

**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

О Лабораторной работе №2

Тема задания: Реализация простого сайта на Django

Выполнил:

Студент Береснев А.С K33402
(Фамилия И.О.) номер группы

Проверил:

Преподаватель Говоров А.И
(Фамилия И.О)

Санкт-Петербург
2020

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

Вариант 1

Список отелей.

Необходимо учитывать название отеля, адрес, описание, типы номеров, стоимость, вместимость, удобства.

Необходимо реализовать следующий функционал:

Регистрация новых пользователей.

Просмотр и резервирование номеров. Пользователь должен иметь возможность редактирования и удаления своих резервирований.

Написание отзывов к номерам. При добавлении комментариев, должны сохраняться период проживания, текст комментария, рейтинг (1-10), информация о комментаторе.

Администратор должен иметь возможность заселить пользователя в отель и выселить из отеля средствами Django-admin.

В клиентской части должна формироваться таблица, отображающая постояльцев отеля за последний месяц.

Модели

```
class Client(AbstractUser):
    username = models.CharField(
        max_length=30, unique=True)
    password = models.CharField(max_length=30)
    is_superuser = models.BooleanField(default=False)

    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    date_of_birth = models.DateField(null=True)
    passport = models.IntegerField(null=True)
    nationality = models.CharField(max_length=30)

class Hotel(models.Model):
    name = models.CharField(max_length=50)
    rating = models.IntegerField()
    adress = models.CharField(max_length=30)
    stars = models.IntegerField()

class Room(models.Model):
    hotel_id = models.ForeignKey(Hotel, on_delete=models.CASCADE)
    room_number = models.IntegerField()
    price = models.IntegerField()
    rating = models.IntegerField(null=True)
    capacity = models.IntegerField()
```

Сущность клиента – является сущностью юзера

Сущность отеля

Сущность комнаты в отеле

```

class ReservedRooms(models.Model):
    client_id = models.ForeignKey(settings.AUTH_USER_MODEL,
                                  on_delete=models.CASCADE)
    room_id = models.ForeignKey(Room, on_delete=models.CASCADE)
    hotel_id = models.ForeignKey(Hotel, on_delete=models.CASCADE)
    start_date = models.DateField()
    end_date = models.DateField(null=True)

class Review(models.Model):
    TYPES = (
        ('5', 'Excellent'),
        ('4', 'Good'),
        ('3', 'Decent'),
        ('2', 'Bad'),
        ('1', 'Very Bad')
    )
    room_id = models.ForeignKey(Room, on_delete=models.CASCADE)
    client_id = models.ForeignKey(settings.AUTH_USER_MODEL,
                                  on_delete=models.CASCADE)
    rating = models.CharField(max_length=5, choices=TYPES)
    description = models.CharField(max_length=1000)

```

Сущность зарезервированных комнат, many-to many между клиентами и комнатами
 Сущность отзывов

Views

Регистрация и логин

```
def registration(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            raw_password = form.cleaned_data.get('password1')
            user = authenticate(username=username, password=raw_password)
            login(request, user)
            return redirect('/')

    else:
        form = RegistrationForm()
        return render(request, 'registration.html', {'form': form})

def logout_view(request):
    logout(request)
    return redirect('/')
```

Отдельная комната

```
def room_view(request, room_id):
    try:
        room = Room.objects.get(pk=room_id)
        hotel = room.hotel_id
    except Room.DoesNotExist:
        raise Http404('Room does not exist')

    reviews = []
    for review in Review.objects.all():
        if review.room_id == room:
            reviews.append(review)

    return render(request, 'room.html', {'room': room, 'hotel': hotel, 'reviews': reviews})
```

Список отелей и вид отдельного отеля

```
class HotelsListView(ListView):
    model = Hotel
    template_name = 'index.html'

def hotel_view(request, hotel_id):
    try:
        hotel = Hotel.objects.get(pk=hotel_id)
    except Hotel.DoesNotExist:
        raise Http404('Hotel does not exist')

    empty_rooms = []
    reserved_rooms = []
    last_clients = []
    for reservation in ReservedRooms.objects.all():
        if (date.today() - reservation.start_date).days < 30 and reservation.hotel_id == hotel:
            last_clients.append(reservation.client_id)
        if reservation.hotel_id == hotel:
            if reservation.start_date < date.today() and reservation.end_date > date.today():
                reserved_rooms.append(reservation.room_id)
            elif reservation.end_date == 'null':
                reserved_rooms.append(reservation.room_id)

    for room in Room.objects.all():
        if (room.hotel_id == hotel) and (room not in reserved_rooms):
            empty_rooms.append(room)

    return render(request, 'hotel.html', {
        'reserved_rooms': reserved_rooms,
        'empty_rooms': empty_rooms,
        'hotel': hotel,
        'last_clients': last_clients})
```

Резервирование отеля

```
def book_view(request, room_id):
    try:
        room = Room.objects.get(pk=room_id)
    except Room.DoesNotExist:
        raise Http404('Room does not exist')

    if request.method == "POST":
        form = BookForm(request.POST)

        if form.is_valid():
            post = form.save(commit=False)
            for reservation in ReservedRooms.objects.all():
                if room == reservation.room_id:
                    if (reservation.start_date <= post.start_date and reservation.end_date >= post.end_date)
                    or (reservation.end_date <= post.start_date and post.end_date >= reservation.end_date)
                    or (form.start_date <= reservation.start_date and form.end_date >= reservation.start_date):
                        raise Http404("Room already reserved on this time")
            post.hotel_id = room.hotel_id
            post.room_id = room
            post.client_id = request.user
            post.save()
            return redirect('/')
        else:
            print(form.errors)
            print(request.user)
    else:
        form = BookForm()

    return render(request, 'book.html', {'room': room, 'form': form})
```

Отзывы

```
def review_view(request, room_id):
    try:
        room = Room.objects.get(pk=room_id)
    except Room.DoesNotExist:
        raise Http404('Room does not exist')

    if request.method == "POST":
        form = ReviewForm(request.POST)
        if form.is_valid():
            post = form.save(commit=False)
            post.room_id = room
            post.client_id = request.user
            post.save()
            return redirect('/')
        else:
            print(form.errors)
            print(request.user)
    else:
        form = ReviewForm()

    return render(request, 'review.html', {'room': room, 'form': form})
```

URLs

```
urlpatterns = [  
    path('', HotelsListView.as_view()),  
    path('hotels/<int:hotel_id>', hotel_view),  
    path('registration/', registration),  
    path('login/', LoginView.as_view(template_name='login.html')),  
    path('logout/', logout_view),  
    path('room/<int:room_id>', room_view),  
    path('book/<int:room_id>', book_view),  
    path('review/<int:room_id>', review_view),  
]
```

Работа программы

Главная страница

[Войти](#)

[Регистрация](#)

•

[Hilton](#)

Kirov, Oktyabrskiy pr. 123

•

[Vyatka](#)

Kirov, Oktyabrskiy pr. 100

Страница отеля

[Войти](#)

[Регистрация](#)

Hilton

Kirov, Oktyabrskiy pr. 123

Empty Rooms

- [63](#)

Capacity:2

Rating:4

- [25](#)

Capacity:8

Rating:5

Reserved Rooms

- [125](#)

Capacity:4

Last Clients

- Nikonov Ilya
-

- Kudashev Iskander
-

Страница комнаты

[Войти](#)

[Регистрация](#)

Hilton

Kirov, Oktyabrskiy pr. 123

Room number:63

Capacity:2

Price per Day:100

Rating:4

Войдите что бы забронировать или написать отзыв

ОТЗЫВЫ

2 | Kudashev Iskander

Поел тараканов на завтрак

Резервирование

[Выйти](#)

Резервирование

Start date:

End date:

Страница регистрации

Регистрация

First name:

Last name:

Username:

Date of birth:

Passport:

Nationality:

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.

Если вы уже зарегистрированы, то можете [войти!](#)

Страница логина

Вход

Username:

Password:

•

Если вы ещё не зарегистрированы, то можете [зарегистрироваться!](#)

[На главную](#)

Написание отзыва

[Выйти](#)

Отзыв о комнате 63

Rating: ▼

Description:

Вывод: Мы овладели практическими навыками и умениями реализации web-сервисов средствами Django. И реализовали букинг отелей.