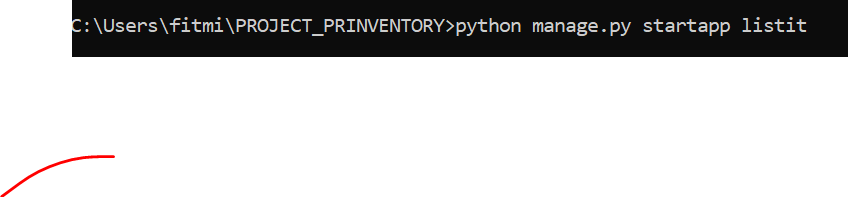
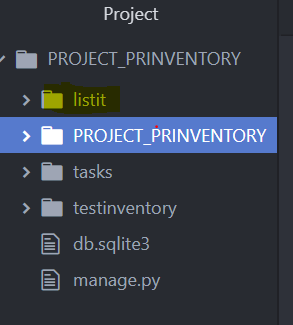
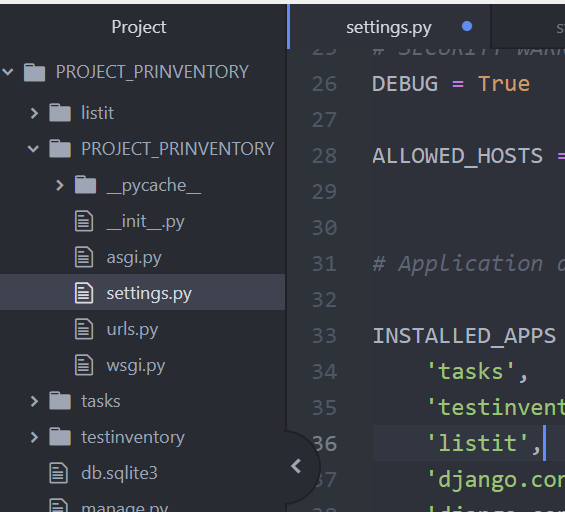
To create a django page we start at the cmd prompt in the directory of our project.

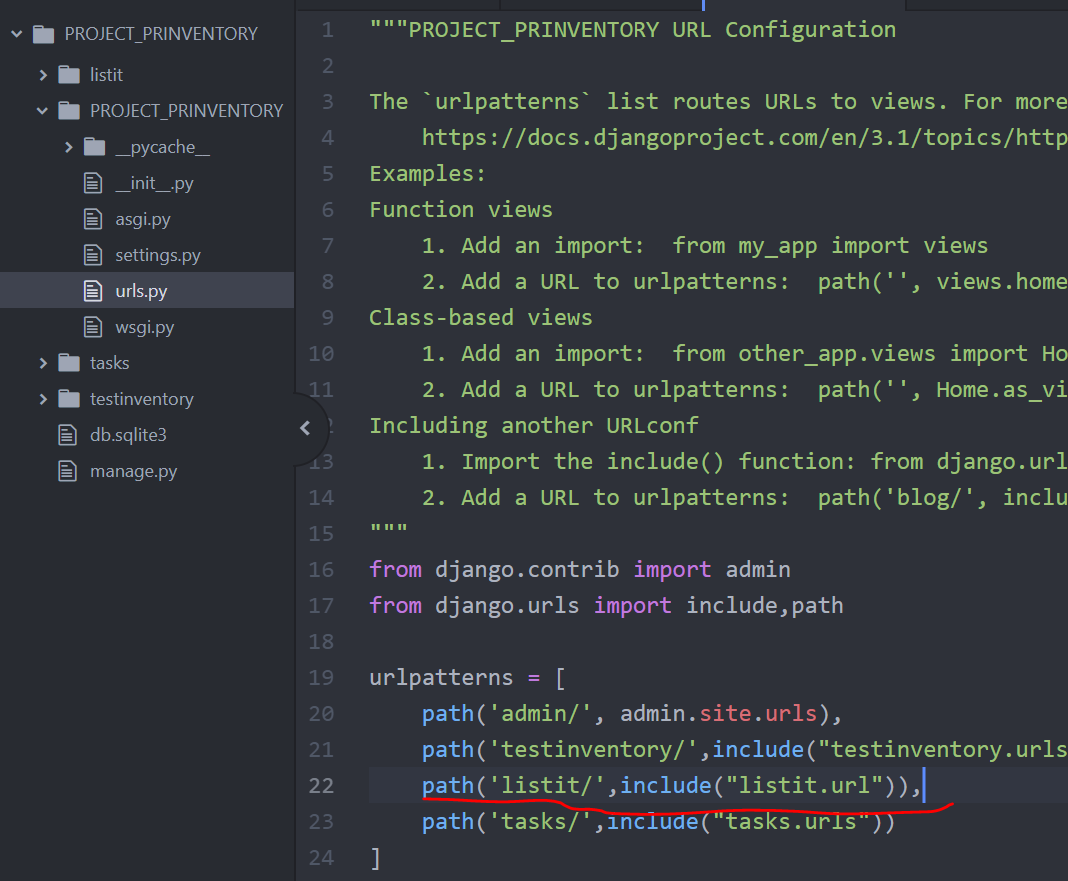
Your “app” (web page) will show itself in your IDE of choice (in this case, atom).

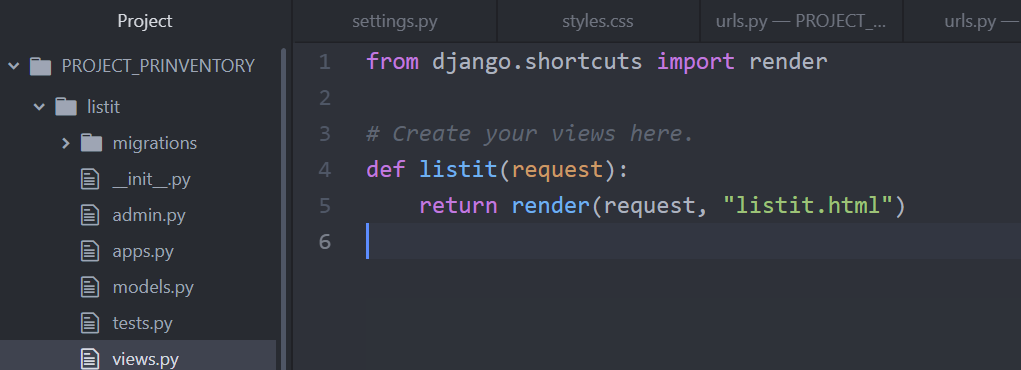


Then we need to go into the settings folder and add the app to “installed apps”.

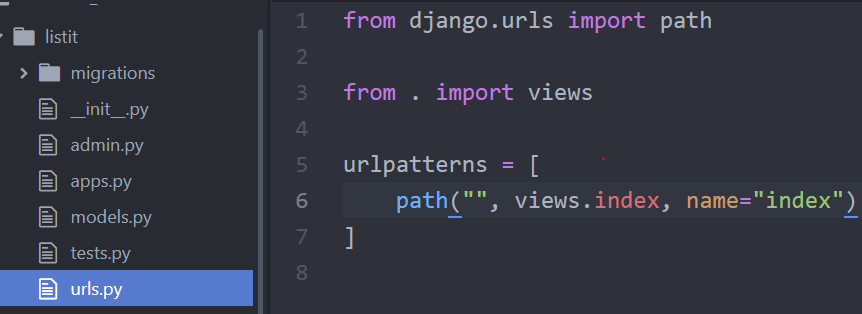
Don’t forget to save.

Then add it to urls.py and don’t forget to save. Note that the path mistakenly ended in .url instead of .urls. This took time to figure out.

Then add the view to listit/views to render the html page. The function should have been def index(request): instead of listit(request):

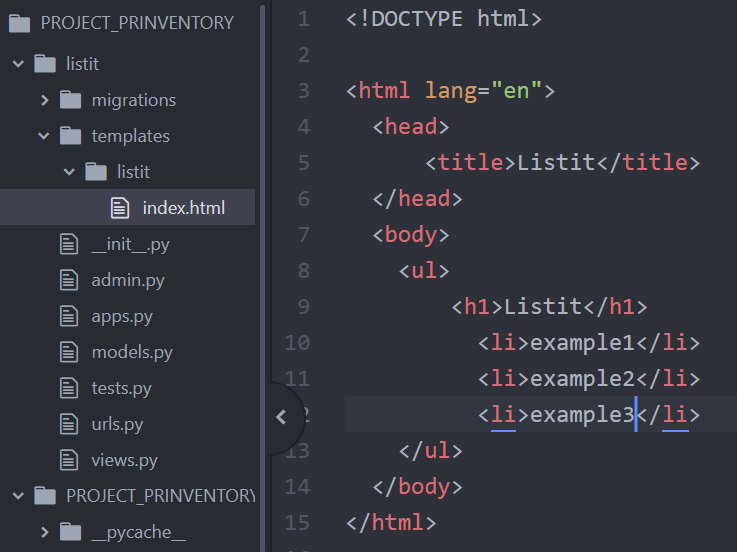


Then create the urls.py file under the listit folder to reference the index html page path.

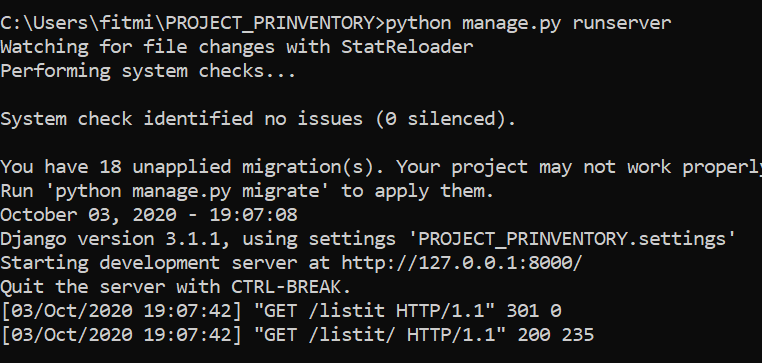


Then create a templates folder under listit. Then create a listit folder under that to create the index file.

Create your html and add as many indexes, urlpaths, views, etc as needed.



Go back to the command prompt to start the server and get the url for your browser to visit the page.



Note that the url must have /listit added to reach the proper site.



The problem I had the most with this relatively simple sequence was how easy it was to miss the minutia that would result in an error. Then I’d be on a hunt for quite a while to find the error.

This would become easier with time and repetition. I wonder if flask is easier.