

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий»

Направление подготовки 09.03.03 «Мобильные и сетевые технологии»

**ОТЧЕТ**

---

**Тема задания:** Реализация простого сайта средствами DJANGO

---

**Выполнил:**

Студент: Ли В.Э., К33402

(ФИО, номер группы)

**Проверил:**

Преподаватель: Говоров А.И.

(ФИО)

Санкт-Петербург  
2020

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

В файле models.py содержатся модели Авиакомпании, Рейса, Места и Комментария (рис 1) и (рис 2).

### models.py

```
from django.db import models
from django.contrib.auth.models import User
from datetime import date

class Airlines(models.Model):
    name = models.CharField(max_length=30)

    def __str__(self):
        return "{}".format(self.name)

class Flight(models.Model):
    company = models.ForeignKey(Airlines, on_delete=models.CASCADE)
    num_flight = models.IntegerField()
    departure = models.TextField(max_length=200, verbose_name='Город, аэропорт') # отлет
    Arrival = models.TextField(max_length=200, verbose_name='Город, аэропорт') # прилет
    NumberOfPackages = models.IntegerField()
    departure_date = models.DateTimeField(null=True)
    Arrival_date = models.DateTimeField(null=True)

    def __str__(self):
        return "{} -> {}".format(self.departure, self.departure)

    @property
    def is_past_due(self):
        return date.today() > self.departure_date

    def count(self):
        return range(self.NumberOfPackages)
```

Рисунок 1

```
class Place(models.Model):
    num_flight = models.ForeignKey(Flight, on_delete=models.CASCADE, verbose_name='Рейс', blank=True,
                                   related_name='numberFlight')
    passenger = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Пассажир', blank=True,
                                   null=True)

    def save(self, *args, **kwargs):
        if self.num_flight.numberFlight.all().count() >= self.num_flight.NumberOfPackages:
            return
        super(Place, self).save(*args, **kwargs)

    def __str__(self):
        return "{} -> {}".format(self.num_flight, self.passenger)

class Comments(models.Model):
    flight = models.ForeignKey(Flight, on_delete=models.CASCADE, verbose_name='Рейс', blank=True, null=True,
                               related_name='Flight')
    author = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Владелец комментария', blank=True,
                               null=True)
    create_date = models.DateTimeField(auto_now=True)
    Rating = models.IntegerField()
    text = models.TextField(verbose_name='Текст комментария')
    status = models.BooleanField(verbose_name='Видимость комментария', default=True)

    def __str__(self):
        return "{} -> {}".format(self.flight, self.author)
```

Рисунок 2

## urls.py

```
from django.contrib import admin
from django.urls import path
from .views import *

urlpatterns = [
    path('', HomeListView.as_view(), name='home'),
    path('detailFuture/<int:pk>', HomeDetailViewFuture.as_view(), name='detailFuture_page'),
    path('delete_comment/<int:id>', delete_comment, name='delete_comment'),
    path('lastflight', LastFlightView.as_view(), name='flight_page'),
    path('detailPast/<int:pk>', HomeDetailViewPast.as_view(), name='detailPast_page'),
    path('login', MyprojectLoginView.as_view(), name='login_page'),
    path('logout', MyprojectLogOutView.as_view(), name='logout_page'),
    path('user', UserPage.as_view(), name='user_page'),
    path('regiser', RegisterUserView.as_view(), name='register_page'),
]
```

Рисунок 3

## views.py

```
from django.contrib.auth import authenticate, login
from django.http import HttpResponseRedirect
from django.shortcuts import render, get_object_or_404, redirect
from django.urls import reverse_lazy, reverse
from django.views.generic import ListView, DetailView, CreateView, UpdateView, DeleteView
from django.contrib.auth.views import LoginView, LogoutView
from django.views.generic.edit import FormMixin
from django.contrib import messages
from .forms import *
from .models import *

class CustomSuccessMessageMixin:
    @property
    def success_msg(self):
        return False

    def form_valid(self, form):
        messages.success(self.request, self.success_msg)
        return super().form_valid(form)

    def get_success_url(self):
        return '%s?id=%s' % (self.success_url, self.object.id)

class HomeListView(ListView):
    model = Flight
    template_name = 'index.html'
    context_object_name = 'flights'

class HomeDetailViewFuture(CustomSuccessMessageMixin, FormMixin, DetailView):
```

Рисунок 4

```
class HomeDetailViewFuture(CustomSuccessMessageMixin, FormMixin, DetailView):
    model = Flight
    template_name = 'detail.html'
    context_object_name = 'flight'
    form_class = PlaceForm
    success_msg = 'вы зарезервированы'

    def get_success_url(self, **kwargs):
        return reverse_lazy('detailFuture_page', kwargs={'pk': self.get_object().id})

    def post(self, request, *args, **kwargs):
        form = self.get_form()
        if form.is_valid():
            return self.form_valid(form)
        else:
            return self.form_invalid(form)

    def form_valid(self, form):
        self.object = form.save(commit=False)
        self.object.num_flight = self.get_object()
        self.object.passenger = self.request.user
        self.object.save()
        return super().form_valid(form)

    def delete_comment(request, id):
        selected_comment = get_object_or_404(Place, id=id)
        if request.user == selected_comment.passenger:
            selected_comment.delete()
        return redirect('home')
```

Рисунок 5

```

64 class HomeDetailViewPast(CustomSuccessMessageMixin, FormMixin, DetailView):
65     model = Flight
66     template_name = 'detailPast.html'
67     context_object_name = 'flight'
68
69     form_class = CommentForm
70     success_msg = 'Коммент опубликован!'
71
72     def get_success_url(self, **kwargs):
73         return reverse_lazy('detailPast_page', kwargs={'pk': self.get_object().id})
74
75     def post(self, request, *args, **kwargs):
76         form = self.get_form()
77         if form.is_valid():
78             return self.form_valid(form)
79         else:
80             return self.form_invalid(form)
81
82     def form_valid(self, form):
83         self.object = form.save(commit=False)
84         self.object.flight = self.get_object()
85         self.object.author = self.request.user
86         self.object.save()
87         return super().form_valid(form)
88
89 class LastFlightView(ListView):
90     model = Flight
91     template_name = 'lastFlights.html'
92     context_object_name = 'flights'
93
94

```

Рисунок 5

```

class MyprojectLoginView(LoginView):
    template_name = 'login.html'
    form_class = AuthUserForm
    success_url = reverse_lazy('home')

    def get_success_url(self):
        return self.success_url

class MyprojectLogOutView(LogoutView):
    next_page = reverse_lazy('home')

class UserPage(ListView):
    model = User
    template_name = 'userPage.html'
    context_object_name = 'user'

class RegisterUserView(CreateView):
    model = User
    template_name = 'register_page.html'
    form_class = RegisterUserForm
    success_url = reverse_lazy('home')
    success_msg = 'Пользователь создан'

    def form_valid(self, form):
        form_valid = super().form_valid(form)
        username = form.cleaned_data["username"]
        password = form.cleaned_data["password"]
        aut_user = authenticate(username=username, password=password)
        login(self.request, aut_user)
        return form_valid

```

Рисунок 6

## forms.py

```

class CommentForm(forms.ModelForm):
    class Meta:
        model = Comments
        fields = ('text', 'Rating',)

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for field in self.fields:
            self.fields[field].widget.attrs['class'] = 'form-control'

class PlaceForm(forms.ModelForm):
    class Meta:
        model = Place
        fields = ('passenger',)

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for field in self.fields:
            self.fields[field].widget.attrs['class'] = 'form-control'

```

Рисунок 7

```

class AuthUserForm(AuthenticationForm, forms.ModelForm):
    class Meta:
        model = User
        fields = ('username', 'password')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for field in self.fields:
            self.fields[field].widget.attrs['class'] = 'form-control'

class RegisterUserForm(forms.ModelForm):
    class Meta:
        model = User
        fields = ('username', 'password')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for field in self.fields:
            self.fields[field].widget.attrs['class'] = 'form-control'

    def save(self, commit=True):
        user = super().save(commit=False)
        user.set_password(self.cleaned_data["password"])
        if commit:
            user.save()
        return user

```

Рисунок 8

## Страница: Будущие рейсы

На располагаются рейсы, дата которых больше сегодняшней.

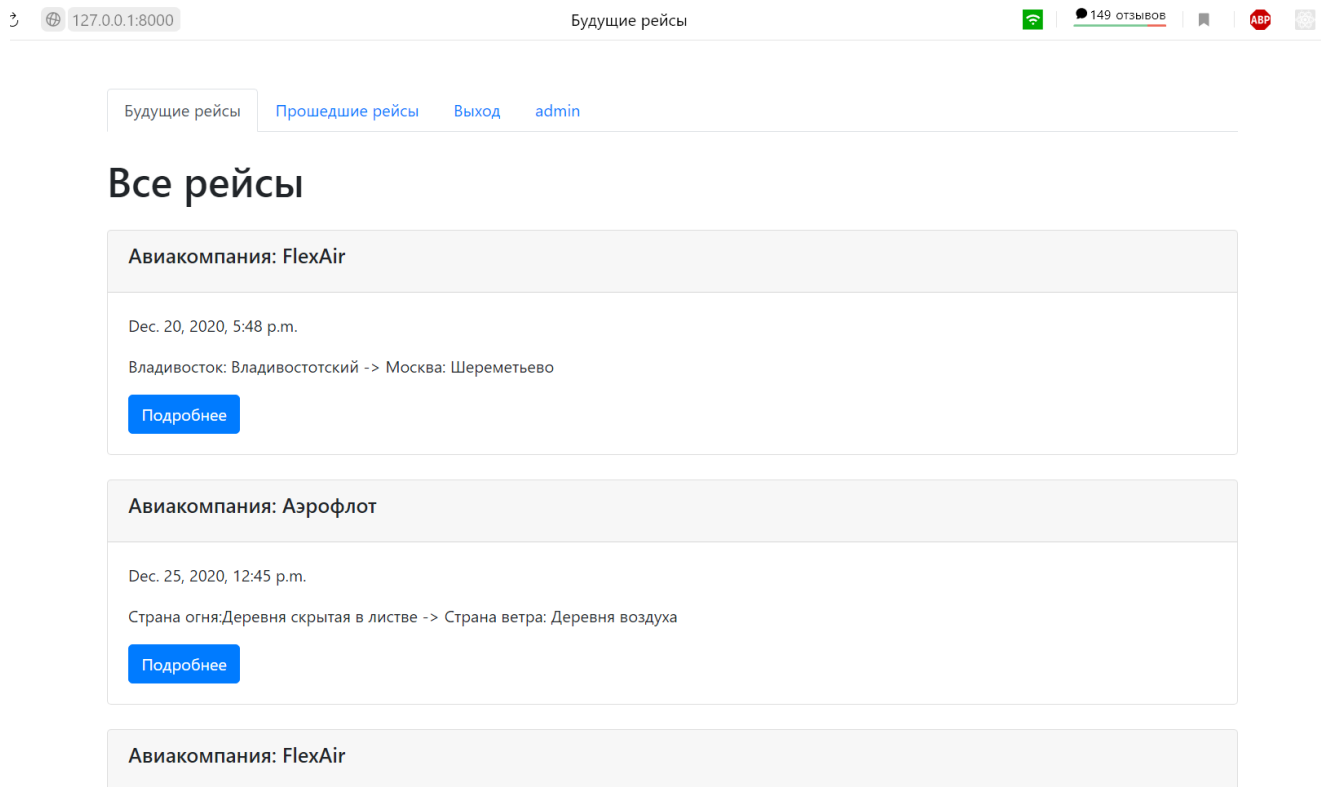


Рисунок 9

Нажав на кнопку “подробнее” пользователь попадает на страницу конкретного рейса, где может зарезервировать себе место, или удалить существующее свое.

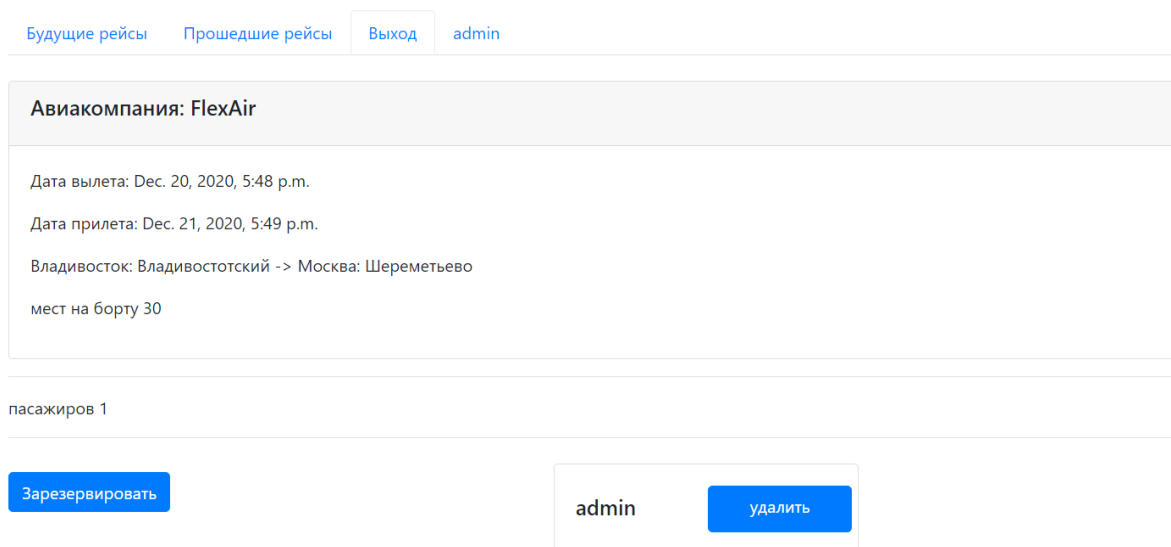


Рисунок 10

В случае, если вошел другой пользователь, то он сможет увидеть, сколько мест зарезервировано. И если зарезервировано мест меньше, чем есть на борту, то он также сможешь зарезервировать себе место. Пользователь может удалять только свои резервации.

[Будущие рейсы](#) [Прошедшие рейсы](#) [Выход](#) [jabka](#)

Авиакомпания: FlexAir

Дата вылета: Dec. 20, 2020, 5:48 p.m.  
Дата прилета: Dec. 21, 2020, 5:49 p.m.  
Владивосток: Владивостотский -> Москва: Шереметьево  
мест на борту 30

пассажиров 2

Зарезервировать

admin

jabka

удалить

Рисунок 11

Страница: “прошедшие рейсы”

[Будущие рейсы](#) [Прошедшие рейсы](#) [Выход](#) [jabka](#)

## Все рейсы

Авиакомпания: FlexAir

Dec. 2, 2020, 5:49 p.m.  
Москва: Шереметьево -> Владивосток: Владивостотский  

Подробнее

Рисунок 12

На конкретном рейсе пользователь может оставить комментарий:

Дата прилета: Dec. 3, 2020, 5:49 p.m.

Москва: Шереметьево -> Владивосток: Владивостотский

Комментариев 2

вы зарезервированны вы зарезервированны вы зарезервированны вы зарезервированны вы зарезервированны вы зарезервированны вы зарезервированны вы зарезервированны

Текст комментария:

Rating:

Добавить

admin - Dec. 4, 2020, 5:56 p.m. - рейтинг: 1  
121ц

admin - Dec. 4, 2020, 5:57 p.m. - рейтинг: 1  
ds

Рисунок 13

Из учетной записи можно также выйти, нажав кнопку “Выйти”, правее располагается имя пользователя.

Если пользователь не авторизован, то вместо кнопки “выход” и имени пользователя есть две другие кнопки, которые переводят на страницу входа или регистрации.

127.0.0.1:8000

Будущие рейсы

Будущие рейсы

Прошедшие рейсы

Вход

Регистрация

Рисунок 14

Будущие рейсы

Прошедшие рейсы

Вход

Регистрация

Авторизация

Вернуться назад

Username:

abka

Password:

.....

войти

Рисунок 15

### **Выводы**

В ходе проделанной лабораторной работы были получены навыки умения и реализации web-сервисов средствами Django