Rails Demo 4: Making things pretty with BootStrap

Bootstrap is a popular front-end library released by Twitter that makes it easy to add professional styling to our web app. This demo continues the blog app you wrote previously.

Step 1: First we need to install Bootstrap into our Rails app.

1. Add the following to your Gemfile

   ```
   gem 'bootstrap', '~> 4.0.0'
   gem 'jquery-rails'
   ```

2. Rename `app/assets/stylesheets/application.css` to `application.scss`
3. Edit `application.scss` :
   - Import Bootstrap: `@import "bootstrap";`
   - Import the other scss files: `@import "*";`
   - Remove all `*= require` and `*= require_tree` statements from this file. (Using `*= require` may interfere with your ability to use Bootstrap mixins.)
4. Edit `app\assets\javascripts\application.js` . Add the following lines above `//= require_tree .`

   ```
   //= require jquery3
   //= require popper
   //= require bootstrap-sprockets
   ```

5. If you are using CodeAnywere, do a "Save All"
6. Remove `Gemfile.lock`
7. Run `bundle install --without production`

Step 2: Show the various templates / components available with BootStrap:

https://getbootstrap.com/docs/4.0/examples/ https://getbootstrap.com/docs/4.0/components/alerts/

Step 3: Generate your home page controller.

```
rails generate controller Home
```

Make sure the controller has an index method:

```
class HomeController < ApplicationController
  def index
  end
end
```

Step 4: Create the home page template: `views/home/index.html.erb`.

Step 5: Add this route to `config/routes.rb`

```
root 'home#index'
```

Step 6: Now Let's add a jumbotron to our homepage!

1. Add the following to `views/home/index.html.erb`:

```
<div class="jumbotron">
  <div class="container">
    <h1>Welcome to my world of bees!</h1>
     <p>The happiness of the bee and the dolphin is to exist. For man it is to know that and to wonder at it.</p>
     <p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more &raquo;</a></p>
   </div>
 </div>
```

2. Add the folowing to `application.scss` after the `import` statements:

```
  body {
   padding-top: 50px;
   padding-bottom: 50px;
   margin: 0;
 }
```

3. Add the folowing to `home.scss` **

```
.jumbotron {
    background-image: asset-url('bees.jpg');
    background-size: cover;
    height: 400px;
}

.jumbotron h1 {
   color: #fff;
   text-align: center;
   margin-bottom: 30px;
   letter-spacing: -1px;
   font-weight: bold;
}

.jumbotron p {
   color: #fff;
   text-align: center;
   margin-bottom: 30px;
   font-weight: bold;
}
```

4. Add `bees.jpg` file to assets/images
   - You can get the bees image here: `https://bit.ly/2VbnAuO`
5. Stop the server; run `rake tmp:cache:clear` ; start the server.

Step 7: Nice! Now let's call out our most recent articles under the jumbotron

```
<div class="container-fluid">
  <!-- Example row of columns -->
  <div class="row">
    <div class="col-md-4">
      <h2>Heading</h2>
      <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. </p>
      <p><a class="btn btn-default" href="#" role="button">View details &raquo;</a></p>
    </div>
    <div class="col-md-4">
      <h2>Heading</h2>
      <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. </p>
      <p><a class="btn btn-default" href="#" role="button">View details &raquo;</a></p>
   </div>
    <div class="col-md-4">
      <h2>Heading</h2>
      <p>Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.</p>
      <p><a class="btn btn-default" href="#" role="button">View details &raquo;</a></p>
    </div>
  </div>

  <hr>

  <footer>
    <p>&copy; 2018 Company, Inc.</p>
  </footer>
</div>
```

Step 8: Let's replace this generated text with actual posts from our blog

In your home controller, implement the index method to do the query.

```
def index
    @posts = Post.last(3)
end
```

Step 9: Now render these posts using a similar HTML structure to that from Step 7

Replace the hardwired Heading divs from step 7 with this:

```
<% @posts.each do |post| %>
   <div class="col-md-4">
      <h2><%= post.title%></h2>
      <p><%=post.article.truncate_words(30)%></p>
      <p>
        <%= link_to 'Read more >', post, class: "btn btn-default"%>
      </p>
   </div>
<% end %>
```

Step 10: Now let's add a standard nav bar to our app throughout.

Create a new partial named `views/layouts/_navigation.html.erb` and place this code in it

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<a class="navbar-brand" href="#">Navbar</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav mr-auto">
  <li class="nav-item active">
    <a class="nav-link navbar-dark" href="/">Home <span class="sr-only">(current)</span></a>
  </li>
  <li class="nav-item">
    <a class="nav-link navbar-dark" href="/authors">Authors</a>
  </li>
   <li class="nav-item">
    <a class="nav-link navbar-dark" href="/posts">Posts</a>
  </li>
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle navbar-dark" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
      Other Sites
    </a>
    <div class="dropdown-menu" aria-labelledby="navbarDropdown">
      <a class="dropdown-item" href="https://www.gvsu.edu">GVSU</a>
      <a class="dropdown-item" href="https://www.cis.gvsu.edu">CIS</a>
      <div class="dropdown-divider"></div>
      <a class="dropdown-item" href="https://www.gatech.edu">Georgia Tech</a>
    </div>
  </li>
</ul>
<form class="form-inline my-2 my-lg-0">
  <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
  <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
</form>
</div>
</nav>
```

Step 11: Now invoke this partial in `views/layouts/application.html.erb`:

```
<body>
    <%= render 'layouts/navigation' %>
   <%= yield %>
</body>
```

Step 12: Let's spruce up our Author listing view.

Modify `views/authors/index.html.erb` to look like this:

```
<% if !notice.blank? %>
    <p id="notice" class="alert alert-success"><%= notice %></p>
<% end %>

<h1>Authors</h1>

<table class="table table-hover">
  <thead>
    <tr>
      <th class="w-25">First Name</th>
      <th class="w-25">Last Name</th>
      <th class="w-25">Email</th>
      <th class="w-25">Actions</th>
      </tr>
    </thead>
    <tbody>
      <% @authors.each do |author| %>
      <tr>
        <td class="w-25"><%= author.fname %></td>
        <td class="w-25"><%= author.lname %></td>
        <td class="w-25"><%= author.email %></td>
        <td class="w-25">
          <%= link_to author, class: "btn btn-default btn-sm" do %>
           Show
          <% end %>
          <%= link_to edit_author_path(author), class: "btn btn-default btn-sm" do %>
          Edit
          <% end %>
          <%= link_to author, method: :delete, data: { confirm: 'Are you sure?' }, cl
ass: "btn btn-danger btn-sm" do %>
          Delete
          <% end %>
        </td>
      </tr>
      <% end %>
    </tbody>
  </table>
  <br/>
  <%= link_to 'New Author', new_author_path, :class => "btn btn-primary" %>
```

Step 13: Let's make the form partial for authors pretty as well.

Notice specifically that we've added bootstrap classes to the various elements.

```
<%= form_with(model: author, local: true) do |form| %>
  <% if author.errors.any? %>
    <div id="error_explanation" class="alert alert-danger">
      <strong><%= pluralize(author.errors.count, "error") %> prohibited this author from being saved:</strong>

      <ul>
      <% author.errors.full_messages.each do |message| %>
        <li><%= message %></li>
      <% end %>
      </ul>
    </div>
  <% end %>
  <div class="form-group">
    <%= form.label :fname , "First Name"%>
    <%= form.text_field :fname, id: :author_fname, class: "form-control" %>
  </div>

  <div class="form-group">
    <%= form.label :lname, "Last Name" %>
    <%= form.text_field :lname, id: :author_lname, class: "form-control" %>
  </div>

  <div class="form-group">
    <%= form.label :email, "Email" %>
    <%= form.text_field :email, id: :author_email, class: "form-control" %>
  </div>

  <div class="form-group">
    <%= form.label :thumbnail, "Profile Image" %>
    <%= form.file_field :thumbnail, id: :author_thumbnail, class: "form-control-file" %>
  </div>

  <div class="actions">
    <%= form.submit class: "btn btn-primary" %>
    <%= link_to 'Cancel', authors_path, class: "btn btn-danger" %>
  </div>
<% end %>
```

Step 14: However, since we added a file upload type input, let's doctor our controller a bit so it can extract the file name.

Add these two lines of code at the top of the create and update methods:

```
thumb_upload = params[:author][:thumbnail]
params[:author][:thumbnail] = thumb_upload.original_filename
```

Step 15: Remove the buttons from `new.html.erb` and the `edit.html.erb`.