

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И
ОПТИКИ**

ФАКУЛЬТЕТ ИКТ

Основы web-программирования

Отчет

по лабораторной работе №1

«Реализация web-сервисов средствами Python»

Выполнила:

Шамахова Александра

Студент группы К3443

Санкт-Петербург

2020

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

Программное обеспечение: Python 3.6, Django 2.2, PostgreSQL *.

Практическое задание:

Реализовать сайт используя фреймворк Django 2.2 и СУБД PostgreSQL *, в соответствии с вариантом задания лабораторной работы.

Вариант: список отелей.

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

Необходимо реализовать следующий функционал:

- Регистрация новых пользователей.
- 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.

Выполнение работы:

Спроектированная БД:



Рисунок 1 – Модель данных

Были созданы проект Django laboratory_work_1 и приложение hotels_list. К проекту подключена БД PostgreSQL hotels при помощи модуля psycopg2.

Модели Hotels (Отели), Rooms (Номера), Reservations (Резервирования), Reviews (Отзывы) описаны в models.py.

```
from django.db import models
from django.contrib.auth import get_user_model
User = get_user_model()

class Hotels(models.Model):
    hotel_name = models.CharField("Название отеля", max_length=100)
    hotel_owner = models.CharField("Владелец отеля", max_length=100)
    hotel_address = models.CharField("Адрес отеля", max_length=500)
    hotel_description = models.TextField("Описание отеля", max_length=1000)
    hotel_capacity = models.IntegerField("Вместимость отеля")
    hotel_class = models.CharField("Класс отеля", max_length=50)

    class Meta:
        verbose_name = "Отель"
        verbose_name_plural = "Отели"

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

Рисунок 2 – Код models.py, таблица Отели

```
class Rooms(models.Model):
    ROOM_TYPES = (
        ('1', '1 гость'),
        ('2', '2 гостя'),
        ('3', '3 гостя'),
        ('4', '4 гостя'),
        ('6', '6 гостей')
    )
    room_hotel = models.ForeignKey(Hotels, on_delete=models.CASCADE)
    room_type = models.CharField("Тип номера", max_length=1, choices=ROOM_TYPES)
    room_price = models.CharField("Цена номера", max_length=20)

    class Meta:
        verbose_name = "Номер"
        verbose_name_plural = "Номера"

    def __str__(self):
        return self.room_type + " " + self.room_price
```

Рисунок 3 – Код models.py, таблица Номера

```

class Reservations(models.Model):
    reservation_user = models.ForeignKey(User, verbose_name="Клиент", on_delete=models.CASCADE)
    reservation_room = models.ForeignKey(Rooms, on_delete=models.CASCADE)
    reservation_start = models.DateField()
    reservation_end = models.DateField()

    class Meta:
        verbose_name = "Резервирование"
        verbose_name_plural = "Резервирования"

```

Рисунок 4 – Код models.py, таблица Резервирования

```

class Reviews(models.Model):
    REVIEW_RANKS = (
        ('1', '1'),
        ('2', '2'),
        ('3', '3'),
        ('4', '4'),
        ('5', '5'),
        ('6', '6'),
        ('7', '7'),
        ('8', '8'),
        ('9', '9'),
        ('10', '10')
    )
    REVIEW_PERIODS = (
        ('Summer', 'Summer'),
        ('Autumn', 'Autumn'),
        ('Winter', 'Winter'),
        ('Spring', 'Spring')
    )
    review_user = models.ForeignKey(User, verbose_name="Клиент", on_delete=models.SET_NULL, null=True)
    review_hotel = models.ForeignKey(Hotels, on_delete=models.CASCADE)
    review_rank = models.CharField("Рейтинг", max_length=2, choices=REVIEW_RANKS)
    review_period = models.CharField("Период проживания", max_length=6, choices=REVIEW_PERIODS)
    review_text = models.TextField("Отзыв", max_length=500)
    review_created = models.DateTimeField("Дата создания", auto_now_add=True, null=True)

    class Meta:
        verbose_name = "Отзыв"
        verbose_name_plural = "Отзывы"

    def __str__(self):
        return "Рейтинг: " + self.review_rank + ". " + self.review_text

```

Рисунок 5 – Код models.py, таблица Отзывы

С помощью `python3 manage.py makemigrations` и `python3 manage.py migrate` созданы миграции.

Все модели были зарегистрированы в `admin.py`. Через административную панель были добавлены примеры в таблицы.

Регистрация и авторизация пользователей реализована через `django-allauth`. Данные о пользователях хранятся в модели `Users`.

В `templates` были созданы шаблоны `hotel_list.html` и `hotel_single.html`.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Отели</title>
</head>
<body>
  {% for hotel in hotels %}
    <h3><a href="{% url 'hotel_single' pk=hotel.id %}">
      {{ hotel.hotel_name }}</a></h3>
    <p>{{ hotel.hotel_address }}</p>
    <hr/>
  {% endfor %}
</body>
</html>

```

Рисунок 6 – Шаблон hotel_list.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{{ hotel.hotel_name }}</title>
</head>
<body>
  <h3>{{ hotel.hotel_name }}</h3>
  Владелец: {{ hotel.hotel_owner }}<br>
  Адрес: {{ hotel.hotel_address }}<br>
  Вместимость: {{ hotel.hotel_capacity }}<br>
  Класс: {{ hotel.hotel_class }}<br>
  <p>{{ hotel.hotel_description }}</p>
  <hr>
  <h5>Номера</h5>
  {% for room in rooms %}
    &#10003; Гостей: {{ room.room_type }}<br>
    {{ room.room_price }}<br>
  {% endfor %}

```

Рисунок 7 – Шаблон hotel_single.html (Часть 1)

```

<hr>
{% for review in reviews %}
    Клиент: {{ review.review_user }}<br>
    Рейтинг: {{ review.review_rank }}<br>
    Сезон: {{ review.review_period }}<br>
    {{ review.review_text }}<br>
    {{ review.review_created }}<br>
    <hr>
{% endfor %}
{% if user.is_active %}
    <form action="" method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Отправить</button>
    </form>
{% else %}
    <h4>Для добавления комментария авторизируйтесь!</h4>
{% endif %}
</body>
</html>

```

Рисунок 8 – Шаблон hotel_single.html (Часть 2)

Во views.py реализованы следующие представления:

hotel_list — функционально выводит список всех отелей и их адреса.

hotel_single — функционально выводит информацию о конкретном отеле, а также о номерах, представленных в нём, и комментарии авторизованных пользователей, также позволяет оставить комментарий (forms.py).

```

def hotel_list(request):
    hotels = Hotels.objects.all()
    return render(request, "hotels/hotel_list.html", {"hotels": hotels})

def hotel_single(request, pk):
    hotel = get_object_or_404(Hotels, id=pk)
    rooms = Rooms.objects.filter(room_hotel=pk)
    review = Reviews.objects.filter(review_hotel=pk)
    if request.method == "POST":
        form = ReviewForm(request.POST)
        if form.is_valid:
            form = form.save(commit=False)
            form.review_user = request.user
            form.review_hotel = hotel
            form.save()
            return redirect(hotel_single, pk)
    else:
        form = ReviewForm()
    return render(request,
                  "hotels/hotel_single.html",
                  {"hotel": hotel,
                  "rooms": rooms,
                  "reviews": review,
                  "form": form})

```

Рисунок 9 – Код views.py

```

from django.forms import ModelForm
from .models import Reviews

class ReviewForm(ModelForm):
    class Meta:
        model = Reviews
        fields = [
            "review_period",
            "review_rank",
            "review_text"
        ]

```

Рисунок 10 – Код forms.py

Адресация настраивается в urls.py. Urls приложения:

```

from django.urls import path
from . import views

urlpatterns = [
    path('', views.hotel_list, name="hotel_list"),
    path('hotel/<int:pk>', views.hotel_single, name="hotel_single"),
]

```

Рисунок 11 – Код hotels_list/urls.py

Urls проекта:

```

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('allauth.urls')),
    path('', include('hotels_list.urls'))
]

```

Рисунок 12 – Код laboratory_work_1/urls.py

Так выглядит основная страница, то есть список отелей:



Рисунок 13 – Список отелей

При нажатии на название отеля происходит переход на страницу информации о конкретном отеле. Выводится информация о владельце отеля, классе, вместимости, адресе и краткое описание отеля, под ней расположена информация о номерах, состоящая из количества человек, на которое они рассчитаны, и их стоимости, ниже следуют отзывы об отеле. Отправка формы с отзывом доступна, если пользователь авторизован, если нет — выводится информация о необходимости регистрации. В отзыве указывается сезон, во время которого пользователь гостил в отеле, рейтинг от 1 до 10 и сам текст.

Sunny Hill

Владелец: Mare Jacob
Адрес: Str.Mare Jacovljevic 9A
Вместимость: 27
Класс: 3 звезды


Санни Хилл — прекрасный выбор для тех, кто хочет восстановить силы. Это место хорошо известно своей романтической атмосферой и близостью к отличным ресторанам и достопримечательностям.


Номера

✓ Гостей: 1
10 000 рублей
✓ Гостей: 6
40 000 рублей
✓ Гостей: 3
25 000 рублей

Клиент: admin
Рейтинг: 9
Сезон: Summer
Замечательно отдохнули, сервис хороший, единственный минус — шумные соседи, приходилось вставать в восемь утра.
1 ноября 2020 г. 14:48

Рисунок 14 – Информация об отеле

Период проживания: 

Рейтинг: 

Отзыв:

Рисунок 15 – Форма отправки отзыва

Так выглядит страница авторизации:

Menu:

- [Sign In](#)
- [Sign Up](#)

Войти

Если у вас ещё нет учётной записи, пожалуйста, сначала [зарегистрируйтесь](#).

Имя пользователя:

Пароль:

Запомнить меня: ☐

[Забыли пароль?](#)

Рисунок 16 – Страница авторизации

Так выглядит страница регистрации:

Menu:

- [Sign In](#)
- [Sign Up](#)

Регистрация

Уже зарегистрированы? [Войдите](#).

Имя пользователя:

E-mail (опционально):

Пароль:

Пароль (ещё раз):

Рисунок 17 – Регистрация

Так выглядит страница выхода:

Menu:

- [Change E-mail](#)
- [Sign Out](#)

Выйти

Вы уверены, что хотите выйти?

Выйти

Рисунок 18 – Выход из аккаунта

Панель администрирования для администратора:

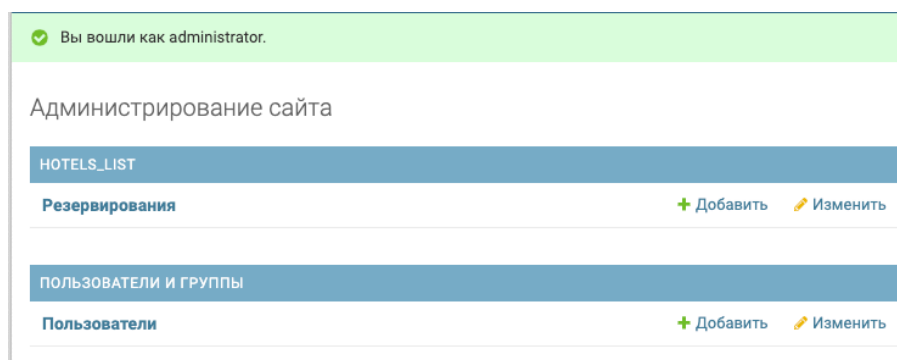


Рисунок 19 – Панель администратора

Панель администрирования суперпользователя:

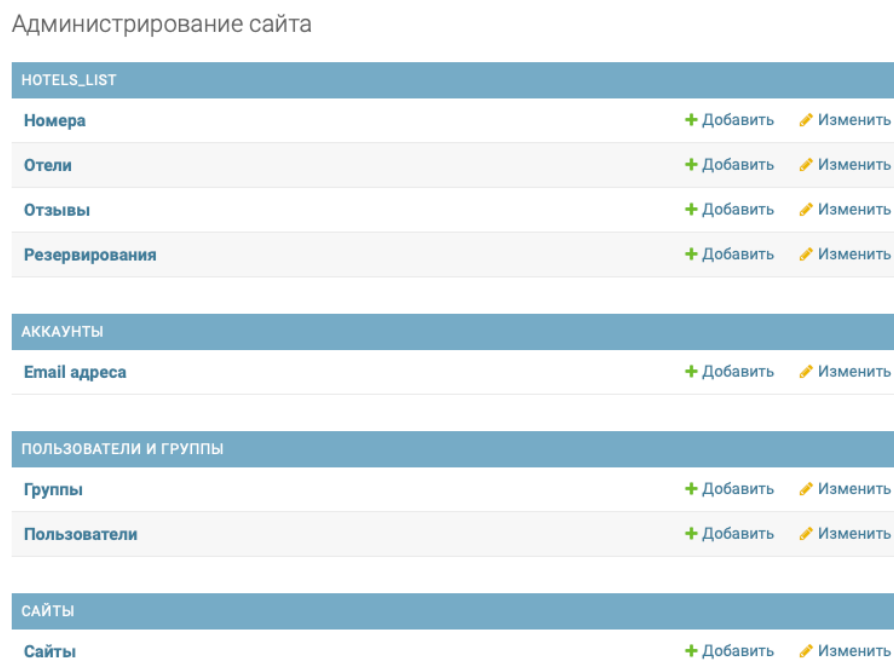


Рисунок 20 – Панель суперпользователя