

Выполнил: Студент группы  
ИУ5-53Б  
Борисов Андрей  
Михайлович

# Задание

Создание прототипа веб-приложения с использованием фреймворка Django.

**Цель лабораторной работы:** изучение возможностей создания прототипа веб-приложения с использованием фреймворка Django.

## Требования к отчету:

Отчет по лабораторной работе должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

## Задание:

1. Создайте прототип веб-приложения с использованием фреймворка Django:
  - Создайте виртуальное окружение.
  - Установите в него Django.
  - Создайте проект и приложение Django.
2. Создайте представления и шаблоны (по желанию можно использовать модели), реализующие концепцию master/detail со следующей функциональностью:
  - На странице master в виде списка HTML выводится информация о трех объектах (например, о трех сортах мороженого). Каждая строка списка представляет собой гиперссылку, при нажатии на которую происходит переход к странице detail.
  - Страница detail содержит детальное описание объекта (сорта мороженого), фотографию, гиперссылку на master-страницу.
  - Фотография относится к статическому содержимому сайта.
  - Страница detail должна выводить данные с использованием таблицы HTML.
  - Шаблон страницы detail получает от представления данные о детальном объекте с использованием контекста.
  - **НЕОБЯЗАТЕЛЬНЫЙ ПУНКТ.** По желанию можно использовать верстку с применением Bootstrap (или аналогичного фреймворка), а также представления на основе классов (class-based views).

# Текст программы

## Models.py

```
from django.db import models

class Customer(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=20)
    surname = models.CharField(max_length=20)
    phone_number = models.CharField(max_length=20)

    class Meta:
        db_table = 'customer'

class Cars(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=20)
    price = models.IntegerField()
    speed = models.IntegerField()
    colour = models.CharField(max_length=10)
    customer = models.ForeignKey(Customer, on_delete=models.PROTECT)

    class Meta:
        db_table = 'cars'
```

## Views.py

```
from django.shortcuts import render

carsList = [
    {'name': 'Audi R8', 'speed': 300, 'cost': 5, 'picture': 'audi.jpg', 'colour': 'белый'},
    {'name': 'Bugatti Veyron', 'speed': 310, 'cost': 7, 'picture': 'bugatti.jpg', 'colour': 'черный'},
    {'name': 'Dodge Viper', 'speed': 290, 'cost': 9, 'picture': 'dodge.jpg', 'colour': 'красный'}
]

def cars(request):
    return render(request, 'cars.html', {'data': {
        'cars': [
            {'name': 'Audi R8', 'id': 1},
            {'name': 'Bugatti Veyron', 'id': 2},
            {'name': 'Dodge Viper', 'id': 3},
        ]
    }})

def car(request, id):
    return render(request, 'car.html', {'data': carsList[id - 1]})
```

## Urls.py

```
from django.contrib import admin
from django.urls import path

from Models_test import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.cars),
    path('car/<int:id>', views.car, name='car_url'),
]
```

## Migrations.py

```
class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Customer',
            fields=[
                ('id', models.IntegerField(primary_key=True, serialize=False)),
                ('name', models.CharField(max_length=20)),
                ('surname', models.CharField(max_length=20)),
                ('phone_number', models.CharField(max_length=20)),
            ],
            options={
                'db_table': 'customer',
            },
        ),
        migrations.CreateModel(
            name='Cars',
            fields=[
                ('id', models.IntegerField(primary_key=True, serialize=False)),
                ('name', models.CharField(max_length=20)),
                ('price', models.IntegerField()),
                ('speed', models.IntegerField()),
                ('colour', models.CharField(max_length=10)),
                ('customer', models.ForeignKey(on_delete=django.db.models.deletion.PROTECT, to='Models_test.customer')),
            ],
        ),
    ]
```

## Base.html

```

    <!doctype html>
    <html lang="en" class="h-100">
    <head>
        <meta charset="utf-8">
        <title>{% block title %}{% endblock %}</title>
    </head>
    <body>
        {% block content %}{% endblock %}
    </body>
    </html>

```

## Car.html

```

    <!doctype html>
    <html lang="en" class="h-100">
    <head>
        <meta charset="utf-8">
        <title>{% block title %}{% endblock %}</title>
    </head>
    <body>
        {% block content %}{% endblock %}
    </body>
    </html>

```

## Cars.html

```

{% extends 'base.html' %}

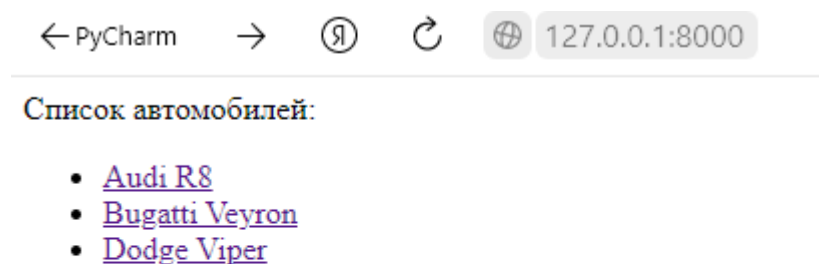
{% block title %}Список авто{% endblock %}

{% block content %}
    <div>Список автомобилей:</div>
    <ul>
        {% for car in data.cars %}
            <li><a href="{% url 'car_url' car.id %}">{{ car.name }}</a> </li>
        {% empty %}
            <li>Список пуст</li>
        {% endfor %}
    </ul>
{% endblock %}

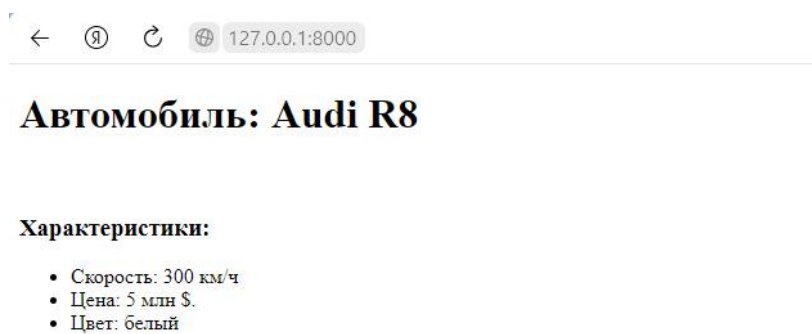
```

# Результаты работы программы

Страница cars.html:



Страница car.html:



## Автомобиль: Bugatti Veyron

### Характеристики:

- Скорость: 310 км/ч
- Цена: 7 млн \$.
- Цвет: черный



## Автомобиль: Dodge Viper

### Характеристики:

- Скорость: 290 км/ч
- Цена: 9 млн \$.
- Цвет: красный

