Homework 3: MVC View

For this homework, you will create a simple, static web page in Rails and practice using the basic version control features of Git.

You will do this homework as a team; however, each member of your team will be responsible for the completion of a particular task.

Step 1. Clone your team's project-skeleton repo

Rather than starting from scratch, your squad is going to start off with a project skeleton. Each team member must clone the skeleton repo, so that each has his/her own copy to work on.

To get going, do the following:

1. SSH into your Ubuntu VM, using the following command:

   ```
   $ vagrant ssh
   ```

2. Change directory (using cd) into the "workspace" folder.

3. Using your host OS's web browser, find your team's repo on GitHub via this link:

   https://github.com/memphis-comp7012/

4. Locate the "HTTPS clone URL" on your team's GitHub-repo page.

5. Back in VM, run the following command such that MYURL is your team's HTTPS clone URL:

   ```
   $ git clone MYURL
   ```

   This command creates a folder with the same name as your team's repo and that contains a working copy of the project code and a copy of the repo.

6. Change directory into this folder. A message from RVM should be displayed, indicating that RVM is working.

7. Install the gems (i.e., library dependencies) that the project needs by running the following command:

   ```
   $ gem install bundler
   $ bundle install
   ```

   This command installs all the libraries (gems) that you'll need to build and run the project.

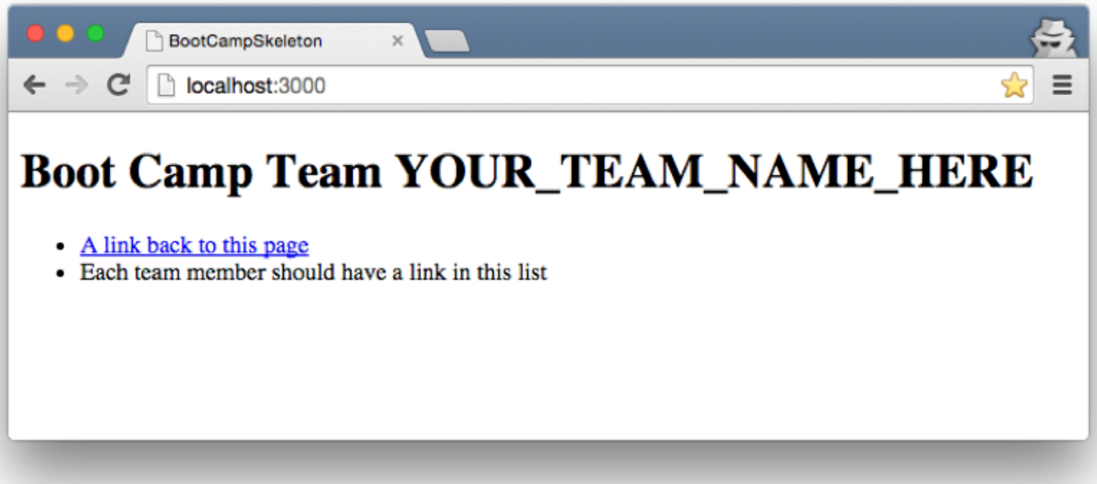
You can now test the skeleton by first starting the Rails server with the following command:

```
$ rails s -b 0.0.0.0
```

And then opening the following URL in a web browser in your host OS:

http://localhost:3000/

You should see a homepage that looks like this:



Boot Camp Team YOUR_TEAM_NAME_HERE

- A link back to this page
- Each team member should have a link in this list

Step 2. Create web pages

Each team member must create two web pages:

1. **Personal profile page.** The page should present fun info about yourself and your interests. Be creative in designing your page, but follow these constraints:
   - Your name must be displayed prominently.
   - The URL path must be your UofM username (e.g., http://localhost:3000/sdflming).
   - You must include a picture of yourself.

2. **Movie review page.** The page should present your review of a movie you've seen. Again, be creative, but follow these constraints:
   - The title of the movie must be prominently displayed.
   - Your score out of 5 stars must be prominently displayed.
   - A picture related to the movie must be included.
   - Some basic data about the movie must be included, such as (but not limited to) the director, the year of release, and the starring actors.
   - Text detailing your review.

Additionally, each of your pages must use each of the following HTML tags at least once (again, some creativity may be required on your part to work all of these in):

- `h1` – Level-1 heading
- `h2` – Level-2 heading
- `p` – Paragraph

- a – Hyperlink (with href)
- img – Image (hint: image files go in app/assets/images/)
- ul/li – Unordered list
- form – Input form containing the following elements:
    - button – Button (doesn't do anything when pressed)
    - datalist – Dropdown list (doesn't do anything other than let you choose an item)
    - input – Radio buttons (type="radio")

Your pages must use these tags in noticeably different ways (i.e., the two pages cannot look similar).

When you create your pages, add the appropriate routes, actions to StaticPagesController, and ERB views.

Step 3. Add links on the home page

You must add hyperlinks to your pages on the home page. This will enable visitors (and graders) to conveniently navigate to each team member's pages.

- The link text for the profile page should be "Profile for *Lastname*, *Firstname*".
- The link text for the movie review page should be "*Firstname*'s review of *Movie Name*".

One member of the team should update the homepage heading to include your team's name. He/she should also delete the example items/links, so that only your team's profile-page links remain.

Step 4. Submit (by tagging) your team's submission

Before you can submit, all team members must have merged their code into the master branch and pushed the updates to GitHub. If a team member does not complete his/her work on time, the rest of the team may submit without his/her contribution.

To submit your team's work, you must "tag" the current commit in the master branch:

```
$ git tag -a hw03v1 -m 'Tagged Homework 3 submission (version 1)'
$ git push origin --tags
```

To grade your work, I will check out the appropriate tag, and run it on my machine.

Note that if for some reason you need to update your submission, simply repeat the tagging process, but increment the version number (e.g., hw03v2, hw03v3, hw03v4, etc.).