Московский государственный технический университет им. Н.Э. Баумана

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчёт по рубежному контролю №2
Вариант 3

Выполнил:
студент группы РТ5-51Б
Борисочкин М.И.
Подпись и дата:

Проверил:
к.т.н., доцент
Гапанюк Ю. Е.
Подпись и дата:

Москва, 2021 г.

Описание задания

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.

Исходные данные

Класс 1	Класс 2
Водитель	Автопарк

Текст программы

models.py

```
from django.db import models
from django.urls import reverse


class Carstation(models.Model):
    """Модель Автопарк"""
    name = models.CharField(max_length=50, verbose_name='Название')
    adress = models.CharField(max_length=150, verbose_name='Адрес')

    class Meta:
        verbose_name = 'Автопарк'
        verbose_name_plural = 'Автопарки'

    def __str__(self):
        return self.name

    def get_absolute_url(self):
        return reverse('Drivers:carstations')


class Driver(models.Model):
    """Модель Водитель"""
    second_name = models.CharField(max_length=50, verbose_name='Фамилия')
    first_name = models.CharField(max_length=50, verbose_name='Имя')
    salary = models.DecimalField(max_digits=7, decimal_places=2,
verbose_name='Заработная плата')
    carstation_id = models.ForeignKey(Carstation, null=True,
on_delete=models.SET_NULL, verbose_name='Автопарк')

    class Meta:
        verbose_name = 'Водитель'
        verbose_name_plural = 'Водители'

    def __str__(self):
        return self.second_name + " " + self.first_name

    def get_absolute_url(self):
        return reverse('Drivers:driver_detail', args=[str(self.id)])
```

RK2\urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('drivers/', include('Drivers.urls'))
]
```

Drivers\urls.py

```
from django.urls import path

from . import views


app_name = 'Drivers'
urlpatterns = [
    path('', views.index, name='index'),
    path('drivers/', views.DriverListView.as_view(), name='drivers'),
    path('drivers/<int:pk>/', views.DriverDetailView.as_view(),
name='driver_detail'),
    path('drivers/add/', views.DriverCreateView.as_view(),
name='driver_add'),
    path('driver/<int:pk>/edit', views.DriverUpdateView.as_view(),
name='driver_edit'),
    path('driver/<int:pk>/delete', views.DriverDeleteView.as_view(),
name='driver_delete'),
    path('carstations/', views.CarstationListView.as_view(),
name='carstations'),
    path('carstations/add/', views.CarstationCreateView.as_view(),
name='carstation_add'),
    path('carstations/<int:pk>/edit', views.CarstationUpdateView.as_view(),
name='carstation_edit'),
    path('carstations/<int:pk>/delete', views.CarstationDeleteView.as_view(),
name='carstation_delete'),
    path('report/', views.report, name='report')
]
```

forms.py

```
from .models import Driver, Carstation
from django.forms import ModelForm, TextInput, NumberInput, Select


class DriverCreateFrom(ModelForm):
    """Форма для добавления водителя"""
    class Meta:
        model = Driver
        fields = ['second_name', 'first_name', 'salary', 'carstation_id']
        widgets = {
            'second_name': TextInput(attrs={'class': 'form-control',
'placeholder': 'Введите фамилию'}),
            'first_name': TextInput(attrs={'class': 'form-control',
'placeholder': 'Введите имя'}),
            'salary': NumberInput(attrs={'class': 'form-control',
'placeholder': 'Введите зарплату'}),
            'carstation_id': Select(attrs={'class': 'form-select'})
        }


class DriverUpdateForm(ModelForm):
    """Форма для редактирования водителя"""
    class Meta:
        model = Driver
        fields = ['salary', 'carstation_id']
        widgets = {
            'salary': NumberInput(attrs={'class': 'form-control',
'placeholder': 'Введите зарплату'}),
            'carstation_id': Select(attrs={'class': 'form-select'})
        }


class CarstationCreateForm(ModelForm):
    """Форма для добавления автопарка"""
    class Meta:
        model = Carstation
        fields = ['name', 'adress']
        widgets = {
            'name': TextInput(attrs={'class': 'form-control', 'placeholder':
'Введите название'}),
            'adress': TextInput(attrs={'class': 'form-control',
'placeholder': 'Введите адрес'})
        }


class CarstationUpdateForm(ModelForm):
    """Форма для изменения автопарка"""
    class Meta:
        model = Carstation
        fields = ['adress']
        widgets = {
            'adress': TextInput(attrs={'class': 'form-control',
'placeholder': 'Введите адрес'})
        }
```

views.py

```
from django.shortcuts import render
from django.views import generic
from django.urls import reverse_lazy

from .models import Driver, Carstation
from .forms import DriverCreateFrom, DriverUpdateForm, CarstationCreateForm,
CarstationUpdateForm


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


class DriverListView(generic.ListView):
    """Список водителей"""
    template_name = 'Drivers/Driver/List.html'
    context_object_name = 'driver_list'

    def get_queryset(self):
        return Driver.objects.order_by('second_name', 'first_name')


class DriverDetailView(generic.DetailView):
    """Информация о конкретном водителе"""
    model = Driver
    template_name = 'Drivers/Driver/Detail.html'


class DriverCreateView(generic.CreateView):
    """Добавление водителя"""
    model = Driver
    template_name = 'Drivers/Driver/Create.html'
    form_class = DriverCreateFrom


class DriverUpdateView(generic.UpdateView):
    """Изменение водителя"""
    model = Driver
    template_name = 'Drivers/Driver/Update.html'
    form_class = DriverUpdateForm


class DriverDeleteView(generic.DeleteView):
    """Удаление водителя"""
    model = Driver
    template_name = 'Drivers/Driver/Delete.html'
    success_url = reverse_lazy('Drivers:drivers')


class CarstationListView(generic.ListView):
    """Список автопарков"""
    template_name = 'Drivers/Carstation/List.html'
    context_object_name = 'carstation_list'

    def get_queryset(self):
        return Carstation.objects.order_by('name')


class CarstationCreateView(generic.CreateView):
    """Добавление автопарка"""
    model = Carstation
```

```
    template_name = 'Drivers/Carstation/Create.html'
    form_class = CarstationCreateForm


class CarstationUpdateView(generic.UpdateView):
    """Изменение автопарка"""
    model = Carstation
    template_name = 'Drivers/Carstation/Update.html'
    form_class = CarstationUpdateForm


class CarstationDeleteView(generic.DeleteView):
    """Удаление автопарка"""
    model = Carstation
    template_name = 'Drivers/Carstation/Delete.html'
    success_url = reverse_lazy('Drivers:carstations')


def report(request):
    top_drivers =
Driver.objects.select_related('carstation_id').filter(salary__gte=50000)\
        .order_by('-salary', 'second_name', 'first_name')
    return render(request, 'Drivers/report.html', {'top_drivers':
top_drivers})
```

templates\Drivers\base.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}BASE{% endblock %}</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css
">
</head>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-
bottom">
      <a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto
text-dark text-decoration-none">
        <span class="fs-4 ms-3">Борисочкин М. И. РТ5-51Б РК2</span>
      </a>

      <ul class="nav nav-pills">
        <li class="nav-item"><a href="{% url 'Drivers:index' %}" class="nav-
link">Главная</a></li>
        <li class="nav-item"><a href="{% url 'Drivers:drivers' %}"
class="nav-link">Водители</a></li>
        <li class="nav-item"><a href="{% url 'Drivers:carstations'%}"
class="nav-link">Автопарки</a></li>
        <li class="nav-item"><a href="{% url 'Drivers:report' %}" class="nav-
link">Отчёт</a></li>
      </ul>
    </header>
{% block content %}
{% endblock %}
</body>
</html>
```

templates\Drivers\index.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Главная{% endblock %}

{% block content %}
    <h1 class="ms-3">Условия рубежного контроля №2 по курсу РИП</h1>
    <p class="ms-3 fs-5">Рубежный контроль представляет собой разработку веб-
приложения с использованием фреймворка Django.
        Веб-приложение должно выполнять следующие функции:</p>
    <ol class="ms-3 fs-5">
        <li>Создайте проект Python Django с использованием стандартных
средств Django.</li>
        <li>Создайте модель Django ORM, содержащую две сущности, связанные
отношением один-ко-многим в соответствии
            с Вашим вариантом из условий рубежного контроля №1.</li>
        <li>С использованием стандартного механизма Django сгенерируйте по
модели макет веб-приложения,
            позволяющий добавлять, редактировать и удалять данные.</li>
        <li>Создайте представление и шаблон, формирующий отчет, который
содержит соединение данных из двух таблиц. </li>
    </ol>
{% endblock %}
```

templates\Drivers\Driver\List.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Водители{% endblock %}

{% block content %}
    <ul class="list-unstyled">
    {% for driver in driver_list %}
        <li class="fs-4 ms-3"><a class="text-decoration-none link-dark"
href="{% url 'Drivers:driver_detail' driver.id %}">
            {{ driver }}</a></li>
    {% endfor %}
    </ul>
    <a href="{% url 'Drivers:driver_add' %}" class="btn btn-primary ms-3 mt-
5"
           role="button">Добавить водителя</a>
{% endblock %}
```

templates\Drivers\Driver\Detail.html

```
{% extends 'Drivers/base.html' %}
{% block title %}{{ driver }}{% endblock %}

{% block content %}
    <ul class="list-unstyled ms-3 fs-4">
        <li><strong>Фамилия:</strong> {{ driver.second_name }}</li>
        <li><strong>Имя:</strong> {{ driver.first_name }}</li>
        <li><strong>Автопарк:</strong> {{ driver.carstation_id }}</li>
        <li><strong>Зарплата:</strong> {{ driver.salary }}</li>
    </ul>
    <a href="{% url 'Drivers:driver_edit' driver.id %}" class="btn btn-
primary ms-3 mt-5"
           role="button">Изменить</a>
    <a href="{% url 'Drivers:driver_delete' driver.id %}" class="btn btn-
danger ms-3 mt-5"
           role="button">Удалить</a>
{% endblock %}
```

templates\Drivers\Driver\Create.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Добавить водителя{% endblock %}

{% block content %}
    <h1 class="ms-3">Новый водитель</h1>
    <form method="post" class="ms-3 col-sm-2">{% csrf_token %}
    {{ form.as_p}}
        <button type="submit" class="btn btn-success">Сохранить</button>
    </form>
{% endblock %}
```

templates\Drivers\Driver\Update.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Изменить водителя{% endblock %}

{% block content %}
    <h1 class="ms-3">Введите новые данные</h1>
    <form method="post" class="ms-3 col-sm-2">{% csrf_token %}
    {{form.as_p}}
        <button type="submit" class="btn btn-success">Сохранить</button>
    </form>
{% endblock %}
```

templates\Drivers\Driver\Delete.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Удалить водителя{% endblock %}

{% block content %}
    <h1 class="ms-3">Удалить водителя</h1>
    <form method="post" class="ms-3 col-sm-3">{% csrf_token %}
    <p>Вы действительно хотите удалить данного водителя?</p>
        <button type="submit" class="btn btn-success">Потвердить</button>
    </form>
{% endblock %}
```

templates\Drivers\Carstation\List.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Автопарки{% endblock %}

{% block content %}
    <dl class="row justify-content-start">
    {% for carstation in carstation_list %}
        <dt class="col-sm-2 fs-4 ms-3 my-1 pe-0">{{ carstation.name }}</dt>
        <dd class="col-sm-3 fs-4 mx-2 my-1 px-0">{{ carstation.adress }}</dd>
        <a href="{% url 'Drivers:carstation_edit' carstation.id %}"
class="btn btn-primary col-sm-1 m-1"
           role="button">Изменить</a>
        <a href="{% url 'Drivers:carstation_delete' carstation.id %}"
class="btn btn-danger col-sm-1 m-1"
           role="button">Удалить</a>
        <div class="col-sm-3 my-1"></div>
    {% endfor %}
    </dl>
    <a href="{% url 'Drivers:carstation_add' %}" class="btn btn-primary col-
sm-2 ms-3 mt-5"
           role="button">Добавить автопарк</a>
{% endblock %}
```

templates\Drivers\Carstation\Create.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Добавить автопарк{% endblock %}

{% block content %}
    <h1 class="ms-3">Новый автопарк</h1>
    <form method="post" class="ms-3 col-sm-2">{% csrf_token %}
    {{ form.as_p}}
        <button type="submit" class="btn btn-success">Сохранить</button>
    </form>
{% endblock %}
```

templates\Drivers\Carstation\Update.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Изменить автопарк{% endblock %}

{% block content %}
    <h1 class="ms-3">Введите новые данные</h1>
    <form method="post" class="ms-3 col-sm-2">{% csrf_token %}
    {{form.as_p}}
        <button type="submit" class="btn btn-success">Сохранить</button>
    </form>
{% endblock %}
```

templates\Drivers\Carstation\Delete.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Удалить автопарк{% endblock %}

{% block content %}
    <h1 class="ms-3">Удалить автопарк</h1>
    <form method="post" class="ms-3 col-sm-3">{% csrf_token %}
    <p>Вы действительно хотите удалить данный автопарк?</p>
        <button type="submit" class="btn btn-success">Потвердить</button>
    </form>
{% endblock %}
```

templates\Drivers\report.html

```
{% extends 'Drivers/base.html' %}
{% block title %}Отчёт{% endblock %}

{% block content %}
    <p class="h2">Отчёт "Водители, зарабатывющие более 50к"</p>
    <table class="table table-info">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">Фамилия</th>
      <th scope="col">Имя</th>
      <th scope="col">Зарплата</th>
      <th scope="col">Автопарк</th>
      <th scope="col">Адрес</th>
    </tr>
  </thead>
  <tbody>
  {% for top_driver in top_drivers %}
      <tr>
      <th scope="row">{{ forloop.counter }}</th>
      <td>{{ top_driver.second_name }}</td>
      <td>{{ top_driver.first_name }}</td>
      <td>{{ top_driver.salary }}</td>
      <td>{{ top_driver.carstation_id.name }}</td>
      <td>{{ top_driver.carstation_id.adress }}</td>
    </tr>
  {% endfor %}
  </tbody>
</table>
{% endblock %}
```

Результат выполнения программы

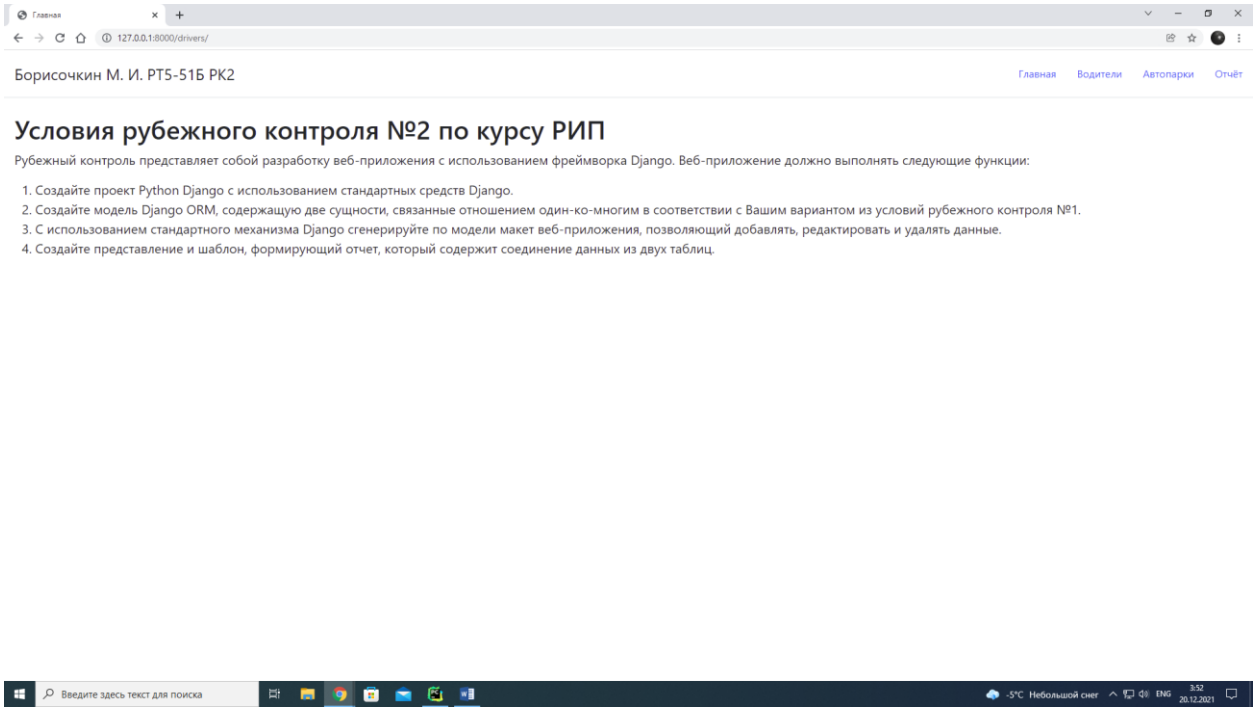

Рисунок 1. Главная страница

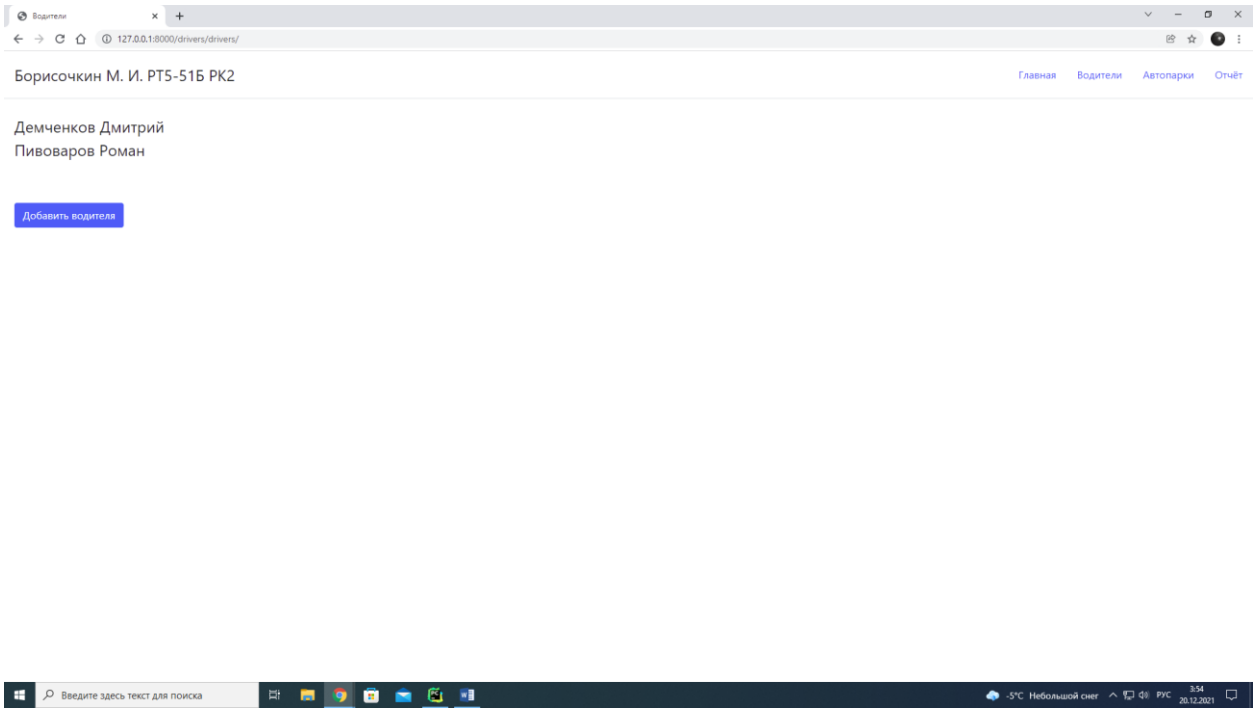

Рисунок 2. Список водителей

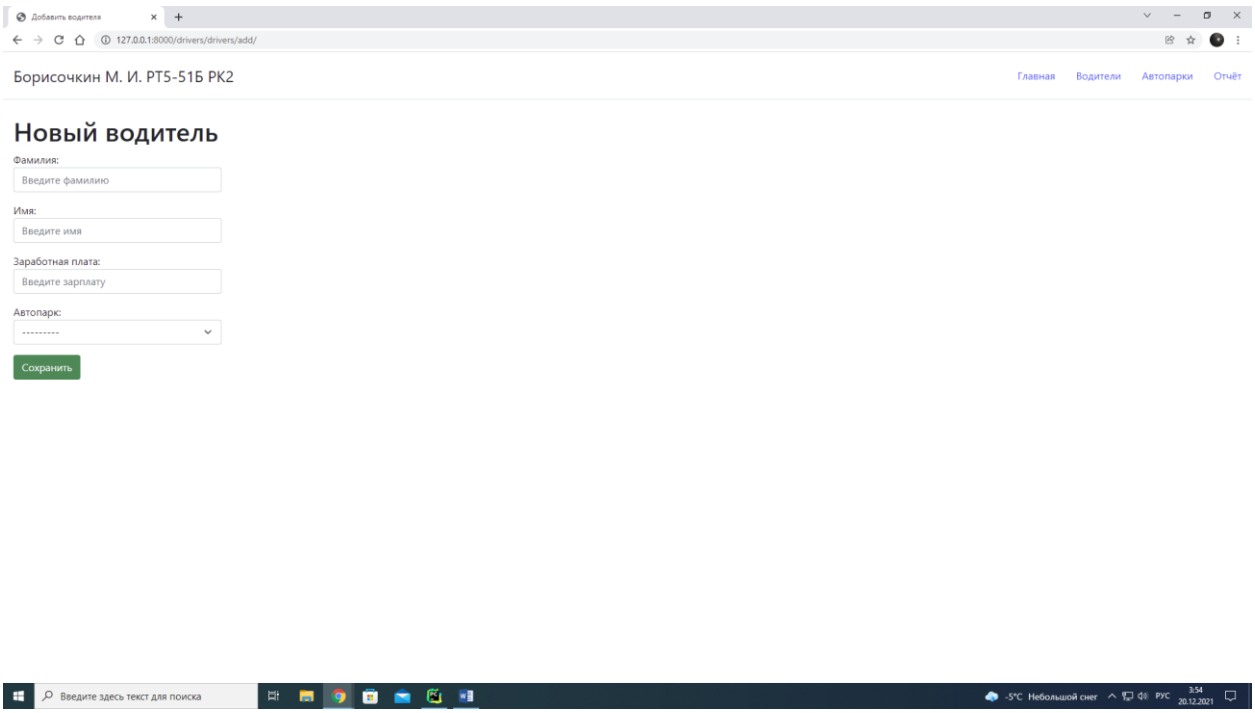

Рисунок 3. Форма добавления водителя

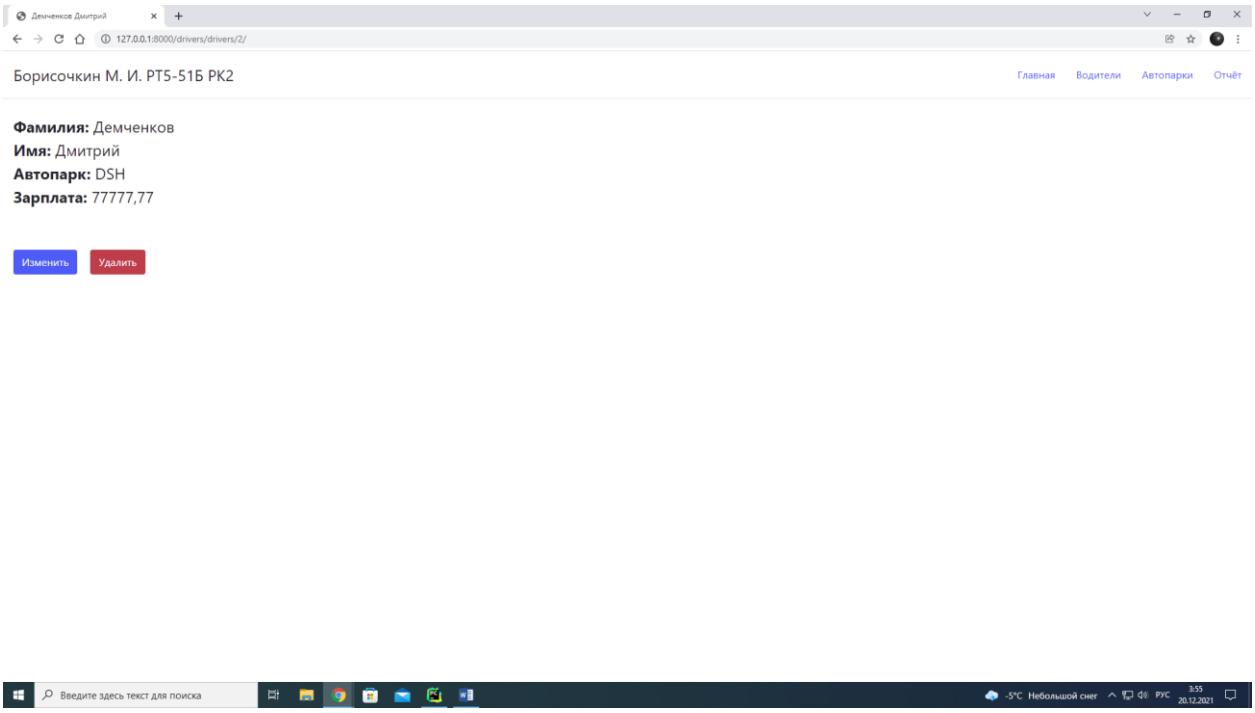

Рисунок 4. Страница детальной информации о водителе

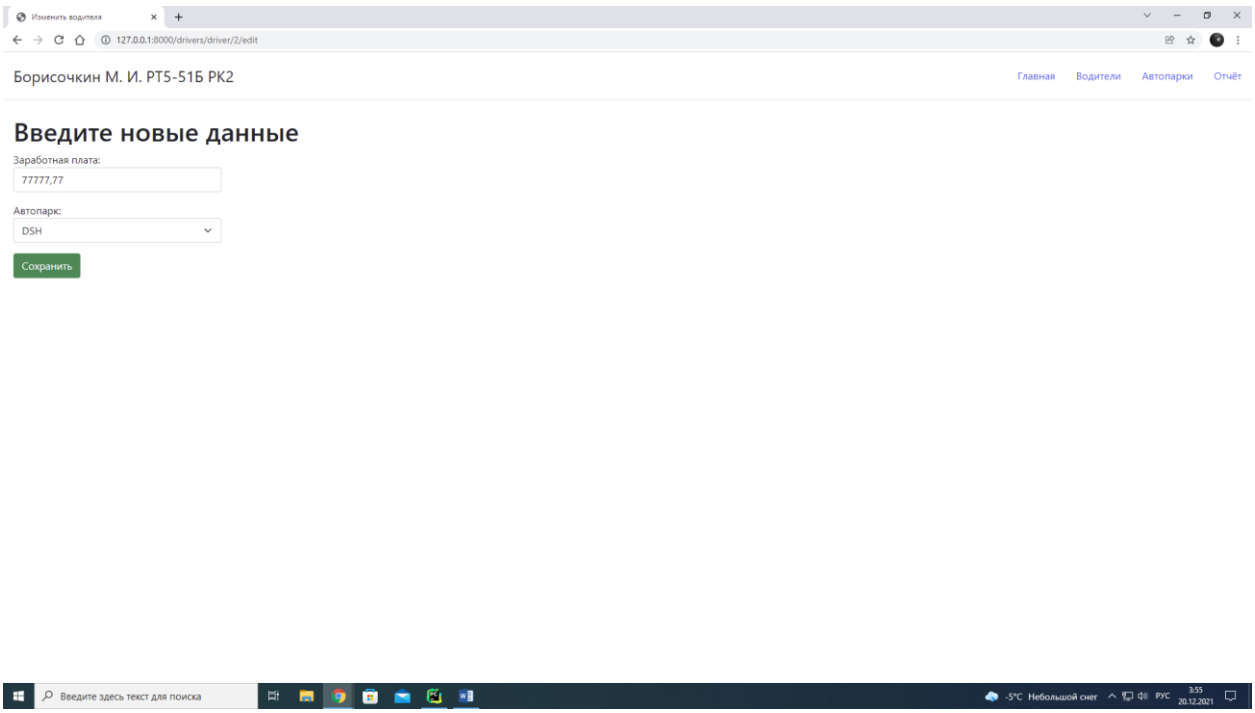

Рисунок 5. Форма изменения данных водителя

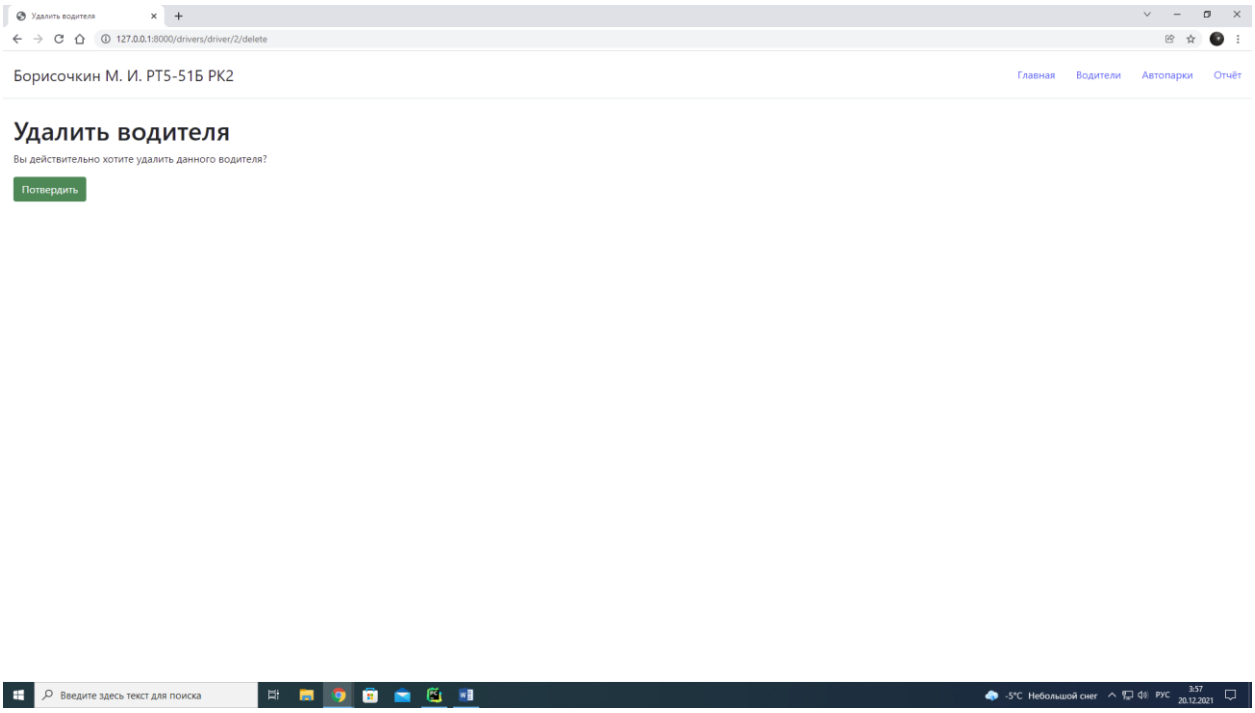

Рисунок 6. Форма удаления данных о водителе

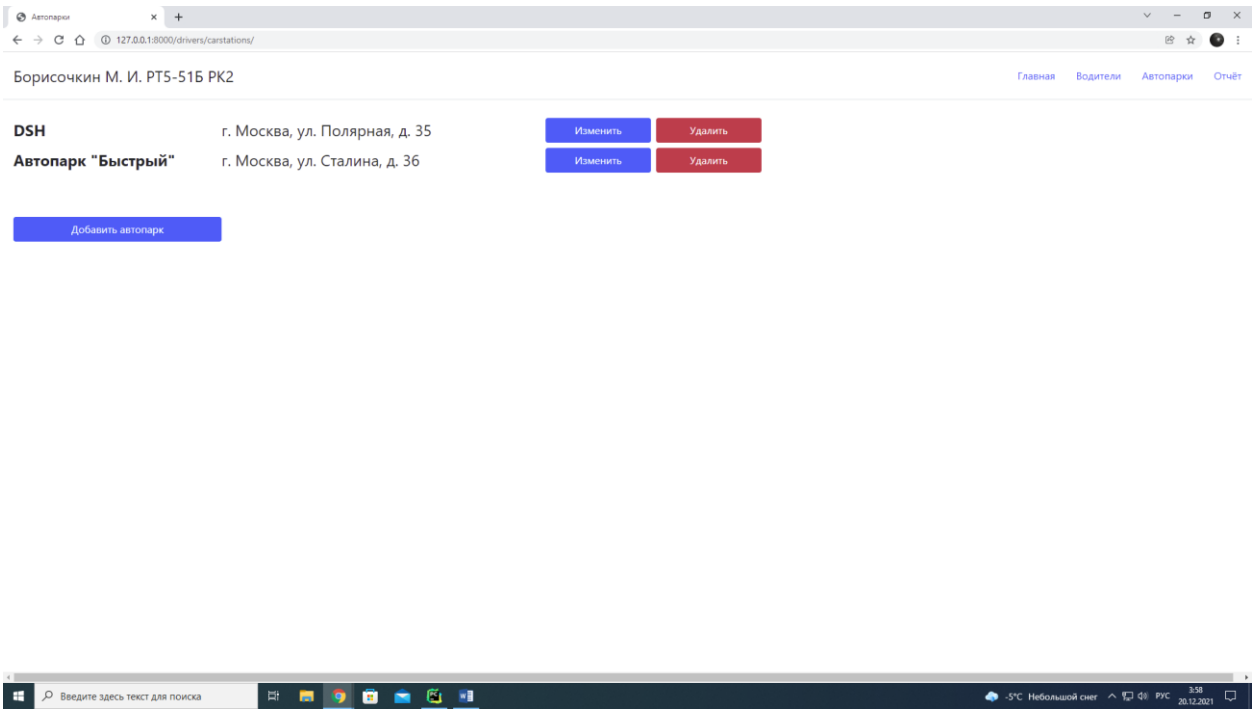

Рисунок 7. Список автопарков

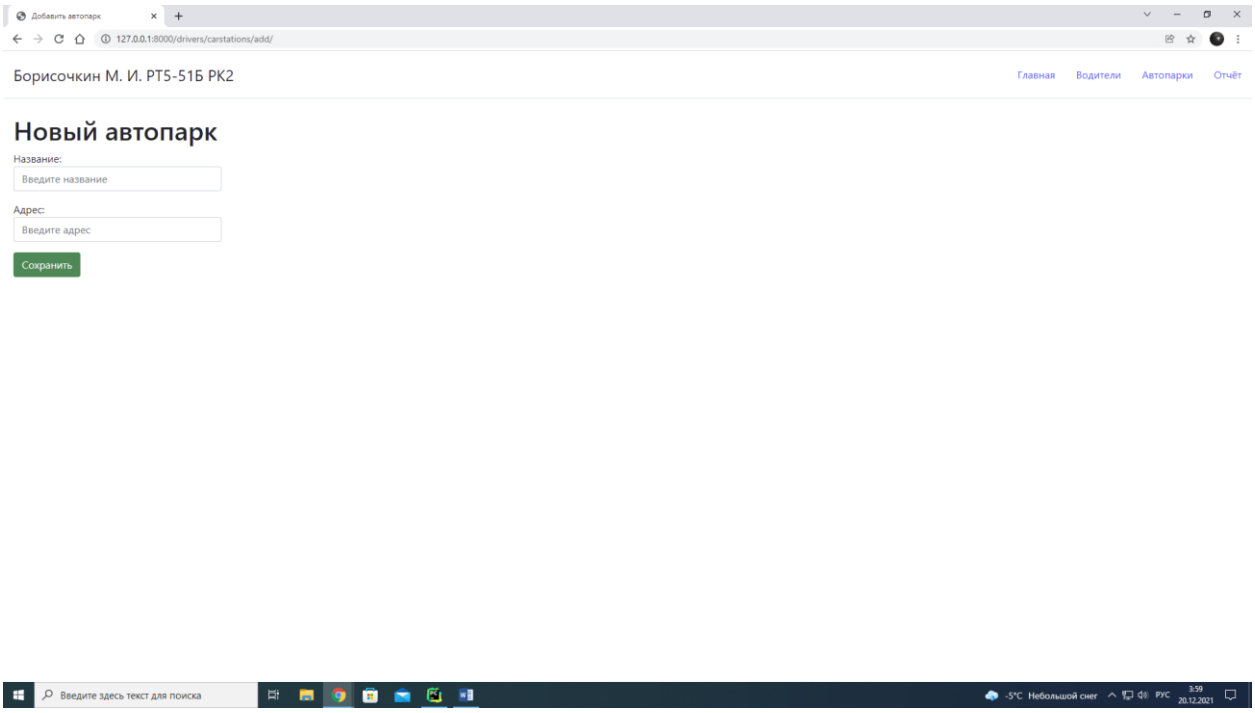

Рисунок 8. Форма добавления автопарка

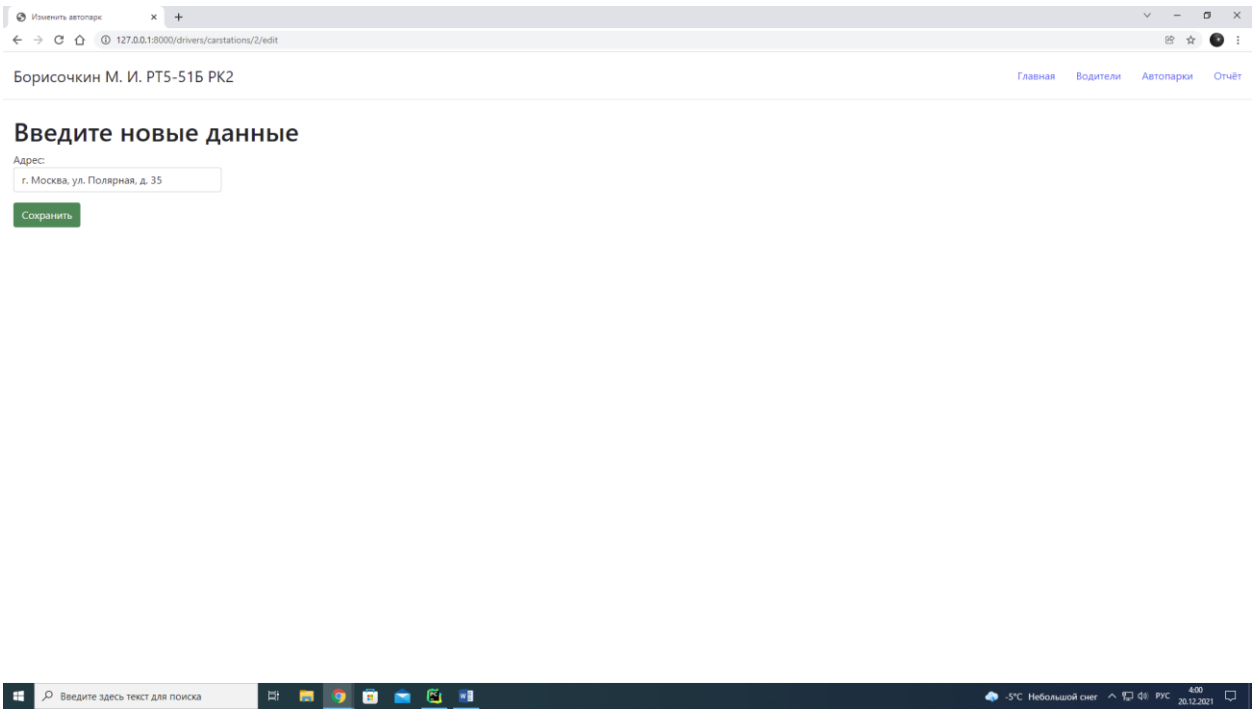

Рисунок 9. Форма изменения данных об автопарке

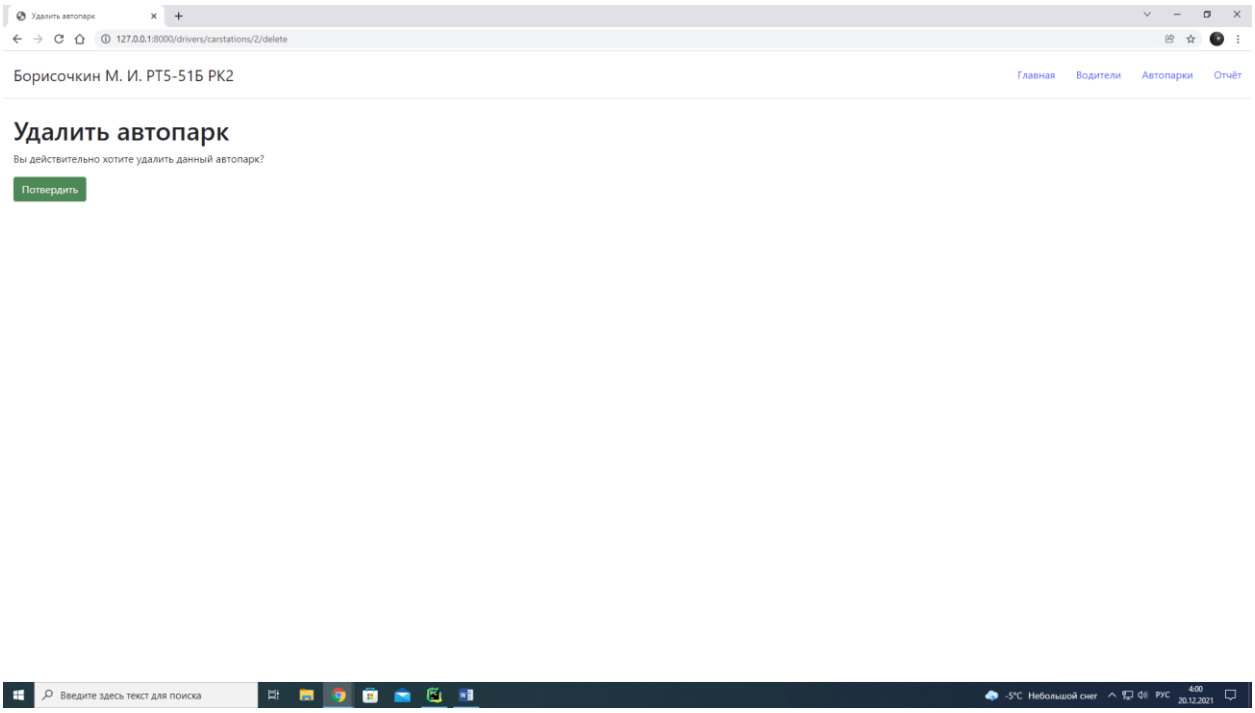

Рисунок 10. Форма удаления данных об автопарке

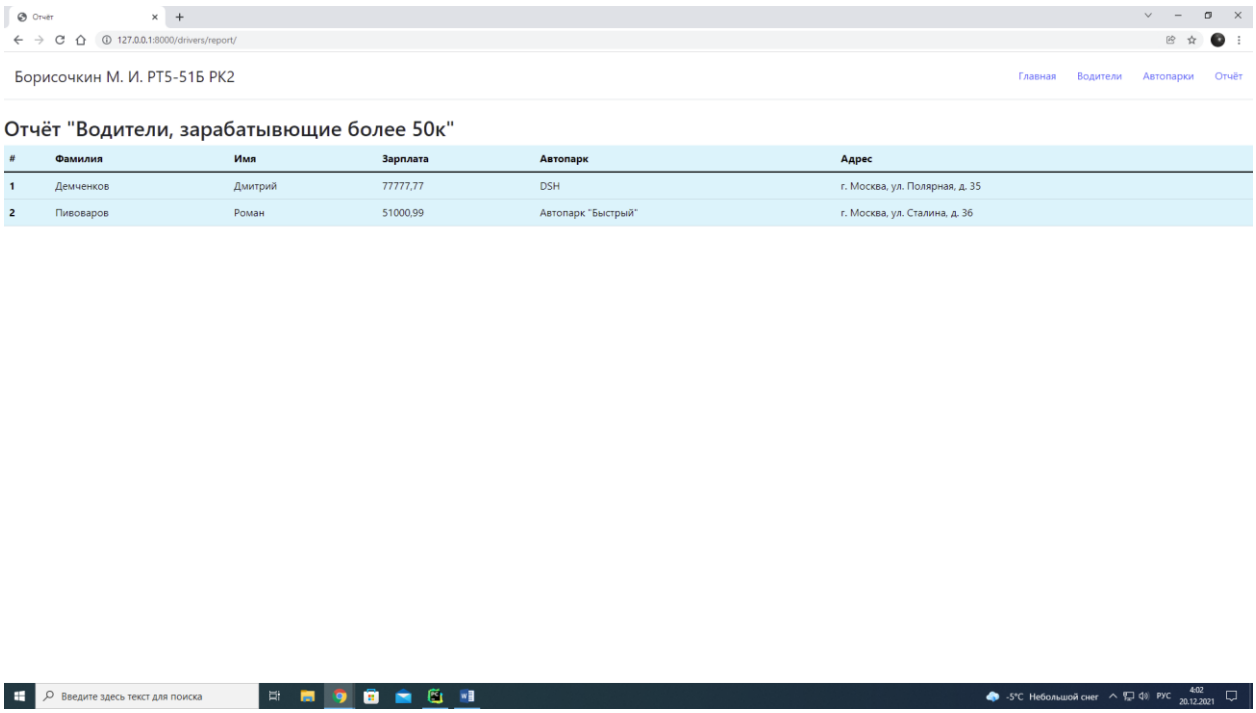

Рисунок 11. Страница с отчётом