

Exercise 2.8: Deploying a Django Application

Reflection Questions :

- *Explain how you can use CSS and JavaScript in your Django web application.*
 - **A:** Through the use of Django-supported static files, developers can make use of things like CSS and JavaScript. These can be used for their respective benefits, such as CSS for styling the design and layout of the rendered page, and JavaScript's many benefits which include numerous dynamic behavior features, advanced user interactivity, AJAX requests, and more.
- *In your own words, explain the steps you'd need to take to deploy your Django web application.*
 - **A:** I will list the basic steps, ignoring the monumental (and seemingly) endless issues that arise throughout these processes. The building of the app (the actual coding) is a minor task compared to enormous undertaking that is getting all the dependencies to be applicable for deployment with Heroku. Perhaps this is the case with any deployment, but I remember having countless problems that took weeks of troubleshooting to overcome the last time I used Heroku in a previous achievement.
 - Create a project repository
 - Create Heroku account
 - Configure static files
 - Configure database to be accessible by Heroku
 - Commit changes to Heroku using the Heroku-client
 - Open the deployed App

- *You've now finished Achievement 2 and, with it, the whole course! Take a moment to reflect on your learning:*
 - What went well during this Achievement?
 - The coding: The coding work, with respect to the actual building of the app through programmatic logic, and the development of those skill sets seemed to go well. It was not without its challenges and hangups. But, when those problems arose, it seemed like a lack of knowledge and skill on my part, which is to be expected when learning, and these scenarios were expected and even welcomed.
 - What's something you're proud of?
 - In light of what I answered in the previous question, I am proud of overcoming a ton of challenging problems. The curriculum would some times not have the exact answers in the code-programming aspects. I sometimes wonder if that was intentional to inspire student research, learning, and resilience to the challenges. I overcome those through varying solutions throughout the achievement.
 - What was the most challenging aspect of this Achievement?
 - Heroku: By far the most challenging part of the achievement was trying to work with Heroku. The deployment of apps and the management of dependencies has been the biggest struggle for me, in general. Heroku, however, stand atop that list. Out of all the time I spent actually coding, I spent at *least* that much time (likely much more, to be honest, when considering that each new attempted problem-solution comes with 15-20 minute wait times to see if the Heroku git-push will be successful) dealing with problems with dependency changes, repository problems, deployment tools, and various file/db management software. I am hoping I can talk to some student peers and/or professionals in the field to convince me that this aspect of my experience is not in simile to the experience of typical professional life of a developer.
 - Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?
 - I suppose I have avoided true candor up until this point; albeit unintentionally. That being said, I must say that I am truly disheartened. I don't really know if I have confidence or not. I don't know if I am doing well or not. I don't really know what I'm doing at all. I can't tell if the way that I research problems (finding others who have had similar problems, and they following their suggested solution verbatim) is what it is to be a developer. Sometimes I just try countless potential solutions until one of them works. Of course, I have some general logic to narrow down which

potential solutions would be tried first. But, the true comprehension of what is going on *within* those solutions is totally beyond my knowledge. In almost every circumstance, the people posting their solutions don't show any particular understanding of what they are doing, either, or why it worked. Even in the circumstances where they *do* explain the solution, they aren't really explaining it. I think they *think* they are explaining it; but they are actually just explaining how they arrived at the solution: explaining how a solution was arrived at, and the logic that makes the solution work, are two *very* different things. The actual intelligence, logic, and "skill" of it all seems to be an afterthought (or even an *avoided* thought) for most developers, from what I can tell. This whole process is almost as far from what I expected it to be when I decided to enter into it. So, in attempting to answer this question, I feel unequipped to answer it. If what I have come to realize is how the industry works (at least at the entry level), then I suppose I am becoming somewhat effective in the work. However, I don't know if this is a problem inherent to the coursework (problems with the curriculum and/or task designs), inherent to the industry, etc etc. Or, if the industry isn't in simile to what I am experiencing. In which case, I would be compelled to say that I have no idea what is going on, or what I may be expected to know skillful on or knowledgeable about; and therefore, have no confidence whatsoever. I'm really lost in a base-level kind of way that prevents me from even plotting an answer to this question.