

**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»

**О Т Ч Е Т**

**Лабораторная работа №2, Web-программирование**

**Тема  
задания:**

Реализация простого сайта средствами Django

---

**Выполнил:  
Студент**

**Чанова С. Ю.**  
Фамилия И.О.

**K33422**  
номер группы

**Проверил:  
Преподаватель**

**Говоров А. И.**  
Фамилия И.О.

**Санкт-Петербург  
2020**

**Цель:** овладеть практическими навыками и умениями реализации web-сервисов средствами Django 3.1.

**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.

### 1. Описание модели данных

Модель данных состоит из 5 сущностей:

- Пользователь (User)
  - *имя, фамилия, email, дата рождения, номер паспорта, номер телефона*

```
class User(AbstractUser):
    birth_date = models.DateField(blank=True, null=True)
    passport = models.CharField(max_length=20, blank=True, null=True)
    phone_number = models.CharField(max_length=11, blank=True, null=True)
    room = models.ManyToManyField('Room', through='Booking')
```

- Комната (Room)

*номер, отель, тип комнаты, цена, вместимость*

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

    class RoomType(models.TextChoices):
        s = 's', "Standard"
        st = 'st', "Studio"
        js = 'js', "Junior Suite"
        su = 'su', "Suite"
        dl = 'dl', "Deluxe"
        fr = 'fr', "Family Room"
        hr = 'hr', "Honeymoon Room"

    room_type = models.CharField(choices=RoomType.choices, max_length=2)
    price = models.CharField(max_length=10)
    num_persons = models.CharField(max_length=1)

    def __str__(self):
        return self.number
```

- Резервирование (Booking) - связующая сущность

*дата въезда, дата выезда, пользователь, комната, заселен*

```
class Booking(models.Model):
    date_in = models.DateField()
    date_out = models.DateField()
    user = models.ForeignKey('User', on_delete=models.CASCADE)
    room = models.ForeignKey('Room', on_delete=models.CASCADE)
    checked_in = models.BooleanField(default=False)
```

- Отель (Hotel)

*название, владелец, адрес*

```
class Hotel(models.Model):
    name = models.CharField(max_length=40)
    owner = models.CharField(max_length=40)
    address = models.CharField(max_length=60)

    def __str__(self):
        return self.name
```

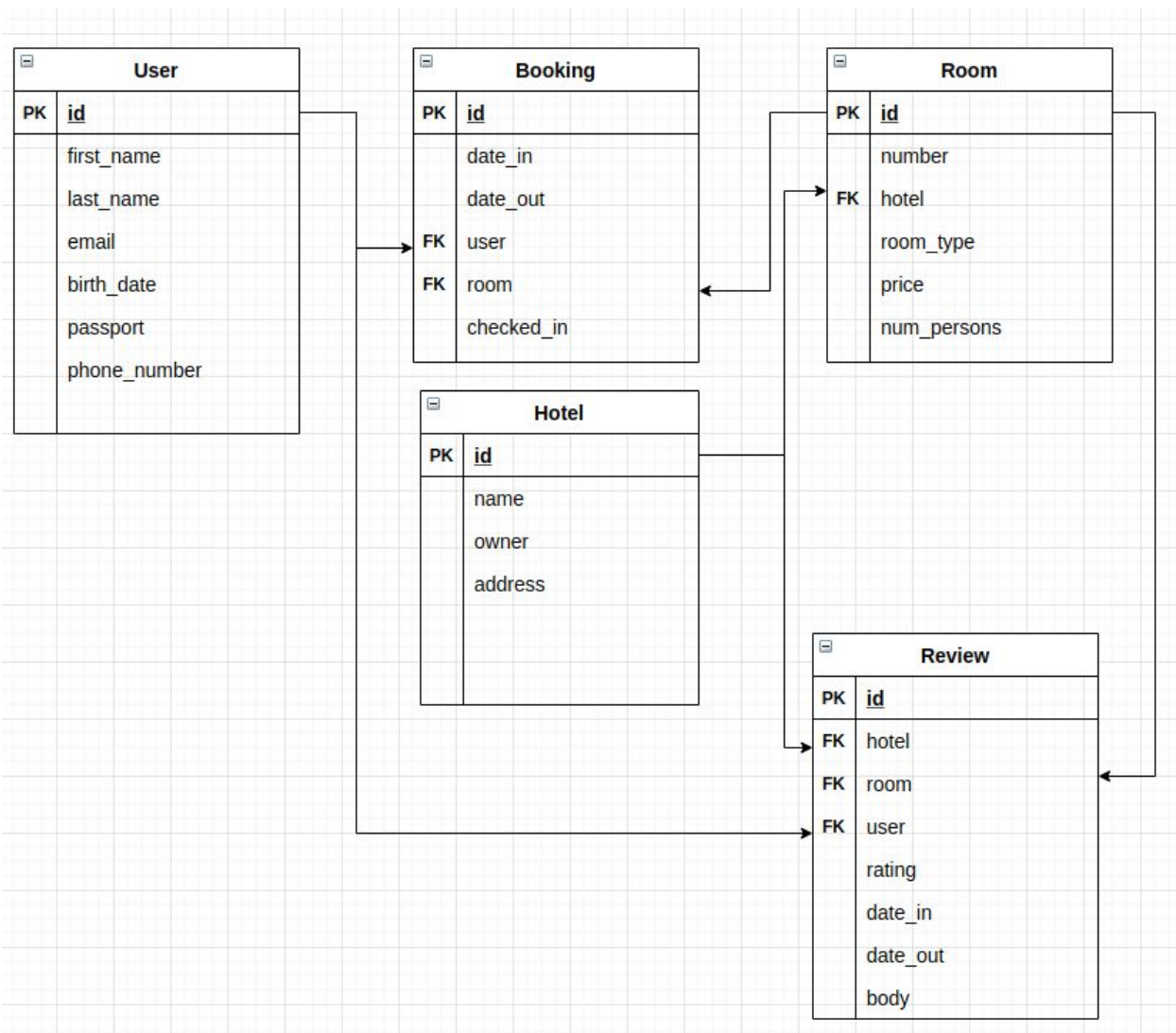
- Отзыв (Review)

*отель, комната, пользователь, рейтинг, дата въезда, дата выезда, текст*

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

    class Rating(models.TextChoices):
        f = '1', "Awful"
        d = '2', "Bad"
        c = '3', "Okay"
        b = '4', "Good"
        a = '5', "Great"

    hotel = models.ForeignKey('Hotel', on_delete=models.SET_NULL, null=True)
    room = models.ForeignKey('Room', on_delete=models.SET_NULL, null=True)
    rating = models.CharField(choices=Rating.choices, max_length=2)
    date_in = models.DateField()
    date_out = models.DateField()
    body = models.CharField(max_length=500)
```



## 2. Описание контроллеров в файле views

- Регистрация

*контроллер: проверка введенных данных и создание пользователя*

```
def register(request):
    if request.method == 'POST':
        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('index')

    else:
        form = RegistrationForm()

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

*форма: юзернейм, email, пароль и его подтверждение*

```
class RegistrationForm(UserCreationForm):

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

- Редактировать профиль

*контроллер:*

```
def edit_profile(request):
    user = request.user

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

    else:
        form = EditProfileForm(instance=user)

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

форма: поля модели User, виджет для удобства ввода даты, метод `clean_email` для проверки уникальности email в системе, переопределение `save`, сохраняя необходимый параметр email

```
class EditProfileForm(forms.ModelForm):
    username = forms.CharField(required=True)
    email = forms.EmailField(required=True)

    class Meta:
        model = User
        fields = ('first_name', 'last_name', 'birth_date', 'passport', 'phone_number')
        widgets = {
            'birth_date': forms.DateInput(attrs={'type': 'date'},
                                           format='%d-%m-%Y')
        }

    def clean_email(self):
        username = self.cleaned_data.get('username')
        email = self.cleaned_data.get('email')

        if email and User.objects.filter(email=email).exclude(username=username).count():
            raise forms.ValidationError('This email is already in use. Please enter a different one.')
        return email

    def save(self, commit=True):
        user = super(EditProfileForm, self).save(commit=False)
        user.email = self.cleaned_data['email']

        if commit:
            user.save()
        return user
```

- Список отелей

```
class ListHotels(ListView):
    model = Hotel
    template_name = 'hotels.html'
    context_object_name = 'hotels'
```

- Список комнат по возрастанию цены

```
class ListRooms(ListView):
    template_name = 'rooms.html'
    context_object_name = 'rooms'

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

- Список комнат 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)
```



- Отдельная страница комнаты

```
class RoomDetail(DetailView):  
    model = Room  
    template_name = 'room.html'  
    context_object_name = 'room'
```

- Создать резервирование

*контроллер: автоматическая инициализация полей Пользователь и Комната в форме, основываясь на 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
```

*форма: виджеты для удобного ввода дат, отключение возможности поменять вручную инициализированные поля, метод clean для корректного ввода дат*

```
class CreateBookingForm(forms.ModelForm):  
  
    class Meta:  
        model = Booking  
        fields = ('user', 'room', 'date_in', 'date_out')  
        widgets = {  
            'date_in': forms.DateInput(attrs={'type': 'date'},  
                                       format='%d-%m-%Y'),  
            'date_out': forms.DateInput(attrs={'type': 'date'},  
                                       format='%d-%m-%Y'),  
        }  
  
    def __init__(self, *args, **kwargs):  
        super(CreateBookingForm, self).__init__(*args, **kwargs)  
        self.fields['user'].disabled = True  
        self.fields['room'].disabled = True  
  
    def clean(self):  
        cleaned_data = super().clean()  
        date_in = cleaned_data.get("date_in")  
        date_out = cleaned_data.get("date_out")  
        today = date.today()  
  
        if date_out < date_in:  
            raise forms.ValidationError("Date in shouldn't be greater than 'Date out'")  
  
        if date_in < today:  
            raise forms.ValidationError("Can't book a room in the past")
```

- Список резервирований конкретного авторизованного пользователя

```
class ListBookings(ListView):
    template_name = 'bookings_list.html'
    context_object_name = 'bookings_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, основываясь на 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
```



форма: виджеты для удобного ввода дат и текста отзыва, отключение возможности поменять вручную инициализированные поля

```
class CreateReviewForm(forms.ModelForm):

    class Meta:
        model = Review
        fields = ('user', 'hotel', 'room', 'rating', 'date_in', 'date_out', 'body')
        widgets = {
            'date_in': forms.DateInput(attrs={'type': 'date'},
                                       format='%d-%m-%Y'),
            'date_out': forms.DateInput(attrs={'type': 'date'},
                                       format='%d-%m-%Y'),
            'body': forms.Textarea(),
        }

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

- Список постояльцев за последний 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(checked_in=True).filter(date_in__gte=last_month)

        return booking_queries
```

### 3. Описание роутеров файла urls

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('django.contrib.auth.urls')),
    path('', include('booking.urls')),
]
```

```
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('register/', register, name='register'),
    path('profile/', profile, name='profile'),
    path('profile/edit', edit_profile, name='edit profile'),
    path('profile/bookings', ListBookings.as_view(), name='bookings'),
    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')
]
```

accounts/ - встроенные django пути для авторизации

<int:pk> - передает первичный ключ в контроллер

#### 4. Скрины работы программы

- Домашняя страница неавторизованного пользователя

Home Hotels Rooms Reviews Guests

You're not logged in :(

- Регистрация

Home Hotels Rooms Reviews Guests

## 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

Home Hotels Rooms Reviews Guests Profile

Welcome, elephant17

- Страница профиля

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

## elephant17's personal page

Username: elephant17  
Email: elephant17@outlook.com  
First name:  
Last name:  
Date of birth: None  
Passport number: None  
Phone number: None  
[Edit](#)

My bookings

- Добавим информацию в профиль

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

## Edit elephant17's profile

First name:

Mihail

Last name:

Bolshoy

Birth date:

mm/dd/yyyy



Passport:

7500 642483

Phone number:

89837762956

Username:

elephant17

Email:

elephant17@outlook.com

Please enter your current username and email to confirm changes

Update

[Cancel](#)

- Просмотрим отели

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

## Available Hotels

Knights Inn  
Brienne Tarth  
"7 Pengwern Terrace, Wallasey",CH45 1HR  
[Show rooms for this hotel](#)

Hotel Crossroads  
James Lanern  
"164 Park Road, Blackpool",FY1 5QN  
[Show rooms for this hotel](#)

Moody Moon  
Lilith Crooke  
"Rhos Gwyn, Brynteg",LL78 8JG  
[Show rooms for this hotel](#)

Show all rooms

- Просмотрим комнаты одного из отелей

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

## Knights Inn Rooms

Room number: 1  
Room type: Standard  
Price per night: 100\$  
Number of persons: 1  
[Details](#)

Room number: 2  
Room type: Deluxe  
Price per night: 200\$  
Number of persons: 1  
[Details](#)

Room number: 3  
Room type: Honeymoon Room  
Price per night: 300\$  
Number of persons: 2  
[Details](#)

Back to hotels

- Выберем комнату

Home Hotels Rooms Reviews Guests Profile

## Room 2 of Knights Inn

Room number: 2  
Room type: Deluxe  
Price per night: 200\$  
Number of persons: 1

[Book this room](#)[Leave a review](#)

[Back to all rooms](#)

- Зарезервируем комнату

Home Hotels Rooms Reviews Guests Profile

## Book a room

User:

Room:

Date in:

Date out:

[Book this room](#)

[Back](#)

- Резервирование появилось в личном кабинете

Home Hotels Rooms Reviews Guests Profile

## elephant17's bookings

Hotel: Knights Inn  
Room: 2  
Date in: Oct. 21, 2020  
Date out: Oct. 24, 2020  
Checked in: **No**

[Delete](#)

---

[Back to profile](#)

- Администратор поменял статус резервирования - пользователь заселился

Home Hotels Rooms Reviews Guests Profile

## elephant17's bookings

Hotel: Knights Inn  
Room: 2  
Date in: Oct. 21, 2020  
Date out: Oct. 24, 2020  
Checked in: **Yes**

Delete [Leave a review of this room |](#)

---

[Back to profile](#)

- Добавим отзыв на комнату

Home Hotels Rooms Reviews Guests Profile

## Add a review

User:

Hotel:

Room:

Rating:

Date in:

Date out:

Body:

Submit



- Список отзывов

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

## Reviews

User: spiritofsofya  
Hotel: Moody Moon  
Room: 33  
Date in: Oct. 13, 2020  
Date out: Oct. 20, 2020  
Rating: Good  
Review: i had a goo time

User: ladybug  
Hotel: Knights Inn  
Room: 1  
Date in: Aug. 5, 2020  
Date out: Aug. 19, 2020  
Rating: Good  
Review: i liked this inn

User: elephant17  
Hotel: Knights Inn  
Room: 2  
Date in: Oct. 21, 2020  
Date out: Oct. 24, 2020  
Rating: Great  
Review: a great hotel with good breakfasts and service

- Список постояльцев за последний месяц

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

## List of guests from the last month

Username	First name	Last name	Hotel	Room
spiritofsofya	Sofya	Chanova	Moody Moon	33
Username	First name	Last name	Hotel	Room
elephant17	Mihail	Bolshoy	Knights Inn	2

**Вывод:** в ходе выполнения лабораторной работы я овладела практическими навыками и умениями реализации web-сервисов средствами Django 3.1.