Technical Documentation

GE02

Git

Linking Local Repo and Github Repo

(Chandler Waller)

To link a previously created local repository and a Github repository you must first go to the github repository on the "Code" page. Upon arriving at the "Code" page you should see a url with a copy button. Copy the URL and return to your local repository. Inside the local repository use the command git remote add origin URL, replace URL with the URL. After doing this you will also need to run the command git push –set-upstream origin. After running these commands your local repository will be linked to the Github repository.

Deleting a Local Repository

(Chandler Waller)

To delete a local repository, you must remove the files entirely. The best way to do this would be to move everything you need out of the repository and then delete the folder in explorer. There are commands to do this through git and terminal, however, none of them seemed to work as well as this.

Git Reset

(Chandler Waller)

In case you accidentally add before creating the git ignore, and assuming you haven't committed anything yet you can use the command "git reset" to undo your add. If you have already committed the add then you will need to revert to the first git version. This can be done with the command "git reset –soft HEAD~1" which undoes the latest commit.

Git Ignore

(Chandler Waller)

To create a git ignore in windows use "notepad .gitignore." the additional dot will prevent an error from occurring and create a .gitignore file instead of a .txt. It is recommended that you add whatever you named your environment as to the git ignore so that you don't upload your entire environment.

Git checkout –b branch_name

(Chandler Waller)

The above command will create a new branch with the name "branch_name". This only works if "branch_name" is not already a branch. If it is a branch, you will simply switch to working in that branch.

Default Branch Reset

To change the default branch on git hub go to your github's settings and there should be a default branch setting on the middle of the page as it loads. Simply change it there.

Django Environment

Creating/Activating Virtual Environment

(Katie Miniter)

Use command ‘python3 -m venv djvenv’ for mac, or ‘py -3 -m venv djvenv’ on windows to create a virtual environment, which you can then activate with ‘source djvenv/bin/activate’ or ‘djvenv\Scripts\activate.bat’ for Windows. This will set your terminal to work inside the virtual environment, and allow you to work in Django, and you will have to activate the virtual environment before working on the app in the future.

Creating Your Django Project

(Katie Miniter)

You can install Django through your virtual environment with ‘pip install django’ and update further with ‘python3 -m pip install –upgrade pip’ or ‘py3 -m pip install –upgrade pip’ (note that since 5.0 django came out, you’ll have to be cautious around how far into the updates you want to go.)

Finally, use ‘django-admin startproject django_project’ to run the project as an admin to the system.

Setting up Personal Environment (Using Python 3.11)

Python must be installed in your terminal/PowerShell/command prompt to set up your person environment. You can do so on windows by typing “python” into your choice of terminal this should bring you to the Microsoft store where you can download the latest version of python.

Once python is installed you can run the command “python -m venv <directory_name> this will create a new directory for your virtual environment.

Activating and deactivating

(Chandler Waller)

Run one of the following commands assuming you are on windows to activate:

```
# In cmd.exe
venv\Scripts\activate.bat
# In PowerShell
Venv\Scripts\Activate.ps1
```

Simply type “deactivate” to deactivate the virtual environment

(In powershell you can delete a virtual environment with rm –r <directory_name>(should be similar in other terminals))

Run server

(Tyler Andrews)

While virtual environment is active and in project folder directory, use python manage.py runserver to activate server. Accessible by going to http://localhost:8000/.

```
(djvenv) C:\Users\Admin\Documents\GitHub\cs3300IndividualWork\cs3300\portfolio>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 13, 2024 - 15:25:29
Django version 4.2, using settings 'django_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Reverting Python Versions (In Powershell)

(Chandler Waller)

If you messed up like me here is a quick guide to go from python 3.12 to 3.11. First install python 3.11, you can do so manually through the microsoft store. Then go to settings->apps->installed apps and delete python 3.12. Go back to powershell and type “Python -v” the very last line that pops up should say 3.11.7 or something along those lines.

Django Lookup error/KeyError admin

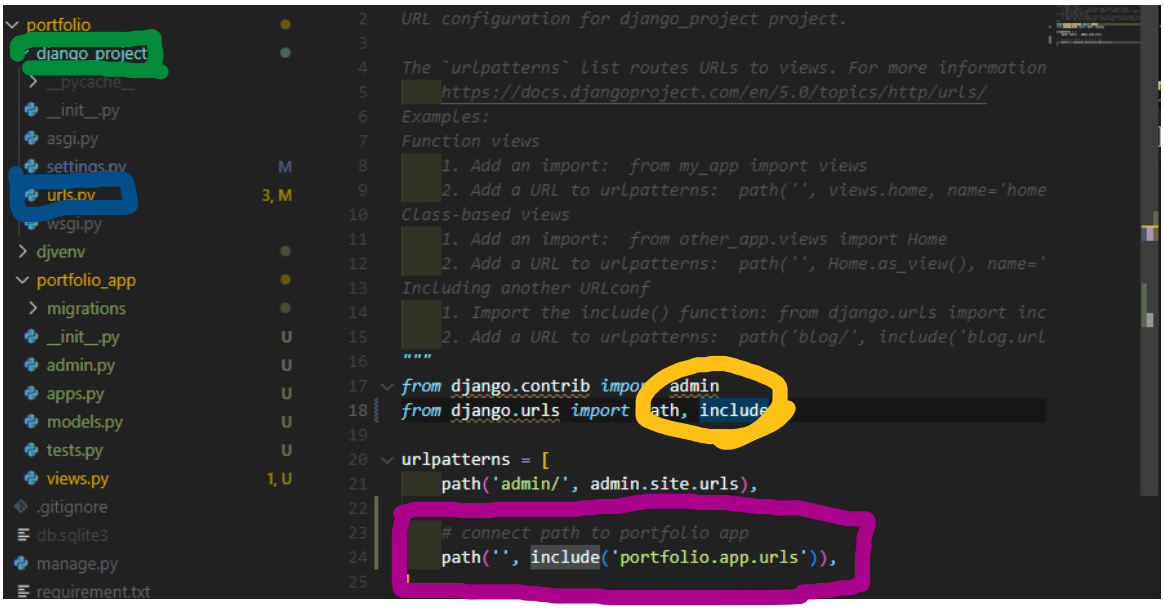
(Chandler Waller)

I am still unsure of exactly how to fix this error but I believe it comes from using “django” instead of “django-admin” when creating your project and app. This can be fixed by completely restating the GE and deleting all the work you have done so far. (Not a recommended fix, just the only way I thought to do it)

Defining URL path and view

(Juhan H)

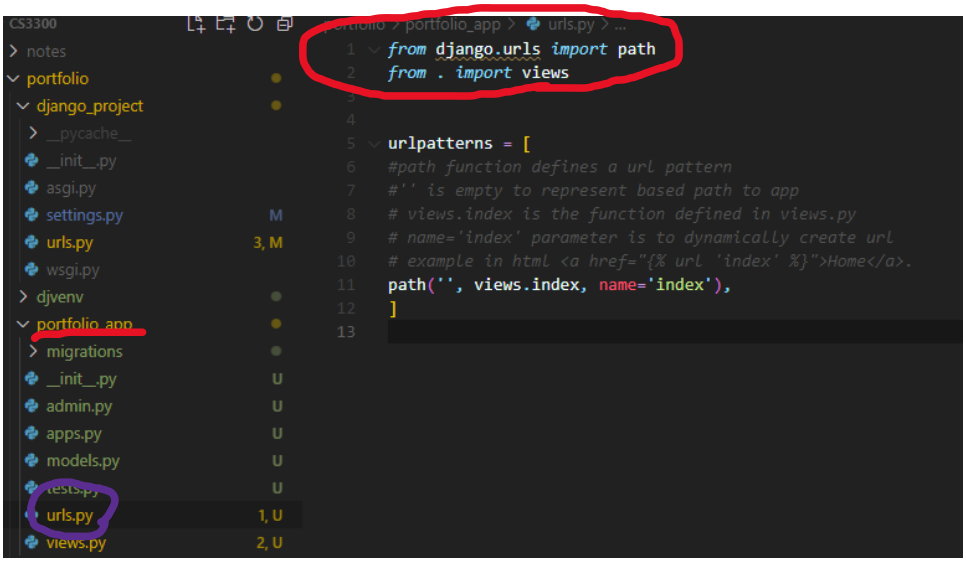
1. Navigate to “name of the project” urls.py and add a path to include specific URL that will be created in the “name of your app” urls.py.



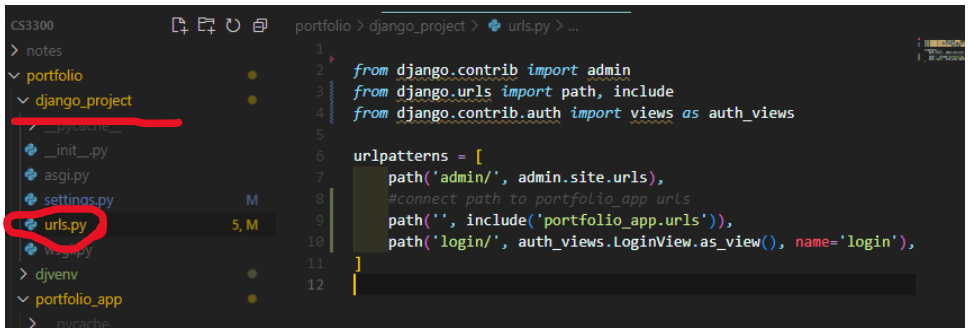
2. Update “name of your app”/views.py views by defining view and add these following:

```
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')
```

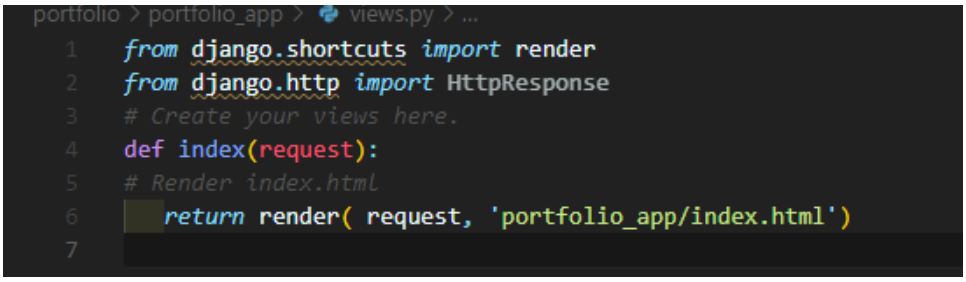
3. Create a urls.py in “name of your app” directory
    a. You must import path and views
    b. Path function defines the URL pattern
    c. Views.index = function defined in views.py



4. Go back to the project\urls.py and add the following:
    a. `from django.contrib.auth import views as auth_views`
    b. And in the urlpatterns:
        i. `path('login/', auth_views.LoginView.as_view(), name='login'),`



5. Update views.py in the “name of your app” directory to the following to get it ready for HTML



Creating HTML templates

(Juhan H)

1. Create a directory “whatever you want to name” in “name of your app” directory

2. Create a new directory named “name of your app” inside of “whatever you want to name” directory.
3. Create a base_template.html in that directory.
    a. This html page will have all the required items for all html pages. Such as a nav menu.
        i. Can't really explain this one, i just copied pasted

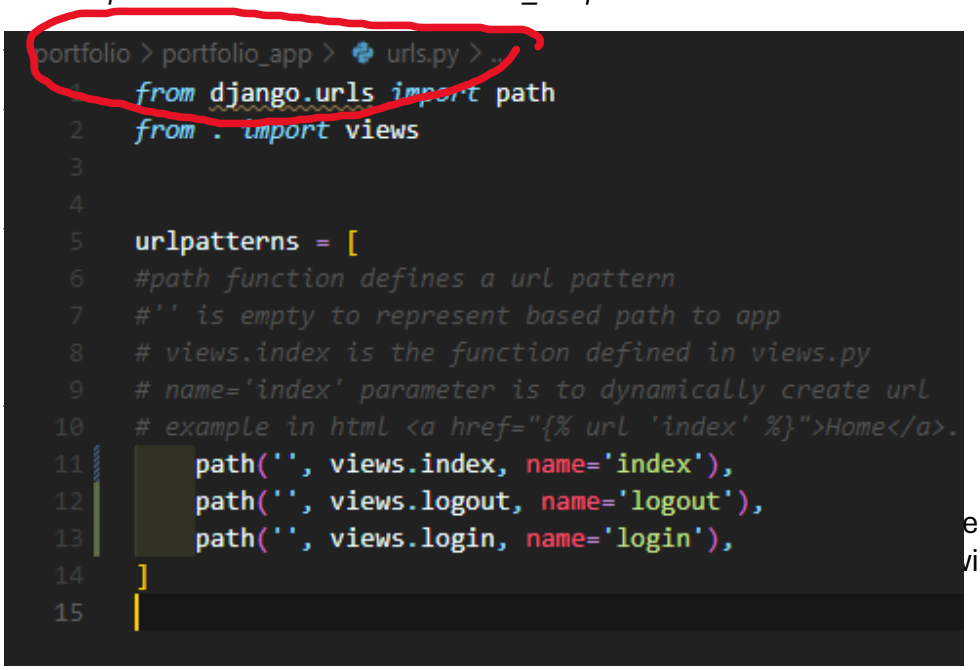
```
{% load static %}


<!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">

       <img src="{% static 'images/uccs_logo.gif' %}">

       <div class="container-fluid">
```

```
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-
expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
          <div class="navbar-nav">
            <!--  {% url 'index' %}  is defined in url path to dynamically
create url -->
            <a class="nav-link active" aria-current="page" href="{% url
'index' %}">Home</a>
            <a class="nav-link" href="#">Menu 2</a>
            <a class="nav-link" href="#">Menu 3</a>
            {% if user.is_authenticated %}
              <a class="nav-link" href="{% url
'logout' %}?next={{ request.path }}">Logout {{user}}</a>
            {% else %}
              <a class="nav-link" href="{% url
'login' %}?next={{ request.path }}">Login</a>
            {% endif %}
          </div>
        </div>
      </div>
    </nav>



      <div class="col-sm-10">
        <!-- add block content from html template -->
        {% block content %}
        {% endblock %}
      </div>
```

```
    </div>
  </div>
 </body>
</html>
```

4. Create index.html file in the templates folder to be the home page

```
<!-- inherit from base.html-->
{% extends "portfolio_app/base_template.html" %}

<!-- Replace block content in base_template.html -->
```



e following and
ving though)

Adding static files

(Tyler Andrews)

To add static files to the webpage, create a folder path: /static/images in the base directory of the project.

Update settings.py file in /django_project directory to include:

import os (more about os.path)

Near the end of settings.py add:

```
STATIC_URL = 'static/'

STATICFILES_DIRS = [

# tell django where static folder is from project base directory

os.path.join(BASE_DIR, 'static')]

# folder to find media and images

MEDIA_URL = '/images/'
```

Note* had issues with '/images/'. Used 'images/' instead.

In base_template.html add:

```
{% load static %}

<!DOCTYPE html>
```

DO THIS ABOVE <!DOCTYPE html>. This tells Django to load the static folder. Should be done for every template; there is only one at the moment.

Wherever an image is needed on the webpage use the following code template:

```
<img src="{% static 'images/<image name>.<gif, png, jpg,

etc.>' %}">
```

Problems and Solutions

Git Problems

Master branch isn't default

If you accidentally push on the second branch instead of merging, or somehow make the second branch the master branch you can fix it by visiting "Default Branch Reset".

_Default_Branch_Reset

GE03

For now, put links to useful info regarding GE03. Will delegate tasks soon™