Homework 5: MVC Controller

For this homework, you will build upon the models from the previous homework to create an actual working web app—mainly by adding controllers and views. Moreover, you will continue to practice with the version-control system, 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.

Each team member will choose one task from the list below to complete. All team members must do a different task. If your team has only three members, then ignore Task 4.

The Tasks

For this homework, there will be four tasks (Tasks 1 through 4), each of which will build upon the corresponding task from Homework 4. As a reminder, here are the model classes and their associated tasks from Homework 4:

Task 1:

Planet
name : string color : string nasa_data : string diameter : integer

Continent
name : string area : integer population : integer description : text

Task 2

City
name : string website : string motto : string population: integer

Country
name : string info_hotline : string world_category : string year_formed : integer

Task 3:

Building
name : string email : string building_type : string stories : integer

Owner
first_name : string last_name : string ssn : string age : integer

Task 4:

Statue
name : string date_created : string medium : string weight : integer

Park
name : string park_type : string year_created : integer area : integer

In particular, each task must do the following.

Create Standard Controller Actions/Pages

Create 5 different seed-data objects for each model class (i.e., in `db/seeds.rb`) for a total of 10. Be sure to use the `create!` method (with a `!`) to create them, so you will get error messages if a validation error occurs.

Create two controllers, each corresponding to a model class. Each controller must have the standard controller actions/pages detailed in class:

- `index`
- `show`
- `new/create`
- `edit/update`
- `destroy`

The pages must have the features demonstrated in class, including inter-page links and notice/alert/error message handling. Be careful to follow the standard Ruby/Rails naming conventions.

As additional constraints, you must:

- Create your controller actions and view ERBs by hand (i.e., not using scaffold).
- Along those lines, your controller actions may not mention "json", nor may you use the `resource` method in `config/routes.rb`.

Create "Lucky" Actions

In addition to the standard actions/pages, you must do the following for each of your controller classes:

- Add an action "`lucky`" that randomly chooses a model object from the database and redirects the browser to the "show" page for that object. (Note that you are doing an HTTP redirect in this case and not rendering a view.)
- Add a route such that, if the controller class is FoosController, it routes from the URL pattern "/foos/lucky" to the "`lucky`" controller action. It should also have the prefix "lucky_foo".
- Add a hyperlink to the bottom of the "index" page that says "I feel lucky" and links to the "`lucky_foo_path`". You should use the `link_to` helper method to create this hyperlink.

Add Code Comments and Links to Home Page

IMPORTANT: Be sure to add your university username (e.g., mine is "sdflming") in a code comment at the top of your controller class files. (Otherwise, how will we know which work is yours?)

Furthermore, you must add hyperlinks to your controllers' index pages on the project home page. This will enable visitors (and graders) to conveniently navigate to each team member's pages.

How to submit your team's work

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, you 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 hw5v1 -m 'Tagged Homework 5 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., hw5v2, hw5v3, hw5v4, etc.).