ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет «Инфокоммуникационных технологий»
Направление подготовки «09.03.03 Мобильные и сетевые технологии»

О Т Ч Е Т

Тема задания: Лабораторная работа №2.
Реализация простого сайта на Django

Выполнил:
Студент Магай Олег (Фамилия И.О.) К33402 номер группы

Проверил:
Преподаватель Говоров А.И (Фамилия И.О)

Санкт-Петербург
2020

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 5

Интерфейс описывает названия конференций, список тематик, место проведения, период проведения, описание конференций, описание место проведения, условия участия.

Необходимо реализовать следующий функционал:

- Регистрация новых пользователей.
- Просмотр конференций и регистрацию авторов для выступлений. Пользователь должен иметь возможность редактирования и удаления своих регистраций.
- Написание отзывов к конференциям. При добавлении комментариев, должны сохраняться даты конференции, текст комментария, рейтинг (1-10), информация о комментаторе.
- Администратор должен иметь возможность указания результатов выступления (рекомендован к публикации или нет) средствами Django-admin.
- В клиентской части должна формироваться таблица, отображающая всех участников по конференциям.

Модели

```python
class Conference(models.Model):
    SCIENCE = "Наука"
    MEDICINE = "Медицина"
    SPORT = "Спорт"
    TOPIC_CHOICES = [
        (SCIENCE, "Наука"),
        (MEDICINE, "Медицина"),
        (SPORT, "Спорт"),
    ]
    name = models.CharField("Конференция", max_length=80)
    topic = models.CharField("Тема", blank=True, choices=TOPIC_CHOICES, max_length=10)
    location = models.CharField("Место проведения", max_length=80)
    start_date = models.DateField("Начало конференции")
    end_date = models.DateField("Окончание конференции")
    description = models.CharField("Описание конференции", max_length=500)
    location_extra = models.CharField("Описание места проведения", max_length=500)
    terms = models.CharField("Условия участия", max_length=300)

    class Meta:
        verbose_name = "Конференция"
        verbose_name_plural = "Конференции"

    def __str__(self):
        return f"{self.name} - {self.topic}"

class Comment(models.Model):
    JOIN = "Принять участие"
    POST = "Опубликовать доклад"
    LOCATION = "Место проведения"
    DIFFERENT = "Другое"
    TOPIC_CHOICES = [
        (JOIN, "Принять участие"),
        (POST, "Опубликовать доклад"),
        (LOCATION, "Место проведения"),
        (DIFFERENT, "Другое"),
    ]
    conference = models.ForeignKey(
        Conference, on_delete=models.CASCADE, verbose_name="Конференция"
    )
    author = models.ForeignKey(
        User, on_delete=models.CASCADE, verbose_name="Автор комментария"
    )
    topic = models.CharField(
        "Тип комментария", blank=True, choices=TOPIC_CHOICES, max_length=50
    )
    text = models.CharField("Текст комментария", max_length=300)

    class Meta:
        verbose_name = "Комментарий"
        verbose_name_plural = "Комментарии"
```

Views

Список конференций:

```
class ConferenceView(generic.ListView):
    model = Conference
    context_object_name = "conferences"
    queryset = Conference.objects.all()
    template_name = "conferences.html"
```

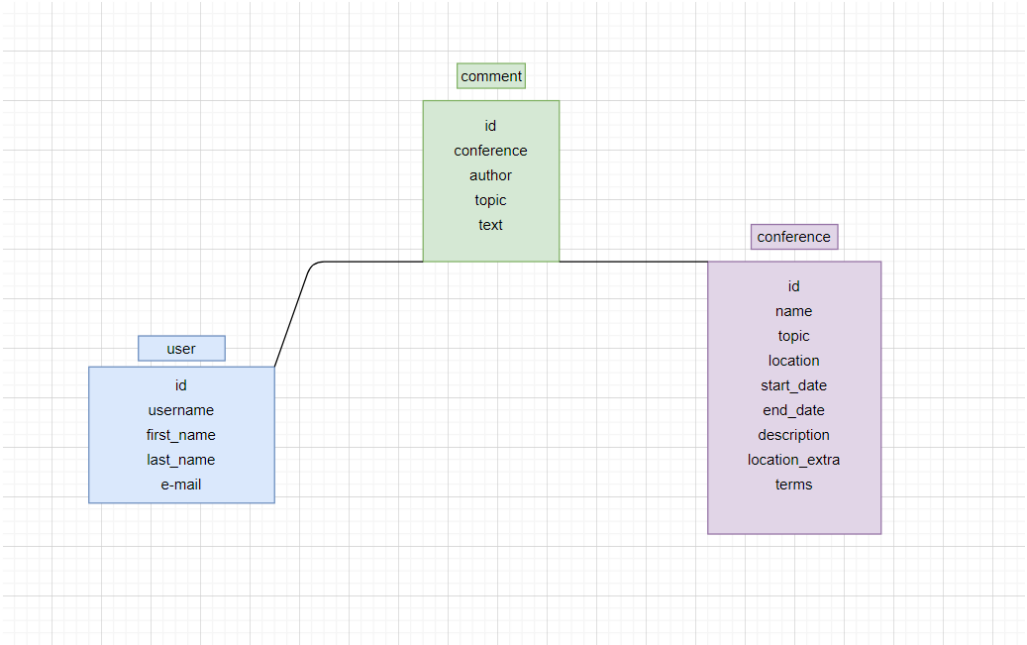
Детальная информация о конференции:

```
class ConferenceDetailView(FormMixin, generic.DetailView):
    model = Conference
    template_name = "conference-detail.html"
    form_class = PostComment

    def get_context_data(self, **kwargs):
        context = super(ConferenceDetailView, self).get_context_data(**kwargs)
        context["form"] = PostComment(
            initial={"conference": self.object, "author": self.request.user}
        )
        context["comments"] = Comment.objects.filter(conference=self.get_object()).all()
        return context

    def post(self, request, *args, **kwargs):
        self.object = self.get_object()
        form = self.get_form()
        if form.is_valid():
            form.save()
        return HttpResponseRedirect(
            reverse("conference-detail", args=(self.object.pk,))
        )
```

Схема базы данных:



Вывод: Мы овладели практическими навыками и умениями реализации web-сервисов средствами Django в ходе реалзиации сайта для научных конференций.