

Part 3 Writeup

For my video screencast I show the functionality of the code in order of:

- Home Page
- New Product
- Product Details
- Update Product
- Delete Product

When creating this App for Sprint1 I learned a lot about software engineering using a framework. Django has a lot of built in assets and architecture that makes building the site quite easy compared to the full understanding of all the code that goes into the framework we use to make our app functional. Building the app was very similar to the portfolio app we built in class as it has a lot of the same assets, CSS/HTML, structure, and site architecture. This allowed for me to even more deeply understand the Django Frameworks structure of Model, View, Template, URL, Statics, and that interaction with the system as a whole all the way to the end User of the site.

Estimating:

My approach to estimating the time to complete different user stories was through looking at how long each of those tasks took me for the portfolio app and extrapolating how long it would take to implement again. I learned estimating is difficult and that I would rather overestimate than underestimate. I overestimated a bit too much though, teaching me that next time I can plan a little less time to complete the tasks.

Sprint Retrospective:

This sprint went well because of my knowledge of the Django MVT structure and previous implementation of the Portfolio app. The resources that helped most were that other app I built as well as the Django documentation itself. The official docs really are the best.

Problems:

I encountered an issue with importing a Picture. I made a static folder. I put an images folder inside it and I put that inside the ecommerce_app folder. This was an issue as in my html reference to it I made that folder /static/images, but I didn't know the command static implicitly uses that static folder. So I was trying to get a file from /static/static/images instead of /static/images. This issue persisted until found from **Inspecting the pages source** as It would render the page but not the image. Inspecting the source showed me the folder reference. This allowed me to fix the issue in my HTML template so that I was not going to the wrong folder path.

Lessons:

I learned a lot of things about:

Frameworks

Django itself

MVT architecture

HTML, CSS/Bootstrap

Project management techniques

Speed of production: My own timeline of task completion

Improvements I would make to process:

I would make a structure for writing the code:

Make everything in order of Models first then for each action within the site we add a URL, VIEW, then HTML Template. This would allow me to make the code from the order that I write it and naming conventions would be easy to maintain as the structure of writing the code lends itself well to linking those names and interconnections well.

I would make templates for the smaller parts of the code for reuse

Ex.

URLs, Models, Views, HTML templates.