



University of Essex

---

Online

---

# **Launching into Cyber Security**

## **Week 11 Seminar**

### **Sammy Danso, PhD**



# Announcement

- E-portfolio
- Individual essay post-mortem
- Programming assignment



# Individual essay – post mortem

- ❑ No names – blinded marking

- ❑ Assessment criteria

- ✓ Knowledge
- ✓ Criticality
- ✓ Use of relevant sources
- ✓ Structure and presentation
- ✓ Academic integrity



# This week's task

- ☐ Intro to Django

- ☐ Walk through some example code



# This week's task

☐ **Intro to Django**

☐ A walk through some example code



# Intro to Django: web development framework

- ☐ High- level web framework - a software that makes it easy to build dynamic websites.
- ☐ Provides an abstract solution to common problems of web development by providing 'shortcuts' to frequent programming tasks.
- ☐ Dynamic website as opposed to static pages – web application.
- ☐ Can retrieve records from a database or carry out a task based on user input.
- ☐ Provides a mechanism of mapping requested URLs to code that handles requests – code to specific to username.



# Intro to Django: web development framework

- ☐ HTML forms can easily be displayed, validated and redisplayed.
- ☐ Supports conversion of user- submitted input into data types for easy manipulation.
- ☐ Ability to separate content from presentation - a template system for look & feel modifications.
- ☐ Provides a mechanism to easily integrate with storage layers.



# Intro to Django: common development frameworks

## **Smarty**

- Focuses on separating application logic and presentation
- Uses a temporary caching approach to facilitate delivering pages

## **Yii framework**

- Focuses on reducing SQL statements and facilitates integration with other platforms
- Provides security, authentication, and professional features to create robust projects rapidly.

## **Zend framework**

- Popular for enterprise-level applications and focuses on modularity, performance and maintenance.
- Allows continues extension and integration

## **Django**

- A high-level Python-based framework for rapid web application development
- Supports the properties of common frameworks

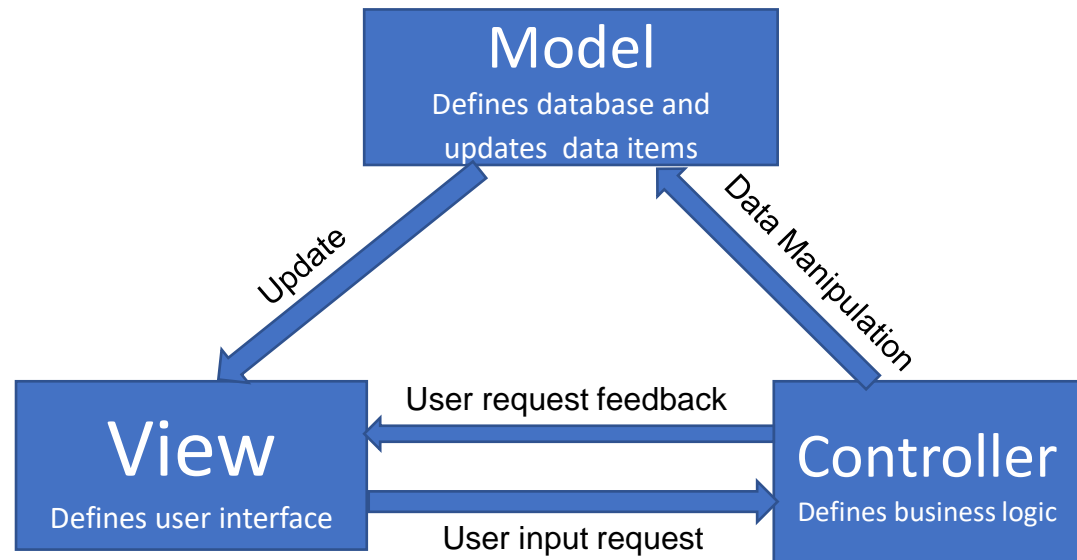




# Intro to Django: specific properties

- ❑ Django encourages loose coupling:  
application should communicate with each other through APIs:

- Template
- Database-access system
- HTTP request/response layer
- Caching



- ❑ Django supports the MVC architecture



# Intro to Django: issues with traditional frameworks e. g PHP

```
<html>
<head><title>Friends of mine</title></head>
<body>

<h1>Friends of mine</h1>

<ul>

<?php
$connection = @mysql_connect("localhost", "my_username", "my_pass");
mysql_select_db("my_database");
$people = mysql_query("SELECT name, age FROM friends");
while ( $person = mysql_fetch_array($people, MYSQL_ASSOC) ) {
?>
<li>
<?php echo $person['name'] ?> is <?php echo $person['age'] ?> years old.
</li>
<?php } ?>

</ul>

</body>
</html>
```



The presentation is tied to the code



# Intro to Django: issues with traditional frameworks e. g PHP

```
<html>
<head><title>Friends of mine</title></head>
<body>

<h1>Friends of mine</h1>

<ul>

<?php
$connection = @mysql_connect("localhost", "my_username", "my_pass");
mysql_select_db("my_database");
$people = mysql_query("SELECT name, age FROM friends");
while ( $person = mysql_fetch_array($people, MYSQL_ASSOC) ) {
?>
<li>
<?php echo $person['name'] ?> is <?php echo $person['age'] ?> years old.
</li>
<?php } ?>

</ul>

</body>
</html>
```

**The database code is tied to the business logic**



# Intro to Django: issues with traditional frameworks e. g PHP

```
<html>
<head><title>Friends of mine</title></head>
<body>

<h1>Friends of mine</h1>

<ul>

<?php
$connection = @mysql_connect("localhost", "my_username", "my_pass");
mysql_select_db("my_database");
$people = mysql_query("SELECT name, age FROM friends");
while ( $person = mysql_fetch_array($people, MYSQL_ASSOC) ) {
?>
<li>
<?php echo $person['name'] ?> is <?php echo $person['age'] ?> years old.
</li>
<?php } ?>

</ul>

</body>
</html>
```

The database connection parameters and backend  
are hard-coded



# Intro to Django: getting started

## Installing Django

- ✓ Django requires Python 2.3 or higher
- ✓ Installation available from <http://www.djangoproject.com/download/>
- ✓ Download the tarball, which will be named something like `Django-1.0.tar.gz`.
- ✓ `tar xzvf Django-*.tar.gz`
- ✓ `cd Django-*`
- ✓ `sudo python setup.py install`



# Intro to Django: getting started

## Starting a project

```
$ django-admin.py startproject myWebStartUp
```

```
$ myWebStartUp / ##container for the project
```

```
manage.py ## A command-line utility that lets you interact with this Django project in various ways.
```

```
myWebStartUp /
```

```
__init__.py ## An empty file that tells Python that this directory should be considered a Python package
```

```
settings.py ## Settings/configuration for this Django project
```

```
urls.py ## The URL declarations for this Django project; a “table of contents” of your Django-powered site
```

```
wsgi.py ## An entry-point for WSGI-compatible web servers to serve your project
```



# Intro to Django: getting started

## starting a server

```
$ python manage.py runserver 8080
```

>>From browser - visit <http://127.0.0.1:8000>

## changing IP address

```
$ python manage.py runserver 0.0.0.0:8080
```



# Intro to Django: getting started

```
Filetree x
STEST-UNIVERSITY-OF-ESSEX-ONLINE1
Hello Codio in Django-1

Hello Codio in Django-1 (master)
├── django01
│   └── myFirstWebApp
│       ├── __init__.py
│       ├── settings.py
│       ├── urls.py
│       ├── wsgi.py
│       └── manage.py
├── myWebStartUp
└── webApp
    ├── .settings
    └── README.md

Terminal
urls.py settings.py Start myWebSt... RunServer x
*
* Your Codio Box domain is: economy-union.codio.io
*
Last login: Fri Dec 4 12:05:29 2020 from 192.168.10.156
python django01/myWebStartUp/manage.py runserver 0.0.0.0:8000
codio@economy-union:~/workspace$ python django01/myWebStartUp/manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 04, 2020 - 12:05:57
Django version 2.2.12, using settings 'myWebStartUp.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
[04/Dec/2020 12:06:11] "GET /admin HTTP/1.1" 301 0
[04/Dec/2020 12:06:12] "GET /admin/ HTTP/1.1" 302 0
[04/Dec/2020 12:06:12] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 1819
[04/Dec/2020 12:06:13] "GET /static/admin/css/base.css HTTP/1.1" 200 16378
[04/Dec/2020 12:06:13] "GET /static/admin/css/responsive.css HTTP/1.1" 200 17944
[04/Dec/2020 12:06:13] "GET /static/admin/css/login.css HTTP/1.1" 200 1233
[04/Dec/2020 12:06:13] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
[04/Dec/2020 12:06:14] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876
[04/Dec/2020 12:06:14] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692
Not Found: /favicon.ico
[04/Dec/2020 12:06:14] "GET /favicon.ico HTTP/1.1" 404 1995
[04/Dec/2020 12:06:37] "POST /admin/login/?next=/admin/ HTTP/1.1" 302 0
[04/Dec/2020 12:06:38] "GET /admin/ HTTP/1.1" 200 3042
[04/Dec/2020 12:06:38] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 412
[04/Dec/2020 12:06:38] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184
[04/Dec/2020 12:06:38] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331
[04/Dec/2020 12:06:38] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380
[04/Dec/2020 12:06:51] "GET / HTTP/1.1" 200 16348
```



# Django admin console

## Django administration

WELCOME, **CODIO** / [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

### Site administration

#### AUTHENTICATION AND AUTHORIZATION

<b>Groups</b>	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
<b>Users</b>	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

#### Recent actions

##### My actions

None available



# Django admin console

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

Available user permissions ?



- admin | log entry | Can add log entry
- admin | log entry | Can change log entry
- admin | log entry | Can delete log entry
- admin | log entry | Can view log entry
- auth | group | Can add group
- auth | group | Can change group
- auth | group | Can delete group
- auth | group | Can view group
- auth | permission | Can add permission
- auth | permission | Can change permission
- auth | permission | Can delete permission
- auth | permission | Can view permission
- auth | user | Can add user
- auth | user | Can change user

Choose all ?

Chosen user permissions ?

contenttypes | content type | Can add content type

Remove all

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.



# Django admin console

## Important dates

Last login:

Date:

Today



Time:

Now



Date joined:

Date:

2020-12-01

Today



Time:

17:16:35

Now



Delete

Save and add another

Save and continue editing

SAVE



### Personal info

First name:

John

Last name:

Smith

Email address:

### Permissions

☒ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☐ Staff status

Designates whether the user can log into this admin site.

☐ Superuser status

Designates that this user has all permissions without explicitly assigning them.



# Intro to Django: getting started

## Where to keep your code

### PHP tradition

- code is kept under web server's document root ie `/var/www`
- code is exposed - not secure!

### Django

- Python code can be kept outside document root e.g `/home/mycode`
- Secured – web server not exposed



# This week's task

☐ Intro to Django

☐ **A walk through some example code**



# Code walk through: creating an app

This example is taken from..

---

**Django Documentation**

*Release 2.2.17.dev20201013064537*

Django Software Foundation

October 13, 2020



# Code walk through: creating an App

```
$ python manage.py startapp polls
```

```
polls /
```

```
    __init__.py
```

```
    admin.py
```

```
    apps.py
```

```
    migrations/
```

```
        __init__.py
```

```
    models.py
```

```
    tests.py
```

```
    views.py
```





# Code walk through: creating an app

## Creating a view for the App

```
from django.http import HttpResponse
```

```
def index(request):
```

```
    return HttpResponse("Hello, world. You're at the polls index.")
```



# Code walk through: creating an app

## Creating a view for the App

```
from django.http import HttpResponse
```

```
def index(request):
```

```
    return HttpResponse("Hello, world. You're at the polls index.")
```

- But...we need to map view to a URL using URLconf file



# Code walk through: creating an app

- Creating a URLconf. for the view

polls /

\_\_init\_\_.py

admin.py

apps.py

migrations/

\_\_init\_\_.py

models.py

tests.py

views.py



# Code walk through: creating an app

- Creating a URLconf. for the view - create a urls file

polls /

\_\_init\_\_.py

admin.py

apps.py

migrations/

\_\_init\_\_.py

models.py

tests.py

views.py

urls.py



# Intro to Django: code walk through

## Content of urls.py

### **##Polls/ urls.py**

```
from django.urls import path
from . import views
urlpatterns = [
    path("", views.index, name='index'),
]
```

### **## myWebStartUp/ urls.py**

```
from django.contrib import admin
from django.urls import include, path
urlpatterns = [
    path('polls/', include('polls.urls')),
    path('admin/', admin.site.urls),
]
```



# Intro to Django: code walk through

## Creating a model

✓ Making sure you have a database setup – myWebStartUp/settings.py

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'myWebStartUp' containing files like 'urls.py', 'views.py', 'models.py', 'tests.py', 'admin.py', 'apps.py', 'migrations', and 'polls'. The code editor shows the 'settings.py' file with the following content:

```
73 # Database
74 # https://docs.djangoproject.com/en/2.2/ref/settings/#databases
75
76 DATABASES = {
77     'default': {
78         'ENGINE': 'django.db.backends.sqlite3',
79         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
80     }
81 }
82
83 # Password validation
84 # https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators
85
86 AUTH_PASSWORD_VALIDATORS = [
87     {
88         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
89     },
90     {
91         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
92     },
93     {
94         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
95     },
96     {
97         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
98     },
99 ]
100
101
```



# Intro to Django: code walk through

## Creating a model

- ✓ Apply settings to the model

\$ `python manage.py migrate`

```
codio@economy-union:~/workspace/myWebStartUp$ 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 admin.0003_logentry_add_action_flag_choices... 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 auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying sessions.0001_initial... OK
codio@economy-union:~/workspace/myWebStartUp$
```



# Intro to Django: code walk through

## Creating a model

- ✓ Define the model – database layout and additional metadata

\$ polls/models.py

```
from django.db import models
```

```
class Question(models.Model):
```

```
    question_text = models.CharField(max_length=200)
```

```
    pub_date = models.DateTimeField('date published')
```

```
class Choice(models.Model):
```

```
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
```

```
    choice_text = models.CharField(max_length=200)
```

```
    votes = models.IntegerField(default=0)
```





# Intro to Django: code walk through

## Activating the model

- ✓ Include app by adding a reference to configuration class in `INSTALLED_APP` setting

```
$ myWebStartUp/settings.py
INSTALLED_APPS = [
    'polls.apps.PollsConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```



# Intro to Django: code walk through

## Activating the model

✓ Apply changes to model

```
$ python manage.py makemigrations polls
```

```
codio@economy-union:~/workspace/myWebStartUp$ python manage.py makemigrations polls
Migrations for 'polls':
  polls/migrations/0001_initial.py
    - Create model Question
    - Create model Choice
```



# Intro to Django: code walk through

## Creating model tables for database

- ✓ View the content of the SQL statement that will create the model tables for the database.

```
$ python manage.py sqlmigrate polls 0001
```

```
codio@economy-union:~/workspace/myWebStartUp$ python manage.py sqlmigrate polls 0001
BEGIN;
--
-- Create model Question
--
CREATE TABLE "polls_question" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "question_text" varchar(200) NOT NULL, "pub
te" datetime NOT NULL);
--
-- Create model Choice
--
CREATE TABLE "polls_choice" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "choice_text" varchar(200) NOT NULL, "votes"
eger NOT NULL, "question_id" integer NOT NULL REFERENCES "polls_question" ("id") DEFERRABLE INITIALLY DEFERRED);
CREATE INDEX "polls_choice_question_id_c5b4b260" ON "polls_choice" ("question_id");
COMMIT;
codio@economy-union:~/workspace/myWebStartUp$
```



# Intro to Django: code walk through

## Create model tables in database

✓ Create the model tables in database

```
$ python manage.py migrate
```



# Intro to Django

## Making changes to model

- ✓ Change your models (in **models.py**).
- ✓ Run **python manage.py makemigrations** to create migrations for those changes
- ✓ Run **python manage.py migrate** to apply those changes to the database.



# Intro to Django

## Testing polls App from API

```
codio@economy-union:~/workspace/myWebStartUp$ python manage.py shell
Python 3.6.5 |Anaconda, Inc.| (default, Apr 29 2018, 16:14:56)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.4.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from polls.models import Choice, Question

In [2]: from django.utils import timezone

In [3]: q = Question(question_text="What's new?", pub_date=timezone.now())

In [4]: q.save()

In [5]: q.id
Out[5]: 1

In [6]: q.question_text
Out[6]: "What's new?"

In [7]: q.pub_date
Out[7]: datetime.datetime(2020, 12, 4, 16, 8, 39, 805744, tzinfo=<UTC>)

In [8]:
```



# Keeping in touch

**Dr Samuel Danso**

**Email:** [samuel.danso@online.essex.ac.uk](mailto:samuel.danso@online.essex.ac.uk)



samdansobe

**Office Hours:**

Wednesday 6.30-7.30pm (18.30-19.30)

Saturday 5-6pm (17.00-18.00)

