Requirements

- Create a Ruby on Rails project (or start with an existing project) and improve the application’s usability and visual design. In addition to standard view templates, use layouts, stylesheets, partials, and helpers to implement the various improvements.

- Modify your application’s layout (located at `app/views/layouts/application.html.erb`) and add a header containing a title and site navigation links.

- Add a reasonable number of CSS styles to your application’s stylesheets (located at `app/assets/stylesheets`) to make your application shiny and attractive.

- Create and use at least one partial within your application to make your view templates DRY. For example, create a partial named `_form.html.erb` to contain your form template code, and render this partial within your existing `new` and `edit` view templates.

- Rails includes many helper methods to assist with rendering your views. Use at least two existing Rails helper methods within your views. Common Rails helpers include `date helpers`, `number helpers`, and `text helpers`.

- In addition to the helper methods included with Rails, you can write your own helper methods to encapsulate your custom view logic. Write at least two custom helper methods, and use each one in at least two different view templates.

- Add a confirmation message to at least one link in your views. For instance, it’s always a good idea to display an “Are you sure?” confirmation dialog when a user performs a destructive action, such as clicking a delete link.

- Using the `image_tag` helper method, add a few images to your views.

- Build a home page for your app. Create an empty controller named `WelcomeController` at `app/controllers/welcome_controller.rb` and create a new view template at `app/views/welcome/index.html.erb`. Next, tell Rails about your new home page by defining the root route. Open the routes configuration file at `config/routes.rb`, search for the commented line that begins with `root :to`, uncomment the line (remove the `#`), and modify it to specify your new controller/action: `welcome#index`. Finally, build your home page by adding content to the view template you created earlier.

Resources

- Use the example from class as a reference when creating your view templates, layouts, stylesheets, partials, and helper methods.

- If you’re new to CSS, read this introduction to learn what CSS is and how it works.

- For a list of CSS properties you may use to style your app, look here and here.

- `CSS Basics` is another great resource for learning all things CSS.

Submission

- Show your completed assignment to the instructor during class or office hours to receive credit.