**Useful Sites**

Ruby on rails guide - <http://edgeguides.rubyonrails.org/index.html>

For validations used on the models - <https://apidock.com/rails/v2.3.8/ActiveRecord/Validations/ClassMethods/validates_length_of>

For more validations - <http://guides.rubyonrails.org/active_record_validations.html>

Basics on queries in the rails console - <http://guides.rubyonrails.org/active_record_querying.html#ordering>

Faker Library - <https://github.com/stympy/faker#fakergameofthrones>

rspec documentation - <http://www.rubydoc.info/gems/rspec-expectations/file/README.md>

Self-referential associations (model stuff, basically joining a table to itself) - <http://railscasts.com/episodes/163-self-referential-association?view=asciicast>

creating polymorphic relationships in the models (db stuff) - <http://culttt.com/2016/01/13/creating-polymorphic-relationships-in-ruby-on-rails/>

**Useful terminal commands (see starting new app for specifics on that)**

rails s - starts the server for a project at <http://0.0.0.0:3000/>

can shut it down with ctrl+c

rails c - opens rails interactive console

Hirb.enable - type this in rails console after installing the gem to make it easier to look at data from the db

**RAILS PROJECTS TO LOOK AS FOR REMINDERS**

Open great\_number\_game in Rails for the very basics, including

* example of session use
* routes page
* flash messages
* getting and using the paramaters passed in from url
* redirects in rails
* using flash messages in the view and basic forms
* if statement in the view (html)

Next, open survey\_form in Rails for

* very basics of forms in rails
* getting the data from forms and using it in the controller
* passing the data from the form as a single chunk of data instead of individual pieces or fields (as an associative array)
* accessing the session data in the view, in the format of an associative array

Next, open integrating\_models\_controllers in Rails for

* Basic look at models (db stuff)
* accessing the db in the controller
* example of plugging the db data into a variable in the controller than can be used in the view
* example of the faker gem for fast create

Next open management\_system the in Rails folder for examples on

* partials
* full RESTful routes and controller operations
* form\_for
* very basic database validations (models folder in the user.rb page)
* if statements as validations for creating/updating data in the db
* and private params

Next open products\_dashboard in rails for

* a look at more complicated models and db
* basic relationships between different models (belongs\_to, has\_many, etc.)
* querying the db with tables that are connected and passing them to the view
* understanding and examples of how querying the db gets you an array of objects
* further use of strong params
* example of the each iterator being used on queried data
* examples of RESTful anchor tags, and how to specify the method (such as delete for the destroy route)
* using forms instead of form\_for
* using hidden inputs in the form to determine the method (post, patch, etc.)

**OTHER USEFUL INFO OR EXAMPLES**

Example of what the ‘resources’ gives you when used in the routes page

Ex from routes page:

Rails.application.routes.draw do

resources :users

Ex:

| **Helper** | **HTTP Verb** | **Path** | **Controller#Action** |
| --- | --- | --- | --- |
| [**Path**](http://0.0.0.0:3000/userss)**/**[**Url**](http://0.0.0.0:3000/userss) |  |  |  |
| users\_path | GET | /users(.:format) | users#index |
|  | POST | /users(.:format) | users#create |
| new\_user\_path | GET | /users/new(.:format) | users#new |
| edit\_user\_path | GET | /users/:id/edit(.:format) | users#edit |
| user\_path | GET | /users/:id(.:format) | users#show |
|  | PATCH | /users/:id(.:format) | users#update |
|  | PUT | /users/:id(.:format) | users#update |
|  | DELETE | /users/:id(.:format) | users#destroy |