

Государственное образовательное учреждение высшего профессионального образования
Санкт-Петербургский национальный исследовательский университет
Информационных Технологий, Механики и Оптики
Факультет инфокоммуникационных технологий

ОТЧЕТ

По Лабораторной работе №1

По Основам Web-программирования

На тему: «Реализация Web-сервисов средствами Python»

Проверил: Говоров А. И.

Дата: «__» _____ 20__ г.

Оценка: _____

Работу выполнила:

Шайдуллина Р. Р.

Студентка группы К3342

Дневного отделения

Санкт-Петербург

2020 г.

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

Оборудование: компьютерный класс.

Программное обеспечение: Python 3.6, Django 2.2, PostgreSQL *.

Практическое задание: реализовать веб-приложение на Django в соответствии с практическим заданием.

Вариант 6. 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).

Ход работы:

1. Разработка модели базы данных.

Модель представлена на рисунке 1.

Основные сущности:

- Racer (автогонщик). Известны их ФИО, команда, автомобиль, описание, опыт и класс.
- Race (гонка). Атрибуты включают название, категорию, дату проведения и победителя.
- Car (автомобиль участвующего в автогонке) с описанием и номером.
- Team (команда, которую представляет автогонщик). Приведены название, страна и количество участников.

Сущности для регистрации пользователей:

- Comment (комментарий). Включает тип комментария, содержание, дату публикации, упоминаемого автогонщика и имя пользователя.
- User (пользователь), его ник и пароль.

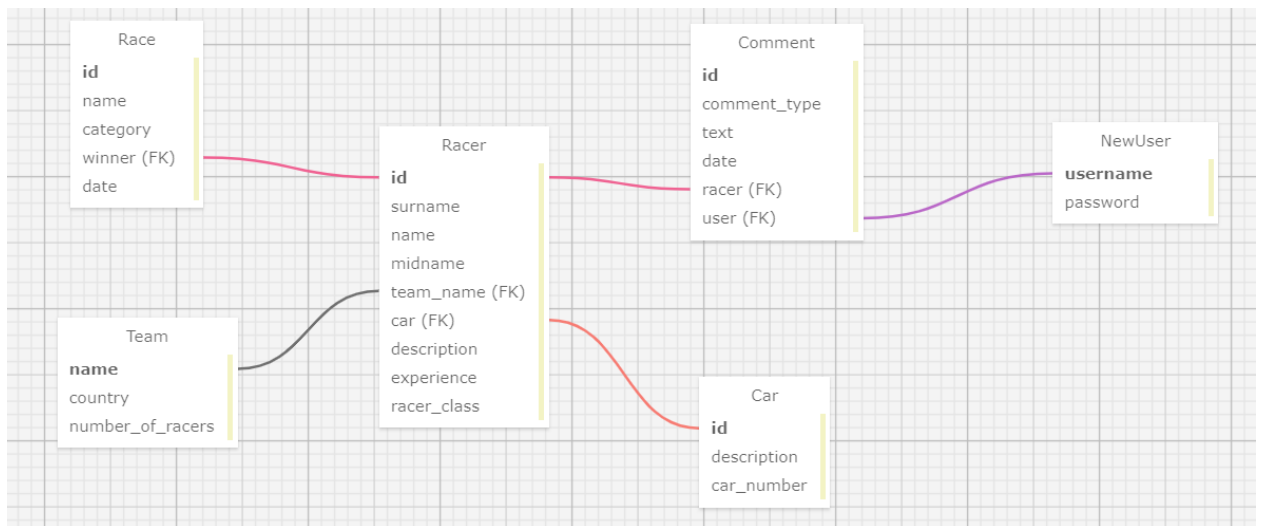


Рисунок 1. Модель базы данных

2. Создание моделей.

Реализация сущностей разработанной модели в файле models.py данных представлена на рисунке 2. Созданы все таблицы из п.1 за исключением User, поскольку была задействована встроенная модель `django.contrib.auth.models.User`.

```

class Car(models.Model):
    class Meta:
        db_table = 'Car'

    car_number = models.CharField(max_length=6)
    description = models.CharField(max_length=100)

    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
  
```

```

class Racer(models.Model):
    class Meta:
        db_table = 'Racer'

    CLASSES = (
        ('A', 'A'),
        ('B', 'B'),
        ('C', 'C'),
        ('D', 'D')
    )

    surname = models.CharField(max_length=30)
    name = models.CharField(max_length=30)
    middlename = models.CharField(max_length=30)
    description = models.CharField(max_length=100)
    experience = models.CharField(max_length=30)
    racer_class = models.CharField(max_length=1, choices=CLASSES)
    team_name = models.ForeignKey(Team, on_delete=models.CASCADE)
    car = models.ForeignKey(Car, on_delete=models.CASCADE)

    def __str__(self):
        #return self.name + ' ' + self.surname + ' ' + self.middlename
        return self.name
  
```

```

class Race(models.Model):

    class Meta:
        db_table = 'Race'

    CATEGORIES = (
        ('OW', 'Open-wheel racing'),
        ('TC', 'Touring car racing'),
        ('SpC', 'Sports car racing'),
        ('PC', 'Production-car racing'),
        ('OM', 'One-make racing'),
        ('TAS', 'Time Attack Series'),
        ('StC', 'Stock car racing'),
        ('R', 'Rallying'),
        ('D', 'Drag racing'),
        ('OR', 'Off-road racing'),
        ('K', 'Kart racing'),
        ('H', 'Historical racing'),
        ('Other', 'Other')
    )

    name = models.CharField(max_length=30)
    category = models.CharField(max_length=5, choices=CATEGORIES)
    date = models.DateField()
    winner = models.ForeignKey(Racer, on_delete=models.CASCADE)

    def __str__(self):
        return self.name


class Comment(models.Model):

    class Meta:
        db_table = 'Comment'

    TYPES = (
        ('Collaboration', 'Collaboration'),
        ('Races', 'Races'),
        ('Other', 'Other')
    )

    comment_type = models.CharField(max_length=20, choices=TYPES)
    text = models.CharField(max_length=100)
    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 2. Таблицы базы данных

3. Создание форм ввода информации.

Формы создаются для ввода новых комментариев пользователями и регистрации новых пользователей (рисунок 3).

```

class AddCommentForm(forms.ModelForm):

    class Meta:
        model = Comment
        fields = ('racer', 'comment_type', 'text')
        labels = {
            'racer': ('Identify who your comment is addressed to'),
            'comment_type': ('Choose the type of your comment'),
            'text': ('Enter your comment'),
        }

class Registration(UserCreationForm):

    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'email', 'password1', 'password2')

    username = forms.CharField(required=True, label='Enter username')
    first_name = forms.CharField(required=True, label='Enter name')
    last_name = forms.CharField(required=True, label='Enter surname')
    email = forms.CharField(required=True, label='Enter e-mail')
    password1 = forms.CharField(required=True, label='Enter password', widget=forms.PasswordInput)
    password2 = forms.CharField(required=True, label='Repeat password', widget=forms.PasswordInput)

```

Рисунок 3. Формы

4. Создание отображений.

Отображения необходимы для вывода информации. В рамках текущей задачи были созданы отображения для показа главной страницы, табло победителей автогонок, а также наглядное представление форм комментариев, регистрации и выхода из аккаунта (рисунок 4). Было опущено создание отображения для входа в аккаунт, поскольку задействован модуль авторизации `django.contrib.auth.views.LoginView`.

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

def leaderboard_view(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, 'leaderboard_view.html', {'info': zip(races, racers, cars)})

def comments(request):
    # all comments
    comments = {}
    cm = Comment.objects.all()
    comments['comments'] = cm
    #racer_names = [c.racer for c in cm]
    #racers = Racer.objects.filter(name__in=racer_names)

    # form
    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) #{'info': zip(comments, racers)}}

def reg(request):
    form = RegistrationForm(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 4. Отображения

5. Полученные интерфейсы.

На всех страницах присутствует навигационное меню. Содержащее следующие пункты:

5.1. Стартовая страница. Выглядит одинаково как для зарегистрировавшегося и авторизовавшегося пользователя, так и для гостя:

- [Main page](#)
- [Racing Leaderboard](#)
- [Comments](#)
- [Register](#)
- [Log in](#)
- [Log out](#)

Board of Auto Race Winners

Choose an option in the navigation menu to continue.

Рисунок 5. Main page

5.2. Табло победителей автогонок.

Аналогично п. 5.1:

- [Main page](#)
- [Racing Leaderboard](#)
- [Comments](#)
- [Register](#)
- [Log in](#)
- [Log out](#)

Leaderboard

- **Race:** Virgin Australia Supercars
 - Category: TC
 - Date: April 18, 2020
- **Winner:** Santino Ferrucci Marco. American racing driver, born May 31, 1998 in Woodbury, Connecticut.
- **Experience:** 2 years
- **Class:** A
- **Team:** Dale Coyne Racing
- **Car:** Aston Martin Vantage GTE (2018)
- **Car number:** q123we
- **Race:** Formula Renault Championship
 - Category: OW
 - Date: March 10, 2020
- **Winner:** Alex Palou Montalbo. Spanish racing driver born in Sant Antoni de Vilamajor, Spain.

Рисунок 6. Racing Leaderboard

5.3. Просмотр всех комментариев.

Аналогично п. 5.1 и 5.2, все существующие комментарии видны и гостям, и авторизованным пользователям.

- [Main page](#)
- [Racing Leaderboard](#)
- [Comments](#)
- [Register](#)
- [Log in](#)
- [Log out](#)

Comments board

Total 7 comments

- **Addressed to:** Daniel
- **Type:** Races
- **Text:** What are the upcoming races?
- **Date:** April 18, 2020
- **User:** regina

- **Addressed to:** Rick
- **Type:** Collaboration
- **Text:** How can I communicate with the racer?
- **Date:** April 18, 2020
- **User:** regina

Рисунок 7. Comments list

5.4. Добавление комментария.

Возможно два варианта: гость видит сообщение о необходимости войти в аккаунт, чтобы внести комментарий. Авторизованный пользователь может добавить комментарий, указав, кому он адресован, какого типа сообщение и сам текст.

Add comment

[Log in](#) to leave a comment.

Add comment

Identify who your comment is addressed to:

Choose the type of your comment:

Enter your comment:

Рисунок 8. Add comment, два варианта

5.5. Регистрация новых пользователей.

Аналогично либо регистрация уже выполнена, либо еще нет. Все поля обязательны, есть ограничения на адрес почты и пароль (типичный вид почты, длина пароля, схожесть с именем пользователя и т. п.):

Registration

You have already registered.

Registration

- This field is required.

Enter username:

- This field is required.

Enter name:

- This field is required.

Enter surname:

- This field is required.

Enter e-mail:

- This field is required.

Enter password:

- This field is required.

Repeat password:

Рисунок 9. Registration

5.6. Вход.

Log in to your account

If you have not created an account yet, then please [sign up](#) first.

Username:

Password:

Рисунок 10. Log in

5.7. Выход.

При нажатии на кнопку «Log out» в навигационном меню происходит выход из аккаунта и перенаправление на главную страницу.

- [Main page](#)
- [Racing Leaderboard](#)
- [Comments](#)
- [Register](#)
- [Log in](#)
- [Log out](#)

Board of Auto Race Winners

Choose an option in the navigation menu to continue.

Рисунок 11. Log out

Вывод: в ходе выполнения лабораторной работы были получены навыки создания web-сервиса средствами языка Python, в частности, фреймворка Django.