Practical Python and Data-Centric Development Milestone Project

Project Introduction

Project purpose:

In this project, you'll build a full-stack site that allows your users to manage a common dataset about a particular domain.

Value provided:

1. Users make use of the site to share their own data with the community, and benefit from having convenient access to the data provided by all other members.
2. The site owner advances their own goals by providing this functionality, potentially by being a regular user themselves. The site owner might also benefit from the collection of the dataset as a whole.

Project Requirements

Main Technologies

- HTML, CSS, JavaScript, Python+Flask, MongoDB
- Additional libraries and external APIs

Mandatory Requirements

A project violating any of these requirements will FAIL

1. Data handling: Build a MongoDB-backed Flask project for a web application that allows users to store and manipulate data records about a particular domain. If you are considering using a different database, please discuss that with your mentor first and inform Student Care.
2. Database structure: Put some effort into designing a database structure well-suited for your domain. Make sure to put some thought into the nesting relationships between records of different entities.
3. User functionality: Create functionality for users to create, locate, display, edit and delete records (CRUD functionality).
4. Use of technologies: Use HTML and custom CSS for the website's front-end.
5. Structure: Incorporate a main navigation menu and structured layout (you might want to use Materialize or Bootstrap to accomplish this).
6. Documentation: Write a README.md file for your project that explains what the project does and the value that it provides to its users.
7. Version control: Use Git & GitHub for version control.
8. Attribution: Maintain clear separation between code written by you and code from external sources (e.g. libraries or tutorials). Attribute any code from external sources to its source via comments above the code and (for larger dependencies) in the README.

9. Deployment: Deploy the final version of your code to a hosting platform such as Heroku.
10. Make sure to not include any passwords or secret keys in the project repository.

Important Notes

No authentication is expected for this project. The focus is on the data, rather than any business logic.

Project Ideas

You can either choose to build the website using your own idea, or take inspiration from the example ideas below.

Project Idea 0

Bring your own idea(s) to life, based on providing value to users to address a specific real or imagined need.

Project Example Idea 1

Create an online cookbook

External user’s goal:

- Find and share recipes

Site owner's goal:

- Promote a brand of cooking tools.

Potential features to include:

- Create a web application that allows users to store and easily access cooking recipes. Recipes would include fields such as ingredients, preparation steps, required tools, cuisine, etc.
- Create the backend code and frontend form(s) to allow users to add new recipes to the site, edit them and delete them.
- Create the backend and frontend functionality for users to locate recipes based on the recipe's fields. You may choose to create a full search functionality, or just a directory of recipes.
- Provide results in a manner that is visually appealing and user friendly.

Advanced potential feature (nice-to-have)

- Build upon the required tools field to promote your brand of kitchen tools (e.g. oven, pressure cooker, etc…).
- Create a dashboard to provide some statistics about all the recipes.

Project Example Idea 2

Create a jargon glossary/dictionary for a particular domain.

External user’s goal:

- Find and share definitions.

Site owner's goal:

- Collect good definitions to eventually publish the dictionary in book form.

Potential features to include:

- Create a web application that allows users to store and easily access definitions for terms in a particular domain, such as web development. You may want to use Wiktionary or Urban Dictionary (potentially NSFW content) for ideas.
- Create the backend code and frontend form(s) to allow users to add new definitions to the site, edit them and delete them.
- Create the backend and frontend functionality for users to search for definitions. You may choose to also provide an alphabetical display of all definitions.

Advanced potential feature (nice-to-have)

- Provide an up-voting functionality for users to express their satisfaction with particular definitions.

Project Example Idea 3

Build a book review and recommendation site.

External user’s goal:

- Find books they would like to read.

Site owner's goal:

- Earn money on each book purchased via a link from the site.

Potential features to include:

- Create a web application that allows users to upload details of books, including book name, author name, link to cover image and any other relevant fields. Allow users to write comments about any book and up-vote it.
- Create the backend code and frontend form(s) to allow users to add new books and reviews to the site, edit them and delete them.

Advanced potential feature (nice-to-have)

- Add a link such as the following to each book page, such that you could conceivably earn money from people looking to buy the book: https://www.amazon.com/s?tag=faketag&k=alice+in+wonderland Note that we do not actually encourage you to create an affiliate link, but rather want to demonstrate how this could work. Instead, for this project, we encourage you to just keep the tag value as something fake. Also, note that in general it's better to link directly to the book's page in the store, but that's a bit more difficult.