

Abinet Kenore
CS390P
HW3
Fall 2018

This is the last part of the continuation on building up a “registration banner” and this time the primary goals are to add a navigation bar and an authentication via using a devise gem. In this part of the project I liked to go further, and I added a home page and footer for the all the views, and I also modified a default CSS to give the page nicer look and adjusted what it looks like in a different screen sizes using a bootstrap.

In this paper I will show you how you can add a home/static page/s on ruby on rails app. I mean when you start a server and route to <http://localhost:xxxx> instead of seeing a ruby on rails home page how you can see the home/landing page you created. I believe you already have some app and you are wondering to add a home page to it.

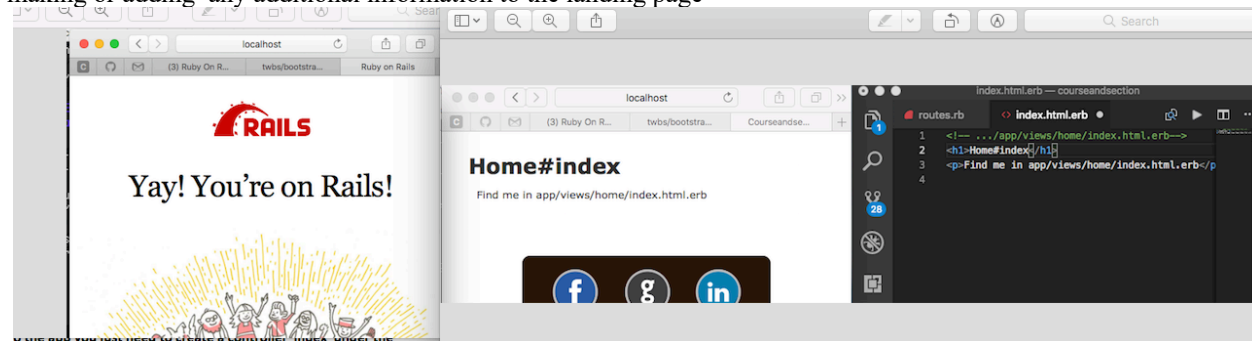
So in order to add a home page to the app you just need to create a controller `index` under the application you created as follow.

Rails g controller Home index

```
Abinets-MacBook-Pro:SCS abiken$ ls
CS390PHWpaper.docx  config          storage
Gemfile             config.ru       test
Gemfile.lock        db             tmp
README.md           lib            vendor
Rakefile            log           ~$390PHWpaper.docx
app                 package.json
bin                 public
Abinets-MacBook-Pro:SCS abiken$ rails g pageName index
```

Next Go to `../config/routes.rb` and add `root to: "home#index"`

Then the landing page `...localhost:3000` looks like default on (left) after modified (middle) what the code looks in view before making or adding any additional information to the landing page



Then visit the website start bootstrap(<https://startbootstrap.com>) and pick the them you liked and click view live and then inspect it on the on the browser and pick the part you are looking for OR view source code on GitHub and see the `index.html` file(I believe this would be the easiest way to get the snippet you are looking for (<https://github.com/BlackrockDigital/startbootstrap-grayscale>) . Also the alternative way would be just to visit the Bootstrap home page and see what's out there for you (<http://getbootstrap.com/docs/4.1/getting-started/introduction/>). After you copied a snippet which liked to add in to you app go to your `...app/views/layouts/application.html.erb` past it before the `<%= yield %>` then you are good modify the href's to the urls like:

```
<ul class="navbar-nav">
  <li class="nav-item active">
    <a class="navbar navbar-light" style="background-color: #2e8b57;"
href="http://localhost:3000/home/index">Home <span class="sr-only">(current)</span></a>
  </li>
```

Good thing about adding a navbar and other features to the layout/application.html.erb is it can be accessed by all the views you have and it makes your code DRY as well.

Hence I said, I added footer to the view let me explain you how you can do it. Under `application.html.erb` `<%= yield %>`

```

</li>
<%= yield %>
<!-- ../views/layouts/application.html.erb-->
<footer>
  <a href="http://facebook.com/abinet.getachew.1" class="social-icon">
    <%= image_tag("facebook.png", :alt => "rss feed") %>
  </a>
  <a href="https://github.com/abikg71" class="social-icon">
    <%= image_tag("github.png", :alt => "rss feed") %>
  </a>
  <a href="https://www.linkedin.com/in/abinet-kenore" class="social-icon">
    <%= image_tag("linkedin.png", :alt => "rss feed") %>
  </a>
  <a href="https://twitter.com/abikenore" class="social-icon">
    <%= image_tag("twitter.png", :alt => "rss feed") %>
  </a>
  <p class="copyright">Copyright 2018, My Q</p>
</footer>
</body>

```

The footer also applies to all the views you have.

Next I will explain to you how it is easy to add an authentication to your ror app using devise gem. At this point I am not going to explain you how you add the gems to the GemFile. You need how to add gems to your ror app read my HW2 paper.

However, when you are installing a devise gem (* <https://www.youtube.com/watch?v=zJYuLebl-Js>)

After you install devise gem you will see a success message

```

Bundle complete! 20 Gemfile dependencies, 85 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
Abinets-MacBook-Pro:SCS abiken$ rails g devise:install
Running via Spring preloader in process 801
  create  config/initializers/devise.rb
  create  config/locales/devise.en.yml
=====

```

Some setup you must do manually if you haven't yet:

1. Ensure you have defined default url options in your environments files. Here is an example of default_url_options appropriate for a development environment in config/environments/development.rb:

```

config.action_mailer.default_url_options = { host: 'localhost', port: 3000 }

```

In production, :host should be set to the actual host of your application.

2. Ensure you have defined root_url to *something* in your config/routes.rb. For example:

```

root to: "home#index"

```

3. Ensure you have flash messages in app/views/layouts/application.html.erb. For example:

```

<p class="notice"><%= notice %></p>
<p class="alert"><%= alert %></p>

```

4. You can copy Devise views (for customization) to your app by running:

```

rails g devise:views

```

```

=====
Abinets-MacBook-Pro:SCS abiken$ rails g devise:install

```

Then go to ... /views/layouts/application.html.erb add

```

<!-- Added -->
<p class="notice">
  <%= notice %>
</p>
<p class="alert">
  <%= alert %>
</p>
<li>
  <%= link_to 'Log out', destroy_user_session_path, method: :delete %>
</li>
<%= yield %>

```

```

Rails.application.routes.draw do
  devise_for :users
  get 'user/sign_up'
  get 'user/sign_in'
  root to: "home#index"
end

```

Add the following code to the controllers you are authenticating

```

before_action :authenticate_user!

```

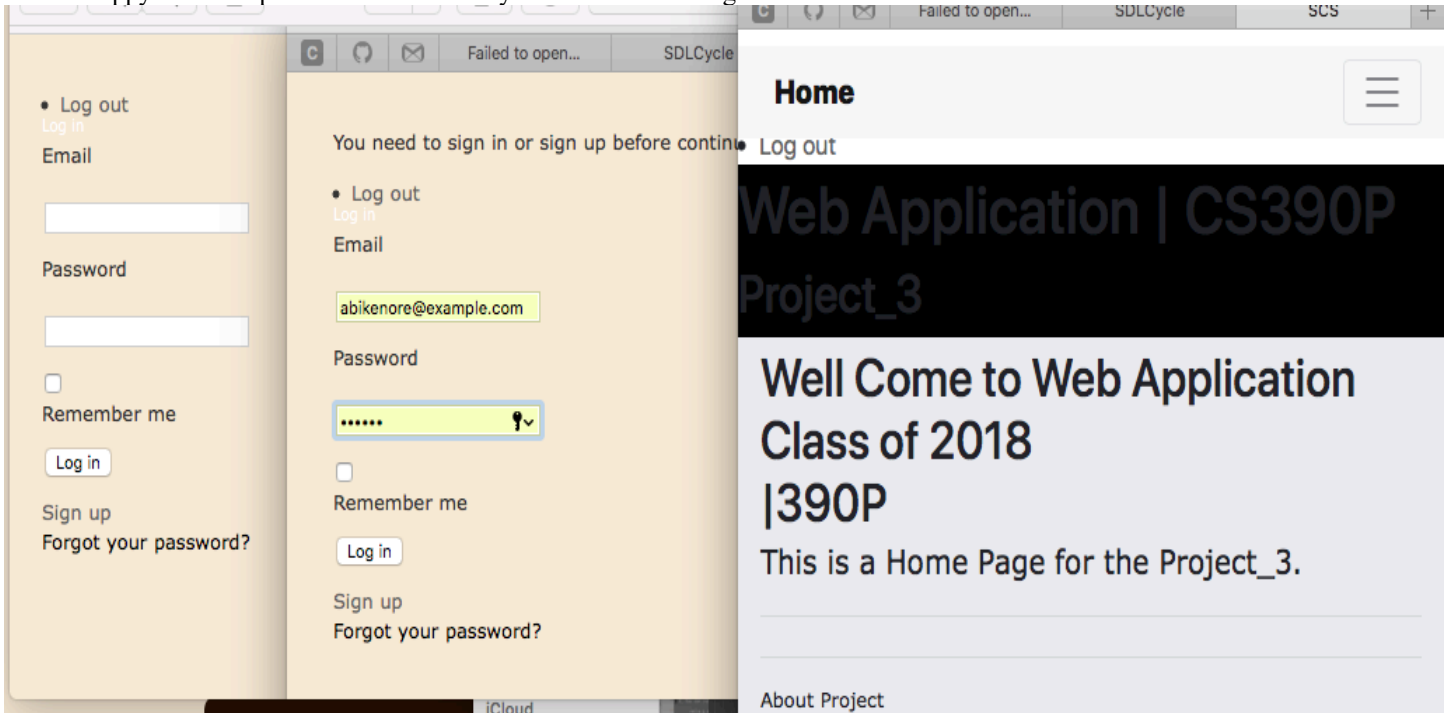
... /app/views/students/index.html.erb

```

<li>
  <%= link_to 'Log out', destroy_user_session_path, method: :delete %>
</li>

```

You can apply the same process to the other files you are authenticating



If you are interested to know how many users created go to your terminal or CMD

```
~/Documents/Fall2018/CS390P/WebAppF2018/HW3/SCS — ruby bin/rails c • fsevent_watch
Abinets-MacBook-Pro:SCS abiken$ rails c
Running via Spring preloader in process 2142
Loading development environment (Rails 5.2.1)
irb(main):001:0> User.count
(1.9ms) SELECT COUNT(*) FROM "users"
=> 1
irb(main):002:0> @user = User.first
User Load (0.4ms) SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ? [
["LIMIT", 1]]
=> #<User id: 1, email: "abikenore@example.com", created_at: "2018-10-25 16:56:08", upda
ted_at: "2018-11-02 20:17:00">
irb(main):003:0> "1", last_sign_in_ip: "1", created_at: "2015-01-
```

The above screenshot shows you that there is one user created and when it was created and the last time user logged in and out and more.

When you are signing up and if you don't fill the form correctly the devise gem will complain you and here is what it looks like.

The screenshot shows a web form with a light beige background. At the top left, there are links for 'Log out' and 'Sign up'. A prominent red error banner at the top states '2 errors prohibited this user from being saved:'. Below this, a list of errors is shown: 'Email can't be blank' and 'Password can't be blank'. The form fields are labeled 'Email', 'Password', and 'Password confirmation'. The 'Email' and 'Password' fields have red borders indicating they are required. The 'Password' field has a note '(6 characters minimum)'. A 'Sign up' button is located below the 'Password confirmation' field. At the bottom left, there is a 'Log in' link.

• Log out
Sign up

2 errors prohibited this user from being saved:

- Email can't be blank
- Password can't be blank

Email

Password
(6 characters minimum)

Password confirmation

Sign up

Log in

Methods/Measurements

I cloned my HW2 project and started modifying the project by adding the requirements for HW3. As stated above adding the navbar and devise gem were the main ones. Hence I already learned how to add a new gem's to the project it wasn't too bad to do the same thing. Also by watching few YouTube videos I was able to add a landing page to the views .

The big point which I leaned on this part of the project was how cool it to use a devise gem to add authentication instead of creating my own credentials.

Bibliography

Fomal

Dr. Beaty's PowerPoint slides, 08Authentication.pptx

Online

https://slackredir.net/link?url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DNf_Si8_szmM

<https://www.youtube.com/watch?v=idgRXCLDWG8>

<https://github.com/twbs/bootstrap-sass>

<https://blog.teamtreehouse.com/static-pages-ruby-rails#setup>

https://www.youtube.com/watch?v=Nf_Si8_szmM

<https://github.com/rails/rails/issues/22584>

<https://rubygems.org/gems/devise/versions/>

<https://www.youtube.com/watch?v=zJYuLebl-Js>