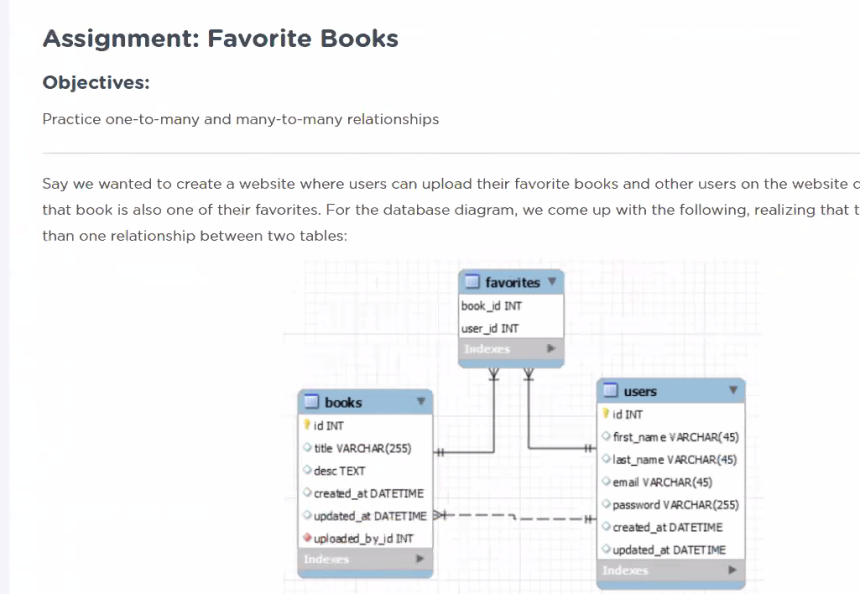
Render is not a redirect

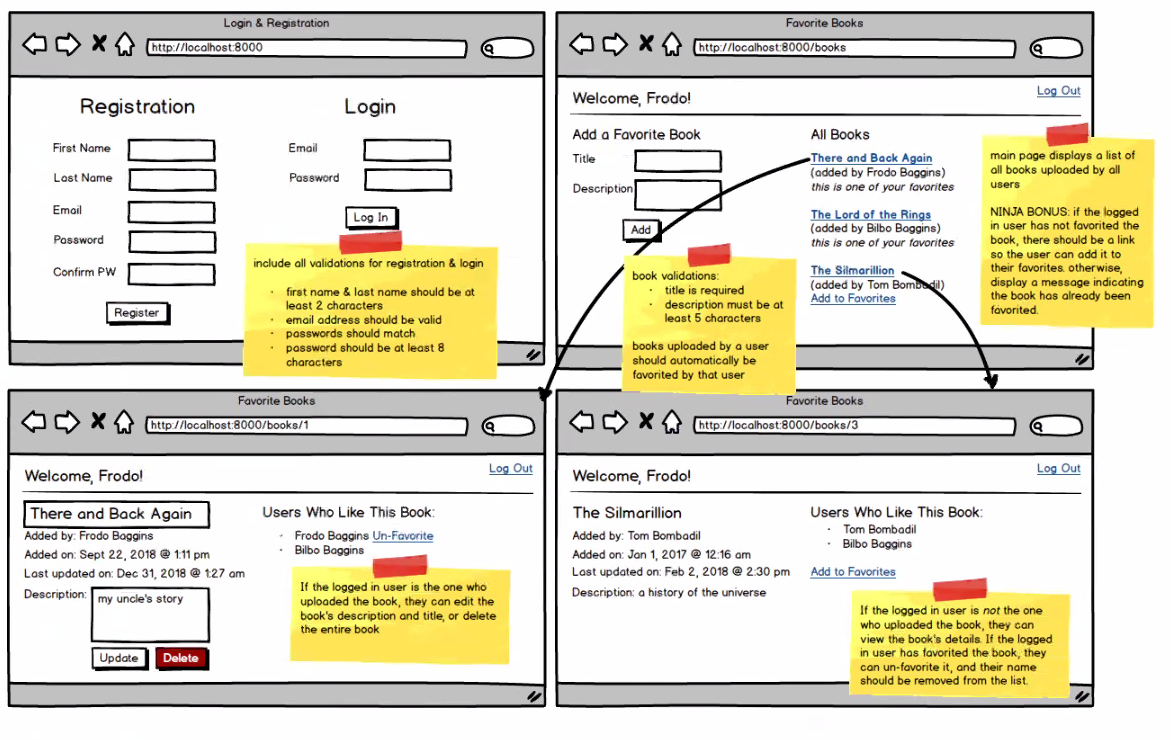
If the route is doing a lot of stuff, then redirect

Pages that we see index.html in the browser address bar are not using a framework

This page in her example is the root, or probably the index.html







As in above, assignments have the route showing in the top

In drawing above, the bottom two are really the same html, same route. These two are generally called “show”, so maybe show1….because they are showing something…and a render template….

The upper right is called a showAll…

Users assignment is a good template for catching route structures

Login/registration

* Route that sends to html

How do we know that the root route is a render…the first route is generally a do little render, but it might populate with something from the database….retrieving more than crunching

If there is a lot of action, then likely to be a redirect

The success route is also a render because it does little, just catching, and showing a page

Logout does not do much, but the root route already does the page I need.

There seems to be an idea of not rendering to the same page in two routes

In the reg.login, I had had two html pages..and two renders…We won’t have two renders for the same html page. Because every html page needs at least one render, that means # routes that render = # html pages

Wes on when to use redirect:

Rule: If sending a form to a route, none of the returns will send directly to template

A route that does have a return render\_template likely does come in straight from a user, but might require a cookie to be set

Claire: this is about the double post

Claire: various pages might check for cookies before sharing data

Redirect if user seems to be up to no good, or lost

Get versus Post:

* Always Get, except when user is submitting form

another little guideline I made up, each form seems to require a route because the form data needs to be caught by something an acted upon

Sunday 4/14/19 p.m.:

here's a fairly simple website. try poking around and ask your self... does this page get data from a server (i.e. is there a list of data of some sort)? is there a form anywhere on this page? if so, I've got a POST route; if not, chances are I'm just rendering an html template. feel free to conceptualize the routes (you don't actually have to build it out) and I'll be happy to give you my opinions

<http://www.shopcottonexchange.com/>

4/17/19 (me is my summarizing what Wes just said)

From Me to Everyone: 05:19 PM

Whenever you see a link or a button…that will be a route

From mifx to Everyone: 05:22 PM

makes sense

From Me to Everyone: 05:22 PM

too many routes....happens when we use different routes that have the same code/structure...only difference is a parameter

From Dietrich Green to Everyone: 05:22 PM

-----------=======

4/17/19

**instructor\_wes**

[6:05 PM](https://dojo.news/coding-dojo/pl/qo47uppj77dduk11iepjufjb3y)

To clarify here, the type of request is really dependent on what the client is trying to do. If you want to get information from the database, use a GET. If you want to create something new in the database use a POST.