

Journal Entry 1: 9/20

Today I started deciding what technologies to use for the application as well as deciding what platform I would make it for. I ultimately decided on using a Python backend as that is what we will be using for our project this semester. I also decided to make the application in a desktop website form since our project will involve making a browser extension for a desktop web browser. I also decided to brush up on my HTML and Python as I haven't used those in a while.

Journal Entry 2: 9/21

I heard from a teammate that React would be a good frontend framework to use. I have never used React before and I know I hate using HTML, so I figured this would be a good way for me to cut down on HTML usage while making a good looking website. I decided to start reading the ReactJS documentation deciding that would be a good place to start.

Journal Entry 3: 9/22

A teammate posted in our slack channel a tutorial for setting up a React app with a Flask backend, so I decided to follow that since it was exactly what I was trying to do. I followed the tutorial for a while then got pretty lost on the React stuff.

Tutorial Used: https://www.youtube.com/watch?v=_RSVoqXWzSw

Journal Entry 4: 9/23

I decided to put that tutorial on hold and went to the ReactJS website again to follow their tutorial since I figured they would know the best way to get started. Soon after I realized that this tutorial was making something very different than the application I was trying to make, so I went back to the previous tutorial on making a React app with a Flask backend. I worked on the tutorial some, but it assumed the audience already knew a great deal about Flask and was mainly showing how to connect React to Flask.

Journal Entry 5: 9/24

I decided that I should start working on making a Flask backend first and foremost, because I prioritized the functionality of the application over the frontend looks. I found a good tutorial on Youtube about making a fully featured web app in Flask. The blog he was making looked pretty close to what I wanted to do, so I started following the tutorial. I initialized a git repo and got to work setting up the rest of the environment.

Link: <https://www.youtube.com/watch?v=MwZwr5Tvyxo>

Journal Entry 6: 9/25

I saw in the tutorial video that he recommended using a virtual environment for Python projects because different projects may depend on different things. So I went and looked up some articles on how to set up a virtual environment and how they worked. Then I set up one of my own within my project folder.

Journal Entry 7: 9/26

I completed following the first tutorial video of the series. I think I'm going to try to get through one video each day until I'm roughly to a point where I have all the requirements that his blog and my application share. This first video got me to a point of having a basic "Hello World" page running on a locally hosted server. I have a route which points to a home page and another which points to an about page.

Journal Entry 8: 9/28

I went through a couple videos today. I was able to use Jinja2 templates and Bootstrap to make a pretty good looking website. I also was able to make a login page and a registration page. They don't do anything right now, but the template and forms are there

Journal Entry 9: 9/29

Today I got the database up and running using SQLAlchemy. I was able to connect the login and registration pages to it so a user account is made. I also have a few tabs and pages which can only be accessed by a logged in user. The user can also now edit their account information

Journal Entry 10: 10/2

I blasted through the rest of the videos until video 8. I think at this point, the website is as feature complete as I'm going to get it from this tutorial. The user can now submit a post which is linked to their user in the database. They can also update and delete this post. All the posts now show up on the home screen. Also, I restructured the program to be a python package.

Journal Entry 11: 10/3

I updated the Post database model to contain some relevant information to a house listing. It now contains things like a pet filter, the cost, the address, and links to images. I decided not to host the images myself, because I think over time, it would take up a lot of server space. I made it so the user can enter URLs of images hosted elsewhere instead. (This is what some big sites like reddit did for a long time) Each post also now has its own page. Pages that don't exist return 404 errors

Journal Entry 12: 10/4

I'm getting close to the deadline, so I'm going to try to implement one more feature. I had left to do the filtering, comment section, and favorites. I figured the comment section would be the easiest to implement, but I ran into quite a few issues along the way. I eventually figured out how to store the structure in the database with backreferences to both the user who posted it as well as the post it is part of. Only registered users can post comments

Summary:

Overall, I've learned a whole lot about web development. I started off knowing pretty much nothing but HTML and basic Python, but now I know how to set up a webserver using Flask, creating a virtual environment for Python projects, connecting to a database with SQLAlchemy, using routes to process the information in an address bar, how to use templates to make writing

HTML less repetitive and easier, and also how to set up forms with WTForms and style them using Bootstrap