## DEPLOY ROTTENPOTATOES AND ENHANCEMENT #1

***Part A —Add some movies to RottenPotatoes, and deploy it to the world***

We have a version of RottenPotatoes that has had some slight modifications for successful deployment on Heroku, and to which we've added an additional handful of movies to make things more interesting. Check out the app on your VM (or other development environment) by running the following commands:

```
git clone git://github.com/saasbook/hw2_rottenpotatoes.git
cd hw2_rottenpotatoes
```

The first command will pull down the latest version of the RottenPotatoes starter code, and the second will move you into the directory which now contains the code.

The next step is to install the necessary gems for the app. Do this by running

```
bundle install --without production
```

The `--without production` part of the command causes the installer to ignore the PostgreSQL gem for your local installation (since that gem will cause problems if you're using a development environment without PostgreSQL installed).

Note that we provided a seed file that seeds the database with a bunch of movies. Take a look at the code in `db/seeds.rb` to review how these work. When you're ready, run

```
rake db:migrate
```

to apply all RottenPotatoes migrations to your local database. Then run

```
rake db:seed
```

to seed your local database with the movies from the seed file.

Verify that you can successfully run the app by using the command

```
rails server
```

If this works properly, you should be able to interact with RottenPotatoes via a web browser as described in the book

and in lecture. Check that you can see a list of all the movies when you access 'localhost:3000/movies'.

Once the above steps have been verified on your development computer, it's time to deploy on Heroku. Create a free Heroku.com (http://www.heroku.com) account if you haven't already, and deploy RottenPotatoes there. Appendix A in the textbook has more information about this procedure, and Heroku has detailed help pages (https://devcenter.heroku.com/articles/quickstart) as well. (Note that your app's name, *something.herokuapp.com*, must be unique among Heroku apps. It's therfore unlikely that the name "rottenpotatoes" will be available. You must either choose a different name or you may just keep the default name Heroku chooses for you when you create a new app.)

Since this is the first deployment of this app on Heroku, its database will be empty. To fix this, *after* you've pushed your app to Heroku, run

```
heroku run rake db:migrate
```

to apply all RottenPotatoes migrations. In this case, we only have one migration file that creates the `Movies` table (which had already been applied when you got your VM). You will also need to run

```
heroku run rake db:seed
```

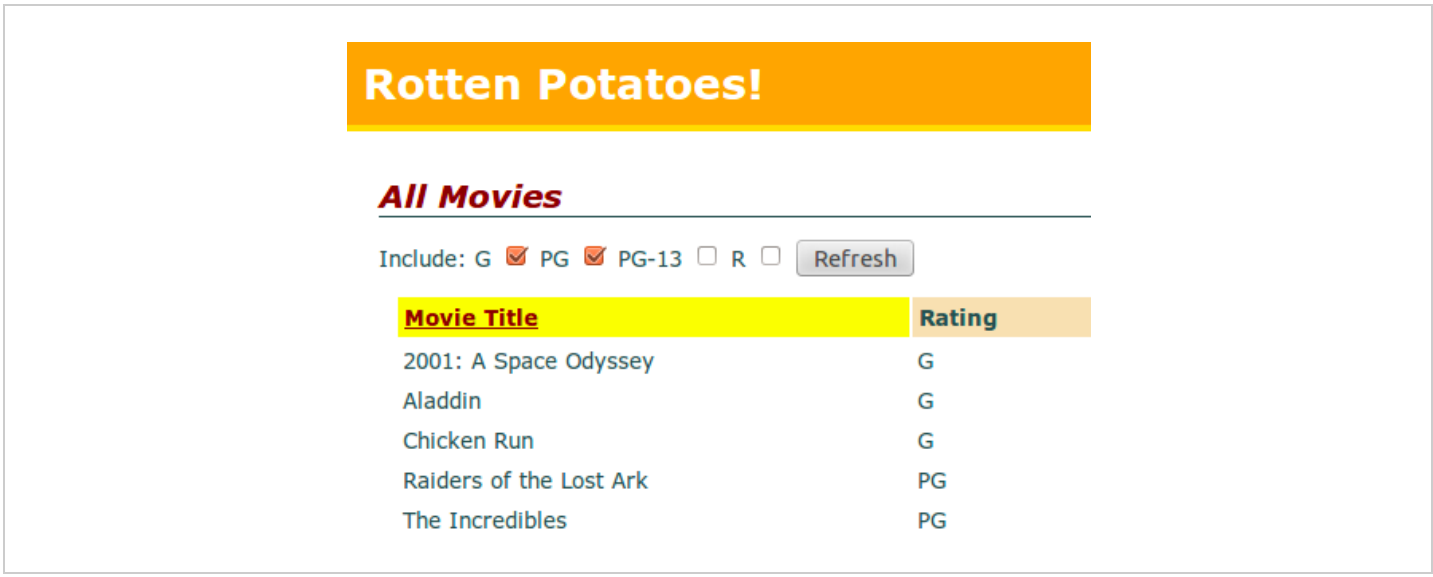
to seed your Heroku app's database with the movies we've created in `seeds.rb`.

Visit your site at *your-app-name.herokuapp.com/movies* to verify that it's working.

***Part B — RottenPotatoes Enhancement #1: Sorting the list of all movies***

You will enhance RottenPotatoes in the following ways:

- On the "All Movies" page, the column headings for "Movie Title" and "Release Date" for a movie should be clickable links. Clicking one of them should cause the list to be reloaded but sorted in ascending order on that column. Namely,
  - clicking the "Movie Title" column should list the movies alphabetically by title (for movies whose names begin with non-letters, the sort order should match the behavior of `String#<=>`), and
  - clicking the "Release Date" column heading should redisplay the list of movies with the earliest-released movies first.
- When the listing page is redisplayed with sorting-on-a-column enabled, the column header that was selected for sorting should appear with a yellow background, as shown below. You should do this by setting controller variables that are used to conditionally set the CSS class of the appropriate table heading to `hilite`, and pasting this simple CSS snippet (http://pastebin.com/zRkyVWZM) into RottenPotatoes's `app/assets/stylesheets/application.css` file.




***Important for grading purposes***

- The link (that is, the `<a>` tag) for sorting by "Movie Title" should have the HTML element id `title_header`.
- The link for sorting by "Release Date" should have the HTML element id `release_date_header`.
- The table containing the list of movies should have the HTML element id `movies`. (This has already been set for you by existing code.)

***Hints and advice***

- The current RottenPotatoes views use the Rails-provided "resourceful routes" helper `movies_path` to generate the correct URI for the movies index page. You may find it helpful to know that if you pass this helper method a hash of additional parameters, those parameters will be parsed by Rails and become available in the `params[]` hash.
- Databases are pretty good at returning collections of rows in sorted order according to one or more attributes. Before you rush to sort the collection returned from the database, look at the documentation for the ActiveRecord `find` and `all` methods and see if you can get the database to do the work for you instead.
- Don't put code in your views! The view shouldn't have to sort the collection itself; its job is just to show stuff. The controller should spoon-feed the view exactly what is to be displayed.

***Submission Directions***

You will submit a text file containing a single line with the URL of your heroku deployment. The url will look like this: `http://your-app-name.herokuapp.com`

Choose Files No file chosen

**Check**

Show Discussion

**New Post**