Weather Api

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('',views.index,name='home'),
    path('delete/<city_name>/',views.delete_city,
name='delete_city')
]
```

Views.py

```
import requests
from django.shortcuts import render, redirect
from .models import City
from .forms import CityForm

def index(request):
    url =
'http://api.openweathermap.org/data/2.5/weather?q={}&
units=imperial&appid=271d1234d3f497eed5b1d80a07b3fcd1
'

    err_msg = ''
    message = ''
    message_class = ''

    if request.method == 'POST':
        form = CityForm(request.POST)

        if form.is_valid():
            new_city = form.cleaned_data['name']
            existing_city_count =
City.objects.filter(name=new_city).count()

            if existing_city_count == 0:
                r =
requests.get(url.format(new_city)).json()
```

```
                if r['cod'] == 200:
                    form.save()
                else:
                    err_msg = 'City does not exist in
the world!'
            else:
                err_msg = 'City already exists in the
database!'

        if err_msg:
            message = err_msg
            message_class = 'is-danger'
        else:
            message = 'City added successfully!'
            message_class = 'is-success'

    form = CityForm()

    cities = City.objects.all()

    weather_data = []

    for city in cities:

        r = requests.get(url.format(city)).json()

        city_weather = {
            'city' : city.name,
            'temperature' : r['main']['temp'],
            'description' :
r['weather'][0]['description'],
            'icon' : r['weather'][0]['icon'],
        }

        weather_data.append(city_weather)

    context = {
        'weather_data' : weather_data,
        'form' : form,
        'message' : message,
        'message_class' : message_class
    }

    return render(request, 'weather/weather.html',
```

```
context)

def delete_city(request, city_name):
    City.objects.get(name=city_name).delete()

    return redirect('home')
```

Models.py

```
from django.db import models

class City(models.Model):
  name = models.CharField(max_length=25)


  def __str__(self):
    return self.name

  class Meta:
    verbose_name_plural = 'cities'
```

Forms.py

```
from django.forms import ModelForm, TextInput
from .models import City

class CityForm(ModelForm):
    class Meta:
        model = City
        fields = ['name']
        widgets = {'name' : TextInput(attrs={'class'
: 'input', 'placeholder' : 'City Name'})}
```

app.py

```
from django.apps import AppConfig
```

```
class WeatherConfig(AppConfig):
    name = 'weather'
```

admin.py

```
from django.contrib import admin
from .models import City

admin.site.register(City)
```

weather app

settings.py

```
"""
Django settings for weatherapp project.

Generated by 'django-admin startproject' using Django
2.2.3.

For more information on this file, see
https://docs.djangoproject.com/en/2.2/topics/settings
/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.2/ref/settings/
"""
import django_heroku
import os

# Build paths inside the project like this:
os.path.join(BASE_DIR, ...)
BASE_DIR =
os.path.dirname(os.path.dirname(os.path.abspath(__fil
e__)))


# Quick-start development settings - unsuitable for
production
# See
https://docs.djangoproject.com/en/2.2/howto/deploymen
t/checklist/
```

```
# SECURITY WARNING: keep the secret key used in
production secret!
SECRET_KEY = 'pq9u13ip%$-
!&8f*%^u$hi92b0_6@q=9t%^*iy9s#(z8_@9lqa'

# SECURITY WARNING: don't run with debug turned on in
production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'weather',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware
',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddlew
are',

'django.contrib.messages.middleware.MessageMiddleware
',

'django.middleware.clickjacking.XFrameOptionsMiddlewa
re',
]

ROOT_URLCONF = 'weatherapp.urls'
```

```
TEMPLATES = [
    {
        'BACKEND':
'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [

'django.template.context_processors.debug',

'django.template.context_processors.request',

'django.contrib.auth.context_processors.auth',

'django.contrib.messages.context_processors.messages'
,
            ],
        },
    },
]

WSGI_APPLICATION = 'weatherapp.wsgi.application'


# Database
#
https://docs.djangoproject.com/en/2.2/ref/settings/#d
atabases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}


# Password validation
#
https://docs.djangoproject.com/en/2.2/ref/settings/#a
uth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttribut
eSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengt
hValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswo
rdValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPassw
ordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/2.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-
files/

STATIC_URL = '/static/'
django_heroku.settings(locals())
```

Urls.py

```
"""weatherapp URL Configuration

The `urlpatterns` list routes URLs to views. For more
information please see:

https://docs.djangoproject.com/en/2.2/topics/http/url
s/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('',
views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import
Home
    2. Add a URL to urlpatterns:  path('',
Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from
django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/',
include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include

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

Wsgi.py

```
"""
WSGI config for weatherapp project.

It exposes the WSGI callable as a module-level
variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/2.2/howto/deploymen
t/wsgi/
```

```
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'weatherapp.settings')

application = get_wsgi_application()
```

Weather

Migrations

000_initial.py

```
# Generated by Django 2.2.3 on 2019-08-17 14:37

from django.db import migrations, models


class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='City',
            fields=[
                ('id',
models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('name',
models.CharField(max_length=25)),
            ],
            options={
                'verbose_name_plural': 'cities',
            },
```

```
        ),
    ]
```

Templates

Weather.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible"
content="ie=edge">
    <title>Weather App</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.
6.2/css/bulma.css" />
    <link rel="shortcut icon"
href="https://cdn3.iconfinder.com/data/icons/bebreeze
e-weather-symbols/691/icon-weather-sunsleetlight-
512.png" type="image/x-icon">
</head>

<body>
    <section class="hero is-primary">
        <div class="hero-body">
            <div class="container">
                <h1 class="title">
                    What's the weather like?
                </h1>
            </div>
        </div>
    </section>
    <section class="section">
        <div class="container">
            <div class="columns">
                <div class="column is-offset-4 is-4">
                    <form method="POST">
                        {% csrf_token %}
```

```
                        <div class="field has-
addons">
                            <div class="control is-
expanded">
                                {{ form.name }}
                            </div>
                            <div class="control">
                                <button type="submit"
class="button is-info">
                                    Add City
                                </button>
                            </div>
                        </div>
                        {% if message %}
                            <div class="notification
{{ message_class }}">{{ message }}<button class="del-
msg delete"
onclick="document.getElementsByClassName('notificatio
n')[0].style.display='none'"></button></div>

                        {% endif %}
                    </form>
                </div>
            </div>
        </div>
    </section>
    <section class="section">
        <div class="container">
            <div class="columns">
                <div class="column is-offset-4 is-4">
                    {% for city_weather in
weather_data %}
                    <div class="box">
                        <article class="media">
                            <div class="media-left">
                                <figure class="image
is-50x50">
                                    <img
src="http://openweathermap.org/img/w/{{
city_weather.icon }}.png" alt="Image">
                                </figure>
                            </div>
                            <div class="media-
content">
```

```
                                    <div class="content">
                                        <p>
                                            <span
class="title">{{ city_weather.city }}</span>
                                            <br>
                                            <span
class="subtitle">{{ city_weather.temperature }}°
F</span>
                                            <br> {{
city_weather.description }}
                                        </p>
                                    </div>
                                </div>
                                <div class="media-right">
                                    <a href="{% url
'delete_city' city_weather.city %}">
                                        <button
class="delete"></button>
                                    </a>
                                </div>
                            </article>
                        </div>
                        {% endfor %}
                    </div>
                </div>
            </div>
        </section>
        <footer class="footer">
        </footer>
</body>

</html>
```