

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

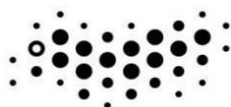
**Лабораторная работа № 2**

на тему  
“Реализация сайта средствами Django”  
по дисциплине **«WEB программирование»**

Выполнила: Исхакова Эмина

Группа: К33422

Проверил: Говоров А.И.



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург, 2021

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

**Практическое задание:**

Реализовать сайт используя фреймворк Django 3 и СУБД PostgreSQL \*, в соответствии с вариантом задания лабораторной работы.

**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 (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.

Структура сайта:

Модели

```
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')

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

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"
```

```
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
```

```
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)
```

```
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)
```

Страницы:

```
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')
```

Необходимые страницы:

[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](#)

## Room 2 of City Centre Hotel

Room number: 2

Room type: Standard

Price per night: 10\$

Number of persons: 1

[Book this room](#)

[Leave a review](#)

# caramel\_nana's bookings

Hotel: City Centre Hotel

Room: 2

Date in: Jan. 27, 2022

Date out: Jan. 28, 2022

Checked in: **No**

Delete

Are you sure you want to cancel the booking of Room 2, City Centre Hotel?

Yes

No, take me back

The screenshot shows the Django administration interface for the 'Hotel' app. The left sidebar contains a menu with categories: 'AUTHENTICATION AND AUTHORIZATION' (Groups), 'HOTEL' (Bookings, Hotels, Reviews, Rooms, Users), and 'HISTORY'. The main content area is titled 'Change booking' and 'Booking object (2)'. It contains the following fields:

- Date in:** 2022-01-27 (Today | 📅). Note: You are 3 hours ahead of server time.
- Date out:** 2022-01-28 (Today | 📅). Note: You are 3 hours ahead of server time.
- User:** caramel\_nana (dropdown menu with edit and add icons).
- Room:** 2 (dropdown menu with edit and add icons).
- Checked in:** ☒ (circled in red).

At the bottom, there are three buttons: 'Delete' (highlighted in red), 'Save and add another', and 'Save and continue editing'. A 'SAVE' button is also visible on the right.

Вывод:

В результате выполнения лабораторной работы были получены практические навыки, умения реализации web-серверов на фреймворке Django.