

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»**

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Дисциплина: Основы web-программирования

**Отчет
по лабораторной работе №1
«Реализация web-сервисов средствами Python»**

Вариант 6 (12)

Выполнила:
Назаренко Ульяна Кирилловна,
студентка группы К3343

Преподаватель:
Говоров Антон Игоревич

Санкт-Петербург,
2020 г.

Лабораторная работа №1

«Реализация web-сервисов средствами Python»

Цель работы: овладеть практическими навыками и умениями реализации web-сервисов средствами Django 2.2.

Задачи:

1. Реализовать сайт используя фреймворк Django 2.2 для соответствующего варианта практического задания.
2. Построить СУБД PostgreSQL *, в соответствии с практическим заданием.

Программное обеспечение: Python 3.7, Django 2.2, MySQL Workbench.

Практическое задание (Вариант 6 (12)): «Табло победителей автогонок»

Табло должно отображать информацию об участниках автогонок: ФИО участника, название команды, описание автомобиля, описание участника, опыт, класс участника.

Необходимо реализовать регистрацию новых пользователей, с целью дальнейшего добавления ими комментариев к турам. При добавлении комментариев должны сохраняться информация об комментаторе, тип комментария (вопрос о сотрудничестве, вопрос о гонках, иное).

Выполнение работы:

1. Модель базы данных «Табло победителей автогонок»

В соответствии с вариантом 6 (12) была разработана модель БД, представленная на Рисунке 1.

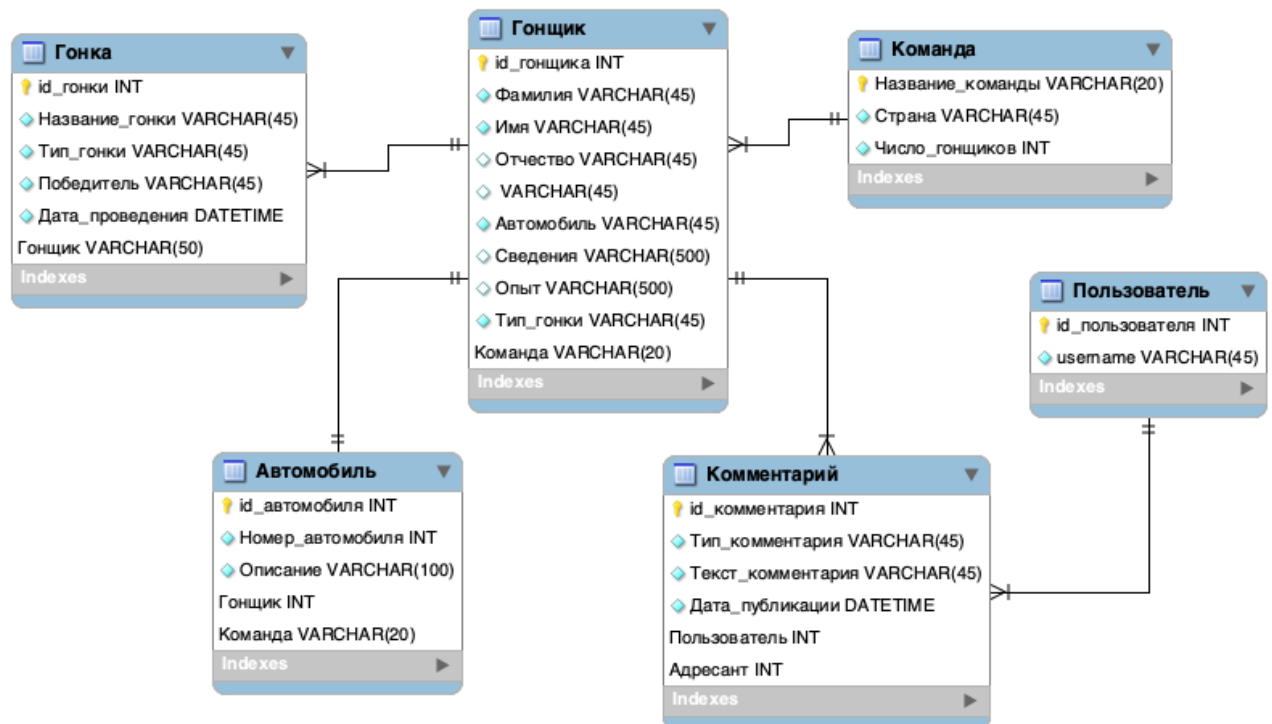


Рисунок 1 – модель базы данных «Табло победителей автогонок»

Основные сущности:

- **Автогонщик (Racer)** – включает ФИО, название команды, класс, описание, опыт и автомобиль;
- **Гонка (Race)** – атрибутами являются название гонки, категория, дата проведения и победитель;
- **Автомобиль (Car)** – автомобиль гонщика, его описание и номер.
- **Team (Команда автогонщика)** – приведены название команды, страна и количество участников.

Сущности для регистрации пользователей:

- **Комментарий (Comment)** – включает тип комментария (вопрос о сотрудничестве, вопрос о гонках, иное), автогонщик, которому адресован комментарий, имя пользователя и дату публикации.
- **Пользователь (User)** – включает логин и пароль.

2. Создание проекта и приложения

Создадим Django проект «*laboratory_work_1*», а также Django приложение «*auto_racing_scoreboard*».

3. Создание моделей

Файл **models.py** хранит следующие модели:

- *Racer* – гонщик (рис.2);
- *Race* – гонка (рис.3);
- *Car* – автомобиль автогонщика (рис.4);
- *Team* – команда автогонщика (рис.4);
- *Comment* – комментарий (рис.5).

Модель *User* взята из стандартной библиотеки «`django.contrib.auth.models`».

```

class Racer(models.Model):

    class Meta:
        db_table = 'Racer'

    CLASSES = (
        ('Class 1', 'Class 1'),
        ('Class 2', 'Class 2'),
        ('Class 3', 'Class 3'),
        ('Class 4', 'Class 4'),
        ('Class 5', 'Class 5'),
    )

    surname = models.CharField(max_length=30)
    name = models.CharField(max_length=30)
    patronymic = models.CharField(max_length=30)
    racer_class = models.CharField(max_length=10, choices=CLASSES)
    description = models.CharField(max_length=500)
    experience = models.CharField(max_length=500)
    team_name = models.ForeignKey(Team, on_delete=models.CASCADE)
    car = models.ForeignKey(Car, on_delete=models.CASCADE)

    def __str__(self):
        return self.name

```

Рисунок 2 – модель «Гонщик»

```

class Race(models.Model):

    class Meta:
        db_table = 'Race'

    CATEGORIES = (
        ('Кольцевая гонка', 'Кольцевая гонка'),
        ('Ралли', 'Ралли'),
        ('Трофи', 'Трофи'),
        ('Гонка на выносливость', 'Гонка на выносливость'),
        ('Автокросс', 'Автокросс'),
        ('Автослалом', 'Автослалом'),
        ('Триал', 'Триал'),
        ('Дрэг-рейсинг', 'Дрэг-рейсинг'),
        ('Спринт', 'Спринт'),
        ('Дрифт', 'Дрифт'),
        ('Картинг', 'Картинг'),
        ('Стритрейсинг', 'Стритрейсинг'),
        ('Иное', 'Иное'),
    )

    name = models.CharField(max_length=30)
    category = models.CharField(max_length=21, choices=CATEGORIES)
    date = models.DateField()
    winner = models.ForeignKey(Racer, on_delete=models.CASCADE)

    def __str__(self):
        return self.name

```

Рисунок 3 – модель «Гонка»

```

class Car(models.Model):

    class Meta:
        db_table = 'Car'

    car_number = models.CharField(max_length=6)
    description = models.CharField(max_length=500)

    def __str__(self):
        return self.description

class Team(models.Model):

    class Meta:
        db_table = 'Team'

    name = models.CharField(max_length=30, primary_key=True)
    country = models.CharField(max_length=30)
    number_of_racers = models.IntegerField()

    def __str__(self):
        return self.name

```

Рисунок 4 – модели «Автомобиль» и «Команда»

```

class Comment(models.Model):

    class Meta:
        db_table = 'Comment'

    TYPES = (
        ('Сотрудничество', 'Сотрудничество'),
        ('Гонки', 'Гонки'),
        ('Иное', 'Иное'),
    )

    comment_type = models.CharField(max_length=20, choices=TYPES)
    text = models.CharField(max_length=1000)
    date = models.DateField(auto_now_add=True)
    racer = models.ForeignKey(Racer, on_delete=models.CASCADE)
    user = models.ForeignKey(User, on_delete=models.CASCADE, null=True)

    def __str__(self):
        return str(self.user)

```

Рисунок 5 – модель «Комментарий»

4. Создание админ панели для разработанной модели данных

```
from django.contrib import admin
from auto_racing_scoreboard.models import Car, Team, Racer, Race, Comment

admin.site.register(Car)
admin.site.register(Team)
admin.site.register(Racer)
admin.site.register(Race)
admin.site.register(Comment)
```

Рисунок 6 – содержимое файла «admin.py»

5. Создание форм

Созданные формы хранятся в файле **forms.py**:

- *Registration* – форма для регистрации нового пользователя (*User*);
- *AddCommentForm* – форма для добавления комментария.

```
class Registration(UserCreationForm):
    class Meta:
        model = User
        fields = ('username', 'surname', 'name', 'patronymic', 'email', 'password1', 'password2')

    username = forms.CharField(required=True, label='Придумайте логин')
    surname = forms.CharField(required=True, label='Введите фамилию')
    name = forms.CharField(required=True, label='Введите имя')
    patronymic = forms.CharField(required=True, label='Введите отчество')
    email = forms.CharField(required=True, label='Введите e-mail')
    password1 = forms.CharField(required=True, label='Введите пароль', widget=forms.PasswordInput)
    password2 = forms.CharField(required=True, label='Повторите пароль', widget=forms.PasswordInput)
```

Рисунок 7 – форма для добавления комментария

```

from django import forms
from django.forms import Textarea
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User
from auto_racing_scoreboard.models import Comment

class AddCommentForm(forms.ModelForm):
    class Meta:
        model = Comment
        fields = ('racer', 'comment_type', 'text')
        labels = {
            'racer': ('Выберите, кому адресован ваш комментарий'),
            'comment_type': ('Выберите тип комментария'),
            'text': ('Напишите комментарий'),
        }

```

Рисунок 8 – форма для добавления комментария

6. Создание контроллеров для обработки данных

Представления размещены в файле **views.py**:

- *main* – вывод главной страницы;
- *scoreboard* – вывод страницы с победителями автогонок и соответствующей информацией;
- *comments* – вывод страницы с комментариями;
- *registration* – вывод страницы регистрации, после успешной регистрации направляет на главную страницу;
- *LogoutFormView* – при выходе из аккаунта направляет пользователя на главную страницу (для отображения входа был задействован модуль авторизации `django.contrib.auth.views.LoginView`.)


```

def main(request):
    return render(request, 'main_page.html')

def scoreboard(request):
    races = Race.objects.all()
    winners = [race.winner for race in races]
    racers = Racer.objects.filter(name__in=winners)
    cars_names = [racer.car for racer in racers]
    cars = Car.objects.filter(description__in=cars_names)
    return render(request, 'scoreboard.html', {'info': zip(races, racers, cars)})

def comments(request):
    comments = {}
    cm = Comment.objects.all()
    comments['comments'] = cm

    form = AddCommentForm(request.POST or None)
    if form.is_valid():
        form = form.save()
        form.user = request.user
        form.save()
        return redirect('comments')
    comments['form'] = form

    return render(request, 'comments.html', comments)

```

Рисунок 9 – формы вывода главной страницы, страницы-табло и страницы с комментариями

```

def reg(request):
    form = Registration(request.POST)
    if form.is_valid():
        form.save()
        username = form.cleaned_data['username']
        password = form.cleaned_data['password1']
        user = authenticate(username=username, password=password)
        login(request, user)
        return redirect('/')
    return render(request, 'registration/registration.html', {'form': form})

class LogoutFormView(FormView):
    def get(self, request):
        logout(request)
        return redirect('/')

```

Рисунок 10 – формы вывода регистрации нового пользователя и выход из сервиса

Папка **templates** хранит шаблоны, которые отвечают за формирование внешнего вида приложения. Они предоставляют специальный синтаксис, который позволяет внедрять данные в код HTML.

Отмечу, что в данной лабораторной работе для удобства используется популярный фреймворк **Bootstrap4**, а также **crispy-forms**, предоставляющие простое отображение форм **Django** без написания лишнего кода и с минимальными установками, что очень упрощает работу.

Содержание **templates**:

- *main_page* – главная страница;
- *menu* – главное меню сайта;
- *comments* – страница с комментариями к турам;
- *scoreboard* – страница с победителями гонок и соответствующая информация о них;

Там же в папке **registration**:

- *registration* – регистрация пользователя;
- *login* – страница входа в аккаунт.

Страницы содержат отображения полей, переданных из контроллера, то есть на данные файлы ссылаются представления из файла **views.py**.

7. Адресация

Файл **urls.py** содержит пути для доступа к страницам.

```
from django.urls import path
from auto_racing_scoreboard import views
from django.contrib.auth.views import LoginView

urlpatterns = [
    path('', views.main, name='main'),
    path('scoreboard/', views.scoreboard, name='scoreboard'),
    path('comments/', views.comments, name='comments'),
    path('register/', views.reg, name='register'),
    path('login/', LoginView.as_view(), name='login'),
    path('logout/', views.LogoutFormView.as_view(), name='logout'),
]
```

Рисунок 11 – пути к страницам в файле «urls.py»

8. Полученные интерфейсы

1) Главная страница



Рисунок 12– Главная страница сервиса «Табло победителей автогонок»

2) Страница с информацией о победителях автогонок

На данной странице отображены гонки, победители в этих гоках и соответствующая информация о них.

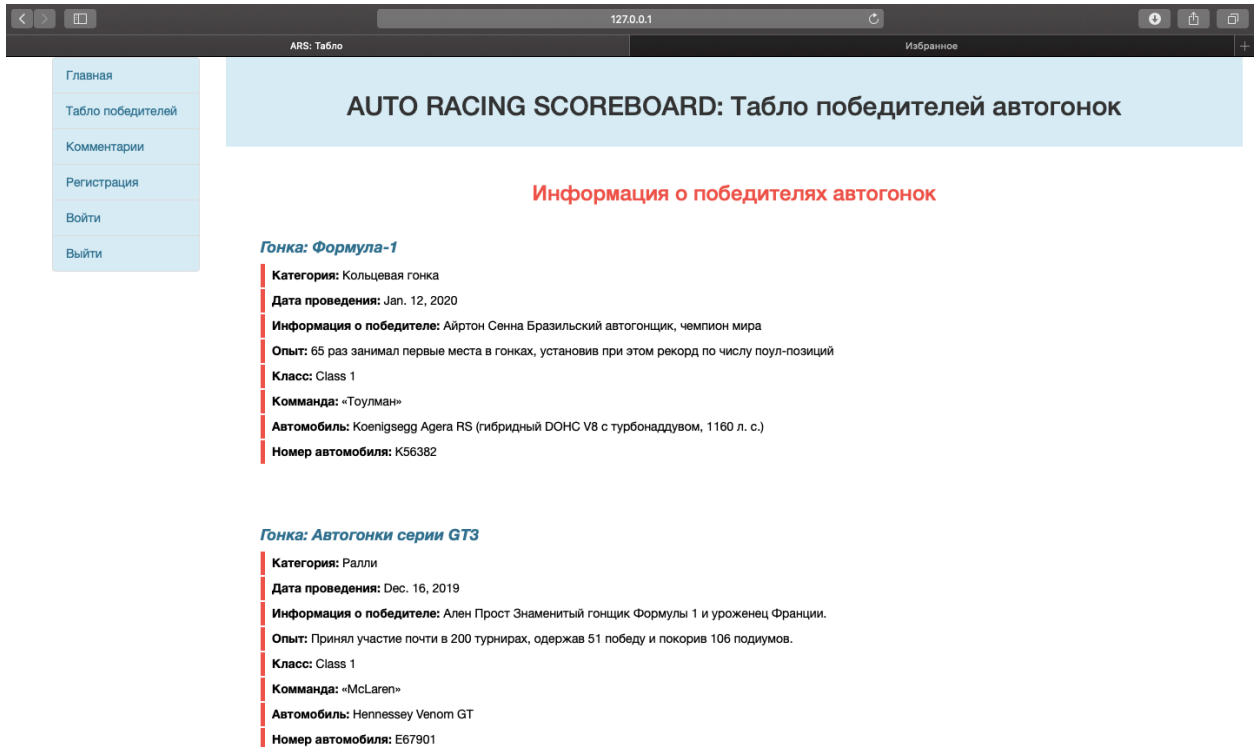


Рисунок 13 – страница сервиса «Табло победителей автогонок» с информацией о победителях автогонок

3) Страница с комментариями

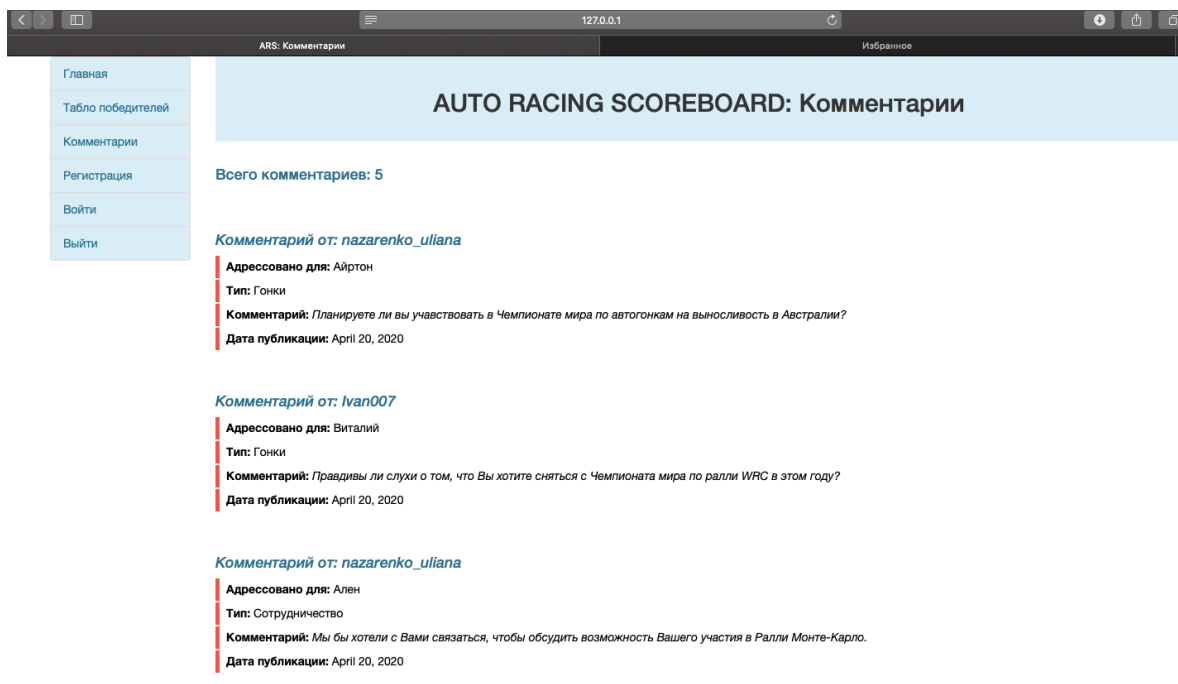


Рисунок 14 – страница сервиса «Табло победителей автогонок» с комментариями

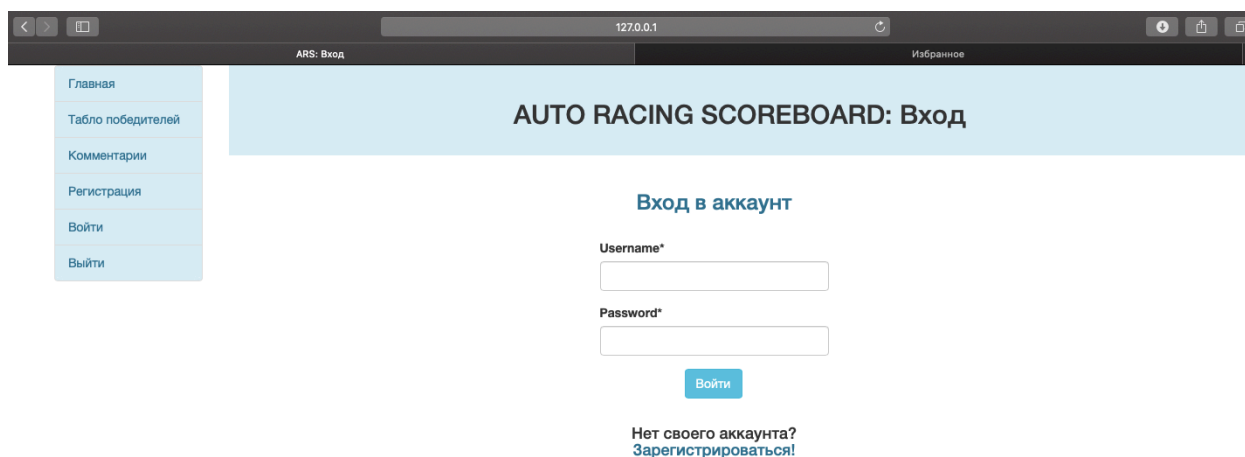
4) Страница регистрации нового пользователя.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1'. The page title is 'ARS: Регистрация'. On the left, there is a navigation menu with links: Главная, Табло победителей, Комментарии, Регистрация, Войти, and Выйти. The main content area has a light blue header with the text 'AUTO RACING SCOREBOARD: Регистрация'. Below the header, it says 'Регистрация нового пользователя'. The registration form consists of several input fields, each preceded by a red vertical bar and a required field indicator. The fields are: 'Придумайте логин:' (value: Ivan007), 'Введите фамилию:' (value: Иванов), 'Введите имя:' (value: Иван), 'Введите отчество:' (value: Иванович), 'Введите e-mail:' (value: Ivanov007@mail.ru), 'Введите пароль:', and 'Повторите пароль:'. Each field has a red vertical bar on the left. Below the fields is a blue button labeled 'Зарегистрироваться'.

Рисунок 15 – страница регистрации аккаунта сервиса «Табло победителей автогонок»

5) Страница входа в аккаунт

После входа в аккаунт, пользователь будет перенаправлен на главную страницу.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1'. The page title is 'ARS: Вход' and the page content is titled 'AUTO RACING SCOREBOARD: Вход'. On the left, there is a vertical navigation menu with links: Главная, Табло победителей, Комментарии, Регистрация, Войти, and Выйти. The main content area has a light blue header with the title 'AUTO RACING SCOREBOARD: Вход'. Below this, the section is titled 'Вход в аккаунт'. It contains two input fields: 'Username*' and 'Password*'. Below the password field is a blue button labeled 'Войти'. At the bottom of the section, there is a link that says 'Нет своего аккаунта? Зарегистрироваться!'.

Рисунок 16 – страница входа в аккаунт сервиса «Табло победителей автогонок»

6) Страница комментирования

Комментировать могут только зарегистрированные пользователи, поэтому после регистрации, авторизованные пользователи могут оставлять свои комментарии в той форме, как показано на рис. 18.

Оставь свой комментарий!
Авторизуйся, чтобы написать комментарий.

Рисунок 17 – Надпись для незарегистрированных пользователей на сайте «Табло победителей автогонок»

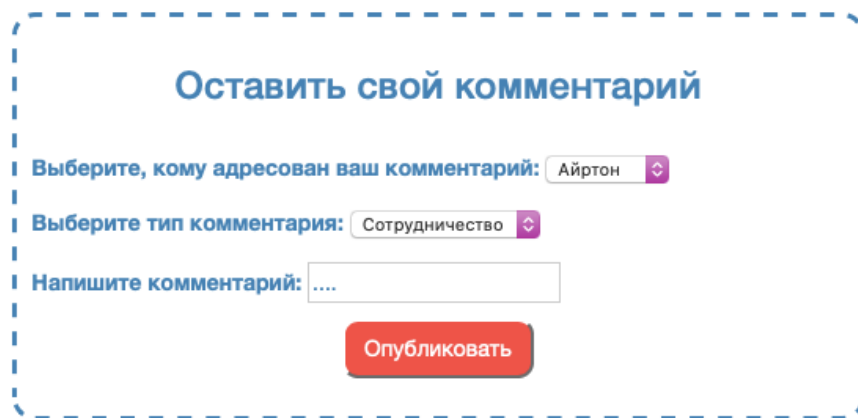


Рисунок 18 – страница добавления комментария на сайте «Табло победителей автогонок»

9. Описание работы с веб-сервисом «Табло победителей автогонок»

Данным сервисом могут пользоваться как зарегистрированные пользователи, так и люди без созданных аккаунтов. Все пользователи могут просматривать табло победителей автогонок и соответствующую информацию о них, а также видеть комментарии, оставленные авторизованными пользователями о турах. Пользователи без аккаунта могут легко зарегистрироваться. После регистрации пользователю предоставляется возможность писать комментарии, адресованные гонщикам.

Выводы:

В ходе данной лабораторной работы были получены практические навыки и умения реализации web-сервисов средствами Django 2.2. Был реализован сайт, по средствам использования фреймворка Django 2.2 для варианта 6 («Табло победителей автогонок») практического задания, была построена модель базы данных «Табло победителей автогонок» в MySQLWorkbench, а также был использован Bootstrap4 для удобства и оптимизации работы над созданием сайта.