

Федеральное государственное автономное образовательное учреждение
высшего образования
"Санкт-Петербургский национальный исследовательский университет информационных
технологий, механики и оптики"

Отчетный лист
по предмету "Web-программирование"

Лабораторная работа № 2
Реализация простого сайта на django

Выполнила: Кустова Екатерина, группа К33421

Проверил: Говоров А.И.

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

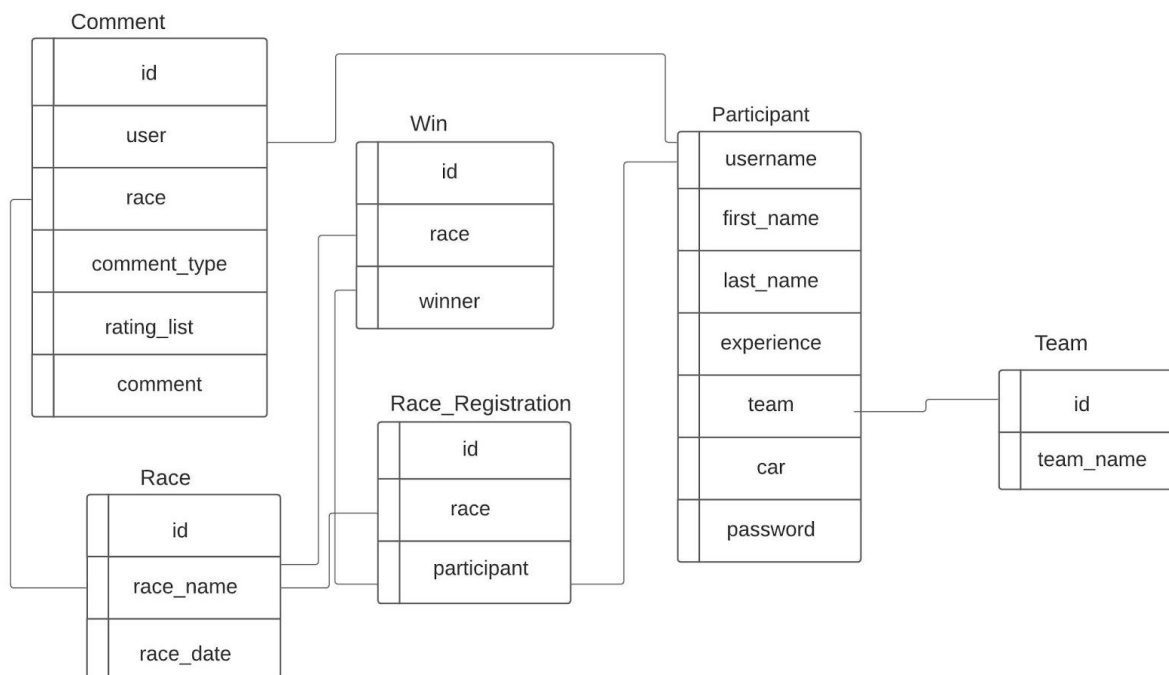
Оборудование: компьютерный класс

Программное обеспечение: Python 3.6+, Django 3, PostgreSQL *

Практическое задание:

Реализовать сайт используя фреймворк Django 3 и СУБД PostgreSQL *, в соответствии с вариантом задания лабораторной работы - Вариант №6 - Табло победителей автогонок.

Описание модели данных:



Реализация сущностей

Участник:

Пользователь сайта является участником. Он может зарегистрироваться, войти в систему, записаться на гонку, отписаться от гонки, оставить о любой гонке комментарий, стать победителем гонки (победитель указывается

администратором). Хранятся данные об имени и фамилии пользователя, его юзернейм, опыт, команда, автомобиль.

```
class Participant(AbstractUser):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    experience = models.CharField(max_length=30)
    team = models.ForeignKey('Team',
on_delete=models.SET_NULL, null=True)
    car = models.CharField(max_length=20)
```

Команда:

По командам участников распределяет администратор. Хранится название команды.

```
class Team(models.Model):
    team_name = models.CharField(max_length=30)
```

Гонка:

Гонки создаются администратором. О гонке хранятся её название и дата заезда.

```
class Race(models.Model):
    race_name = models.CharField(max_length=30)
    race_date = models.DateField()
```

Регистрация на гонку:

Пользователь может зарегистрироваться на гонку. Будут храниться данные о зарегистрированном пользователе и гонке, на которую он зарегистрирован.

```
class Race_Registration(models.Model):
    race = models.ForeignKey('Race', on_delete=models.CASCADE)
    participant = models.ForeignKey('Participant',
on_delete=models.CASCADE)
```

Победа в гонке:

Администратор может указать победителя для каждой гонки.

```
class Win(models.Model):
    race = models.ForeignKey('Race', on_delete=models.CASCADE)
    winner = models.ForeignKey('Race_Registration',
on_delete=models.CASCADE)
```

Комментарий:

Пользователи могут оставлять комментарии - будут храниться имя пользователя, оставившего комментарий, название гонки, о которой он оставляет комментарий, тип комментария, его оценка гонки и сам комментарий.

```
class Comment(models.Model):
    user = models.ForeignKey('Participant',
on_delete=models.SET_NULL, null=True)
    race = models.ForeignKey('Race', on_delete=models.CASCADE)

    class CommentType(models.IntegerChoices):
        partnership = 1
        race_question = 2
        other = 3
    comment_type =
models.IntegerField(choices=CommentType.choices, null=True)
    class Rating(models.IntegerChoices):
        disgusted = 1
        very_unsatisfied = 2
        unsatisfied = 3
        below_neutral = 4
        neutral = 5
        above_neutral = 6
        satisfied = 7
        very_satisfied = 8
        amazed = 9
        best_of_the_best = 10

    rating_list = models.IntegerField(choices=Rating.choices,
null=True)
    comment = models.CharField(max_length=3000)
```

Контроллеры

Профиль (просмотр информации о пользователе):

```
def profile(request):
    context = {}
    context["dataset"] = Participant.objects.all()
    return render(request, 'profile.html')
```

Регистрация пользователя:

```
def registration(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)

        if form.is_valid():
            form.save()
            return redirect('main')

    else:
        form = RegistrationForm()

    context = {'form': form}
    return render(request, 'registration.html', context)
```

Вход пользователя в систему:

```
class LoginFormView(FormView):
    form_class = AuthenticationForm
    template_name = "main.html"
    success_url = "/profile/"

    def form_valid(self, form):
        self.participant = form.get_user()
        login(self.request, self.participant)
        return super(LoginFormView, self).form_valid(form)
```

Регистрация пользователя на гонку:

```
class Registrations(ListView):
    template_name = 'race_registrations.html'
    context_object_name = 'race_registrations.html'
    def get_queryset(self):
```

```
        self.participant = self.request.participant.pk
    return
Race_Registration.objects.filter(participant=self.participant
)
```

Просмотр регистраций пользователя:

```
class Registrations(ListView):
    template_name = 'race_registrations.html'
    context_object_name = 'race_registrations.html'
    def get_queryset(self):
        self.participant = self.request.participant.pk
    return
Race_Registration.objects.filter(participant=self.participant
)
```

Все гонки:

```
def races(request):
    context = {}
    context["dataset"] = Race.objects.all()
    return render(request, "races.html", context)
```

Результаты гонок:

```
def results(request):
    context = {}
    context["dataset"] = Win.objects.all()
    return render(request, 'results.html', context)
```

Форум с комментариями (и возможностью добавить свой):

```
def forum(request):
    context = {}
    context["dataset"] = Comment.objects.all()

    form = CommentForm(request.POST)
    context['form'] = form

    if form.is_valid():
```

```
object = form.save(commit=False)
object.participant = request.participant
object.date = datetime.datetime.now()
object.save()
return redirect('/forum/')
return render(request, "forum.html", context)
```

Формы

Форма регистрации:

```
class RegistrationForm(UserCreationForm):
    username = forms.CharField(max_length=30, required=True,
label="Login")
    first_name = forms.CharField(max_length=30, required=True,
label="First Name")
    last_name = forms.CharField(max_length=30, required=True,
label="Last Name")
    experience = forms.CharField(max_length=30, required=True,
label="Your Experience")
    car = forms.CharField(max_length=30, required=True,
label="Car")

    class Meta:
        model = Participant
        fields = ('username', 'first_name', 'last_name',
'experience', 'car')
```

Форма регистрации на гонку:

```
class RaceRegistrationForm(forms.ModelForm):
    class Meta:
        model = Race_Registration
        fields = ('race', 'participant')

    def __init__(self, *args, **kwargs):
        super(RaceRegistrationForm, self).__init__(*args,
**kwargs)
```

```
self.fields['race'].disabled = True
self.fields['participant'].disabled = True
```

Форма создания комментария:

```
class CommentForm(forms.ModelForm):
    class Meta:
        model = Comment
        fields = ('user', 'comment', 'comment_type',
                  'rating_list', 'race')
        widgets = {
            'comment': forms.Textarea(),
        }

    def __init__(self, *args, **kwargs):
        super(CommentForm, self).__init__(*args, **kwargs)
        self.fields['user'].disabled = True
```

Получившиеся страницы сайта и их интерфейс

Регистрация:

Registration

Login:

First Name:

Last Name:

Your Experience:

Car:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

Вход в систему:

Main page

Log in or create a new account

Log in

Username:

Password:

[Registration](#)

Профиль пользователя:

My profile

Username: Captain

First name: James

Last name: Kirk

Experience: No

Car: Enterprise

[My registrations](#)

[Add race registration](#)

[All races](#)

Запись на гонку:

Sign up for a race

[My profile](#)

[My registrations](#)

Race
Race1

Просмотр записей пользователя с возможностью отменить запись:

My registrations

Race: Race1

Delete

[Back to profile](#)

Просмотр всех существующих гонок:

All races

Race: Race

Date: Jan. 25, 2021

Race: Race1

Date: Jan. 27, 2021

[Results](#)

[Forum](#)

[Main](#)

Форум с возможностью добавить свой комментарий:

Comments

◦ This field is required.

User:

◦ This field is required.

Comment:

◦ This field is required.

Comment type:

◦ This field is required.

Rating list:

◦ This field is required.

Race:

Submit Race

• no rev Race1

Результаты гонок:

Results

Race: Race

Winner: admin

[All races](#)

Вывод:

В ходе выполнения лабораторной работы были получены практические навыки реализации web-сервисов средствами Django. Был реализован сайт - табло автогонок с возможностью зарегистрироваться на гонку и оставить о ней комментарий. Табло фиксирует победителей автогонок.