Week 3

1. **Django Project Structure**

Create project with this command django-admin startproject <project\_name>. This will create a series of files in a directory with the project\_name.

File manage.py is used to execute commands with Django project

Within directory settings.py is used to hold configuration settings while urls.py is used to hold routes.

**To start server use command:**

* Python manage.py startserver

**To create new application:**

* Python manage.py startapp <app\_name>
* Add <app\_name> to setings.py under INSTALLED\_APPS

**How to add a view/route in an application**

* Within views.py, you can create a view by creating a new function and having it take an argument called request.
  + This request is the HTTP request sent by clients
  + The name of the function would be the name of the route
* Create a separate urls.py within the root of the application folder that was created
* Within urls.py create a list with variable name called urlpatterns
  + This list should contain all routes that exists for this app. Each element should therefore be 1 url path.
  + To create a url path, import path from django.urls library. Each path takes 2 arguments
* Link project wide urls.py and app urls.py

Note: Within 1 Django project can have multiple applications as different applications can be used for different things.

**How Django processes requests?**

Urls.py acts as a url dispatcher. To design URLs for an app, a Python module called URLconf (informally named) must be created. This is normally called urls.py. This module is pure Python code and is a mapping between URL path expression and views (python functions).

1. Django determines the root URLconf module to use. Ordinarily, this is the value of the [ROOT\_URLCONF](https://docs.djangoproject.com/en/3.2/ref/settings/#std:setting-ROOT_URLCONF) setting [This is the urls.py contained at the project level], but if the incoming HttpRequest object has a [urlconf](https://docs.djangoproject.com/en/3.2/ref/request-response/" \l "django.http.HttpRequest.urlconf" \o "django.http.HttpRequest.urlconf) attribute (set by middleware), its value will be used in place of the [ROOT\_URLCONF](https://docs.djangoproject.com/en/3.2/ref/settings/#std:setting-ROOT_URLCONF) setting.
2. Django loads Python module and looks for variable urlpatterns. This should be a sequence of of django.urls.path() and/or django.urls.re\_path() instances
3. Django runs through each URL pattern in order and stops at the first one that matches the requested URL matching against the attribute path\_info in either path() or re\_path() instances
4. Django imports and calls the given view – which is normally a Python function. The view gets passed the following arguments
   1. An instance of the HTTP request
   2. If the matched URL contains no named groups, then matches from regex are provided as positional arguments.
   3. The keyword arguments are made up of any named parts matched by the path expression that are provided, overridden by any arguments specified in the optional kwargs argument to [django.urls.path()](https://docs.djangoproject.com/en/3.2/ref/urls/" \l "django.urls.path" \o "django.urls.path) or [django.urls.re\_path()](https://docs.djangoproject.com/en/3.2/ref/urls/" \l "django.urls.re_path" \o "django.urls.re_path).
5. If no URL pattern matches, or if exception raised, Django invokes an error handling view.