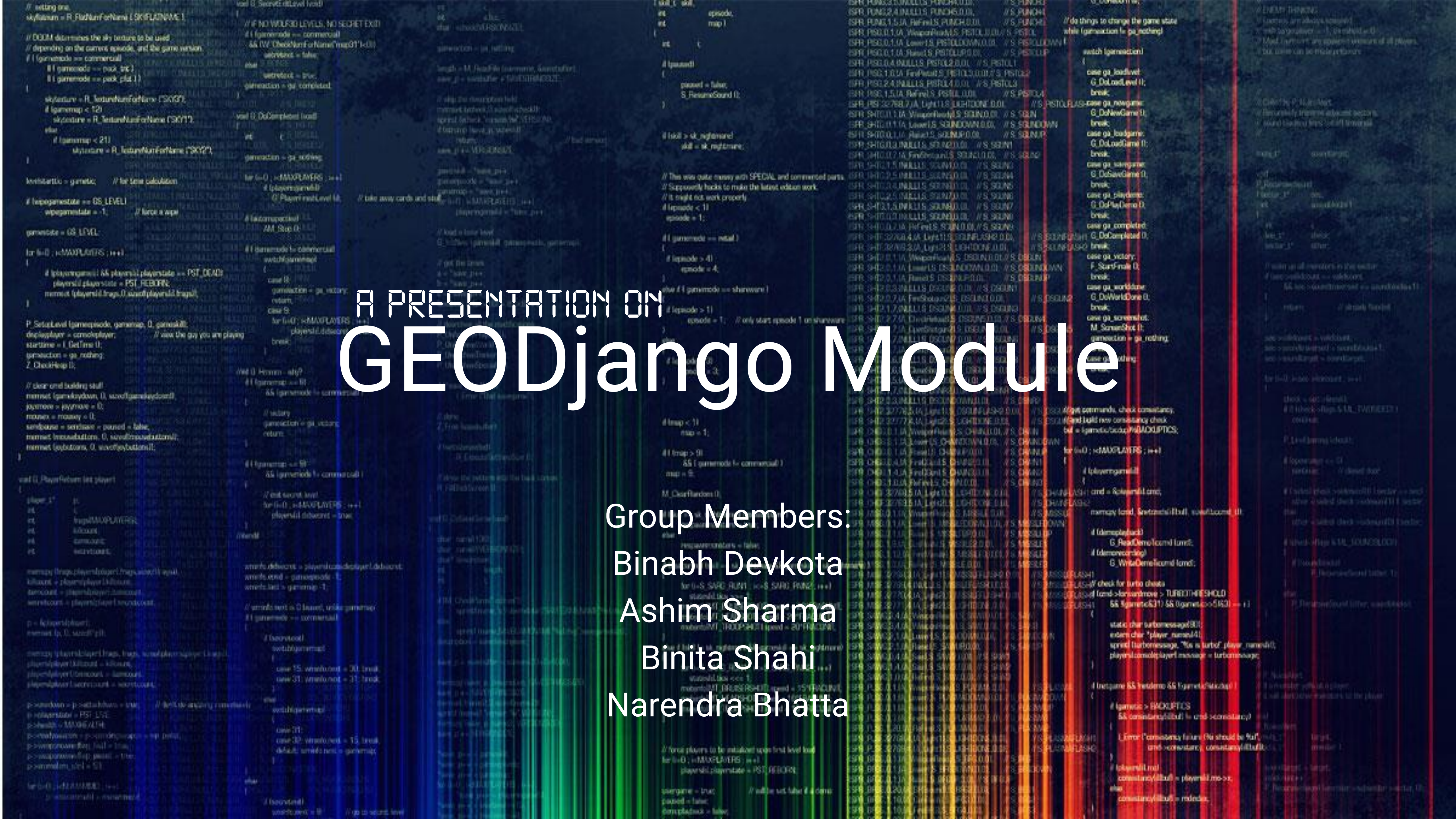


A PRESENTATION ON

GEODjango Module

Group Members:

Binabh Devkota

Ashim Sharma

Binita Shahi

Narendra Bhatta

Django

- A Python framework for web development
- “The web framework for perfectionists with deadlines”
- Its Free and Open Source
- 1500+ contributors on GitHub
- Follows model-view-template (MVT) architectural pattern
- Used by: Youtube, DropBox, Quora, Instagram, Google, Yahoo Maps, Reddit etc.

GeoDjango

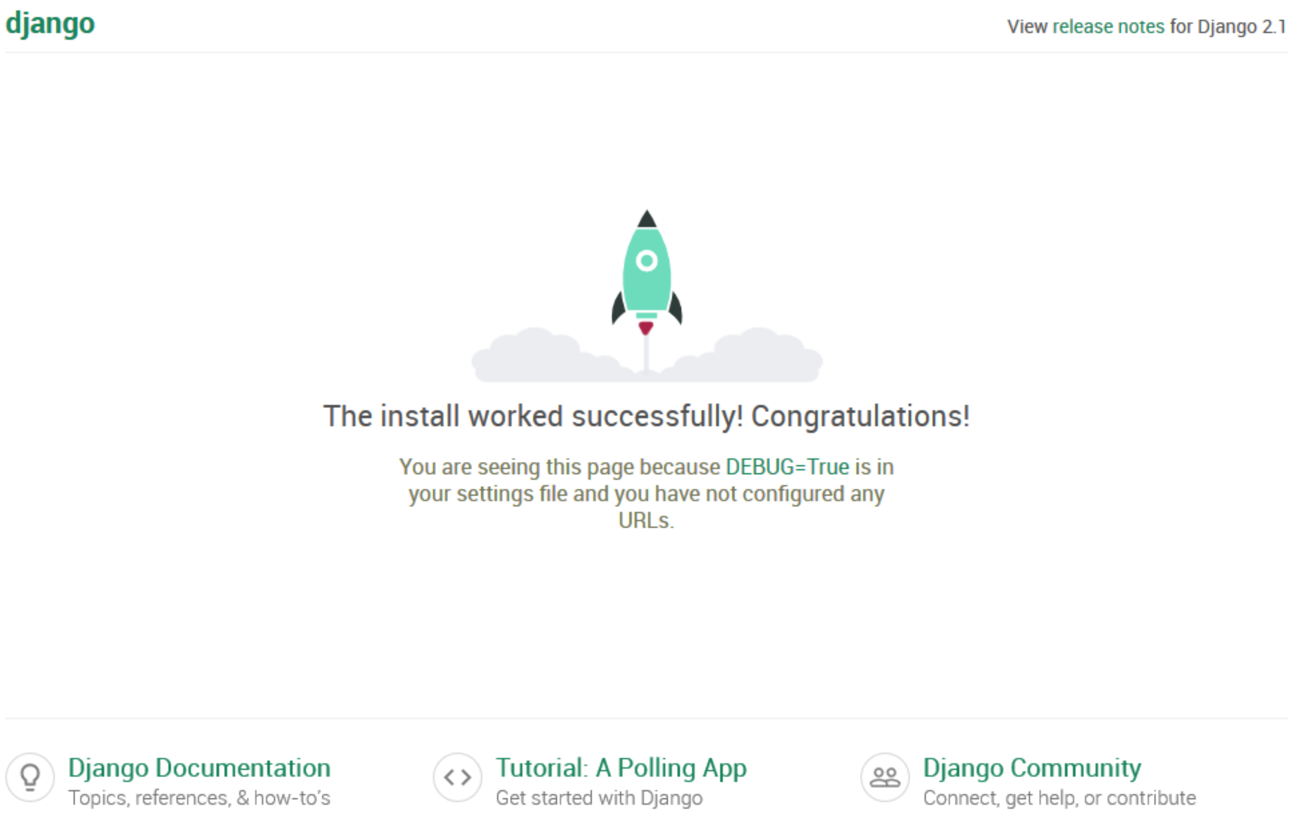
- django.contrib.gis
- A world-class geographic Web framework
- Power of Python/Django on spatial data
- Representing data as map on browser
- Supports variety of data formats
- Wide range of database for backend

Installation and setup

- Setting up the virtual environment [optional]
  - `..\>pip install virtualenv`
  - `..\>virtualenv environment_name`
  - `..\>cd environment_name\Scripts`
  - `..\environment_name\scripts\>activate`
  - `(environment_name) ..\>`
- Installing using pip:
  - `..\>pip install django`
- Starting Project:
  - `..\>django-admin startproject project_name`
- Running Project:
  - `..\>python manage.py runserver [port_number]`

See in browser

- Type in browser’s address bar: `localhost:[port_number]`



django

View release notes for Django 2.1

The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

Django Documentation
Topics, references, & how-to’s

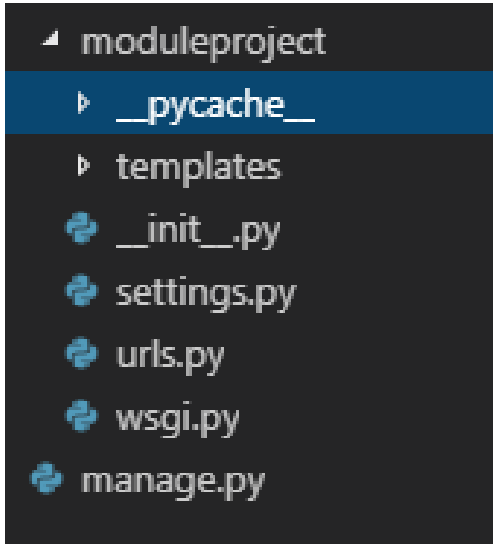
Tutorial: A Polling App
Get started with Django

Django Community
Connect, get help, or contribute

DEMO TIME

File structure of project

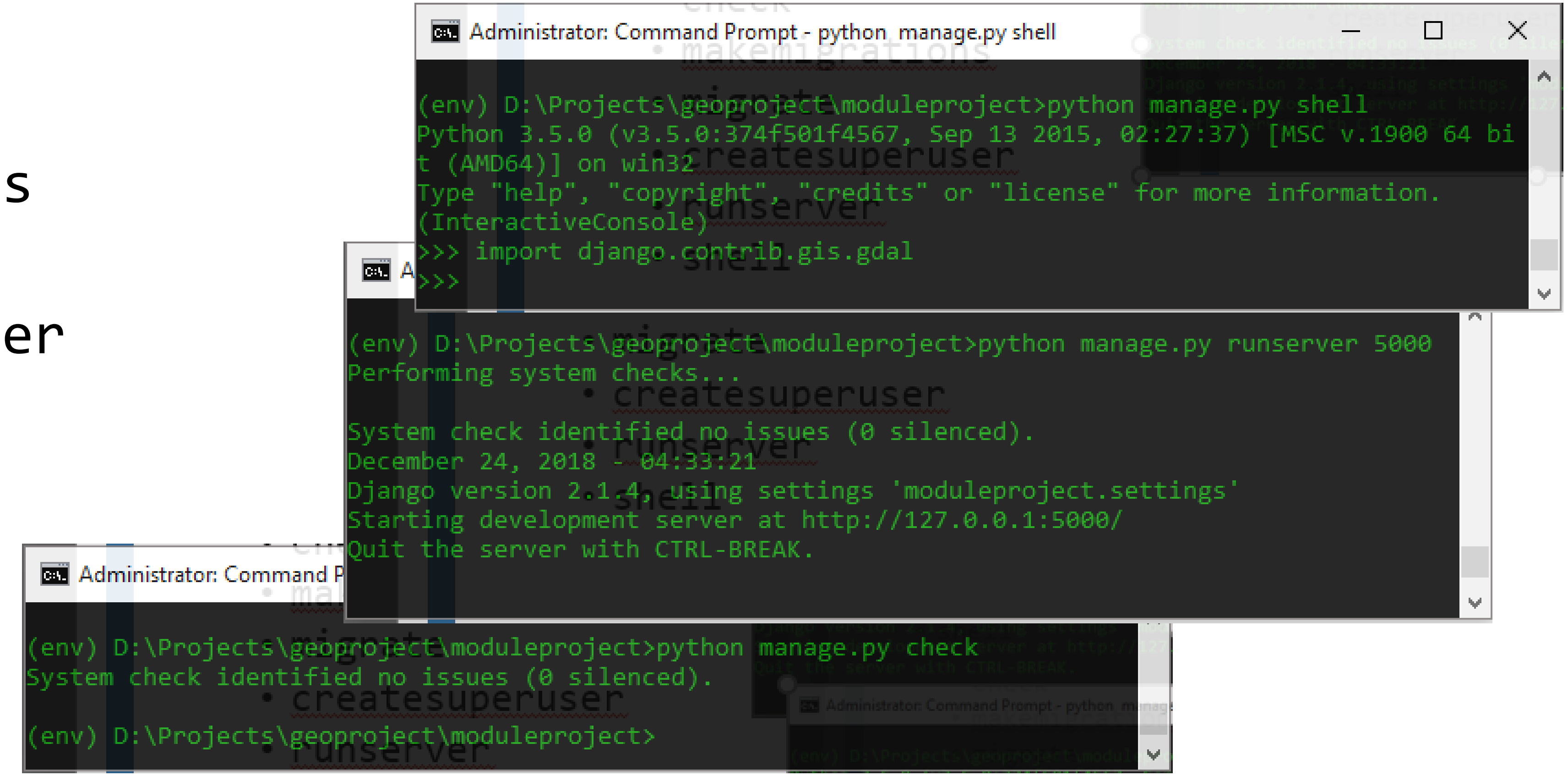
- Main Directory
  - Project Folder
    - __pycache__ //Folder used by system
    - __init__.py //Recognition as django project
    - setting.py //Contains various settings for our project
    - urls.py //Contains urls list and action upon urls entered
    - wsgi.py //ignore this
  - manage.py //used for managing our project



```
◢ moduleproject
    ▷ __pycache__
    ▷ templates
      __init__.py
      settings.py
      urls.py
      wsgi.py
  manage.py
```

manage.py : Usage

- Syntax
  - `..\>python manage.py [command]`
- Commands:
  - check
  - makemigrations
  - migrate
  - createsuperuser
  - runserver
  - shell



```
(env) D:\Projects\geoproject\moduleproject>python manage.py shell
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bi
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> import django.contrib.gis.gdal
>>>
```

```
(env) D:\Projects\geoproject\moduleproject>python manage.py runserver 5000
Performing system checks...

System check identified no issues (0 silenced).
December 24, 2018 - 04:33:21
Django version 2.1.4, using settings 'moduleproject.settings'
Starting development server at http://127.0.0.1:5000/
Quit the server with CTRL-BREAK.
```

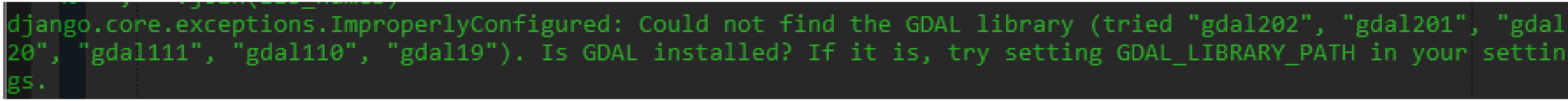
```
(env) D:\Projects\geoproject\moduleproject>python manage.py check
System check identified no issues (0 silenced).

(env) D:\Projects\geoproject\moduleproject>
```

Shell commands

```
>>>from django.contrib.gis.gdal import DataSource
>>>ds=DataSource(r“D:\Projects\Datas\district\Nepal_District.shp”)
>>>layer = ds[0]
>>>print(layer.fields)
>>>print(len(layer))
>>>print(layer.geom_type)
>>>print(layer.srs)
```

Fixing the GDAL error



```
django.core.exceptions.ImproperlyConfigured: Could not find the GDAL library (tried "gdal202", "gdal201", "gdal
20", "gdal111", "gdal110", "gdal19"). Is GDAL installed? If it is, try setting GDAL_LIBRARY_PATH in your settin
gs.
```

- Add this to settings.py file

```
import os
if os.name == 'nt':
    import platform
    OSGEO4W = r"C:\OSGeo4W"
    if '64' in platform.architecture()[0]:
        OSGEO4W += "64"
    assert os.path.isdir(OSGEO4W), "Directory does not exist: " + OSGEO4W
    os.environ['OSGEO4W_ROOT'] = OSGEO4W
    os.environ['GDAL_DATA'] = OSGEO4W + r"\share\gdal"
    os.environ['PROJ_LIB'] = OSGEO4W + r"\share\proj"
    os.environ['PATH'] = OSGEO4W + r"\bin;" + os.environ['PATH']
```

DEMO TIME

Database

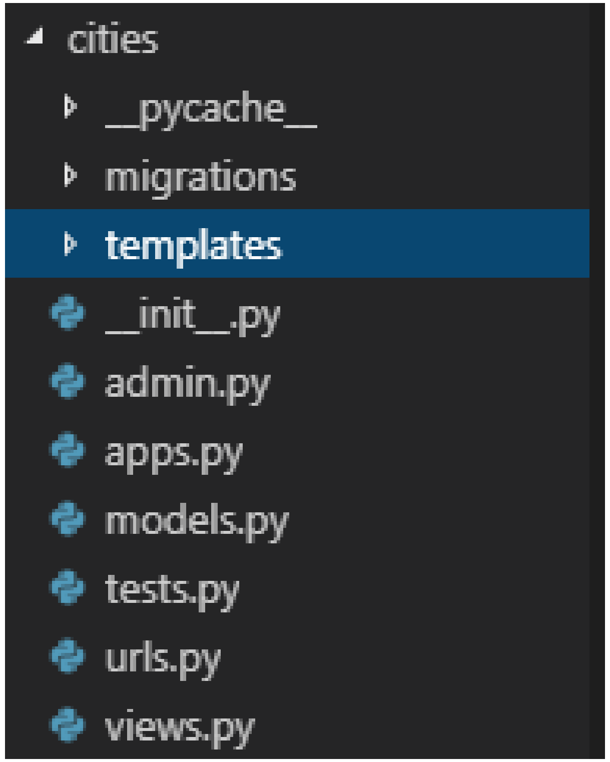
- Connection details in `settings.py` file:

```
DATABASES = {
    'default': {
        'ENGINE':'django.db.backends.postgresql',
        'NAME': 'moduleproject',
        'USER': 'postgres',
        'PASSWORD': 'postgres',
        'HOST': '127.0.0.1',
        'PORT': '5432',
    }
}
```

- Above code may vary according to your database configurations
- To connect and see the tables in database made by django
  - `..\>pip install  psycopg2`
  - `..\>python manage.py check`
  - `..\>python manage.py makemigrations`
  - `..\>python manage.py migrate`

The concept of Apps

- (Models, views, templates) together
- Command:
  - `..\>python manage.py startapp app_name`
- File structure:
  - ➢ app_folder
    - ➢ __pycache__ //System folder
    - ➢ migrations //Files for migrations
    - ➢ __init__.py //For recognition
    - ➢ admin.py //Registering apps in admin area
    - ➢ apps.py //about the app
    - ➢ models.py //Create models here
    - ➢ tests.py //ignore
    - ➢ urls.py //it is not initially there, for managing urls
    - ➢ views.py //creating views on html templates using models



```
cities
  __pycache__
  migrations
  templates
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
```

Inside settings.py

- Registering the created apps

```
INSTALLED_APPS = [
    'django.contrib.admin',
  ……. .
    'myapp',
    'myanotherapp',
]
```

Moving to spatial side of things

- Modifying database:

```
DATABASES = {
     'default': {
          ~~'ENGINE':'django.db.backends.postgresql',~~
          'ENGINE':'django.contrib.gis.db.backends.postgis',
          …..
     }
}
```

- Importing the required package:

```
INSTALLED_APPS = [
     'django.contrib.admin',
     …..
     'django.contrib.gis',
]
```

- `..\>python manage.py check`

Inside cities app

- models.py

```
from django.db import models
from django.contrib.gis.db import models as geomodels
class City(models.Model):
    name = models.CharField(max_length=100, blank=False)
    geometry = geomodels.PointField()
    class Meta:
        # order of drop-down list items
        ordering = ('name',)
        # plural form in admin view
        verbose_name_plural = 'cities'
```

- views.py

```
from django.shortcuts import render
from django.views.generic import DetailView
from .models import City
class CitiesDetailView(DetailView):
    template_name = 'cities/city-detail.html'
    model = City
```

DEMO TIME

Additionals

- Leaflet
  - an open-source JavaScript library for mobile-friendly interactive maps
  - https://leafletjs.com/
  - ../>pip install django-leaflet
  - In settings.py:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    …..
    'leaflet',
]
```

Web Template

- Code in `templates/disthq/disthqs.html`

```
{% load leaflet_tags %}
{% leaflet_css %}
{% leaflet_js %}
<div>
    <h2><center>District Headquaters of Nepal</center></h2>
    {% leaflet_map "main" callback="map_init" %}
</div>
<script type="text/javascript">
    function map_init(map, options) {
        // get point lat and lon
        // zoom to point & add it to map
        var obj = JSON.parse("{{ my_geo_json | escapejs }}");
        console.log(obj)
        map.setView([27.700769, 85.300140], 6);
        L.geoJSON(obj,{onEachFeature: function (feature, layer)
{layer.bindPopup('<p><b>'+feature.properties.dist_name+'</b><br>Name:
'+feature.properties.vdc_name+'<br> Zone:
'+feature.properties.zone_name+'</p>');}}).addTo(map);
    }
</script>
```

Resources

- https://www.djangoproject.com/
- https://docs.djangoproject.com/en/2.1/ref/contrib/gis/
- https://stackoverflow.com/
- https://github.com/
- https://leafletjs.com

DEMO TIME

THANK YOU

ANY QUERIES ?