

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

Факультет «Инфокоммуникационных технологий»

Направление подготовки «45.03.04 Интеллектуальные системы в гуманитарной сфере»

**ОТЧЕТ**

**Web-программирование**

**Лабораторная работа №2: Реализация сайта средствами Django**

**Выполнила:** Грицай А. И. группа К33421

**Проверил:** Говоров А. И.

**Санкт-Петербург  
2020**

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

**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 1: Список отелей.**

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

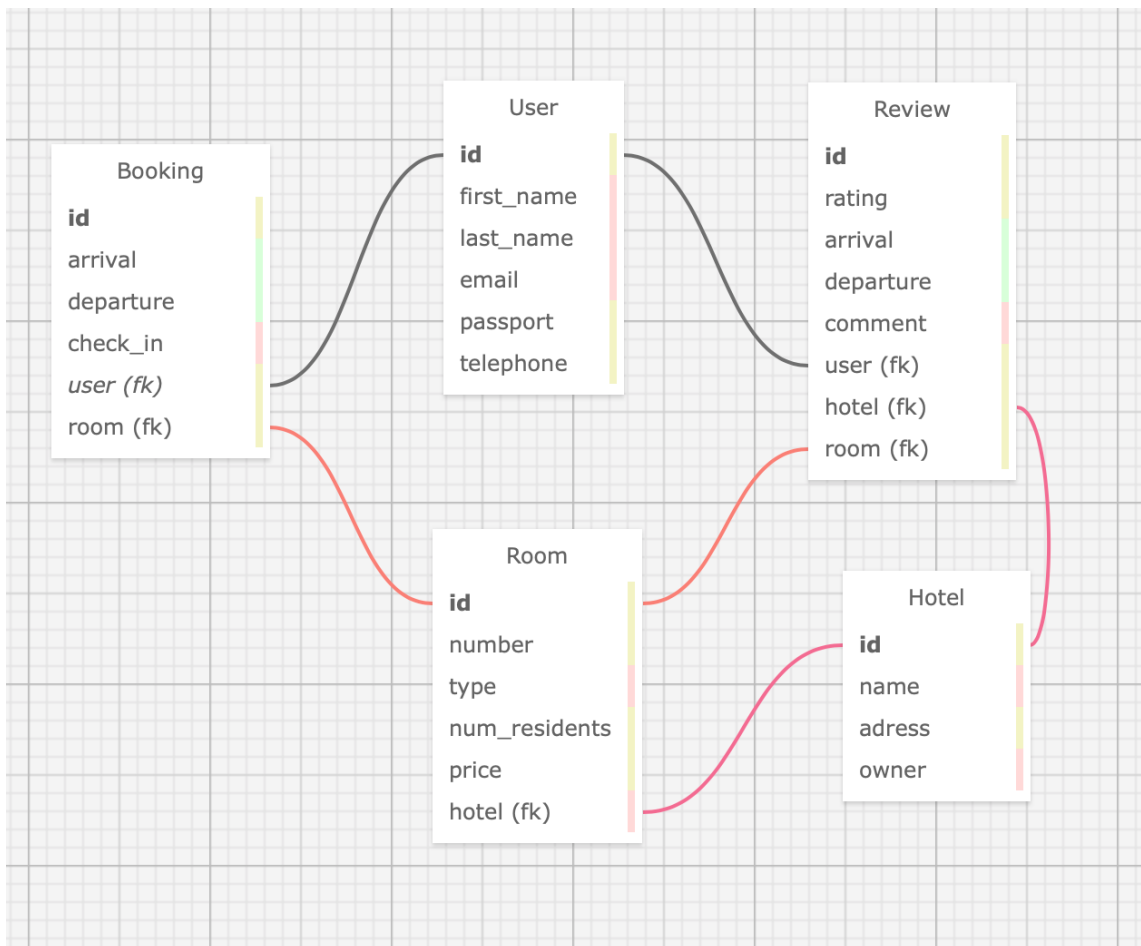
Необходимо реализовать следующий функционал:

- 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, 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.

### **Ход работы**

#### **1. Описание модели данных**

В модели данных 5 сущностей:



1. Сущность **User** (пользователь) включает атрибуты – имя, фамилия, электронная почта, номер паспорта и телефон.

```
class User(AbstractUser):
    passport = models.CharField(max_length=10, blank=True, null=True)
    telephone = models.CharField(max_length=11, blank=True, null=True)
    room = models.ManyToManyField('Room', through='Booking')
```

2. Сущность **Booking** (резервирование) включает атрибуты – прибытие, отправление, заселение и fk к сущностям user и room.

```
class Booking(models.Model):
    arrival = models.DateField()
    departure = models.DateField()
    check_in = models.BooleanField(default=False)
    user = models.ForeignKey('User', on_delete=models.CASCADE)
    room = models.ForeignKey('Room', on_delete=models.CASCADE)
```

3. Сущность **Room** (номер) включает атрибуты – номер, отель (fk), тип номера, цена за сутки, количество гостей.

```
class Room(models.Model):
    number = models.CharField(max_length=4)
    hotel = models.ForeignKey('Hotel', on_delete=models.CASCADE)

    class RoomType(models.TextChoices):
        SGL = 'SGL', "Single"
        DBL = 'DBL', "Double"
        TRPL = 'TRPL', "Triple"
        FR = 'FR', "Family Room"
        DL = 'DL', "DeLuxe"

    room_type = models.CharField(choices=RoomType.choices, max_length=4)
    price = models.CharField(max_length=10)
    guests = models.CharField(max_length=1)
```

4. Сущность **Hotel** (отель) включает атрибуты – название отеля, адрес, владелец отеля.

```
class Hotel(models.Model):
    name = models.CharField(max_length=50)
    address = models.CharField(max_length=100)
    owner = models.CharField(max_length=50)
```

5. Сущность Review (отзыв) включает атрибуты – пользователь (fk), оценка, прибытие, отправление, комментарий, отель (fk), номер (fk).

```
class Review(models.Model):
    user = models.ForeignKey('User', on_delete=models.SET_NULL, null=True)

    class Rating(models.TextChoices):
        a = '1', "Awful"
        b = '2', "Bad"
        c = '3', "Not bad"
        d = '4', "Well"
        e = '5', "Excellent"

    rating = models.CharField(choices=Rating.choices, max_length=2)
    arrival = models.DateField()
    departure = models.DateField()
    comment = models.CharField(max_length=1000)
    hotel = models.ForeignKey('Hotel', on_delete=models.SET_NULL, null=True)
    room = models.ForeignKey('Room', on_delete=models.SET_NULL, null=True)
```

## 2. Описание контроллеров в файле views.py

```

# стартовая страница
def index(request):
    return render(request, 'index.html')

# страница профиля
def profile(request):
    return render(request, 'profile.html')

# регистрация, контроллер – проверка ввода данных, создание пользователя
def registration(request):
    if request.method == 'POST':
        form = RegForm(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('index')

    else:
        form = RegForm()

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

# контроллер редактирование профиля
def edit_profile(request):
    user = request.user

    if request.method == 'POST':
        form = EditProfForm(request.POST, request.FILES, instance=user)
        if form.is_valid():
            form.save()
            return redirect('profile')

    else:
        form = EditProfForm(instance=user)

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

# список отелей
class ListHotels(ListView):
    model = Hotel
    template_name = 'hotels.html'
    context_object_name = 'hotels'

# список номеров, отсортированный по названию отелей
class ListRooms(ListView):
    template_name = 'rooms.html'
    context_object_name = 'rooms'

    def get_queryset(self):
        return Room.objects.all().order_by('hotel')

```

```

# номера конкретного отеля
class ListHotelRooms(ListView):
    template_name = 'rooms_by_hotel.html'
    context_object_name = 'hotel'

    def get_queryset(self):
        self.hotel = get_object_or_404(Hotel, pk=self.kwargs['pk'])
        return Room.objects.filter(hotel=self.hotel)

# детали о конкретном номере
class RoomDetail(DetailView):
    model = Room
    template_name = 'room.html'
    context_object_name = 'room'

# создать резервирование
# контроллер: автоматическая инициализация полей пользователь и Комната
# в форме по информации о комнате с предыдущей страницы и
# авторизованном пользователе
class CreateBooking(CreateView):
    form_class = CreateBookingForm
    model = Booking
    template_name = 'create_booking.html'
    context_object_name = 'booking'
    success_url = '/profile/bookings'

    def get_initial(self):
        initial = super(CreateBooking, self).get_initial()
        initial = initial.copy()
        initial['user'] = self.request.user.pk
        initial['room'] = get_object_or_404(Room, pk=self.kwargs['pk'])
        return initial

```

```

# список резервирований пользователя
class ListBookings(ListView):
    template_name = 'booking_list.html'
    context_object_name = 'booking_list'

    def get_queryset(self):
        self.user = self.request.user.pk
        return Booking.objects.filter(user=self.user)

# удалить резервирование
class DeleteBooking(DeleteView):
    model = Booking
    template_name = 'delete_booking.html'
    context_object_name = 'booking'
    success_url = '/profile/bookings'

# СПИСОК ОТЗЫВОВ
class ListReviews(ListView):
    model = Review
    template_name = 'reviews.html'
    context_object_name = 'reviews'

```

```

# создать отзыв
# контроллер: автоматическая инициализация полей пользователь, отель
# и комната в форме по информации о комнате с предыдущей страницы
# и авторизованном пользователе
class CreateReview(CreateView):
    model = Review
    form_class = CreateReviewForm
    template_name = 'create_review.html'
    context_object_name = 'review'
    success_url = '/reviews'

    def get_initial(self):
        initial = super(CreateReview, self).get_initial()
        initial = initial.copy()
        room = Room.objects.get(pk=self.kwargs['pk'])
        room_hotel = getattr(room, 'hotel')

        initial['user'] = self.request.user.pk
        initial['hotel'] = room_hotel
        initial['room'] = room

        return initial

# список гостей за месяц
class ListGuests(ListView):
    template_name = 'last_month_guests.html'
    context_object_name = 'guests_list'

    def get_queryset(self):
        last_month = datetime.today() - timedelta(days=30)
        booking_queries = Booking.objects.filter(check_in=True).filter(arrival__gte=last_month)

        return booking_queries

```

### 3. Описание роутеров файла urls.py

urls.py проекта

```

urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('django.contrib.auth.urls')),
    path('', include('hotels.urls')),
    url(r'^favicon\.ico$', RedirectView.as_view(url='/static/images/favicon.ico')),
]

```

accounts/ - встроенные django пути для авторизации

urls.py приложения

```

urlpatterns = [
    path('', index, name='index'),
    path('hotels/', ListHotels.as_view(), name='hotels'),
    path('rooms/', ListRooms.as_view(), name='rooms'),
    path('hotels/<int:pk>', ListHotelRooms.as_view()),
    path('rooms/<int:pk>', RoomDetail.as_view()),
    path('rooms/<int:pk>/book', CreateBooking.as_view(), name='booking'),
    path('registration/', registration, name='registration'),
    path('profile/', profile, name='profile'),
    path('profile/edit', edit_profile, name='edit_profile'),
    path('profile/bookings', ListBookings.as_view(), name='booking'),
    path('profile/bookings/delete/<int:pk>', DeleteBooking.as_view(), name='delete_booking'),
    path('reviews/', ListReviews.as_view(), name='review_list'),
    path('rooms/<int:pk>/add_review', CreateReview.as_view(), name='add_review'),
    path('last_guests/', ListGuests.as_view(), name='last_guests')
]

```

<int:pk> - передает первичный ключ в контроллер

## 4. Скриншоты работы программы

### Страница неавторизованного пользователя

You're not logged in, please log in or create a new account

### Регистрация нового пользователя

#### Registration

Username:  Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Email address:

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.

Register

### Домашняя страница авторизованного пользователя

Welcome, Lilia

### Профиль

## Lilia's personal page

Username: Lilia

Email: lilia@gmail.com

First name:

Last name:

Passport number: None

Telephone: None

[Edit](#)

My bookings



## Добавление информации в профиль

[Home](#) [Hotels](#) [Rooms](#) [Reviews](#) [Guests](#) [Profile](#)

[Log out](#)

### Edit Lilia's profile

First name:

Lilia

Last name:

Taris

Passport:

4321567890

Telephone:

89997776655

Username:

Lilia

Email:

lilia@gmail.com

Please enter your current username and email to confirm changes

[Update](#)

[Cancel](#)

[Home](#) [Hotels](#) [Rooms](#) [Reviews](#) [Guests](#) [Profile](#)

[Log out](#)

### Lilia's personal page

Username: Lilia

Email: lilia@gmail.com

First name: Lilia

Last name: Taris

Passport number: 4321567890

Telephone: 89997776655

[Edit](#)

[My bookings](#)

## Отели в системе

[Home](#) [Hotels](#) [Rooms](#) [Reviews](#) [Guests](#) [Profile](#)

[Log out](#)

### Available Hotels

Astoria Hotel

Alice Bribe

190000, St. Petersburg, Bolshaya Morskaya, 39

[Show rooms in this hotel](#)

UNDERWOOD Art Room

Georgy Almanov

191002, St. Petersburg, st. Rubinshteina, 23

[Show rooms in this hotel](#)

Boutique Hotel North Flower

Andrey Vivanov

191040, St. Petersburg, st. Marata, 40

[Show rooms in this hotel](#)

[Show all rooms](#)

## Astoria Hotel Rooms

Room number: 1  
Room type: Single  
Price per night: 14900  
Number of guests: 1

[Details](#)

Room number: 2  
Room type: Double  
Price per night: 20500  
Number of guests: 2

[Details](#)

Room number: 3  
Room type: DeLuxe  
Price per night: 35000  
Number of guests: 3

[Details](#)

[Back to hotels](#)

## Room 1 of Astoria Hotel

Room number: 1  
Room type: Single  
Price per night: 14900  
Number of guests: 1

[Book this room](#)

[Back to all rooms](#)

## Book a room

User:

Room:

Arrival:

Departure:

[Book this room](#)

[Back](#)

# Резервация номера в личном кабинете

HomeHotelsRoomsReviewsGuestsProfile

Log out

Lilia's bookings

Hotel: Astoria Hotel  
Room: 1  
Arrival: Jan. 19, 2021  
Departure: Jan. 29, 2021  
Check in: **No**  

Delete

Back to profile

# Администратор меняет статус check in

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Hotels › Bookings › Booking object (1)

AUTHENTICATION AND AUTHORIZATION

GroupsAdd

HOTELS

BookingsAdd

HotelsAdd

ReviewsAdd

RoomsAdd

UsersAdd

Change booking

HISTORY

Arrival:2021-01-19Today

Departure:2021-01-29Today

☒ Check in

User:Lilia

Room:1

Delete

Save and add another

Save and continue editing

SAVE

# Пользователь заселился

HomeHotelsRoomsReviewsGuestsProfile

Log out

Lilia's bookings

Hotel: Astoria Hotel  
Room: 1  
Arrival: Jan. 19, 2021  
Departure: Jan. 29, 2021  
Check in: **Yes**  

Delete

Back to profile

Можно добавить отзыв о номере

[Home](#) [Hotels](#) [Rooms](#) [Reviews](#) [Guests](#) [Profile](#)

[Log out](#)

## Room 1 of Astoria Hotel

Room number: 1

Room type: Single

Price per night: 14900

Number of guests: 1

[Book this room](#)

[Leave a review on this room](#)

[Back to all rooms](#)

[Home](#) [Hotels](#) [Rooms](#) [Reviews](#) [Guests](#) [Profile](#)

[Log out](#)

## Create a review

User:

Hotel:

Room:

Rating:

Arrival:

Departure:

Comment:

[Submit](#)

Отзыв на странице reviews

[Home](#) [Hotels](#) [Rooms](#) [Reviews](#) [Guests](#) [Profile](#)

[Log out](#)

## Reviews

User: Lilia

Hotel: Astoria Hotel

Room: 1

Arrival: Jan. 19, 2021

Departure: Jan. 29, 2021

Rating: Excellent

Review: Nice room, gorgeous atmosphere, good food.

Список гостей за последний месяц

## List of guests from the last month

Username	First name	Last name	Hotel	Room
Lilia	Lilia	Taris	Astoria Hotel	1

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки и умения для реализации web-сервисов средствами Django.