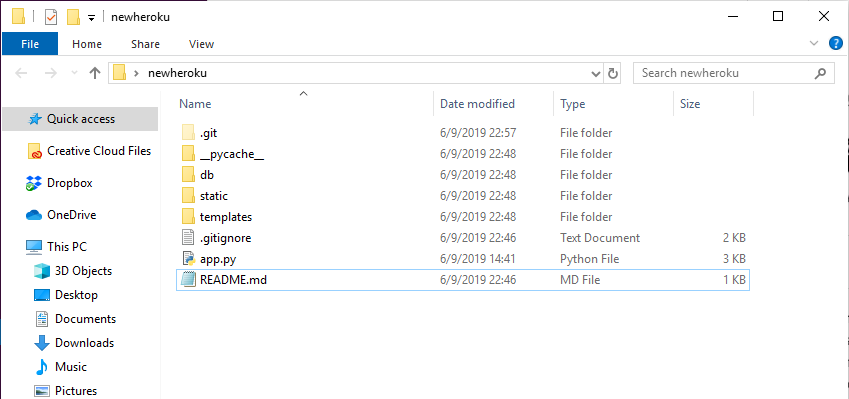
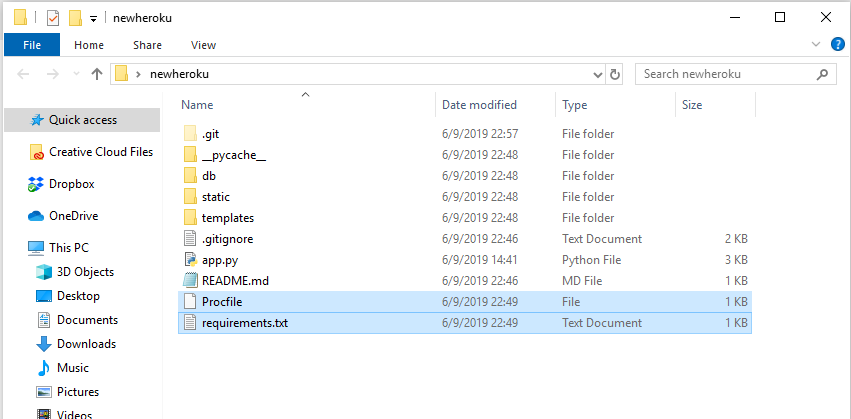
Hosting to Heroku

1. You will need a Heroku account. If you opened one during the “prework” section of this bootcamp, please use that account. If not, please go to [www.heroku.com](http://www.heroku.com) and click Sign Up for Free. Enter your first name, last name, and email address; for Role, select Student; for Country, select United States; for Primary development language, select Python. Check your email inbox for a confirmation message, and follow the instructions provided. Be sure to remember your username and password!
2. You will need to create a new conda environment. Please do so as follows: (Note that you do NOT have to type the dollar sign for any of the commands.)
   1. Open a GitBash or Terminal window. (Exactly where you open it doesn’t matter for this step.)
   2. $conda create -n ForHeroku python=3.6
   3. $source activate ForHeroku
   4. $conda install pip
   5. $pip install gunicorn
   6. $pip install pandas
   7. $pip install flask
   8. $pip install flask-sqlalchemy
   9. Close the GitBash or Terminal window.
3. Create a new repository on GitHub and name it anything you’d like. I’ve named mine newheroku. Do this the same way you’ve done it for every other Python assignment.
4. Use Git to clone the new repository to your laptop, similarly to the way you’ve done this on past assignments. This will create your working directory.
5. Populate the working directory with the starter files for the assignment. To do this, copy the starter files from the class repo, the same way you’ve done this on past assignments.
6. When you’ve finished populating your working directory, it should look like the following. (Note: I have my system configured to show hidden files, which is why you see .git. If your system is configured differently, you may not see this file. For more information on how to show hidden files, please use Google.)



1. Now your system is set up properly and you’re free to write and debug your code, just as you’ve done on previous assignments. While you do this, don’t worry about deploying your application to Heroku. Heroku will happen later.
2. During development, you will need to activate the ForHeroku environment you created earlier. Otherwise, your packages won’t be imported properly.
3. During development, please use localhost to host your application, just like we’ve done in class. To do this, execute either of the following commands. Note that these must be run from your working directory. These two commands are largely interchangeable, and the differences between them aren’t important now.
   1. $FLASK\_APP=app.py flask run
   2. $python app.py
4. Once your application is running on localhost, you can view it using Chrome. To do so, go to <http://127.0.0.1:5000> or <http://localhost:5000>. These are two different addresses for the exact same server, which runs internally in your laptop. This is how I’ve demonstrated the process in class, and this is how you may have done it on previous homework assignments.
5. If Chrome doesn’t seem to update properly when you make changes to your code, you may have to clear the cache. Use Google to look for help if you need it.
6. Each time you’d like to test some changes to your code, you’ll need to restart the Flask server. To stop it first, type CTRL-C. (Note that you may have to type CTRL-C several times.) To restart it, use the command above.
7. Once your application is ready to host on Heroku, you’ll need to add two additional files to your working directory. These additional files tell Heroku how to create the environment it needs to host your application properly. Create these files as follows:
   1. $pip freeze > requirements.txt
   2. $echo web: gunicorn app:app > Procfile
8. After these new files have been created, your working directory should look like this:



1. Use Git to add, commit, and push your updated files to the repository. Now you’re ready to deploy your application to Heroku.
2. Log in to Heroku.com, choose to create a new application, and give it a unique name.
3. Select GitHub as your Deployment method.
4. Enter the name of the GitHub repository that contains this new assignment (i.e., the one you created above), and connect to this repo.
5. Click Enable Automatic Deploys.
6. Click Deploy Branch. Once the deployment process is finished, click the button for View.
7. You should now see your application running in Heroku.