**Django Review: Setup**

1. Setup VM
   1. python -m venv <mydirectory>
   2. <mydirectory>\Scripts\activate OR <mydirectory>\bin\activate
   3. pip install Django
   4. pip install mysqlclient
2. Create project
   1. django-admin startproject <projectname>
   2. Navigate to <project name> directory
3. Create MYSQL DB
   1. Login to mysql
   2. create database <database name>
4. Configure settings.py
   1. Import os module: **import os**
   2. Change timezone: **TIME\_ZONE = ‘Hongkong’**
   3. Add base directory for static files (right below STATIC\_URL):  STATIC\_ROOT = os.path.join(BASE\_DIR, 'static')
   4. Configure allowed hosts for deployment: **ALLOWED\_HOSTS = [‘127.0.0.1’, ‘localhost’]**
   5. Change database to mysql:

**DATABASES = {**

**'default': {**

**'ENGINE': 'django.db.backends.mysql',**

**'USER': 'root',**

**'NAME': '<the database name>',**

**'PASSWORD': 'your root password',**

**'HOST': 'localhost',**

**'PORT': '3306',**

**}**

**}**

1. Migrate (in the directory containing manage.py, project directory)
   1. python manage.py migrate
2. Create the application
   1. python manage.py startapp <application name>
3. Configure the application (in this example, the application name is posapp)
   1. Project Level setting.py - include name of app in “Installed apps”

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'posapp'

]

* 1. Project Level urls.py - include urls.py of the app

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('posapp.urls'))

]

1. Create view
   1. View function to load hello\_world.html

from django.shortcuts import render

def hello\_world(request):

return render(request, 'posapp/hello\_world.html')

1. Create template directory and html files (in the app directory)
   1. create a templates directory
   2. inside the templates directory create another directory with the same name as you app
   3. Copy the html files into this directory
2. Create a static directory for static files (in the app directory)
   1. create a static directory
   2. copy the static files there
3. Adjust the HTML file so that it loads properly when the django url is typed
   1. {% load static %}
   2. Convert the links to {% static ‘</file location/>’ %}

**Django Review: Setup**

**Activity**

Create/code a user interface for the CRUD module of items that are sold. The module need not be working yet but make sure that the user interface has provisions for:

1. Displaying all available items
2. Adding a new item (will need a form for this)
3. Deleting a specific item
4. Updating the details of a specific item (could reuse the form for adding a new item)