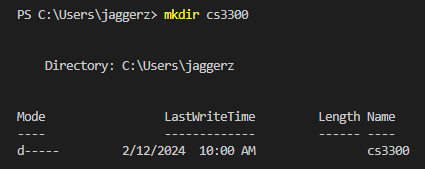
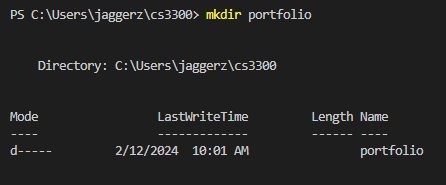
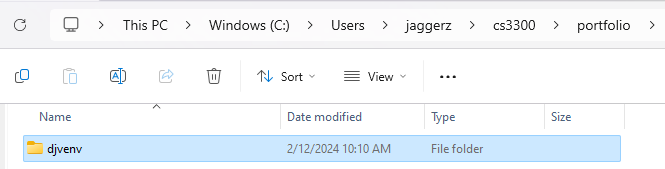
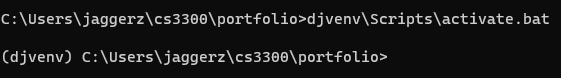
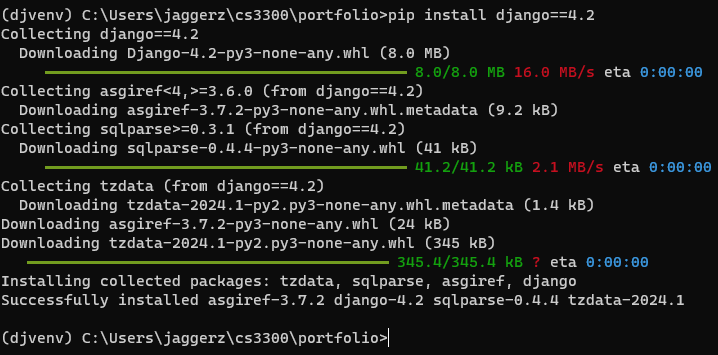
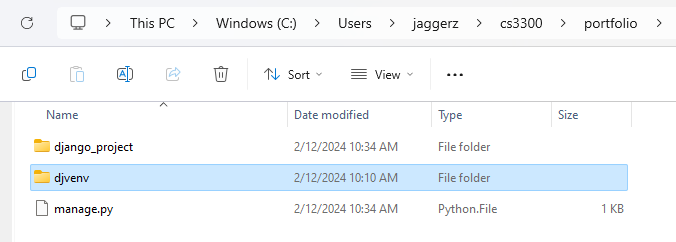
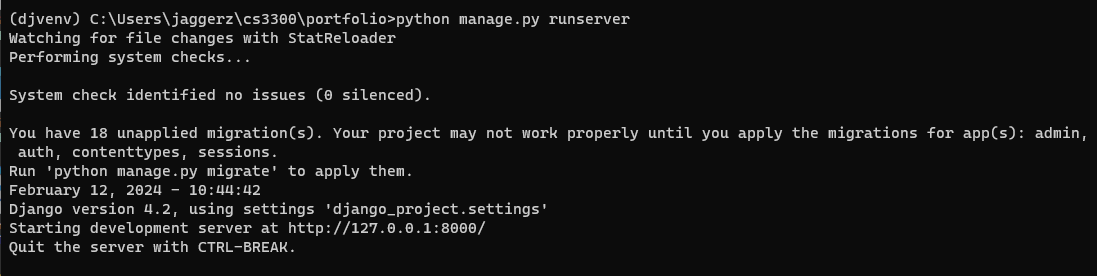
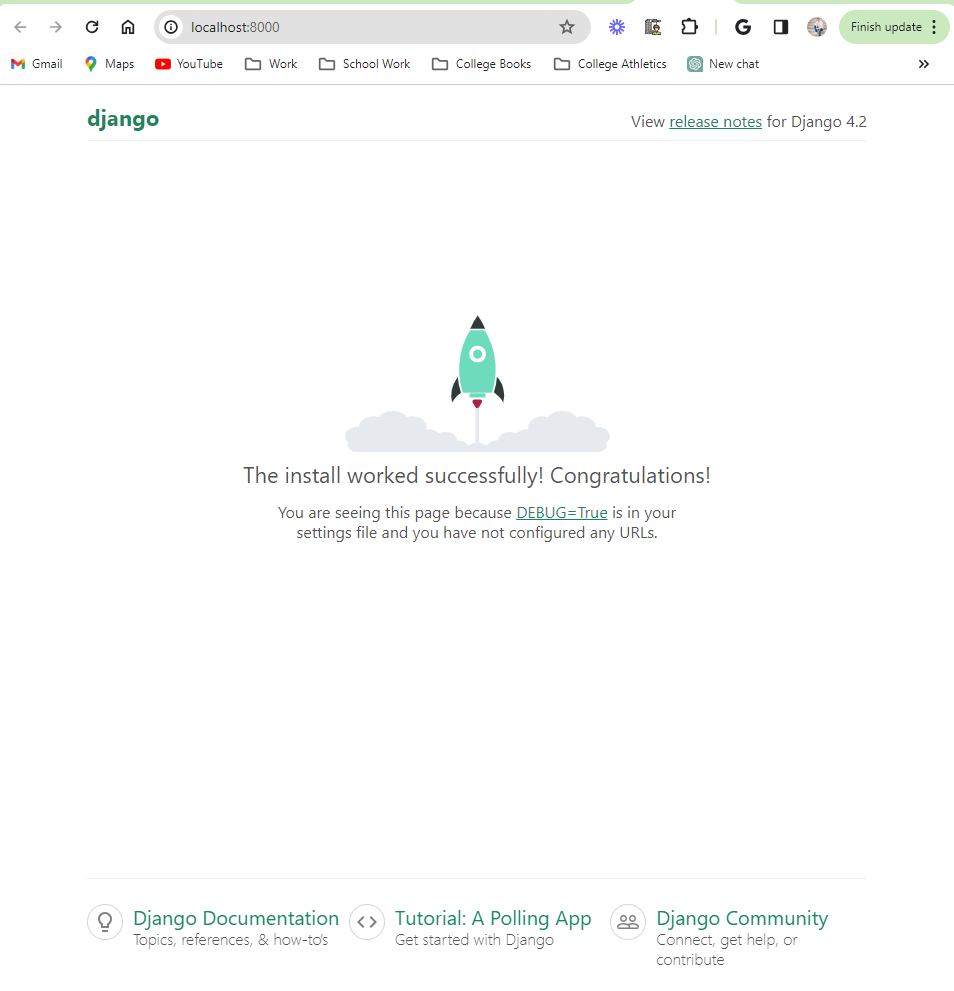
Task 2.1 Create Virtual Environment and Django Project

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

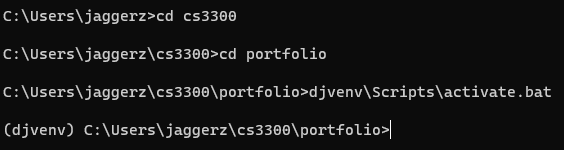


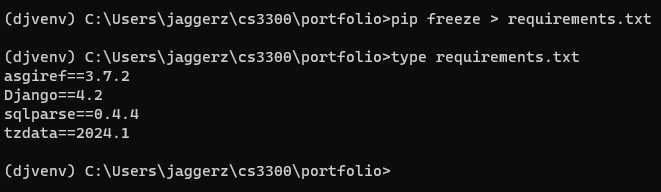
1. Create cs3300 folder then create portfolio folder for GE project
2. Make the folder for cs3300 by using the command mkdir cs3300
3. Go into the folder by using the command cd cs300
4. Make the folder for portfolio by using the command mkdir portfolio
5. Go into the folder by using the command cd portfolio
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!

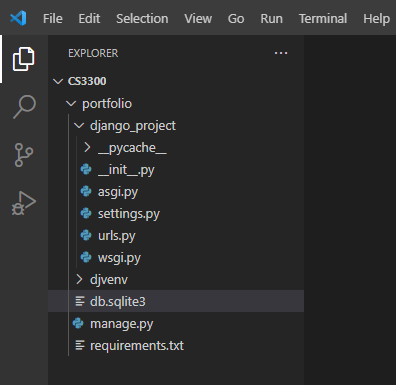


1. Open the second terminal and activate the virtual environment.Tip: I like to have 3 terminal windows tabbed: one for

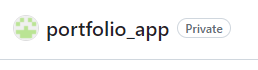
* venv to run commands
* venv starting server and stopping server
* git/gihub commands

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

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.

