Capstone

Designing and implementing a web application of your own with Python and JavaScript.

Overview

The final project is your opportunity to design and implement a dynamic website of your own. So long as your final project draws upon this course’s lessons, the nature of your website will be up to you, subject to some constraints as indicated below.

Requirements

In this project, you are asked to build a web application of your own. The nature of the application is up to you, subject to a few requirements:

- Your web application must be sufficiently distinct from the other projects in this course (and, in addition, may not be based on the old CS50W Pizza project), and more complex than those.
    - A project that appears to be a social network is *a priori* deemed by the staff to be indistinct from Project 4, and should not be submitted; it will be rejected.

    - A project that appears to be an e-commerce site is *strongly suspected* to be indistinct from Project 2, and your `README.md` file should be very clear as to why it’s not. Failing that, it should not be submitted; it will be rejected.
- Your web application must utilize Django (including at least one model) on the back-end and JavaScript on the front-end.
- Your web application must be mobile-responsive.

The most common cause for failure of the final project is not spending enough effort on this next instruction. **Read it completely**. Your `README.md` file should be minimally multiple paragraphs in length, and should provide a comprehensive documentation of what you did and, if applicable, why you did it. Ensure you allocate sufficient time and energy to writing a `README.md` that you are proud of and that documents your project thoroughly, and that distinguishes this project from others in the course and defends its complexity.

- In a `README.md` in your project’s main directory, include a writeup describing your project, and specifically your file **MUST** include all of the following:
    - **Under its own header within the `README` called `Distinctiveness and Complexity`**: Why you believe your project satisfies the distinctiveness and complexity requirements, mentioned above.
    - What’s contained in each file you created.
    - How to run your application.
    - Any other additional information the staff should know about your project.
- If you’ve added any Python packages that need to be installed in order to run your web application, be sure to add them to a `requirements.txt` file!
- Though there is not a hard requirement here, a `README.md` in the neighborhood of 500 words is likely a solid target, assuming the other requirements are also satisfied.

Failure to adhere to these requirements **WILL** result in a failing grade for the project, and you will need to wait and resubmit.

Beyond these requirements, the design, look, and feel of the website are up to you!

How to Submit

1. Visit this link, log in with your GitHub account, and click **Authorize cs50**. Then, check the box indicating that you’d like to grant course staff access to your submissions, and click **Join course**.
2. Install Git and, optionally, install `submit50`.

When you submit your project, the structure of your `web50/projects/2020/x/capstone` branch should match, in general, the file structure of Projects 1, 2, 3, and 4. Your branch should also not contain any code from any other projects, only this one. Failure to adhere to this file structure will likely result in your submission being rejected.

Your `README.md` file must also be at the highest level of your project. That is to say it should exist at `https://github.com/me50/USERNAME/blob/web50/projects/2020/x/capstone/README.md` (where `USERNAME` is your own GitHub username as provided in the form, below). If it doesn’t, reorganize your repository as needed to match this paradigm.

3. If you’ve installed `submit50`, execute

   ```
   submit50 web50/projects/2020/x/capstone
   ```

   Otherwise, using Git, push your work to `https://github.com/me50/USERNAME.git`, where `USERNAME` is your GitHub username, on a branch called `web50/projects/2020/x/capstone`.
4. Record a screencast not to exceed 5 minutes in length, in which you demonstrate your project’s functionality. Be certain that every requirement of the specification, above, is demonstrated in your video. There’s no need to show your code in this video, just your application in action; we’ll review your code on GitHub. Upload that video to YouTube (as unlisted or public, but not private) or somewhere else.

5. Submit this form.

You can then go to https://cs50.me/cs50w to view your current progress!