

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И
ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЕТ
о лабораторной работе №2
«Реализация простого сайта на django»
по дисциплине
«Web программирование»

Выполнила
Голуб А.Л.
группа К33421

Проверил
Говоров А. И.

Санкт-Петербург
2021

Задание: табло отображения информации об **авиаперелетах**

Хранится информация о номере рейса, авиакомпании, отлете, прилете, типе (прилет, отлет), номере гейта.

Необходимо реализовать следующий функционал:

- Регистрация новых пользователей.
- Просмотр и резервирование мест на рейсах. Пользователь должен иметь возможность редактирования и удаления своих резервирований.
- Администратор должен иметь возможность зарегистрировать на рейс пассажира и вписать в систему номер его билета средствами Django-admin.
- В клиентской части должна формироваться таблица, отображающая всех пассажиров рейса.
- Написание отзывов к рейсам. При добавлении комментариев, должны сохраняться дата рейса, текст комментария, рейтинг (1-10), информация о комментаторе.

Модель данных (models.py)

```
# user - пользователь / пассажир
class User(AbstractUser):
    username = models.CharField(max_length=30, unique=True)
    password = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    first_name = models.CharField(max_length=30)
    passport = models.CharField(max_length=10, blank=True, null=True)

    def __str__(self):
        if self.is_superuser:
            return 'superuser'
        return self.last_name + ' ' + self.first_name

# flight - рейс
class Flight(models.Model):
    passengers = models.ManyToManyField(settings.AUTH_USER_MODEL,
    through='Reservation')
    number = models.CharField(max_length=30)
    airline = models.CharField(max_length=30)
    departure = models.DateField()
    arrival = models.DateField()
    gate = models.CharField(max_length=10)

    DEPARTURE = 'Departure'
    ARRIVAL = 'Arrival'
    type = models.CharField(
        max_length=30,
        choices=[(DEPARTURE, 'Departure'), (ARRIVAL, 'Arrival')]
    )

    def __str__(self):
        return self.airline + ' ' + self.number
```

```

# reservation - резервирование
class Reservation(models.Model):
    passenger = models.ForeignKey(settings.AUTH_USER_MODEL,
    on_delete=models.CASCADE)
    flight = models.ForeignKey(Flight, on_delete=models.CASCADE)
    seat = models.CharField(max_length=15)
    ticket = models.CharField(max_length=15, blank=True, null=True)

    def __str__(self):
        return 'Резервирование'

# review - отзыв
class Review(models.Model):
    author = models.ForeignKey(settings.AUTH_USER_MODEL,
    on_delete=models.SET_NULL, null=True)
    flight = models.ForeignKey(Flight, on_delete=models.SET_NULL, null=True)
    rating = models.IntegerField([(i, i) for i in range(1, 10)])
    comment = models.CharField(max_length=400, blank=True, null=True)

```

Контроллеры (views.py)

```

# create user
class UserCreateView(CreateView):
    model = User
    fields = ['username', 'password', 'last_name', 'first_name', 'passport']
    success_url = '/success'

# view user info
class UserRetrieveView(DetailView):
    model = User

# create reservation
class ReservationCreateView(CreateView):
    model = Reservation
    form_class = ReservationForm
    success_url = '/success'

# view reservation
class ReservationRetrieveView(DetailView):
    model = Reservation

# update reservation data
class ReservationUpdateView(UpdateView):
    model = Reservation
    form_class = ReservationForm
    success_url = '/success'

# delete reservation
class ReservationDeleteView(DeleteView):
    model = Reservation
    success_url = '/success'

```

```

# list passengers on specified flight
class PassengerList(ListView):
    model = Reservation
    template_name = 'passenger_list_view.html'

# write review
class ReviewCreateView(CreateView):
    model = Review
    form_class = ReviewForm
    success_url = '/success'

# success
def success(request):
    return render(request, 'success.html')

```

Маршрутизация (urls.py)

```

'user/register/' # create user
'user/<int:pk>/' # view user info

'reservation/create' # create reservation
'reservation/<int:pk>' # view reservation info
'reservation/edit/<int:pk>' # edit reservation
'reservation/delete/<int:pk>' # delete reservation

'flight/<int:pk>' # view list of passengers on a specified flight
'review' # write review

path('success', success) # success url

```

Примеры работы программы

Регистрация пользователя

←
→
↺
🏠
localhost:8000/user/register/

Username:

Password:

Last name:

First name:

Passport:

Просмотр информации о пользователе

← → ↺ 🏠 ⓘ localhost:8000/user/6/

Григорьева Лидия

Фамилия: Григорьева

Имя: Лидия

Паспортные данные: 1212121212

Создание резервирования

← → ↺ 🏠 ⓘ 127.0.0.1:8000/reservation/create

Passenger: Григорьева Лидия ▾

Flight: Aeroflot 65 ▾

Seat: 30

Ticket:

Submit

Просмотр резервирования

← → ↺ 🏠 ⓘ 127.0.0.1:8000/reservation/1

Резервирование

Пассажир: Григорьева Лидия

Рейс: 65

Авиакомпания: Aeroflot

Время вылета: Nov. 6, 2021

Время прилета: Nov. 6, 2021

Тип: Arrival

Гейт: 4

Место: 30

Номер билета: 1

Изменение резервирования

← → ↻ 🏠 ⓘ 127.0.0.1:8000/reservation/edit/1

Passenger: Григорьева Лидия ▾

Flight: Aeroflot 65 ▾

Seat: 30

Ticket: 1

Submit

Отмена резервирования

← → ↻ 🏠 ⓘ 127.0.0.1:8000/reservation/delete/1

Вы уверены, что хотите отменить резервацию?

Пассажир: **Григорьева Лидия**

Рейс: 65

Авиакомпания: Aeroflot

Время вылета: Nov. 6, 2021

Время прилета: Nov. 6, 2021

Тип: Arrival

Гейт: 4

Место: **30**

Номер билета: **1**

Confirm

Список зарегистрированных на рейс

← → ↻ 🏠 ⓘ 127.0.0.1:8000/flight/1

Пассажир: Григорьева Лидия

Место: 30

Номер билета: 1

Пассажир: Иванов Иван

Место: 21

Номер билета: 2

Оставить отзыв

Author: Григорьева Лидия ▾

Flight: Aeroflot 65 ▾

Rating: 6 ▾

Comment: Еда невкусная|

Submit