

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Web-программирование

Отчет

Лабораторная работа №2

Реализация простого сайта средствами Django

Выполнил:
Новиков Г. В.

Группа:
К33402

Проверил:
Говоров А. И.

Санкт-Петербург

2022 г.

Цель работы

Овладеть практическими навыками и умениями реализации web-сервисов средствами Django 2.2.

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.

Выполнение работы

Веб-сервис имеет такие представления как: `RegistrationView` (для регистрации), `FlightListView` (для списка рейсов), `FlightDetailView` (для просмотра подробной информации о рейсе), `BookingListView` (для просмотра списка броней), `BookingCreateView` (для создания брони), `BookingUpdateView` (для обновления брони).

```

9      # View for sign up
10     class RegistrationView(FormView):
11         form_class = UserCreationForm
12         template_name = "users/sign_up.html"
13         success_url = "/"
14
15     def form_valid(self, form):
16         form.save()
17         username = form.cleaned_data.get("username")
18         raw_password = form.cleaned_data.get("password1")
19         user = authenticate(username=username, password=raw_password)
20         login(self.request, user)
21         return super().form_valid(form)
22
23
24     # View for Table
25     class FlightListView(ListView):
26         template_name = "flights/flight_list.html"
27         model = Flight
28
29
30     # View for Flight details
31     class FlightDetailView(DetailView):
32         template_name = "flights/flight_detail.html"
33         model = Flight
34
35
36     # View for Booking
37     class BookingListView(LoginRequiredMixin, ListView):
38         template_name = "booking/booking_list.html"
39         model = Booking
40         login_url = "/signin"
41
42     def get_queryset(self):
43         return Booking.objects.filter(user=self.request.user)
44
45

```

```

46     # View for add Booking
47     class BookingCreateView(LoginRequiredMixin, CreateView):
48         template_name = "booking/booking_add.html"
49         model = Booking
50         fields = ["flight", "seats"]
51         success_url = "/booking_list"
52         login_url = "/signin"
53
54     def get_initial(self):
55         initial = super().get_initial()
56         if "flight" in self.request.GET:
57             initial["flight"] = self.request.GET.get("flight")
58         return initial
59
60     def form_valid(self, form):
61         form.instance.user = self.request.user
62         return super().form_valid(form)
63
64
65     # View for update Booking
66     class BookingUpdateView(LoginRequiredMixin, UpdateView):
67         template_name = "booking/booking_update.html"
68         model = Booking
69         fields = ["review_text", "review_number"]
70         success_url = "/booking_list"
71         login_url = "/signin"
72
73     def get_queryset(self):
74         return Booking.objects.filter(user=self.request.user)
75

```

Также есть 3 модели данных: Company (авиакомпания), Flight (рейс) и Booking (бронирование).

```
6 # Company model
7 class Company(models.Model):
8     class Meta:
9         verbose_name = "Company"
10        verbose_name_plural = "Companies"
11
12        name = models.CharField("Name", max_length=255, unique=True)
13
14    def __str__(self):
15        return str(self.name)
16
17
18 # Flight model
19 class Flight(models.Model):
20     class Meta:
21         verbose_name = "Flight"
22         verbose_name_plural = "Flights"
23
24     class TypeChoices(models.IntegerChoices):
25         DEPARTING = 0, "Start"
26         ARRIVING = 1, "End"
27
28        number = models.CharField("N", max_length=255, unique=True)
29        from_city = models.CharField("From city", max_length=255)
30        to_city = models.CharField("To city", max_length=255)
31        departure_time = models.DateTimeField("Start time")
32        arrival_time = models.DateTimeField("End time")
33        type = models.IntegerField("Start/end", choices=TypeChoices.choices)
34        gate = models.CharField("Gate", max_length=255)
35        company = models.ForeignKey(Company, verbose_name="Company", on_delete=models.CASCADE,
36                                   related_name="flights")
37
38    def __str__(self):
39        return str(self.number)
```

```
42 # Booking model
43 class Booking(models.Model):
44     class Meta:
45         verbose_name = "Booking"
46         verbose_name_plural = "Bookings"
47
48        flight = models.ForeignKey(Flight, verbose_name="Flight", on_delete=models.CASCADE, related_name="bookings")
49        user = models.ForeignKey(User, verbose_name="User", on_delete=models.CASCADE, related_name="bookings")
50        seats = models.IntegerField("Count", validators=[MinValueValidator(1)])
51        review_text = models.TextField("Review", default="")
52        review_number = models.IntegerField("Score", null=True, validators=[MinValueValidator(1), MaxValueValidator(10)])
53
54    def __str__(self):
55        return str(self.id)
```

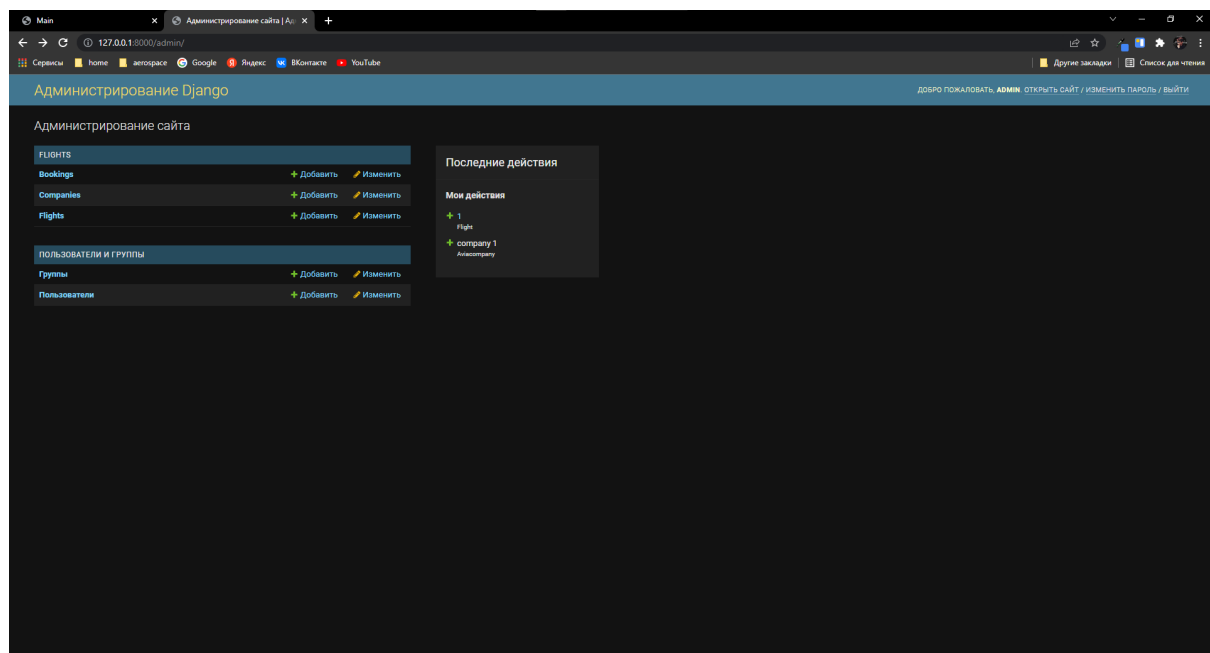
Список роутеров:

```

11 urlpatterns = [
12     path("signin/", LoginView.as_view(next_page="/", template_name="users/sign_in.html")),
13     path("signup/", RegistrationView.as_view()),
14     path("logout/", LogoutView.as_view(next_page="/")),
15     path('admin/', admin.site.urls),
16     path("", TemplateView.as_view(template_name="index.html")),
17     path("", include("flights.urls")),
18     path("", include("users.urls")),
19     path("flight_list/", FlightListView.as_view()),
20     path("flight_detail/<int:pk>/", FlightDetailView.as_view()),
21     path("booking_create/", BookingCreateView.as_view()),
22     path("booking_update/<int:pk>/", BookingUpdateView.as_view()),
23     path("booking_list/", BookingListView.as_view())
24 ]

```

Панель администратора в Django:



Список полётов:

| No | Company | From city | Start time | To city | End time | Type | Gate |
|----|-----------|-----------|------------------------|---------|------------------------|-------|-------------------------|
| 1 | company 1 | A | 19 марта 2022 г. 12:13 | B | 19 марта 2022 г. 12:13 | Start | Details |
| 2 | company 1 | B | 19 марта 2022 г. 12:14 | D | 19 марта 2022 г. 12:14 | Start | Details |

Вывод

В результате выполненной работы было освоена разработка веб-приложений с помощью фреймворка Django (создание моделей, представлений, роутеров). Также была использована встроенная admin панель Django.