Team 1-4

GitRDone

Table of Contents

Task 2.1 Create Virtual Enviornment …………………………………………………………2

Task 2.2 Create Local Git and Github Repository...…………………………………………6

Task 2.3 Create Portfolio App………………………………………………………………….7

Task 2.4 Define URL Path and View………………………………………………………….8

Task 2.5 Define HTML Templates...…………………………………………………………10

Task 2.6 Add Static Files……………………………………………………………………..11

Jagger Zlotoff – Tasks 2.1 & 2.2

Elijah Casillas – Tasks 2.3 & 2.4

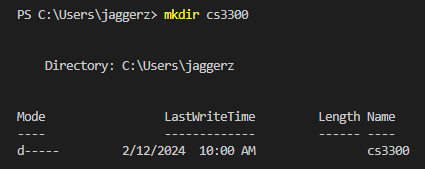
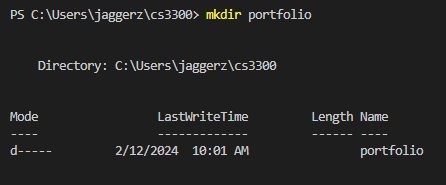
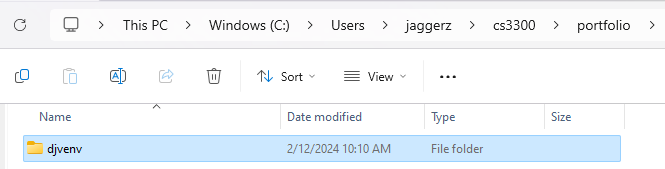
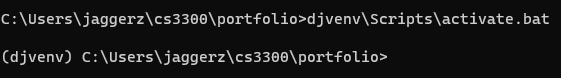
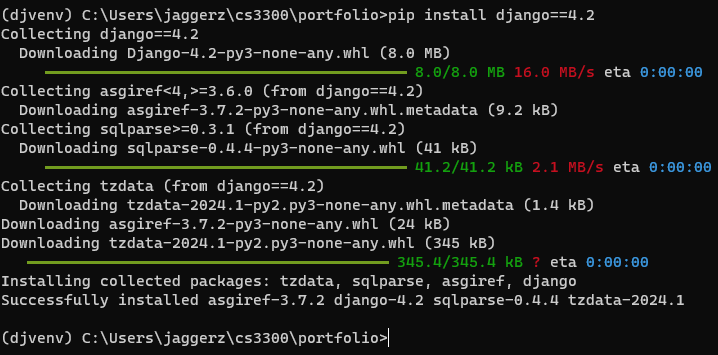
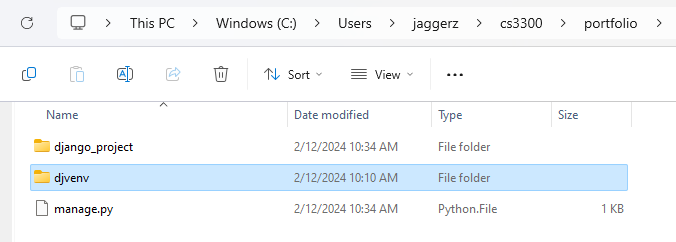
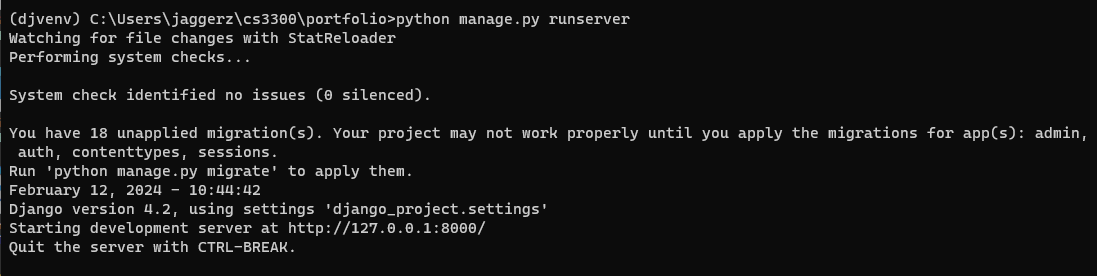
Lucas Haas – Task 2.5

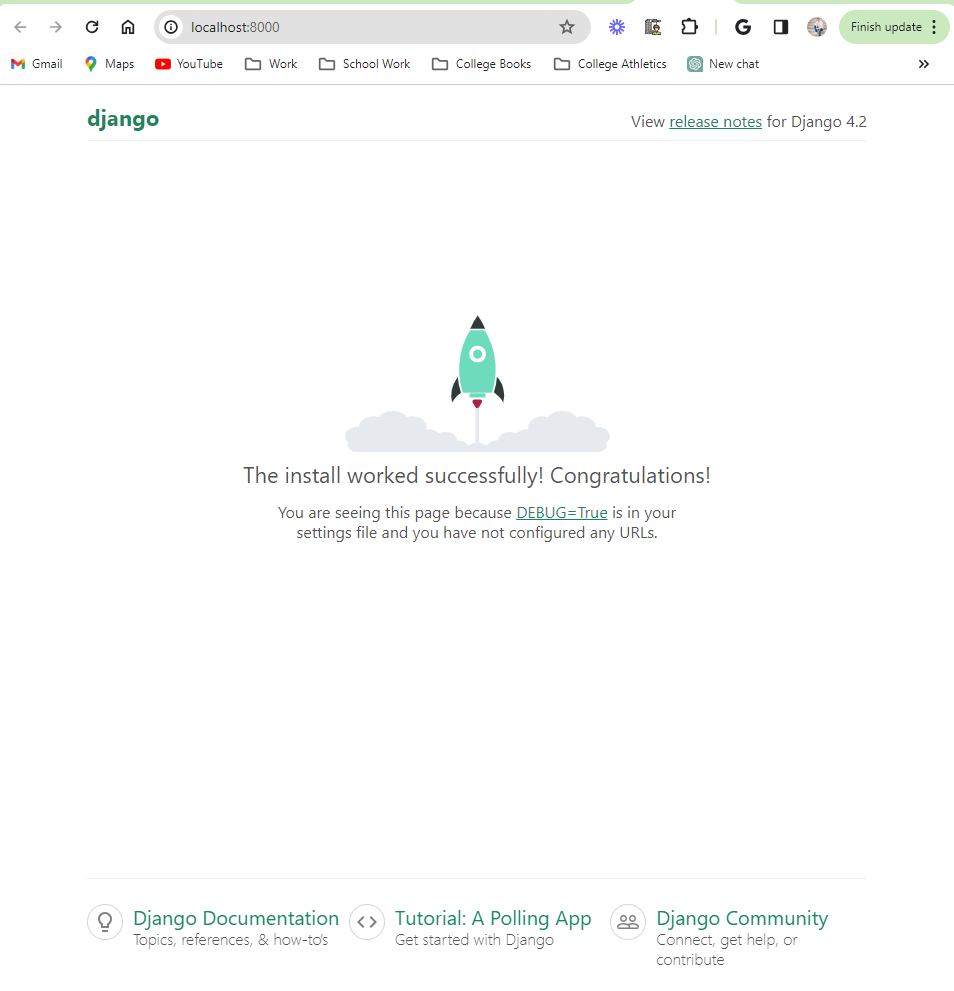
Jacob Larsen – Task 2.6

Task 2.1 Create Virtual Environment and Django Project (Jagger Zlotoff)

1. Open the command/terminal window and check your python version. You should be using 3.11. Use command python3 –version

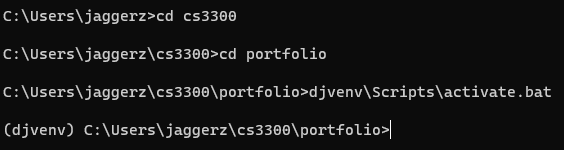


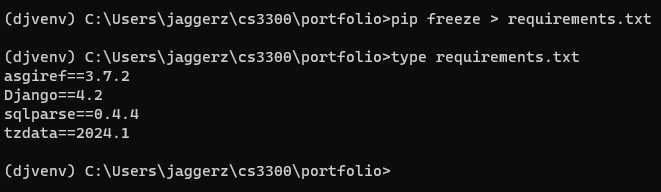
1. Create cs3300 folder then create portfolio folder for GE project
2. Make the folder for cs3300 by using the command mkdir cs3300
3. Go into the folder by using the command cd cs300
4. Make the folder for portfolio by using the command mkdir portfolio
5. Go into the folder by using the command cd portfolio
6. Create a virtual environment. Use command python3 -m venv djvenv
7. Activate the virtual environment (Note: when in windows make sure you're in command prompt). Use command djvenv\Scripts\activate.bat
8. Install django in the virtual environment (make sure it is version 4.2)( Make sure your virtual environment is active, indicated by ‘(djvenv)’ before proceeding to this step). Use command pip install django==4.2 
9. Upgrade pip. Use command python -m pip install –upgrade pip
10. Create a django project. Use the command django-admin startproject django\_project .
11. Reorder directory structure for ease of use. Because we added the ‘.’ at the end it creates the project in the current directory, which already simplifies the structure. 
12. Run server and ignore migration warnings. Use the command python manage.py runserver 
13. See if the installation worked by going to <http://localhost:8000/> and you should see. Celebrate!

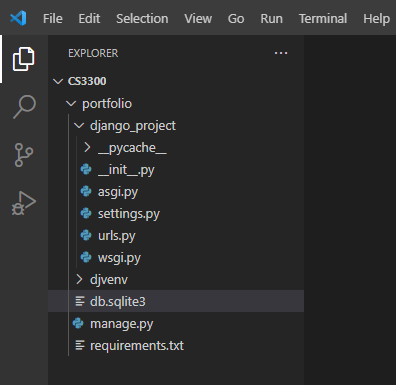


1. Open the second terminal and activate the virtual environment.Tip: I like to have 3 terminal windows tabbed: one for

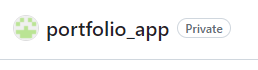
* venv to run commands
* venv starting server and stopping server
* git/gihub commands

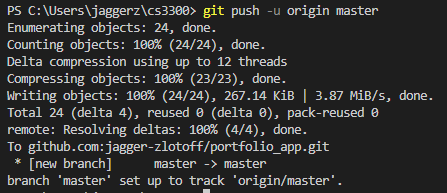
Navigate to your project directory where the ‘djvenv’ directory is located using the cd command then activate the virtual environment using the command djvenv\Scripts\activate.bat

1. From the virtual environment create a requirements file of what is installed. Use command pip freeze > requirements.txt and use the command type requirements.txt to list and check all the contents of the file. 
2. After all those steps are done your IDE if using VS code should look something like this.



Task 2.2 Create Local Git and Github Repository (Jagger Zlotoff)

1. Create a Git repository on the root directory in our case cs3300. Use command git init 
2. Create a .gitignore file. Use the command New-Item .gitignore -Type File
3. Then paste the content of the link into the .gitignore file. [Python.gitignore](https://github.com/github/gitignore/blob/main/Python.gitignore)
4. Add all the files to your Git staging area. Use the command git add . 
5. Commit the changes. Use the command git commit -m “Initial commit”
6. Create a private repository on GitHub. Log into your GitHub account, go to the plus sign in the top right corner, click on new repository, make the name for the repository, change it to private, and do not initialize with README, license, or .gitignore. 
7. Link the local repository to the remote. Use the command git remote add origin <REMOTE\_URL>
8. Push your local repository to the Github remote. Use the command git push -u origin master.



Task 2.3 Create Portfolio App (Elijah Casillas)

1. **Activate your virtual environment**
   1. This resource helped on Windows to [activate](https://stackoverflow.com/questions/46896093/how-to-activate-virtual-environment-from-windows-10-command-prompt) virtual environment
   2. Just change venv to dj\_venv (when in portfolio directory)
2. No extra steps
3. **Configure Settings: Edit your django\_project's settings.py file, add your app to the INSTALLED\_APPS list and add support for authenticating users:**
   1. Specifically add this A screen shot of a computer code

      Description automatically generated

Task 2.4 Define URL path and view (Elijah Casillas)

**Before you start please ensure that your gitignore is setup correctly!**

Solutions currently in place for this section include restarting, so ensure following of steps

* 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....
  + Just add the required code into the file using VS Code
* Update portfolio\_app/views.py views by defining the following view for the home page.
  + Same here
* Create urls.py file in portfolio\_app that contains a path to the defined view
  + For this step copy the urls from djangoproject into the portfolio app folder
  + Then edit the code
* Run Start server command-python manage.py runserver
  + This step may cause you issues. You may find you get an error where the module django\_project is not found with a slew of other errors.
  + First try to ensure these steps work correctlyA screenshot of a computer

    Description automatically generated
  + If the problem persists the solution i did was to delete everything and restart GE02 with the corrected commands djvenv\Scripts\activate for windows in portfolio folder
  + This is the only sure way that django installs correctly
* No extra steps needed for the rest of section 2.4

Task 2.5 Define HTML Templates (Lucas Haas)

1. Create a file with convention <name>.html

A screen shot of a computer

Description automatically generated

1. Edit file with notepad, VSCode or any other text editor
   1. Markup tags: [HTML Cheat Sheet - Learn web development | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Learn/HTML/Cheatsheet)
   2. HTML docs typically start with <!DOCTYPE html>
   3. Make sure to check open (i.e. <head>) and end (i.e. </head>) tags
   4. { % thing % } is replaced by item named thing from the django project settings.py (if it won’t work, make sure to check spelling on variable and if it exists to replace item)

A screenshot of a computer

Description automatically generated

Extras:

* HTML checkers can help find syntax errors for you (django syntax not always supported)

Task 2.6 Add static files (Jacob Larsen)

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 a new folder in the portfolio folder named “static” and create a folder inside of static named “images”

A screenshot of a computer

Description automatically generated

1. Download the UCCS logo from this link: <https://brand.uccs.edu/sites/g/files/kjihxj1416/files/inline-images/uccs-signature-email.gif>
2. Rename the logo to “uccs\_logo.png”
3. place it into the images folder



3) Update the settings.py file inside the django\_project folder

1. At the top (above “from pathlib import path) add “import os”

A black background with white text

Description automatically generated

1. Near the bottom, under “#static files (CSS, JS, Images), add the following code:

STATIC\_URL = ‘static/’

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, ‘static’)

]

MEDIA\_URL = ‘/images/’

4) At this point the process may differ from person to person depending on a few things, all paths are covered below:

Go to base\_template.html and look at the very top of the document, if there is a {% load static %} already above <!DOCTYPE html> then you can skip the following steps

If the {% load static %} is not there:

1. Add {% load static %} above <!DOCTYPE html>
2. Search for the <a class=”navbar-brand” href=”#”>Navbar</a> and replace it with <img src=”{%static ‘images/uccs\_logo.png’ %}”>

5) check the website to make sure all the code worked by heading to 127.0.0.1:8000 and seeing if the logo appears, if not make sure the image is named “uccs\_logo.png” in *every* instance it appears.

6) congrats! You’re done! Push it to the branch and merge once version is documented!