

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО**

Факультет «Инфокоммуникационных технологий»

Направление подготовки «45.04.04 Интеллектуальные системы в гуманитарной сфере»

О Т Ч Е Т

Лабораторная работа №2

Тема: «Реализация простого сайта средствами Django»

Выполнила:

**Студентка Белова А.С.
К33421**

Проверил:

Преподаватель Говоров А.И.

Санкт-Петербург

2021

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

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 (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.

Выполнение работы

Файл models.py

```
from django.db import models
from django.contrib.auth.models import AbstractUser
from django.contrib.auth import get_user_model

class Guest(AbstractUser):
    last_name = models.CharField(max_length=30)
    first_name = models.CharField(max_length=30)
    birth_date = models.DateField(null=True)
    passport = models.CharField(max_length=30, null=True)

    def __str__(self):
        return self.last_name + " " + self.first_name

class Hotel(models.Model):
    name = models.CharField(max_length=30)
    owner = models.CharField(max_length=30)
    address = models.CharField(max_length=40, null=True)
    description = models.CharField(max_length=100, null=True)

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

```

class Room(models.Model):
    Guest = get_user_model()
    number = models.IntegerField()
    type = models.CharField(max_length=30, null=True)
    price = models.IntegerField(null=True)
    capacity = models.IntegerField(null=True)
    amenities = models.CharField(max_length=30, null=True)
    hotel = models.ForeignKey(Hotel, on_delete=models.CASCADE, null=True)
    guest = models.ManyToManyField(Guest, through='Accommodation')

    def __str__(self):
        return f"{self.number} | {self.hotel}"

class Accommodation(models.Model):
    Guest = get_user_model()
    check_in_date = models.DateField()
    check_out_date = models.DateField()
    guest = models.ForeignKey(Guest, on_delete=models.CASCADE, null=True)
    room = models.ForeignKey(Room, on_delete=models.CASCADE, null=True)

    def __str__(self):
        return f"{self.guest} | {self.check_in_date} | {self.check_out_date}"

```

```

class Comment(models.Model):
    Guest = get_user_model()
    hotel = models.ForeignKey(Hotel, on_delete=models.CASCADE, null=True)
    comment = models.CharField(max_length=200)
    rating = models.IntegerField(null=True)
    guest = models.ForeignKey(Guest, on_delete=models.CASCADE, null=True)
    accommodation = models.ForeignKey(Accommodation, on_delete=models.CASCADE, null=True)

    def __str__(self):
        return f"{self.guest} | {self.rating}"

```

Файл views.py

```
from django.shortcuts import render, get_object_or_404
from django.http import Http404
from .models import Guest, Room, Accommodation, Hotel, Comment
from .forms import GuestForm, AccommodationForm, CommentForm
from django.views.generic import UpdateView, DeleteView
from django.views.generic.list import ListView
import datetime
```

```
def guest_list(request):
    data = {"guests": Guest.objects.all()}
    return render(request, "guest_list.html", data)
```

```
def guest_create(request):
    data = {}
    form = GuestForm(request.POST or None)
    if form.is_valid():
        form.save()
    data["form"] = form
    return render(request, "guest_create.html", data)
```

```
def room_list(request):
    data = {"rooms": Room.objects.all()}
    return render(request, "room_list.html", data)
```

```
def book(request):
    data = {}
    form = AccommodationForm(request.POST or None)
    if form.is_valid():
        form.save()
    data["form"] = form
    return render(request, "book.html", data)
```

```
def book_list(request):
    data = {"accoms": Accommodation.objects.all()}
    return render(request, "accom_list.html", data)
```

```

def last_month(request):
    data = {}
    try:
        month = datetime.date.today() - datetime.timedelta(days=31)
        data["accoms"] = Accommodation.objects.all().filter(check_in_date__gte=str(month),
                                                            check_out_date__lte=str(datetime.date.today()))
        print(data)
    except Accommodation.DoesNotExist:
        raise Http404("No guests this month yet :(")
    return render(request, "accom_list.html", data)

def accommodation_list(request):
    data = {"accoms": Accommodation.objects.all()}
    return render(request, "accom_list.html", data)

def home(request):
    return render(request, "home.html")

```

```

class AccomUpdate(UpdateView):
    model = Accommodation
    template_name = "accom_update.html"
    fields = ["check_in_date", "check_out_date", "guest", "room"]
    success_url = "/accom/list/"

class AccomDelete>DeleteView):
    model = Accommodation
    template_name = "accom_delete.html"
    success_url = "/accom/list/"

class HotelList(ListView):
    model = Hotel
    template_name = 'hotel.html'
    all_hotels = Hotel.objects

def hotel_view(request, pk):
    hotel = Hotel.objects.get(id=pk)
    comments = Comment.objects.filter(hotel=hotel)
    return render(request, 'hotel_detail.html', {'hotel': hotel, 'comments': comments})

```

```

def comment(request, pk):
    obj = get_object_or_404(Hotel, id=pk)
    author = request.user
    if request.method == 'POST':
        form = CommentForm(request.POST)
        if form['comment'].value():
            if form['rating'].value():
                if form.is_valid():
                    com = form.save(commit=False)
                    com.author = author
                    com.hotel = obj
                    com.save()
            else:
                form = CommentForm()
    return render(request, 'review.html', {'form': form})

```

Файл urls.py

```

from django.urls import path
from .views import *

urlpatterns = [
    path("guest/list/", guest_list),
    path("guest/create/", guest_create),
    path("room/list/", room_list),
    path("book/", book),
    path("book/list/", book_list),
    path("month/", last_month),
    path("accom/list/", accommodation_list),
    path("accom/<int:pk>/update/", AccomUpdate.as_view()),
    path("accom/<int:pk>/delete/", AccomDelete.as_view()),
    path("home/", home),
    path('hotel/list/', HotelList.as_view()),
    path('hotel/<str:pk>', hotel_view),
    path('hotel/review/<str:pk>', comment)
]

```

Результат

- Главная страница

Home

[Sign Up](#)
[Our guests](#)
[Book a room](#)
[All bookings](#)
[Our rooms](#)
[Our hotels](#)
[Update booking](#)
[Delete booking](#)
[Last month](#)

- Регистрация пользователей

New user

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

Password:

Last name:

First name:

Birth date:

Passport:

[Home](#)

- Список добавленных пользователей

Username: itto
Last Name: Arataki
First Name: Itto
Birth date: Nov. 9, 1970
Passport: 1104 456389

Username: kazuha
Last Name: Kaedehara
First Name: Kazuha
Birth date: July 13, 1985
Passport: 1523 786855

Username: ayaka
Last Name: Kamisato
First Name: Ayaka
Birth date: Dec. 16, 1999
Passport: 1448 963844

[Home](#)

- Бронирование номера

Book a room

Check in date:

Check out date:

Guest:

Room:

[Home](#)

- Список номеров

Rooms

Number: 666
Type: luxury
Price: 100000
Capacity: 8
Amenities: parking, free wifi, pool
Hotel: Super Hotel

Number: 1408
Type: luxury
Price: 10000
Capacity: 3
Amenities: mini-fridge, room service
Hotel: Super Hotel

Number: 21
Type: Studio
Price: 100
Capacity: 1
Amenities: water and light
Hotel: Okay Hotel

Number: 13
Type: Duplex
Price: 200
Capacity: 2
Amenities: free water
Hotel: Okay Hotel

[Home](#)

- Добавление комментариев

Text:

Rating:

Guest:

Accommodation:

[Home](#)

Hotel Super Hotel

Name: Super Hotel

Owner: Diluc

Address: Mondstadt

Description: Guess you can rest a bit here, it's cool

Comments:

Guest: Arataki Itto

Rating: 9

Text: nice hotel nice me

Guest: Kamisato Ayaka

Rating: 10

Text: wow!!! it was really cool

[Comment](#) [Home](#)

- Изменение брони

Check in date:

Check out date:

Guest:

Room:

[Home](#)

- Удаление брони

Do you want to delete the accommodation "Garza Judith | 2021-11-01"?

[Home](#)