

Django - Web Framework

Sopan Shewale

Aegis

SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Agenda - Day 1

- Introduction to Django
- Starting Project and Starting Application
- Templates
- Developing views
- URLConf
- Heroku Deployment
- Models
- Admin Interface

Agenda - Day 2

- Web Application - use Matplotlib, Numpy or Pandas to deal with data (simple one)
- Rest API framework
- git - Managing your code versions
- Heroku - Deploy your Application

Django Framework

Web Development Framework - based on Python Programming

Why Django?

With Django, you can take Web applications from concept to launch in a matter of hours.

Django takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

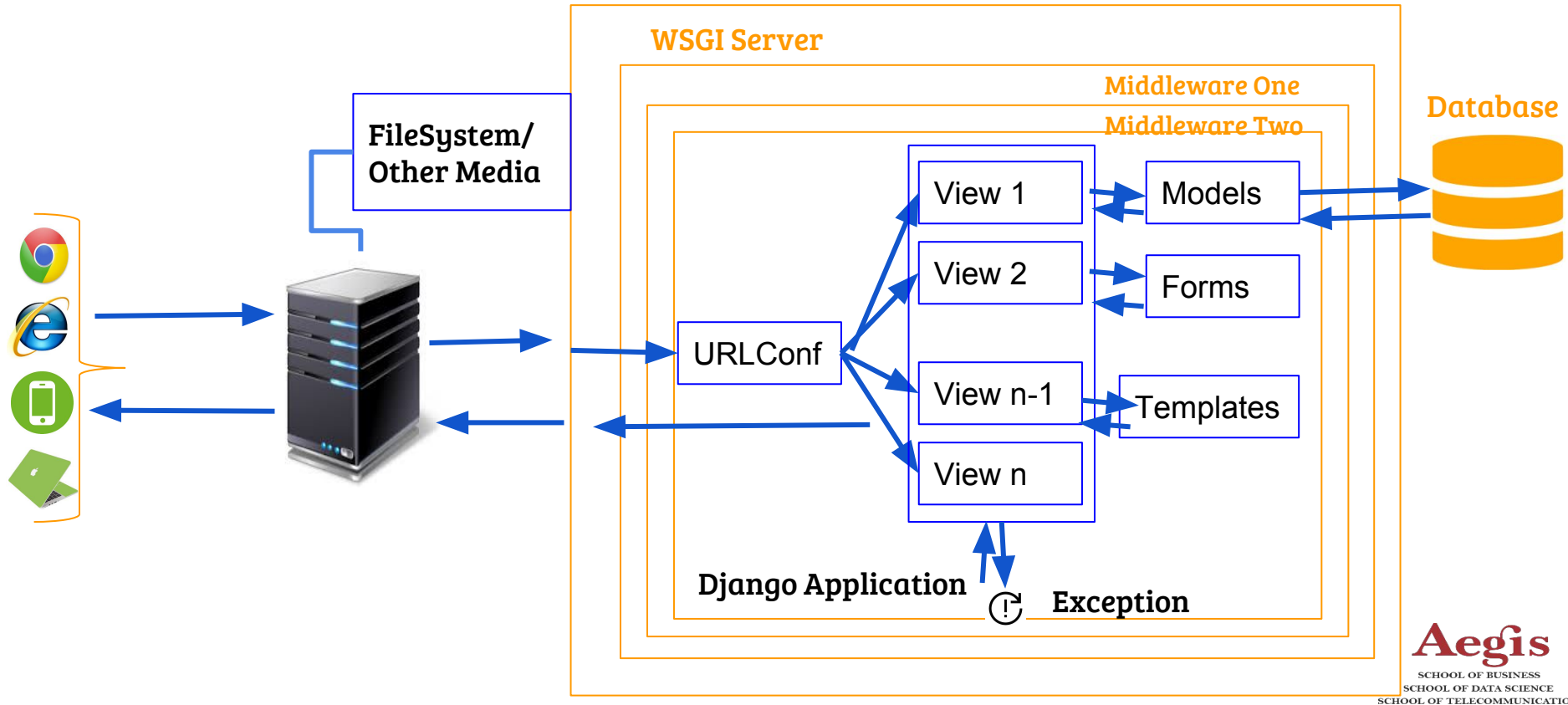
- Fast
- Fully Loaded
- Secure
- Scalable
- Versatile - used for multiple purposes

*Easy - But not so easy, But Large Community
Online Documentation - Most updated*

Aegis

SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Request Processing in Django Application



Django - Project Details

- Project Name: “Lychee”
- Application Name: “temperature”
(Possible Future Apps - whether)

Setting up Project Infrastructure

```
$ pip install virtualenv
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/ed/ea/e20b5cbebf45d3096e8138ab74eda139595d827677f38e9dd543e6015bdf/virtualenv-15.2.0-py2.py3-none-any.whl (2.6MB)
    100% |████████████████████████████████████████| 2.6MB 4.2MB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-15.2.0
```

```
$ virtualenv lychee_env
Using base prefix '/anaconda'
New python executable in /Users/sopanshewale/lychee/lychee_env/bin/python
copying /anaconda/bin/python => /Users/sopanshewale/lychee/lychee_env/bin/python
copying /anaconda/bin/./lib/libpython3.6m.dylib => /Users/sopanshewale/lychee/lychee_env/lib/libpython3.6m.dylib
Installing setuptools, pip, wheel...done.
```

```
$ source lychee_env/bin/activate
(lychee_env) $
(lychee_env) $
(lychee_env) $ python --version
Python 3.6.1 :: Continuum Analytics, Inc.
(lychee_env) $
```

Setting up Project Infrastructure (Cont ...)

```
(lychee_env) $ pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/23/91/2245462e57798e9251de87c88b2b8f996d10ddcb68206a8a020561ef7bd3/Django-2.0.5-py3-none-any.whl (7.1MB)
    100% |████████████████████████████████████████| 7.1MB 686kB/s
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/dc/83/15f7833b70d3e067ca91467ca245bae0f6fe56ddc7451aa0dc5606b120f2/pytz-2018.4-py2.py3-none-any.whl (510kB)
    100% |████████████████████████████████████████| 512kB 4.3MB/s
Installing collected packages: pytz, django
Successfully installed django-2.0.5 pytz-2018.4
(lychee_env) $
(lychee_env) $ python
Python 3.6.1 |Continuum Analytics, Inc.| (default, May 11 2017, 13:04:09)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> print (django.VERSION)
(2, 0, 5, 'final', 0)
>>>
```


Django - Project Architecture

```
(lychee_env) $ django-admin startproject lychee
(lychee_env) $ tree lychee
lychee
├── lychee
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── manage.py

1 directory, 5 files
(lychee_env) $
```

- `__init__.py`: It says the “lychee” directory is Python package
- `settings.py` : Django Project settings (e.g. database configuration details)
- `urls.py`: Pattern details to URL's
- `wsgi.py`: Important file for deployment (Web Server utilize this file)

Starting Project

```
(lychee_env) $ python manage.py runserver  
Performing system checks...
```

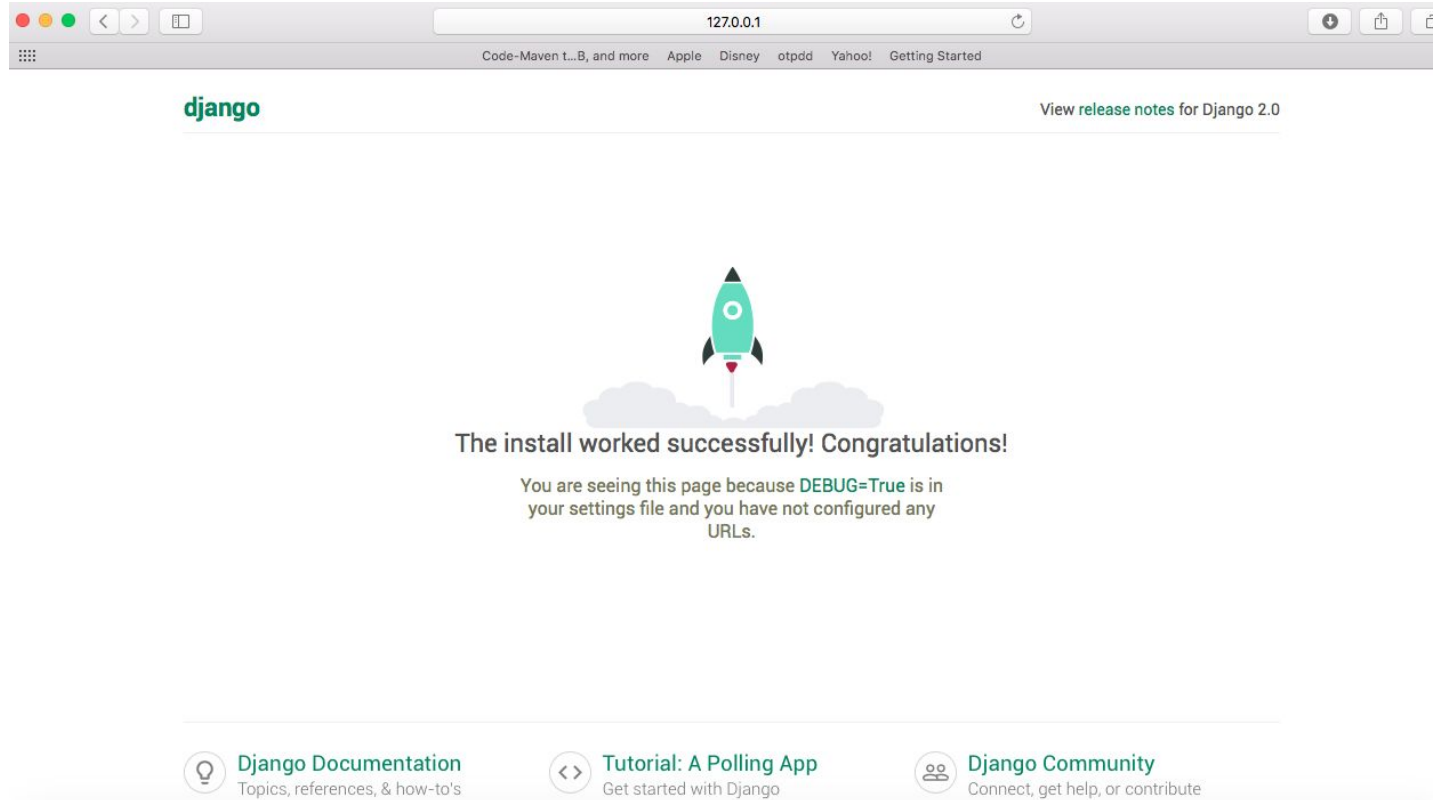
```
System check identified no issues (0 silenced).
```

```
You have 14 unapplied migration(s). Your project may not work properly until you apply the migrations f  
or app(s): admin, auth, contenttypes, sessions.  
Run 'python manage.py migrate' to apply them.
```

```
May 10, 2018 - 14:48:22
```

```
Django version 2.0.5, using settings 'lychee.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.
```

Starting project (Cont ...)



Starting project (Cont ...)

```
(lychee_env) $ tree
.
├── db.sqlite3
├── lychee
│   ├── __init__.py
│   ├── __pycache__
│   │   ├── __init__.cpython-36.pyc
│   │   ├── settings.cpython-36.pyc
│   │   ├── urls.cpython-36.pyc
│   │   └── wsgi.cpython-36.pyc
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── manage.py

2 directories, 10 files
(lychee_env) $
```

Starting project (Cont ...)

System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

```
(lychee_env) $ python manage.py makemigrations
No changes detected
(lychee_env) $ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
(lychee_env) $
```

Starting project (Cont ...)

```
sqlite> .tables
auth_group                auth_user_user_permissions
auth_group_permissions    django_admin_log
auth_permission           django_content_type
auth_user                 django_migrations
auth_user_groups          django_session
sqlite>
```

manage.py makemigrations
Created many Databases

Starting project (Cont ...)

You can also change the “hostname/IP” and “Port” Number to run the application

```
(lychee_env) $ python manage.py runserver 0:8181
Performing system checks...

System check identified no issues (0 silenced).
May 10, 2018 - 15:09:00
Django version 2.0.5, using settings 'lychee.settings'
Starting development server at http://0:8181/
Quit the server with CONTROL-C.
```

You are on 0 (i.e - any IP/name of your machine) and Port - 8181

Starting project (Cont ...)

A Django project is collection of

- *Configurations*
- *applications*

that together make up a given web application or website

Each Application - typically handles one task/feature of the Web Application. The applications can be used (or integrated) in other Django Projects with minimum efforts.

Let us get started with Application

```
(lychee_env) $ python manage.py startapp temperature
```

```
(lychee_env) $ tree
.
├── db.sqlite3
├── lychee
│   ├── __init__.py
│   ├── __pycache__
│   │   ├── __init__.cpython-36.pyc
│   │   ├── settings.cpython-36.pyc
│   │   ├── urls.cpython-36.pyc
│   │   └── wsgi.cpython-36.pyc
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── manage.py
└── temperature
    ├── __init__.py
    ├── admin.py
    ├── apps.py
    ├── migrations
    │   └── __init__.py
    ├── models.py
    ├── tests.py
    └── views.py

4 directories, 17 files
(lychee_env) $
```

Django Application

- **__init__.py**: Indicates Python to treat directory as package
- **models.py**: A place to store applications mode (defines entities relations, types etc)
- **tests.py**: Application test cases
- **Admin.py**:

“**views.py**” and “**models.py**” - most important and frequently used files to develop application

Django Application (Cont ...)

lychee/settings.py

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

Django Application (cont ...)

temperature/views.py

```
from django.shortcuts import render

# Create your views here.
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, world. You are developing Temperature Application")
```

Django Application (Cont ...)

temperature/urls.py

```
from django.urls import path

from . import views

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

Django Application (Cont ...)

lychee/urls.py

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

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

Django Application (Cont ...)

```
(lychee_env) $ python manage.py runserver 0:8181
Performing system checks...
```

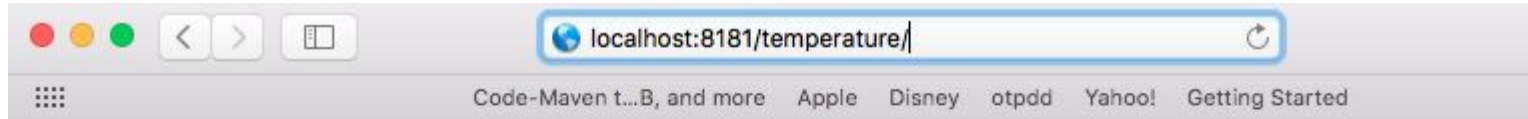
```
System check identified no issues (0 silenced).
```

```
May 10, 2018 - 18:48:15
```

```
Django version 2.0.5, using settings 'lychee.settings'
```

```
Starting development server at http://0:8181/
```

```
Quit the server with CONTROL-C.
```



Hello, world. You are developing Temperature Application

```
[10/May/2018 18:48:31] "GET /temperature HTTP/1.1" 301 0
[10/May/2018 18:48:31] "GET /temperature/ HTTP/1.1" 200 55
```

Starting Application - Summary of Actions

1. `$python manage.py startapp <appname>` - Helps create new application
2. Tell your Django project about the new application - Edit `INSTALLED_APPS` tuple in your project's `settings.py` file.
3. Add `urls.py` file in your application directory
4. Modify your project “`urls.py`” - add mapping to your new application (via `include`)
5. In your application's `view.py`, create the required views ensuring that they return a `HttpResponse` object

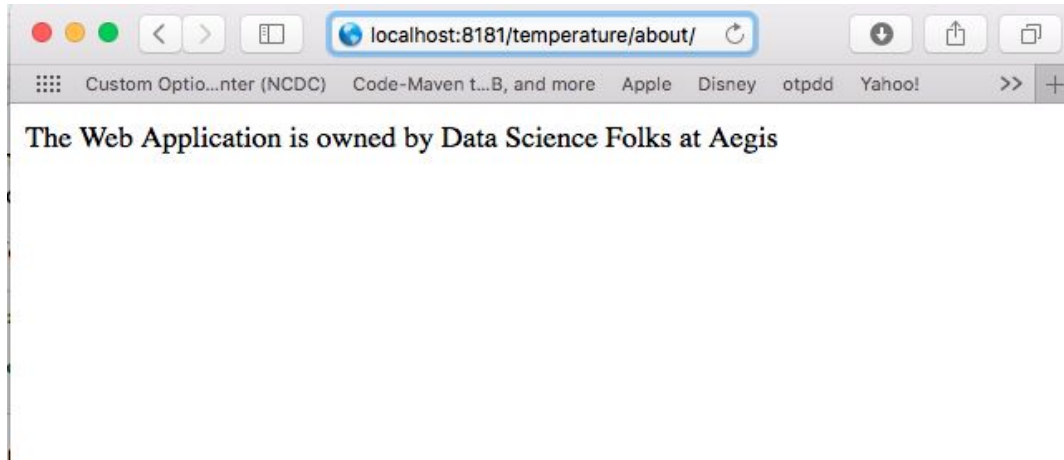
Task - 1

Create “http://localhost:8181/temperature/about” view

-Add “**about**” method in temperature/views.py module

-Update “temperature/urls.py” to support “temperature/about” mapping

-Do you need that restart in your development server?



settings

<project>/settings.py has all the configuration details about your project.

- Database configuration

- Static File Location Details

- Template Details

- May be, you would like to configure a few Constants/Variables used everywhere in your project (e.g. DEPLOYMENT_TYPE=development)

settings (cont...)

```
root@ba896a66dd36:/app# more restbook/settings/development.py
from .base import * # noqa: F403
DEBUG = True
SECRET_KEY = '_1)+c%uy*haml7*vjcf0ubl2h0!bwbed((83&q0#f9i-po!!@8'
ALLOWED_HOSTS = ['*'],
APPLICATION_ENV='dev'
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.
        'NAME': 'postgres',
        'USER': 'postgres'
        'PASSWORD': 'postg
        'HOST': 'postgres'
        'PORT': '',
    }
}
```

```
SMTP = {
    'smtp_server': 'smtp.gmail.com',
    'smtp_port' : 587,
    'mail_user' : 'sopan.shewale@fyra.biz',
    'mail_password': 'xxxxx',
    'from_mail' : 'sopan.shewale@fyra.biz',
}
```

Database

<https://docs.djangoproject.com/en/2.0/ref/settings/#databases>

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

Static Files

Let us configure static file storage directory

lychee/settings.py

```
STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
]
```

Added lychee_logo.png in “static/img/” directory

```
(lychee_env) $ tree static/
static/
├── img
│   └── lychee_logo.png
1 directory, 1 file
```

static/img/lychee_logo.png

“mapped to

http://localhost/static/img/lychee_logo.png

Aegis

SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

static files (cont ...)



Static files (Cont ...)

- Store images
- CSS files and Libraries, Frameworks (e.g. Bootstrap)
- Javascript Files and Libraries (e.g. jQuery)
- Any static content required for your Web Application

Templates

- It's programming language itself
- Django supports powerful templating Engine
 - Helps Rendering/presenting content on browser/any client

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': ["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',  
            ],  
        },  
    ],  
]
```

Templates (Cont ...)

BACKEND - template engine class implementing Django's template backend API.

Possible default Options:

- `django.template.backends.django.DjangoTemplates`
- `django.template.backends.jinja2.Jinja2`

Most engines load templates from files

- **DIRS** defines a list of directories where the engine should look for template files
- **APP_DIRS** tells whether the engine should look for templates inside installed applications

Template (Cont ...)

Let us add template to display our first page

template/temperature/index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Temperature Application </title>
</head>
<body>

<h1>"Hello, world. You are developing Temerature Application"</h1>

</body>
</html>
```

Template (Cont ...)

temperature/views.py

```
from django.shortcuts import render

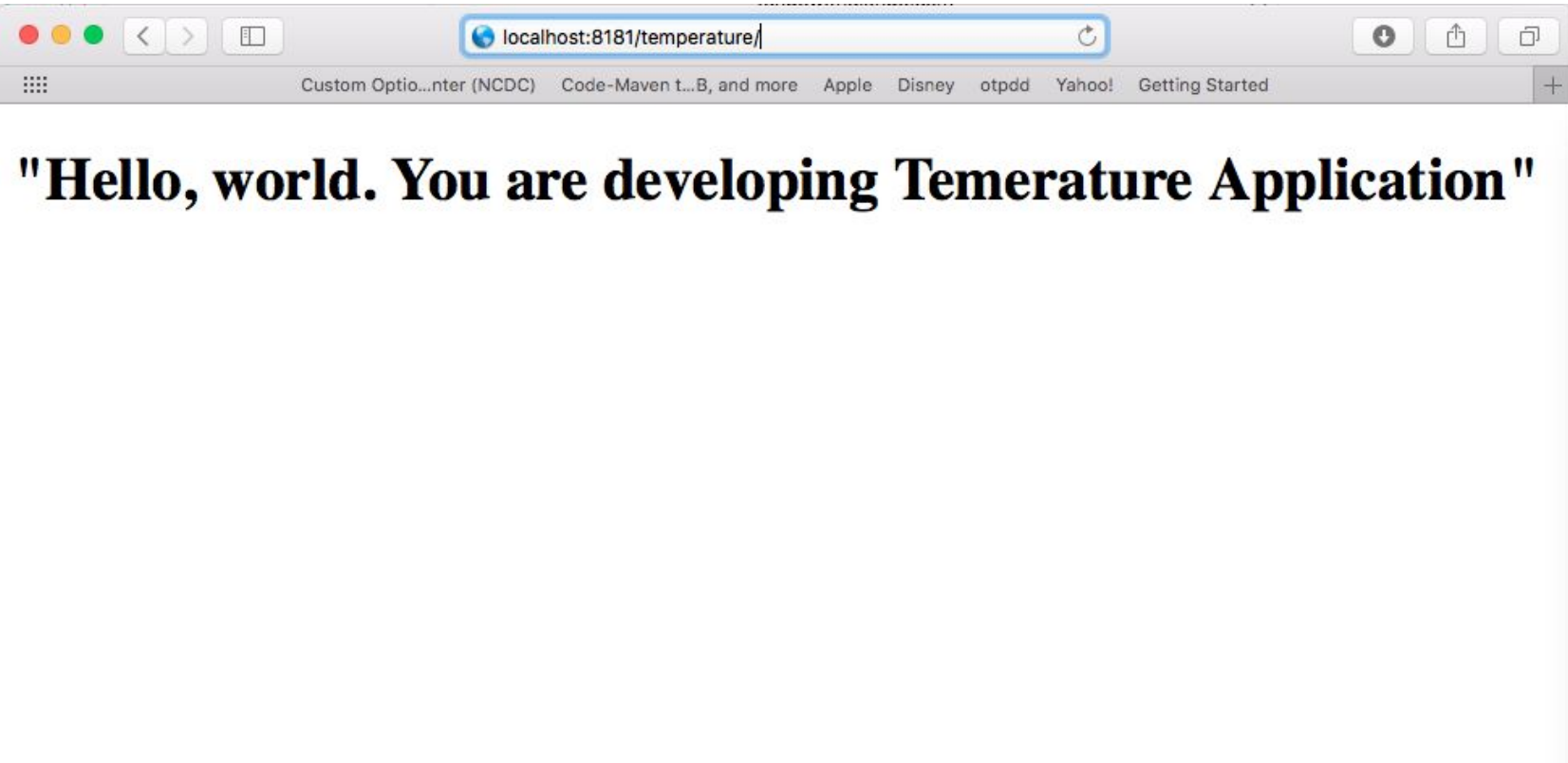
# Create your views here.
from django.http import HttpResponse

def index(request):
    #return HttpResponse("Hello, world. You are developing Temerature Application")
    return render(request, 'index.html')

def about(request):
    return HttpResponse("The Web Application is owned by Data Science Folks at Aegis")
```

“Index.html”
Mapped to
template/temperature/index.html

Template (Cont ...)



Template (Cont ...)

template/temperature/sample_temp.html

```
{% extends "base_generic.html" %}

{% block title %}{{ section.title }}{% endblock %}

{% block content %}
<h1>{{ section.title }}</h1>

{% for story in story_list %}
<h2>
  <a href="{{ story.get_absolute_url }}">
    {{ story.headline|upper }}
  </a>
</h2>
<p>{{ story.tease|truncatewords:"100" }}</p>
{% endfor %}
{% endblock %}
```

Template (Cont ...)

```
<ul>
{% for athlete in athlete_list %}
    <li>{{ athlete.name }}</li>
{% endfor %}
</ul>

{% if athlete_list %}
    Number of athletes: {{ athlete_list|length }}
{% elif athlete_in_locker_room_list %}
    Athletes should be out of the locker room soon!
{% else %}
    No athletes.
{% endif %}
```

Template (Cont ...)

[template/temperature/index.html](#)

```
<!DOCTYPE html>
```

```
{% load staticfiles %}
```

```
<html>
```

```
<head>
```

```
    <title>Temperature Application </title>
```

```
</head>
```

```
<body>
```

```

```

```
<hr>
```

```
<h1>"Hello, world. You are developing Temperature Application"</h1>
```

```
</body>
```

```
</html>
```

```
{% load static %}
```

Template (Cont ...)



Task - 2

“About Page” - Add template for “/temperature/about” URL. Also add logo to the page.

one_lychee.tar.gz

two_lychee.tar.gz

Views - do more!

Let us display records via “views” method

temperature/Temperature.py

```
from datetime import datetime
from django.utils import timezone

class Temperature(object):
    def __init__(self, temperature =0, max_temp = 0, min_temp = 0, record_day = None):
        self.temperature = temperature
        self.max_temp = max_temp
        self.min_temp = min_temp
        if record_day is None:
            self.record_day = timezone.now
        else:
            self.record_day = record_day
```

Views (cont ...)

Look at “records” method from temperature.views

```
def records(request):
    weather_1 = Temperature(
        temperature=random.randint(20,40),
        max_temp=random.randint(30,40),
        min_temp=random.randint(20,30)
    )
    weather_2 = Temperature(
        temperature=random.randint(20,40),
        max_temp=random.randint(30,40),
        min_temp=random.randint(20,30)
    )
    weather_3 = Temperature(
        temperature=random.randint(20,40),
        max_temp=random.randint(30,40),
        min_temp=random.randint(20,30)
    )
    weather_4 = Temperature(
        temperature=random.randint(20,40),
        max_temp=random.randint(30,40),
        min_temp=random.randint(20,30)
    )
    weather_list = [weather_1, weather_2, weather_3, weather_4]
    context = {
        'temp':weather_list
    }

    return render(request, 'weather.html', context)
```

Views (cont ...)

temperature/templates/weather.html

```
{% extends 'base.html' %}
<!DOCTYPE html>

{% load staticfiles %}

{% block content %}

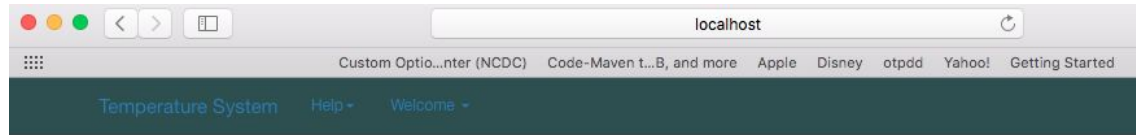


<h3>Weather Records</h3>

<ul class="list-group">
{% for record in temp %}
    <li class="list-group-item"><b>Day Details : {{ record.record_day }} </b></li>
    <li class="list-group-item">Current Temp : {{ record.temperature }} </li>
    <li class="list-group-item">Max Temp : {{ record.max_temp }} </li>
    <li class="list-group-item">Min Temp : {{ record.min_temp }} </li>
    <hr>
{% endfor %}
</ul>

{% endblock %}
```

Views (Cont ...)



Weather Records

| |
|---------------------------------------|
| Day Details : May 11, 2018, 5:20 p.m. |
| Current Temp : 32 |
| Max Temp : 40 |
| Min Temp : 23 |

| |
|---------------------------------------|
| Day Details : May 11, 2018, 5:20 p.m. |
| Current Temp : 40 |
| Max Temp : 37 |
| Min Temp : 26 |

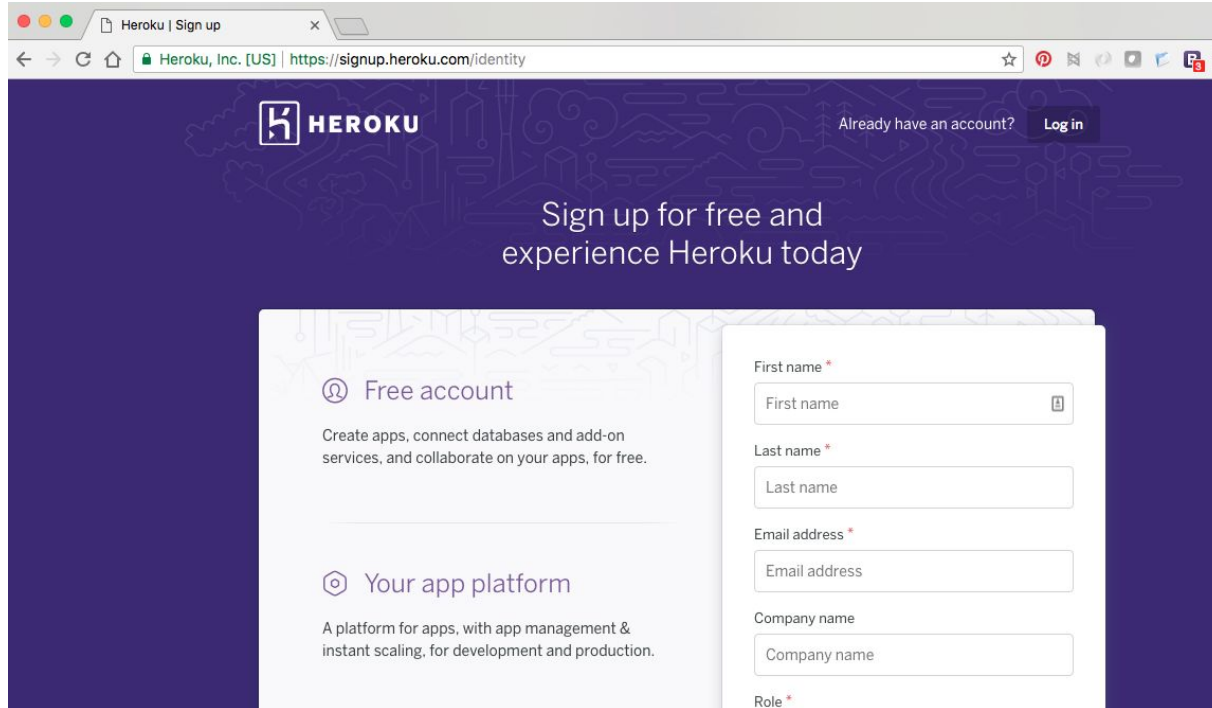
| |
|---------------------------------------|
| Day Details : May 11, 2018, 5:20 p.m. |
| Current Temp : 20 |

Task-3

Using “Temperature Class” - print weeks temperature records.
You need to create week days list using today’s date.

Heroku Deployment

<https://signup.heroku.com/identity>



The screenshot shows a web browser window with the Heroku sign-up page. The browser's address bar shows the URL <https://signup.heroku.com/identity>. The page has a dark blue background with a subtle pattern of white icons. The Heroku logo is in the top left, and a 'Log in' button is in the top right. The main heading is 'Sign up for free and experience Heroku today'. Below this, there are two main sections: 'Free account' and 'Your app platform'. The 'Free account' section includes a sub-heading and a description. The 'Your app platform' section includes a sub-heading and a description. On the right side, there is a sign-up form with fields for 'First name', 'Last name', 'Email address', 'Company name', and 'Role'. Each field has a corresponding label and a small icon.

Heroku | Sign up

Heroku, Inc. [US] | <https://signup.heroku.com/identity>

HEROKU

Already have an account? [Log in](#)

Sign up for free and experience Heroku today

Free account

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

Your app platform

A platform for apps, with app management & instant scaling, for development and production.

First name *

First name

Last name *

Last name

Email address *

Email address

Company name

Company name

Role *

Heroku Deployment

Install Command Line utility for Heroku

```
(lychee_env) hari-sadu:lychee sopanshewale$ heroku login
heroku: Enter your login credentials
Email [sopan.shewale@gmail.com]:
Password: *****
Logged in as sopan.shewale@gmail.com
```

```
(lychee_env) hari-sadu:lychee sopanshewale$ heroku create lychee-temp  
Creating  lychee-temp... done  
https://lychee-temp.herokuapp.com/ | https://git.heroku.com/lychee-temp.git  
(lychee_env) hari-sadu:lychee sopanshewale$
```

-create requirement.txt

-create Procfile

-Create Pipfile

- git init
- git add .
- git commit -am "my lychee app"
- git remote add heroku <https://git.heroku.com/lychee-temp.git>
- git push heroku master

Assignment

Create Account at Heroku

Deploy application at Heroku

Enable Static Files Delivery

Aegis

SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Models and Databases

- Almost all hassle is taken care by Django's *object relational mapping (ORM)*
- Django encapsulates databases tables through models
- model is a Python object that describes your data model/table
- That helps you manipulate the Python object than playing with SQL Queries

lychee/settings.py

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

Models (Cont ...)

temperature/models.py

```
from django.db import models
from django.utils.translation import ugettext_lazy as _
from django.utils import timezone
from datetime import datetime

class Temperature(models.Model):
    temperature = models.PositiveIntegerField(
        verbose_name=_('Current Temperatuer in Cel'),
        default = 0,
    )
    min_temp = models.PositiveIntegerField(
        verbose_name=_('Current Temperatuer in Cel'),
        default = 0,
    )
    max_temp = models.PositiveIntegerField(
        verbose_name=_('Current Temperatuer in Cel'),
        default = 0,
    )
    entry_time = models.DateTimeField(
        default= timezone.now,
        verbose_name=_('Entry Time'),
        blank=True, null=True,
    )
```

```
(lychee_env) hari-sadu:lychee sopanshewale$ python manage.py makemigrations temperature
Migrations for 'temperature':
  temperature/migrations/0001_initial.py
    - Create model Temperature
```

```
(lychee_env) hari-sadu:lychee sopanshewale$ more temperature/migrations/0001_initial.py
# Generated by Django 2.0.5 on 2018-05-12 07:14

from django.db import migrations, models
import django.utils.timezone

class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Temperature',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('temperature', models.PositiveIntegerField(default=0, verbose_name='Current Temperatuer in Cel')),
                ('min_temp', models.PositiveIntegerField(default=0, verbose_name='Current Temperatuer in Cel')),
                ('max_temp', models.PositiveIntegerField(default=0, verbose_name='Current Temperatuer in Cel')),
                ('entry_time', models.DateTimeField(blank=True, default=django.utils.timezone.now, null=True, verbose_name='Entry Time')),
            ],
        ),
    ]
```



```
(lychee_env) hari-sadu:lychee sopanshewale$ python manage.py migrate
```

Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions, temperature

Running migrations:

```
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying sessions.0001_initial... OK
Applying temperature.0001_initial... OK
```



```
sqlite> .tables
auth_group                django_admin_log
auth_group_permissions    django_content_type
auth_permission           django_migrations
auth_user                 django_session
auth_user_groups          temperature_temperature
auth_user_user_permissions
sqlite> █
```

Admin Interface

Create Admin User

```
(lychee_env) $ python manage.py createsuperuser
Username (leave blank to use 'sopanshewale'): info@aegis.com
Email address: info@aegis.com
Password:
Password (again):
Superuser created successfully.
(lychee_env) $
```

localhost:8181/admin

How to Deplo...n on Heroku? Custom Optio...nter (NCDC) Code-Maven t...B, and more Apple Disney otpdd Yahoo! Getting Started

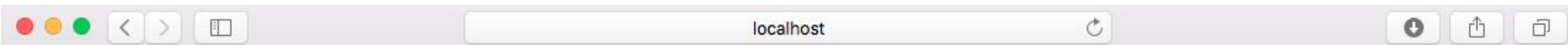
Log in | Django site admin Temperature System Custom Options - Daily Summaries | Cli

Django administration

Username:

Password:

Log in



Django administration

WELCOME, **INFO@AEGIS.COM**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home » Authentication and Authorization

Authentication and Authorization administration

| AUTHENTICATION AND AUTHORIZATION | | |
|----------------------------------|-----------------------|--------------------------|
| Groups | + Add | ✎ Change |
| Users | + Add | ✎ Change |

```
from django.contrib import admin  
  
from temperature.models import Temperature  
admin.site.register(Temperature)
```

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

 Change

Users

+ Add

 Change

TEMPERATURE

Temperatures

+ Add

 Change

Recent actions

My actions

None available

Aegis

SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

thx