространенная и востребованная услуга. В любом городе работает множество автопрокатных контор - от крупных до самых маленьких. Процедура аренды автомобиля проста. От клиента требуется, чтобы его возраст был не менее 21 года и не превышал 70 лет. Служащему арендной компании надо предъявить паспорт и водительское удостоверение (международное). К моменту заключения договора удостоверение должно быть действительно не менее 2 лет.

В стоимость арендной платы должно входить следующее:

1. неограниченный пробег автомобиля;
2. доставка клиенту автомобиля в черте города;
3. ремонт или замена автомобиля в случае технической неисправности, кроме повреждения покрышек и ветрового стекла;
4. полная страховка на случай ДТП, произошедшего не по вине клиента;
5. страховка, покрывающая ущерб, нанесенный автомобилю в ДТП по вине клиента, сверх определенной суммы (но если на момент ДТП водитель находился в состоянии алкогольного опьянения, страховка не выплачивается);
6. страховка пассажиров (кроме водителя) от несчастных случаев (себя водитель может застраховать за дополнительную плату);
7. налоги. Обычно машину доставляют с полным баком, но и вернуть в автопрокатную контору ее нужно также с полным баком
8. Автомобили застрахованы от всех рисков на условиях КАСКО и ОСАГО. В случае ДТП ответственность Клиента составляет величину залога – 200$, весь остальной ущерб, нанесённый автомобилю, покрывает страховая компания.
9. Процесс оформления аренды автомобиля прост и понятен. Требуется лишь минимальный комплект персональной информации и 15 минут свободного времени.

Разработка данного приложения должна вестись на языке программирования Python с использованием фреймворка Django. Дополнительных требований к среде разработки, а также версии среды и фреймворка не предъявляется

* 1. Разработка схемы базы данных

Первым этапом и самым главным этапом в процессе проектирования и создания базы данных, является разработка модели. Основными конструктивными элементами сущности, связи между ними и их свойства (атрибуты). База данных создаётся для информационного обслуживания пункта авто-проката. БД должна содержать данные об автомобилях, сдающихся в прокат или находящихся в прокате, о клиентах. В соответствии с предметной областью система строится с учётом следующих особенностей:

1. Каждый автомобиль сдается в рамках аренды;
2. Автомобиль может быть арендован одним клиентом.
3. Аренда оформляется на одного клиента;
4. Аренда включает тариф на прокат
5. За каждым автомобилем закрепляется оператор
6. Каждый автомобиль выпущен определенной фирмой; имеет определенное топливо и класс
7. Существует предел кол-ва автомобилей, которыми может оперировать авто-прокат.

Логическая схема базы данных представлена на рисунке 1.

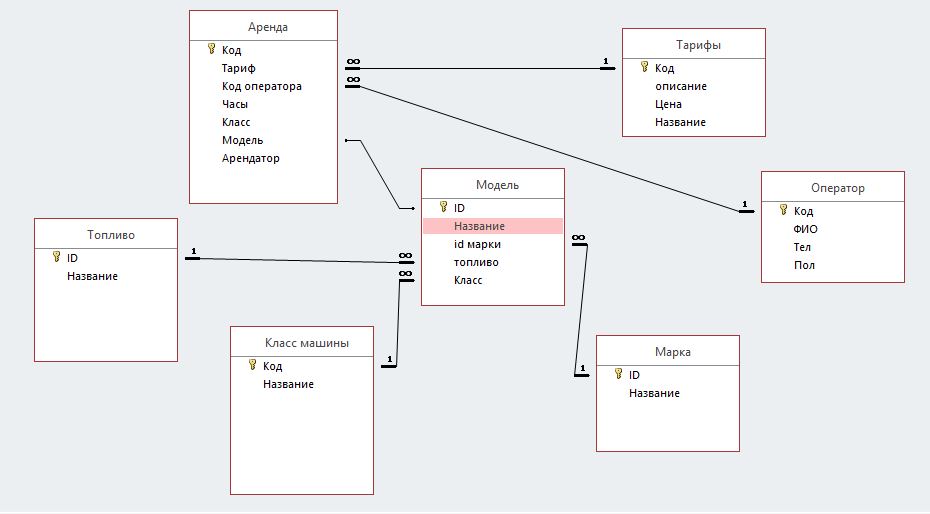


Рис. 1 - Логическая схема базы данных.

В БД входят следующие сущности

1. Аренда. Содержит всю необходимую информацию для аренды автомобиля
2. Модель автомобиля. Содержит информацию об автомобиле, который можно арендовать.
3. Марка. Содержит все марки автомобилей компании.
4. Класс автомобиля. Содержит классы автомобиля
5. Топливо. Содержит виды топлива.
6. Тарифы. Содержит всю информацию о тарифах на аренду
7. Оператор. Содержит информацию об операторе, который будет закреплен за конкретным автомобилем.

Физическая схема базы данных представлена на рисунке 2.

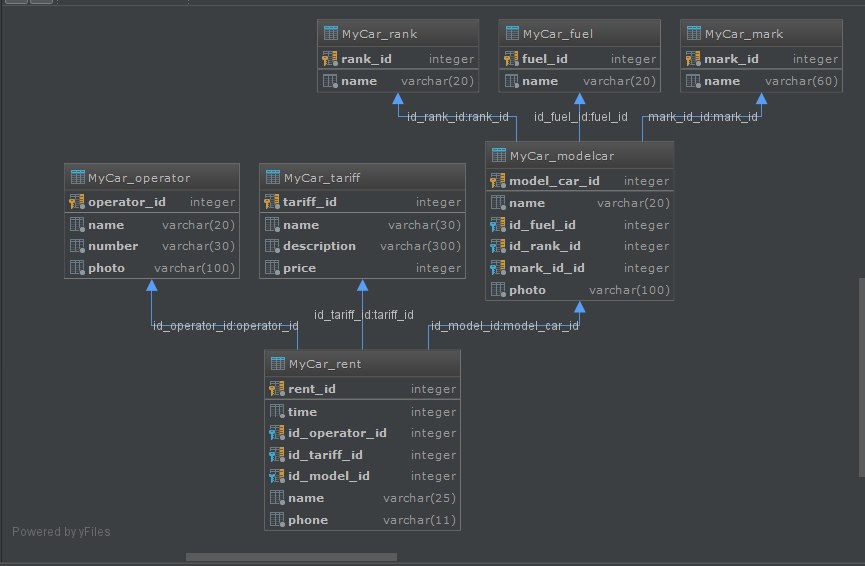


Рис. 2 - Физическая схема базы данных.

Структура данных сущностей содержится в разделе 2.3.

* 1. Разработка структур данных, предназначенных для представления сущностей в оперативной памяти

Файл models.py(Модели)

Сущность **Тариф**

class Tariff(models.Model):

tariff\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=30)

description = models.CharField(max\_length=300)

price = models.IntegerField()

Сущность **Операторы**

class Operator(models.Model):

operator\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=20)

number = models.CharField(max\_length=30)

photo = models.ImageField()

Сущность **Топливо**

class Fuel(models.Model):

fuel\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=20)

Сущность **Класс машины**

class Rank(models.Model):

rank\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=20)

Сущность **Марка**

class Mark(models.Model):

mark\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=60)

Сущность **Модель**

class ModelCar(models.Model):

model\_car\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=20)

id\_rank = models.ForeignKey(Rank, verbose\_name="Класс")

id\_fuel = models.ForeignKey(Fuel, verbose\_name="Топливо")

mark\_id = models.ForeignKey(Mark, verbose\_name="Марка")

photo = models.ImageField(blank=True, upload\_to='MyCar/media')

Сущность **Аренда**

class Rent(models.Model):

rent\_id = models.AutoField(primary\_key=True)

id\_tariff = models.ForeignKey(Tariff, verbose\_name="Тариф")

id\_model = models.ForeignKey(ModelCar, verbose\_name="Модель")

id\_operator = models.ForeignKey(Operator, verbose\_name="Оператор")

name = models.CharField(verbose\_name='Арендатор', max\_length=25, default='')

phone = models.CharField(verbose\_name='Телефон', max\_length=11, default='')

time = models.PositiveIntegerField()

* 1. Выбор СУБД

Для реализации данной курсовой работы была выбрана СУБД PostgreSQL 10.

PostgreSQL — это объектно-реляционная система управления базами данных (ОРСУБД, ORDBMS), основанная на [POSTGRES, Version 4.2](http://db.cs.berkeley.edu/postgres.html) — программе, разработанной на факультете компьютерных наук Калифорнийского университета в Беркли. В POSTGRES появилось множество новшеств, которые были реализованы в некоторых коммерческих СУБД гораздо позднее.

PostgreSQL — СУБД с открытым исходным кодом, основой которого был код, написанный в Беркли. Она поддерживает большую часть стандарта SQL и предлагает множество современных функций:

1. сложные запросы
2. внешние ключи
3. триггеры
4. изменяемые представления
5. транзакционная целостность
6. многоверсионность

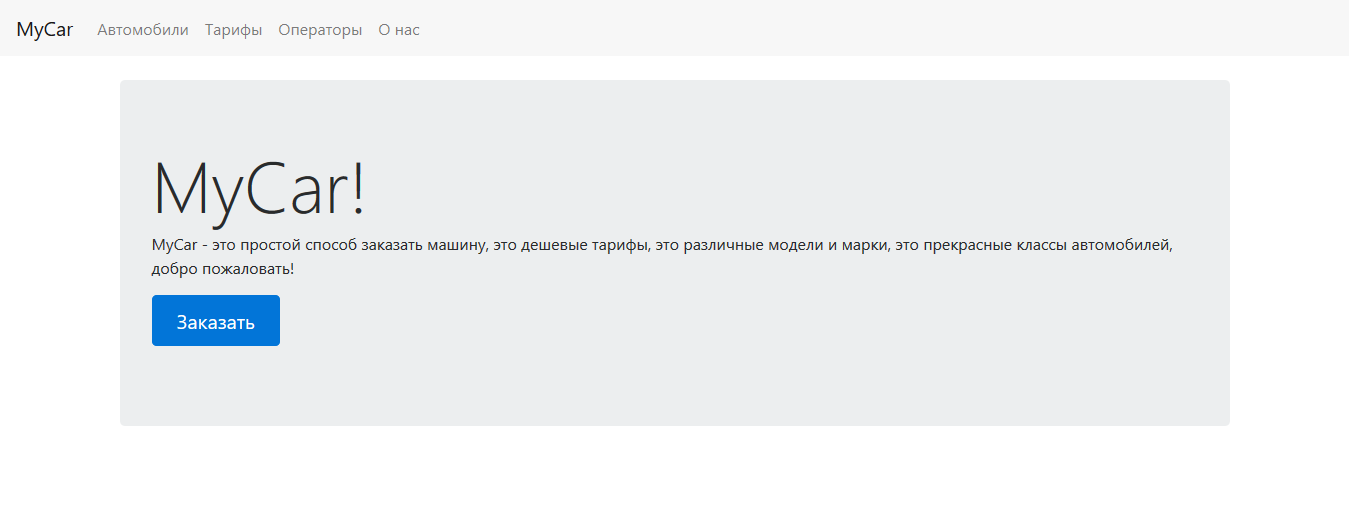
Кроме того, пользователи могут всячески расширять возможности PostgreSQL, например, создавая свои

1. типы данных
2. функции
3. операторы
4. агрегатные функции
5. методы индексирования
6. процедурные языки

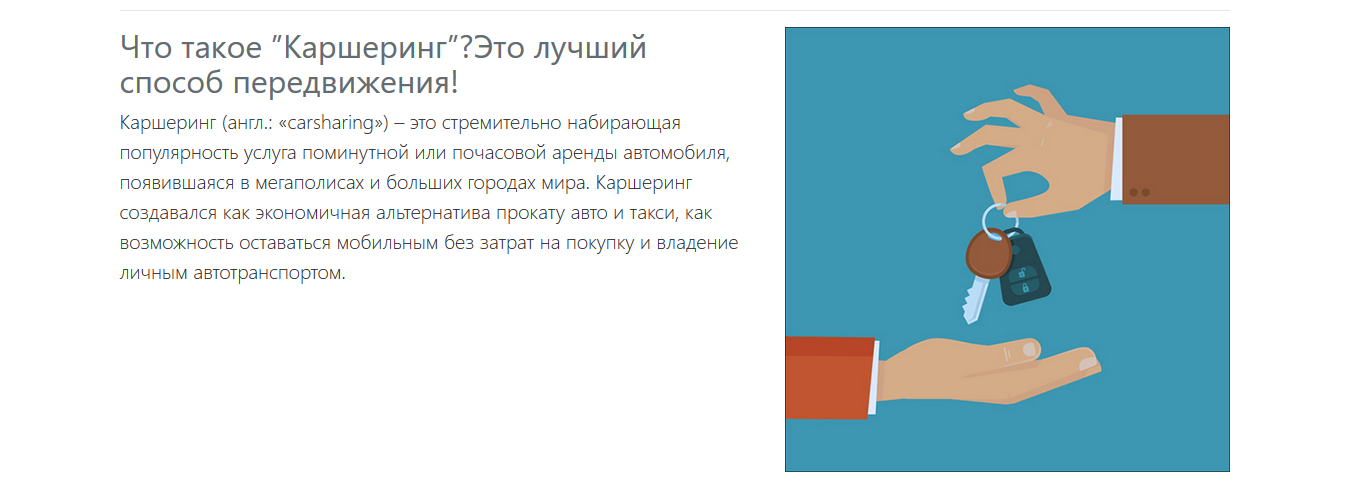
А благодаря свободной лицензии, PostgreSQL разрешается бесплатно использовать, изменять и распространять всем и для любых целей — личных, коммерческих или учебных.

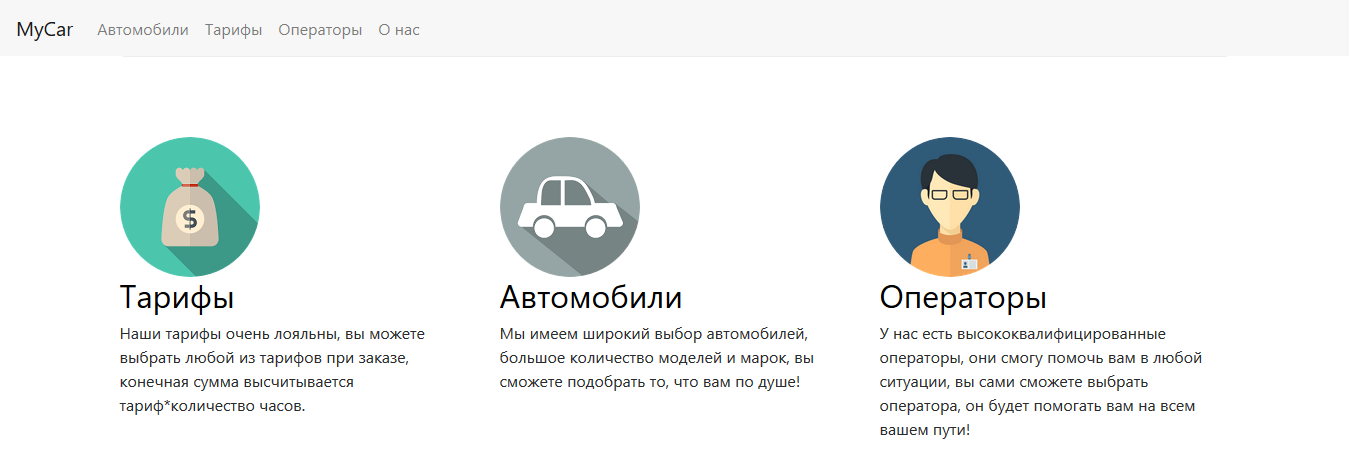
* 1. Разработка структуры веб приложения

«Главная» страница

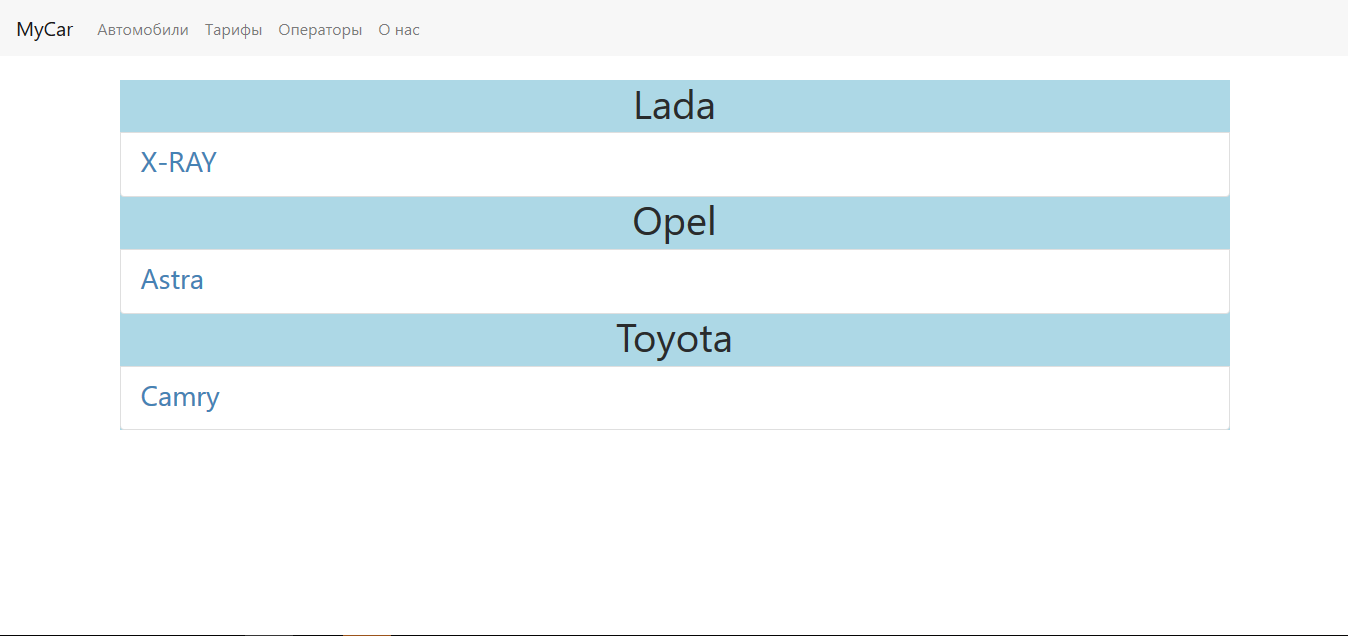




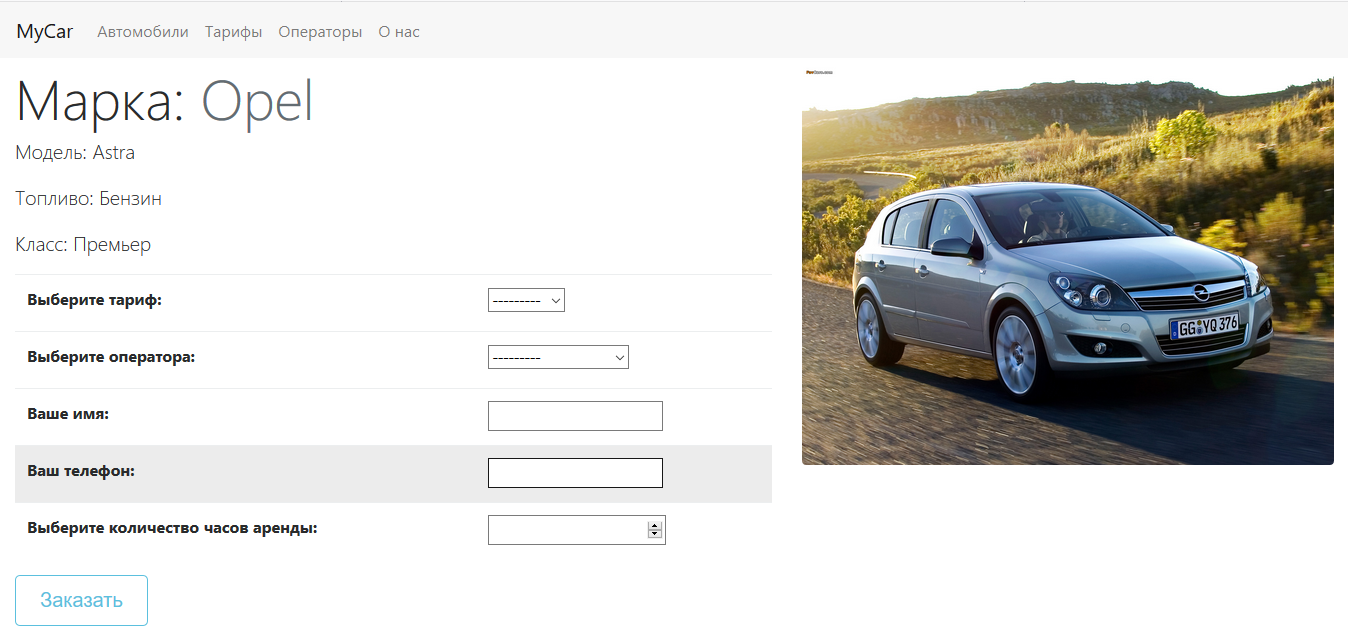




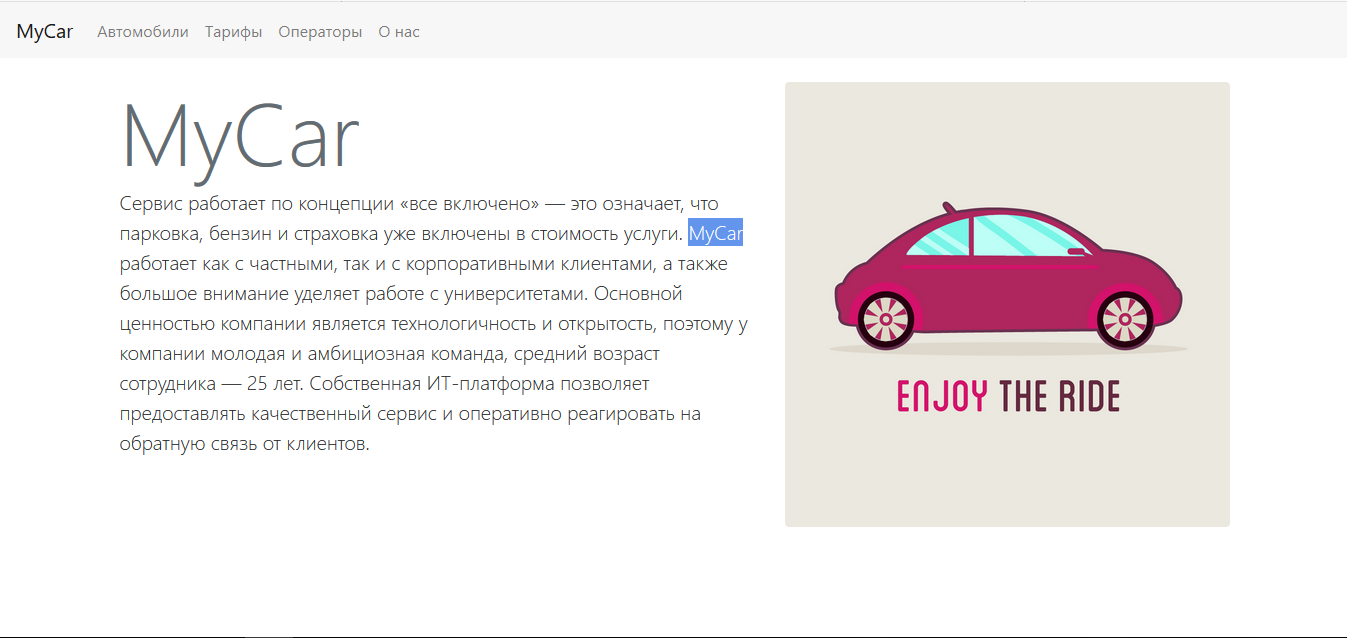
Страница «Автомобили»



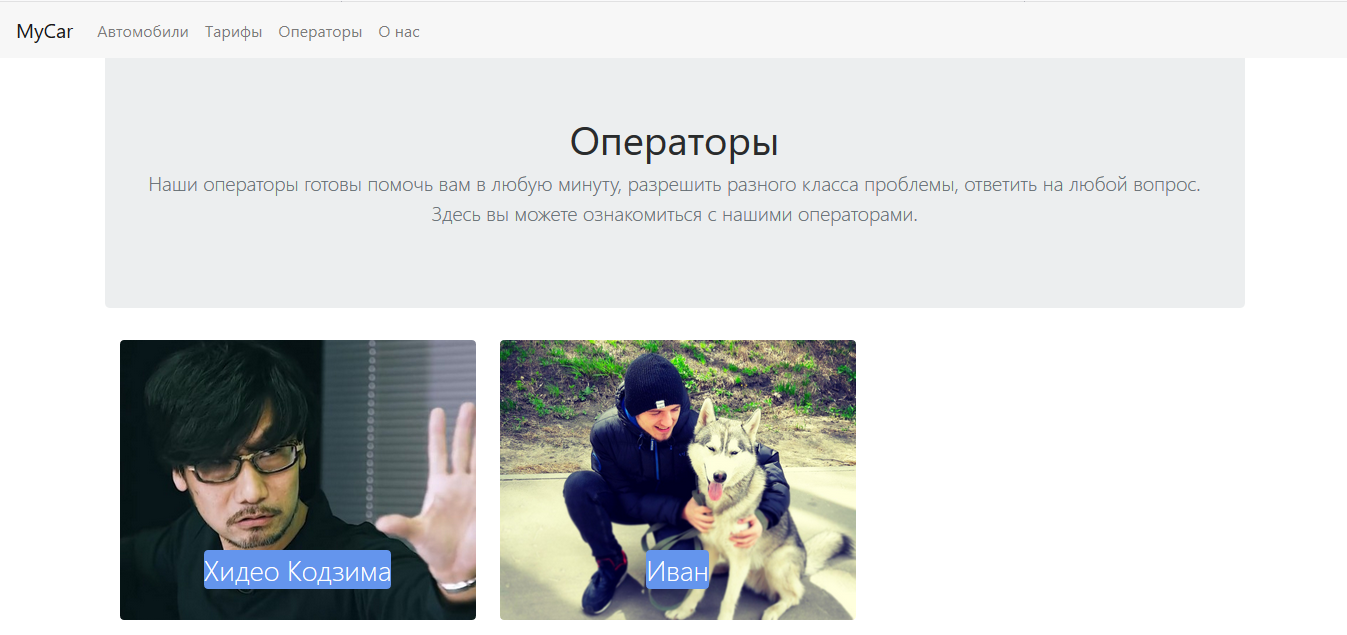
Страница «Марка-Модель-Заказ»



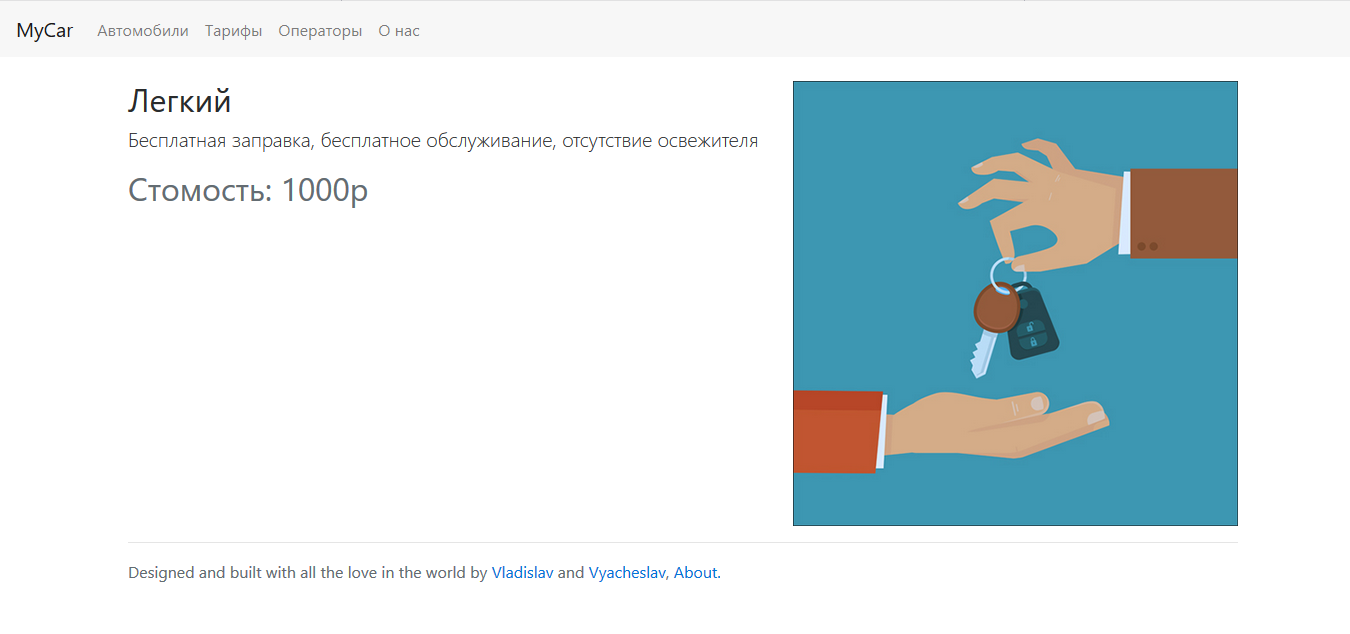
Страница «о нас»



Страница «оператор»

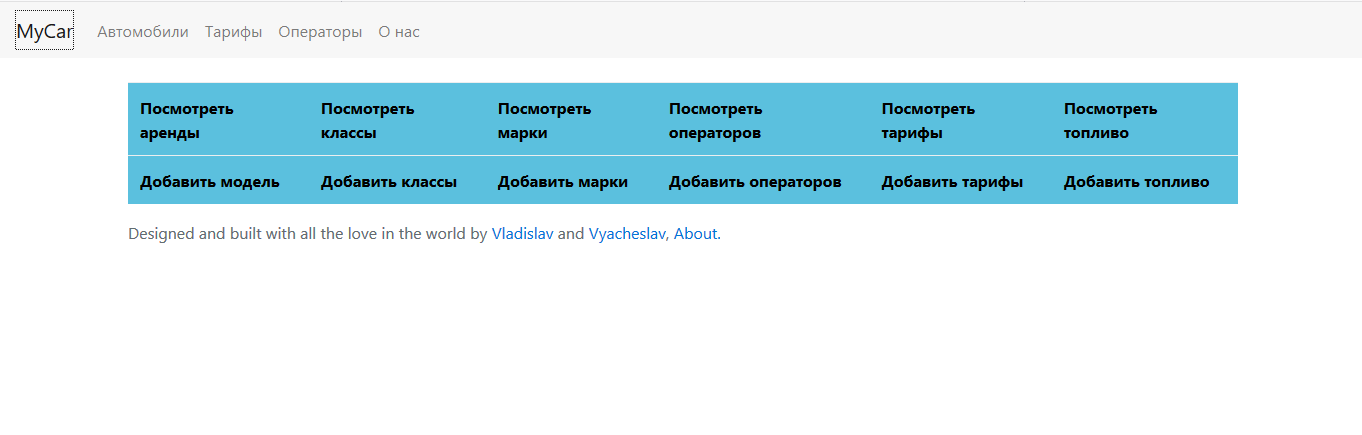


Страница «Тарифы»

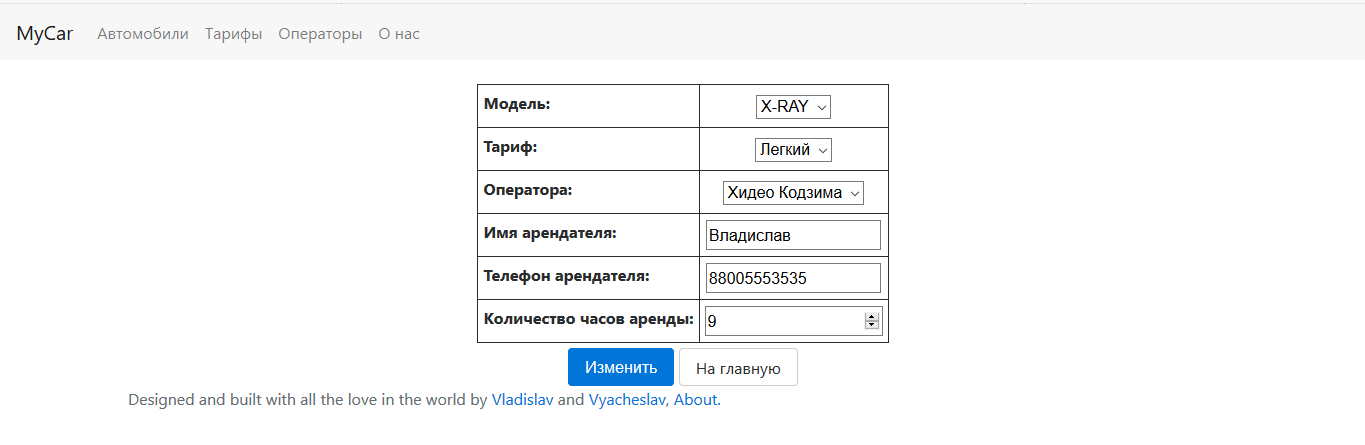


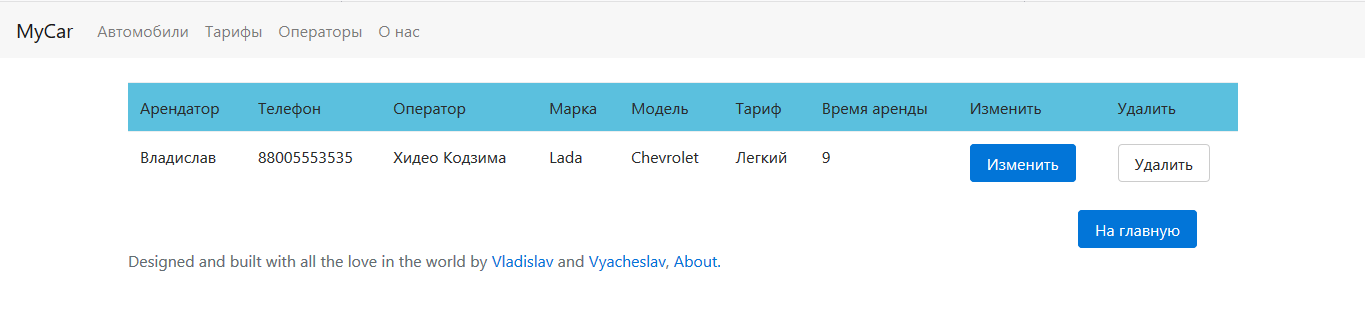
Зона администрирования. По аналогии строятся страницы управления «Марки», «Модели», «Операторы», «Тарифы», «Топливо».

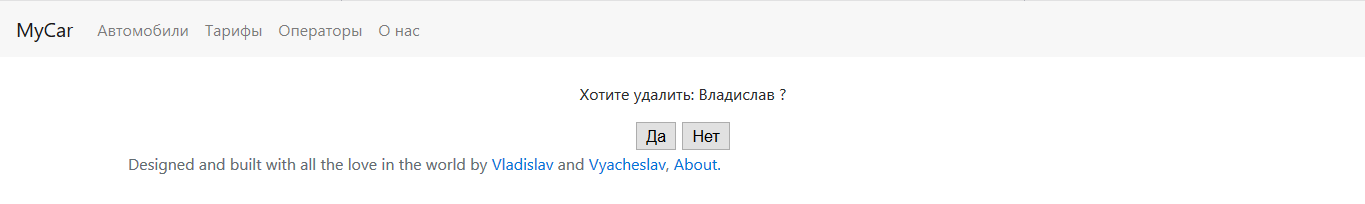
Меню



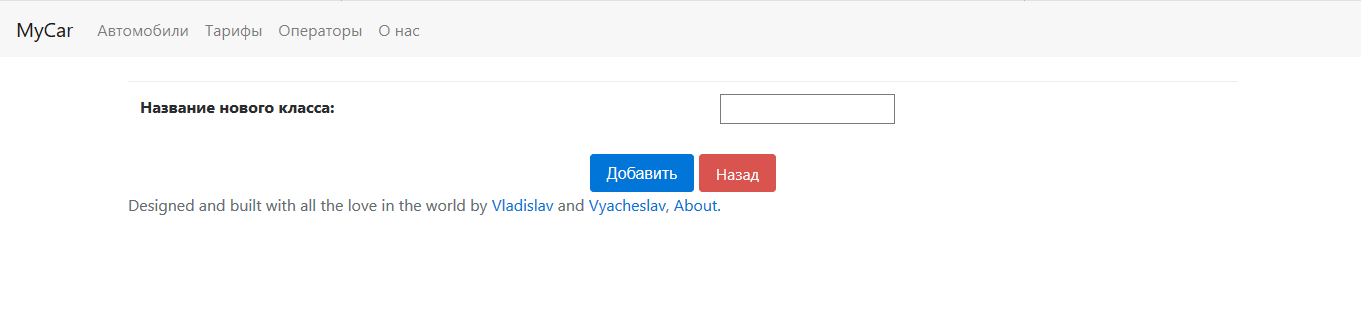
Аренда (просмотр, изменение, удаление)

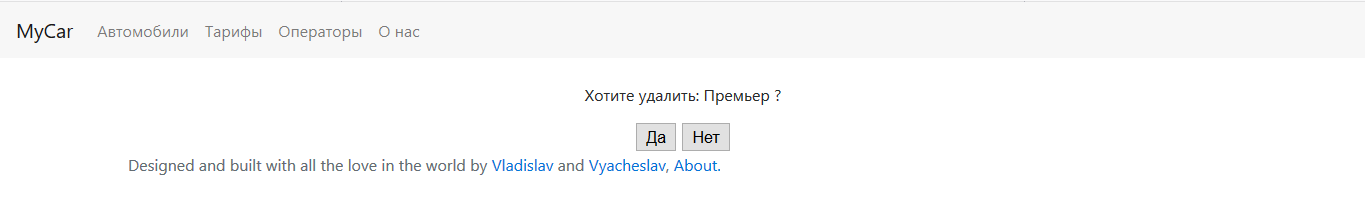
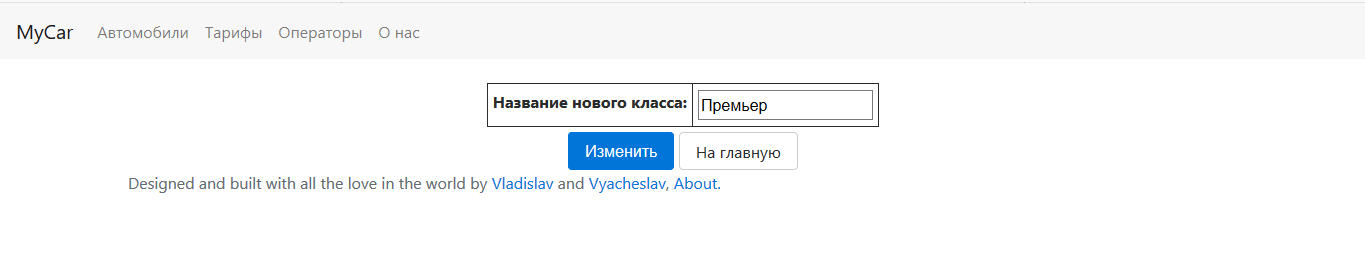
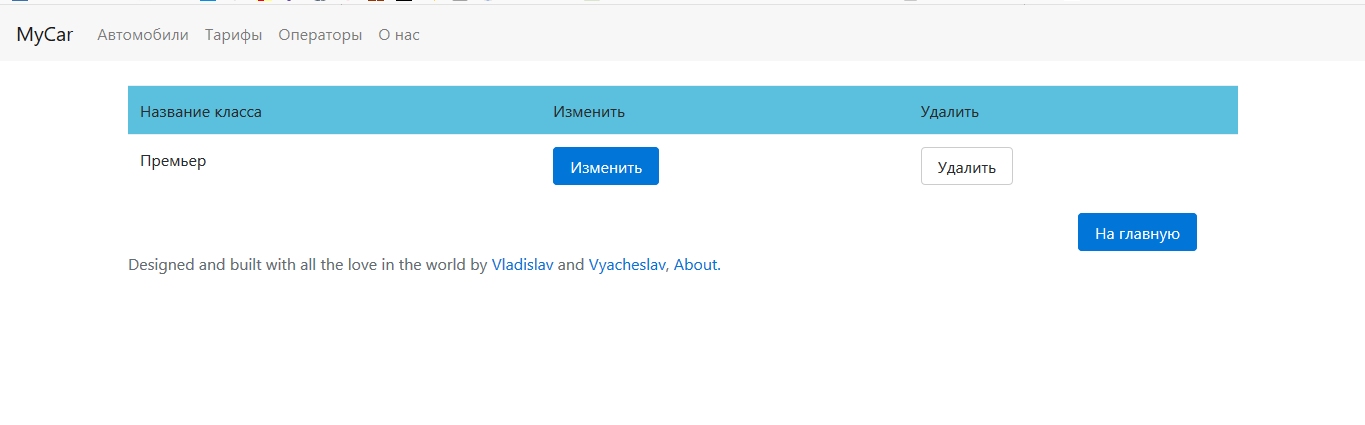






Класс машины (добавление, просмотр, изменение, удаление)





* 1. Разработка программного кода, реализующего функциональные требования к приложению

Чистая, элегантная схема URL-ов – это важная часть качественного приложения. Django позволяет проектировать URL-адреса как вы пожелаете, без ограничений “фреймворка”.

Файл **urls.py** содержит ссылки для перехода между страницами сайта. Всего используется 35 ссылок.

Примеры ссылок:

url(r'^main\_page/$', views.main\_page, name="main\_page"),

url(r'^auto/$', views.auto, name="auto"),

url(r"^model/(?P<model\_id>\d+)$", views.model, name="model").

Файл **view.py** содержит функции для функционирования сайта. **View**, или представление, — это то место, где размещается «логику» работы приложения. Оно запросит информацию из models (модели), и передаст её в templates (шаблоны). Каждая функция начинается со слова Def. Всего используется 34 функции.

Пример функций:

def main\_page(req):

mark\_list = Mark.objects.all()

return render(req, "main\_page.html", locals())

def auto(req):

mark\_list = Mark.objects.all()

model\_list = ModelCar.objects.all()

return render(req, "autos.html", locals())

Будучи веб фреймверком, Django позволяет динамически генерировать HTML. Самый распространенный подход - использование шаблонов. Шаблоны содержат статический HTML и динамические данные, рендеринг которых описан специальным синтаксисом, работу с БД *PostgreSQL*. Всего используется 10 шаблонов HTML страниц для работы с веб приложением пользователя и 11 шаблонов HTML страниц администрирования.

Пример шаблона **Main\_page.html**

{% extends "base.html" %}

{% load static %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="jumbotron first">

<h1 class="display-3">MyCar!</h1>

<p>MyCar - это простой способ заказать машину, это дешевые тарифы, это различные модели и марки, это прекрасные классы автомобилей, добро пожаловать!</p>

<p><a class="btn btn-primary btn-lg" href="{% url "auto" %}" role="button">Заказать</a></p>

</div>

</div>

<div class="container marketing" style="margin-top: 80px;">

<div class="row">

<div class="col-lg-4">

<img class="rounded-circle first" src="{% static "img/1.png" %}" alt="Generic placeholder image" width="140" height="140">

<h2><a href="{% url "tariff" %}" class="thoth">Тарифы</a></h2>

<p class="thoth">Наши тарифы очень лояльны, вы можете выбрать любой из тарифов при заказе, конечная сумма высчитывается тариф\*количество часов.</p>

</div><!-- /.col-lg-4 -->

<div class="col-lg-4">

<img class="rounded-circle first" src="{% static "img/2.png" %}" alt="Generic placeholder image" width="140" height="140">

<h2><a href="{% url "auto" %}" class="thoth">Автомобили</a></h2>

<p class="thoth">Мы имеем широкий выбор автомобилей, большое количество моделей и марок, вы сможете подобрать то, что вам по душе!</p>

</div><!-- /.col-lg-4 -->

<div class="col-lg-4">

<img class="rounded-circle first" src="{% static "img/3.png" %}" alt="Generic placeholder image" width="140" height="140">

<h2><a href="{% url "operators" %}" class="thoth">Операторы</a></h2>

<p class="thoth">У нас есть высококвалифицированные операторы, они смогу помочь вам в любой ситуации, вы сами сможете выбрать оператора, он будет помогать вам на всем вашем пути!</p>

</div><!-- /.col-lg-4 -->

</div><!-- /.row -->

<!-- START THE FEATURETTES -->

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7">

<h2 class="featurette-heading"> <span class="text-muted">Что такое ”Каршеринг”?Это лучший способ передвижения!</span></h2>

<p class="lead">Каршеринг (англ.: «carsharing») – это стремительно набирающая популярность услуга поминутной или почасовой аренды автомобиля, появившаяся в мегаполисах и больших городах мира. Каршеринг создавался как экономичная альтернатива прокату авто и такси, как возможность оставаться мобильным без затрат на покупку и владение личным автотранспортом.</p>

</div>

<div class="col-md-5">

<img class="featurette-image img-fluid mx-auto" src="{% static "img/4.png" %}" alt="Generic placeholder image">

</div>

</div>

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7 push-md-5">

<h2 class="featurette-heading"> <span class="text-muted">Каршеринг – это свобода передвижения на автомобиле без необходимости:</span></h2>

<p class="lead">

<ul>

<li>приобретать автомобиль</li>

<li>оплачивать страховку и транспортный налог</li>

<li>заправлять машину</li>

<li>посещать ТО</li>

<li>готовить автомобиль по сезону</li>

<li>посещать мойку и шиномонтаж</li>

<li>оплачивать парковку</li>

</ul>

</p>

</div>

<div class="col-md-5 pull-md-7">

<img class="featurette-image img-fluid mx-auto" src="{% static "img/5.png" %}" alt="Generic placeholder image">

</div>

</div>

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7">

<h2 class="featurette-heading">Почему MyCar?</h2>

<p class="lead">MyCar – это новый стандарт городской мобильности: «Заказать. Завести. Поехать!» Это использование современного автомобиля без обременений и лишних расходов. Стать участником программы каршеринга просто: зарегистрироваться на сайте и загрузить на сайт фотографии необходимых документов: паспорта, прав, фото с паспортом и подписанные соглашение и согласие о присоединении.</p>

</div>

<div class="col-md-5">

<img class="featurette-image img-fluid mx-auto" src="{% static "img/6.png" %}" alt="Generic placeholder image">

</div>

</div>

<hr class="featurette-divider">

<!-- /END THE FEATURETTES -->

<!-- FOOTER -->

</div><!-- /.container -->

{% endblock content %}

Приложение 1. Исходный код

**Project**

Manage.py

#!/usr/bin/env python

import os

import sys

if \_\_name\_\_ == "\_\_main\_\_":

os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "CSProject.settings")

try:

from django.core.management import execute\_from\_command\_line

except ImportError:

# The above import may fail for some other reason. Ensure that the

# issue is really that Django is missing to avoid masking other

# exceptions on Python 2.

try:

import django

except ImportError:

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?"

)

raise

execute\_from\_command\_line(sys.argv)

Settings.py

"""

Django settings for CSProject project.

Generated by 'django-admin startproject' using Django 1.11.7.

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 = '%s6nq%rgn800t(pys0^kpwxng-x9=&wh-!u0et\_o(xj7\_z3)d3'

# 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',

'MyCar',

]

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 = 'CSProject.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 = 'CSProject.wsgi.application'

# Database

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

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql\_psycopg2',

'NAME': 'CarSharing',

'USER': 'postgres',

'PASSWORD': '1234',

'HOST': 'localhost',

'PORT': '5432',

}

}

# 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/'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, 'static'),

]

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

MEDIA\_URL = '/media/'

Urls.py

"""CSProject 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 django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

url(r'^admin/', admin.site.urls),

url(r'^', include("MyCar.urls"))

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

Wsgi.py

"""

WSGI config for CSProject project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see

https://docs.djangoproject.com/en/1.11/howto/deployment/wsgi/

"""

import os

from django.core.wsgi import get\_wsgi\_application

os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "CSProject.settings")

application = get\_wsgi\_application()

**MyCar**

Admin.py

from django.contrib import admin

from MyCar.models import \*

# Register your models here.

admin.site.register(Tariff)

admin.site.register(Operator)

admin.site.register(Rank)

admin.site.register(ModelCar)

admin.site.register(Rent)

admin.site.register(Fuel)

admin.site.register(Mark)

Apps.py

from django.apps import AppConfig

class MycarConfig(AppConfig):

name = 'MyCar'

Forms.py

from django import forms

from MyCar.models import \*

class RentForm(forms.ModelForm):

id\_model = forms.ModelChoiceField(queryset=ModelCar.objects.all(), widget=forms.HiddenInput)

class Meta:

model = Rent

fields = ['id\_model','id\_tariff','id\_operator','name','phone','time']

labels = {

"id\_model": "Модель",

"id\_tariff": "Выберите тариф",

"id\_operator": "Выберите оператора",

"name": "Ваше имя",

"phone": "Ваш телефон",

"time": "Выберите количество часов аренды",

}

class RankForM(forms.ModelForm):

class Meta:

model = Rank

fields = ['name']

labels = {

'name': 'Название нового класса',

}

class MarkForm(forms.ModelForm):

class Meta:

model = Mark

fields = ['name',]

labels = {

'name': 'Название новой марки',

}

class OperatorForm(forms.ModelForm):

class Meta:

model = Operator

fields = ['name', 'photo', 'number']

labels = {

'name': 'Имя нового оператора',

'photo': 'Фото нового оператора',

'number': 'Номер телефона нового оператора',

}

class TariffForm(forms.ModelForm):

class Meta:

model = Tariff

fields = ['name','description','price']

labels = {

'name': 'Название нового тарифа',

'description': 'Короткое описание нового тарифа',

'price': 'Стоимость нового тарифа',

}

class FuelForm(forms.ModelForm):

class Meta:

model = Fuel

fields = ['name']

labels = {

'name': 'Название нового топлива',

}

class ModelForm(forms.ModelForm):

class Meta:

model = ModelCar

fields = ['name', 'mark\_id', 'id\_rank', 'id\_fuel', 'photo']

labels = {

'name': 'Название новой модели',

'id\_fuel': 'Топливо новой модели',

'id\_rank': 'Название новой модели',

'mark\_id': 'Марка новой модели',

'photo': 'Фото новой модели',

}

class RentEditForm(forms.ModelForm):

class Meta:

model = Rent

fields = ['id\_model','id\_tariff','id\_operator','name','phone','time']

labels = {

"id\_model": "Модель",

"id\_tariff": "Тариф",

"id\_operator": "Оператора",

"name": "Имя арендателя",

"phone": "Телефон арендателя",

"time": "Количество часов аренды",

}

Models.py

from django.db import models

# Create your models here.

class Tariff(models.Model):

tariff\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=30)

description = models.CharField(max\_length=300)

price = models.IntegerField()

class Meta():

verbose\_name = "Тариф"

verbose\_name\_plural = "Тарифов"

def \_\_str\_\_(self):

return self.name

class Operator(models.Model):

operator\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=20)

number = models.CharField(max\_length=30)

photo = models.ImageField()

class Meta():

verbose\_name = "Оператор"

verbose\_name\_plural = "Операторы"

def \_\_str\_\_(self):

return self.name

class Fuel(models.Model):

fuel\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=20)

class Meta():

verbose\_name = "Топливо"

verbose\_name\_plural = "Топлива"

def \_\_str\_\_(self):

return self.name

class Rank(models.Model):

rank\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=20)

class Meta():

verbose\_name = "Класс"

verbose\_name\_plural = "Классов"

def \_\_str\_\_(self):

return self.name

class Mark(models.Model):

mark\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=60)

class Meta():

verbose\_name = "Марка"

verbose\_name\_plural = "Марок"

def \_\_str\_\_(self):

return self.name

class ModelCar(models.Model):

model\_car\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=20)

id\_rank = models.ForeignKey(Rank, verbose\_name="Класс")

id\_fuel = models.ForeignKey(Fuel, verbose\_name="Топливо")

mark\_id = models.ForeignKey(Mark, verbose\_name="Марка")

photo = models.ImageField(blank=True, upload\_to='MyCar/media')

class Meta():

verbose\_name = "Модель"

verbose\_name\_plural = "Модели"

def \_\_str\_\_(self):

return self.name

class Rent(models.Model):

rent\_id = models.AutoField(primary\_key=True)

id\_tariff = models.ForeignKey(Tariff, verbose\_name="Тариф")

id\_model = models.ForeignKey(ModelCar, verbose\_name="Модель")

id\_operator = models.ForeignKey(Operator, verbose\_name="Оператор")

name = models.CharField(verbose\_name='Арендатор', max\_length=25, default='')

phone = models.CharField(verbose\_name='Телефон', max\_length=11, default='')

time = models.PositiveIntegerField()

class Meta():

verbose\_name = "Аренда"

verbose\_name\_plural = "Аренд"

def \_\_str\_\_(self):

return self.name

Tests.py

from django.test import TestCase

# Create your tests here.

Urls.py

from django.conf.urls import url

from MyCar import views

urlpatterns = [

url(r'^main\_page/$', views.main\_page, name="main\_page"),

url(r'^auto/$', views.auto, name="auto"),

url(r"^model/(?P<model\_id>\d+)$", views.model, name="model"),

url(r"^tariff/$", views.tariff, name="tariff"),

url(r"^operators/$", views.operators, name="operators"),

url(r"^about/$", views.about, name="about"),

url(r"^admin\_menu/$", views.admin\_menu, name="admin\_menu"),

url(r"^show\_rent/$", views.show\_rent, name="show\_rent"),

url(r"^show\_rank/$", views.show\_rank, name="show\_rank"),

url(r"^show\_mark\_models/$", views.show\_mark, name="show\_mark"),

url(r"^models/(?P<id\_mark>\d+)$", views.show\_models, name="show\_models"),

url(r"^fuel/$", views.show\_fuel, name="show\_fuel"),

url(r"^operator/$", views.show\_operator, name="show\_operator"),

url(r"^tariffs/$", views.show\_tariffs, name="show\_tariffs"),

url(r"^add\_rank/$", views.add\_rank, name="add\_rank"),

url(r"^add\_mark/$", views.add\_mark, name="add\_mark"),

url(r"^add\_operator/$", views.add\_operator, name="add\_operator"),

url(r"^add\_tariff/$", views.add\_tariff, name="add\_tariff"),

url(r"^add\_fuel/$", views.add\_fuel, name="add\_fuel"),

url(r"^add\_model/$", views.add\_model, name="add\_model"),

url(r"^edit\_rank/(?P<rank\_id>\d+)$", views.edit\_rank, name="edit\_rank"),

url(r"^edit\_mark/(?P<mark\_id>\d+)$", views.edit\_mark, name="edit\_mark"),

url(r"^edit\_model/(?P<model\_id>\d+)$", views.edit\_model, name="edit\_model"),

url(r"^edit\_tariff/(?P<tariff\_id>\d+)$", views.edit\_tariff, name="edit\_tariff"),

url(r"^edit\_fuel/(?P<fuel\_id>\d+)$", views.edit\_fuel, name="edit\_fuel"),

url(r"^edit\_rent/(?P<rent\_id>\d+)$", views.edit\_rent, name="edit\_rent"),

url(r"^edit\_operator/(?P<operator\_id>\d+)$", views.edit\_operator, name="edit\_operator"),

url(r"^delete\_rent/(?P<rent\_id>\d+)$", views.delete\_rent, name="delete\_rent"),

url(r"^delete\_rank/(?P<rank\_id>\d+)$", views.delete\_rank, name="delete\_rank"),

url(r"^delete\_operator/(?P<operator\_id>\d+)$", views.delete\_operator, name="delete\_operator"),

url(r"^delete\_tariff/(?P<tariff\_id>\d+)$", views.delete\_tariff, name="delete\_tariff"),

url(r"^delete\_fuel/(?P<fuel\_id>\d+)$", views.delete\_fuel, name="delete\_fuel"),

url(r"^delete\_mark/(?P<mark\_id>\d+)$", views.delete\_mark, name="delete\_mark"),

url(r"^delete\_model/(?P<model\_car\_id>\d+)$", views.delete\_model, name="delete\_model"),

]

Views.py

from django.shortcuts import render, redirect

from MyCar.models import \*

from MyCar.forms import \*

import time

# Create your views here.

def main\_page(req):

mark\_list = Mark.objects.all()

return render(req, "main\_page.html", locals())

def auto(req):

mark\_list = Mark.objects.all()

model\_list = ModelCar.objects.all()

return render(req, "autos.html", locals())

def model(req, model\_id):

car = ModelCar.objects.get(model\_car\_id=model\_id)

form = RentForm(initial={

'id\_model': car.model\_car\_id

})

if req.method == "POST":

form = RentForm(req.POST, initial={

'car': car

})

if form.is\_valid():

form.save()

time.sleep(3)

return redirect('main\_page')

else:

return render(req, "model.html", locals())

return render(req, "model.html", locals())

def tariff(req):

how = 0

tariff\_list = Tariff.objects.all()

return render(req, "tariff.html", locals())

def operators(req):

operator\_list = Operator.objects.all()

return render(req, "operators.html", locals())

def about(req):

return render(req, "about.html", locals())

def admin\_menu(req):

return render(req, "admin/admin\_menu.html", locals())

def show\_rent(req):

rent\_list = Rent.objects.all()

return render(req, "admin/rent/show.html", locals())

def show\_rank(req):

rank\_list = Rank.objects.all()

return render(req, "admin/Rank/show.html", locals())

def show\_mark(req):

mark\_list = Mark.objects.all()

return render(req, "admin/Mark/show.html", locals())

def show\_models(req, id\_mark):

mark = Mark.objects.get(mark\_id=id\_mark)

model\_list = ModelCar.objects.all()

return render(req, "admin/Model/show.html", locals())

def show\_fuel(req):

fuel\_list = Fuel.objects.all()

return render(req, "admin/Fuel/show.html", locals())

def show\_operator(req):

operator\_list = Operator.objects.all()

return render(req, "admin/Operator/show.html", locals())

def show\_tariffs(req):

tariffs\_list = Tariff.objects.all()

return render(req, "admin/Tariff/show.html", locals())

def add\_rank(req):

form = RankForM()

if req.method == "POST":

form = RankForM(req.POST)

if form.is\_valid():

form.save()

return redirect('show\_rank')

else:

return render(req, "admin/admin\_add.html", locals())

return render(req, "admin/admin\_add.html", locals())

def add\_mark(req):

form = MarkForm()

if req.method == "POST":

form = MarkForm(req.POST)

if form.is\_valid():

form.save()

return redirect('show\_mark')

else:

return render(req, "admin/admin\_add.html", locals())

return render(req, "admin/admin\_add.html", locals())

def add\_operator(req):

form = OperatorForm()

if req.method == "POST":

form = OperatorForm(req.POST, req.FILES)

if form.is\_valid():

form.save()

return redirect('show\_operator')

else:

return render(req, "admin/admin\_add.html", locals())

return render(req, "admin/admin\_add.html", locals())

def add\_tariff(req):

form = TariffForm()

if req.method == "POST":

form = TariffForm(req.POST)

if form.is\_valid():

form.save()

return redirect('show\_tariffs')

else:

return render(req, "admin/admin\_add.html", locals())

return render(req, "admin/admin\_add.html", locals())

def add\_fuel(req):

form = FuelForm()

if req.method == "POST":

form = FuelForm(req.POST)

if form.is\_valid():

form.save()

return redirect('show\_fuel')

else:

return render(req, "admin/admin\_add.html", locals())

return render(req, "admin/admin\_add.html", locals())

def add\_model(req):

form = ModelForm()

if req.method == "POST":

form = ModelForm(req.POST, req.FILES)

if form.is\_valid():

form.save()

return redirect('admin\_menu')

else:

return render(req, "admin/admin\_add.html", locals())

return render(req, "admin/admin\_add.html", locals())

def edit\_rank(req, rank\_id):

rank = Rank.objects.get(rank\_id=rank\_id)

form = RankForM(instance=rank)

if req.method == "POST":

form = RankForM(req.POST, instance=rank)

if form.is\_valid():

form.save()

return redirect('show\_rank')

else:

return render(req, "admin/admin\_edit.html", locals())

return render(req, "admin/admin\_edit.html", locals())

def edit\_mark(req, mark\_id):

mark = Mark.objects.get(mark\_id=mark\_id)

form = MarkForm(instance=mark)

if req.method == "POST":

form = MarkForm(req.POST, instance=mark)

if form.is\_valid():

form.save()

return redirect('show\_mark')

else:

return render(req, "admin/admin\_edit.html", locals())

return render(req, "admin/admin\_edit.html", locals())

def edit\_model(req, model\_id):

model = ModelCar.objects.get(model\_car\_id=model\_id)

form = ModelForm(instance=model)

if req.method == "POST":

form = MarkForm(req.POST, req.FILES, instance=model)

if form.is\_valid():

form.save()

return redirect('admin\_menu')

else:

return render(req, "admin/admin\_edit.html", locals())

return render(req, "admin/admin\_edit.html", locals())

def edit\_tariff(req, tariff\_id):

tarif = Tariff.objects.get(tariff\_id=tariff\_id)

form = TariffForm(instance=tarif)

if req.method == "POST":

form = TariffForm(req.POST, instance=tarif)

if form.is\_valid():

form.save()

return redirect('show\_tariffs')

else:

return render(req, "admin/admin\_edit.html", locals())

return render(req, "admin/admin\_edit.html", locals())

def edit\_fuel(req, fuel\_id):

fuel = Fuel.objects.get(fuel\_id=fuel\_id)

form = FuelForm(instance=fuel)

if req.method == "POST":

form = FuelForm(req.POST, instance=fuel)

if form.is\_valid():

form.save()

return redirect('show\_fuel')

else:

return render(req, "admin/admin\_edit.html", locals())

return render(req, "admin/admin\_edit.html", locals())

def edit\_rent(req, rent\_id):

rent = Rent.objects.get(rent\_id=rent\_id)

form = RentEditForm(instance=rent)

if req.method == "POST":

form = RentEditForm(req.POST, req.FILES, instance=rent)

if form.is\_valid():

form.save()

return redirect('show\_rent')

else:

return render(req, "admin/admin\_edit.html", locals())

return render(req, "admin/admin\_edit.html", locals())

def edit\_operator(req, operator\_id):

operator = Operator.objects.get(operator\_id=operator\_id)

form = OperatorForm(instance=operator)

if req.method == "POST":

form = OperatorForm(req.POST, req.FILES, instance=operator)

if form.is\_valid():

form.save()

return redirect('show\_operator')

else:

return render(req, "admin/admin\_edit.html", locals())

return render(req, "admin/admin\_edit.html", locals())

def delete\_rent(request, rent\_id):

rent = Rent.objects.get(rent\_id=rent\_id)

name\_delete = rent.name

if request.method == "POST":

if 'yes' in request.POST.keys() and request.POST['yes']:

rent.delete()

return redirect("show\_rent")

else:

return redirect("show\_rent")

return render(request, 'admin/admin\_delete.html', locals())

def delete\_rank(request, rank\_id):

rank = Rank.objects.get(rank\_id=rank\_id)

name\_delete = rank.name

if request.method == "POST":

if 'yes' in request.POST.keys() and request.POST['yes']:

rank.delete()

return redirect("show\_rank")

else:

return redirect("show\_rank")

return render(request, 'admin/admin\_delete.html', locals())

def delete\_mark(request, mark\_id):

mark = Mark.objects.get(mark\_id=mark\_id)

name\_delete = mark.name

if request.method == "POST":

if 'yes' in request.POST.keys() and request.POST['yes']:

mark.delete()

return redirect("show\_mark")

else:

return redirect("show\_mark")

return render(request, 'admin/admin\_delete.html', locals())

def delete\_operator(request, operator\_id):

operator = Operator.objects.get(operator\_id=operator\_id)

name\_delete = operator.name

if request.method == "POST":

if 'yes' in request.POST.keys() and request.POST['yes']:

operator.delete()

return redirect("show\_operator")

else:

return redirect("show\_operator")

return render(request, 'admin/admin\_delete.html', locals())

def delete\_tariff(request, tariff\_id):

tariff = Tariff.objects.get(tariff\_id=tariff\_id)

name\_delete = tariff.name

if request.method == "POST":

if 'yes' in request.POST.keys() and request.POST['yes']:

tariff.delete()

return redirect("show\_tariffs")

else:

return redirect("show\_tariffs")

return render(request, 'admin/admin\_delete.html', locals())

def delete\_fuel(request, fuel\_id):

fuel = Fuel.objects.get(fuel\_id=fuel\_id)

name\_delete = fuel.name

if request.method == "POST":

if 'yes' in request.POST.keys() and request.POST['yes']:

fuel.delete()

return redirect("show\_fuel")

else:

return redirect("show\_fuel")

return render(request, 'admin/admin\_delete.html', locals())

def delete\_model(request, model\_car\_id):

model = ModelCar.objects.get(model\_car\_id=model\_car\_id)

name\_delete = model.name

if request.method == "POST":

if 'yes' in request.POST.keys() and request.POST['yes']:

model.delete()

return redirect("show\_mark")

else:

return redirect("show\_mark")

return render(request, 'admin/admin\_delete.html', locals())

Style.css

h2 a:hover{

text-decoration: none;

color: steelblue;

transition: color 0.3s ease-in;

}

h2 a{

color: black;

}

.bg:hover {

color: black;

background-color: lightblue;

text-decoration: none;

transition: background-color 0.3s ease-in;

}

.nonedec:hover {

color: black;

text-decoration: none;

}

.list-group-item:hover{

background-color: aliceblue;

transition: background-color 0.3s ease-in;

}

.list-group{

background-color: lightblue;

}

li a{

/\*font-weight: bold;\*/

}

thead tr td a:hover{

text-decoration: none;

color: black;

}

thead tr td:hover{

background-color: lightblue;

transition: background-color 0.3s ease-in;

}

td a{

color: black;

font-weight: bold ;

}

ul li{

list-style: none;

}

li a{

color: steelblue;

}

.content{

height: 100vh;

}

@keyframes strt{

0%{

opacity: 0;

}

100%{

opacity: 1;

}

}

.first{

animation: strt 1s ease;

}

.second{

animation: strt 2s ease;

}

.third{

animation: strt 3s ease;

}

.thoth{

animation: strt 4s ease;

}

**Templates**

about.html

{% extends "base.html" %}

{% block content %}

{% load static %}

<div class="container" style="margin-top: 80px;">

<div class="content">

<div class="row">

<div class="col-md-12">

<div class="row featurette">

<div class="col-md-7 first">

<h2 class="featurette-heading display-2"> <span

class="text-muted">MyCar</span></h2>

<p class="lead">

Сервис работает по концепции «все включено» — это означает, что парковка, бензин и страховка уже включены в стоимость услуги. <span style="background-color: cornflowerblue; color: white"> MyCar </span> работает как с частными, так и с корпоративными клиентами, а также большое внимание уделяет работе с университетами.

Основной ценностью компании является технологичность и открытость, поэтому у компании молодая и амбициозная команда, средний возраст сотрудника — 25 лет.

Собственная ИТ-платформа позволяет предоставлять качественный сервис и оперативно реагировать на обратную связь от клиентов.

</p>

</div>

<div class="col-md-5">

<img src="{% static "img/9.jpg" %}" class="img-fluid rounded second" alt="Responsive image">

</div>

</div>

</div>

</div>

</div>

</div>

</div>

{% endblock content %}

autos.html

base.html

{% load staticfiles %}

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<meta name="description" content="">

<meta name="author" content="">

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

<link rel="stylesheet" href="{% static 'img' %}">

<title>{% block title %}{% endblock title %}</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css" integrity="sha384-rwoIResjU2yc3z8GV/NPeZWAv56rSmLldC3R/AZzGRnGxQQKnKkoFVhFQhNUwEyJ" crossorigin="anonymous">

</head>

<body>

{% include 'header.html' %}

{% block content %}

{% endblock content %}

{% include 'footer.html' %}

<script src="https://code.jquery.com/jquery-3.1.1.slim.min.js" integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k9Omg/a/EheAdgtzNs3hpfag6Ed950n" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js" integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIxUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-vBWWzlZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVkE+jo0ieGizqPLForn" crossorigin="anonymous"></script>

</body>

</html>

footer.html

<footer class="bd-footer text-muted">

<div class="container">

<p>Designed and built with all the love in the world by <a href="#">Vladislav</a> and <a href="#">Vyacheslav</a>, <a href="{% url "about" %}">About.</a></p>

</div>

</footer>

header.html

<nav class="navbar navbar-toggleable-md navbar-light bg-faded fixed-top">

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

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

</button>

<a class="navbar-brand" href="{% url "main\_page" %}">MyCar</a>

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

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link bg" href="{% url "auto" %}">Автомобили</a>

</li>

<li class="nav-item">

<a class="nav-link bg" href="{% url "tariff" %}">Тарифы</a>

</li>

<li class="nav-item">

<a class="nav-link bg" href="{% url "operators" %}">Операторы</a>

</li>

<li class="nav-item">

<a class="nav-link bg" href="{% url "about" %}">О нас</a>

</li>

</ul>

</div>

</nav>

main\_page.html

{% extends "base.html" %}

{% load static %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="jumbotron first">

<h1 class="display-3">MyCar!</h1>

<p>MyCar - это простой способ заказать машину, это дешевые тарифы, это различные модели и марки, это прекрасные классы автомобилей, добро пожаловать!</p>

<p><a class="btn btn-primary btn-lg" href="{% url "auto" %}" role="button">Заказать</a></p>

</div>

</div>

<div class="container marketing" style="margin-top: 80px;">

<div class="row">

<div class="col-lg-4">

<img class="rounded-circle first" src="{% static "img/1.png" %}" alt="Generic placeholder image" width="140" height="140">

<h2><a href="{% url "tariff" %}" class="thoth">Тарифы</a></h2>

<p class="thoth">Наши тарифы очень лояльны, вы можете выбрать любой из тарифов при заказе, конечная сумма высчитывается тариф\*количество часов.</p>

</div><!-- /.col-lg-4 -->

<div class="col-lg-4">

<img class="rounded-circle first" src="{% static "img/2.png" %}" alt="Generic placeholder image" width="140" height="140">

<h2><a href="{% url "auto" %}" class="thoth">Автомобили</a></h2>

<p class="thoth">Мы имеем широкий выбор автомобилей, большое количество моделей и марок, вы сможете подобрать то, что вам по душе!</p>

</div><!-- /.col-lg-4 -->

<div class="col-lg-4">

<img class="rounded-circle first" src="{% static "img/3.png" %}" alt="Generic placeholder image" width="140" height="140">

<h2><a href="{% url "operators" %}" class="thoth">Операторы</a></h2>

<p class="thoth">У нас есть высококвалифицированные операторы, они смогу помочь вам в любой ситуации, вы сами сможете выбрать оператора, он будет помогать вам на всем вашем пути!</p>

</div><!-- /.col-lg-4 -->

</div><!-- /.row -->

<!-- START THE FEATURETTES -->

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7">

<h2 class="featurette-heading"> <span class="text-muted">Что такое ”Каршеринг”?Это лучший способ передвижения!</span></h2>

<p class="lead">Каршеринг (англ.: «carsharing») – это стремительно набирающая популярность услуга поминутной или почасовой аренды автомобиля, появившаяся в мегаполисах и больших городах мира. Каршеринг создавался как экономичная альтернатива прокату авто и такси, как возможность оставаться мобильным без затрат на покупку и владение личным автотранспортом.</p>

</div>

<div class="col-md-5">

<img class="featurette-image img-fluid mx-auto" src="{% static "img/4.png" %}" alt="Generic placeholder image">

</div>

</div>

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7 push-md-5">

<h2 class="featurette-heading"> <span class="text-muted">Каршеринг – это свобода передвижения на автомобиле без необходимости:</span></h2>

<p class="lead">

<ul>

<li>приобретать автомобиль</li>

<li>оплачивать страховку и транспортный налог</li>

<li>заправлять машину</li>

<li>посещать ТО</li>

<li>готовить автомобиль по сезону</li>

<li>посещать мойку и шиномонтаж</li>

<li>оплачивать парковку</li>

</ul>

</p>

</div>

<div class="col-md-5 pull-md-7">

<img class="featurette-image img-fluid mx-auto" src="{% static "img/5.png" %}" alt="Generic placeholder image">

</div>

</div>

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7">

<h2 class="featurette-heading">Почему MyCar?</h2>

<p class="lead">MyCar – это новый стандарт городской мобильности: «Заказать. Завести. Поехать!» Это использование современного автомобиля без обременений и лишних расходов. Стать участником программы каршеринга просто: зарегистрироваться на сайте и загрузить на сайт фотографии необходимых документов: паспорта, прав, фото с паспортом и подписанные соглашение и согласие о присоединении.</p>

</div>

<div class="col-md-5">

<img class="featurette-image img-fluid mx-auto" src="{% static "img/6.png" %}" alt="Generic placeholder image">

</div>

</div>

<hr class="featurette-divider">

<!-- /END THE FEATURETTES -->

<!-- FOOTER -->

</div><!-- /.container -->

{% endblock content %}

model.html

{% extends "base.html" %}

{% block content %}

<div class="container-fluid" style="margin-top: 80px;">

<div class="row">

<div class="col-md-12">

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7" >

<h2 class="featurette-heading display-4 first">Марка: <span class="text-muted">{{ car.mark\_id }}</span></h2>

<p class="lead second">Модель: {{ car.name }}</p>

<p class="lead second">Топливо: {{ car.id\_fuel }}</p>

<p class="lead second">Класс: {{ car.id\_rank }}</p>

<form action="" method="POST">

{% csrf\_token %}

<table class="table table-hover bg-light third">

{{ form.as\_table }}

</table>

<input type="submit" value="Заказать" class="btn btn-outline-info btn-lg order" >

</form>

</div>

<div class="col-md-5">

<img src="{{ car.photo.url }}" class="img-fluid rounded second" alt="Responsive image">

</div>

</div>

<hr class="featurette-divider">

</div>

</div>

</div>

<script src="https://cdn.rawgit.com/alertifyjs/alertify.js/v1.0.10/dist/js/alertify.js"></script>

<script language="JavaScript">

var doc = document,

elem = doc.getElementsByClassName("order")[0];

elem.addEventListener("click",function () {

alertify.alert("Ваш заказ принят, ожидайте звонка оператора, спасибо!");

},true);

</script>

{% endblock %}

Operators.html

{% extends "base.html" %}

{% block content %}

{% load static %}

<div class="container" style="margin-top: 80px;">

<div class="row">

<section class="jumbotron text-center">

<div class="second">

<h1 class="jumbotron-heading">Операторы</h1>

<p class="lead text-muted">Наши операторы готовы помочь вам в любую минуту, разрешить разного класса проблемы, ответить на любой вопрос. Здесь вы можете ознакомиться с нашими операторами.</p>

</div>

</section>

</div>

<div class="row">

{% for operator in operator\_list %}

<div class="col-md-4" style="margin-bottom: 15px;">

<div class="second rounded" style='background-image: url("{{ operator.photo.url }}");max-height: 280px; min-height: 280px; max-width: 356px; min-width: 356px; background-position: center; background-size: cover'>

<h3 align="center" style="padding-top: 60%"> <span class="rounded" style="background: cornflowerblue; color: #fff; font-weight: 100">{{ operator.name }}</span></h3>

</div>

</div>

{% endfor %}

</div>

</div>

{% endblock content %}

tariff.html

{% extends "base.html" %}

{% block content %}

{% load static %}

<div class="container" style="margin-top: 80px;">

<!-- START THE FEATURETTES -->

{% for tar in tariff\_list %}

<div class="row featurette">

<div class="col-md-7 first">

<h2 class="featurette-heading">{{ tar.name }} <span class="text-muted"></span></h2>

<p class="lead">{{ tar.description }}</p>

<h2><span class="text-muted">Стомость: {{ tar.price }}р</span></h2>

</div>

<div class="col-md-5 second">

<img class="featurette-image img-fluid mx-auto" src="{% static "img/4.png" %}"

alt="Generic placeholder image">

</div>

</div>

<hr class="featurette-divider">

{% endfor %}

</div>

</div>

{% endblock content %}

**Admin**

Fuel/show.html

{% extends "base.html" %}

{% block title %}

Топливо

{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="row">

<div class="col-md-12">

<table class="table table-hover" style="background-color: white">

<thead class="bg-info">

<tr>

<td>Название топлива</td>

<td>Изменить</td>

<td>Удалить</td>

</tr>

</thead>

{% for fuel in fuel\_list %}

<tr>

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

<td><a href="{% url "edit\_fuel" fuel.fuel\_id %}" class="btn btn-primary" role="button">Изменить</a>

</td>

<td><a href="{% url "delete\_fuel" fuel.fuel\_id %}" class="btn btn-secondary" role="button">Удалить</a></td>

</tr>

{% endfor %}

</table>

</div>

</div>

<div class="row">

<div class="col-md-2 offset-md-10">

<a href="{% url "admin\_menu" %}" class="btn btn-primary" role="button">На главную</a>

</div>

</div>

</div>

{% endblock content %}

Mark/show.html

{% extends "base.html" %}

{% block title %}

Марки

{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="row">

<div class="col-md-12">

<table class="table table-hover" style="background-color: white">

<thead class="bg-info">

<tr>

<td>Название марки</td>

<td>Изменить</td>

<td>Удалить</td>

</tr>

</thead>

{% for mark in mark\_list %}

<tr>

<td><a href="{% url "show\_models" mark.mark\_id %}">{{ mark.name }}</a></td>

<td><a href="{% url "edit\_mark" mark.mark\_id %}" class="btn btn-primary" role="button">Изменить</a>

</td>

<td><a href="{% url "delete\_mark" mark.mark\_id %}" class="btn btn-secondary" role="button">Удалить</a></td>

</tr>

{% endfor %}

</table>

</div>

</div>

<div class="row">

<div class="col-md-2 offset-md-10">

<a href="{% url "admin\_menu" %}" class="btn btn-primary" role="button">На главную</a>

</div>

</div>

</div>

{% endblock content %}

Model/show.html

{% extends "base.html" %}

{% block title %}

Модели {{ mark.name }}

{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="row">

<div class="col-md-12">

<table class="table table-hover" style="background-color: white">

<thead class="bg-info">

<tr align="center" style="font-weight: bold">

<td>

Модель

</td>

<td colspan="2" style="font-weight: bold;">

{{ mark.name }}

</td>

</tr>

</thead>

{% for model in model\_list %}

{% if model.mark\_id.mark\_id == mark.mark\_id %}

<tr align="center">

<td>

{{ model.name }}

</td>

<td><a href="{% url "edit\_model" model.model\_car\_id %}" class="btn btn-primary" role="button">Изменить</a>

</td>

<td><a href="{% url "delete\_model" model.model\_car\_id %}" class="btn btn-secondary" role="button">Удалить</a>

</td>

</tr>

{% endif %}

{% endfor %}

</table>

</div>

</div>

<div class="row">

<div class="col-md-2 offset-md-10">

<a href="{% url "admin\_menu" %}" class="btn btn-primary" role="button">На главную</a>

</div>

</div>

</div>

{% endblock content %}

Operator/show.html

{% extends "base.html" %}

{% block title %}

Операторы

{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="row">

<div class="col-md-12">

<table class="table table-hover" style="background-color: white">

<thead class="bg-info">

<tr>

<td>Имя</td>

<td>Фото</td>

<td>Телефон</td>

<td>Изменить</td>

<td>Удалить</td>

</tr>

</thead>

{% for operator in operator\_list %}

<tr>

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

<td><img src="{{ operator.photo.url }}" alt="" class="rounded" style="min-height: 200px; max-height: 200px; min-width: 200px; max-width: 200px;" ></td>

<td>{{ operator.number }}</td>

<td><a href="{% url "edit\_operator" operator.operator\_id %}" class="btn btn-primary" role="button">Изменить</a>

</td>

<td><a href="{% url "delete\_operator" operator.operator\_id %}" class="btn btn-secondary" role="button">Удалить</a></td>

</tr>

{% endfor %}

</table>

</div>

</div>

<div class="row">

<div class="col-md-2 offset-md-10">

<a href="{% url "admin\_menu" %}" class="btn btn-primary" role="button">На главную</a>

</div>

</div>

</div>

</div>

{% endblock content %}

Rank/show.html

{% extends "base.html" %}

{% block title %}

Классы

{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="row">

<div class="col-md-12">

<table class="table table-hover" style="background-color: white">

<thead class="bg-info">

<tr>

<td>Название класса</td>

<td>Изменить</td>

<td>Удалить</td>

</tr>

</thead>

{% for rank in rank\_list %}

<tr>

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

<td><a href="{% url "edit\_rank" rank.rank\_id%}" class="btn btn-primary" role="button">Изменить</a>

</td>

<td><a href="{% url "delete\_rank" rank.rank\_id%}" class="btn btn-secondary" role="button">Удалить</a></td>

{% endfor %}

</table>

</div>

</div>

<div class="row">

<div class="col-md-2 offset-md-10">

<a href="{% url "admin\_menu" %}" class="btn btn-primary" role="button">На главную</a>

</div>

</div>

</div>

{% endblock content %}

Rent/show.html

{% extends "base.html" %}

{% block title %}

Аренда

{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="row">

<div class="col-md-12">

<table class="table table-hover" style="background-color: white">

<thead class="bg-info">

<tr>

<td>Арендатор</td>

<td>Телефон</td>

<td>Оператор</td>

<td>Марка</td>

<td>Модель</td>

<td>Тариф</td>

<td>Время аренды</td>

<td>Изменить</td>

<td>Удалить</td>

</tr>

</thead>

{% for rent in rent\_list %}

<tr>

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

<td>{{ rent.phone }}</td>

<td>{{ rent.id\_operator.name }}</td>

<td>{{ rent.id\_model.mark\_id.name}}</td>

<td>{{ rent.id\_model.name }}</td>

<td>{{ rent.id\_tariff.name }}</td>

<td>{{ rent.time }}</td>

<td><a href="{% url "edit\_rent" rent.rent\_id %}" class="btn btn-primary" role="button">Изменить</a>

</td>

<td><a href="{% url "delete\_rent" rent.rent\_id %}" class="btn btn-secondary" role="button">Удалить</a></td>

{% endfor %}

</table>

</div>

</div>

<div class="row">

<div class="col-md-2 offset-md-10">

<a href="{% url "admin\_menu" %}" class="btn btn-primary" role="button">На главную</a>

</div>

</div>

</div>

{% endblock content %}

Tariff/show.html

{% extends "base.html" %}

{% block title %}

Марки

{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px;">

<div class="row">

<div class="col-md-12">

<table class="table table-hover" style="background-color: white">

<thead class="bg-info">

<tr>

<td>Название</td>

<td>Короткое описание</td>

<td>Цена</td>

<td>Изменить</td>

<td>Удалить</td>

</tr>

</thead>

{% for tariff in tariffs\_list %}

<tr>

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

<td>{{ tariff.description }}</td>

<td>{{ tariff.price }}</td>

<td><a href="{% url "edit\_tariff" tariff.tariff\_id %}" class="btn btn-primary" role="button">Изменить</a>

</td>

<td><a href="{% url "delete\_tariff" tariff.tariff\_id %}" class="btn btn-secondary" role="button">Удалить</a></td>

</tr>

{% endfor %}

</table>

</div>

</div>

<div class="row">

<div class="col-md-2 offset-md-10">

<a href="{% url "admin\_menu" %}" class="btn btn-primary" role="button">На главную</a>

</div>

</div>

</div>

{% endblock content %}

admin\_add.html

{% extends "base.html" %}

{% block title %}Admin add{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px">

<div class="row">

<div class="col-md-12">

<form enctype="multipart/form-data" action="" method="POST">

{% csrf\_token %}

<div align="center">

<table class="table table-hover bg-light">

{{ form.as\_table }}

</table>

<div>

<input type="submit" value="Добавить" class="btn btn-primary">

<a href="{% url "admin\_menu" %}" class="btn btn-danger">Назад</a>

</div>

</div>

</form>

</div>

</div>

</div>

{% endblock content %}

admin\_delete.html

{% extends 'base.html' %}

{% block title %}Delete{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px">

<form action="" method="POST">

{% csrf\_token %}

<div align="center">

<p>Хотите удалить: {{ name\_delete }} ?</p>

<input type="submit" value="Да" name="yes">

<input type="submit" value="Нет" name="no">

</div>

</form>

</div>

{% endblock content %}

admin\_edit.html

{% extends "base.html" %}

{% block title %}Admin\_Edit{% endblock title %}

{% block content %}

<div class="container" style="margin-top: 80px">

<form enctype="multipart/form-data" action="" method="POST">

{% csrf\_token %}

<div align="center">

<table border="1 solid black" cellpadding="5" cellspacing="0" style="text-align: center">

{{ form.as\_table }}

</table>

<div style="margin-top: 5px">

<input type="submit" value="Изменить" class="btn btn-primary">

<a href="{% url "admin\_menu" %}" class="btn btn-secondary"> На главную </a>

</div>

</div>

</form>

</div>

{% endblock content %}

admin\_menu.html

{% extends "base.html" %}

{% block content %}

<div class="container" style="margin-top: 80px">

<div class="row">

<div class="col-md-12">

<table class="table table-hover" style="background-color: white">

<thead class="bg-info">

<tr>

<td><a href="{% url "show\_rent" %}">Посмотреть аренды</a></td>

<td><a href="{% url "show\_rank" %}">Посмотреть классы</a></td>

<td><a href="{% url "show\_mark" %}">Посмотреть марки</a></td>

<td><a href="{% url "show\_operator" %}">Посмотреть операторов</a></td>

<td><a href="{% url "show\_tariffs" %}">Посмотреть тарифы</a></td>

<td><a href="{% url "show\_fuel" %}">Посмотреть топливо</a></td>

</tr>

<tr>

<td><a href="{% url 'add\_model' %}">Добавить модель</a></td>

<td><a href="{% url "add\_rank" %}">Добавить классы</a></td>

<td><a href="{% url "add\_mark" %}">Добавить марки</a></td>

<td><a href="{% url "add\_operator" %}">Добавить операторов</a></td>

<td><a href="{% url "add\_tariff" %}">Добавить тарифы</a></td>

<td><a href="{% url "add\_fuel" %}">Добавить топливо</a></td>

</tr>

</thead>

</table>

</div>

</div>

</div>

{% endblock %}