

Project folder structure:

/ - Root Application Directory

app/ - Main content of application

config/ - Configuration files

vendor/ - Custom gems you've written/included

db/ - Database migrations, seed file, schema

lib/ - Additional custom code

test/ OR spec/ - Unit tests, Functionality/Integration tests

log/ - Development and Production logs

/Gemfile - List of gems to bundle with the application

/Gemfile.lock - List of gems and their versions being used in the application

/readme.md - Readme documentation for the project. Usually contains stack info and information the developer needs to know.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

app/assets - javascript, stylesheets, and images

app/controllers - controller files to coordinate views with models

app/models - model files to modify data being stored

app/views - view files to display data to the end user. Html, json, some javascript

app/views/layouts - The files located in layouts control the overall look and feel of the application

app/views/\*/ - Each folder within the views dir map to actions in a controller. Its imperative this has the same naming convention as the controllers.

example: pages (views dir), pages\_controller.rb (controller), page.rb (model)

app/helpers - helper methods to be shared across views

app/mailers - email templates

config/routes.rb - Routes for http GET and POST commands. Connects the views to actions in the controllers.

config/database.yml - Database connection information for Development, Test, and Production environments.

config/secrets.yml - Used for storing private keys to be used in your application.

While providing the basic structure of the application, I'm going to spend some time going over an example of what each team can expect.

For each team responsible for individual pages, the majority of what you contribute will be in your specific view folder,.A minority of your work can contain some assets if custom styling, javascript or hosted images are needed.

**Example**

RMI home page team:

Your html page is found at:

app/views/pages/home.html.erb

From here you can create partials within the pages view directory (app/views/pages/) that can be accessed within the home page if needed. Generally partials are html files that are generic enough to be applicable to multiple views. Code refactoring basically.

An example of partials is the \_form.html.erb file found within both the pages and links views directories.

\_form.html.erb is called in both the new, and edit views. Its used to create new pages/links and to update the information for existing pages/links.