Project 2
Rails site with data

Due: 1 April 2024

For the second project you'll build another small website in Ruby on Rails, this time involving data (and some of the other things you've been figuring out on your own).

The topic is close to home as we ramp up towards registration season: you'll pick a department or program at Longwood (other than computer science or math), and construct a multi-page website that presents information about that program, including its faculty and some of the courses it offers.

The website you build should have the following properties:

- At least three pages, including a homepage or landing page (associated with the root route `'/'`), with appropriate meaningful metadata and content, cleanly styled, with HTML as purely structural as possible, with the visual styling reserved for CSS/SASS.
- The styling should make use of flexbox layout and media queries to look differently good on at least two device types or aspect ratios.
- Pages should have logo, navigation bar (using `link_to` and shared layouts with `provide` and `yield`).
- The project should include substantial use of the automatic testing facilities in Rails to effectively test that individual pages and the relationships between them are working.
- One of the pages (not the root page) should have a listing of courses offered in the program, in tabular format, including at least the course number and title (possibly more depending on your design choices).
  - Each element in the table on the listing page should be a link to an individual page about that course (see below).
  - The table should use partials to cleanly organise the code for displaying each course.
- One of the pages (not the root page) should have a listing of the the faculty in the department, including at least their name (possibly more depending on your design choices).

  - Each element of the listing should be a link to an individual page about that professor (see below).
  - The listing should use partials to cleanly organise the code for displaying each professor.

- The data should be editable in a standard CRUD/REST/Rails way. The pages handling creation and update should make use of partials to tightly connect their content and layout (and reduce code duplication).

- The pages and links that modify the database should require at least limited authentication in order to view and execute them. This project does not require a full-blown system to create new users, set passwords, etc but should have some way to authenticate and remember that a user has create/edit permission. (You need to tell me in a readme or somewhere how to access this functionality!)

- Data should be at least minimally validated before saving it to the database, and in case of invalid input, the user should be provided with a reasonable experience (notify of the problem, give a chance to edit appropriately).

- A group of the pages will be pages for each individual course. Each such page, viewing that course's resource, should display at a minimum the course number, title, a description of the course, and when it was last taught and by whom, and when it is next scheduled to be taught.

- Another group of the pages will be pages for each individual professor. Each such page, viewing that professor's resource, should display at a minimum their name, photo,[1] office number, and a brief description.

- Either the course pages should actively link to (not just print the name of) the professor who most recently taught the course, or the professor pages should contain a listing of courses they've recently taught (with links), or both (design choice!).

- Regardless of other design choices or the current contents of the database, your website should cleanly handle cases where a professor has not yet taught courses, a course has not yet been taught, etc, without erroring or crashing.

[1] Subject to change but stay tuned

To the extent that you base your design ideas on other sites you've seen, *cite your sources* by indicating your inspirations in comments in the HTML. Similarly, any content lifted for this project should be cited either in comments or actually visibly in the page footer.

Handing in

Hand this project in using the `handin` script for course `cmsc210` and assignment `proj2`. If you are doing your development on some machine other than the lab machines (totally ok, and not difficult—just install ruby, and then run the various `gem install` commands and so forth), upload your stuff to the server *and verify that it works in the lab* before handing in. I'll test your code by running `bundle install` if necessary and then running `rails test` and `rails server` (and also reading parts of your code to check things that wouldn't be visible from those two commands). So you should make sure it works there.

It's due at 8pm on 1 April.