Austin Jameson

February 6, 2024

CPSC 4911 Sprint 0 Notes

Please take notes on the technologies that your team has decided to research. There are some examples provided next to each prompt, but these lists are not exhaustive.

\*\*Note: Due to the nature of some web-development frameworks (i.e., SvelteKit, Django, Next, …) your frontend and backend may be hosted by the same technology. In this case, please take notes on the specific parts of the framework in the sections below.

**Frontend Language(s) (Javascript / Typescript / JSX / Svelte)**

\*\*Note: Backend languages such as PHP and Python will template/render some variation of JavaScript and HTML to display in the browser. If you are using a backend framework, recognize what language it will be rendering.

URL(s):

Notes: Probably going to use JavaScript for this since most team members have a decent handle on it. Will most likely be used for dynamically creating HTML elements, event listeners for buttons and such, interacting with the API, etc.

**Frontend Framework/Architecture (Vue / React / Svelte[Kit] / Raw)**

\*\*Note: If using a backend framework with a raw frontend language please take notes on how the backend processes and renders front end code into a completed file (e.g., PHP will echo HTML from the server into the returned file)

URL(s): <https://www.w3schools.com/react/react_getstarted.asp>

https://www.techmagic.co/blog/why-we-use-react-js-in-the-development/

Notes: Using React. Seems like for testing and experimenting purposes you can just code it directly into the HTML which is convenient. Must use npm to create a react environment in actual application. Article says it’s relatively easy to use and not super complicated.

**Back End Language (PHP, Javascript, Python)**

URL(s): https://www.turing.com/kb/javascript-for-backend-development

Notes: I’ve already used both PHP and JavaScript a decent amount, especially during CPSC 3750 (Web App Dev). I have a decent handle on using JS. It will probably be the simplest option for interacting with the database.

**Back End Framework/Architecture (Laravel, Django, Flask, Node.js)**

\*\*Note: Though it is not usually recommended and is often harder, you may choose to use the raw version of some of these languages. As a general recommendation you should look to use a trusted framework for efficiency and security reasons.

URL(s): https://www.freecodecamp.org/news/how-to-create-a-react-app-with-a-node-backend-the-complete-guide/

Notes: When researching React, it seems that it pairs best with Node.js so that is probably what we will go with. The link is a good guide for setting up the React environment with Node.js. Apparently, Node.js is one of the simpler backend frameworks so it’ll be good to learn.

**Relational Databases (PostgreSQL, MySQL)**

\*\*Note: Teams may find views and triggers to be beneficial as well as standardize operational queries. Please emphasize learning joins, grouping, ordering, and filtering. Finally, take notes on technologies/libraries in your backend language used to connect to the server.

URL(s): <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/option1-task1-create-rds-database.html>

https://docs.aws.amazon.com/opsworks/latest/userguide/customizing-rds-connect-create.html

Notes: We are going to use MySQL and RDS for the database. I already set up a database server in the learner lab and started experimenting with adding fields and columns and stuff.

**\*\*DEPENDENT ON SERVERLESS DECISIONS\*\***

**Web Servers (Apache, Nginx)**

\*\*Note: Emphasize research on permissions, services, virtual hosts, proxy passing and certifications.

URL(s): <https://www.hostinger.com/tutorials/what-is-apache>

<https://httpd.apache.org/docs/2.4/mod/mod_proxy.html>

<https://httpd.apache.org/docs/2.4/vhosts/>

Notes: