

CSCB20 - Assignment 3

Build Something Cool

Context

You've just spent the past term learning how to build a CRUD web-app using HTML/CSS/JavaScript/HTTP/. Forget Okayreads, forget databases defined by your instructors, now it's just time to play and use what you've learned.

Instructions

In a team of up to 3 people. Build something cool!

Is... is that it...?

Yes. That's all the instructions. Build something cool using everything we've learned this term. Don't make what we want you to make, make what you want to make. (Good thing you've spent most of your tutorial time working towards exactly this, huh?)

Are there any requirements?

Your submission must do at least the following:

- Use HTML + CSS to create consistent, well designed pages that respond sensibly to user input
- Use JavaScript to make the pages dynamic and responsive
- Use Flask + HTTP to host your pages and communicate with a web browser using both GET and POST messages
- Use Flask + SQL to store and retrieve data

Evaluation

The project will be marked out of 40 total marks as follows:

Fundamentals (15 marks)

Front End:

- HTML templates properly formatted
- HTML Forms generating HTTP POST messages
- CSS used properly to maintain a consistent and maintainable style
- Well documented JavaScript files used to provide dynamic components

Back End:

- Well documented Flask file handling GET and POST messages
- SQL tables well defined with sensible integrity constraints
- Flask used to generate SQL CREATE, INSERT and SELECT statements

Design (15 marks)

Front End:

- HTML templates using Flask to avoid redundant code
- CSS and JavaScript well integrated for maintainability
- Good use of fundamental design: Grid or Flexbox systems used to ensure code works and looks good on screens of various aspect ratios and sizes

Back End:

- SQL databases well designed for maintainability and eventual growth
- Advanced and complex SQL queries going beyond minimal CRUD elements
- Use of sessions and data management to ensure persistent user login across pages

Extension (10 marks)

These items are more flexible (and will probably require a higher effort/reward ratio), but could include things such as:

- Using features beyond basics covered in lectures/tutorials
- Going above and beyond in styling site or polishing user interface
- Building a real “passion project” that does something beyond the technical limitations and solves a real world problem or makes the world better in some way

Submission

You will need to submit your `.py` Flask file, as well as your `templates` and `static` directories and associated files as a `.zip` or `.rar` file (Note: when zipping your files, do not include your `venv` directory. If you have trouble with this ask for help from your TAs or on Piazza).

You should also include a file called `instructions.txt` that has the names and student numbers of all group members (please list the names as they would appear in Quercus so we can find you to attribute the marks), along with a description of your site, and any instructions or features that would not be obvious to the graders (e.g., if you need to do something special to create an admin account, or have a specific feature only open to certain types of users)