GE02 Django Portfolio App Set Up

**Purpose**:

* Communicate effectively in a variety of professional contexts within a team, doing oral or written presentations, and creating technical documents.
* Function effectively as a member/leader of a team engaged in scrums while participating in different roles
* Apply various tools and agile principles utilizing concepts (scrum, behavior-driven development, pair programming , version control
* Apply computer science theory such as utilizing design patterns for software architecture, higher-order functions, metaprogramming, to improve the maintainability, modularity and reusability to build a SaaS application

**Effort**: Collaborative Assignment see [CS3300 Academic Integrity](https://docs.google.com/document/d/1cORsFi1YrqW5ChfJu0G67Fjm8HwEMse47DVqXfEn2n4/edit#heading=h.w1yj4lpdz8sh)

**Points**: 40 (**see rubric in canvas**)

**Deliverables**: Each person submits their own word/pdf document of the GE with their own answers. You will also submit a separate document for your professional communication of your learnings so far. DO NOT SUBMIT A ZIP FILE.

**Due Date**: See Canvas

[1 Team Information](#_heading=h.heplw3609se6)

[2 Set Up Portfolio App](#_heading=h.786x4ldpj26k)

[2.1 Create Virtual Environment and Django Project](#_heading=h.f1aa5twnb3ew)

[2.2 Create Local Git and Github Repository](#_heading=h.mow0vh1kshn8)

[2.3 Create Portfolio App](#_heading=h.2d7mqrmj40j8)

[2.4 Define URI path and view](#_heading=h.emfqpx5ewz97)

[2.5 Create HTML Template](#_heading=h.pvce2la26m1o)

[2.6 Add static files](#_heading=h.rlcgtluuhjy)

[3 Apply Technologies](#_heading=h.fxmyl9f070dm)

[Professional Written Communication of Technical Content](#_heading=h.e3n5kf230hzu)

**Description:** Explore setting up your first SaaS app using the django framework to build a portfolio app by reading documentation, troubleshooting, collaborating in your team and asking detailed questions when you need help outside of your team. Start building your understanding of Client-Server model, HTTP, URIs when developing SaaS apps.

Try setting up your environment on your own first. If an issue arises, spend 20 minutes trying to troubleshoot on your own. Then collaborate with your team. If no one on the team can help then reach out to the appropriate discord channel . Provide the following details if you want someone to help you.

* What step did you do when something went wrong?
* What have you tried so far?
* What resources did you already use to problem solve?

# 1 Team Information

Read your assignment for the GE02 Team Roles in Canvas before you begin.

|  |
| --- |
| What is your role?  **Scrum Team Member** |
| What is your responsibility for this GE02 Sprint’s Documentation?  My responsibility is to document on 2.2 and report any successes, failures, or general things to do in order to succeed. |

# 2 Set Up Portfolio App

Some Resources to help you get started

[Setting up a Django development environment](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/development_environment) and [Django Getting Started](https://www.w3schools.com/django/django_getstarted.php) discuss installations depending on OS.

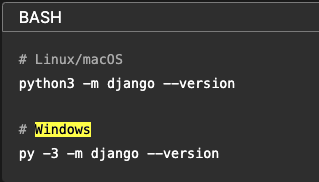
I tried both of these virtual environments but find the first one to be easier to manage and is the one shown in steps below

* <https://python.land/virtual-environments/virtualenv> - this is the one that I included in this documention
* <https://virtualenvwrapper.readthedocs.io/en/latest/index.html>

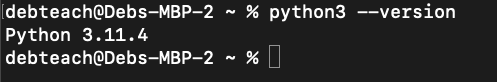
# 

## 2.1 Create Virtual Environment and Django Project

These instructions show examples of working on a Mac for the commands. Note how you must run the commands on your environment.

* + 

1. Open the command/terminal window and check your python version. You should be using 3.11.



1. Create cs3300 folder then create portfolio folder for GE project

mkdir cs3300

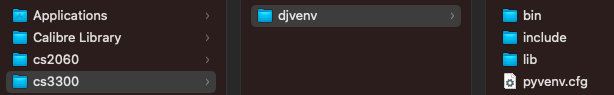
cd cs3300

mkdir portfolio

cd portfolio

1. Create virtual env

python3 -m venv djvenv

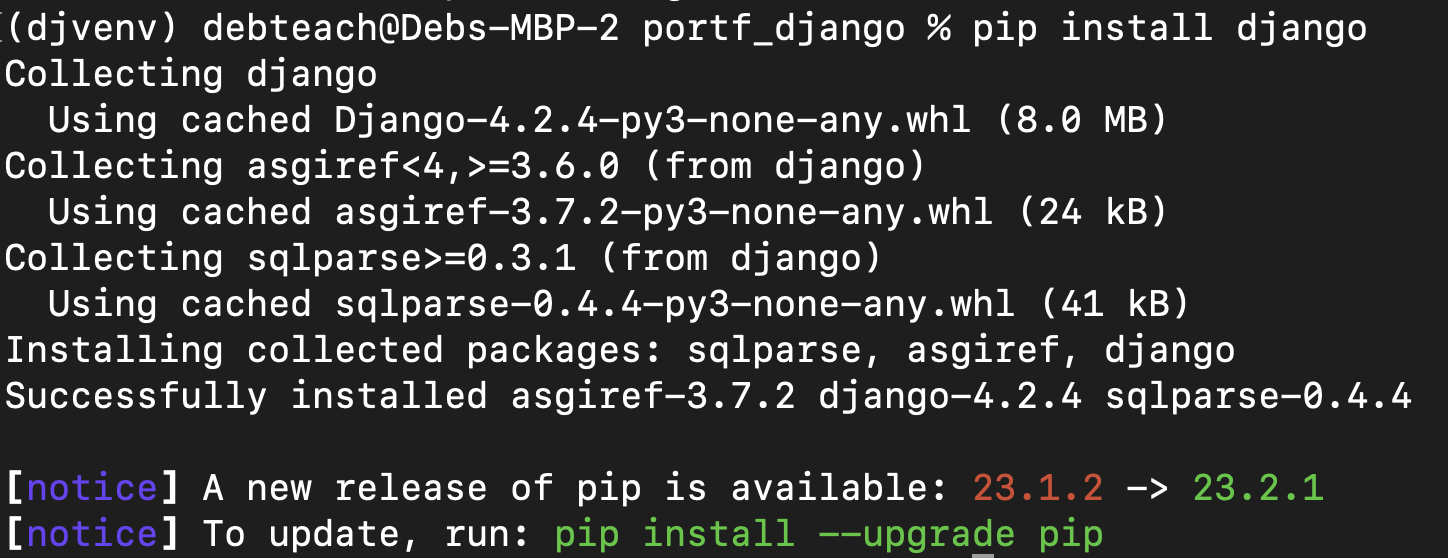


1. Activate virtual environment

source djvenv/bin/activate

1. Install django in virtual environment

pip install django



1. Upgrade pip

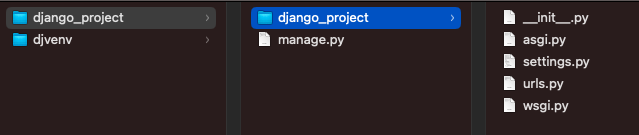
python3 -m pip install --upgrade pip

1. Create django project

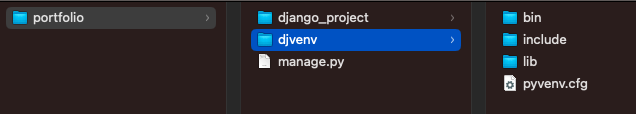
django-admin startproject django\_project

1. Reorder directory structure for ease of use

Change from



to



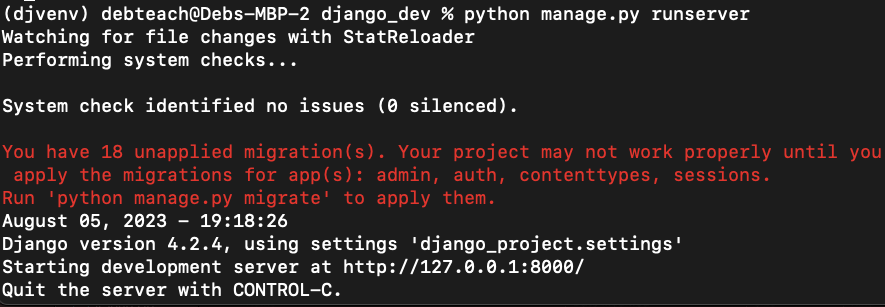
mv django\_project/manage.py ./

mv django\_project/django\_project/\* django\_project

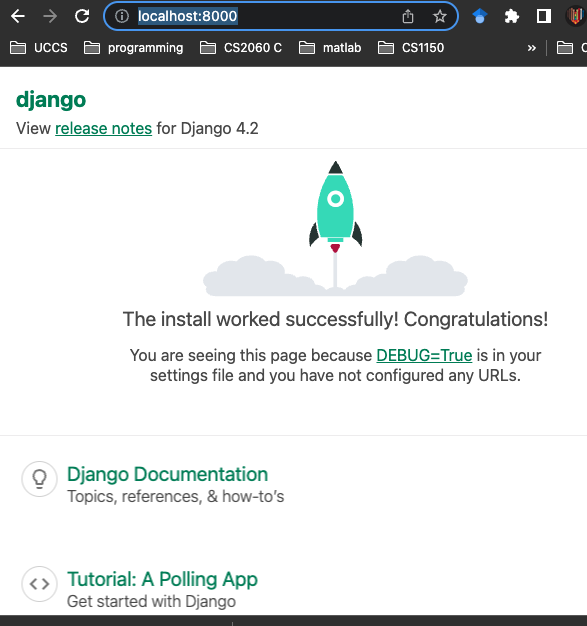
rm -r django\_project/django\_project/

1. Run server and ignore migration warnings.

python manage.py runserver



1. See if the installation worked by going to <http://localhost:8000/> and you should see. Celebrate!



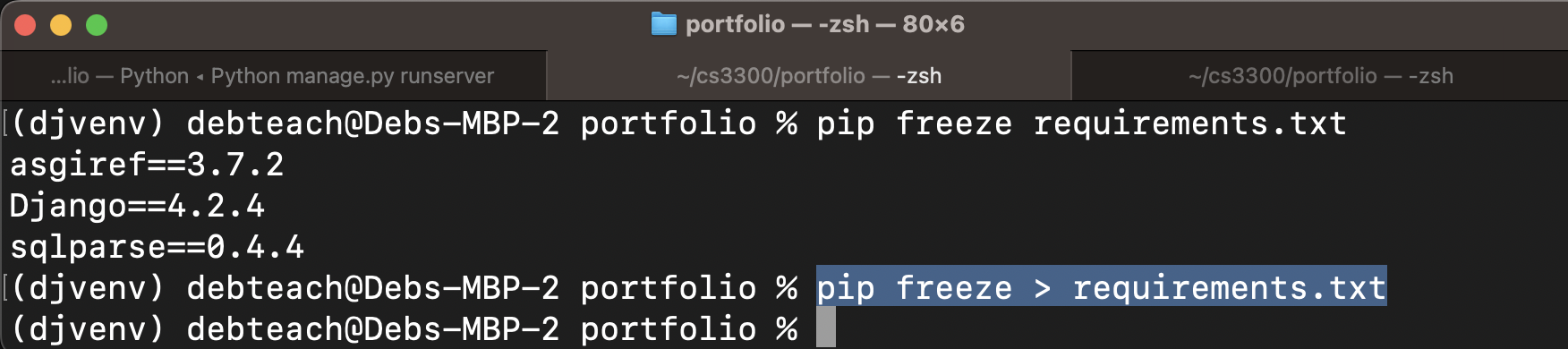
1. Open the second terminal and activate the virtual environment.

Tip: I like to have 3 terminal windows tabbed: one for

* venv to run commands
* venv starting server and stopping server
* git/gihub commands

1. From virtual environment create requirements file of what is installed [Check all installed Python packages with pip list/freeze](https://note.nkmk.me/en/python-pip-list-freeze/)

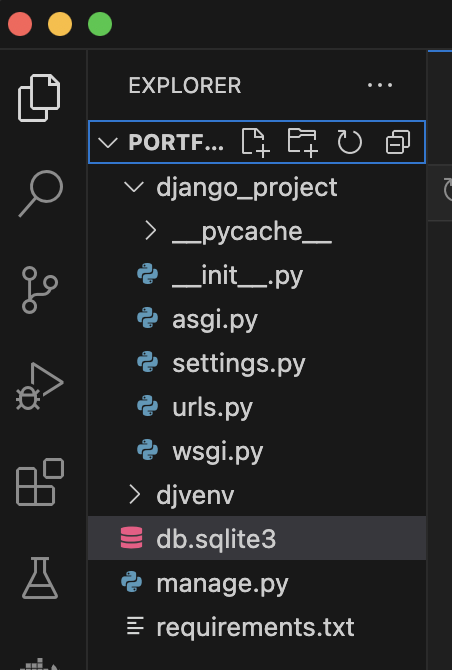
pip freeze > requirements.txt



**Manage Project in VS Code**

You can use a different IDE but I will be demonstrating with VS Code.

Open the project in Visual Studio Code: In Visual Studio Code, go to `File -> Open Folder` and select the folder that contains your Django project.



## 2.2 Create Local Git and Github Repository

Once your app is working you want to create a GitHub Repository from an existing Local directory/folder in your computer. This is different from the team repository. This initial versioning will be on the main branch. Search for resources to help you create your repos from an existing local directory. Read information below for using git and github for versioning your django project

* Store your code in github as a private repository.
* You will need to add a git ignore file so that when committing it doesn’t check certain files in. See [Ignoring files - GitHub Docs](https://docs.github.com/en/get-started/getting-started-with-git/ignoring-files) and then you can use this example for [Python.gitignore](https://github.com/github/gitignore/blob/main/Python.gitignore). Here are some updates I made to my file
  + For # Environments I added /djvenv to ignore my virtual environment
  + the mac I added .DS\_Store to my gitignore file.
* **Note:** Deb made mistakes when doing the gitignore and had to search for help on why it didn’t ignore something. <https://stackoverflow.com/questions/39933600/how-to-ignore-folder-in-github-correctly>

## 2.3 Create Portfolio App

You will add the next part for your portfolio app in a branch called Sprint01.

Here you will create a portfolio app for your project. Remember your indenting when editing python code.

1. Activate your virtual environment
2. From the django-portfolio directory run the following command to create the portfolio app. Explore what is created in the portfolio\_app folder (open in VS code or your IDE).

django-admin startapp portfolio\_app

1. Configure Settings: Edit your django\_project's settings.py file, add your app to the INSTALLED\_APPS list and add support for authenticating users:

INSTALLED\_APPS = [

# ...

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'Django.contrib.staticfiles',

# Add your app name here

'portfolio\_app',

]

# Add support for authenticating users

AUTHENTICATION\_BACKENDS = [

'django.contrib.auth.backends.ModelBackend',

]

## 2.4 Define URI path and view

Resource: Watch [URLS and Views | Django Framework (3.0) Crash Course Tutorials (pt 2)](https://www.youtube.com/watch?v=QvTyqta3OJo&list=PL-51WBLyFTg2vW-_6XBoUpE7vpmoR3ztO&index=3&t=1s). Read [Django Tutorial Part 5: Creating our home page - Learn web development](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Home_page) and [Django Tutorial Part 5: Creating our home page - Learn web development](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Home_page) just to get some background information but the project directions are below.

* Open django\_project urls.py to add a path below the admin path to include the specific urls that will be created in the the portfolio\_app urls.py. You need to import include.

**from django.contrib import admin**

**from django.urls import path, include**

urlpatterns = [

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

#connect path to portfolio\_app urls

path('', include('portfolio\_app.urls')),

]

* Update portfolio\_app/views.py views by defining the following view for the home page.

from django.shortcuts import render

from django.http import HttpResponse

# Create your views here.

def index(request):

# Render the HTML template index.html with the data in the context variable.

return HttpResponse('home page')

* Create urls.py file in portfolio\_app that contains a path to the defined view

from django.urls import path

from . import views

urlpatterns = [

#path function defines a url pattern

#'' is empty to represent based path to app

# views.index is the function defined in views.py

# name='index' parameter is to dynamically create url

# example in html <a href="{% url 'index' %}">Home</a>.

path('', views.index, name='index'),

]

* Run Start server command

python manage.py runserver

* Open <http://127.0.0.1:8000> and you should see home page



* Update portfolio\_app/views.py views by defining the following view for the home page.

from django.shortcuts import render

from django.http import HttpResponse

# Create your views here.

def index(request):

# Render index.html

return render( request, 'portfolio\_app/index.html')

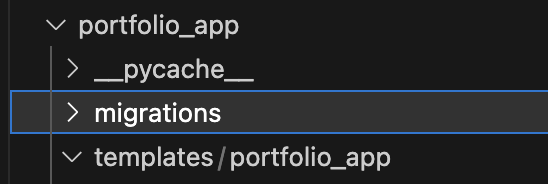
* Open <http://127.0.0.1:8000> What do you think is the reason for the issue? In the next part you will fix it.
* Version on your sprint01 branch and update your remote repository.

## 2.5 Create HTML Template

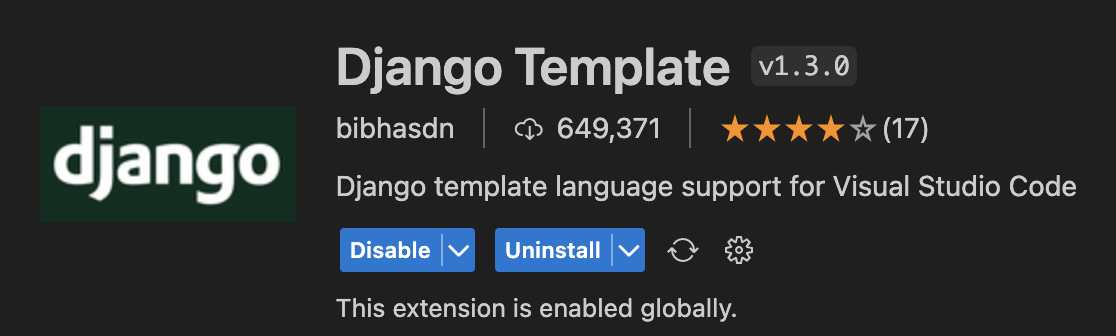
Django has a certain configuration for finding html templates. Set up the following.

templates

1. Create folder called templates in portfolio\_app
2. Create folder called portfolio\_app in templates



1. Create a **base\_template.html** file in the **templates/portfolio\_app** folder that will contain the html needed on every page, such as the navigation menu. Paste the following into it. Notice the django [**block** tag](https://vegibit.com/django-block-content/)s {% block content %}{% endblock %} that allow for inheritance. The HTML for the nav bar is from [Navbar · Bootstrap v5.3](https://getbootstrap.com/docs/5.3/components/navbar/). Later you will learn how to implement bootstrap to create an html page. Do not worry if you do not understand the HTML code given to you below.
2. {% load static %}
3. <!DOCTYPE html>
4. <html lang="en">
5. <head>
6. <title>UCCS CS Students</title>
7. <meta charset="utf-8">
8. <meta name="viewport" content="width=device-width, initial-scale=1">
9. <title>Bootstrap demo</title>
10. <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">
11. </head>
12. <body>
13. <div class="container-fluid">
14. <!-- Navbar-->
15. <nav class="navbar navbar-expand-lg bg-body-tertiary">
16. <img src="{% static 'images/uccs\_logo.gif' %}">
17. <div class="container-fluid">
18. <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
19. <span class="navbar-toggler-icon"></span>
20. </button>
21. <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
22. <div class="navbar-nav">
23. <!-- {% url 'index' %} is defined in url path to dynamically create url -->
24. <a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>
25. <a class="nav-link" href="#">Menu 2</a>
26. <a class="nav-link" href="#">Menu 3</a>
27. {% if user.is\_authenticated %}
28. <a class="nav-link" href="{% url 'logout' %}?next={{ request.path }}">Logout {{user}}</a>
29. {% else %}
30. <a class="nav-link" href="{% url 'login' %}?next={{ request.path }}">Login</a>
31. {% endif %}
32. </div>
33. </div>
34. </div>
35. </nav>
36. <div class="col-sm-10">
37. <!-- add block content from html template -->
38. {% block content %}
39. {% endblock %}
40. </div>
41. </div>
42. </div>
43. </body>
44. </html>

I suggest exploring how to format your code easily. For example, I am using the extensiontion Django Template in VS code. 

When I set it up it didn’t work at first so I found this solution and it worked. <https://stackoverflow.com/questions/42170561/vscode-html-autoformat-on-django-template>

1. Create an **index.html** file in the templates/portfolio\_app folder to be the home page for the app.

<!-- inherit from base.html-->

{% extends "portfolio\_app/base\_template.html" %}

<!-- Replace block content in base\_template.html -->

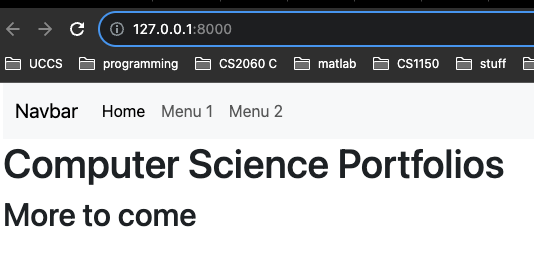
{% block content %}

<h1>Computer Science Portfolios</h1>

<h2>More to come from [and your name goes here]</h2>

{% endblock %}

1. Restart our server and open <http://127.0.0.1:8000> and you should see your index.html using the base\_template.html. For example.



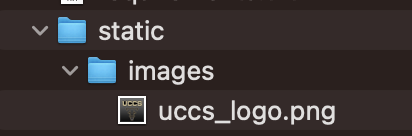
1. Version on your sprint01 branch and update your remote repository.

## 2.6 Add static files

Resources:

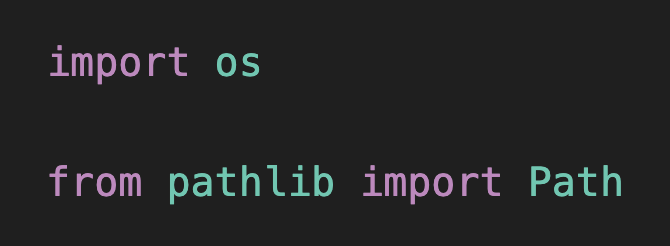
* [Static Files & Images | Django Framework (3.0) Crash Course Tutorials (pt 4)](https://www.youtube.com/watch?v=kqyfEz7TNI0&list=PL-51WBLyFTg2vW-_6XBoUpE7vpmoR3ztO&index=4)
* [os.path — Common pathname manipulations — Python 3.11.5 documentation](https://docs.python.org/3/library/os.path.html)

1. Create folder structure to store static files
   1. Make a directory called “static” in portfolio folder
   2. Create a folder called “images” in static
   3. Add a logo. I downloaded [UCCS logo](https://brand.uccs.edu/sites/g/files/kjihxj1416/files/inline-images/uccs-signature-email.gif) and called it **uccs\_logo.gif**



1. Update settings.py so Django can find the static files.
   1. At top of settings.py file add

import os



* 1. Then add the following near the bottom of the settings file to tell Django where to find folder based on building the path from the BASE\_DIR defined in the settings.py file already

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC\_URL = 'static/'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, 'static')

]

MEDIA\_URL = '/images/'

1. Add image information to base template
   1. At top of base\_template.html add {% load static %} above <!DOCTYPE html>

{% load static %}

<!DOCTYPE html>

* 1. Replace

<a class="navbar-brand" href="#">Navbar</a>

With

<img src="{% static 'images/uccs\_logo.gif' %}">

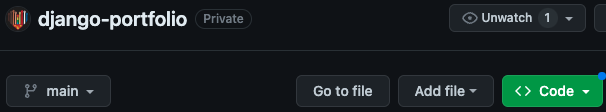
1. Open <http://127.0.0.1:8000> and you should see the logo.
2. Version on your sprint01 branch and update your remote repository.
3. Update your main branch by merging it with sprint01 branch code and tag the code as GE02. Do not delete sprint01 branch.

# 3 Apply Technologies

These are individual responses and not team responses.

1. Put a screenshot of your home page showing the UCCS image.

|  |
| --- |
|  |

1. Go to your github remote repository, click and take a screenshot of your code files and directories that are in the main branch

|  |
| --- |
| Not exactly sure what clicking on “code” does, but this is my code files and directories in history. |

1. Go to your github remote repository, and take a screenshot showing you have a main branch and sprint01 branch

|  |
| --- |
|  |

1. Take a screenshot showing your two branches in github - main and sprint01.

|  |
| --- |
|  |

1. What is the requirements.txt file you created useful for? When should you update it? How do you update it?

|  |
| --- |
| The requirements.txt file is used for displaying what dependencies the software uses and what versions should be used. This should be updated whenever you update/jump-back in versions for any apps/softwares and when new apps or dependencies are added. You update it with pip freeze, which creates a snapshot of all programs currently in use and directs that text output to a file |

1. Explain the python virtual environment you set up using venv. Include what it is and at least 2 reasons why you should use a virtual environment.

|  |
| --- |
| The python virtual environment is a “world” where code and any installed packages are made separate from any other projects on the drive. By creating an environment, you are creating a sectioned off area where any packages implemented by pip are set away from any other environment. You should use a virtual environment because it allows for project separation, making sure that apps, packages, or files from separate environments don’t interact. Another reason to use a virtual environment is environments are easy to reproduce. Since all files, packages, etc. are tied together, it makes it easy to “copy and paste” a project from one computer to the next. |

1. List any issues encountered by you or someone else and how you approached resolving them.

|  |
| --- |
| 1. I had issues setting up Django & testing my version at the beginning of the GE. I approched this issue by searching online and found that, when installing python, my environment variables in my computer were not set up properly. 2. I forgot to include my .gitignore and David was able to give me the correct files to not include. 3. I had an issue with the login/logout set up. David was able to explain the issue to me and give me advice on the multiple avenues I could take to fix it. I was also made aware that this issue is online, so I could’ve googled it aswell, so I feel a little bad that I asked David. |

# 4 Professional Communication of Technical Content

This is just a start of your understanding of different architectures, design patterns and concepts used when learning a framework to build SaaS apps.The goal is to see how you grow over the semester in your learnings.

Create a separate professional document (word or pdf) where main ideas are clearly presented with supporting evidence.

* Present and overview of the concepts about client server, url, HTTP Requests, Django MVT(model, view, template) in relation to what you set up in Django in this GE.
* Include resources used such as lectures, books, and your own research. You do not need to include citations in the writing but list the resources at the end.
* This should contain technical vocabulary,  **your own** words, code snippets, diagrams (digitally or drawn on paper).Do not use an image that someone else created.
* About 1 to 1 ½ pages. Should not be more than two pages.