

Отчет по РК № 2 по курсу
"Разработка Интернет-Приложений"

Выполнил:
Студентка
группы
ИУ5-55Б
Богданова В.В.

Задание:

Рубежный контроль представляет собой разработку веб-приложения с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.\

«Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список всех Водителей, у которых фамилия заканчивается на «ов», и названия их Автопарков.

«Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список Автопарков со средней зарплатой Водителей в каждом Автопарке, отсортированный по средней зарплате.

№ варианта	Класс 1	Класс 2
3	Водитель	Автопарк

Текст программы:

rk2/urls.py

```
from django.contrib import admin
from django.urls import path, include
from drivers import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name="main"),
    path('driver/', include([
        path('', views.read_driver, name='read_driver'),
        path('create/', views.create_driver, name="create_driver"),
        path('update/<int:driver_id>', views.update_driver,
name="update_driver"),
        path('delete/<int:driver_id>', views.delete_driver,
name="delete_driver"),
    ])),
]
```

```

    path('autopark/', include([
        path('', views.read_autopark, name='read_autopark'),
        path('create/', views.create_autopark, name="create_autopark"),
        path('update/<int:autopark_id>/',
            views.update_autopark, name="update_autopark"),
        path('delete/<int:autopark_id>/',
            views.delete_autopark, name="delete_autopark"),
    ])),
    path('report/', views.report, name="report"),
]

```

drivers/models.py

```

from django.db import models

class Autopark(models.Model):
    name = models.CharField(max_length=256, verbose_name="Autopark name")

    def __str__(self):
        return self.name

class Driver (models.Model):
    name = models.CharField(max_length=256, verbose_name="Driver name")
    salary = models.PositiveIntegerField(verbose_name="Driver's salary")
    autopark = models.ForeignKey(
        Autopark,
        on_delete=models.SET_DEFAULT,
        null=True,
        default=None,
        related_name="drivers"
    )

    def __str__(self):
        return self.name

```

drivers/views.py

```

from django.db.models.aggregates import Avg
from django.http.response import HttpResponseRedirect
from django.shortcuts import get_list_or_404, get_object_or_404, redirect, render
from .models import Autopark, Driver
from django.db import models

def index(request):
    return render(request, 'index.html')

# drivers

def read_driver(request):

```

```

drivers = Driver.objects.all()
return render(request, 'driver/driver_list.html', {'drivers': drivers})

def create_driver(request):
    if request.method == 'GET':
        autoparks = Autopark.objects.all()
        return render(request, 'driver/create_driver.html', {"autoparks":
autoparks})
    else:
        dto = {}
        for key in request.POST:
            if key in Driver.__dict__:
                dto[key] = request.POST[key]
        dto['autopark'] = get_object_or_404(
            Autopark, pk=request.POST['autopark'])
        new_driver = Driver(**dto)
        new_driver.save()
        return redirect('read_driver')

def update_driver(request, driver_id):
    if request.method == 'GET':
        autoparks = Autopark.objects.all()
        driver = get_object_or_404(Driver, pk=driver_id)
        return render(request, 'driver/update_driver.html', {"driver": driver,
"autoparks": autoparks})
    else:
        driver = get_object_or_404(Driver, pk=driver_id)
        for key in request.POST:
            if key in driver.__dict__ and key != 'autopark':
                setattr(driver, key, request.POST[key])
        if 'autopark' in request.POST:
            setattr(driver, 'autopark', get_object_or_404(
                Autopark, pk=request.POST['autopark']))
        driver.save()
        return redirect('read_driver')

def delete_driver(request, driver_id):
    driver = get_object_or_404(Driver, pk=driver_id)
    driver.delete()
    return redirect(request.META.get('HTTP_REFERER'))

# autoparks

def read_autopark(request):
    autoparks = Autopark.objects.all()
    return render(request, 'autopark/autopark_list.html', {'autoparks':
autoparks})

def create_autopark(request):
    if request.method == 'GET':
        return render(request, 'autopark/create_autopark.html')
    else:
        new_autopark = Autopark(name=request.POST['name'])

```

```

        new_autopark.save()
        return redirect('read_autopark')

def update_autopark(request, autopark_id):
    if request.method == 'GET':
        autopark = get_object_or_404(Autopark, pk=autopark_id)
        return render(request, 'autopark/update_autopark.html', {"autopark":
autopark})
    else:
        autopark = get_object_or_404(Autopark, pk=autopark_id)
        for key in request.POST:
            if key in autopark.__dict__:
                setattr(autopark, key, request.POST[key])
        autopark.save()
        return redirect('read_autopark')

def delete_autopark(request, autopark_id):
    autopark = get_object_or_404(Autopark, pk=autopark_id)
    autopark.delete()
    return redirect(request.META.get('HTTP_REFERER'))

# REPORT

def report(request):
    # return HttpResponse('♥')
    found_drivers = Driver.objects.filter(name__endswith='OB')
    average_salaries = []
    for autopark in Autopark.objects.all():
        average_salaries.append({"autopark": autopark, "salary":
round(Driver.objects.filter(
            autopark=autopark.pk).aggregate(Avg('salary'))['salary__avg'] or 0)})
    return render(request, 'report.html', {"found_drivers": found_drivers,
"average_salaries": sorted(average_salaries, key=lambda salary: salary['salary'])

```

templates/inc/base.html

```

{% load static %}

<!doctype html>
<html lang="en" class="h-100">

<head>
    <!-- ===== CHARSET AND VIEWPORT ===== -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- ===== CACHE TURN OFF ===== -->
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <!-- ===== COPY RIGHT ===== -->
    <meta name="author" content="IU5-55 Valeria Bogdanova">
    <title>
        IU5-55 Bogdanova RK2

```

```

</title>

<!-- ===== BOOTSTRAP CORE CSS ===== -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWbQ78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
<!-- ===== SHARED STYLES ===== -->
<link rel="stylesheet" href="{% static 'css/navbar.css' %}">
</head>

<body class="d-flex flex-column h-100">
  <header>
    <!-- Navbar -->
    {% include 'inc/navbar.html' %}
  </header>

  <!-- Begin page content -->
  <main role="main" class="flex-shrink-0">
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <!-- Content -->
          {% block content %} {% endblock content %}
        </div>
      </div>
    </div>
  </main>

  <footer class="footer mt-auto py-3">
    <!-- Footer -->
  </footer>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

</html>

```

templates/inc/navbar.html

```
{% load static %}
<nav class="navbar navbar-expand-sm navbar-dark bg-dark p-3">
  <ul class="navbar-nav nav-tabs ml-auto">
    <li class="nav-item">
      <a href="{% url 'main' %}" class="nav-link">Main Page</a>
    </li>
    <li class="nav-item">
      <a href="{% url 'read_driver' %}" class="nav-link">All drivers</a>
    </li>
    <li class="nav-item">
      <a href="{% url 'create_driver' %}" class="nav-link">Create driver</a>
    </li>
    <li class="nav-item">
      <a href="{% url 'read_autopark' %}" class="nav-link">All autoparks</a>
    </li>
    <li class="nav-item">
      <a href="{% url 'create_autopark' %}" class="nav-link">Create
autopark</a>
    </li>
    <li class="nav-item">
      <a href="{% url 'report' %}" class="nav-link">Report</a>
    </li>
  </ul>
</nav>
```

templates/driver/driver_card.html

```
{% load static %}

<div class="card mb-3">
    <div class="row g-0">
        <div class="col-md-4">
            
        </div>
        <div class="col-md-8">
            <div class="card-body">
                <h5 class="card-title">{{ driver.name }}</h5>
                <p class="card-text">{{ driver.autopark.name }}</p>
                <p class="card-text">
                    This is a wider card with supporting text below as a natural
                    lead-in to additional content. This content is a little bit longer.
                </p>
                <p class="card-text"><small class="text-muted">{{ driver.salary
            }}</small></p>
                <a href="{% url 'delete_driver' driver.pk %}" class="btn btn-
            danger"> Delete </a>
                <a href="{% url 'update_driver' driver.pk %}" class="btn btn-
            primary"> Update </a>
            </div>
        </div>
    </div>
</div>
```

templates/driver/driver_form.html

```
<div class="mb-3">
    <label for="driverName" class="form-label">Driver Name</label>
    <input type="text" name="name" value="{{ driver.name }}" required="required"
class="form-control" id="driverName" />
</div>

<div class="mb-3">
    <label for="driverSalary" class="form-label">Driver Salary</label>
    <input type="number" min="0" name="salary" value="{{ driver.salary }}"
required="required" class="form-control" id="driverSalary" />
</div>

<div class="mb-3">
    <label for="driverAutopark" class="form-label">Select Autopark</label>
    <select name="autopark" class="form-select" aria-label="Default select
example" id="driverAutopark">
        {% for autopark in autoparks %}
            <option value="{{ autopark.pk }}">{{ autopark.name }}</option>
        {% endfor %}
    </select>
</div>
```


templates/driver/driver_list.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">All Drivers</h1>
</div>
<!-- Page Content -->
<div class="d-flex flex-column">
    {% for driver in drivers %} {% include 'driver/driver_card.html' with
driver=driver %} {% endfor %}
</div>
{% endblock content %}
```

templates/driver/create_driver.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">New Driver</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'create_driver' %}" method="POST" class="form">
        {% csrf_token %} {% include 'driver/driver_form.html' with driver=driver
%}
        <button type="submit" class="btn btn-primary">Create</button>
    </form>
</div>
{% endblock content %}
```

templates/driver/update_driver.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Update Driver</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'update_driver' driver.pk %}" method="POST" class="form">
        {% csrf_token %} {% include 'driver/driver_form.html' with driver=driver
%}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock content %}
```

templates/autopark/autopark_card.html

```
{% load static %}

<div class="card m-3" style="width: 18rem">
    
    <div class="card-body">
        <h5 class="card-title">{{ autopark.name }}</h5>
        <p class="card-text">
            Some quick example text to build on the card title and make up the
            bulk of the card's content.
        </p>
        <a href="{% url 'delete_autopark' autopark.pk %}" class="btn btn-danger">
Delete </a>
        <a href="{% url 'update_autopark' autopark.pk %}" class="btn btn-primary">
Update </a>
    </div>
</div>
```

templates/autopark/autopark_form.html

```
<div class="mb-3">
    <label for="autoparkName" class="form-label">Autopark Name</label>
    <input type="text" name="name" value="{{ autopark.name }}" required="required"
class="form-control" id="autoparkName" />
</div>
```

templates/autopark/autopark_list.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">All Autoparks</h1>
</div>
<!-- Page Content -->
<div class="d-flex flex-wrap">
    {% for autopark in autoparks %} {% include 'autopark/autopark_card.html' with
autopark=autopark %} {% endfor %}
</div>
{% endblock content %}
```

templates/autopark/create_autopark.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">New Autopark</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'create_autopark' %}" method="POST" class="form">
```

```
        {% csrf_token %} {% include 'autopark/autopark_form.html' with
autopark=autopark %}
        <button type="submit" class="btn btn-primary">Create</button>
    </form>
</div>
{% endblock content %}
```

templates/autopark/update_autopark.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Update Autopark</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'update_autopark' autopark.pk %}" method="POST"
class="form">
        {% csrf_token %} {% include 'autopark/autopark_form.html' with
autopark=autopark %}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock content %}
```

templates/report.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Report Page</h1>
</div>
<!-- Page Content -->
<div>
    <h1>Drivers with name ends with 'ob'</h1>
    {% for driver in found_drivers %} {% include 'driver/driver_card.html' with
driver=driver %} {% endfor %}
</div>

<div>
    <h1>Average salary</h1>
    <div class="d-flex flex-wrap">
        {% for avg_salary in average_salaries %}
        <div class="d-flex flex-column align-center">
            <h3 class="text-center">Average salary {{ avg_salary.salary }}</h3>
            {% include 'autopark/autopark_card.html' with
autopark=avg_salary.autopark %}
        </div>
        {% endfor %}
    </div>
</div>
{% endblock content %}

```

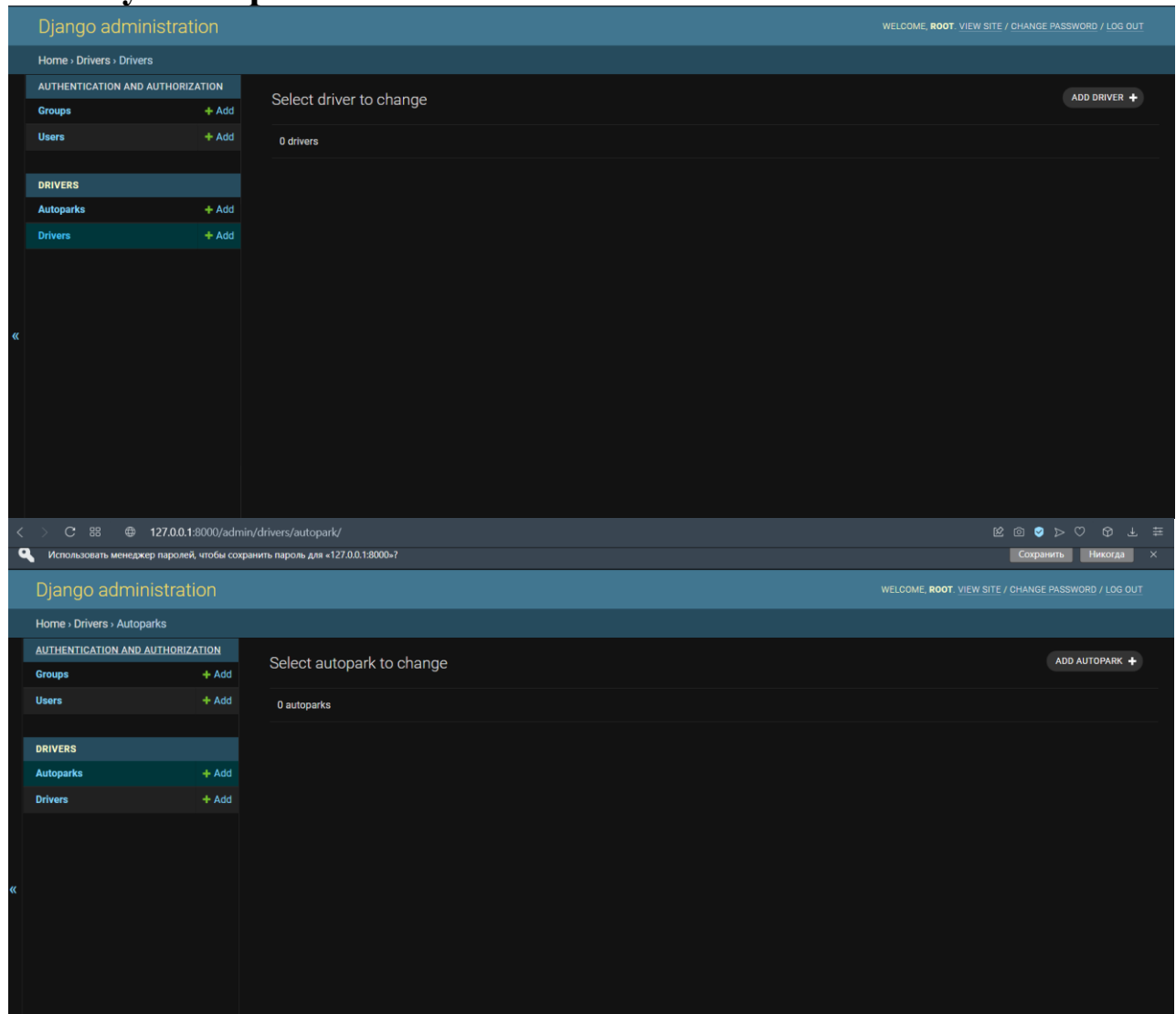
templates/index.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Main Page</h1>
</div>
<!-- Page Content -->
<div>
    
</div>
{% endblock content %}

```

Результат работы:



Main Page



New Autopark

Autopark Name

Тачки

Create

All Autoparks



Тачки

Some quick example text to build on the card title and make up the bulk of the card's content.

Delete

Update

New Autopark

Autopark Name

Форсаж

Create

All Autoparks



Тачки

Some quick example text to build on the card title and make up the bulk of the card's content.

[Delete](#)[Update](#)

Форсаж

Some quick example text to build on the card title and make up the bulk of the card's content.

[Delete](#)[Update](#)

New Driver

Driver Name

Driver Salary

Select Autopark

[Create](#)

All Drivers



Сергей

Тачки

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

50000

[Delete](#)[Update](#)

[Main Page](#) [All drivers](#) [Create driver](#) [All autoparks](#) [Create autopark](#) [Report](#)

New Driver

Driver Name

Геннадий

Driver Salary

20000



Select Autopark

Форсаж

Create

[Main Page](#) [All drivers](#) [Create driver](#) [All autoparks](#) [Create autopark](#) [Report](#)



All Drivers

	<div>Сергей</div> <div>Тачки</div> <div>This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</div> <div>50000</div> <div>DeleteUpdate</div>
	<div>Геннадий</div> <div>Форсаж</div> <div>This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</div> <div>20000</div> <div>DeleteUpdate</div>

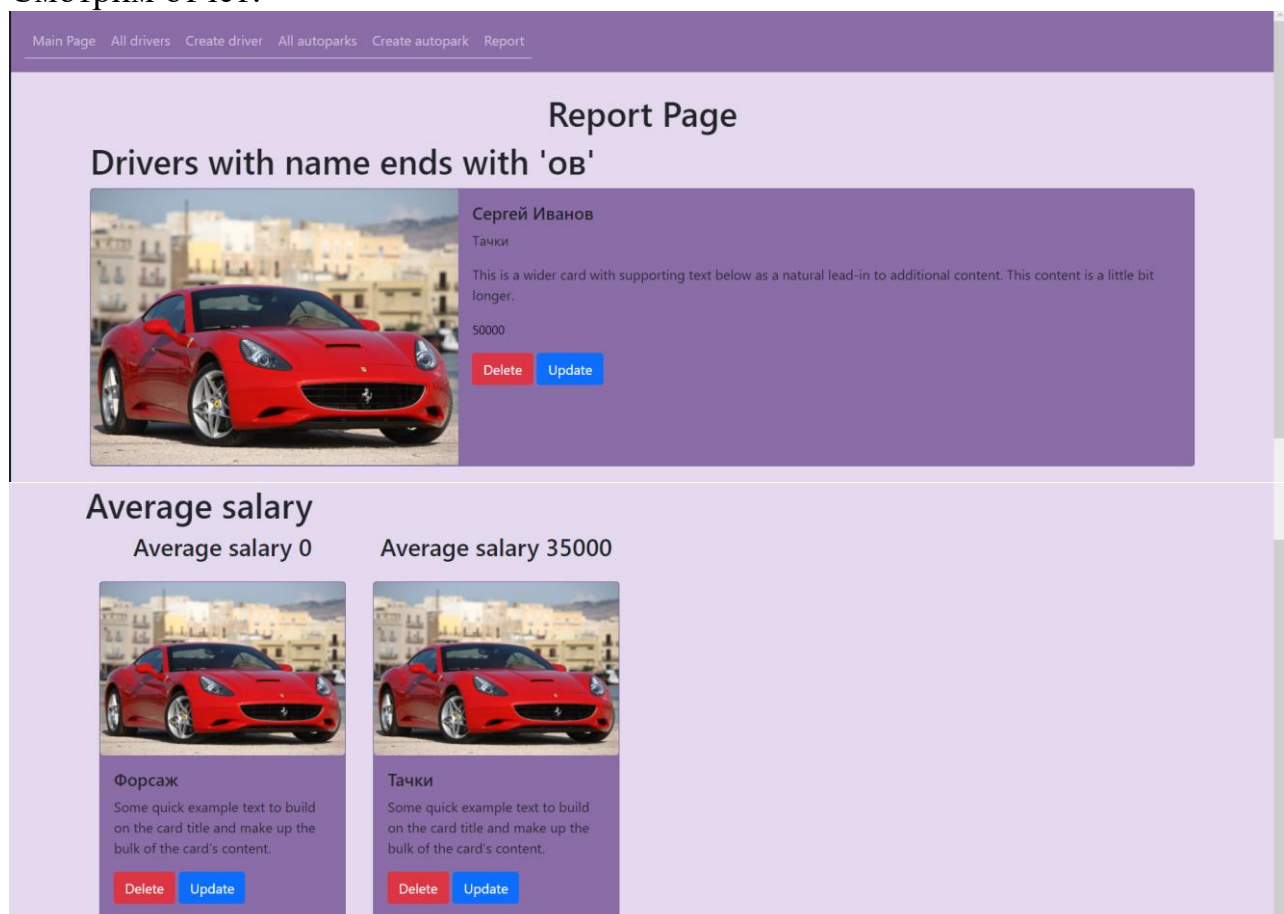
По кнопке update меняем имена водителей

[Main Page](#) [All drivers](#) [Create driver](#) [All autoparks](#) [Create autopark](#) [Report](#)

All Drivers

	<div>Сергей Иванов</div> <div>Тачки</div> <div>This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</div> <div>50000</div> <div>DeleteUpdate</div>
	<div>Геннадий Сериченко</div> <div>Тачки</div> <div>This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</div> <div>20000</div> <div>DeleteUpdate</div>

Смотрим отчет:



Django administration

WELCOME **root** / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Drivers > Autoparks

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

DRIVERS

Autoparks + Add

Drivers + Add

Select autopark to change

ADD AUTOPARK +

Action: Go 0 of 2 selected

☐ АУТОПАРК

☐ Форсаж

☐ Тачки

2 autoparks

Django administration

WELCOME **root** / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Drivers > Drivers

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

DRIVERS

Autoparks + Add

Drivers + Add

Select driver to change

ADD DRIVER +

Action: Go 0 of 2 selected

☐ DRIVER

☐ Геннадий Сериченко

☐ Сергей Иванов

2 drivers