Programming Assignment 2

For this assignment students will be adding an association and validations to the blog that they started in the previous assignment.

For help and more in-depth instructions on this assignment, please refer to the Module 3 Lectures 4 and 5 Videos.

This corresponds to iterations 2 and 3 of the blog app in the lecture videos. Please make sure to use the same names used in the videos for all Rails variables, classes, etc.. The grading scheme depends on you following the exact instructions provided in the videos.

1. Post "has many" Comments association

Create a "has many" relationship for posts to comments. The previous assignment already made sure the post_id was part of the comments table. For this part, you simply need to add the necessary relationships to the post and comment models. Be sure to add a destroy dependency to the comments relationship in the posts model (this makes sure that the comments associated with a post will get destroyed whenever a post is destroyed).

Then open up the rails console (rails c) to test this association. Here is the console input to test the association.

```
rails c
>> c = Comment.new()
>> p= Post.new()
>> p.comments << c
>> puts p.comments
```

2. Create nested resource

Create a nested resource for comments within posts in the routes file.

In the command line, run rake routes to test and view your new nested resources.

3. Add Validations

Please add the following database validations to your blog application:

- Validate presence of the Post's title
- Validate presence of the Post's body
- Validate presence of the Comment's post_id

- Validate presence of the Comment's body

4. Test and Push

Test the functionality of your app.

When you are satisfied with the app, push it up to your Bitbucket account under a public repository. Finally, follow the instructions on the assignment page to submit your Bitbucket URL to the automated grader queue.