Stack:

Backend

***SQL* ->** Database, describes table columns, raw relational data
***Django* ->** Python Model View Controller framework, describes database schema, structure, outputs JSON to frontend via a set of routes/endpoints

Frontend

***Typescript* ->** Javascript superset with static typing, language we will be using to build out frontend
***React* ->** Receives JSON from backend through a set of routes, describes blocks of html code on the webpage.
***CSS* ->** Describes styling of given html blocks based on class names, we will be using Emotion Styled components for simplicity

Dev Tools

Before learning about any framework in particular, I highly recommend familiarizing yourself with the browser dev tools. Here are a great set of tutorials for Chrome dev tools specifically. Learning about the dev tools allows you to mess around with any component or webpage and view/understand its structure.

Most important *Elements/CSS* - Describes literal page layout with html and css

https://developers.google.com/web/tools/chrome-devtools/css

Less important *Console* - Less useful but still good to understand, browser console similar to a terminal

https://developers.google.com/web/tools/chrome-devtools/console

Less important *Network* - Describes network request/response packets, goes over a lot of the concepts Clint talked about on Monday

https://developers.google.com/web/tools/chrome-devtools/network

React

Most important *React Starter Tutorial*- tic tac toe game https://reactjs.org/tutorial/tutorial.html

Less important *Function components/Hooks*

https://reactjs.org/docs/hooks-state.html

CSS

Less important *CSS intro tutorials* - helpful to understand, not as crucial as React

https://www.w3schools.com/css/

Less important *Emotion Styled Components* - Library we can use with React, more simple than using default CSS

https://emotion.sh/docs/styled