

НИУ ИТМО
ФАКУЛЬТЕТ ИКТ

Основы web-программирования

Отчет

по лабораторной работе №1

«Реализация web-сервисов средствами Python»

Выполнила:

Суздальцева Маргарита

Студентка группы К3343

Проверил:

Говоров Антон Игоревич

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 *.

Задание

Вариант 3. Табло отображения информации об авиа перелётах

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

Необходимо реализовать регистрацию новых пользователей, с целью дальнейшего добавления ими комментариев к информации об авиа перелете. При добавлении комментариев должны сохраняться информация об комментаторе, тип комментария (информация о задержке, изменении гейта или иное).

Ход работы

Спроектированная БД:

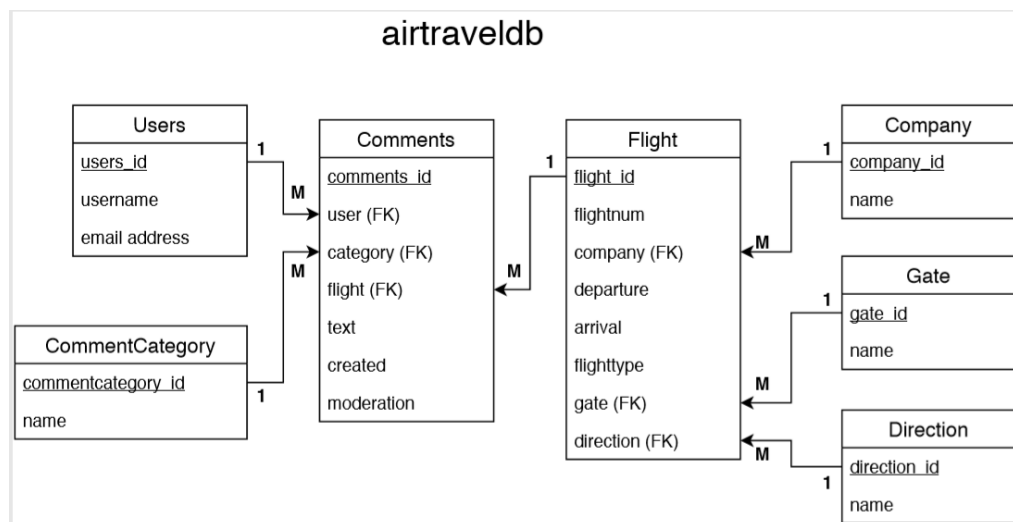


Рисунок 1 – Модель данных airtraveldb

Были созданы проект Django *airtravel* и приложение *flightsapp*. К проекту подключена БД PostgreSQL *airtraveldb* при помощи модуля *psycopg2*.

Модели Company (Авиакомпания), Gate (Гейт), Direction (Направление), Flight (Авиаперелёт), CommentCategory (Категория комментария), Comments (Комментарии) описаны в models.py.

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

class Company(models.Model):
    """Класс авиакомпаний"""
    name = models.CharField("Название", max_length=50)

    class Meta:
        verbose_name = "Авиакомпания"
        verbose_name_plural = "Авиакомпании"

    def __str__(self):
        return self.name

class Direction(models.Model):
    """Класс направлений рейсов"""
    name = models.CharField("Название", max_length=50)

    class Meta:
        verbose_name = "Направление"
        verbose_name_plural = "Направления"

    def __str__(self):
        return self.name

class Gate(models.Model):
    """Класс гейтов"""
    name = models.CharField("Название", max_length=50)

    class Meta:
        verbose_name = "Гейт"
        verbose_name_plural = "Гейты"

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

Рисунок 2 – Код models.py (часть 1)

```

class Flight(models.Model):
    """Класс авиа перелётов"""
    flightnum = models.CharField("Рейс", max_length=7)
    company = models.ForeignKey(
        Company,
        verbose_name = "Компания",
        on_delete = models.CASCADE)
    departure = models.DateTimeField("Отлёт", default=timezone.now, auto_now=False, auto_now_add=False)
    arrival = models.DateTimeField("Прилёт", default=timezone.now, auto_now=False, auto_now_add=False)
    TYPES = (
        ('D', 'отлёт'),
        ('A', 'прилёт'),
    )
    flighttype = models.CharField("Тип", max_length=1, choices=TYPES)
    gate = models.ForeignKey(
        Gate,
        verbose_name = "Гейт",
        on_delete = models.SET_NULL,
        null=True)
    direction = models.ForeignKey(
        Direction,
        verbose_name = "Направление",
        on_delete = models.CASCADE)

class Meta:
    verbose_name = "Авиа перелёт"
    verbose_name_plural = "Авиа перелёты"

def __str__(self):
    return self.flightnum

```

Рисунок 3 – Код models.py (часть 2)

```

class CommentCategory(models.Model):
    """Класс категорий комментариев"""
    name = models.CharField(max_length=50)

class Meta:
    verbose_name = "Категория"
    verbose_name_plural = "Категории"

def __str__(self):
    return self.name

class Comments(models.Model):
    """Класс комментариев"""
    user = models.ForeignKey(
        User,
        verbose_name = "Пользователь",
        on_delete = models.CASCADE)
    category = models.ForeignKey(
        CommentCategory,
        verbose_name = "Категория",
        on_delete = models.SET_NULL, null=True)
    flight = models.ForeignKey(
        Flight,
        verbose_name = "Авиа перелёт",
        on_delete = models.CASCADE)
    text = models.TextField("Комментарий")
    created = models.DateTimeField("Дата добавления", auto_now_add=True, null=True)
    moderation = models.BooleanField("Модерация", default=False)

class Meta:
    verbose_name = "Комментарий"
    verbose_name_plural = "Комментарии"

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

```

Рисунок 4 – Код models.py (часть 3)

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

Все модели были зарегистрированы в `admin.py`, через админку в БД добавлены несколько примеров рейсов.

Регистрация, авторизация пользователей для последующего комментирования реализованы с помощью модуля *django-allauth*. Данные о пользователях хранятся в модели `Users`.

В templates были созданы **шаблоны** `flightslist.html`, `flightsingle.html`.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <h1>Информация об авиа перелётах</h1>
  <a href="{% url 'flights_type' pk=1 %}"><button>Все рейсы</button></a>
  <a href="{% url 'flights_type' pk=2 %}"><button>Отлёты</button></a>
  <a href="{% url 'flights_type' pk=3 %}"><button>Прилёты</button></a><br>
  <hr/>
  {% for flight in flights %}
    <b>{{ flight.flightnum }}</b> |
    {{ flight.company }} |
    {{ flight.departure }} |
    {{ flight.arrival }} |
    <b>{{ flight.get_flighttype_display }}</b> |
    GATE {{ flight.gate }} |
    {{ flight.direction }} |
    <a href="{% url 'flight_single' pk=flight.id %}">Подробнее</a></li>
  <hr/>
  {% empty %}
    Нет информации о перелётах!
  {% endfor %}
</body>
</html>
```

Рисунок 5 – Шаблон `flightslist.html`

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <a href="{% url 'flights_list' %}">Все рейсы</a>
  <h2>Информация о рейсе</h2>
  <ul style="list-style-type:none">
    <li><b>№ рейса:</b> {{ flight.flightnum }}</li>
    <li><b>Авиакомпания:</b> {{ flight.company }}</li>
    <li><b>Отлёт:</b> {{ flight.departure }}</li>
    <li><b>Прилёт:</b> {{ flight.arrival }}</li>
    <li><b>Тип:</b> {{ flight.get_flighttype_display }}</li>
    <li><b>№ гейта:</b> {{ flight.gate }}</li>
    <li><b>Направление:</b> {{ flight.direction }}</li>
  </ul>
  <hr/>
  <h4>Комментарии</h4>
  {% for comment in comments %}
  <div>
    Пользователь: {{ comment.user }}<br>
    {{ comment.text }}<br>
    Категория: {{ comment.category }}<br>
    Добавлен {{ comment.created }}<br><br>
  </div>
  {% endfor %}
  {% if user.is_active %}
    <form action="" method="post">
      {% csrf_token %}
      {{ form.as_p }}
      <button type="submit">Отправить</button>
    </form><br>
  <a href="/accounts/logout/">Выйти</a> из аккаунта
  {% else %}
    Чтобы оставить комментарий - <a href="/accounts/login/">авторизуйтесь</a>.
  {% endif %}
</body>
</html>

```

Рисунок 6 – Шаблон flightsingle.html

Во views.py реализованы следующие **представления**:

flights_list – функционально выводит список всех рейсов, использует шаблон flightlist.html

flights_type – данная функция фильтрует рейсы в зависимости от типа (делит на отлёты и прилёты), также использует шаблон flightlist.html. Отлёты сортируются по времени отлёта, прилёты – по времени прилёта.

flight_single – функционально выводит информацию о конкретном рейсе, комментарии к нему, форму для комментирования (задана в forms.py).

```

from django.shortcuts import render, get_object_or_404, redirect
from django.http import Http404, HttpResponse
from .models import Company, Direction, Gate, Flight, Comments
from django.views.generic.list import ListView
from .forms import CommentForm

def flights_list(request):
    """Список всех рейсов"""
    flights = Flight.objects.all().order_by('arrival')
    return render(request, "flightslist.html", {"flights": flights})

def flights_type(request, pk):
    """Фильтр рейсов в зависимости от типа (разделение на отлёты и прилёты)"""
    flights = Flight.objects.all()
    if pk == 1:
        flights = flights.order_by('arrival')
    elif pk == 2:
        flights = flights.filter(flighttype='D').order_by('departure')
    elif pk == 3:
        flights = flights.filter(flighttype='A').order_by('arrival')
    return render(request, "flightslist.html", {"flights": flights})

def flight_single(request, pk):
    """Вывод информации о конкретном рейсе"""
    flight = get_object_or_404(Flight, id=pk)
    comment = Comments.objects.filter(flight=pk, moderation=True)
    if request.method == "POST":
        form = CommentForm(request.POST)
        if form.is_valid():
            form = form.save(commit=False)
            form.user = request.user
            form.flight = flight
            form.save()
            return redirect(flight_single, pk)
    else:
        form = CommentForm()
    return render(request, "flightsingle.html", {"flight": flight, "comments": comment, "form": form})

```

Рисунок 7 – Код views.py

```

from django.forms import ModelForm
from .models import Comments

class CommentForm(ModelForm):
    class Meta:
        model = Comments
        fields = [
            "text",
            "category"
        ]

```

Рисунок 8 – Код forms.py

Адресация настраивается в urls.py. Urls приложения:

```

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

urlpatterns = [
    path('', views.flights_list, name="flights_list"),
    path('single/<int:pk>', views.flight_single, name="flight_single"),
    path('filter/<int:pk>', views.flights_type, name="flights_type"),
]

```

Рисунок 9 – Код flightsapp/urls.py

Urls проекта:

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

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

Рисунок 10 – Код airtravel/urls.py

Так выглядит основная страница, а то есть табло отображения информации о рейсах аэропорта:



Рисунок 11 – Информация об авиа перелётах

При нажатии кнопки «Отлёты»:

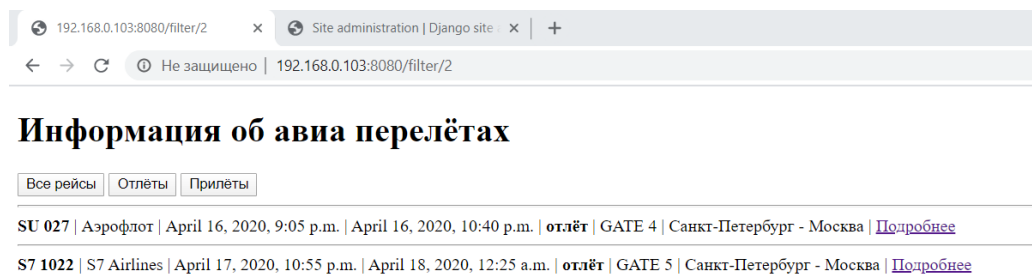
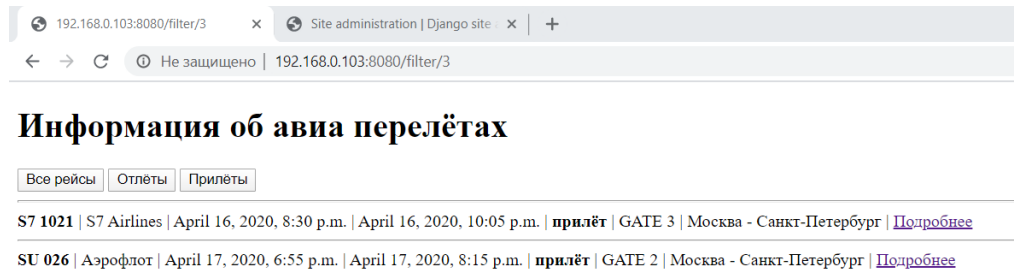


Рисунок 12 – Таблo отлётов

Аналогично с прилётами:



Информация об авиа перелётах									
<div>Все рейсыОтлётыПрилёты</div>									
S7 1021 S7 Airlines April 16, 2020, 8:30 p.m. April 16, 2020, 10:05 p.m. прилёт GATE 3 Москва - Санкт-Петербург Подробнее									
SU 026 Аэрофлот April 17, 2020, 6:55 p.m. April 17, 2020, 8:15 p.m. прилёт GATE 2 Москва - Санкт-Петербург Подробнее									

Рисунок 13 – Табло прилётов

При нажатии кнопки «подробнее» происходит переход на страницу информации о конкретном полёте. Дублируется информация с основной страницы списком, под ней расположены оставленные комментарии. Здесь же находится форма, в которой пользователь может оставить комментарий (есть выбор типа комментария – задержка рейса, изменение гейта, другое). Отправка формы доступна, если пользователь авторизован. Если нет – ему предложат авторизоваться. Комментарии проходят модерацию в админ-панели и отображаются только если `moderation=True`.

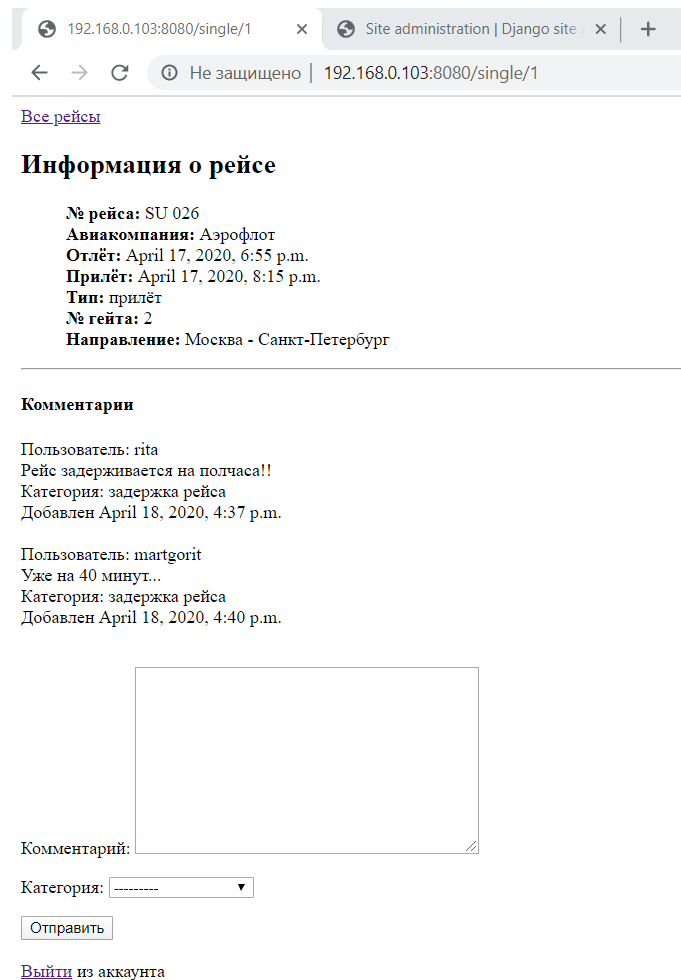


Рисунок 14 – Информация о рейсе, комментарии, форма

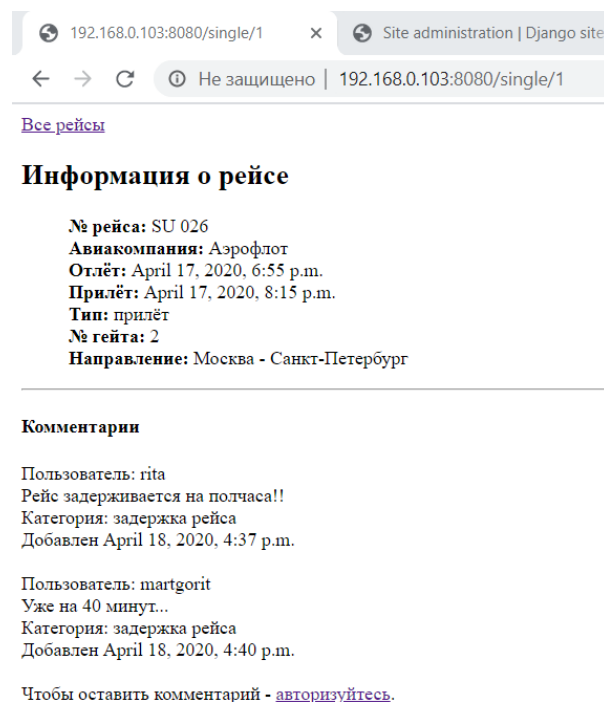


Рисунок 15 – Так страница выглядит для неавторизованного пользователя

Так выглядит страница авторизации:

Sign In

Site administration | Django site

← → ↻ Не защищено | 192.168.0.103:8080/accounts/login/

Messages:

- You have signed out.

Menu:

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

Sign In

If you have not created an account yet, then please [sign up](#) first.

Username:

Password:

Remember Me: ☐

[Forgot Password?](#)

Рисунок 16 – Авторизация

Админ-панель со всеми моделями:

192.168.0.103:8080

Site administration | Django site

← → ↻ Не защищено | 192.168.0.103:8080/admin/

Site administration

ACCOUNTS

Email addresses	+ Add	Change
-----------------	-------	--------

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

FLIGHTSAPP

Авиа перелёты	+ Add	Change
Авиакомпании	+ Add	Change
Гейты	+ Add	Change
Категории	+ Add	Change
Комментарии	+ Add	Change
Направления	+ Add	Change

SITES

Sites	+ Add	Change
-------	-------	--------

SOCIAL ACCOUNTS

Social accounts	+ Add	Change
Social application tokens	+ Add	Change
Social applications	+ Add	Change

Recent actions

My actions

- rita Комментарий
- SU 026 Авиа перелёт
- martgorit Комментарий
- rita Комментарий
- + другое Категория
- + изменение гейта Категория
- + задержка рейса Категория
- + SU 027 Авиа перелёт
- + S7 1021 Авиа перелёт
- + S7 1022 Авиа перелёт

Рисунок 17 – Панель администрирования