GE02 Django Portfolio 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 explanation of question 3 in section 3 learnings and reflections. DO NOT SUBMIT A ZIP FILE.

**Due Date**: See Canvas

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

[2 Create Virtual Environment and Django Project](#_heading=h.786x4ldpj26k)

[Manage Project in VS Code](#_heading=h.9gomil77gszl)

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

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

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

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

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

[3 Learnings and Reflections](#_heading=h.fxmyl9f070dm)

[4 Agile Team Member Roles Only Need to Complete](#_heading=h.xs4kbrqtyc4y)

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 tech lead can reach out to the appropriate discord channel (#django has been added). Think about how to provide details to ask a question to help others be efficient with helping you.

# 1 Team Information

Team name:

Team members and roles - If you should not have the same role as last GE. Remember you are all taking turns in each role.

| **Roles and Responsibilities** | **Names** |
| --- | --- |
| **Scrum Coach**  - see [Team Scrum Coach Role](https://docs.google.com/document/d/1GR04En225yi5699ZT5YfhHyLCjrdrBgr/edit) assignment on canvas.  - submit this and the scrum coach assignment  - communicate in teams-scrum-coach channel in discord if you have questions on the assignment |  |
| Tech Lead Django  - see[Team Technical Lead Role](https://docs.google.com/document/d/1Xx4bqHQNuvfuxozaaf5tZc7bhP_dUdlc/edit#heading=h.86wae0otsxbo) assignment on canvas.  - technical documentation can be stored in the team github  - communicate in discord django area if you have issues that others might be able to help resolve |  |
| Team Member  - Participates in team meetings and in discord  - Help with technical documentation  - Help with troubleshooting issues and understanding concepts |  |
| Team Member  - Participates in team meetings and in discord  - Help with technical documentation  - Help with troubleshooting issues and understanding concepts |  |
| Team Member  - Participates in team meetings in discord  - Help with technical documentation  - Help with troubleshooting issues and understanding concepts |  |

Some Resources to help you get started

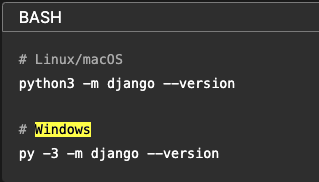
* [Setting up a Django development environment](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/development_environment) and [Django Getting Started](https://www.w3schools.com/django/django_getstarted.php) discuss installations depending on OS.
  + I tried both of these virtual environments but find the first one to be easier to manage and is the one shown in documentation
    - <https://python.land/virtual-environments/virtualenv> - this is the one that I included in this documention
    - <https://virtualenvwrapper.readthedocs.io/en/latest/index.html>
* [Check all installed Python packages with pip list/freeze](https://note.nkmk.me/en/python-pip-list-freeze/)

Individually add your own resources below that were helpful in setting up a django portfolio app. This will be helpful as your team collaborates to keep technical documentation.

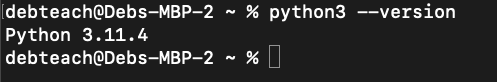
| **Resources** |
| --- |

# 2 Create Virtual Environment and Django Project

These instructions show examples of working on a Mac

* + 

1. From the preparation in L02 you should have installed python, VScode, git and have a github repository. Open the command/terminal window and check your python version.



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

mkdir cs3300

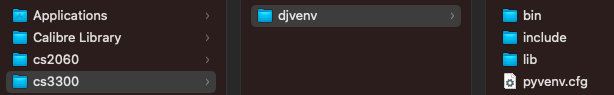
cd cs3300

mkdir portfolio

cd portfolio

1. Create virtual env

python3 -m venv djvenv

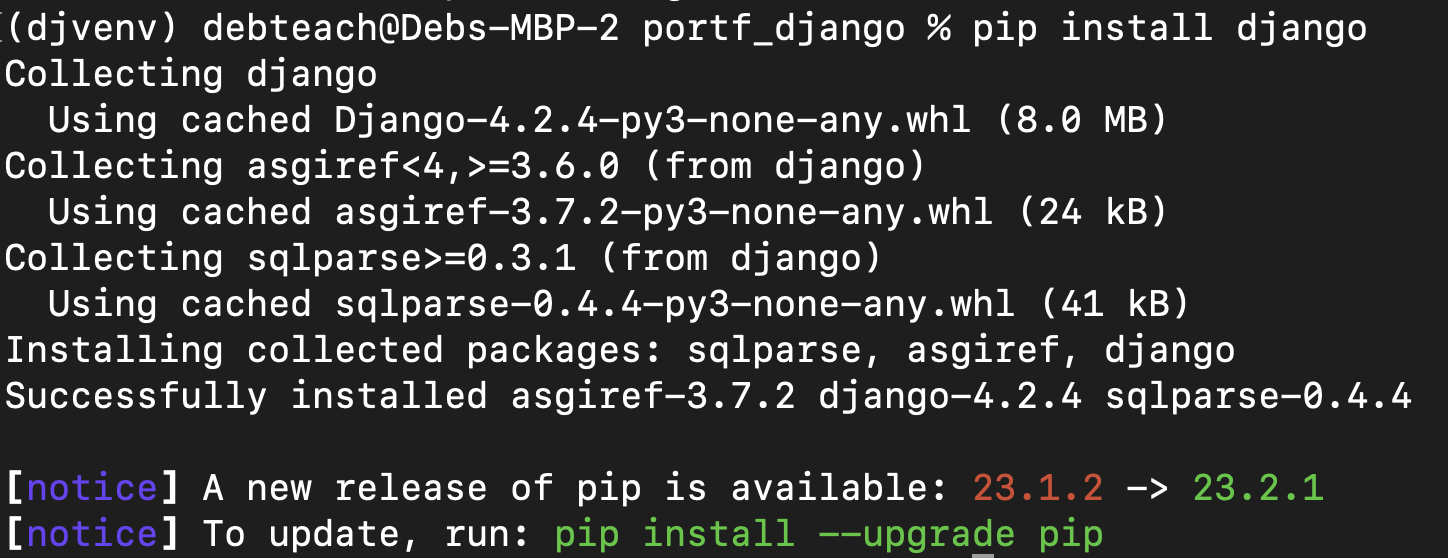


1. Activate virtual environment

source djvenv/bin/activate

1. Install django in virtual environment

pip install django



1. Upgrade pip

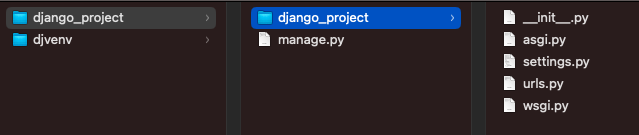
python3 -m pip install --upgrade pip

1. Create django project

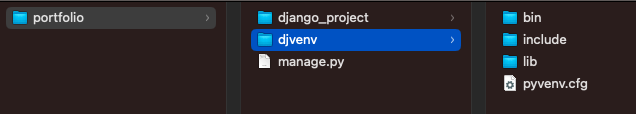
django-admin startproject django\_project

1. Reorder directory structure for ease of use

Change from



to



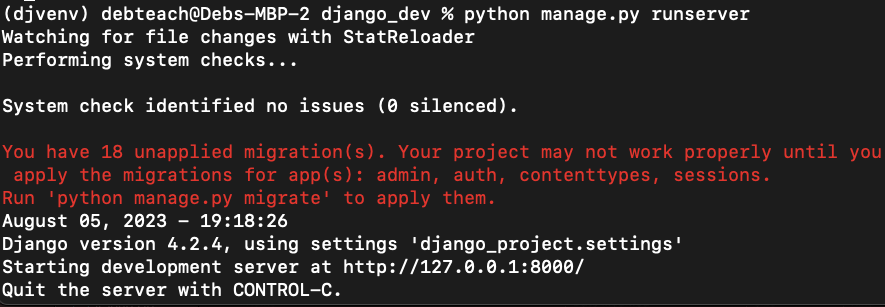
mv django\_project/manage.py ./

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

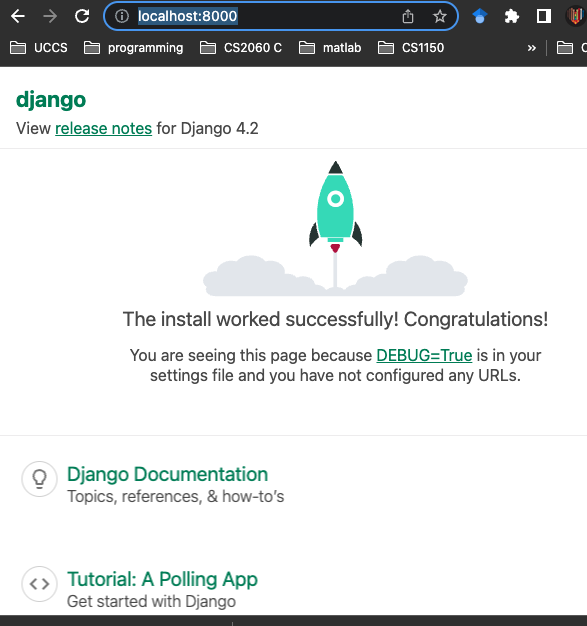
rm -r django\_project/django\_project/

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!



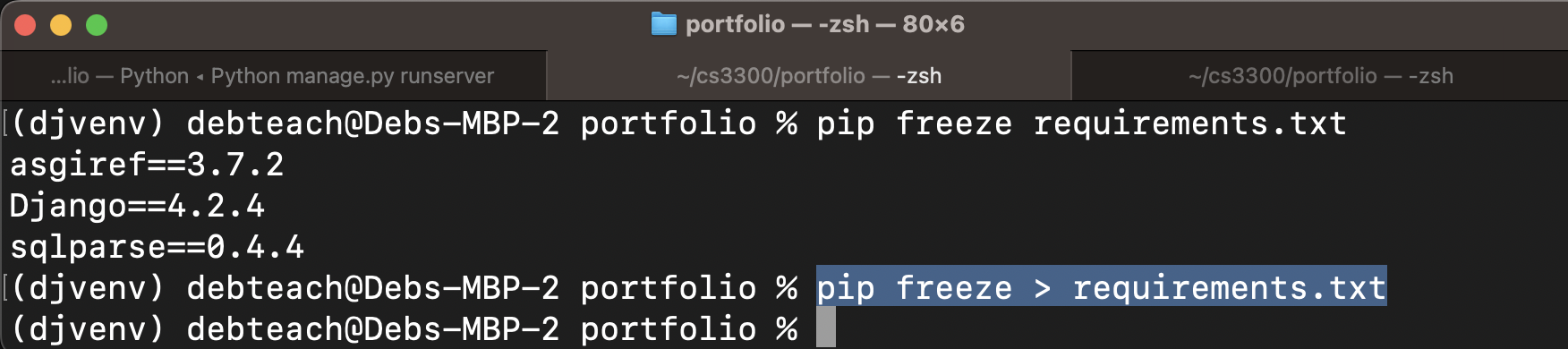
1. Open the second terminal and activate the virtual environment.

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

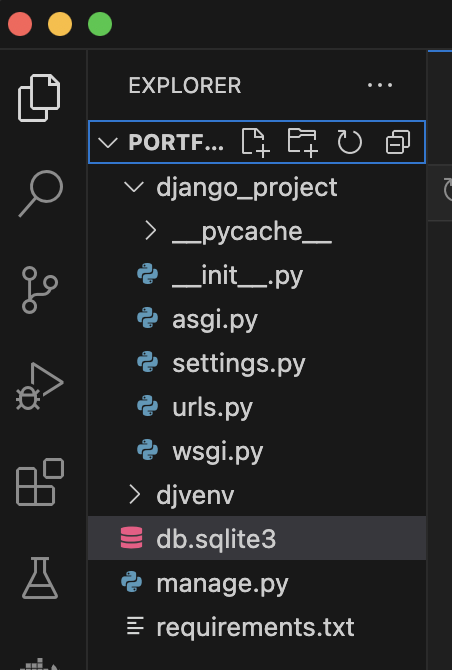
pip freeze > requirements.txt



## Manage Project in VS Code

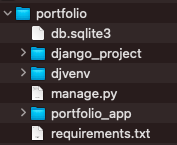
You can use a different IDE but I will be demonstrating with VS Code. See the preparation in L02 on setting up VS Code.

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.



## 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 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.
* Keep track of your git commands used to fill in the information below.
  + Note: Only paste command and not the entire command line prompt. For example put **git status** instead of **debteach@Debs-MBP-2 django-portfolio % git status**
* **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>

List and explain the commands you used to add the current files and commit base project to the main branch and store them in a private remote github repository.

Paste the command(s) and explain each command in your own words. Do not copy and paste from other resource and do not use AI tools such as ChatGPT

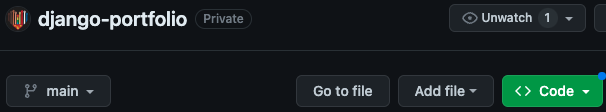
For example my first command:

From the local portfolio folder create a git repository for my local django project folder on main branch

git init -b main

1. Put your explanations and commands below.

|  |
| --- |

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

|  |
| --- |

You will add the next part for your portfolio app in a branch called iter01

Paste and explain the command(s) to create a branch and work on that branch on your local repository.

|  |
| --- |

Take a screenshot of your terminal window showing you are on the iter01 branch.

|  |
| --- |

List any issues you encountered and any resources that helped resolve them

|  |
| --- |

## Create Portfolio App

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',

]

1. Version on your iter01 branch and update your remote repository. Put commands used below and take a screenshot showing your two branches in github.

|  |
| --- |

## 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 you do not need to follow these steps. The project directions are below.

1. 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')),

]

1. 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')

1. 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'),

]

1. Run Start server command

python manage.py runserver

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



1. Explain the concepts about client server, url, http requests, Django MVT(model, view, template) using examples from your app set up and the code you just added. You can use screenshots. Again this is just a start of your understanding and the goal is to see how you grow over the semester in your understanding. Create at least one diagram digitally or on paper to help with your explanation. Do not take an image that someone else created.

|  |
| --- |

1. 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')

1. Open <http://127.0.0.1:8000> and paste a screenshot of the top portion of the page stating the issue. Then explain why you think this is happening.

| Screenshot of what is displayed: |
| --- |
| What do you think is the reason for the issue? In the next part you will fix it. |

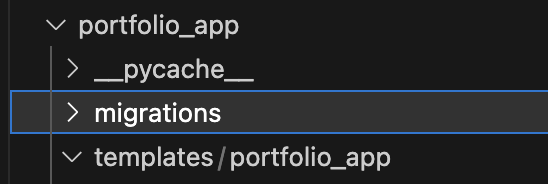
1. Version on your iter01 branch and update your remote repository.

## 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.

<!DOCTYPE html>

<html lang="en">

<head>

<title>UCCS CS Students</title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Bootstrap demo</title>

<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" />

</head>

<body>

<div class="container-fluid">

<!-- Navbar -->

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="container-fluid">

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

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Menu 1</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Menu 2</a>

</li>

</ul>

</div>

</div>

</nav>

<!-- add block content from html template -->

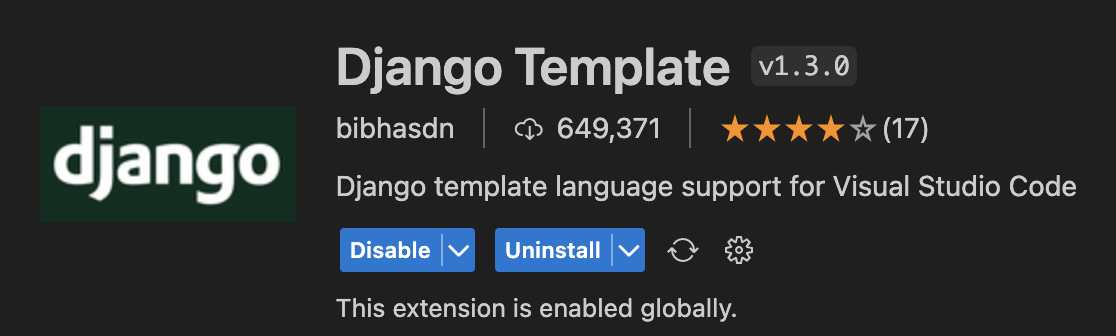
{% block content %}

{% endblock %}

</div>

</body>

</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 -->

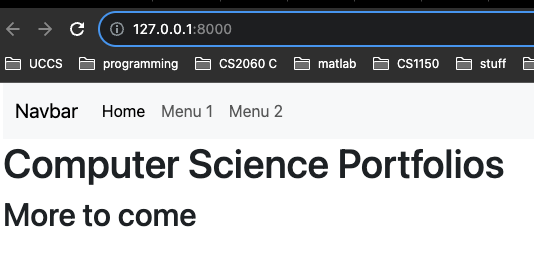
{% block content %}

<h1>Computer Science Portfolios</h1>

<h2>More to come</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 as shown below.



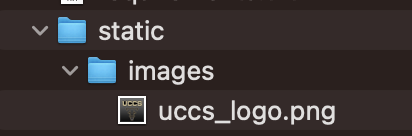
1. Version on your iter01 branch and update your remote repository.

## Add static files

Resources:

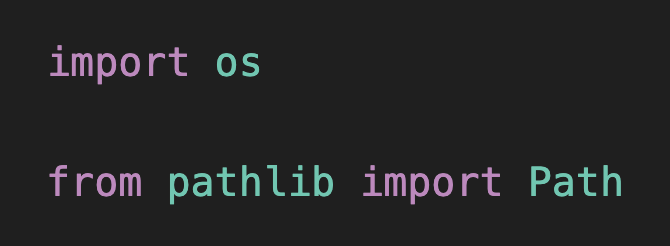
* [Static Files & Images | Django Framework (3.0) Crash Course Tutorials (pt 4)](https://www.youtube.com/watch?v=kqyfEz7TNI0&list=PL-51WBLyFTg2vW-_6XBoUpE7vpmoR3ztO&index=4)
* [os.path — Common pathname manipulations — Python 3.11.5 documentation](https://docs.python.org/3/library/os.path.html)

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

|  |
| --- |

1. Version on your iter01 branch and update your remote repository.
2. Update your main branch by merging it with iter01 branch code and tag the code as release 1.0. Do not delete iter01 branch

| Put the commands below you used to complete |
| --- |
| Put a screenshot of your github repository showing your tag |

# 3 Learnings and Reflections

Explanations - **provide elaboration of the concepts using supporting evidence and vocabulary.** Provide code snippets or your own drawings/electronic images that you created. Remember you are not an expert in these concepts but share your learnings so far.

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

|  |
| --- |

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.

|  |
| --- |

1. Explain HTTP communications and relate it to what you set up in Django. Use the lectures and your research to help you. This should contain **your own** words, code snippets, diagrams and include vocabulary to explain as if you are helping a new team member. **Upload as a separate document (word or pdf).**
2. Reflect on your individual work. How did it go compared to the first GE? What worked well in how you approached working on this GE? What might you want to do differently for the next part?

|  |
| --- |

1. Reflect on your team collaboration. How did it go compared to the first GE? What worked well in how your team approached working together? What might help improve your team for working together for the next part?

|  |
| --- |

# **4 Agile Team Member Roles Only Need to Complete**

**Team Member:**

Share how you contributed to the team

|  |
| --- |

Explain how it went for you being the team member. What was one of your strengths in this role and would be a growth area for you?

|  |
| --- |

Summarize any issues and resolutions encountered by the team.

|  |
| --- |