

DJANGO

An overview of Django



COS216
AVINASH SINGH
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF PRETORIA

Django - Overview



Open-source, released July 2005



Developed and Maintained by the Django Software Foundation



Built using Python



Authors: Adrian Holovaty, Simon Willison



Cross Platform



Needs Python to run

HOW TO USE DJANGO?

- Firstly, you need to install Python <https://www.python.org/downloads/>
- Install Django via pip
\$ pip install Django@3.2
- Test installation
\$ python -m django --version
- Create a new project
\$ python -m django startproject covid

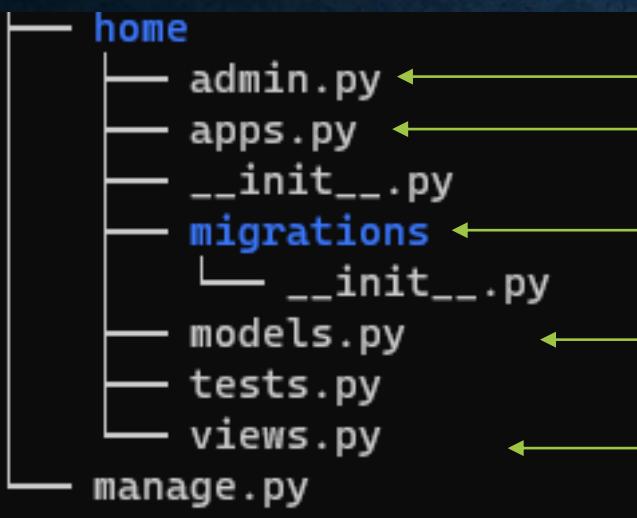
PROJECT STRUCTURE



CREATE AN APP

- Now that we have a project, we will create an app

```
$ python manage.py startapp home
```



Used for the admin view of Django
App Config file

Migration Folder for DB

Your Data models, that map to your DB

This is where the magic happens

CREATE MODELS

- Models are classes that allow for data representation and data mapping to a database
- Django makes use of models to remove the need of using SQL commands

```
from django.db import models
from datetime import datetime
# Create your models here.

class Country(models.Model):
    name = models.CharField(max_length=30)
    code = models.CharField(max_length=2)

class Cases(models.Model):
    country = models.ForeignKey(Country, null=True, on_delete=models.CASCADE)
    newConfirmed = models.IntegerField()
    totalConfirmed = models.IntegerField()
    newDeaths = models.IntegerField()
    totalDeaths = models.IntegerField()
    newRecovered = models.IntegerField()
    totalRecovered = models.IntegerField()
    timestamp = models.DateTimeField(default=datetime.now)
```

```
{
  "ID": "be8af200-1b53-437a-b013-d00749c1af5e",
  "Message": "",
  "Global": {
    "NewConfirmed": 126234,
    "TotalConfirmed": 183481426,
    "NewDeaths": 2888,
    "TotalDeaths": 3973736,
    "NewRecovered": 122333,
    "TotalRecovered": 120793551,
    "Date": "2021-07-05T06:12:18.847Z"
  },
  "Countries": [
    {
      "ID": "b6330678-a9c3-461f-bfa1-9e7924865df7",
      "Country": "Afghanistan",
      "CountryCode": "AF",
      "Slug": "afghanistan",
      "NewConfirmed": 0,
      "TotalConfirmed": 120216,
      "NewDeaths": 0,
      "TotalDeaths": 4962,
      "NewRecovered": 0,
      "TotalRecovered": 74480,
      "Date": "2021-07-05T06:12:18.847Z",
      "Premium": {}
    }
  ]
}
```

MIGRATIONS

- From the models, Django will build tables in a SQL database for you
 - \$ **python manage.py makemigrations <app_name>**
 - \$ **python manage.py migrate**
- This will then populate the migrations folder with a script of what migrations need to happen. This makes it easier to update models on the Fly, but remember with SQL and structured data, additions are easy, but restructuring and changing IDs are not.

DJANGO ADMIN

- Django Admin is fairly powerful, by default Django Admin is enabled, and is very customizable.
- The admin view allows you to manage and manually add data to your models (not recommended), however, it allows for quick management, rather than exploring the DB.
- In order to view the admin pages, an admin user is needed.

```
$ python manage.py createsuperuser
```

VIEW

home/views.py

```
from django.shortcuts import render

# Create your views here.

def stats(request):
    data = {
        'success': True,
        'message': "Hello Word!",
    }
    return render(request, 'home/index.html', data)
```

- Django views is the core of all functionality in a Django project. Each app will have its own views.py.
- The views allow you to use templates as HTML and pass data to it.
- Django follows an MTV (model-template-view) architecture

VIEW - TEMPLATES

home/templates/home/index.html

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="utf-8">  
    <title>Covid19 listings</title>  
</head>  
<body>  
    Success: {{ success }} <br/>  
    Message: {{ message }}  
  
</body>  
</html>
```

- Template folder
- Template data

URLS

home/urls.py

- In order for Django to know what to display when the urls.py file is used to declare your routes for your app.

```
# urls.py

from django.conf.urls import url
from django.urls import path, include, re_path

from home import views

urlpatterns = [
    path('', views.stats, name='stats'),
]
```

URLS

covid/urls.py

- In order for Django to know what to display when the urls.py file is used to declare your routes for your app. The main urls situated in the project_name/urls.py is the master urls, this is where you would add other app urls.

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

urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'^$', include('home.urls'))
]
```

SETTINGS - APPS

covid/settings.py

- The Django settings.py is the heart of the Django project, this is a sensitive file that stores the configurations and even passwords for the entire Django project. Now that we have created and configured the new app **home**, it now needs to be added as an installed app.

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

SETTINGS - MIDDLEWARE

covid/settings.py

- The middleware helps with a lot of security aspects, each middleware adds a layer of security that does a lot of prevention of attacks for you.

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```

SETTINGS - DATABASES

covid/settings.py

- The settings also hold the configuration to the database, for this purpose we going to use a sqlite3 database.

```
DATA BASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': str(os.path.join(BASE_DIR, "db.sqlite3")),
    }
}
```

SETTINGS - TEMPLATES

covid/settings.py

```
TEMPLATES = [
{
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': [os.path.join(BASE_DIR, 'templates')],
    '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',
        ],
    },
},
]
```

- Set the DIRs

RUN THE SERVER

- Once we have the initial set up completed, we can now run the Django Server

```
$ python manage.py runserver
```

Go to <http://127.0.0.1:8000/>

```
Success: True  
Message: Hello Word!
```

MORE FEATURES

- Now that we have the basis covered, let us dive into more features
- What we will be exploring:
 - ✓ Making use of an External API
 - ✓ Parsing the data into the template
 - ✓ Using template logic (for, if, etc.)
 - ✓ Making use of Bootstrap
 - ✓ Creating Forms
 - ✓ Saving data
 - ✓ Viewing saved data

ADDING GLOBAL STATS

```
API = "https://api.covid19api.com/"

def stats(request):
    summary = requests.get(API + "summary").json()
    summary["Global"]["Date"] = datetime.strptime(
        summary["Global"]["Date"], '%Y-%m-%dT%H:%M:%S.%fZ')
    data = {
        'success': True,
        'global': summary["Global"],
        'countries': summary["Countries"],
    }
    return render(request, 'home/index.html', data)
```

home/views.py

- Make use of the requests library to get data, and parse it as json and send it to the template

ADDING GLOBAL STATS

```
<div class="container">
    <div class="card border-primary mb-3" >
        <div class="card-header">Global Statistics <span style="float:right">{{ global.Date }}</span></div>
        <div class="card-body">
            <p class="card-text">
                <button type="button" class="btn btn-primary">
                    New Confirmed <span class="badge bg-secondary">{{ global.NewConfirmed }}</span>
                </button>
                <button type="button" class="btn btn-primary">
                    Total Confirmed <span class="badge bg-dark">{{ global.TotalConfirmed }}</span>
                </button>
                <button type="button" class="btn btn-danger">
                    New Deaths <span class="badge bg-secondary">{{ global.NewDeaths }}</span>
                </button>
                <button type="button" class="btn btn-danger">
                    Total Deaths <span class="badge bg-dark">{{ global.TotalDeaths }}</span>
                </button>
                <button type="button" class="btn btn-success">
                    New Recovered <span class="badge bg-secondary">{{ global.NewRecovered }}</span>
                </button>
                <button type="button" class="btn btn-success">
                    Total Recovered <span class="badge bg-dark">{{ global.TotalRecovered }}</span>
                </button>
            </p>
        </div>
    </div>
</div>
```

Global Statistics

July 4, 2021, 4:35 p.m.

New Confirmed 194926

Total Confirmed 183198785

New Deaths 4323

Total Deaths 3968150

New Recovered 127247

Total Recovered 120541181

RESULT

ADDING STATS PER COUNTRY

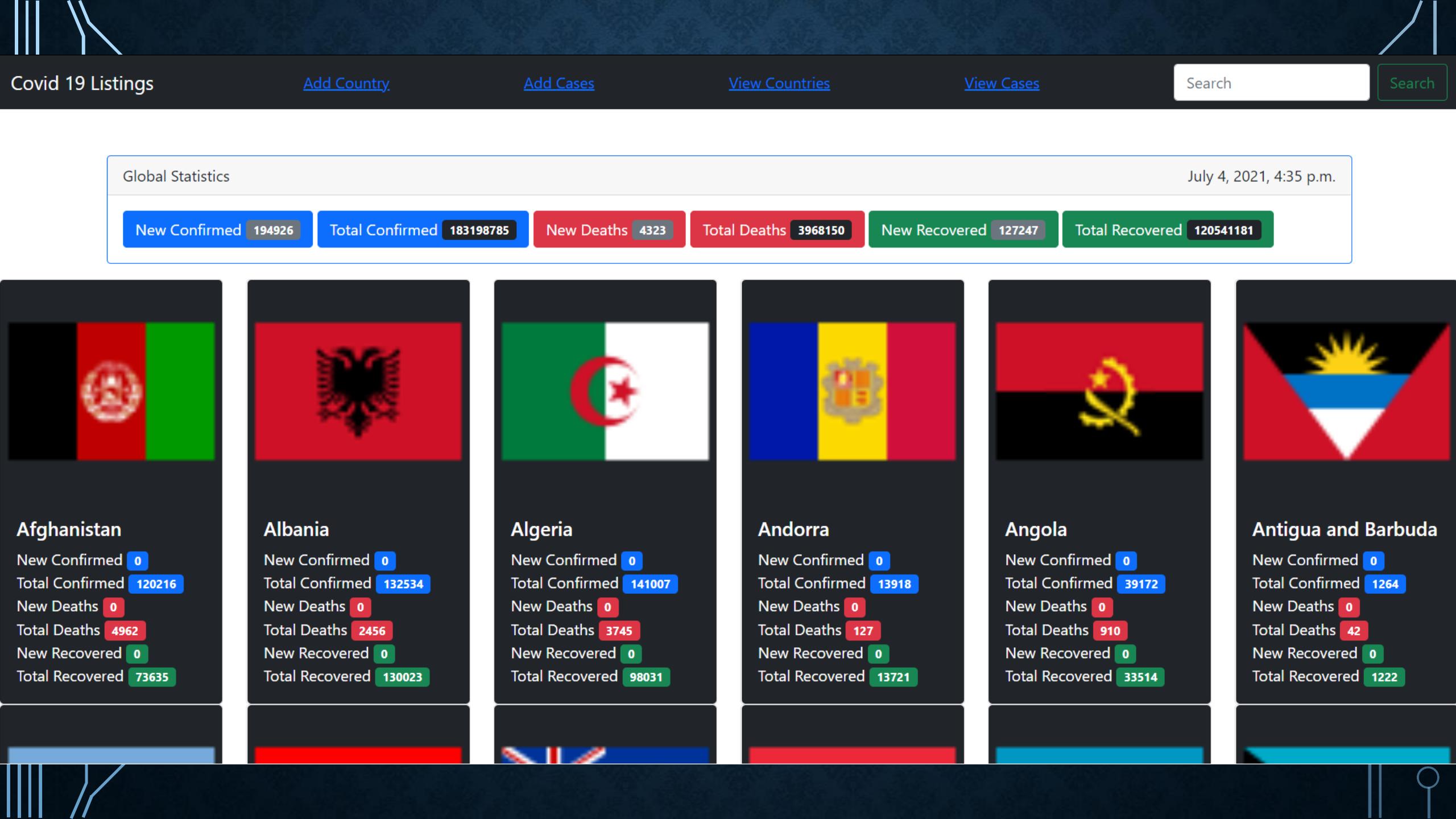
```
<nav class="navbar navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand">Covid 19 Listings</a>
    <a href="/addCountry">Add Country</a>
    <a href="/addCases">Add Cases</a>
    <a href="/viewCountries">View Countries</a>
    <a href="/viewCases">View Cases</a>
    <form class="d-flex">
      <input class="form-control me-2" id="search" type="search" placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success" type="button" >Search</button>
    </form>
  </div>
</nav>
```

ADDING STATS PER COUNTRY

```
<div class="row row-cols-6">
    {% for c in countries %}
        <div class="col" slug="{{c.Slug}}>
            <div class="card">
                <div class="card bg-dark text-white">
                    
                    <div class="card-body">
                        <h5 class="card-title">{{ c.Country }}</h5>
                        <p class="card-text">
                            New Confirmed <span class="badge bg-primary">{{ c.NewConfirmed }}</span> <br/>
                            Total Confirmed <span class="badge bg-primary">{{ c.TotalConfirmed }}</span><br/>
                            New Deaths <span class="badge bg-danger">{{ c.NewDeaths }}</span><br/>
                            Total Deaths <span class="badge bg-danger">{{ c.TotalDeaths }}</span><br/>
                            New Recovered <span class="badge bg-success">{{ c.NewRecovered }}</span><br/>
                            Total Recovered <span class="badge bg-success">{{ c.TotalRecovered }}</span>
                        </p>
                    </div>
                </div>
            </div>
        {% endfor %}
    </div>
```

ADDING STATIC SEARCH

```
<script>
    // Not a perfect search
    $("#search").keyup(function() {
        $(".col").filter(function( index ) {
            return $( this ).attr( "slug" ).indexOf($("#search").val().toLowerCase()) >= 0 ;
        }).show();
        $(".col").filter(function( index ) {
            return $( this ).attr( "slug" ).indexOf($("#search").val().toLowerCase()) == -1 ;
        }).hide();
    });
</script>
```





Covid 19 Listings

[Add Country](#)

[Add Cases](#)

[View Countries](#)

[View Cases](#)

sou



Search

Global Statistics

July 4, 2021, 4:35 p.m.

New Confirmed 194926

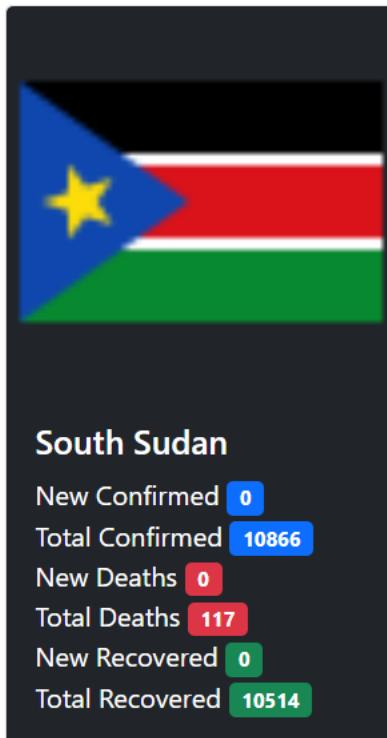
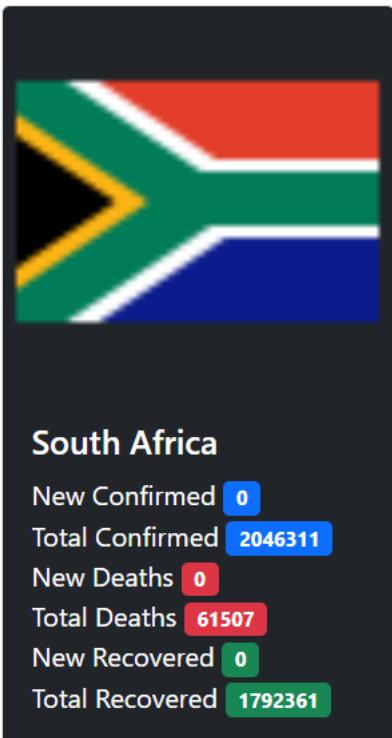
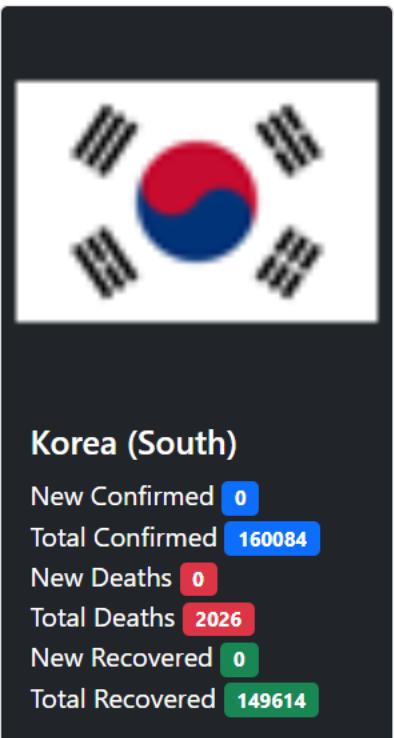
Total Confirmed 183198785

New Deaths 4323

Total Deaths 3968150

New Recovered 127247

Total Recovered 120541181



ADDING FORMS

home/forms.py

```
from django import forms
from home.models import *

class AddCountry(forms.ModelForm):
    class Meta:
        model = Country
        fields = ('name', 'code')

class AddCases(forms.ModelForm):
    class Meta:
        model = Cases
        fields = ('country', 'newConfirmed', 'totalConfirmed',
                  'newDeaths', 'totalDeaths', 'newRecovered', 'totalRecovered')
```

USING FORMS

home/views.py

```
from home.forms import *

def addCountry(request):
    message = ""
    if request.method == "POST":
        country = AddCountry(data=request.POST)
        if country.is_valid():
            # Your job, check for duplicates before saving
            country.save()
            message = "Added Country Successfully"
    data = {
        'form': AddCountry(),
        'name': "Country",
        'message': message
    }
    return render(request, 'home/add.html', data)
```

- Detect if method is POST
- Use the Form to validate data
- Use the form to save to DB

USING FORMS

```
<div class="container">
    <h2>Add {{ name }}</h2>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Submit">
    </form>
    {{ message }}
</div>
```

- Create HTML Form with POST
- **Add the CSRF prevention**
- **Let Django handle the Forms Elements**

Covid 19 Listings

Add Country

Name:

Code:

Covid 19 Listings

Add Country

Name:

Code:

Added Country Successfully

RESULT

VIEW COUNTRY

```
def viewCountries(request):
    results = Country.objects.all()
    table_columns = [{'title': "Id"}, {'title': "Name"}, {'title': "Code"}]
    table_data = []
    for r in results:
        table_data.append([r.id, r.name, r.code])
    data = {
        'name': "Countries",
        'table_columns': table_columns,
        'table_data': table_data
    }
    return render(request, 'home/view.html', data)
```

DATA TABLES

```
<div class="container">
  <h2>View {{ name }}</h2>
  <table id="data" class="display compact">
</div>

<script>
  $("#data").dataTable({
    data: {{ table_data | safe }},
    columns: {{ table_columns | safe }},
  });
</script>
```

- Uses the **datatables** JS library

SET THE URLs

```
# urls.py

from django.conf.urls import url
from django.urls import path, include, re_path

from home import views

urlpatterns = [
    path('', views.stats, name='stats'),
    path('addCountry', views.addCountry, name='addCountry'),
    path('viewCountries', views.viewCountries, name='viewCountry'),
    path('addCases', views.addCases, name='addCases'),
    path('viewCases', views.viewCases, name='viewCases')
]
```

View Countries

Show 10 entries

Id	Name	Code
1	Test	XA
2	South Africa	ZA
3	Australia	AU

Showing 1 to 3 of 3 entries

Search:

Previous

1

Next

RESULT

TESTING

- <https://docs.djangoproject.com/en/3.2/topics/testing/overview/>

```
from django.test import TestCase
from myapp.models import Animal

class AnimalTestCase(TestCase):
    def setUp(self):
        Animal.objects.create(name="lion", sound="roar")
        Animal.objects.create(name="cat", sound="meow")

    def test_animals_can_speak(self):
        """Animals that can speak are correctly identified"""
        lion = Animal.objects.get(name="lion")
        cat = Animal.objects.get(name="cat")
        self.assertEqual(lion.speak(), 'The lion says "roar"')
        self.assertEqual(cat.speak(), 'The cat says "meow"')
```

FINAL STRUCTURE

```
$ tree
covid
├── asgi.py
├── __init__.py
├── settings.py
├── urls.py
└── wsgi.py
db.sqlite3
home
├── admin.py
├── apps.py
├── forms.py
├── __init__.py
└── migrations
    ├── 0001_initial.py
    └── __init__.py
        └── pycache
            └── 0001_initial.cpython-37.pyc
            └── __init__.cpython-37.pyc
models.py
templates
└── home
    ├── add.html
    ├── index.html
    └── view.html
tests.py
urls.py
views.py
manage.py

6 directories, 22 files
```

DJANGO ADMIN

home/admin.py

```
from django.contrib import admin
from home.models import *
# Register your models here.

admin.site.register(Cases)
admin.site.register(Country)
```

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

	Add	Change
Groups		
Users		

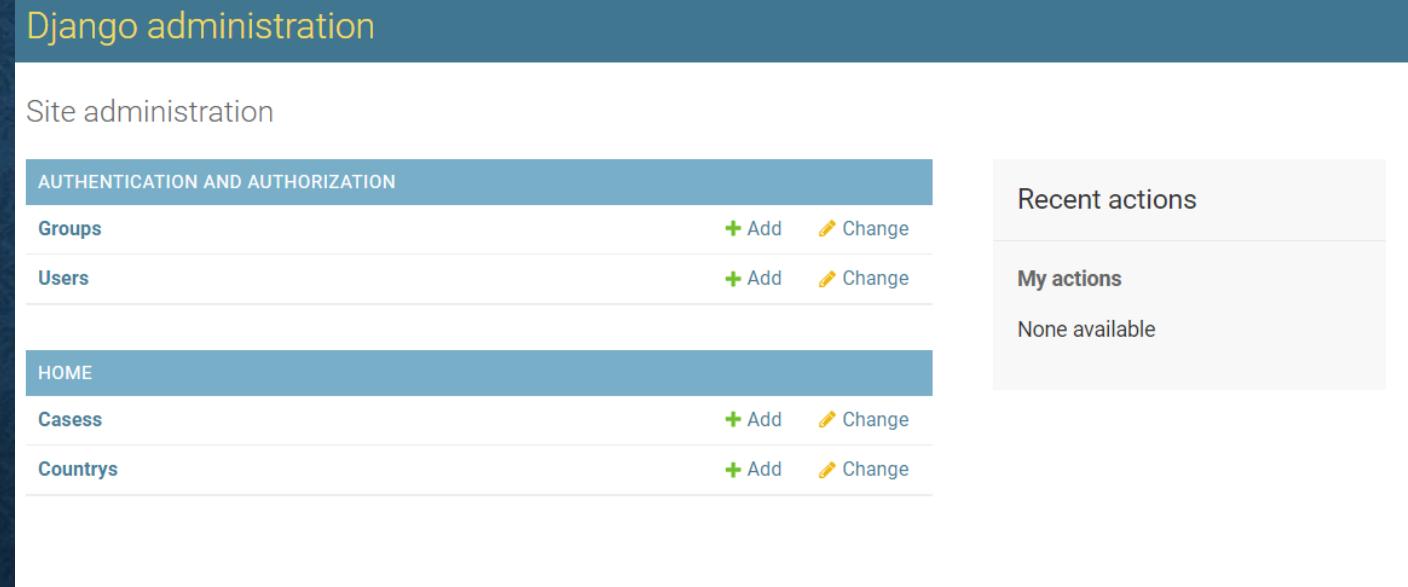
HOME

	Add	Change
Cases		
Country		

Recent actions

My actions

None available



<http://localhost:8000/admin/>

DJANGO ADMIN – MANUAL ADDITION

Django administration

WELCOME, AVINASH. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Home › Cases › Add cases

Add cases

Country:  

NewConfirmed:
South Africa
Australia

TotalConfirmed:

NewDeaths:

TotalDeaths:

NewRecovered:

TotalRecovered:

Timestamp:

Date: Today | 

Time: Now | 

Note: You are 2 hours ahead of server time.

RESOURCES

- <https://www.djangoproject.com/start/>
- <https://docs.djangoproject.com/en/3.2/>
- <https://docs.djangoproject.com/en/3.2/intro/tutorial01/>
- <https://djangopackages.org/> (**For pre-existing apps**)
- <https://datatables.net/examples/index>
- <https://getbootstrap.com/docs/5.0/components/badge/>
- <https://getbootstrap.com/docs/5.0/components/card/>



