

University of Arizona, CSc 337

Final Project

Thanks to Victoria Kirst of Google for parts of this document

Deadlines

Proposal: 11:30pm 4/5

Code Submission: 11:30pm 4/24

Presentation: 4/25 and 4/30 in class

For the final project, you will be creating a web app using the key technologies we have practiced throughout the semester. Instead of giving you a specific web app to replicate, you are given the freedom to choose what you'd like to make.

We are giving you some restrictions on what you can make, such as a list of technical and style requirements.

You may work alone but you are encouraged to work with another student in a group.

Unlike the homework assignments, the final project **cannot be turned in late.**

Technology requirements

In your final project, you must include the following technologies:

Frontend: HTML/CSS

- Use classes and ids
- Use floating
- Change 2+ default font properties
- Change 4+ box model properties (border, padding, margin)
- Must pass validators

Frontend: JavaScript

- Listen for 2+ events
 - Can be the same event type, like 2 click events
- Use `fetch()` to talk to your backend
- Validate user input

Backend

- Use Node and ExpressJS
- Save persistent data in a database or file
- Include at least 1 GET or POST route

Misc

- You need to incorporate all these technologies into one cohesive web app.
 - We will not be grading on things like how interesting your project idea is, how original your idea is, how "realistic" it is, etc. However, your project also can't be a set of totally nonsensical code that happens to hit our checklist of technologies. Your project should be a web app, even if it's a very small web app, or a silly web app, or a dumb web app.
- Your project should be relatively bug-free.
 - It's hard for us to judge whether you are using the technology correctly if your code is very buggy.
 - We will not deduct points for trivial bugs or polish bugs, but we will deduct points for things like if the one GET in your project is not actually working.
 - This is another reason why you should aim small in your project scope.
- Your project should be deployed to Heroku

Style requirements

These style requirements should be no surprise, as we've enforced them all semester.

HTML/CSS:

- Use tags semantically, e.g. don't use <div> for every single element on your page.
- Use descendent selectors to reduce redundancy in CSS and HTML
- You must write raw HTML and CSS, as we have done all semester.
 - i.e. don't use SASS or compiled CSS/HTML

JavaScript:

- Avoid global variables. Use the module pattern.
- OK to use globals for constants, or other reasonable scenarios
- Add/remove/toggle classes from classList instead of modifying style directly, unless you have to modify style directly (e.g. if you are calculating the value of a CSS property in JavaScript)
- You must write raw JavaScript, as we have done all semester.
 - You may not use jQuery or Prototype.
 - You may not use any frontend frameworks, such as React, Angular ect.

Backend:

- Your backend must be written using the Node and Express libraries
- It is better to store data in a database than in text files
- HTTP methods should be used in ways that are compatible with the method definition. For example:
 - Use GET for retrieving data. Do not write data in a GET handler.
 - Use POST for saving data. Do not use POST to display a page.
 - Don't use query parameters with POST

Presentation

You will be required to do a 5 minute demonstration of your page in class on 4/25 or 4/30. This presentation will be informal. All you will need to do is demonstrate how your web app works and explain what it does.

Proposal

You must turn in a project proposal by 11:30pm on 4/5. This proposal must include the following information:

- Your name and your partner's name (if you are working in a pair)
- A paragraph describing what your app will do
- A paragraph describing what requests you will send to the server and what data will be stored on the server.